UNIVERSIDAD
DE GRANADA

universität
uulm

User-centred Proactive Dialogue Modelling for Trustworthy Conversational Assistants

Programa de Doctorado en Tecnologías de la Información y la Comunicación

Fakultät für Ingenieurwissenschaften, Informatik und Psychologie

A doctoral thesis jointly supervised between the University of Granada and Ulm University and submitted in fulfilment of the requirements for the academic degrees of

Doctor (University of Granada)

and

Dr. –Ing. (Ulm University)

by

Matthias Peter Kraus

Supervisors: Prof. Dr. Dr.-Ing. Wolfgang Minker
Prof. Dr. Zoraida Callejas

2022

Editor: Universidad de Granada. Tesis Doctorales
Autor: Matthias Peter Kraus
ISBN: 978-84-1117-573-9
URI: https://hdl.handle.net/10481/77693

Abstract

The current wave of artificial intelligence and technological advancements has brought intelligent assistants into our daily lives. Such personal assistants help us with simple tasks, like providing news or weather information, smart home control, or entertainment using natural language. Despite their appraisal as intelligent entities, personal or conversational assistants are in general still stuck in the role of butlers and reactive bystanders that act upon commands.

For enhancing the cooperation capability of these systems and unfolding their technical competencies to the fullest, the integration of proactivity has become an emerging research topic in this area. Proactivity implies that technical systems, such as conversational assistants, possess the ability to detect a user's need for assistance and to initiate appropriate actions accordingly. Even though related work shows the potential benefits of proactive behaviour with regard to human-machine cooperation, the acceptance of proactive technology is still low due to an expectation gap between system behaviour and user requirements.

For closing this gap, we propose to equip proactive systems with proactive dialogue management in order to include the user in the system's decision processes and negotiate appropriate actions. However, how to computationally model timely and relevant proactive dialogue without giving the user the perception of being controlled or invading their privacy is an open question. Inappropriate proactive behaviour may have devastating effects on the cooperation and lead to diminished trust in the system which may compromise the acceptance of this technology. Therefore, this work aims at providing accepted and trustworthy proactive assistance by developing socially and task-effective dialogue models with the overall goal of improving the cooperation between humans and machines. For this, three major contributions are provided.

As the first contribution, we present a proactive dialogue model for human-machine cooperation. This concept builds upon two exploratory pilot studies observing the user perception of state-of-the-art approaches for inferring user and system requirements of proactive dialogue for application in cooperative contexts. Based on the outcome of the initial studies we conduct an requirement analysis and provide a taxonomy of proactive dialogue for cooperation. Here, we introduce proactive dialogue act types which represent different autonomy levels of proactive dialogue behaviour. Proactive dialogue in general is considered as the initiation of supporting dialogues for facilitating task execution. Besides, we propose a cognitive system architecture with the goal of implementing proactive dialogue in a technical system using methods of artificial intelligence and human-computer interaction.

As a second contribution, we present the design and evaluation of four user-centred proactive dialogue strategies based on the developed proactive dialogue model. Here, the goal is to provide an understanding of the effects of proactive dialogue design on the cooperation not only from a usability point of view but also from a social, user-centred perspective including a system's trustworthiness. For this, we develop and implement several conversational assistance prototypes, both low- and high-fidelity, that are capable of proactive dialogue. In laboratory and more realistic user studies, we shed light on the effects of proactive dialogue on a system's usability as well as human-computer trust dependent on task context, user characteristics, and state. These experiments allow to synthesise guidelines for the implementation of user-centered proactive dialogue management into cooperative conversational assistants.

As a third and last contribution, we fuse the gained understanding of the social impact of proactive dialogue for implementing user-centred proactive dialogue models with the goal of achieving trusted and task-effective conversational assistants for improving cooperation. In this regard, we provide findings considering the user expectations of user-adaptive proactive dialogue and the feasibility of utilising a trust measure for dialogue adaptation. For enabling statistically-driven adaptation methods, a proactive dialogue data corpus is collected and annotated with several features including trust. Based on the provided data, we advance the state-of-the-art for computationally modelling trust during conversational cooperation and present approaches for real-time prediction of trust during dialogue. Evaluation of the trust predictors shows the utility of our approach by achieving reasonable recall and accuracy. Trust prediction is then included in a conversational assistant for realising trust-adaptive proactive dialogue management. For dialogue management, we develop and implement a rule-based and reinforcement learning approach. The high trustworthiness and usability of trust-adaptive proactive dialogue management are proven in a user simulator study, for which a new socially aware user simulator has been developed.

In summary, we provide the first user-centred approach for integrating the concept of proactivity in human-computer dialogue. Here, we enhance the social awareness of artificially intelligent systems by equipping them with the ability to reason about their own trustworthiness during cooperation and adapt their proactive dialogue behaviour accordingly. Finally, this enables machines to provide more human-like and natural decision-making for appropriately assisting humans in complex task environments. This forms an important step on the way from mere conversational assistants to personal advisors.

Resumen

El auge actual de la inteligencia artificial y los avances tecnológicos ha propiciado que los asistentes inteligentes sean cada vez más frecuentes en nuestra vida cotidiana. Estos asistentes personales nos ayudan en tareas sencillas, como informarnos de las últimas noticias, escuchar el parte meteorológico, controlar elementos en un hogar inteligente o acceder a entretenimiento, todo mediante comandos en lenguaje natural. A pesar de ser considerados en muchas ocasiones entidades inteligentes, los asistentes conversacionales personales siguen en general estancados en el rol de mayordomos o espectadores reactivos que actúan en función de las órdenes que reciben por parte del usuario. Para mejorar la capacidad de cooperación de estos sistemas y desarrollar su potencial al máximo, la proactividad se ha convertido en un tema de investigación emergente. El que los sistemas conversacionales estén dotados de proactividad implica que tengan la capacidad de detectar las necesidades de sus usuarios y actuar en consecuencia. Aunque los trabajos relacionados con este tema han subrayado los beneficios que la proactividad ofrece para mejorar la cooperación hombre-máquina, la aceptación de la tecnología proactiva sigue siendo escasa debido a la diferencia entre el comportamiento del sistema y las expectativas de los usuarios. Para abordar esta problemática, proponemos dotar a los sistemas de una gestión proactiva del diálogo para incluir al usuario en los procesos de decisión del sistema y poder discernir de forma más adecuada cuáles son las acciones que se esperan. No se trata de una tarea obvia, pues surge el reto de modelar computacionalmente un diálogo proactivo oportuno en el contexto en el que se produce y relevante para el usuario, sin que éste tenga la percepción de ser controlado o de que se está invadiendo su privacidad. Un comportamiento proactivo inadecuado puede tener efectos devastadores en la cooperación y conducir a una disminución de la confianza en el sistema, lo que puede comprometer la aceptación de esta tecnología. Por lo tanto, este trabajo tiene como objetivo proporcionar asistencia proactiva aceptada y fiable mediante el desarrollo de modelos de diálogo que mejoren la cooperación entre humanos y máquinas. Con tal fin, se ofrecen tres contribuciones principales. Como primera contribución, presentamos un modelo de diálogo proactivo para la cooperación hombre-máquina. El concepto presentado se basa en dos estudios piloto exploratorios en contextos de cooperación hombre-máquina, en los que se observa la percepción que tienen los usuarios de los enfoques más avanzados para inferir sus expectativas y necesidades y gestionar en consecuencia el diálogo de forma proactiva. A partir de los resultados de estos estudios, realizamos un análisis de estos requisitos y proporcionamos una taxonomía de diálogo proactivo cooperativo. A continuación, introducimos tipos de actos de diálogo proactivo que representan los diferentes niveles de autonomía del comportamiento de diálogo proactivo, entendiendo éste como la iniciación de diálogos de apoyo para facilitar la ejecución de tareas. Además, proponemos una arquitectura de sistema cognitivo con el objetivo de implementar el diálogo proactivo utilizando métodos de inteligencia artificial y de interacción persona-ordenador. Como segunda contribución, presentamos el diseño y la evaluación de cuatro estrategias de diálogo proactivo centradas en el usuario y basadas en el modelo de diálogo proactivo desarrollado. En este caso, el objetivo es comprender los efectos del diseño del diálogo proactivo en la cooperación, no sólo desde el punto de vista de la usabilidad, sino también desde una perspectiva social centrada en el usuario, incluyendo entre otros aspectos la confianza que éste deposita en el sistema. Para ello, desarrollamos e implementamos varios prototipos de asistencia conversacional con capacidad proactiva que hemos evaluado en entornos de laboratorio con distintos tipos de tareas cooperativas, tipos de usuario y estados de los mismos. Estos experimentos han permitido identificar directrices para la implementación de la gestión proactiva del diálogo centrada en el usuario para asistentes conversacionales cooperativos. Como tercera y última contribución, tenemos en cuenta la comprensión obtenida del impacto social del diálogo proactivo para implementar modelos de diálogo proactivo centrados en el usuario que permitan desarrollar asistentes conversacionales fiables y eficaces. En este sentido, aportamos conclusiones que tienen en cuenta las expectativas de los usuarios sobre el diálogo proactivo adaptado y la viabilidad de utilizar medidas de confianza para adaptar el diálogo. Para la utilización de métodos de adaptación estadísticos, se ha recolectado un corpus de diálogos proactivos que ha sido anotado con diversas características incluyendo la confianza. Basándonos en estos datos, avanzamos el estado del arte del modelado computacional de la confianza durante la cooperación a través de diálogos, y presentamos diversos enfoques para la predicción dinámica de la confianza durante el transcurso de la conversación. La evaluación de los predictores de confianza muestra la utilidad de nuestro enfoque alcanzando tasas de acierto apreciables. La predicción de la confianza se imbuye en el sistema conversacional durante la gestión proactiva del diálogo, que hemos implementado utilizando enfoques basados en reglas así como basados en aprendizaje automático. La pertinencia y usabilidad del gestor proactivo del diálogo adaptable a la confianza se ha demostrado mediante un estudio con simuladores de usuario, para el cual se ha desarrollado un simulador de usuario que considera diversos parámetros del diálogo social. En resumen, aportamos un enfoque novedoso centrado en el usuario para integrar el concepto de proactividad en el diálogo hombre-máquina. En este caso, mejoramos los sistemas conversacionales dotándolos de la capacidad de razonar sobre la confianza que inspiran en el usuario durante la cooperación y de adaptar su diálogo proactivo en consecuencia. Esto permite a las máquinas proporcionar una toma de decisiones más natural y parecida a la humana, asistiendo adecuadamente a los usuarios en la resolución de tareas complejas que requieren cooperación y asistencia. Esto constituye un paso importante en el camino para transformar los sistemas conversacionales de meros transmisores reactivos de información en verdaderos asesores personales.

Acknowledgements

For a successful completion of this thesis, various people have provided valuable support. First and foremost, I want to thank my supervisor Prof. Wolfgang Minker (Ulm University, Germany) for his dedicated support and belief in my work over the last years. Without his constant efforts and advice, none of this would have been possible. Also, I want to express my highest gratitude to my Joint-PhD supervisor Prof. Zoraida Callejas (Granada University, Spain) for her interest in my work and very helpful feedback. During our visits to Granada, here vast knowledge helped to gain new insights and helped to shape the focus of this work. In this regard, I further would like to thank my tutor Manuel Noguera and also David Griol (both Granada University, Spain) for their interest in my research. In addition, much appreciation goes to Elisabeth André (University Augsburg, Germany) for reviewing this thesis.

University and working life would not have been same without my awesome (former) colleagues. I had the pleasure to work with so many talented and nice people who have been always ready for offering a sympathetic ear. Here, I want to thank Annalena Aicher, Oleg Akhtiamov, Denis Dresvyanskiy, Dmitrii Fedotov, Isabel Feustel, Danila Mamontov, Juliana Miehle, Louisa Pragst, Niklas Rach, and Sabine Wieluch. A very special thanks goes to Nicolas Wagner for the extraordinary fruitful collaboration and sharing so many great moments. I learned a lot from our discussions and always appreciated your opinion and important contributions. Sharing an office with you made this experience twice as fun and it felt great to have a good friend as colleague.

Further, I appreciate the support from the staff at the Institute of Communications Engineering at Ulm University. Especially, I want to thank Werner Teich, Fe Hägele, and Michaela Baumann for their admistrative expertise and kind support. Additionally, I am grateful for my former Bachelor and Master students, namely Prince Attrams, Diana Betancourt Lopez, Viktoria Dettenhofer, Jianping Dong, Philipp Dörzenbach, Fabian Fischbach, Timo Häge, Pascal Jansen, Christoph Kunder, Ron Riekenbrauck, Philip Seldschopf, Tibor Tonn, and Nico Untereiner.

Finally, but most importantly, I would like to thank my friends and family. My friends have provided me with the necessary distraction for accomplishing this step in my life. My parents constantly supported and cared for me through the ups and downs of this thesis and helped me to focus on the relevant aspects of life. In addition, I want to thank my brother and my sisters for keeping me grounded and their significant influence on me for pursuing this degree.

Contents

List of Figures

List of Tables

List of Abbreviations

AI artificial intelligence

AIML artificial mark-up language

ANN artificial neural network

ANOVA analysis of variance

API application program interface

ASR automatic speech recogntion

AU action units

BDI belief-desire-intention

BFI big-five-inventory

CA conversational assistant

CNN convolutional neural networks

DAMSL dialog act markup in several layers

DIY do-it-yourself

DM dialogue management

DQN deep-q-network

DS dialogue system

eA extended accuracy

ECL extraneous cognitive load

FACS facial action coding system

GCL germane cognitive load

GRU gated recurrent unit

GUI graphical user interface

HCI human-computer interaction

HCT human-computer trust

HHI human-human interaction

HMI human-machine interaction

HRI human-robot interaction

HTML hypertext markup language

HTN hierarchical task network

HTTP hypertext transfer protocol

ICL intrinsic cognitive load

IG information gain

IP interface-proactivity

IQ interaction quality

JSON javascript object notation

KL Kullback-Leibler

LoA level of autonomy

LORA level of robot autonomy

LSTM long short term memory

LUIS language understanding intelligent service

MC monte carlo

MDP markov decision process

MFCC mel-frequency cepstral coefficients

ML machine learning

MLP multi-layer perceptron

MSE mean squared error

NARS negative attitudes towards robots scale

NLU natural language understanding

OWL W3C web ontology language

P precision

PANAS positive and negative affect schedule

POMDP partially observable decision process

RL reinforcement learning

RNN recurrent neural network

SASSI subjective assessment of speech system interfaces

SSL secure sockets layer

SUS system usability scale

SVM support vector machine

TD temporal difference

TOM theory of mind

UAR unweighted average recall

UEQ user experience questionnaire

WoZ wizard-of-oz

XGB extreme gradient boosting

1. Introduction

Significant technological advances in engineering and information technology have resulted in computers and technical devices becoming an integral part of our daily lives. One of the main reasons for this development is that interaction with technical systems is as easy as ever before. In the past, the necessity of learning a special communication language, e.g. text- or touch-based commands, was focal for operating computers or machines. More recently there has been a paradigm shift towards a more intuitive interaction using natural language. This transition is most prominently visible considering the rise of personal chatbot assistants like Apple's SIRI or Microsoft's CORTANA, and voice-activated smart speakers like Amazon's ALEXA or GOOGLE HOME. These devices all have in common, that they allow users to formulate their needs in their language and have the ability to respond or act accordingly. In doing so, even technical inexperienced users are capable of efficiently solving various tasks through human-machine collaboration, e.g. restaurant search (Henderson et al., 2014) or smart home control (Sciuto et al., 2018). Thus, the capability to understand natural language and reason about an appropriate system response for contributing to a specific user goal turns machines into conversational assistant (CA).

We use the term CA in this thesis for a computer system that is able to provide assistance for specific tasks using natural language. The overarching term in research and industry for intelligent systems that are able to converse with humans is (spoken) dialogue system (DS). Generally, a DS is a user interface allowing interaction with a computer application in a dyadic manner, i.e. both the user and the computer participate in a conversation by taking turns. The development of DSs has been a popular research topic for over half a century. One of the earliest well-known examples is ELIZA (Weizenbaum, 1966), which imitates a Rogerian therapist using pattern matching. ELIZA had no real understanding of language, but only reformulated a user's utterances or relied on predefined phrases. Since then, the development of dialogue systems has taken giant leaps. The formalisation of speech and dialogue enabled machines to understand and interpret language (Allen and Perrault, 1980; Searle, 1969). This in turn allowed to endow computer systems with a decision logic for adequate response generation. How DSs make decisions has evolved greatly over the last decades, beginning with rather rigid rule- and plan-based approaches (e.g. see Allen et al. (1995)) to flexible data-based approaches utilising sophisticated methods of machine learning (ML) (e.g see Young et al. (2013)). The performance and versatility of contemporary realisations of DSs let machines sound more and more natural and human-like. While ELIZA was identified easily as a computer, recent systems are harder to distinguish from humans. For example, Google's DUPLEX which can perform well-structured tasks on user-demand, such as restaurant reservations, is able to sound very similar to a human.

For this reason, there exist opinions which propose to let the system identify itself as a machine in order to not fool its interlocutor (O'Leary, 2019). However, sounding like an intelligent being does not necessarily imply helpful intelligent behaviour.

1.1. Motivation

Humans see computers to some extent as social actors, and therefore apply social rules during communication with them (Nass et al., 1994; Nass and Moon, 2000; Nass and Yen, 2012). As a result, humans put similar expectations to interactions with computers as with real persons. For example, having a naturally sounding voice makes users think that computers seemingly possess intelligence.

However, if a CA only acts reactively during the interaction and stays on standby, this may lead to a mismatch in the users' expectations about how such an intelligent entity should be assisting. Consequently, there exists a "gulf" (Luger and Sellen, 2016) between expectations on CAs and their capabilities implicating that they are still rather perceived as tools and not as true collaboration partners (Clark et al., 2019; Zamora, 2017). Furthermore, despite their ability to sound natural and being advertised as "intelligent", CAs are mostly applied for rather simple tasks by human standards. For example, communication with smart speakers is primarily based on one-shot interactions, e.g. "Alexa/Siri/Google, what's the weather today?". These systems only allow a few follow-up questions and are required to be invoked using command-based language. Also in academia, task-oriented systems are mainly restricted to reactive assistance such as question answering (e.g. see Weston et al. (2015); Guo et al. (2018)) or form-filling interactions (e.g. see Williams et al. (2014); Bordes et al. (2016)). In form-filling, also known as slot-filling interactions, the system step-wisely collects the required user information for fulfilling a specific task, e.g. requesting information about the type of food and price range in a restaurant search for querying a database (Henderson et al., 2014). Although the system usually has the initiative in form-filling dialogues, it acts reactively as it only serves as an information source and does not actively influence a user's decision-making. Even Google's Duplex does not truly collaborate with a user, as it acts in the name of the user, like a middle man, exactly as it was told to do. Thus, they act as butlers or bystanders, but not as equal partners in cooperation. For this reason, current CAs might not yet be able to escape their narrow, non-complex, and restricted task domains that require a little trust or human logic and have negligible consequences in case of system failure (Zamora, 2017).

To turn CAs into truly helpful agents also in more complex task environments where consequences of failure are more severe, computers need to be equipped with several types of intelligence besides technical functionality (Chaves and Gerosa, 2021). Concerning conversational systems, *conversational intelligence* seems to be of utmost importance for leveraging CAs to the next level. Chaves and Gerosa (2021) describe a chatbot's conversational intelligence as the ability "to actively participate in the conversation and to demonstrate awareness of the topic discussed, the evolving conversational context, and the flow of the dialogue". In conclusion, this type of intelligence enables systems to become better cooperation partners by contributing more actively to the interaction.

Some social characteristics related to conversational intelligence are *proactivity*, *conscientiousness*, and *communicability* (Chaves and Gerosa, 2021). A conversational system's ability to convey to users its internal features and interactive principles can be labelled as communicability. Conscientiousness is a system's capability to show understanding of the conversational context and to reasonably interpret user input (Morrissey and Kirakowski, 2013). Proactivity is manifested by a system's capacity to engage in a conversation autonomously, suggest topics, or provide further information. Proactivity seems to be the driving force for expressing a system's conversational intelligence, as both conscientiousness and communicability, are supported by adequate proactive conversation (Chaves and Gerosa, 2021). Accordingly, equipping CAs with proactive dialogue may be the key to improving cooperation and fostering user acceptance.

For getting a clearer idea of the concept of proactivity, consider the following example of proactive dialogue in human-human interaction (HHI). Primarily, proactive dialogue occurs in various situations that require two or more humans to cooperate. For illustration, we provide a dialogue between a guest and waiter with the cooperative goal of ordering food and drinks at a restaurant:

GUEST: Good evening, we would like to order dinner, please!
WAITER: Yes, of course. What would you like for starters?
GUEST: Let me see...
WAITER: I suggest tomato soup. May I serve you some?
GUEST: Yes, please.
WAITER: Ok, what do you want as the main dish?
GUEST:: Hm, just a second... what do you suggest?
WAITER: The "Coq au vin" is great. Do you want to order?
GUEST: That sounds good. But I'm vegan so I'll take the pumpkin gnocchi.
WAITER: Good choice. I'll serve a dry white wine with that.
WAITER: I can offer you some vegan dishes for dessert.
GUEST: Alright, what do you recommend?
WAITER: ...

Here, the cooperation starts with deciding on starters. After the user hesitates the waiter becomes proactive and suggests a dish without being asked. Afterward, the guest has to decide on the main dish. Here, the waiter stays reactive and waits for the user to make a move. The decision on which drink to take is taken by the waiter on behalf of the guest, using his context knowledge that dry white wine is suitable for this kind of food. For deciding on desserts, the waiter takes into account his knowledge about the particular guest and notifies him that vegan dishes for dessert are available. As can be seen from the example, proactive dialogue actions may take different types including notifications, suggestions, taking actions on behalf of the user, or simply staying reactive. Which type of action is appropriate in a given situation is quite complex and may depend on the context, specific user information, the user state, and the relationship between the cooperation partners. The benefits of adequate proactive dialogue expressed by the waiter, however, are more clear: it facilitates the cooperation process, as the waiter guides the guest in the food ordering process which may lead to faster and more successful task completion. In addition, if the guest perceives the waiter's behaviour to be trustworthy, it can increase the guest's trust in the waiter and the establishment, which may, in turn, increase the probability of the guest returning to the restaurant in the future.

Similarly, recent research has shown the benefits of proactivity in conversational systems. Considering human-chatbot interaction, Chaves and Gerosa (2021) identify five major benefits of proactive behaviour: To provide additional, useful information for increasing a chatbot's naturalness and the user's enjoyment of the interaction; To motivate users and keep the conversation alive by suggesting new topics and disclosure about the system's knowledge; To naturally recover the chatbot from failure; To improve conversation productivity in task-oriented dialogue by improving task efficiency; To guide and engage users for facilitating decision-making or learning. In interactions with robots and voice assistants, proactive dialogue has also been found to positively contribute to a system's helpfulness (Peng et al., 2019), usefulness (Schmidt and Braunger, 2018), and user satisfaction during collaborative tasks (Baraglia et al., 2016).

The intriguing question, however, is how to engineer and model a CA's proactive dialogue capacities for taking advantage of these benefits. As Meurisch et al. (2020) point out, the adoption and acceptance of proactive technology are still low due to a mismatch between system operation and user expectations. This mismatch is mostly due to issues concerning the system's *timing and relevance* of proactive actions, the user's *perception of being controlled*, and *privacy* (Chaves and Gerosa, 2021). Timing and relevance of proactive actions are crucial for the system to avoid being obtrusive and getting the user out of the task flow (McFarlane and Latorella, 2002). For example, Portela and Granell-Canut (2017) found that a system taking the initiative in an untimely manner was perceived as annoying and harmed engagement. Additionally, poorly timed interruptions can be perceived as disruptive (Chaves and Gerosa, 2018; Liao et al., 2016). As a result, users may experience higher frustration and require an increased mental effort for situation comprehension (Adamczyk and Bailey, 2004). This ultimately leads to distrust and thus non-acceptance of the system (Muir, 1994).

Closely related to the previous challenge is the perception of being controlled by proactive interventions. The arguably best negative example for this case is Microsoft's former office assistant CLIPPIT. The proactive assistant developed during the LUMIÈRE project (Horvitz, 1998) was intended to provide suggestions for a better or easier task execution, e.g. writing of a letter. However, CLIPPIT interrupted users at inappropriate moments during task execution, providing non-helpful assistance, behaving highly obtrusively (Bickmore and Picard, 2005). This culminated in the rejection of the assistant which has even been named one of the "50 worst inventions of all time" (Times, 2010). Finally, privacy issues with a system's proactivity are a concern, as such behaviour is often deemed too intrusive and perceived as an act of user surveillance (Meurisch et al., 2020). For example, Duijvelshoff (2017) reported that a proactive chatbot that was integrated into a work-related group chat resulted in privacy concerns. Users thought that the chatbot represented their superiors' interests which may result in disengaged and discomforted users.

The state-of-the-art research currently provides little knowledge about how to model timely and relevant proactive dialogue for overcoming these shortcomings. Consequently, state-of-the-art proactive conversational systems lack reliability and competence resulting in impaired user trust, which is crucial during cooperation with a technical system (Muir, 1994; Parasuraman and Riley, 1997). Therefore, the main research goal of this thesis is to improve cooperation between humans and computers by developing a sound proactive dialogue model for CAs. As user trust and the system's task effectiveness are fundamental aspects of successful cooperation, the CA's proactive dialogue behaviour needs to be trustworthy and show high usability. For achieving this goal, it is necessary to understand proactivity not only from a technical point of view but also from a social, user-centred perspective. Thus, it needs to be investigated how proactive dialogue is perceived by users, especially concerning perceived trustworthiness and usability. Furthermore, humans are highly individual beings possessing different characteristics manifested by their personality, age, gender, knowledge, experiences, preferences, moral attitudes, etc. Therefore, the usability and trustworthiness of proactive system actions may differ depending on the specific user. Additionally, the perception of proactivity is also subject to change as interaction takes place in a dynamic environment. For this reason, when developing proactive systems it is necessary to consider not only user-specific features but also the current situation, i.e. contextual properties related to a specific task (task type, task complexity, task progress), the dialogue, (type of user or system action or interaction length), and the user's current situation (user activity, emotional or affective state).

Technically, the main challenge is to implement and optimise adequate proactive dialogue strategies. In this context, adequate implies a conversational system's ability to convey its assistance behaviour competently and reliably avoiding disruptive and obtrusive system interventions. Therefore, we implement various proactive dialogue strategies and evaluate their impact on cooperation. Using the knowledge gained from the effects of the user-centred proactive dialogue strategies on the cooperation, we develop a user model allowing for real-time measurement of the impact of proactive dialogue on user trust. We propose to implement this user model into a dialogue system to endow it with the ability to reason about the trustworthiness of its actions. Such awareness is then used to augment a system's capability to tailor its proactive dialogue accordingly to improve cooperation.

In summary, we will provide novel work regarding user-centred proactive dialogue modelling, the development of proactive dialogue strategies, and how to equip a proactive system with self-awareness of its trustworthiness. Furthermore, we will propose machine-learning methods of using this information to influence a system's decisions on whether to become proactive and to what extent. The research contributions of the work described in this thesis are presented in the next section.

1.2. Contributions

To improve the cooperation between humans and machines by the means of trusted and task-effective proactive dialogue modelling, our research contributions may be divided into three parts:

1. Proactive dialogue modelling for human-machine cooperation
2. Design of user-centred proactive dialogue strategies and their effects on cooperation
3. Implementation and evaluation of user-centred proactive dialogue strategies for trustworthy CAs with high usability

As a first step, we carry out exploratory work in the form of two pilot studies conducted in the wild to establish an intuition of the user perception of current proactive interaction approaches considering usability and human-computer trust (HCT). Further, the results of these studies are used to distill user as well as system requirements for the application of proactive dialogue for human-machine cooperation. Based on the outcome of the exploratory studies and related work, we contribute a novel proactive dialogue model for CAs. Here, we define cooperation processes as dialogues, where supporting sub-dialogues initiated by the system during task execution are conceptualised as *proactive dialogues*. Further, we introduce four different proactive dialogue act types representing different degrees of autonomous system behaviour. For realising proactive dialogue, we contribute a novel cognitive system architecture combining artificial intelligence (AI) and human-computer interaction (HCI) methods.

To understand the effect of the proactive dialogue model on cooperation, we present four novel approaches to proactive dialogue design, implement them in CA prototypes, and provide evaluations with a focus on their impact on trustworthiness and usability. In two studies under laboratory conditions, we shed light on the relations between the different degrees of proactive dialogue and the HCT relationship. Particularly, novel insights are gained concerning the application of proactive dialogue dependent on the task context as well as different user states. Additionally, we deepen the understanding of the influence of various user characteristics, including personality, gender, technical affinity, and domain expertise, on the perception of proactive dialogue. Besides its impact on trust, also results for the task effectiveness of different levels of proactive dialogue are reported. In two experiments in realistic task scenarios under less restricted conditions, we show the portability of our concept in sophisticated prototypes for real-world application and confirm the results from our laboratory studies. Here, we also provide evidence of the applicability of the developed cognitive architecture for enabling CAs to conduct a proactive dialogue.

Finally, we implement novel approaches for the development of user-centred proactive dialogue management (DM). For this, two novel evaluation frameworks utilising an interactive video method and a serious dialogue game approach are developed. The interactive video method is used to validate HCT as adaptation criteria for implementing user-centred proactive dialogue. The serious dialogue game approach is integrated into a novel data collection set-up and used for the creation of a new proactive dialogue data corpus annotated with several features including trust. Based on the data corpus, we then introduce a novel user model that allows the prediction of the CA's trustworthiness online during dialogue. The user model is implemented using ML-algorithms and proved to provide reliable predictions for including trust in the dialogue state. This enabled the implementation of novel trust-adaptive proactive dialogue behaviour. In this regard, we present a rule-based strategy and a statistically-driven strategy using reinforcement learning (RL). For realising the statistically-driven strategy, we provide a novel approach for including trust and task metrics in a reward function. In addition, we present a new corpus-based user simulator for training the RL-based DM module and to test different proactive dialogue strategies in user simulation. We further present a user simulator study comparing adaptive and static proactive dialogue strategies. Here, we provide novel results and evidence that particularly RL-based trust-adaptive proactive dialogue strategies are promising for improving the cooperation with CAs.

In summary, this work has three main contributions for improving the state-of-the-art in cooperation with CAs. Firstly, we advance the proactive capabilities of a CA by providing a structured proactive dialogue model comprising a new taxonomy and cognitive architecture for its sound realisation. Secondly, we improve the proactive dialogue design by gaining an in-depth understanding of the relations between user, context, and dialogue as well as the effects of proactive dialogue strategies on the system's trustworthiness and usability during cooperation. Finally, we improve the machine's anticipation of the need of proactive system behaviour by fusing user-, context-, and dialogue information for a more socially and task-intelligent decision-making on whether to act reactively or in leveled proactive ways. The work described in this thesis has been carried out within a Joint-PhD programme between Ulm University and Universidad de Granada.

1.3. Outline

The structure of this work is as follows: In Chapter 2, we summarise and describe all relevant backgrounds that this work is based on to provide a common ground of knowledge and understanding for this work.

In Chapter 3, we provide a comprehensive literature review on the research activities in the domain of proactive human-machine interaction (HMI), and user-centred DS.

In Chapter 4, we present two pilot studies for gaining insights on how state-of-the-art proactive interaction behaviour can be transferred into the dialogue domain and how this influences the user perception regarding human-machine cooperation.

In Chapter 5, we describe the development of our proactive dialogue model including a user and system requirement analysis based on the previous exploratory studies and related work, a taxonomy for proactive dialogue, and a cognitive architecture for implementing proactive dialogue in CAs.

Based on the developed proactive dialogue model, we describe the design of four novel user-centred proactive dialogue strategies and their impact on the cooperation in Chapter 6. Here, the aim is to determine the selection of the appropriate level of proactive dialogue, i.e. proactive dialogue act type, dependent on the specific user type and context information for improving the human-computer cooperation by increasing a system's trustworthiness and usability. Further, we confirm the portability of our approach to realistic environments and the utility of the designed cognitive architecture.

In Chapter 7, we subsequently describe the implementation of user-centred proactive DM comprising an investigation of the applicability of trust as a measure for dialogue adaptation, the development of a trust recognition module, the creation of a socially-aware user simulator, and the development of trust-adaptive proactive dialogue strategies. Further, we present user simulator studies showing the benefits of our approach for achieving the stated goals of this thesis. Finally, we conclude this thesis by reviewing our research contributions and providing promising future research areas in Chapter 8.

1.4. Note on Copyrighted Material

Parts of this work have been contributed to scientific publications. Accordingly, it contains figures, tables, and ideas we have previously been published elsewhere, and parts of our work are conveyed literally or with minor adaptations. Figures and tables that were published previously or have only been slightly modified are additionally marked by the corresponding reference in the caption. A complete list of publications that originated from this work can be found at the end of this thesis.

2. Background

For understanding the presented work on the development and implementation of proactive dialogue models, we first provide the necessary background and explain the relevant fundamentals. Primarily, this thesis focuses on DSs as the central component of autonomous technical systems, such as CAs. For this reason, we elucidate contemporary DS architectures, relevant dialogue models as well as strategies, and evaluation methods for measuring dialogue quality and user experience. For user-adaptive proactive dialogue modelling, we will largely rely on data-driven statistical methods. Therefore, we introduce available data collection methods and describe two ML approaches relevant for statistical DSs: unsupervised learning and RL. Concerning unsupervised learning, we will explain often applied ML algorithms, namely decision tree, support vector machine (SVM), and artificial neural network (ANN). For explaining RL, we introduce relevant concepts, such as markov decision process (MDP), and describe popular algorithms for realising RL. Finally, some psychological concepts are presented. These concepts need to be addressed, as one of the main aims of this thesis was to study the psychological and social impact of proactive dialogue on the user. Further, we consider user-adaptation approaches for proactive dialogue, which require the system to know the user's psychological state to improve cooperation. Therefore, we cover psychological aspects that are most relevant in a cooperative context. Here, we provide background on a theory of mind (TOM), which addresses fundamental knowledge about how technical systems may form beliefs about the mental states of their users. In human-machine cooperation, trust is an elementary mental state to consider. For this, several relevant trust concepts and models are presented. Additionally, other mental states, such as cognitive load and cognitive-affective user states are explained. As the perception of proactive dialogue is supposed to be quite dependent on the individuality of users, we included the concept of personality into our considerations and describe different personality models.

2.1. Dialogue Systems

A DS is an interface that allows humans to use natural language when interacting with computers or machines. Exchanging information using natural communication modalities such as spoken or written language seems to be a trivial task to humans, as the various processes involved are mostly perceived subconsciously. On second sight, however, the nature of the task is quite complex, e.g. see Lindsay and Norman (1972); Card (1981). To illustrate the difficulty of this, a model for human information processing called the *human processor* by Card (1981) is provided as an example: The human senses (auditory, visual, ...) perceive external signals (stimuli), which then are pre-processed and transformed in electric impulses for subsequent processing. A representation of the stimuli is then saved in a sensory register. The so-called *perceptual processor* takes the representation as input and refines the information using the long-term memory, which contains knowledge about how to perform tasks (procedural knowledge) and what to do to accomplish them (declarative knowledge). The result is then processed by the *cognitive processor*, which then plans actions taking into consideration knowledge provided by the short-term memory, being responsible for decision-making and memory search, as well as the long-term memory. Subsequently, the actions are controlled by a *motoric processor*, which instructs the human effectors (e.g. speech production) to perform them. Analogously, researchers from various disciplines have tried to describe interactive systems using computational models based on human processing capacities. As a result, nowadays there exists a range of various architectures, methods, and models for processing natural language and providing systems with the ability to conduct dialogues.

2.1.1. Architectures

For structuring the complex processes which are involved in the conversation, an abstraction of the problem has been provided in the form of a DS architecture. Generally, a DS needs to solve the problems of recognising user input, understanding the underlying meaning or intention, deciding on an appropriate system response, and generating a natural system output. Currently, there exist two fundamental architectural approaches: a *modular* and an *end-to-end* framework. While the modular approach relies on several individual components for providing natural language interaction, the end-to-end approach makes use of an ensemble of ANN for direct language generation depending on the input. In the following, these two approaches are described briefly.

Modular Framework

Following a modular approach, a DS distributes the different tasks for conducting a dialogue to individual components: Semantic en- and de-coding components transform textual user input into a machine-interpretable form, so-called dialogue acts, and convert the system's output back to natural language. For deciding the next system output, the dialogue manager takes into account various pieces of information, for example, the user's last input and/or the dialogue context, and chooses an appropriate response. In case a system assists by using spoken language, modules for converting speech into text and vice versa can be included. Further, other modalities different from language can be processed by conversational systems. Such systems are described as multi-modal DSs and are mostly applied in embodied agents, such as virtual or robotic assistants (André and Pelachaud, 2010). Contrarily to purely text- or voice-based applications, embodied systems possess more human-like features, e.g. extremities or a face. This allows them, for example, to generate gestures (Mitra and Acharya, 2007) or facial expressions (Pelachaud and Poggi, 2002). However, also combinations of input modalities, e.g. speech and affective or emotional user state, can be used by the dialogue manager to consider more information during the decision-making process (Pittermann and Pittermann, 2006).

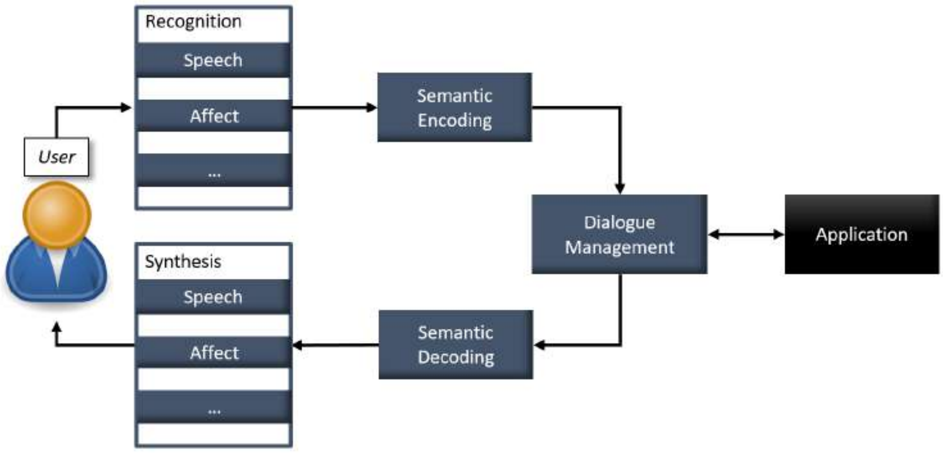


Figure 2.1.: Modular architecture of a DS.

The modular dialogue architecture is visualised in Fig. 2.1. The individual components of this architecture are described in more detail in the following paragraphs.

Recognition: Raw input signals, e.g. audio when recognising speech or video for facial expression recognition, are processed by recognition modules. As this work focuses on textual and spoken interaction, the individual steps for solving the speech recognition task are highlighted. The component executing these steps is called automatic speech recogntion (ASR) module. Here, the problem is to find the most probable word sequence given an acoustic feature vector (Jurafsky and Martin, 2020; Yu and Deng, 2016). For achieving this, the audio signal is firstly pre-processed to reduce noise and channel distortions. Subsequently, the signal is digitalised using Fourier transformation and relevant features for speech recognition are extracted from the signal spectrum. Typically, the mel-frequency cepstral coefficients (MFCC) are used, as the human ear is only capable of perceiving certain frequency bandwidths. Based on feature vectors containing MFCC the most probable string of words is estimated using statistical methods. In the past, a combination of Hidden Markov models (Juang and Rabiner, 1991) (acoustic model) and so-called n-grams (language model) was used. While the acoustic model maps the observed feature vectors to word sequences using a dictionary, the language model augments the recognition including language-specific information, e.g. grammar or syntactic rules, by learning statistical dependencies between hypothesised words. The current state-of-the-art for speech recognition, however, is the application of encoder-decoder models which are realised using recurrent neural network (RNN) or transformer networks (Yu and Deng, 2016). These models have shown to significantly decrease error rates for word recognition (Yu and Deng, 2016). For recognising affect or emotion from audiovisual signals, similar approaches relying on deep learning architectures have obtained promising results (Kim et al., 2013).

Semantic Encoding: For extracting the meaning of the user input, speech, or in the case of a multi-modal system other modalities, e.g. images or visual information, are semantically encoded. Semantic encoding is conducted by generating a semantic representation of the user input based on formal structures. We illustrate this approach by looking at speech processing, which employs a natural language understanding (NLU) module. NLU can be realised in various ways including syntax-driven approaches, rule-based semantic grammars, or nowadays commonly ML methods (McTear, 2020). Utilising a syntax-driven approach, the input is analysed based on the composition of the user's utterance. Therefore, the sentence is split into its syntactic components, e.g. noun or verb phrases, according to a specific grammar. For this, often context-free grammars and lexical rules are applied (Tur and De Mori, 2011).

Another approach is to use semantic grammars. Such grammars are rule-based coding schemes that capture the semantics of speech input in the form of a semantic frame or dialogue act. Contrarily to syntactic-based approaches, where the sentence components are categorised according to their syntactic function, semantic grammars structure the sentence depending on the communicative function of its constituents. An example is shown in Fig. 2.2. For creating a standardised taxonomy of communicative functions, several proposals have been made. One of the first attempts was provided by Searle (1969). According to his speech act theory, communication functions can be classified. For example, it can be differentiated between *assertives* that address the state of a current situation, e.g. stating, claiming, or suggesting; *directives* intending to commit the addressee to do something, e.g. ordering or commanding; or *commissives* that attempt to commit the speaker to do something, e.g. promising or threatening. Based on this preliminary work, several taxonomies for describing dialogues have been developed (Traum and Hinkelman, 1992; Core and Allen, 1997). One of the most used taxonomies is the dialog act markup in several layers (DAMSL) that defines a set of primitive communicative actions that can be used to semantically analyse dialogues (Core and Allen, 1997). DAMSL differentiates between forward- and backward-looking dialogue acts. While forward-looking acts include statements and requests, backward-looking implies for instance the expression of agreement with the previously provided information or answering requests.

Although these rule-based grammars can capture fine-grained distinctions in the input, it is still required to manually craft rules for every possible input. Whereas hand-crafting rules may be beneficial in rather small conversation domains, this process becomes more and more expensive and time-consuming with increasing domain complexity, also requiring an expert grammar author. Therefore, statistical methods using ML are predominant nowadays; which receive different names, e.g. intent classification or dialogue act tagging. Here, the NLU-task is modelled as a classification task: dialogue utterances labelled with semantic concepts, e.g. using the DAMSL, are the input to the classifiers. For predicting the dialogue acts, different machine-learning methods have shown to be valuable e.g. SVM (Schuurmans and

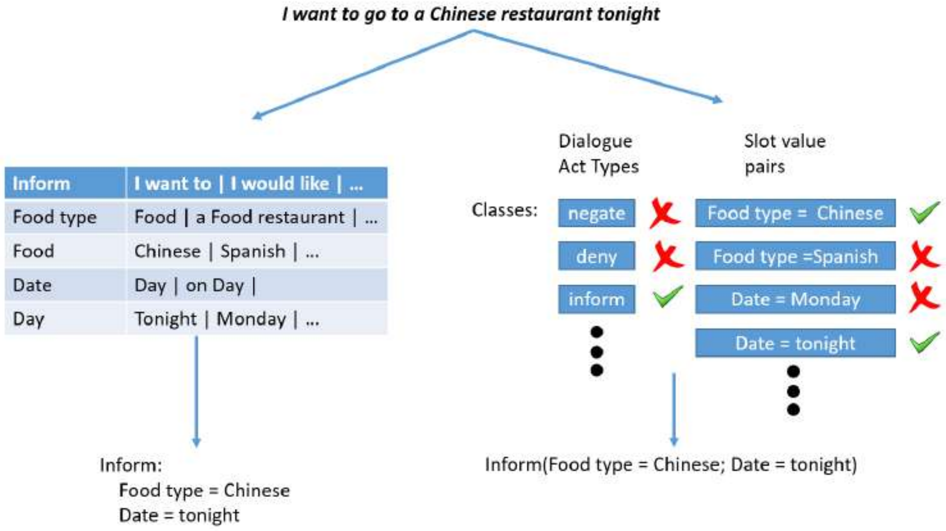


Figure 2.2.: Depiction of a NLU process. Left: NLU using semantic grammars. Right: NLU using classification.

Frasincar, 2019) or convolutional neural networks (CNN) (Collobert and Weston, 2008). However, deep neural networks have been proven to outperform classical ML methods (Tur et al., 2018; Sarikaya et al., 2014). Besides the classification of the dialogue act or intent (what the user wants), also relevant information on the word level can be classified. This process is known as entity extraction (e.g. see (Tur and De Mori, 2011)). An example is provided in Fig. 2.2. As can be observed, the utterance "I want to go to a chinese restaurant tonight" can be either parsed using predefined grammars or classes for predicting the appropriate dialogue acts and entities. The semantic representation of the user utterance is then conveyed to the DM module to select an adequate response

Dialogue Management: The dialogue manager is the decision-making module in the architecture of a modular DS. It controls the flow of the dialogue, interprets the user's semantic input, and interacts with external services or applications for accomplishing different tasks (McTear, 2020). For some tasks, e.g. question-answering or so-called chit chats that include small talk, DM does not necessarily require context information for achieving its purpose. However, in cooperative dialogue, where both the user and the system equally contribute to the conversation for fulfilling a goal, DM needs to keep track of the conversation and use other relevant information for providing adequate assistance. Considering the example presented in Fig 2.2. *Inform(food_type = Chinese, date=tonight*, DM takes this semantic representation as input, verifies if some additional information, e.g. price range, is missing and either proceeds in querying a restaurant database for providing the user with information or asks the user for the relevant information.

For realisation, two main concepts are involved in the DM process: dialogue state and dialogue policy (Young et al., 2013):

The *dialogue state* contains relevant information for keeping track of the conversation (McTear, 2020). For this, several knowledge sources are used: The dialogue history contains information about the dialogue participants' contributions so far. This information is often represented in the form of an agenda-based data structure, that also contains knowledge about what information still has to be gathered from the user depending on the previous input. A domain model represents a system's "world knowledge", i.e. concepts and information for a specific task domain, such as several food types or price ranges in the restaurant search domain. This knowledge can be retrieved from a database that can be structured in the form of a knowledge graph or ontology, for example. Besides conversation- and domain-related knowledge, the dialogue state can also include user-specific information, e.g. age, gender, or preferences, and relevant dynamic user states. For example, in the semantic encoding modules of multi-modal systems, the user's affective state can be categorised based on facial expression features found in images or videos and mapped to states, such as anger or fear (McCrae and John, 1992).

The *dialogue policy* determines the next system action dependent on the last user input and current dialogue state information. For selecting the next system action, decisions have to be made, for example, whether the system needs to request or provide information, and if previously provided information needs to be clarified. Here, different dialogue strategies may be applied. Such strategies can be pre-defined in advance, e.g. the system proceeds to the next dialogue step if the confidence that a user input has been correctly understood exceeds a specified threshold, or otherwise asks for clarification. Moreover, strategies can be deployed dynamically by taking into account the dialogue state's knowledge sources. These sources can also be used to adapt the dialogue to the relevant user- and/or context-related information.

As DM is the focal point of this thesis, more detailed information about how dialogue state and policy can be modelled is provided in Section 2.1.2. Furthermore, we highlight fundamental concepts of dialogue strategies and provide relevant work in user-centred DM in Section 3.2.

Semantic Decoding: The DM module produces system output in the form of a high-level semantic representation. Considering our dialogue example, a possible system output would be to ask the user for his preferred price range, represented in the dialogue act "*Request(price_range)*" For being understandable to humans, the natural text needs to be generated from this representation. This process is called semantic decoding. For decoding abstract system actions in a textual representation, there exist several options: The simplest way is to use templates, e.g. see Reiter and Dale (1997). Here, either a template for each system action is stored in a look-up table, e.g "Which price range should your restaurant have?", or a template containing placeholders is stored which are dynamically filled during run-time, e.g. "Which $< frame >$ should your restaurant have?".

Even though this approach is rather inflexible, it can be effective for a manageable amount of system actions, which usually is the case for early prototype systems. To create a more dynamic and diversified text generation, other more sophisticated semantic decoding methods have been developed. This includes AI planning approaches (Reiter and Dale, 2000), statistical approaches relying on tagged dialogue corpora using utterance classes (Oh and Rudnicky, 2002), supervised learning approaches using recurrent neural networks (Wen et al., 2015), end-to-end approaches (Dušek et al., 2020), as well as RL approaches (Rieser et al., 2014). In case the system output needs to be conveyed using natural speech, a synthesis module uses the textual representation to generate speech signals. Furthermore, the text can be accompanied by using additional modalities. In this case, the semantic decoding module of an anthropomorphic agent may select, for example, adequate gestures or facial expressions to create a more realistic user experience (Hartmann et al., 2005).

Synthesis: A synthesis module is responsible for the transformation of the system output representation in the desired format, i.e. for producing gestures, animations are generated by adjusting an avatar's facial features, while for speech production text is converted into a sequence of phonemes. In early systems, unit selection techniques were applied (Hunt and Black, 1996). Here, linguistic units are retrieved from a large speech database to convert a text into a sequence of sounds (Taylor, 2009). The units are selected according to how well they represent the target specification of an utterance and the quality of a concatenation of individual units. The target, as well as the units, can be any mixture of acoustic and linguistic features, e.g, phonemes or diphones with pronunciation features. For finding the best fit between units and target specification, as well as the best combination of units, a cost function needs to be minimised. After finding the best sequence, the speech waveform is finally synthesised from the concatenated units' spectral and excitation parameters by applying a speech synthesis filter. More recently, new approaches using end-to-end deep neural networks are applied for directly mapping words to the appropriate speech signal, e.g. see Prenger et al. (2019).

As described, the modular DS architecture follows a pipeline approach, where the output of one module is the input of the following. A characteristic of this approach is that each module needs to be implemented and fine-tuned separately. This allows reusing already implemented modules fast and easily. Further, the integration of external services, e.g. ASR or NLU, is facilitated and the architecture is easily extensible. This makes the modular approach especially useful for developing new prototypes. However, a downside of the approach is that the modification of one module can result in negative effects on the others. This also known under the term "knock-on effects" (McTear, 2020). This possibly makes the system more difficult to maintain and to identify module-specific errors. Additionally, this problem renders modular architectures not very transferable to new domains. To counteract these downsides and to leverage the knowledge that can be obtained from big data sets recently an end-to-end approach emerged. Even though this approach is not used in the scope of this work, we provide a short introduction.

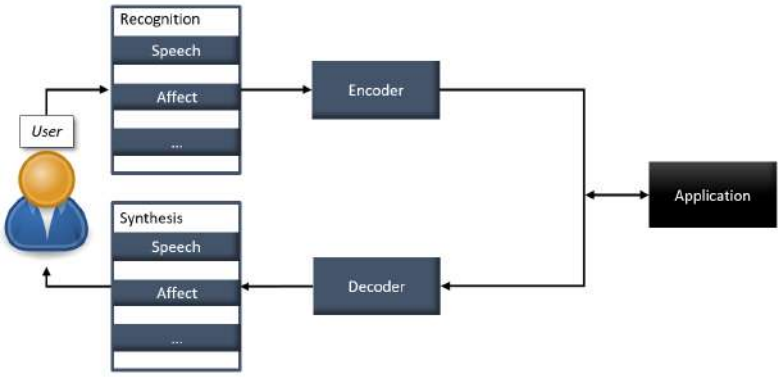


Figure 2.3.: End-to-End architecture of a DS.

End-to-End Framework

End-to-End DSs unify semantic en- and decoding, as well as DM into one module consisting of two main components: Encoder and Decoder. *Encoding* implies the processing and representation of user input, while *decoding* is the output generation. The basic idea of this approach constitutes that conversation is modelled by predicting the next output in the dialogue given some previous user input. Thus, there exists a sequence-to-sequence mapping of input to output. For creating these mappings, primarily neural network approaches are applied, e.g. see Bordes et al. (2016). First, the encoder network creates a context vector that represents the user input. Afterward, the decoder network uses this vector to create an output. In the following the two components are described (McTear, 2020):

Encoder: The input of the encoder forms a sequence of words, affective states, or other relevant user input. For an illustration of the end-to-end process, we take the example used for describing the DM module from the previous section. In the first step, the word sequence needs to be converted into a numerical form to be processed by the encoder network. For this, word embedding is used to generate a unique real-number vector given a word that represents its meaning and its relation to other words in the vocabulary (McTear, 2020). Currently, two-word embedding techniques are primarily used: One-hot-encoding and so-called WORD2VEC-approaches. Using one-hot-encoding, categorical values, e.g. words in a vocabulary, are transformed into fixed-size vectors. The size of the vector equals the number of categorical values to be transformed. For distinguishing the values, each but one vector cell is filled with zeros. The exception is the cell that represents the position of a particular categorical value which is encoded with a "1". Using WORD2VEC (Mikolov et al., 2013), categorical values are represented in a so-called semantic space. Here, each categorical value can be identified by its unique position in the space. Furthermore, semantically similar values are closer together in the space. For transforming a categorical value in a semantic space, it is represented as a numerical vector in

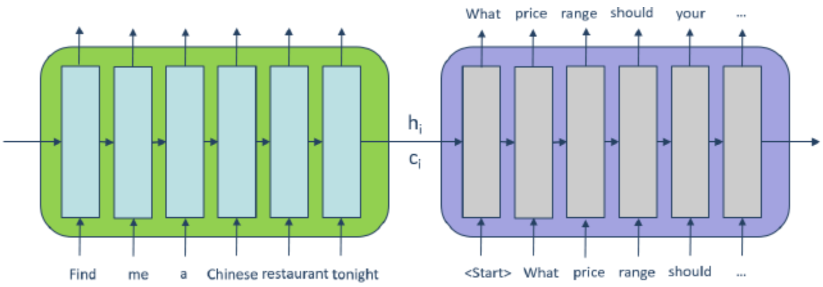


Figure 2.4.: Sequence-to-sequence dialogue modelling.

which cell values are learned from large text corpora. The word embeddings are then used by deep neural networks for creating the context vector that represents the input sequence.

Decoder: The decoding component of a end-to-end framework takes one element of the context vector at a time and generates an output sequence (McTear, 2020) (see Fig. 2.4). The mappings from input to output sequence are learned from large corpora of dialogues. For generating the output sequence, there exist two methods: autoregressive generation and retrieval-based generation (McTear, 2020): Using autoregressive generation the word that is processed is conditioned on the word generated by the network at the previous time step, and the context vector. On the contrary, using retrieval-based generation, a pre-defined response is retrieved from a data source (dialogue corpus) by matching against the input.

End-to-end approaches have gained widespread popularity in research nowadays, where they have been used for several applications. For example, Bordes et al. (2016) studied their application for task-oriented dialogue in the restaurant reservation domain. However, end-to-end systems are mostly studied in the context of open-domain chatbots which conversations can span a wide variety of topics, e.g. Google's Meena (Adiwardana et al., 2020) or Facebook's Blenderbot (Roller et al., 2020). Despite their popularity, end-to-end systems still have several problematic issues. For example, see McTear (2020) for an overview. One of the main problems is the generic response problem, which concerns the often bland or uninformative responses of such systems, e.g. *"Ok."* or *"I'm not sure."*. Further, they are prone to semantic inconsistencies, i.e. their responses are inconsistent with their previous responses. For example, they may state different cities when asked for their current habitat. Last but not least, for end-to-end systems to work on a relatively reliable level, they need to be trained on huge data sets. This makes them rather impracticable for deployment in domains with scarce data, and for early prototype development for specific applications. In addition, it is still unclear how to integrate social user information, e.g. emotion or trust, in such systems. For these reasons, we exclusively rely on the usage of modular architectures in the scope of this thesis.

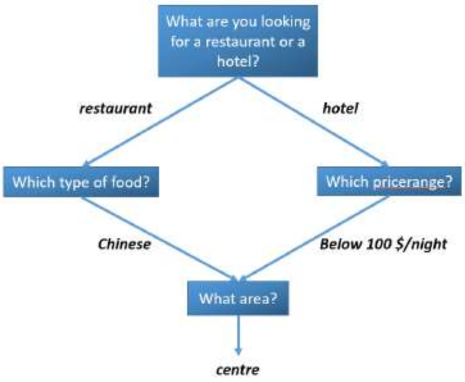


Figure 2.5.: Typical callflow in the touristic domain.

2.1.2. Models

As outlined in the introduction, the main objective of the presented work is the development of a dialogue model that allows CAs to interact with users proactively. Therefore, it is necessary to define a dialogue model and explain different kinds of modelling approaches. Generally, dialogue modelling can be defined as follows: "A dialogue model is an abstract model that is used to describe the structure of the dialogue between a user and an interactive computer system" (Green, 1986). Here, the structure determines how the dialogue state is represented, which actions a user and the system may take, the methods to decide when to take which action, what dialogue information is relevant, and how the decision influences the dialogue state.

The dialogue model is then used by DM to take control over the content and flow of the dialogue. This is provided by interpreting the user input in the context of the dialogue, evaluating the relevance of user requests, identifying and recovering from recognition and understanding errors, tracking the dialogue history, and updating the current dialogue state. Since ELIZA several approaches for modelling the dialogue and thus the design of the DM module have been developed and extensively researched. The three main approaches are introduced in the following:

Finite State-based

The simplest model for modelling dialogue is the finite state-based approach (McTear, 2004). Here, the dialogue structure is represented as a state transition network, where the nodes are the possible dialogue steps and network paths depict the valid dialogue flow. Thereby, the user is taken through the dialogue following a sequence of predetermined states. The decisions the DM may take are modelled as pre-scripted rules which are based on the possible user input. Usually, the transition network can be visualised in the form of a call flow (see Fig. 2.5). The main advantage of a finite-state dialogue model is its simplicity which makes it particularly useful for rapid prototyping. As the user's input is limited to single predefined words or phrases, a full natural language processing setup is not necessary, Instead, a simple speech recognition for recognising keywords suffices. Furthermore, a simplified DM component can be used being easy and fast to develop. Usually, the flow of the dialogues is system-directed which makes such system quite reliable, as users are not allowed to deviate from the defined network transitions. This also makes finite-based systems suitable for well-structured tasks, e.g. restaurant information, or bus scheduling, where dialogues can be scripted easily. However, considering a real-world application of finite state-based models has several disadvantages. First, the approach is problematic for covering complex domains. As the DM's decisions must be pre-defined in advance, a system designer has to think of all possible paths through the dialogues which can easily become intractable for larger domains, especially when dealing with non-atomic task structures. Furthermore, the models are quite inflexible, as users cannot take the initiative and deviate from the current dialogue path. In such cases, a system would ignore deviation and only ask irrelevant questions. In addition, users must know the predefined words or phrases due to the restricted vocabulary, which may lead to a high error rate for speech recognition. For providing systems with more flexibility, finite-state approaches have been enhanced to information state-based approaches.

Information State-based

The dialogue structure of the information-state approach is also represented in a state transition model. However, the dialogue is not defined as a sequence of predetermined states, but based on a distinct set of information that is called frames or slots (Ginzburg et al., 1996). Dialogue slots are pre-defined entities of a specific domain, which are structured as attribute-value pairs. For an information retrieval task, e.g. restaurant search, labels for slots could be food type, price range, location, etc. Via a dialogue, the system then needs to retrieve values for these slots from the user, e.g. Chinese, expensive, and west side of town, for identifying the user's goal. When all necessary information is gathered a database is queried for providing the user with the desired information. Contrary to finite state-based approaches, the dialogue state transitions are not pre-defined but dependent on which kind of information has been provided by the user. Thus, the dialogue evolves dynamically. Larsson and Traum (2000) provided a theoretical foundation for the information state-based approach based on five key concepts. These concepts serve as blueprint for implementing a DM using an information state for decision-making. The five fundamental concepts are:

Informational components are different kinds of knowledge sources for modelling the desired system behaviour. Such knowledge sources can differ whether they carry information about the context or the interaction itself. For example, a user's internal state (e.g. goals, intentions, attitudes) can be seen as context information, whereas information about the interaction may include the dialogue length, duration, or misunderstandings, for example. As such informational components can also be categorised into static knowledge bases, which handle information that is not subject to change throughout a dialogue or knowledge bases that handle dynamic aspects of the conversation.

Formal representation describe the way how the informational components are modelled, e.g. which data structures (lists, ontologies) are used and how accessible (private vs. shared) the data is.

Dialogue moves are basically dialogue acts. As previously described, they form an abstract representations of possible user and system actions.

Update rules constitute a set of predefined rules that formalise the change of the information state as the dialogue progresses. Each rule has preconditions and effects. Preconditions, e.g. a certain slot has been filled, need to be satisfied for the rule to become executable. Effects are changes made to the information state.

Update strategy is a mechanism that decides which rule should be executed among a set of rules which preconditions are met. Furthermore, there may exist domain-independent update rules looking at the quality of the input recognition and tracking possible understanding problems for executing generic protocols to deal with them.

The major advantage of using an information state-based approach is its greater flexibility for the user. While a finite-state-based system guides the user through a static dialogue flow, the information state allows the user to actively influence the dialogue path. For example, over-answering may be allowed, i.e. users may provide the system with more slots than the system asked for and hence shorten the dialogue. Furthermore, the approach is attractive as it encourages a declarative formulation of the required knowledge sources and the rules for computing the state transitions. From a designer's perspective that makes the system easier to maintain and to transfer the model to different domains.

The update strategy can be hand-crafted (Larsson and Traum, 2000) or automatically learned (Levin and Pieraccini, 1997). For the latter, the task of finding an appropriate dialogue strategy is formulated as an optimisation problem. Here, the information state-based model is described as an MDP) that allows to automatically learn a dialogue strategy for a given application using RL. In Section 2.2, we will provide a more detailed overview of the concept of RL and MDPs.

As user actions are subject to uncertainty, due to imperfect speech and intention recognition, the system only has a belief about what was said and does not take into account that this belief could be false. For including uncertainty into the DM process Young et al. (2013) extended the information state approach to a hidden information state model. Here, it is assumed that the dialogue states are not directly observable. The dialogue state can only be estimated by making observations, i.e. the user actions. Thus, there exists a probability distribution over multiple information states. For allowing decision-making under uncertainty, an information state-based dialogue model formulated as a partially observable decision process (POMDP) has been introduced (Williams and Young, 2007). However, this approach is not the subject of this work. Hence, it is not described more in detail.

Plan-Based

The previously described models, finite and information state-based, are most useful for providing rather soft assistance, i.e. the dialogue manager interacts with a combination of databases and web services for information retrieval and question answering. For providing a deeper and more intelligent kind of assistance, e.g. for collaborative problem solving and decision making, querying a database or web service might not be sufficient. In collaborative interactions with assistant systems, users expect the system to further one's plans and goals (Allen et al., 2019). For fulfilling this expectancy, CAs need to understand what plans and goals a user has and be able to reason about them. Additionally, CAs can facilitate achieving one's goals by performing adequate actions. In accordance, CAs themselves need to create plans for performing actions and for reasoning about their actions (McTear, 2020). Thus, task-oriented DSs need to be equipped with reasoning and planning components for becoming truly cooperative CAs.

First work regarding the integration of planning in DS was provided by Cohen and Perrault (1979) as well as Allen and Perrault (1980). Their work presents a theoretical model for plan-based speech acts to recognize or construct plans for collaborative dialogue. An evolution of this theory is the well-known belief-desire-intention (BDI) model (Cohen and Levesque, 1990; Bratman et al., 1988) that sets the basis to endow intelligent agents with rational behaviour, including planning and reasoning. This enables them to form their plans and generate a set of actions in order to reach them based on internal and external states. Typically, AI planning approaches are used for generating an agent's action sequences for accomplishing its goals. The three main components of the BDI-framework are:

Belief: This concept represents the agent's information state, that is, its available knowledge about the world including itself and other agents.

Desire: This concept represents the motivational state of the agent. Thus, desires constitute objectives or an ideal state of the environment that the agent would like to achieve.

Intention: This concept represents a subset of desires, an agent is committed to achieve. Therefore, an agent develops a sequence of actions based on its beliefs for attaining its goals.

For deciding which desires become intentions and how to select intentions for becoming agent actions, a BDI model makes use of practical reasoning. A deliberation component is used for strategic thinking and decides what desires need to be currently accomplished. The result is a set of intentions. Afterward, a means-ends reasoning component deals with tactical planning and selects actions that should be performed to accomplish the set of committed intentions. As a result, a set of plans and actions is generated. In conversational systems, the actions would be represented as communicative action schemes. Such action schemes consist of preconditions and effects. To achieve a desire, a conversational agent has to plan a sequence of communicative acts. This sequence starts with the communication act having the initial belief state as precondition and ends with the communication act having the goal state as an effect.

Practical realisations of the BDI-model can be found in the development of prototypical dialogue managers, e.g. see Sadek et al. (1997) or Bohus and Rudnicky (2009). More recently, Galescu et al. (2018) provided an extension to the development of BDI-based dialogue managers by including concepts of collaborative problem solving (Allen et al., 2002). However, these plan-based approaches for developing a collaborative CA are incomplete as they lack the ability to be user-adaptive and to enable personalisation. In this regard, Biundo and Wendemuth (2016) specify the concept of Companion-technology which aims at intelligently adapting an assistant's functionality to individual user requirements. According to the authors, a cooperative CA or so-called companion, combines the cognitive abilities of planning, knowledge reasoning, and adaptive dialogue to provide individualised assistance. Empirical evaluations of companion systems (Bercher et al., 2014; Behnke et al., 2019c) demonstrate the acceptance and usefulness for users in real-world application scenarios. In this work, we follow the line of research of cooperative CA, as we study assistance systems with cognitive abilities in this thesis.

2.1.3. Concepts of Dialogue Strategies

In CAs, the DM component has to make a recurring decision: which action to take that suits best the current context and situation for pursuing a specific goal. Thus, a system follows a certain strategy or policy for achieving this. Depending on the type of task and the specific target of a system there exist different kinds of strategies. For example, if a system's goal is to persuade a user then it needs to find adequate argumentative strategies. An example of such strategies is described by Rach et al. (2018). In the tutoring domain, an assistance system aims at increasing the users' learning gain or their motivation (Graesser et al., 2001; Litman and Silliman, 2004). Here, a system has to find appropriate engagement and motivational strategies, e.g. the timely use of appraisals or reflective dialogues.

However, the primary goal of task-oriented systems is to increase task success, e.g. see Wen et al. (2016); Su et al. (2015); Litman and Pan (2002). For this, a system needs to make sure to keep the user engaged in the task and to correctly process the user's input. Therefore, strategies have been defined to clarify user information (*grounding*) and to decide who should have the *initiative* during which point of the dialogue to progress with the task. The selection of appropriate initiative and grounding strategies has also shown to be beneficial for another important goal in HCI: enabling user satisfaction or providing a high interaction quality, e.g. see Ultes et al. (2015); Ultes (2019); Hastie et al. (2002); Forbes-Riley and Litman (2006). As the dialogue initiative selection and grounding strategies are also important for the realm of proactive dialogue models, we will explain those two concepts in more detail in this section.

Grounding

One of the major challenges in DSs is to deal with uncertainty and errors due to ambiguity of language and the sensory limitations of current systems (Skantze, 2007). Addressing this issue, however, is not the task of an individual during the interaction. All dialogue participants are required to collaborate for identifying positive or negative evidence of understanding. In doing so, private knowledge and beliefs can be shared with each other for reaching a common ground (Clark, 1996). This process is understood under the term *grounding.* There exist two aspects of grounding: either as initiated by a provider of information aiming for clarification, justification, or conviction; Or as initiated by a receiver of information to resolve misunderstandings or disagreement (Gregor and Benbasat, 1999):

Initiated by receiver: In case a DS detects potential misunderstandings, e.g. a certain confidence threshold for the recognised user utterance has not been exceeded, it may ask the user for a confirmation of the relevant piece of information. As a response, the user may validate or decline the system's assumption. For the confirmation of user information, a DS may use either an explicit or an implicit strategy. Using an explicit strategy, e.g. "Do you want a Chinese restaurant?", the current dialogue is halted and the user needs to affirm or deny the system's statement. Using an implicit strategy, e.g. "What price range should the Chinese restaurant have?", the dialogue continues to the next state by restating the previously provided information. Comparing the two strategies, the implicit one may provide a smoother and more efficient dialogue. However, false information can be confirmed unnoticed by the user more easily, which can foster the clarification of information more cumbersome, especially for novice users who are not familiar with the system's mechanisms.

Initiated by provider: A common phenomenon in human-human dialogue is to reason about dialogue actions and identify their causes, i.e. to find a reason behind a certain utterance or action. For this, explanations are used which can either have an intentional, e.g "I want to go to a Chinese restaurant because I like noodles" or functional purpose, e.g. "The system has a button to enable its speech recognition". As autonomous systems able to make independent decisions, CAs can use explanative behaviour for providing the user with additional information about its inner processes, i.e. how a system came to a specific decision that may be unclear to the user. In doing so, a CA proactively integrates the user into its decision processes. Depending on the goal which should be accomplished by explaining, it can be differentiated into five types of explanations: transparency (how the system came to this decision), justification (why is this decision appropriate), learning (provide the user with information about the domain), and relevance (why is this decision relevant) (Sørmo and Cassens, 2004).

While confirmation dialogues are a basic functionality of DSs due to the inherent uncertainty of language, explanative behaviour has only been recently integrated. For example, Nothdurft et al. (2012) have developed an adaptive explanation architecture, that can select an appropriate type of explanation depending on the context- and a user profile.

However, explanations have become a hot topic in HCI, especially concerning the interpretability of machine-learning behaviour which is now predominantly found in DSs (e.g. see Du et al. (2019)). In this thesis, explanations are relevant for justifying and clarifying proactive behaviour.

Initiative

Generally, it can be differentiated between *task* and *dialogue* initiative in the domain of intelligent CAs. According to Chu-Carroll and Brown (1998) "an agent is said to have the task initiative if she is directing how the agent's task should be accomplished, i.e., if her utterances directly propose actions that she believes the agent(s) should perform." Such utterances may propose particular domain actions, e.g. the ordering of Chinese food in a decision-making scenario about a certain type of restaurant, or problem-solving actions, e.g. constructing a plan on how to decide what are the necessary elements for making the decision. In HCI, either the user, the system, or both can have the task initiative. In mixed-initiative interactions, a user and an autonomous agent collaborate for solving tasks by taking interleaving actions.

The dialogue initiative can be divided similarly into three types during conversational interactions according to McTear (2020) and Litman and Pan (2002):

User-directed: Here, the user initiates and controls the dialogue. However, this strategy mostly supports so-called one-shot interactions, where the user issues a question or command and the system reacts. Hence, this strategy is typically applied in smart speakers and smart home assistants, where the user can repeatedly ask about different topics.

System-directed These dialogues are led by the system. Here, McTear distinguishes between three types: the system initiates an interaction to deliver a reminder or notification (proactive); instructional dialogues in which the user starts the dialogue and the system repeatedly guides the user with little input from the user; the previously described slot-filling dialogues, in which a user commences the dialogue with requesting a service and the system takes over the command of the interaction posing a set of questions for working out the user's preferences and helps with task completion.

Mixed-initiative: The term for this kind of initiative may not be confused with the term mixed-initiative interaction for human-machine collaboration on a specific task. In the sense of dialogue initiative, mixed-initiative either refers to a system's ability to ask open-ended and specific questions, while providing the user with more freedom when answering questions in tasks-oriented dialogues (Litman and Pan, 2002). Otherwise, the term can also be applied in open-domain interactions where the conversation can span a variety of topics and both the system and the user have control over the dialogue flow (McTear, 2020).

2.1.4. Evaluation Methods

To compare the performance or quality of individual conversational systems and to measure the effect of different dialogue strategies, evaluation concepts are necessary. Therefore, various types of measures and methods for evaluation have been developed. For example, system evaluations can be conducted with real or simulated users using objective or subjective measures. In the following, the dialogue evaluation measures relevant to the studies conducted in the scope of this thesis are briefly described. Furthermore, we discuss the evaluation with both real users and by applying a user simulator. During the conceptualisation phase of proactive dialogue models, studies were conducted with real users to find out specific characteristics of proactive interaction, while the performance of the implemented model is tested using simulated users. A more detailed descriptions of methods for dialogue evaluation can be found in McTear (2020) and Pietquin and Hastie (2013).

Objective

Objective measures are typically derived from logs of interactions with users. There exist different measures for the evaluation of each module of the DS. However, in this thesis, we concentrate on the evaluation of DSs in general, as the impact of proactive dialogue strategies on the performance of a CA is investigated. Relevant measures for the studies conducted in this work are *Task Success*, *Dialogue Duration*, and *Compliance*:

Task Success implies how well a system can perform a given task. In slot-filling dialogues, task success is usually measured by checking the correctly identified values of a user goal. For CAs applied in mixed-initiative tasks, however, there exist no slots. Therefore, other measures are necessary. For example, a binary measure whether the task at hand was accomplished by the human-machine team or not. For testing the proactive dialogue model in this work, we apply numerical scores to individual decisions in sequential problem-solving tasks. For example, a task step can have multiple different options to which different numerical values (0,10,20) are attached for representing the quality of a decision.

Dialogue Duration describes the length of a dialogue. In most applications, efficient task solving is desired. Thus, dialogue strategies that lead to task success in less duration are found to be superior to others.

Compliance indicates whether a user follows the lead or suggestions of a system or not. In proactive systems, the user is provided with different kinds of notifications, suggestions, or persuasive messages. In case the user agrees to use a suggestion by the proactive systems, the user is compliant. Otherwise, the system would be not successful. Compliance can also be evaluated using binary measures for “success” or “fail”.

Subjective

As the interaction with a DS includes human beings, another way to evaluate systems is to let user's express their personal opinions about a system's behaviour or operation. This is typically conducted relying on questionnaires. For representing the users' opinions as numerical values, users rate statements about their perception of the system on a scale. One of the most common scale types constitutes the Likert scale. A Likert scale is commonly used to measure attitudes, knowledge, perceptions, values, and behavioral changes. Usually, Likert scales consist of five, seven, or nine points for providing users with the possibility to express their agreement or disagreement with a statement in a more fine-grained way. For subjectively evaluating DSs or strategies, there exists a wide range of psychological and technical questionnaires. For the studies presented in this thesis, we primarily make use of questionnaires for measuring the following aspects:

Quality: Three aspects play a major role in measuring the quality of a conversational system (Möller, 2004): the effectiveness, i.e. the degree of accuracy and completeness of users solving a task with the system, efficiency, i.e. the effort of solving a task with respect to task outcome, and satisfaction, i.e. the users' subjective opinions about the usefulness and usability of the system. One of the earliest questionnaires measuring this aspect was provided by Brooke (1996). He invented the system usability scale (SUS) containing ten items for measuring the quality of a wide variety of products and services, e.g. software, websites and applications. Recommended for the subjective quality evaluation of telephone-based spoken DSs is the ITU-T Rec. P.851 (P.851, 2003). This questionnaire comprises multiple items relating to the quality of information obtained from the system, speech input/output capabilities, a system's interaction behavior, perceived system personality, impression on the user, and perceived task fulfillment (Möller, 2004). While we make use of the previous two questionnaires for some experiments presented in this thesis, primarily the subjective assessment of speech system interfaces (SASSI) questionnaire by Hone and Graham (2000) is used. The advantage of SASSI is that it is accepted and predominantly used in the dialogue community. Thus, its usefulness is validated and the results are comparable among different applications. Furthermore, it offers a set of distinguishable factors for measuring the system's quality. SASSI contains sub-scales for measuring annoyance, user satisfaction, cognitive demand, speed, and habitability. Except for the concept of habitability, the meaning of each sub-scale should be comprehensible. According to the authors a "habitable system may be defined as one in which there is a good match between the user's conceptual model of the system and the actual system" (Hone and Graham, 2000). Thus, this concept measures the congruence of the user's expected system behaviour and actual system actions.

User Experience: The quality evaluation of conversational systems focuses on specific system attributes and capabilities. For including more user-related aspects into the evaluation process, several user experience questionnaires have been developed. In this thesis, we make use of the long and shortened version of the user experience questionnaire (UEQ) (Laugwitz et al., 2006). The long-version of the UEQ contains six scales with 26 items: attractiveness, i.e. do users like or dislike the system; perspicuity, i.e. is it easy to get familiar with and learn how to use the system; efficiency, i.e. can users solve their tasks without unnecessary effort; dependability, i.e. does the user feel in control of the interaction; stimulation, i.e is it exciting and motivating to use the system; novelty, i.e. is the system innovative and creative and catch the interest of users. Attractiveness is a pure valence dimension. Perspicuity, efficiency, and dependability are pragmatic quality aspects (goal-directed), while stimulation and novelty are hedonic quality aspects (not goal-directed) and represent the "feeling" about the system. The shortened version only contains 10 items and uses the sub-categories pragmatic and hedonic qualities. Furthermore, we measured the motivation to use a system using the intrinsic motivation inventory developed by McAuley et al. (1989). This inventory contains dimensions for measuring the user's interest-enjoyment, perceived competence, effort-importance, and tension-pressure. However, as this questionnaire has overlapping dimensions with other used questionnaires, we mainly considered only single dimensions of this survey.

Acceptance: For measuring the potential usage of a system in the real world, the acceptance of such technology is inevitable. It can be assumed that a person with a positive usage attitude will use a specific technology (Davis et al., 1989). This attitude can be named behavioural acceptance, which depends on the factors "perceived usefulness" and "perceived ease of use" (Davis et al., 1989). Under the term "perceived usefulness" Davis et al. (1989) understand "the prospective user's subjective probability that using a specific application system will increase his or her job performance within an organizational context", and by "perceived ease of use", the extent to which a user expects the system to be free of expense. The greater the benefit of an information system and the simpler its usability, the sooner the user is ready to use the new system (Davis et al., 1989). For measuring the acceptance of a proactive dialogue model, we use the acceptance scale developed by Van Der Laan et al. (1997). This scale consists of nine items and evaluates system acceptance on two dimensions: a Usefulness scale and an affective Satisfying scale.

A detailed description of the questionnaires used in this thesis can be found in the Appendix. Objective and subjective evaluation metrics can be used to measure the perception of a dialogue strategy or a conversational system in general. The evaluation can be either conducted using real or simulated users. The differences between these two evaluation paradigms is presented in the following.

Real Users vs. User Simulation

Experiments with real users can either take place under laboratory conditions or "in the wild", i.e. in real-life scenarios (McTear, 2020). In a laboratory setting, study participants interact with a DS in scripted and limited scenarios. For rating the interactions, they complete a questionnaire at the end of the session or at different time steps during the ongoing conversation. Using this approach, the evaluation is strictly controlled and a wide variety of different scenarios can be explored and investigated. This ensures a high within-test and between-test reliability that allows the collection of extensive data sets. Furthermore, this allows more easily collect immediate feedback from participants, as interviews are possible and study participants may be encouraged to use the thinking-aloud method (Lewis, 1982). In this way, the users' cognitive processes may be better understood. However, the major problem with evaluations under laboratory conditions is that they may not reflect real-life usage which may negatively influence the validity of the measurements and the results may not generalise well.

These disadvantages can be alleviated using an "in the wild setting", where study subjects interact with a real system to accomplish a real task. For acquiring participants for these kinds of studies crowd-sourcing platforms like Amazon Turk (Jurcıcek et al., 2011) or clickworker [1] may be applied. These platforms offer a convenient way for collecting data based on a diverse sample enhancing the validity of the results. However, studies in the wild require to have already have a full-functioning system, which may be difficult to obtain during the early stages of the development cycle, so simplifications have to made. In addition, users have more freedom to act in a realistic environment with may endanger the standardisation of the study setup.

A problem for both, laboratory and in-the-wild studies, is recruiting a sufficient number of users for allowing a valid interpretation of data. For achieving a sufficient participant number, often more than 30 participants need to be recruited which can make it quite expensive to setup user studies. This especially holds true in exploration studies, where the outcome is not easily predictable and a variety of options need to be explored. Therefore, user simulation techniques have been developed. The idea of a user simulator is to interact with a DS pretending to be a real user. On the one hand, this allows testing DS prototypes with a large number of simulated "subjects". On the other, user simulation enables the collection of large amounts of data, which is a fundamental for training statistical, data-driven DSs (McTear, 2020). Additionally, they facilitate the exploration of dialogue strategies that may be difficult to obtain in studies with real users and that may not be apparent in existing dialogue corpora (McTear, 2020). Typically, user simulation produces output in the form of semantic representations of user actions (Eckert et al., 1997; Levin et al., 2000; Schatzmann et al., 2006; Lee and Eskenazi, 2012).

However, there also exist alternative approaches that can generate natural language utterances, e.g see Kreyssig et al. (2018). López-Cózar et al. (2003) even presented a user simulator that generates spoken utterances based on pre-recorded speech files. Considering the development of user simulators, it can be distinguished between two methods: rule-based and corpora-based. Rule-based methods rely on hand-crafted rules and a range of user profiles. A rule-based user simulator produces a static output for a given DS and system action (Pietquin, 2005). For incorporating the inherent uncertainty of language in user simulation and to model more natural user behaviour, probabilistic data-driven approaches have been developed. One of the earliest stochastic models was developed by Eckert et al. (1997) et al. who made use of conditional probabilities in the form of bigrams. Here, the dialogue context of the last turn was included for generating simulated

[1] www.clickworker.de

user responses. This was then used to train a stochastic task-oriented dialogue model modelled as an MDP (Levin et al., 2000). More realistic user simulators for task-oriented dialogue were developed by Schatzmann et al. (2006) and Lee and Eskenazi (2012), for example. Both approaches formulate a specific user goal model for representing the user's intention. Further, they apply a user model for generating a user action dependent on the last system action and the user's goal and use an error model for simulating speech recognition and understanding errors. Besides only simulating task behaviour, there is interest in modelling also the user's social behaviour which is more relevant to the work described in this thesis. For example, Jain et al. (2018a) propose an data-driven approach for modelling both kinds of behaviours. For this, their user simulator comprised four modules: a user model for generating distinct types of users having specific tasks and social goals; a social rapport estimator for predicting the level of rapport that is experienced by the user at every turn; a user dialogue manager that takes into account this information, besides user model and past system actions, for deciding on the next task- or socially-related user action. Further, a reward model can be included if the user simulator is intended to train a dialogue model using RL.

To automatically train dialogue strategies and to generate stochastic user models, statistical methods are necessary. Statistical methods relevant to this thesis's work are described in the following.

2.2. Statistical Methods for Dialogue Systems

In the previous section, we have described various DS components and modules that make use of statistical models without explaining their underlying principles. As most models are based on ML techniques, we provide a brief introduction to the ML algorithms that are used in this thesis. The introduction is mostly based on the popular textbook provided by Alpaydin (2020). There, the author describes ML as "...programming computers to optimize a performance criterion using example data or past experience". Typically, an algorithm, i.e. a sequence of instructions for transforming defined input to defined output, is used to solve specific computational problems. However, for some tasks, there is no knowledge of an appropriate algorithm, e.g. spam detection. Here, ML helps to "learn" an algorithm for solving tasks based on example data, e.g. a set of e-mails labeled as spam or no spam. Learning in this context implies that the computer tries to approximate the process underlying the generation of the data. Thus, it may not be able to identify the complete process but can detect particular patterns. These patterns may then be used to predict the future, gain knowledge from data, or both. Usually, it can be differentiated between three types of ML:

Supervised Learning: Supervised learning aims to learn a mapping from a set of input features to an output, where the correct value or label of the output is known. Considering the spam detection example, a set of input features could be specific elements of the email, e.g. wording of the document or the sender name. Here, the output could be labeled as a Boolean value representing if the specific email is spam or no spam. A supervised learning algorithm would then try to identify the correct mappings based on a set of labeled data. Typically, a supervised learning problem can be defined as a regression or as a classification task. Regression is used for estimating a numerical value, while classification tries to assign input features to one of multiple output classes that may be structured in an ordinal or categorical manner.

Unsupervised Learning: Here, there exist no output labels. The goal of unsupervised learning algorithms is to detect hidden structures in the input data without human intervention. These structures are learned by finding reoccurring patterns in the data. Popular unsupervised learning algorithms comprise clustering, where the aim is to find groupings of input features; association, which finds relationships between elements of the data set; and dimensionality reduction, which is a feature selection method for identifying the most relevant features or merging several less meaningful data into a new important feature.

Reinforcement Learning: These kinds of learning problems address finding an appropriate sequence of actions, i.e a policy. Thereby, an action is only good if it is part of a good policy. Here, the policy can be described as good if it optimises a specific cost function. For learning a good policy, this ML method needs to evaluate the quality of a policy and learn from past action sequences for finding new, better policies. A trial-and-error approach is utilised during the learning process, where the system receives rewards for taking "good" actions.

In this thesis, we utilised solely supervised and RL methods. Therefore, only algorithms from these methods are explained in more detail in the following sections. As data is a pivotal aspect of these learning algorithms, we first address the most important data collection methods for DS.

2.2.1. Data Collection Methods

According to Budzianowski et al. (2018), there exist three types of methods for collecting dialogue data: machine-to-machine, human-to-human, and human-to-machine. Machine-to-machine dialogue data is collected by simulating interaction outlines between an artificial user and a system bot via dialogue self-play (Shah et al., 2018). For generating a more diverse data set, crowd workers are then recruited for paraphrasing the utterances. This approach is useful for generating data for building task-oriented DSs. However, procedural turn-taking, i.e. sequential planning or decision-making task, as well as the handling of unstructured data is not covered by this approach. This approach is also highly influenced by the quality and capabilities of the user simulator.

The arguably ideal way of collecting dialogue data are HHIs, being the most natural approach and providing a high diversity of dialogues. With the rise of social networks, the idea of recording publicly accessible conversations emerged. Relying on unsupervised clustering algorithms, the Twitter data set (Ritter et al., 2010) consists of an open domain collection with more than a million of conversations extracted from Twitter. Similarly, the Ubuntu corpus (Lowe et al., 2015) provides chats in the area of technical support.

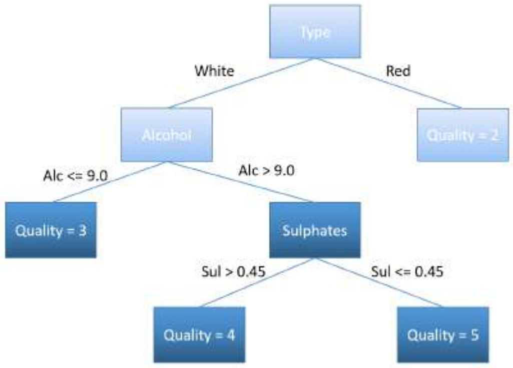


Figure 2.6.: Example of a decision tree predicting the quality of wine based on the type, alcohol and sulphates concentration.

The disadvantage of this type of collections is that parts of the data contain unusable texts and spellings and thus require adequate cleaning. Additionally, the majority of these conversations are not goal-oriented dialogues, while being mostly used to train end-to-end DSs (Lowe et al., 2017). A special type of human-to-human data collections are wizard-of-oz (WoZ) datasets (Kelley, 1984), in which a human wizard simulates system behaviour. Here, dialogues follow a pre-defined script designating the potential actions the wizard is allowed to take. The simulated system behaviour appears thus logically consistent and human-like. To increase the quality and diversity of dialogues, the MultiWOZ approach (Budzianowski et al., 2018) was developed containing 10000 dialogues in different domains obtained by employing crowd workers. The procedure for collecting such a dataset is, however, cumbersome and resource-intensive, since it requires additional human work-time for the data collections as well as for the subsequent transcription and labelling. This considerably limits the total number of samples and can lead to an inconsistent system due to the dissimilar behaviour of different wizards.

As the last type, human-to-machine data collection is conducted with interactions between users and existing DSs. Naturally, the prerequisite for this approach is that a DS is available and that all necessary functions have been implemented. This in turn facilitates the annotation process as it is possible to extract objective features directly during the experiment. So far, there already exist quite several such developed corpora, including the “Let’s Go Bus Information system” corpus from the Carnegie Mellon University (Black and Eskenazi, 2009).

2.2.2. Decision Trees

One of the simplest and most intuitive methods of supervised ML form *decision trees*. As indicated by the name, the prediction problem is structured in a tree-like format. Consequently, the elements of a decision tree are decision nodes, leaf nodes, and branches. For predicting the target value of an arbitrary input vector, each decision node is labeled with a specific feature of the vector. The outgoing branches of a decision node represent a decision rule for this specific feature. The branches either lead to other decision nodes or a leaf node labeled with a target value or a probability distribution over all possible target values. For illustration, consider an example of a toy classification problem predicting wine quality as depicted in Fig. 2.6.

Decision trees are generated by identifying data set attributes that split observations so that the resulting subsets are as distinguishable as possible. For identifying the attributes that are best for making splitting decisions, the expected information gain from each attribute with regard to the target variable is calculated. In information theory (Shannon, 1948), the expected information gain (IG) is the reduction of the information entropy H, i.e. a measure of uncertainty of the decision problem. Entropy is a fundamental concept for indicating uncertainty or loss in machine-learning, and can be formally described given a discrete random variable X, with possible outcomes $x_1, ..., x_n$, which occur with probability $\mathrm{P}(x_1), ..., \mathrm{P}(x_n)$ as follows:

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log \mathrm{P}(x_i) \tag{2.1}$$

The decision tree learning algorithm uses the relative entropy between the parent node and its children for selecting the best attributes as decision nodes. This is done recursively in a top-down fashion until no more information gain can be achieved. A more detailed overview of decision trees can be found in Breiman et al. (2017).

For improving the prediction performance, ensembles of decision trees are used, called *random forests* (Breiman, 2001). The basic idea of random forests is that a large amount of relatively uncorrelated trees that operate together are more accurate than any singular tree. For modelling random forests, there exist two methods: bagging and boosting (see Fig. 2.7:

Bagging: Using this approach the trees are built independently and in parallel. For building the forest, each tree is trained using random sampling with replacements. Furthermore, random sub samples of the input features are used by each tree for generating the decision nodes as previously described. Each tree is grown to the largest extent possible. No pruning, i.e. removing sections of the tree that have redundant information, is applied. The final prediction is based on the results of all decision trees. For regression tasks, the average of the decision trees' values is used. Majority voting is used in case of a classification problem.

Boosting: This method implies combining multiple weak learners (single decision trees) to a strong learner. The main principle of this method is to sequentially build trees that use information about the prediction errors their predecessors made and apply pruning. Using this information the performance of the subsequent models can be iteratively increased. For example, the decision tree learning algorithms Adaboost (Freund et al., 1996) and extreme gradient boosting (XGB) (Chen and Guestrin, 2016) make use of this method. Boosting usually leads to better prediction results as ensembles using the Bagging approach and is hence favoured in ML.

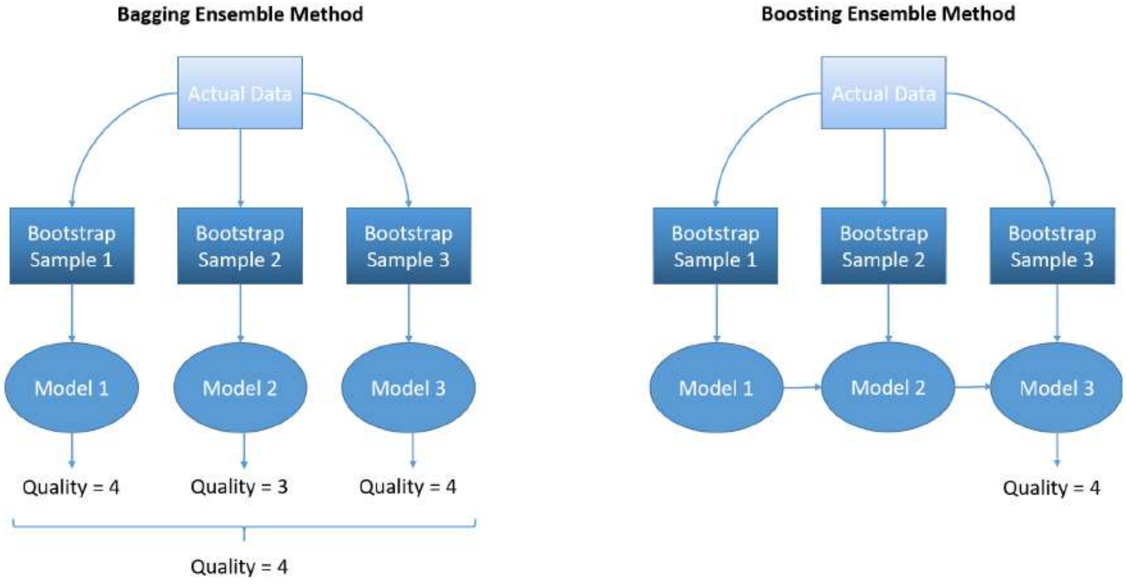


Figure 2.7.: Boosting Methods.

As only the XGB-algorithm is used in the scope of this thesis, a few details of its model specification are provided. XGB is a tree-boosting ML model based on ensembles of decision tree using K additive functions to predict the output based on the prediction outcome f_k of each decision tree:

$$\hat{y_i}^{(t)} = \sum_{k=1}^{K} f_k(x_i) = \hat{y_i}^{(t-1)} + f_t(x_i),\ \ f_k \in F \tag{2.2}$$

where x_i is the i-th feature vector, F is a collection of k trees, $\hat{y_i}$ is the predicted target value, and t is the amount of training rounds. For training the model an objective function equalling the sum of a loss function L and a regularisation term Ω is optimised.

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{2.3}$$

l measures the predictive power of the model between the real target value y_i and $\hat{y_i}$, while the regularisation term Ω measures the complexity of trees and reduces overfitting for stable prediction.

XGB uses iterative gradient descent for optimising the objective function in an additive manner. This is conducted by adding new trees at each training round. The trees are built using an exact greedy algorithm. The algorithm iteratively finds the optimal splitting of a decision tree by using residuals, i.e. the difference between the predicted and true values, and measuring the gain of each split. The tree is then pruned using the regularisation term Ω for enhancing the generalisation of the prediction.

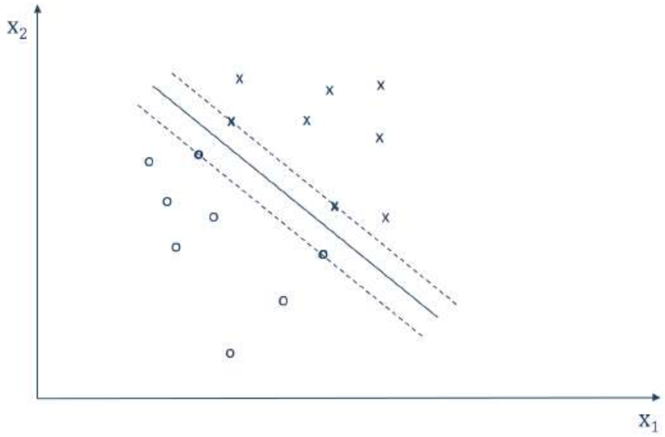


Figure 2.8.: Support Vector.

2.2.3. Support Vector Machine

Another popular algorithm for supervised prediction tasks are SVMs, introduced by Vapnik (2000). They have shown to be excellent at solving various problems such as digit recognition, computer vision, and text categorization (Kecman, 2005). In the following, the basics of the SVM are explained based on the books by Kecman (2005) as well as Steinwart and Christmann (2008).

Typically for supervised learning problems, information in the form of a data set consisting of a high-dimensional input vector $\mathbf{x}$ and output labels y. The data set can be described as $D = \{\mathbf{x_i}, y_i \in X \times Y\}, i = 1, l$, where l represents the amount of training data pairs. Furthermore, the underlying probability distributions are unknown, which requires the training process to perform distribution-free learning. The goal of an SVM is to find a linear separating hyperplane that can predict as best as possible the output label of an unknown input vector. For this, the learning problem is stated to find a non-linear mapping function $y = f(\mathbf{x})$.

For explaining the operating principle of an SVM, a binary classification problem is considered y is a scalar value. In the case of multi-class SVM, y would be represented as a vector.

In a binary classification problem, a set of training examples $(\mathbf{x_1}, y_1), ..., (\mathbf{x_n}, y_n)|\ \mathbf{x_i} \in \mathcal{R}^n, y_i \in \{-1, 1\}$ is used to create a hyper-plane separating two classes $\{-1, 1\}$ with the goal to maximise the margin between samples of each class. For simplifying visualisation, only a two-dimensional input space, i.e $\mathbf{x_i} \in \mathcal{R}^{\in}$, is used as an example. In Fig. 2.8, two-dimensional training samples as well as the trained linear separating hyperplane are illustrated. The hyperplane separates instances of two classes 1 (illustrated as circles) and -1 (illustrated as crosses). The exact position of the optimal hyperplane is determined by the instances of each class that are nearest to the plane. These instances are called support vectors (drawn in bold). Optimal, in this sense, implies that the model generalises well on unseen data, i.e. the hyperplane is able to correctly classify new data as best as possible.

The separating hyperplane can be mathematically described as a decision function:

$$d(\mathbf{x}, \mathbf{w}, b) = w^T x + b = \sum_{i=1}^{n} w_i x_i + b = 0 \tag{2.4}$$

where w denotes the normal vector of an input instance and its offset bias b. To classify an unknown sample the following decision rule is applied:

$$\hat{y} = sgn[d(\mathbf{x}, \mathbf{w}, b)] = \begin{cases} +1, & d(\mathbf{x}, \mathbf{w}, b) > 0 \\ -1, & d(\mathbf{x}, \mathbf{w}, b) \leq 0 \end{cases} \tag{2.5}$$

Depending on the position of the training sample to the hyper-plane, class 1 or -1 is assigned to the unknown sample. Multi-class problems are solved by reducing the problem to several binary classification problems according to a one-vs-one scheme. For finding the optimal hyperplane among a possibly infinite set of planes that linearly separate the classes, the margin between the hyperplane and the support vectors of each class needs to maximised. For indicating the distance between a support vector and the hyperplane, parallel canonical hyperplanes are used. These can be mathematically described as $w^T x + b = 1$, respectively $w^T x + b = -1$. Thus, the largest margin between the canonical hyperplane can be described as $M = \frac{2}{\|\mathbf{w}\|}$. In order to maximise this margin, the norm of the separating hyperplane's normal weight vector $\|\mathbf{w}\| = \sqrt{(\mathbf{w^T w})}$ needs to be minimised. As a result, the following objective function needs to be minimised:

$$Obj = \frac{1}{2}\mathbf{w^T w} \tag{2.6}$$

under the constraint that each input data point must lie on the correct side of the margin. This can be described as

$$y_i(\mathbf{w}^T \mathbf{x}_i - b) \geq 1, \ \forall \ 1 \leq i \leq n \tag{2.7}$$

For solving this problem, the saddle point of the Lagrange functional can be used stochastic gradient descent can be used. However, the mathematical description of these methods is out of the scope of this thesis.

SVMs have been designed to solve linearly separable problems. However, in real-world scenarios, the distributions of the input features are too complex to be classified using linear separation. For solving such non-linear problems, the so-called *Kernel-Trick* is applied. Here, a non-linear kernel function is used to transfer the input features to a higher dimension until the data is linearly separable. The original features space can be transformed using various non-linear kernels. These comprise:

- Polynomial kernel, with a variable polynomial degree d: $K(x, x_i) = ((x \cdot x_i) + 1))^d$
- Gaussian radial basis function: $K(x, x_i) = \exp(-\gamma \|x - x_i|^2)$ for $\gamma > 0$
- Sigmoid function or hyperbolic tangent: $K(x, x_i) = \tanh(\kappa x \cdot x_i) + 1 + c)$ for some (not every) $\kappa > 0$ and $c < 0$

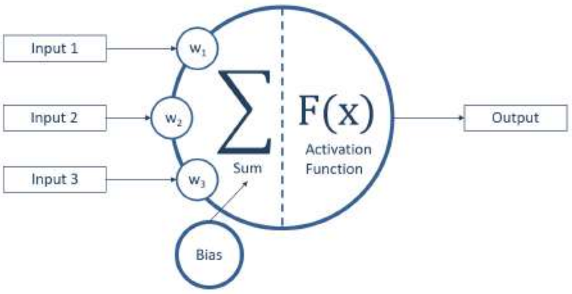


Figure 2.9.: Artificial Neuron.

2.2.4. Artificial Neural Networks

Due to the recent advances in computational processing power and the availability of big data, information processing using ANN has received wide recognition. Their learning capabilities and flexibility have made them popular for solving various tasks including pattern recognition, classification and regression, and function approximation. The elementary component of ANNs is artificial neurons, which are inspired by the biological neurons of the mammal nervous system (Rosenblatt, 1958). Biological neurons are cells that transmit information to other neurons using electrical and chemical signals. For allowing this kind of communication, a biological neuron possesses dendrites for receiving information and uses a "sum-and-threshold" (Bishop et al., 1995) method for producing an output via the neuron's axons. These axons are connected to the dendrites of other neurons, thus creating a dense network for information processing. Especially, the adoption of a simplified "sum-and-threshold" method for artificial neurons has proven to be useful for determining optimal discriminating functions (Bishop et al., 1995). Mathematically, the operating methods of a neuronal processing unit can be described as follows:

$$\hat{y}(k) = F\left(\sum_i w_i(k) \cdot x_i(k) + b\right) \tag{2.8}$$

with $\hat{y}(k)$ denoting the output of the k-th neuron. The output is calculated based on a non-linear activation or transfer function. The function's input is the sum of the data features of an input vector $x_i(k) \in \mathbf{x}$, that are individually weighted by multiplying a factor $w_i(k)$ and a neuron specific bias b. The components of an artificial neuron are illustrated in Fig. 2.9. The activation function can take various forms depending on the problem at hand. In the original paper about the multi-layer perceptron (MLP) by Rosenblatt (1958), one of the first considering neural networks, a threshold or step function is used that takes the value 1 if the input exceeds a certain value, while outputting 0 otherwise. Nowadays, usually rectified linear units, sigmoid, and hyperbolic tangent functions are used (Nielsen, 2015). For generating predictions or to make classifications, a high number of neurons are interconnected creating an ANN. Based on the books by Nielsen (2015) and Bishop et al. (1995), the learning problem and different structures of an ANN are explained in the following.

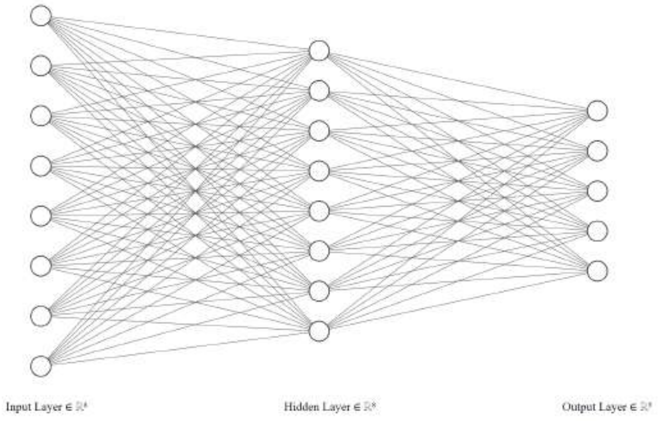

Input Layer ∈ $\mathbb{R}^8$
Hidden Layer ∈ $\mathbb{R}^8$
Output Layer ∈ $\mathbb{R}^5$

Figure 2.10.: Neural net.

The typical structure of an ANN is arranged using three different kinds of layers:

Input Layer: This layer poses the interface between input features and the network and has the same size as the input vector. For example, if the data vector contains eight entries, then the input layer would consist of eight neurons.

Hidden Layer(s): As an intermediate layer between input and output, there may exist one or more deeply connected hidden layers. Deeply connected implies that neurons of the hidden layers receive the outputs of all neurons of the previous layer and transmit their output to all neurons of the next layer.

Output Layer: The output of an ANN is provided by an output layer. The number of neurons of this layer and their activation functions are selected dependent on the problem to solve. For example, one output neuron is often sufficient for solving a regression or binary classification problem. For classification tasks the number of output neurons equals the number of classes. For classification tasks, the activation function of the output layer is often a softmax function. Softmax represents the output as a probability distribution, as the sum of all output activations equals 1, and individual outputs are valued between 0 and 1. This allows to interpret the network's output as its estimate of the probability that it is correct.

In the simplest configuration, the output of each neuron in a hidden layer is conveyed as input to the neurons of the succeeding layer. This structure is known as a *feed-forward network*, where information flows in one direction, from the input to the output. The structure is depicted in Fig. 2.10. As usual for supervised learning problems, ANNs are trained by adjusting its parameters for minimising an objective or loss function. For this, usually, the quadratic cost function, or mean squared error (MSE), and the cross-entropy function are used. These functions measure the difference between the annotated "correct" output and the network's estimated output.

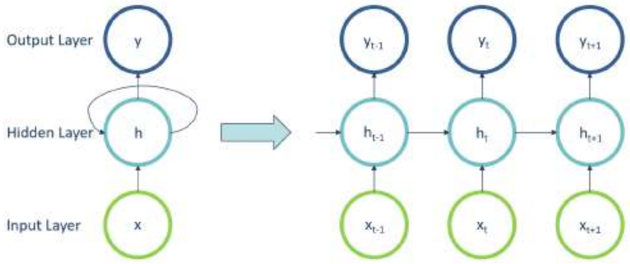


Figure 2.11.: Recurrent neural net.

The MSE can be described mathematically as

$$Obj = \frac{1}{2n} \sum_{i}^{k} (y_i - \hat{y_i})^2 \tag{2.9}$$

and the cross-entropy can be noted as

$$Obj = \frac{1}{n} \sum_{i}^{k} [y_i \; ln \; \hat{y_i} + (1 - y_i) \; ln \; (1 - \hat{y_i})] \tag{2.10}$$

with n being the number of vectors in the training data, k the number of output neurons, y_i the correct output value, and $\hat{y_i}$ and networks predicted output value. For minimising the network's error, and thus its objective function, e.g. stochastic gradient descent and error back-propagation (Rumelhart et al., 1986)can be used. In doing so, the weights w and bias b are automatically trained. In principle, the training of an ANN works as follows: First, weights are initialised usually using small random numbers. Afterwards, the network calculates an output based on some input data and measures the error. Subsequently, the error is propagated backward through the network using gradients to determine the influence of different weights on the error function. Finally, a learning algorithm, such as stochastic gradient descent, can be used for updating the network's parameters. The network is trained iteratively through multiple epochs and usually using mini-batches of the training sample. For training ANNs, several so-called hyper-parameters can be adjusted to improve the performance of a network. This includes a learning rate α, number of epochs, the size of the mini-batches, and using different regularisation methods (e.g. l_2 regularisation, dropout) for making the ANN better at generalising to unseen input data. Furthermore, the number of layers and neurons, as well as different kinds of activation functions can be used. For tuning these hyper-parameters, there exist only heuristics up to this date. Therefore, finding the correct configurations of a network is quite complex and usually, this process is automated using greedy search methods, e.g. grid search.

Another interesting variant of ANNs are RNN that includes dynamic changes over time in their model. Here, the basic idea is that a neuron not only feeds forward information but also has a connection of its output back it its input. Therefore, they are called recurrent networks, as there exists a backward information flow, i.e. information from the

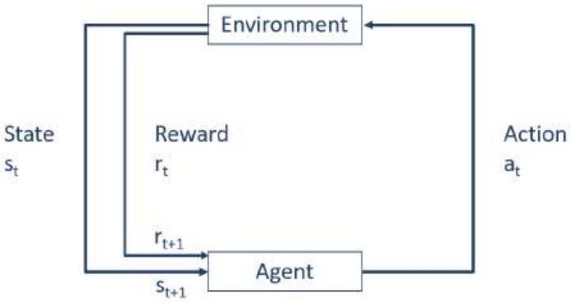


Figure 2.12.: Reinforcement learning environment.

past can be used in the current process. This allows the model to store information over time, and to create a kind of memory for providing context. For this reason, RNNs are particularly useful for learning temporal and sequential dependencies. Fig. 2.11 illustrates the topology of an RNN, which can be unfolded for representing temporal relations. For training RNNs, a variant of back-propagation, called "back-propagation through time" is used.

A major drawback of RNNs, however, is that it can only make use of limited context, as back-propagation through time fails for longer sequences. This is due to the either vanishing or exploding outputs of the network neurons, which is also named the vanishing or exploding gradient problem (Bengio et al., 1994). To remedy this problem , specific neurons such as gated recurrent unit (GRU) (Chung et al., 2014) and long short term memory (LSTM) cells (Hochreiter and Schmidhuber, 1997) can be applied. Both neuron structures use a gating mechanism for updating and resetting their memory. This allows to store or forget specific information over time, which also prevents the gradients form exploding. LSTMs and GRUs provide comparable performance but a GRU-based networks are significantly faster to compute.

2.2.5. Reinforcement Learning

In the following, the concept of RL is briefly described based on the books by Sutton and Barto (2018) and Szepesvári (2010). RL is a sequential decision-making problem. An agent interacts with an environment by taking actions in different situations or states, As an effect the agent receives a positive or negative reward and transitions to a new state, where the process starts over again. A reward r_t is a scalar feedback value that indicates how good the agent is doing at time step t. The goal of RL is for the agent to find an optimal policy (sequence of actions) $\pi(s|a) = P(A = a|S = s)$ that maximises the expected discounted rewards R_t, also noted as return. For achieving the goal, the agent learns by trial-and-error under usage of the environment's feedback (reward) based on its own actions and experiences. The RL problem is depicted in Fig. 2.12. It can be formally modelled as an MDP. An MDP can be described mathematically as tuple (S, A, P, R, γ) where S is a finite set of world states and A denotes a finite set of actions an agent can

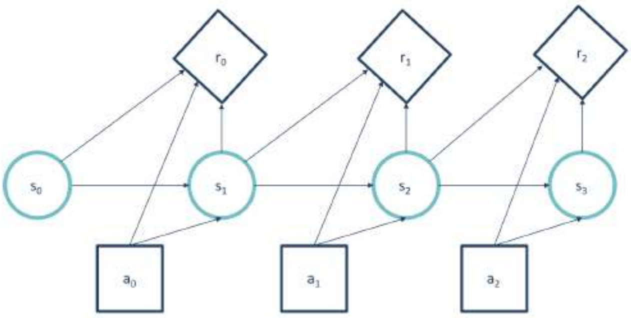


Figure 2.13.: Markov decision process.

execute. P describes the transition probability function $P^a_{ss'} = P[s_{t+1} = s'|S_t = s, Aa_t = a]$ determining the probability of transitioning to state $s' \in S$ after taking action $a \in A$ in state $s \in S$. R is the reward function $R^a_s = E[R_{t+1}|S_t = s, A_t = a]$ which describes the expected next (immediate) reward. γ is a discount factor ranging between 0 and 1, that affects the importance of future rewards. A fundamental aspect of MDPs is the Markov property $P[S_{t+1}|S_t] = P[S_{t+1}|S_1, ..., S_t]$ indicating that the current agent state contains all relevant information about the interaction history. The learning objective is to maximise the total discounted reward for time step t

$$Obj = G_t = R_{t+1} + \gamma R_{t+2} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2.11}$$

Fig. 2.13 illustrates an MDP. As previously mentioned, for maximising the reward an RL-agent needs to find an optimal policy. Therefore, it is required to estimate the expected return starting from state s, and subsequently following policy π: $v_\pi(s) = E_\pi[G_t|S_t = s]$. This function is called *state-value function* and is measure for how good it is to for an agent to be a in specific state following a specific policy. Similarly, it can be measured how good it it is to take a specific action in a given state, and then following a particular policy. This is denoted as *action-value function* or Q-function: $q_\pi(s, a) = E[G_t|S_t = s, A_t = a]$. Both value functions can be decomposed into recursive formulas called Bellman equations. This property is fundamental for solving MDPs, as they designate the relationship between the value of a state and its successive states. Mathematically, both Bellman equations can be described as

$$v_\pi(s) = E_[R_t + \gamma v_\pi(s_{t+1})|S_t = s] = \sum_{a \in A} \pi(a|s) q_\pi(s, a) \tag{2.12}$$

$$q_\pi(s, a) = E[R_t + \gamma q_\pi(s_{t+1}, a_{t+1})|S_t = s, A_t = a] = r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a) v_\pi(s') \tag{2.13}$$

An optimal action-value function describes the best possible action selection in a MDP, as it is the maximum action-value function over all policies $q_*(s, a) = \max_\pi q_\pi(s, a)$. Using the Bellman equations, the optimal value function can also be expressed in terms of the optimal action-value function $v_*(s, a) = \max_a q_*(s, a)$. Optimal policies can be found by solving the Bellman equations. However, there exists no closed-form solution. Therefore, iterative methods need to be applied. Here, it is distinguished between model-based and model-free methods. In realistic scenarios, the agent usually has incomplete knowledge about the dynamics of the environment, i.e. the state transition probabilities are unknown. For this reason, we only describe model-free methods in the scope of this thesis. In case a model is not available, it is essential to estimate the action-value function q_*. Without a model, the state-value function v_* is not sufficient for determining a policy. Thus, it is required to estimate the value of each action for making inferences about the best policy. Model-free RL comprises two phases, also known as General Policy Iteration: Prediction, i.e. estimate the action-value function and evaluate the current policy of an MDP, and Control, i.e. optimise the action-value function and find the best policy. The optimal value functions can be learned by sampling experience in realistic or simulated environments. Here, primarily two approaches are used to learn an optimal policy: monte carlo (MC) learning and temporal difference (TD) learning. The MC-method samples and averages the return for each state-action pair for estimating the value function. According to the law of large numbers, the empirical mean of these estimates converges to the true expected value as the number of samples increases. The estimate is updated here after observing a complete episodes of training samples based on the actual return. Contrary, TD uses bootstrapping, i.e. the estimate of the value function of a specific state s is updated based on the estimate after subsequent n-steps of an episode. Thus, the estimation is updated towards an estimated return instead of the actual return. For this reason, TD learning is usually more efficient than MC and is not restricted to episodic problems which require terminal states.

Both approaches of RL suffer from the exploration and exploitation dilemma. The agent has to compromise between exploration for finding new information about the environment and exploitation which uses the information to maximise the reward. For finding new, potentially better policies, continual exploration needs to be ensured. For this, *ϵ-greedy exploration* is usually applied. There, a greedy action, i.e. an action that maximises the state-action value, is chosen with probability 1 - ϵ. Otherwise, with probability ϵ a random action is chosen. In doing so, the policy can be continuously improved.

For learning an optimal strategy, it can also be differentiated between two learning control mechanisms: *On-policy* and *Off-policy* methods. Using on-policy methods, the policies used for Prediction and Control are the same. Contrary, off-policy methods make use of two separate policies. A *target policy* $\pi(a|s)$ is evaluated for computing $q_\pi(s, a)$, while actions are selected following a *behavioural policy* $\mu(a|s)$. This allows learning an optimal target policy using an exploratory behaviour policy. Off-policy methods are more powerful and lead to better generalisation, due to their ability to learn from old policies and different sources, e.g. humans or other agents. For both methods there exist various MC and TD algorithms. Due to the limited scope of this thesis, we only consider the Off-policy TD algorithm Q-learning algorithm in the following. Applying Q-learning, the next action is selected by sampling from the behavioral policy $A_{t+1} \sim \mu(\cdot|S_t)$, while an alternative action sampled from the target policy $A' \sim \pi(\cdot|S_t)$ is used for evaluation.

Consequently, the action-values are updated towards the value of the alternative action

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma Q(S_{t+1}, A') - Q(S_t, A_t)) \tag{2.14}$$

For finding an optimal policy, both behaviour and target policies are improved. While the target policy is greedy $\pi(S_{t+1}) = \max_{A'} Q(S_{t+1}, A')$, the behaviour policy is ϵ-greedy to allow for exploration. Thus, the update equation can be simplified to

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \max_{A'} \gamma Q(S_{t+1}, A') - Q(S_t, A_t)) \tag{2.15}$$

After training, The optimal strategy can then be directly derived, as $\pi(a|s)^* = \max_A Q(s, a)$. In Q-learning, the action-value function is represented by a lookup table, where each state-action pair has an entry $q(s, a)$. However, for large MDPs with a high number of states and actions it is impractical to learn the value of each state-action value individually. Therefore, generalisation from examples need to applied in order to infer from seen states to unseen states. For this, function approximation methods are applied. For example, ANNs can be used as a function for mapping states to action-q-value pairs. The combination between neural networks as function approximators and Q-learning is called a deep-q-network (DQN) (Mnih et al., 2015), which we briefly describe in the following.

Here, a neural network is used to approximate the action-value function $Q(s, a; \phi) \approx Q * (s, a)$, with ϕ denoting the network's weights. The network's structure consists of an input layer with the size of the state features, one or more hidden layers, and an output layer that comprises neurons equalling the number of possible actions in the MDP. Similarly as described in the previous section, the goal is to update the weights of the network to optimise an objective function. Concerning Q-learning, the objective function for DQNs can be mathematically described as

$$Obj(\phi_i) = E_{s,a,r,s' \sim D_i} \left[\left(r + \gamma \max_{a'} Q(s', a'; \phi_i^-) - Q(s, a; \phi_i) \right)^2 \right] \tag{2.16}$$

reducing the MSE in the Bellman equation. Further, $e_t = (s_t, a_t, r_t, s_{t+1})$ are the agent's experiences at each time step following a ϵ-greedy behavioural policy for exploration. The experiences are stored in a data set $D_t = (e_1, ..., e_t)$, also known as the replay buffer.

For training the DQN, mini-batches of the replay buffer are randomly sampled. This approach is called *experience replay* and is more data efficient and allows to stabilise the learning of networks parameters. Furthermore, a separate network for generating the target Q-values is used in the Q-learning update. Usually, some older network with the weights ϕ_i^- is used for generating the target values following an greedy policy. The weights of this network are kept fixed for a specified number of time steps to ensures the stability of the training process. For updating the weights and to finally learn the optimal Q-value function a variant of the stochastic gradient descent algorithm is used.

2.2.6. Evaluation Methods

For measuring and comparing the performance of different machine-learning algorithms, evaluation metrics are necessary. Depending on the problem and used algorithm, it can

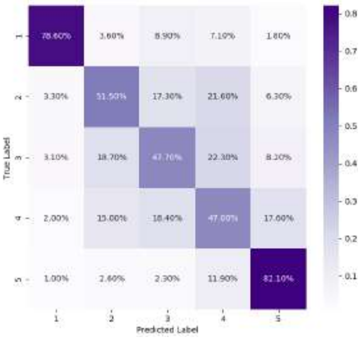


Figure 2.14.: Example of a confusion matrix.

be chosen from a variety of metrics. As only supervised learning for classification and RL are considered in the scope of this thesis, we limit the description of evaluation variables for these approaches. RL in the DS domain is used to find optimal dialogue policies or strategies based on hand-crafted rewards. For this reason, the result of RL is a dialogue strategy and hence can be evaluated using the typical dialogue evaluation metrics as described in the previous section. For supervised learning approaches considering classification, standard metrics are derived from a so-called confusion matrix. The confusion matrix counts correct and incorrect predictions of the classifier concerning the true class labels. The matrix rows denote the predicted values, while the columns correspond to the actual classes. Fig 2.14 illustrates an exemplary confusion matrix. Several confusion matrix-based metrics are explained in the following. Another important aspect of evaluating machine-learning methods is generalisation.

For training a classifier a training data set is used. Consequently, the classifier is fit to this specific data set with the goal to minimise the prediction error, i.e. to minimise the number of false predictions. For this reason, evaluating algorithms on the same data set which was used for training does not reveal anything about its true performance. Therefore, algorithms are trained and tested on distinct data sets, named training and test data set. This allows evaluating the performance of the classifiers on previously unseen data for creating an objective and unbiased measurement. In doing so, the generalisation of an algorithm can be established. In the following, different metrics for measuring the classification performance and the most important approach for generalisation *cross-validation* are described.

Metrics

One of the most used metrics for evaluating classifiers is *accuracy.* Formally, it can be described as follows based on a confusion matrix

$$accuracy = \frac{\sum_{i=1}^{K} TP(C_i)}{\sum_{i=1}^{K}\sum_{j=1}^{K} C_{i,j}} \tag{2.17}$$

with K being the number of classes, TP being the *true positive* classification, i.e. the prediction was correct, C_i indicating the confusion matrix entries for class i, and $C_{i,j}$ denoting the confusion matrix entry at the respective row and column. Usually, accuracy works very well for categorical class labels, that have no natural order. However, in the case of ordinal classes, the distance between the wrong prediction to the real class is important which is not represented in the accuracy metric. Therefore, an extended accuracy (eA) measure can be computed for ordinal class distributions (Rach et al., 2017). Here, the amount of guesses in which the classification was wrong only by one class is computed. The percentage δ of these guesses in relation to the total amount of class-wise occurrences can be derived directly from the confusion matrix C. Adding this value to the accuracy gives a percentage of usable predictions of the classifier $eA = accuracy + \delta$ with

$$\delta = \frac{1}{N}\left(\sum_{k=1}^{K-1} C_{k,k+1} + \sum_{k=2}^{K} C_{k,k-1}\right) \tag{2.18}$$

with N the number of total entries of C and K the number of classes, i.e. the dimension of C. For imbalanced data sets, i.e. one or more classes appear significantly less in the data set than other classes, the accuracy measure can be misleading. In such cases, other metrics are more reliable. For example, precision (P) may be used for measuring the proportion of correct classifications of a specific label to all samples that are classified as the respective label. Based on the confusion matrix, precision can be defined as follows

$$P = \frac{1}{K}\sum_{i=1}^{K} \frac{TP(C_i)}{TP(C_i) + FP(C_i)} \tag{2.19}$$

with K being again the number of classes, TP being the true positive classification of class C_i , and FP being the *false positive* classification of the respective class. Further, unweighted average recall (UAR), i.e. the arithmetic average of all class-wise recalls, may be used

$$UAR = \frac{1}{K}\sum_{i=1}^{K} \frac{TP(C_i)}{TP(C_i) + FN(C_i)} \tag{2.20}$$

with all the previously introduced variables being the same and FN being the *false negative* classification of the respective class, i.e. samples that were falsely labeled to belong to the specific class.

In addition, the F_1-score, i.e. the harmonic mean between precision and recall, may be used

$$F_1 = 2 \cdot \frac{P \cdot UAR}{P + UAR} \tag{2.21}$$

Furthermore, *linearly weighted Cohen's* κ, and *Spearman's* ρ were used as evaluation metrics. Cohen's Kappa κ (Spitzer et al., 1967) measures the relative agreement between two sets of ratings and is defined as

$$\kappa = \frac{p_{a(w)} - p_{s(w)}}{1 - p_{s(w)}} \tag{2.22}$$

where $p_{a(w)}$ is the observed agreement, and $p_{s(w)}$ is the chance agreement. Hence, $\kappa = 1$ for perfect agreement and $\kappa = -1$ for perfect disagreement. However, this version of Cohen's Kappa does not work well for ordinal scaled class labels, as the disagreement between the labels is not weighted accordingly, i.e. the distance between the disagreed labels is not reflected. Therefore, Cohen introduced a linearly weighted version of Cohen's κ. Here, three matrices are used for the measurement.

$$\kappa = 1 - \frac{\sum_{i=1}^{K} \sum_{j=1}^{K} w_{ij} x_{ij}}{\sum_{i=1}^{K} \sum_{j=1}^{K} w_{ij} m_{ij}} \tag{2.23}$$

with k being the number of classes, C_{ij} being a particular entry of the observed confusion matrix, w_{ij} being the weighting factor between label i and j in the weights matrix, and m_{ij} being the respective entry in the expected matrix based on chance agreement.

Spearman's rank correlation coefficient Rho ρ is a non-parametric measure for the rank correlation between two variables and describes how well one variable can be expressed by the other (Spearman, 1904).

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{2.24}$$

where d represents the pairwise distances of the ranks of the observations x_i and y_i and n is the number of samples. Thus, $\rho = 1$ if observations have identical ranks and $\rho = -1$ if observations have fully opposed ranks.

Cross-Validation

Usually, supervised learning classifiers are trained on a limited amount of labeled data that only represent a small-scale model of the "true world". For being applicable in realistic environments, the predictor must be able to handle previously unseen data observations, otherwise, it will fail. Therefore, testing on unseen data is crucial for testing the performance of a classifier. The goal of supervised learning is to perform the best possible on such test data, as it can be concluded that the model will also be able to generalise. The arguably best way to obtain better generalisation and less biased results, especially for smaller data sets, is to apply m-fold cross-validation. Here, the data set is split equally into m disjoint sets. Typically, splits of size $m = 5$ or $m = 10$ are used. For classifier training, one subset is selected as test set, while the other $m-1$ subsets are used for training. After training, the classifier is tested on the chosen subset. Subsequently, a different subset as before is selected as test data and the cycle is repeated. This kind of data rotation stops after all m subset have been selected as test data. The final performance value of the trained model can then be calculated by taking the performance averages of each fold. There also exists stratified m-fold cross-validation approaches for imbalanced data sets to ensure that the minority classes are available in all subsets.

2.3. Psychological Models for Human-Computer Cooperation

Human factors play a fundamental role in human-computer cooperation as computer-based assistants aim to work alongside humans. Therefore, it is necessary to understand how humans are affected by CAs, and which cognitive concepts and processes are involved during dialogue with a CA. In the following, we present five concepts that are relevant for the work presented in this thesis.

2.3.1. Theory of Mind

For cooperating adequately and advancing towards problem-solving in the interest of users, a CA needs to understand to some extent how they “think” and “feel”. In doing so, a computer may be able to predict user behaviour. This allows deciding whether the initiation of a dialogue can be helpful or not. Theoretically, a CA might infer, for example, user confusion from specific cues and the interaction history which allows it to deduce that the user is possibly looking for help. Here, a system could exploit this knowledge to act in advance and show goodwill. This in turn could foster trust in the user towards the system as we explain in the next section. The development of trust can subsequently be used as a positive feedback and allows the system to evaluate and adapt its own behaviour. However, cooperation is dyadic process. Thus, the user also needs to be able to understand a computer’s “thinking” in order to form expectations and be able to develop trust in its behaviour for a successful interdependence. In psychology, the concept of understanding and predicting the inner processes of others is understood as *TOM*.

According to the definition by Margolis et al. (2012) “Theory of mind refers to the cognitive capacity to attribute mental states to self and others”. TOM is also often referred to “mindreading” or “mentalising”, for example. Generally, mental states include perceptions, bodily feelings, emotional states, and propositional feelings (beliefs, desires, intentions). The attribution of these states can be made verbal as well as non-verbal and are an important aspect of social life (Cuzzolin et al., 2020). For understanding this social phenomenon, its underlying processes are investigated. Here, the main question is, how do cognitive systems form beliefs or judgments about others’ mental states that are not directly observable. According to social cognitive research there exist two main theories for explaining this:

Theory-theory: Theory theorists argue that psychologically competent humans understand and predict thought and action by using implicit knowledge, a so-called folk-psychological TOM (Carruthers and Smith, 1996). This knowledge is a theory one has for attributing mental state concepts to oneself and others. The theory is developed by oneself using causal-explanatory generalisations. This implies that humans create generalisations about the usage of mental concepts by making observations through interaction with their environment. These generalisations can be understood as a set of rules that map observable input to certain mental states, mental states to other mental states, and mental states to observable outputs (Margolis et al., 2012). These rules are stored in a kind of mental module for making inferences about the mental state of oneself and others.

Simulation-theory: Another approach of TOM considers the simulation of mental states, also called empathy theory (Gordon, 1986). Here, the process of predicting the behaviour by others is to simulate their mental states by trying to create similar mental states of their own as surrogates of the others. Thus, this process can also be described as "stepping in one shoe", where humans use their minds to model another's for predictions (Margolis et al., 2012). Thus, mental concepts are simulated, by imagining the situation of others and consequently generating the thoughts or actions attributed.

Even though the role of TOM in consciousness is unclear and controversial (Carruthers and Smith, 1996), it has drawn lots of interest from the AI community. For example, Cuzzolin et al. (2020) argue that an AI with the capacity of having a TOM would greatly increase trust in such a system, for the reason we outlined before. Therefore, several researchers have investigated a machine's TOM (Rabinowitz et al., 2018). They propose that an artificial agent could learn autonomously how to model other agents using limited data. For accomplishing this, e.g. inverse RL (Abbeel and Ng, 2004), Bayesian TOM (Baker et al., 2011) or game theory (Yoshida et al., 2008) approaches have been investigated. Most prominent is the work by Rabinowitz et al. (2018), who let an ANN learn to predict the future behaviour of previously unseen deep RL agents by using behavioural traces of a high number of different agents for training. This process, which they label as "meta-learning", allows them to bootstrap predictions about agents' characteristics and mental states. In experiments, they showed that their framework passes classic TOM tasks such as the "Sally-Anne" test (Baron-Cohen et al., 1985) of recognising that others can hold false beliefs about the world. Passing the "Sally-Anne"-test is widely believed as the core requirement for the manifestation of an TOM, i.e. the ability to understand that others have their own beliefs that may not correlate with reality. Transferring these findings to the domain of CAs may allow predicting user behaviour and their internal states from a large amount of interaction observations. One of these states that is essential for successful human-computer cooperation is the concept of trust, which is described in the following.

2.3.2. Trust in Human-Computer Interaction

Trust is a fundamental concept in interpersonal relationships and has been extensively studied over the past 60 years by social and organisational scientists as well as psychologists. In one of the earliest works on trust in relationships, Deutsch (1960) defined trust as the ability of a party (the trustor) to have confidence in the actions of another person (the trustee) under the assumption that this decision may lead to either harmful or beneficial consequences. In his work, trust is described as the belief in another person's ability and the intention to produce a benefit, even though a violation of trust may have negative consequences to a greater extent than a trust fulfilment has benefits.

Rotter (1980) described trust as a stable personality trait and highlighted the benefits of high trust in the interaction between people. A theoretical model of trust in romantic relationships was provided by Rempel et al. (1985). Contrary to Rotter, they proposed trust to evolve out of experience and interaction. Hence, trust was seen as a dynamic variable. Based on this precondition, predictability, dependability, and faith are seen as the main aspects of trust. The predictability of a partner is determined among others by consistent and recurring behaviour. Dependability is defined by the partner's characteristics and qualities, e.g. attributes such as reliability and honesty. While predictability and dependability are mainly influenced by the partner, the third construct of faith is one's personal belief in the goodwill of their partner in face of an uncertain future.

Similarly, Mayer et al. (1995) investigated trust in organisational structures based on the three factors of perceived trustworthiness of another party: ability, benevolence, and integrity. Ability is related to another party's set of skills or competencies to complete certain tasks in a specific domain. The extent to which a trustee is believed to want to do good to a partner is described as benevolence. Integrity implies that a trustee follows a set of moral rules that a trustor finds acceptable. The authors specified trust to be a willingness to take risks and to be vulnerable to the actions of another party "based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party" (Mayer et al., 1995). Their model also includes an outcome loop, which compares the perceived trustworthiness and actions of a trustee for trust updates. For example, when the trustor takes a risky action that results in a positive outcome, the trustee's perceptions are enhanced. Trust can also be seen as a construct of cooperativeness, as trustor and trustee are interdependent (Simpson, 2007). The trustor needs to cooperate with the trustee to accomplish a certain goal and requires that the trustor believes in the reliability and helpfulness of the trustee.

The concepts of trust are transferable to interactions with computers and machines. Particularly, since computers are perceived as social actors and social rules are believed to be applicable in such interactions to some extent (Nass et al., 1994; Madhavan and Wiegmann, 2007). With the increasing progress in automation and conversational AI, machines could assist in complex task domains by providing guidance and advice. This makes humans vulnerable to the decisions of their programmed "partner" and thus a trustworthy human-machine relationship is indispensable. Otherwise, the system is possibly not accepted and becomes obsolete. Schaefer et al. (2016) describe the effect as the 'no trust – no use' principle. This is also stated in the earlier works on trust in automation by Muir (1987); Muir and Moray (1996), who hypothesise that independent of the "intelligence" or finesse of an autonomous system, users will reject a system when it is not perceived as trustworthy. In literature, this phenomenon is known under the term under reliance (Parasuraman and Riley, 1997; Lee and See, 2004). An example for this is the false alarm problem often occurring with fire detectors (Parasuraman and Riley, 1997). In case the false alarm ratio is too high, people may disuse the device. even though this could have negative consequences. Contrarily, over reliance in automation may lead to misuse because people may overestimate the competence of a system (Parasuraman and

Riley, 1997). Therefore, trust calibration is necessary, in which a user sets an appropriate trust level corresponding to the machines trustworthiness and uses it in accordance with its abilities and limits (Muir, 1987). Transferring the notion of trust to the CA domain, we understand the term trust in the scope of this thesis according to the definition provided by Lee and See (2004) as "the attitude that an agent will help achieve an individual's goal in a situation characterized by uncertainty and vulnerability". In the following, we present relevant trust models concerning HCI.

Models

Fundamental research on modelling trust has been done in the domain of autonomy research (Lee and See, 2004; Hoff and Bashir, 2015; Schaefer et al., 2016; Muir and Moray, 1996). However, CA can be considered as an autonomous system, hence human-autonomy models of trust may be transferable. For CA design, two models stemming from autonomy research are of particular interest. Schaefer et al. (2016) as well as Hancock et al. (2011) describe a model for categorising factors that influence trust in automation and robots respectively. In the following, we will refer to the Schaefer et al. model as it is an extension to Hancock's model and provides a more general view on trust in automation.

Schaefer et al. (2016) proposed three factors that are fundamental for modelling trust: the human, the autonomous partner, and the environment. Each factor has specific characteristics that influence the human-automation trust relationship. Considering the human element, individual traits, states, cognitive factors, and emotive factors play a decisive role in trust development:

Traits: There exists some evidence that age, gender, culture, and personality influence a human's trusting behaviour. However, the power and direction of these relationships have not been consistent across different experiments. An interesting human trait is a person's trust propensity. This trait defines how willing a human is to trust others or machines. Therefore, it can serve as a baseline for predicting the initial HCT level (Merritt and Ilgen, 2008; Merritt et al., 2013; Jian et al., 2000).

States: The user states stress, fatigue, and attentional control, i.e. the setting and flexibly shifting of focus, are highly correlated with the perception of the interaction with autonomous systems. There also exists a tendency that mood and affect directly influence trust development.

Cognitive Factors: These factors comprise a human's self-perceived ability to use automation, their technical understanding, and expectancy concerning the automation. A human's self-perceived ability to use automation depends on the degree of performance capability, workload, and system as well as domain expertise. The technical understanding of the system is influenced by the ease of learning the interaction, and prior experience with the same and similar systems. For example, as it has been shown that trust develops over time, experience directly influences the trust development (Merritt and Ilgen, 2008). Expectancy relates to the TOM and how a user predicts the system's perceived usefulness and the perceived benefits of the automation. Furthermore, the reputation of a system influences its trustworthiness

Emotive Factors: Attitudes towards the automation, confidence in the automation, as well as comfort and satisfaction with the automation also impact the human-automation trust relationship. For example, it has been shown that positive emotions, e.g. happiness, positively affect trust in a system.

Trust antecedents of the autonomous partner are system-specific features and capability related features:

Features: The features of an autonomous system comprise its level of automation, mode of communication, appearance or anthropomorphism, intelligence, and personality. The level of autonomy (LoA) is related to the amount of control a system exhibits and correlates to some degree with a system's proactive behaviour, which will be explained more in detail in Chapter 3. There exists evidence that there exists no simple correlation between trust and LoA. However, if a person finds a specific level as "good", this usually results in greater trust in the system. The mode of communication describes whether the system interacts via visual, auditory, or tactile cues. The appearance, anthropomorphism, intelligence, and personality of an autonomous system should be designed following its expected capabilities. Otherwise, there may exist a mismatch which could result in the system falling into the uncanny valley (Mori et al., 2012).

Capability: It has been found that especially an autonomous system's capabilities, e.g. competence, reliability, and predictability, are proven to have a large impact on the trust relationship (Muir and Moray, 1996; Muir, 1994; Lee and Moray, 1994). This involves the appropriateness of cues and feedback, effective communication, as well as consistent behaviour. Furthermore, a system's ability to behave adaptively has been shown to affect its trustworthiness.

Furthermore, environmental factors such as the task/context and team collaboration characteristics need to be considered:

Team Collaboration: In collaboration with autonomous systems, it has been shown that team composition (e.g., size, diversity, roles, and characteristics), allocation of roles, i.e. whether the system is in- or out-of-group, societal impact, and task interdependence are relevant to consider when designing for a trustworthy system.

Task/Context: Considering the task and context in which a human interacts with an autonomous system, the main factors influencing trust are the risk and uncertainty of the situation, as well as the task type respectively the task difficulty.

Including these factors, Hoff and Bashir (2015) presented a three-layered model of trust. The layers and interaction between the different layers are depicted in Fig. 2.15. Dispositional trust represents a user's long-term tendency to use an autonomous systems dependent on individual characteristics. The other two layers, situational and learned trust are

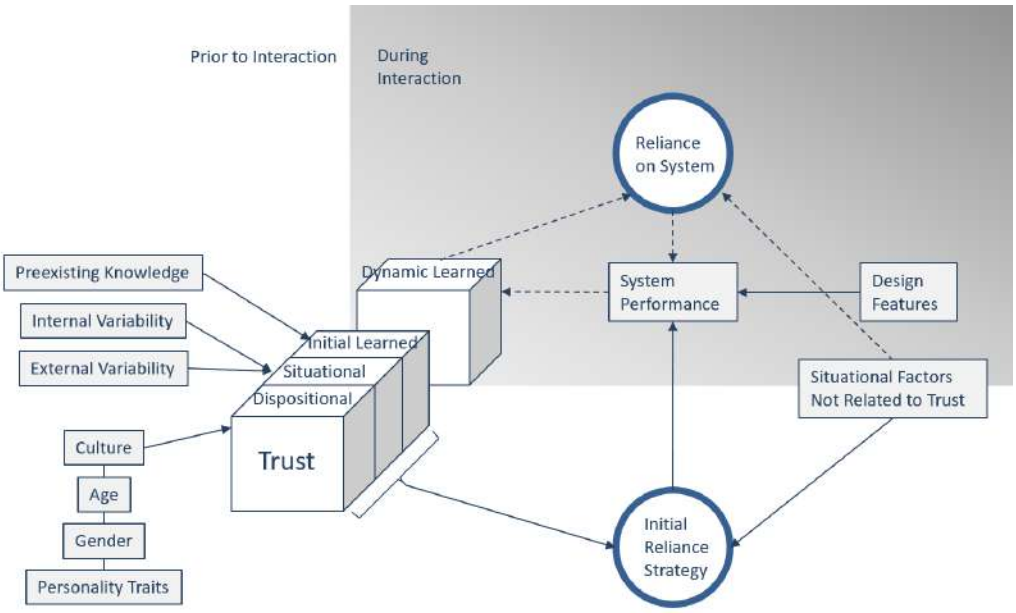


Figure 2.15.: Full model of factors that influence trust in automation. The dotted arrows represent factors that can change within the course of a single interaction (Schaefer et al., 2016; Hancock et al., 2011).

controlled by the user's experience either with the environment, i.e. task type and context but also a user's self-confidence or mood, or specific features of the autonomous system. Taking into account the dynamic nature of trust on a more short-term level, the authors distinguish between initially learned trust depending on preexisting knowledge and dynamically learned trust that is possible to change during an interaction being subject to system performance and design. CAs can be described as artificial agents cooperatively guiding humans using natural language dialogue. As CAs can be seen as some kind of autonomous system, many concepts of trust in automation are supposed to be easily transferable. However, due to the ability to conduct natural dialogue and interact on a more complex information level, some idiosyncrasies need to be taken into consideration. For example, a dialogue can have a social, e.g., small talk (Schneider, 1988), or utilitarian purpose, e.g. solving a cooperative task. Although there exists work studying the effects of socio-emotional dialogue, small talk, empathetic reactions, and voice characteristics of a CA on the user's perceived trust (see Rheu et al. (2021) for an overview), most users rather see CAs still as “tools” and use the term trust concerning a system's performance or privacy (Clark et al., 2019). Therefore, trust in CAs resembles more the concept of computer believability or credibility as presented by Tseng and Fogg (1999). Here, a system's trustworthiness and expertise form the bases for its credibility, whereby under the term trustworthiness the quality of information (unbiased, truthful, honest) is understood. However, besides the content an agent provides, also its behaviour greatly influences its relationship with the user. Trust in the system's behaviour in utilitarian terms is mostly

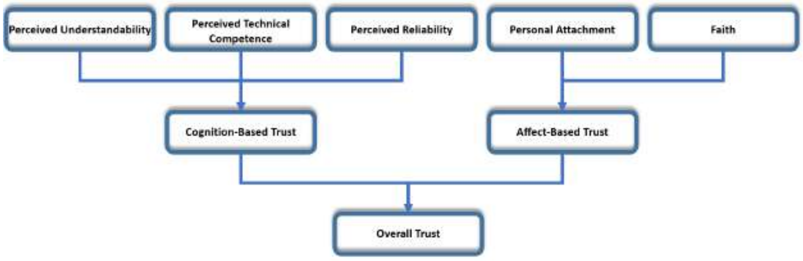


Figure 2.16.: According to Madsen and Gregor (2000), HCT is based on the two foundations cognitive- and affect-based trust. Each base of trust comprises several sub-concepts.

related to its performance with regard to consistency and reliability (Lee and See, 2004). In this context, much research focuses on a system's capability to conduct explanation dialogues for mitigating the effects of system failures by providing transparency which in turn increases trust (e.g see Nothdurft et al. (2012) or Glass et al. (2008)).

Measurement

Measuring trust in CAs is complicated because trust is multi-faceted and also a latent variable that cannot be observed directly. For this reason, several approaches for assessing the HCT relationship have been proposed. The most used methods are subjective measurements in the form of self-reported questionnaires (Madsen and Gregor, 2000; Gulati et al., 2019; Malle and Ullman, 2021). Other possibilities include (psycho-)physiological signal-based methods, e.g. EEG (Ajenaghughrure et al., 2019) or eye movements (Riegelsberger et al., 2003), as well as combinations between subjective questionnaires and objective body signals (Khalid et al., 2016). Furthermore, trust can be estimated observing the user's past experience with a system, e.g., with regard to the system's performance (Guo and Yang, 2020). Other work describes that trust can be assessed by observing user behaviour. For example, it can be measured via people's choices in an economic game like the classic prisoner's dilemma (Torre et al., 2018). Thus, different types of user behaviour would implicitly indicate higher or lower levels of trust. Within the scope of conversational systems, it has been shown that behaviour such as self-disclosure (e.g. see Laban et al. (2021)), or reciprocity (e.g. see Zonca et al. (2021)) could implicitly indicate higher levels of trust. Further, corpus analysis of linguistic features (Scissors et al., 2009) or non-verbal cues (Lee et al., 2013) in human-human dialogues can be used as a foundation for predicting trust in conversational agents.

In the scope of this work, we primarily relied on using validated psychological questionnaires for measuring subjective trust in proactive dialogue behaviour. For this, the Trust in Automated Systems scale (Jian et al., 2000) and some variants (Kraus, 2020) were used, where subjects could agree or disagree with statements about the system's impression. Sub-components of trust were measured using the HCT-model by Madsen and Gregor (2000). The model is visualised in Fig. 2.16. This hierarchical model relates to five fundamental components of trust: personal attachment and faith form the bases for affect-based trust while perceived understandability, perceived technical competence, and perceived reliability are the bases for cognition-based trust. Affect-based trust refers to a long-term human-computer relationship, being established through frequent interactions with a system. In contrast, cognition-based trust refers to a more short-termed trust. For the latter, mostly the functionality and usability of a system are of importance. Each trust component is measured by user ratings of five statement items, whereas the agreement is represented on a Likert scale.

2.3.3. Personality

Personality is a psychological construct of habitual behaviour and cognitive and emotional patterns that evolve from biological and environmental factors (DeYoung et al., 2009). Generally, psychologists agree that personality can be defined as traits or differentiable characteristics than can only be inferred from behaviour and experience (McCrae and Costa Jr, 2008). According to a theory by McCrae and Costa Jr (2008), all adults can be characterised by a series of personality traits that influence behavioural patterns as well as thoughts and feelings. These traits are developed during childhood and reach maturity in adulthood, thereafter they remain stable. Furthermore, individuals make personality-characteristic adaptations in reaction to external influences. These adaptions form a person's attitudes, behaviours, skills, and relationships. There exists scientific evidence that personality correlates with job success, attractiveness, marital satisfaction, and happiness (e.g. see a review by Alves et al. (2020)). Regarding the influence on the perception of user interfaces, Alves et al. (2020) state that personality affects the way the user perceives the interface design, the way they interact, and their acceptance of a system. Studies of the effects of personality on the interaction with conversational systems have shown the positive effect of matching personalities on the user experience (Mairesse and Walker, 2010; Nass and Lee, 2001), trust and system likeability (Braun et al., 2019). However, recent works suggest that the dimensions of an agent's personality differ significantly from human personality traits (Poushneh, 2021; Völkel et al., 2020).

Models

There exist various personality models using a different amount and nomenclature of specific traits. However, three fundamental models have been developed during the past century. Eysenck (1963, 1966) proposed a model that categorises three personality traits: extraversion, neuroticism, and psychoticism. Extraversion is described by Eysenck, as a combination of sociability and impulsiveness. Neuroticism is seen as the inability of a person to remain emotionally stable, i.e. neurotic people tend to have poor emotional adjustments, mood swings, issues of trust, etc. Psychoticism implies a generally low impulse control and refers to traits such as a person's lack of empathy, cruelty, or lonerism, while its counterpart socialization refers to altruistic, empathetic, and cooperative traits. The theory behind Eysenck's personality types is based on genetically-based personality differences stemming from biological processes. However, also conditioning and socialisation

Personality	*Level*	
	High	*Low*
Openness	Imaginative, Creative, Original, Prefer variety, Curious, Liberal	Down-to-earth, Uncreative, Conventional, Prefer routine, Uncurious, Conservative
Conscientiousness	Conscientious, Hardworking, Well-organized, Punctual, Ambitious, Persevering	Negligent, Lazy, Disorganized, Late, Aimless, Quitting
Extraversion	Affectionate, Joiner, Talkative, Active, Fun-loving, Passionate	Reserved, Loner, Quiet, Passive, Sober, Unfeeling
Agreeableness	Softhearted, Trusting, Generous, Acquiescent, Lenient, Good-natured	Ruthless, Suspicious, Stingy, Antagonistic, Critical, Irritable
Neuroticism	Worrying, Temperamental, Self-pitying, Self-conscious, Emotional, Vulnerable	Calm, Even-tempered, Self-satisfied, Comfortable, Unemotional, Hardy

Table 2.1.: The big five traits and their respective characteristics adopted from John et al. (1999).

play a role in personality development. Another early model was Cattell's 16 factors theory (Cattell et al., 1970). Cattell argued that three factors were insufficient for describing the personality of a person and more dimensions are necessary. For identifying the number of traits, he made use of factor analysis, a statistical method for identifying clusters of intercorrelated individual elements. Using this method, Cattell found sixteen source traits, i.e. underlying traits, as opposed to surface traits that are observable characteristics.

The model, however, which has become the most accepted and validated up to this date is the big five-factor model (Tupes and Christal, 1961; Norman, 1963; McCrae and John, 1992). The model is also known under the acronym OCEAN following the five personality factors it is named after: Openness to experience, Conscientiousness, Extraversion, Agreeableness, Neuroticism. In comparison with Eysenck's model, three factors have been included in the OCEAN model. Openness relates to a person's level of intellectual curiosity, creativity, and preference for novelty and variety. Conscientiousness is associated with a person's tendency towards self-discipline, dutifulness, and goal-directed behaviour. Agreeableness describes a person's tendency towards compassion and cooperative behaviour, as well as trust. A list of personality traits describing each factor is provided in Table. 2.1. The OCEAN model has been developed by independent researchers using empirical methods. It shows support for cross-cultural application and applies to people of any age. Furthermore, the factors are conceptualised as a spectrum rather than extreme categories.

Measurement

In the domain of conversational systems, research has found methods for detecting a user's personality based on linguistic features (Mairesse and Walker, 2006), conversational/interaction patterns (Stachl et al., 2020), and multimodal features including facial and speech recognition (Batrinca et al., 2012; Fung et al., 2016). However, due to the technical challenges of this method, questionnaires are still the primary assessment tool of personality factors. For each of the models described in the previous section, there exists an accompanying questionnaire. The Eysenck Personality inventory comprises 57 questions that can be answered either using yes or no responses (Eysenck and Eysenck, 1975). The questionnaire developed by Cattell et al. (1970) consists of 164 statements about a person, that can be rated on a 5-point Likert to express a person's agreement. Costa and McCrae (1989) similarly designed a questionnaire containing 240 statements. Here, the rating is also conducted on a 5-point Likert scale. In dialogue research, however, the large number of questions of these surveys are deemed rather impracticable for usage in study design due to time constraints. For this reason, a shortened version of the questionnaire for rating the Big Five personality factors developed by Rammstedt et al. (2013) was used in the scope of this thesis. The big-five-inventory (BFI)-10 consists of ten statements to be rated on a 5-point Likert Scale. For each personality factor, the questionnaire contains two statements.

2.3.4. Cognitive Load Theory and Cognitive-Affective States

This section introduces two concepts, the *cognitive load theory* and *cognitive-affective states*, that are primarily relevant in problem-solving or learning task scenarios. Consequently, they also need to be considered in the context of CAs which are specifically designed for such scenarios. The concept of cognitive load theory (Chandler and Sweller, 1991; Sweller, 1988) describes the amount of effort that humans exert during cognitive processes, e.g. reasoning and thinking. As the human's working memory is limited in capacity, both for processing information and memorising information, also the amount of mental processes a human can handle is restricted (Paas et al., 2010). For problem-solving or learning tasks, this implies that material has to be designed in such a way that the usage of the working memory is optimised. Otherwise, the task would congest the cognitive capacity and lead to a sub-optimal outcome. Cognitive-affective states combine the human's cognitive and also affective processes, such as emotions, moods, and feelings (Baker et al., 2010). The idea of combining these two aspects stems from the idea that emotions are systematically affected by the knowledge and goals of the user, and vice versa (Mandler, 1984). For example, cognitive processes, e.g. causal reasoning, deliberation, goal appraisal, and planning processes, influence the experiences of emotion (Baker et al., 2010). On the other hand, Gasper and Clore (2000) showed that emotions have an effect on verbal expressions, non-verbal expressions (e.g. facial expressions and body postures), and cognitive processes (e.g. decision making, and information retrieval). Due to these complex relationships between affect and cognition, several researchers (e.g. see D'Mello and Graesser (2011); Baker et al. (2010)) identified some user states during learning and problem-solving as a blend of affect and cognition. This resulted in the connotation of a cognitive-affective user state. Keeping track of cognitive-affective user states in HCI may lead to a better and new understanding of the user's needs (Dimoka et al., 2012).

Models

Cognitive load theory is closely related to the working processes and architecture of the human memory, particularly the short-term memory. As previously described, the short-term memory is limited and used as working memory for processing information (Card, 1981; Baddeley, 1994). Therefore, models describing cognitive load characterise different sources that influence the capacities of the working memory. Generally, it can be distinguished between three independent sources of memory load, namely intrinsic cognitive load (ICL), extraneous cognitive load (ECL), and germane cognitive load (GCL) (Sweller, 2010):

ICL represents the inherent load induced by the content itself. Hence, it can not be changed by the learning material and is caused by the complexity/difficulty of the task.

ECL arouses from the instructional design of the learning material. This kind of load is linked to mental processes that are not relevant to the task itself, like searching for or narrowing information.

GCL is directly linked to the learning process itself. A high germane load indicates that learners are engaged with the task and focus their mental resources on learning processes, e.g., the construction of learning schemes (Mayer and Moreno, 2002).

A model for cognitive-affective user states has been developed by Baker et al. (2010). The authors take into account that certain affects occur predominantly during cognitive activities and learning that are relevant for assistance systems (Baker et al., 2010): boredom, frustration, confusion, engagement/flow, delight, and surprise. Engagement/flow is the state of engagement with a task correlated with focused attention, intense concentration, and complete involvement. Delight and surprise can also be considered positive user states but mostly occur after a task has been completed or an important insight has been unveiled. These states are short-lived, while engaged concentration is more persistent (Baker et al., 2010; D'Mello and Graesser, 2011). Although being considered a negative affective state, confusion is positively correlated with learning gain, because it induces self-reasoning in the user. Related to failure, dissatisfaction, making mistakes, and giving up, frustration and boredom should be avoided or at least users should not get stuck in these states long-term. All cognitive-affective states are situated within the well-known valence-arousal model by (Russell, 2003) (see Fig. 2.17).

This valence-arousal model allows decomposing of affective states into specific dimensions: valence and arousal. Affective valence ranges from negative to positive or displeasure to pleasure, while affective arousal ranges from low-energy to high-energy or deactivation to activation. These observations make clear that depending on which

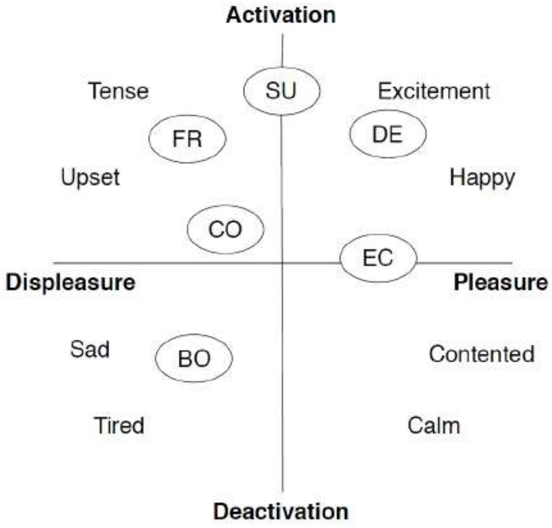


Figure 2.17.: Two-dimensional valence-arousal model showing the position of each cognitive-affective state according to Baker et al. (2010): BO - boredom; CO - confusion; DE - delight; EC - engaged concentration/ flow; FR - frustration; SU - surprise.

cognitive-affective state the user is in, different kinds of assistance or no assistance at all are necessary. As emotions influence cognitive processing, negative affects need to be kept low, especially in problem-solving and decision-making (Hudlicka, 2003).

Measurement

For measuring cognitive load it can be differentiated between three different approaches: subjective rating scales, dual-task measurements, and psycho-physiological measurements. Paas (1992) developed one of the first questionnaires for subjectively measuring cognitive load. However, the questionnaire only measures overall cognitive load and does not distinguish between the individual types. Therefore, extended questionnaires including the different types of cognitive load have been developed recently. An overview can be found in Zheng (2018). In this thesis, a questionnaire developed by Klepsch et al. (2017) is used. It measures all three types - ICL, ECL, and GCL - separately. The questionnaire consists of 12 items, with 4 items per type.

Dual-task measurements provide a direct measure. Here, two tasks are executed simultaneously by a user. Under the assumption that the user's cognitive resources can be evenly spread between two tasks that require an equal amount of information, the secondary task performance is a plausible proxy measure for the cognitive load induced by the first task (Zheng, 2018).

Finally, psycho-physiological measurements make use of sensory information (e.g. heart rate, skin temperature, EEG, pupilometry) for estimating the user's cognitive load using regression or classification analysis (Zheng, 2018). Using sensory information allows to measure cognitive processes in a temporal context. This makes psycho-physiological measurements also a promising approach for measuring cognitive-affective states. Here, several modalities have been researched ranging from facial expression, heart rate, and speech signals, to multimodal features (Picard, 2000; Corneanu et al., 2016; Schuller et al., 2009, 2011). For detecting affect in a non-intrusive manner, capturing facial expressions is the most used approach. In the scope of this, the Affectiva software was used for cognitive-affective user state detection (McDuff et al., 2013). Affectiva analyses spontaneous facial expressions with facial emotion recognition algorithms, trained using their repository based on a large database of faces from a variety of different countries and morphological groups.

2.4. Summary

This chapter dealt with providing fundamental background information for the reader to understand the work presented in this thesis.

Knowledge of DS architectures and models is required to understand interaction procedures between humans and machines. Further, this serves as a foundation for understanding the prototypes developed in this thesis. Concepts of dialogue strategies are important for putting in context the proactive dialogue strategies that combine elements of grounding and initiative. Finally, we gave an overview of evaluation methods that are relevant for understanding the different kinds of user studies presented in this thesis. Here, we also explained the concept the user simulation which is necessary to understand for evaluation purposes, but also in the context of automatically training data-driven proactive dialogue strategies.

As this work also presents novel work on using statistical methods for developing proactive dialogue models, we reviewed methods for collecting data for statistical dialogue modelling and introduced several ML algorithms that were applied in this thesis. Decision trees, SVM, and ANN, were primarily used for user state recognition, while RL was utilised for creating user-adaptive proactive dialogue. Here, also evaluation methodologies for comparing the applicability of the different ML approaches were presented.

Finally, we gave an overview of relevant psychological concepts and models with regard to HCI. Background on the TOM is necessary for understanding how computers may predict user behaviour and internal states from observations which can subsequently be used to adapt the dialogue considering social cues. One social cue that is of particular interest in the scope of this thesis is the psychological concept of trust that is relevant for appropriate cooperation between system and user. Therefore, we summarised trust models relevant for HCI and discuss measurement methods. Besides trust, we also considered a user's personality, cognitive load, and cognitive-affective states in relation to proactive assistance behaviour in various studies presented in this thesis. For this reason, we also reviewed relevant models and measurements of these concepts for enabling their applicability during human-machine cooperation. In this work, we study the influence of proactive dialogue on cooperation and its adaptation to specific users and their contexts. Regarding this, we introduce related work on proactivity in HMI and user-centred dialogue in the next chapter. There, we also provide an in detail explanation of the differences and added value of our approach.

3. Related Work

This work aimed at improving the cooperation between users and CAs by including user-adaptive proactive dialogue behaviour. For enhancing cooperation concerning task- and socially-related aspects, the domains of proactive HMI and user-centred DS are necessary to consider. Therefore, this chapter reviews the state-of-the-art in both areas. First, we provide definitions of proactive behaviour and elucidate the concept concerning HHI and HMI. Considering proactivity in HMI, several modelling approaches were reviewed in the sub-domains of interaction with autonomous systems, human-robot interaction (HRI), and HCI. For allowing comparison with our approach, a particular focus was set on related work and systems that considered proactivity in a conversational context. Moreover, the current state of knowledge of the user perception of proactive interaction with robots and computers is presented. Here, also conversational systems were prioritised. This information was subsequently used for modelling proactive dialogue in CAs.

Secondly, user-centred approaches for developing DSs were summarised. Here, we stressed the importance of user modelling and made distinctions between two types of use-centred dialogue modelling approaches: static and dynamic user adaption. For both types, we present various approaches on how to include user-related information for adapting the dialogue in several ways. Concerning our work, it was important to get an understanding of which type of user information may be applicable for determining the need for proactive behaviour. Further, this provided insights into the design and adaptation of proactive dialogue. At the end of each section, we discuss extensively the differences and similarities between related work and our proposed approach. Also, we provide explanations for manifesting the novelty of our approach.

3.1. Proactive Human-Machine Interaction

For understanding the term proactivity in the context of HMI, it is first necessary to consider the definitions of proactive behaviour in HHI. Here, the concept of proactive behaviour has been extensively studied in the domain of organisational psychology and management (Crant, 2000; Frese and Fay, 2001; Parker et al., 2006). For example, Crant defines proactive behaviour in organisations as "taking initiative in improving current circumstances or creating new ones; it involves challenging the status quo rather than passively adapting to present conditions" (Crant, 2000). Another definition was provided by Grant and Ashford (2008), who describe proactive behaviour as an "anticipatory action that employees take to impact themselves and/or their environments". In addition, there have been identified several characteristics that can be attributed to proactive behaviour. For example, proactivity is supposed to have a long-term focus and intends to predict future states, is action-oriented and goal-directed, while also being persistent and self-starting (Crant, 2000; Frese and Fay, 2001). In contrast, reactive behaviour is about reacting to environmental demands, only doing what one is told, and not about developing plans to deal with possible difficulties (Frese and Fay, 2001). Proactive behaviour can also either be as a personality trait and behavioural predisposition (Bateman and Crant, 1993) or as a context-induced behaviour (Morrison and Phelps, 1999). For example, Bateman and Crant (1993) constructed a 17-item scale for assessing a person's tendency towards proactive behaviour. They found correlations between proactive behaviour and the personality traits conscientiousness (goal-oriented and implying persistence toward reaching closure on an objective) and extraversion (seeking new experiences and activities). Furthermore, it was found to correlate with the need for achievement and dominance. Crant (2000) described organisational culture and norms, as well as different situational cues, e.g. socialisation of newcomers as context-related features that influence proactive behaviour. Further, Crant proposed to use a cost/benefit approach for addressing cognitive processes by which people decide when to become proactive or not. This stems from the idea that people evaluate the social costs and other risks before taking proactive actions. Experiments in this research field have shown the positive effects of proactive behaviour at work. Proactivity leads to higher job performance and team performance and is associated with leadership and innovation (Crant, 2000). Additionally, it refines one's intrinsic motivation and self-regulation (Frese and Fay, 2001) while also creating coworker trust in work environments and positively contributing to socialisation (Parker et al., 2006).

Having described proactive behaviour in HHI, we now investigate the concept in HMI. Here, it can be associated with a machine's ability to act autonomously. Automation is understood as the ability of technology to perform tasks or parts of tasks that were formerly executed by a human (Parasuraman and Riley, 1997). However, the people's demand for a machine's capability to deliberately take action depends on its intelligence and purpose. For example, industrial machines are intended to automatically take over tedious or labor-intensive tasks. Contrarily, CAs or robotic agents used in collaborative tasks will be expected to express another kind of autonomous action-taking. Here, users will expect the system to actively contribute to problem-solving, to integrate them into its decision processes, and communicate naturally for action alignment and grounding. Thus, users will apply a more human-like notion of automation or rather self-awareness that involves cognitive processes. In the scope of this thesis, we understand proactive behaviour under this notion of autonomous system behaviour. For modelling adequate proactive behaviour, we review different approaches and systems in the following section, starting with initial models in automation research.

3.1.1. Modelling Approaches of Proactive Human-Machine Interaction

For modelling human interaction with automation, Sheridan and Verplank (1978) introduced the notion of LoA. In their work, how humans and computers can cooperate was divided into ten levels, ranging from offering no assistance to completely autonomous behaviour. The individual levels are listed in Table 3.1. The intention behind this model was to aid automation designers in deciding on the mixture of human and machine decision-

Level of Autonomy	*Description*
1.	The computer offers no assistance; the human must take all decisions and actions.
2.	The computer offers a complete set of decision/action alternatives.
3.	The computer narrows the selection down to a few.
4.	The computer suggests one alternative.
5.	The computer executes that suggestion if the human operator approves
6.	The computer allows the human a restricted time to veto before automatic execution
7.	The computer executes automatically, then necessarily informs the human
8.	The computer informs the human only if asked
9.	The computer informs the human only if it, the computer, decides to.
10.	The computer acts completely autonomously

Table 3.1.: Levels of Autonomy according to Sheridan and Verplank (1978).

making for a task at hand. Further, the authors suggested that smooth communication and mutual understanding are crucial if the computer takes over control from the human and vice versa. However, they did not describe any communication strategies in this regard. Parasuraman et al. (2000) later refined this model for answering the question of which system functions should be automated and to what extent based on the system's technical capabilities. For this, they applied the LoA to different functional domains or types of system functions. These functions are based on a simplified four-stage model of human information processing, namely information acquisition (sensing and registration of input data), information analysis (cognitive functions, e.g reasoning), decision selection, and action implementation (machine execution of the choice of action). Each function can vary in its degree of automation. However, the LoA of a function does not need to be fixed during run-time but may be adaptive to situational aspects, e.g dynamically changing environmental factors, such as specific events or user intentions. This concept is known under the term adaptive autonomy (e.g. see Kaber et al. (2001); Byrne and Parasuraman (1996); Scerbo (1996)). By adapting the level of autonomy, tasks are dynamically allocated between the user and the system depending on the context (Byrne and Parasuraman, 1996). LoA have been extensively studied in unmanned objects, e.g. drones (Zhou et al., 2019), autopilots (Anderson et al., 2018), and automated driving (Flemisch et al., 2008; Walch et al., 2016; Biondi et al., 2019). The more autonomous a vehicle gets, the more responsibilities, e.g. steering and acceleration/deceleration, are taken away from the operator. This allows an operator to stay in a supervisory role over the vehicle. However, in case of events where the system is unable to make safe decisions in the operator's interest, a hand-over or possible alternatives need to be communicated actively. Therefore, in autonomous driving, for example, research focuses on predicting opportune moments when to initiate an interaction (Kim et al., 2019; Park et al., 2020) and how to communicate the information (Walch et al., 2016; Park et al., 2020). Depending on user preferences or situational characteristics, e.g., emergency or non-emergency, the style, and content of the interaction (alert, suggestion, hand-over) can be altered.

Due to their highly autonomous nature, robotic systems have become one of the major research streams for studying proactive interaction. Trying to structure the process of a robot's proactive behaviour, Peng et al. (2019) identified three elements: anticipation, initiation of action, and target of impact.

Anticipation refers to the robot's capability to sense its surroundings for predicting future environmental states or the human's intentions. This would subsequently allow a robot to decide when to take anticipatory actions, i.e. to behave proactively. For making assumptions about the situation and the human, there exist various methods. For example, Grosinger et al. (2016) developed a robot with planning capabilities that took into account context- and time-related measures to decide when to take action. The robot could make assumptions about the human's activity using a simple user model that contained the user's location and daytime. Based on the activity, the robot would proactively interact with the user. Liu et al. (2018) presented a shopping assistant robot that learns the appropriate moments for being proactive using ANNs. In their work, they captured the user's motion and speech data to make assumptions about the user's behaviour. Furthermore, they defined yield actions, i.e. representations of the moment when a user yields his turn and does nothing, for initiating proactive behaviour. This data together with a representation of the interaction history was then fed to the neural network for producing the desired robot behaviour. Anticipation of user behaviour has also been used to decide on ways how to approach humans acceptably. For example, Kato et al. (2015) identified a user's intention to interact with a robot based on analysing the user's trajectories and body postures. Additionally, a human's gaze can be interpreted to assume potential user actions (Huang and Mutlu, 2016).

Initiation of action addresses a robot's autonomy in the functional domains of decision selection and action implementation. Thus, this element is closely related to the robot's LoA during the interaction. Following the categorisation by Sheridan et al., Beer et al. (2014) described ten level of robot autonomy (LORA). LORA range from manual teleoperation to full autonomy. At the lower levels, the HRI is generally controlled by the human, however, the robot may assist to some degree with action implementation, e.g., the robot automatically steers to avoid a collision with an obstacle in case a user navigates the robot inappropriately. At the intermediate levels, both interaction partners create plans to achieve a task, however, the human only has supervisory control, whereas the robot takes all actions. At the highest levels, the robot performs all actions of the task with the user only providing a high-level abstract goal. Naturally, higher LORA also allow for a more proactive initiation of interaction, which is essential for cooperative and social robots. Even at a high autonomous level, a robot should interact with the human to some degree due to the human-out-of-the-loop phenomenon in automation that may cause performance problems (Endsley and Kiris, 1995). The proactive initiation of interaction can span various modalities and purposes. For example, a robot can verbally initiate to offer help, if a user runs into problems during executing a task (Cramer et al., 2009), or making unsolicited suggestions or remarks to the user (Peng et al., 2019; Liu et al., 2018; Rau et al., 2013). Furthermore, a robot may also proactively initiate physical actions, e.g. reach the robot's arm out to take something from the user (Pandey et al., 2013) or manipulate objects without explicitly asking (Baraglia et al., 2016) in joint task scenarios.

Target of impact denotes simply the addressee of the proactive actions. This can be either a human or group of humans in collaborative task scenarios (Wagner et al., 2021; Peng et al., 2019) or other entities in a multi-agent task scenario (Lou et al., 2012).

Observing the concept of proactivity in the HCI domain, proactive behaviour can be primarily associated with recommender systems as well as CAs. Recommendation systems are proactive per se in most cases, as they provide suggestions without explicit user request (Rook et al., 2020; Shah, 2018). Under the term recommendation system, a decision support tool is understood that autonomously provides user-adapted advice on items to ease people's navigation in large product or information spaces (Rook et al., 2020). Such systems are, for example, applied for assisting in restaurant search (Christakopoulou et al., 2016), online shopping (Linden et al., 2003), or to recommend movies Cai and Chen (2020) by observing context features and user preferences. A major issue in the development of recommender systems is how to improve the accuracy of the content provided about the user's needs (Christakopoulou et al., 2018; Ikemoto et al., 2019; Rook et al., 2020). For generating accurate recommendation models, user modelling approaches are essential. For example, content-based recommender systems (Pazzani and Billsus, 2007) model the user by characteristics of the liked or disliked items. Systems based on collaborative filtering help users in decision-making by considering the opinions of other people who share similar interests (Lu et al., 2015). Research on *when* and *how* to recommend is limited, but recently context-related factors, such as the user's activity (Dingler et al., 2018; Shah, 2018), for deciding when to act as well as dialogue-based approaches for how to provide suggestions have been studied (Cai and Chen, 2020).

As proactivity can be seen as a cooperative trait and has already been identified as a major part of conversational intelligence (Chaves and Gerosa, 2021), proactive behaviour has become a prerequisite to consider when developing personal and especially CAs. In the following, we will use the term CA also for describing personal assistants as the definitions are blurry and often interchangeable, e.g. see Sarikaya (2017). Proactive behaviour in CAs has several definitions. For example, proactive assistance in CAs can be defined as an "agent taking an action to assist the user without the user's explicit request"(Sarikaya, 2017). Similarly, Nothdurft et al. (2015b) define proactive behaviour as "an autonomous, anticipatory system-initiated behaviour, with the purpose to act in advance of a future situation, rather than only reacting to it".

Proactive behaviour in CAs is based on the principle of mixed-initiative interaction (Horvitz, 1999). In mixed-initiative interactions, a user and an autonomous assistant are collaboratively working together for solving tasks. For assisting, the agent may operate in a reactive, where it acts only upon user request, or in a proactive mode, where it

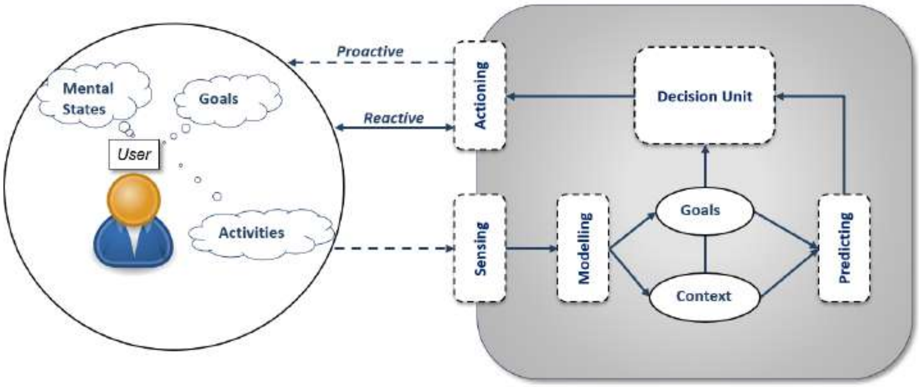


Figure 3.1.: Reference model of digital personal assistant integrating reactive and proactive behaviors based on the work of Meurisch et al. (2017).

initiates actions autonomously. In the proactive mode, the intelligent agent is required to track the user's activities and goals. For example, Horvitz (1998) describe this process as "background assistance tracking", which just observes the user during work. Further, the agent needs to reason about the costs and benefits of taking automated actions. In this context, the assistant may initiate a proactive dialogue to communicate and negotiate a system's decision process for minimizing the risk of such "speculative assistance".

For integrating proactive behaviour in CAs, Sarikaya (2017) and Meurisch et al. (2017) provided reference models/architectures that are quite similar in their construction. For illustration, we therefore only describe the model by Meurisch et al. in the following. This model is depicted in Fig. 3.1. For making inferences about their environment, proactive CAs use sensors that can track, for example, a user's activities and physical or psychological states. Using this information, the CA would be able to model the user's goals or intention and context. This further allows for goal-aware prediction of future contextual states. Based on this knowledge, the system may then make goal-based decisions, and initiate intelligent actions. The way such a CA can cooperate in a mixed-initiative interaction can be characterized by the respective proactive mode on the interface-proactivity (IP) continuum introduced by Isbell and Pierce (2005). They transferred the autonomy levels based on previous work by Sheridan and Verplank (1978) to the domain of HCI, resulting in five different levels of proactive assistance behavior. The IP continuum ranges from zero, i.e. the user acts completely autonomously, to full automation, i.e. the assistant acts completely on behalf of the user. The gradations between these two extremes are warnings that tell the user to pay attention, notifications that tell the user exactly what to pay attention to, and suggestions that give the user multiple decision options. The more proactive a system becomes, the more it takes control and responsibility away from the user. This also increases the risk of failure, since there is a possibility that the system will perform actions that are inconsistent with the user's goal due to not asking for confirmation.

This may potentially threaten a healthy human-computer relationship (Isbell and Pierce, 2005). Therefore, most current applications in this area are concerned with notification or suggestion management, since the cost-benefit ratio is more controllable. For example, there are a number of works on desktop (Iqbal and Bailey, 2008), smartphone notifications (Lopez-Tovar et al., 2015), and proactive suggestions for interactive television (Ferraz de Abreu et al., 2019). However, there also exist some system prototypes, that span multiple points of the IP-continuum and can perform tasks fully autonomously. For comparison with our work, we present the most important of these assistants in the following.

The Cognitive Assistant that Learns and Organizes called Calo (Yorke-Smith et al., 2012) was a proactive assistant that helped users with task management in an office environment. For example, it could assist with organising meetings, reminding of important activities, delegating work to colleagues, and collaboratively working on specific processes, e.g. hiring a new employee. Therefore, it relied on the BDI-framework (see Section 2.1.2) for modelling the user's goals and knowledge. Furthermore, it applied a workflow tracker to detect the user's current activities. This allowed the system to collect user-specific, e.g. preferences, and context-related information, e.g. electronic to-do lists, schedules, and current activities. Based on this information, the system could reason about the cost-benefit of proactive behavior and was able to adjust its level of proactivity accordingly. For example, if the workload was detected to be high, the system could suggest transferring work to others, automatically preparing background material for the meeting, or offering a reminder. The cost-benefit value was calculated using system-related metrics such as the urgency of proactive behaviour, the cost of a mistake or interruption, and the confidence of the current workflow state. The decision on which autonomy level to choose was based on heuristics relying on user-stated advice and fixed rules. Thus, proactive behaviour was only modelled for the specific task and the rules were required to be predefined. Further, the agent only interacted using textual messages and no speech.

The Reflective Agent with Distributed Adaptive Reasoning called Radar was also a personal assistant that could help office employees to solve their tasks more efficiently (Faulring et al., 2010; Garlan and Schmerl, 2007). Among other tasks, it could help to reduce email overload, i.e. the difficulty of people to handle a large number of emails, by providing adequate strategies such as filtering emails according to various criteria. For this, it comprised several modular components. So-called task specialists were intended to assist users in executing particular tasks, e.g. schedule management, email-handling. The specialists were adaptive knowledge bases. These bases contained static information on how to conduct specific tasks and learned knowledge about user-preferred methods in task execution. A task manager coordinated the specialists while tracking current activities and prioritising individual tasks. For deciding when and how a user should be interrupted, i.e. whether a meeting was scheduled automatically or an interaction was initiated, a dialogue manager was used. This module contained knowledge about a specific user's attention and interruption policies. Here, proactivity was also designed only for a rigid domain, and prior user knowledge was required for making adequate decisions. Besides, only textual interaction was considered.

A new version of Radar was published more recently (Grover et al., 2020) that used state-of-the-art planning techniques instead of such case-based reasoning for proving proactive decision support. The new system could, for example, help a fire chief in the process of developing a sequential plan to control fire in a building. Here, the system proactively assisted with resource management, e.g. the water tanks to be sent, the number of ambulances to be called, and the areas to be secured. For this, the system automatically generated the respective planning problem in the background, analysed the possible solutions, and highlighted the resources required for solving the planning problem. Proactive assistance was provided by incorporating knowledge about the user's capabilities and time constraints. This was then used to increase the situational awareness of users by suggesting or notifying them that alternatives would be available. In their work, however, proactive behaviour was only represented as a list of abstract plan steps, and no natural language was used. Also, prior user knowledge is required for adequate proactive behaviour.

Another proactive agent called Virma (Virtual Insurance Risk Management Advisor) (L'Abbate et al., 2005) was a German virtual personal insurance and finance assistant with a focus on risk management counseling for small enterprises. For assistance, it relied on existing expert knowledge guiding users during the elicitation of the required data to create a user-specific risk portfolio. During unclear or problematic situations, users could take the initiative by asking questions or expressing uncertainty while performing tasks. However, also proactive interventions were possible. For this, the system evaluated a set of constraints at each interaction step for deciding about the necessity of proactive action. The constraints were based on static user profiles, e.g. business sector, expertise. For example, one task of the finance assistant was to elicit information about the user's business activities. For this, users could select activities from a pre-defined list. Depending on the user profile the system could act differently. In case the business sector was known and the user had low expertise, the system would trigger a sub-dialogue by presenting a refined set of information based on previous sessions by users with the same profession. In doing so, the system would proactively pre-select adequate information instead of showing the full list of options. The decision, of when to act proactively was pre-defined as a set of artificial mark-up language (AIML) rules. Similar to recommender systems, proactive behaviour was here solely used for reducing the information space and not in a conversational manner.

Besides how to act proactively, also the decision, of when proactive behaviour should be initiated, has been identified as a crucial topic in HCI (Nothdurft et al., 2015b). Determining the timing of proactive system behavior is complicated and mostly related to research on managing task interruptions in AI (McFarlane and Latorella, 2002). To minimise interruptions, it is imperative to gather knowledge about the user's context (Iqbal and Bailey, 2008; Cha et al., 2020; McFarlane and Latorella, 2002). For example, Iqbal and Bailey (2008) used interruption points in task execution to develop a notification delivery strategy. Similar to the research on recommender systems, personal contextual features have also been explored for initiating an interaction with a smart speaker (Cha et al., 2020). In addition, device- or user-related features can be used to trigger proactive actions: Lopez-Tovar et al. (2015) determined the timing of smartphone notifications as a function of device location and state, while Segal et al. (2018) predicted time steps at which the user could disengage from a crowdsourcing task to trigger motivational messages. Additionally, the need for assistance could be assumed based on the prediction of the user's affective state. In this context, D'Mello et al. (2006) inferred affective states such as boredom or confusion from interaction patterns in a mixed-initiative dialogue.

So far, we only considered goal-directed proactive systems. However, there also exists proactive behaviour that is not necessarily goal-oriented. For example, Wu et al. (2019) described proactivity as the act of leading the dialogue and actively changing the discussion topic, while keeping the dialogue natural, coherent and engaging. Similarly, Yoshino and Kawahara (2015) understood the act of actively presenting or recommending topics related to the current interaction as proactive behaviour. Here, proactivity was especially used for resolving ambiguous user demands, by showing possible candidates for the query instead of doing nothing. However, non-goal-directed behaviour was not considered in this thesis.

3.1.2. User Perception of Proactive Human-Machine Interaction

Based on preceding considerations of proactive behaviour in HRI and HCI, we will now present related work regarding the user perceptions of proactive HMI.

In Baraglia et al. (2016), proactive non-verbal behaviour of a robotic assistance system was evaluated in an object placement joint task execution scenario. For this, they conducted a study with 18 participants comparing three different robot versions: a reactive robot that only executed tasks when users requested help; a robot initiated semi-proactive help when it discovered that help was required; and a proactive version that executed tasks independent of the user. Their results showed that there was no difference concerning total task duration. However, participants subjectively rated the reactive robot to be lazy, slow, and hesitant when compared to the more proactive versions. Further, the proactive versions were subjectively more liked by study participants.

Rau et al. (2013) proposed a design of social robot interaction for assisting in decision-making tasks. Here, two levels of proactive interaction (high vs. low) were compared. The authors presented a WoZ-study, in which participants had to complete a sea survival task in collaboration with a remotely controlled robot. The HRI was modeled in such a way that the robot provided its opinions on how to solve the task in natural spoken language. The robot's opinion was either provided proactively or reactively. In the study, the influence of the robot's proactivity on decisions, trust, robot credibility, and user workload was evaluated. The results demonstrated that trust in the robot was higher in the low-level (reactive) than in the high-level condition. However, cognitive load showed no significant difference between the LoA.

In another study on social interaction, Peng et al. (2019) studied the design and evaluation of an autonomous robot in a decision-making-support scenario. Proactive interaction with the robot was designed in three dimensions, i.e low, medium, and high. Here, the lowest level of proactive behavior could be considered reactive behavior. For evaluation of their approach, a within-subject study using the WoZ paradigm was conducted.

The user's task in the study was to select gifts for fictitious personas while being assisted by a robotic shopping assistant. A human wizard took the role of the assistant and tried to infer the participants' need for assistance and triggered the proactive interactions. The results showed, that a medium-level of proactive interaction was perceived as the most helpful. Furthermore, a high degree of autonomy was disliked and perceived as less appropriate compared to the other two conditions.

Schmidt et al. (2020) investigated user acceptance of proactive voice assistants in cars. For this, they conducted a driving simulator study in which a system could express proactive behaviour based on particular driving events. For example, the proactive voice assistant could suggest to the user the nearest gas station in case fuel was low, or propose rerouting in case of traffic jams. Comparing such a proactive in-car voice assistant to a reactive version of the system, the authors found that proactive behaviour was generally well-received in an experiment with 42 subjects. However, most participants wished for the possibility to deactivate proactive behaviour in certain situations. In addition, they found no differences between the conditions concerning the user's cognitive load.

Considering the original Radar system, Steinfeld et al. (2007a,b) conducted a study for measuring the effects of the proactive system concerning a less proactive version. In a study with 66 participants, they found that the proactive version was able to perform the planning task better than the more reactive version and was also found to be more useful. Grover et al. (2020) evaluated a more recent version of Radar which assisted study participants to interactively plan their studies, e.g. which study courses to take dependent on various factors. The authors compared several versions of the system: a system that could only provide plan suggestions, one that only could validate the effects of the user's plan choices, a system capable of both features, and a baseline variant. The results of a study with 56 participants showed that the variants providing plan suggestions were more efficient in task execution and also increased user satisfaction. Further, providing plan validations and suggestions proved to be more effective in teaching users how to construct study plans. Between domain experts and novice users, no differences were found.

Glass et al. (2008) conducted interviews with Calo users for evaluating the effect of the system's behaviour on the user's perceived trust. Although the authors primarily focused on the effect of explanations and transparency behaviour of trust, they also briefly considered the effect of the system's autonomy. In interviews with four subjects, they found that people tended to trust Calo only if they were also able to verify its behaviour. This included the ability to override erroneous behaviour. The authors identified that users seemed to trust the system more after time, where users could observe the performance of the system. Generally, lower autonomy was trusted more and the relationship between trust and autonomy depended on the magnitude of the system's actions in the real world.

Iqbal and Bailey (2008) conducted an experiment for examining the effects of timing (scheduled at identified breakpoints vs. immediate) and content (task-related vs. general) of proactive desktop notifications. During the experiment, 16 study subjects had to design a floor plan for a model workspace in a computer science building. A notification system would provide information according to the described notification timing and content conditions. For the timing of actions, they further differentiated between fine, medium, and coarse scheduled notifications. These levels represented the hierarchy of sub-tasks of an overall problem or activity, e.g. fine described that the notification was triggered after a small task such as manipulating the design of a small element of the workspace, while coarse indicated that the user switched to a completely different activity. The results showed that task-relevant notifications should be scheduled after fine or medium activities, while general information may be provided at a coarse level. Further, the experiment showed that users experienced lower frustration and reacted faster when notifications were scheduled instead of being provided immediately. However, urgent messages may be delivered immediately or at fine breakpoints. Generally, initiating interruptions at an inappropriate point of time or in the wrong way could be perceived as disruptive and obtrusive. Particularly, this could negatively influence the user's perceived trust in the system (Jenkins et al., 2016; McFarlane and Latorella, 2002).

Meurisch et al. (2020) explored user expectations of proactive AI systems based on interviews with 272 people. In their work, the proactive level of an intelligent assistant was differentiated in four levels: *reactive*, *proactive support I* (notifications and alerts), *proactive support II* (personalised recommendations), and *autonomous support* (making decisions and taking actual actions). The results of their interviews showed that users are generally in favour of proactive support across several task domains. Particularly, users are willing to receive proactive support for social interactions, e.g. arranging appointments with a personal assistant. Further, they are open to proactive behaviour in physical health support systems, as well as smart home and task management. However, users did not wish to be proactively supported in the mental health domain. Investigating the general acceptance of the different proactive levels, the authors discovered that proactive support II was the most favoured followed by reactive and proactive support I. Completely autonomous support was the least preferred. In addition, Meurisch et al. studied the relations between the five personality traits, some selected socio-demographic traits (age, gender, ...), and the proactive levels. Here, the results revealed that the age of the user relates to the acceptance of proactive behaviour. They found that elderly people tend to prefer reactive systems as compared to younger study participants. Especially, elderly with low degrees of the personality traits openness to experience and high levels of conscientiousness are more likely to opt for a reactive system. People with higher levels of openness and lower levels of extraversion seem to accept a system with proactive support level II. However, no relations between such user-specific features and proactive support I were found. Generally speaking, they found that the user perception of proactive behaviour is user-dependent and also domain-dependent, and might be also related to a user's tendency to fear a loss of control (Meurisch et al., 2017; Sankaran and Markopoulos, 2021) which could also result in mistrust and frustration.

For recommender systems, most work on the user perceptions of these systems addresses the accuracy of the suggestions dependent on the user needs and expectations. It has been found that an accurate system contributes positively to its perceived trustworthiness (Rook et al., 2020) and the user's satisfaction with suggestions (Cai and Chen, 2020). Yoshino and Kawahara (2015) showed that proactive recommendations of new topics encourage interaction with the system.

3.1.3. Conclusion and Research Gaps

Generally, the concept of proactivity in HMI is closely related to computer autonomy and may be divided into several levels called LoA that indicate different ways a computer can autonomously offer assistance. These levels have been transferred to HRI and user interface design, but have not yet been considered for conversational and dialogue design. To close this research gap, this work deals with transferring the concept of LoA into the dialogue domain by specifying novel *proactive dialogue act types.*

Further, the LoA may be applied to different types of system functions. Regarding DS, for example, the ASR module would be considered to act according to the highest LoA as it acts completely autonomously. In this thesis, we consider the LoA in the context of DM for defining the degree of decision-making and action implementation during dialogue. As previously noted, we define proactivity in CAs as their ability to actively contribute to problem-solving, integrate users in their decision processes, and communicate naturally for action alignment and grounding. In this regard, our definition follows the principles of adequate mixed-initiative interaction design and applies them in a conversational context.

Related work also revealed that proactivity in HMI may be described in the form of a structured process including anticipation of the need for proactive behaviour, initiation of action, and target of impact. This process was implemented in some form by all exemplary proactive systems that were presented as related work. Further, the reference model of proactive digital personal assistants by Meurisch et al. (2017) also inherited such a process structure. In this thesis, we adopt this process model for application in CAs and conceptualise a novel cognitive architecture for enabling proactive behaviour in conversational systems. For this, we define cooperation in decision-making and problem-solving as a dialogue problem, where the problem of whether to become proactive and to what extent needs to be decided on a turn-level basis. Contrary to related work, we include dialogue information for anticipating the need for proactive behaviour and realise the initiation of action in the form of dialogue acts. In our work, the target of impact is a singular user.

Furthermore, the decision whether to become proactive was mostly made dependent on specific task information and user states, including a user's expertise level, workflow, or user preferences. However, this information was usually pre-defined in advance resulting in quite rigid proactive strategies. For this reason, we consider a more dynamic approach in this thesis by integrating real-time measurements of user states, such as cognitive-affective user states, user uncertainty, as well as context information, e.g. user activity, in the dialogue model for determining the need of initiating proactive conversation.

Another problem of adequate proactive system behaviour concerns the timing of action execution. Even though timing showed to influence the user's perception of proactive behaviour, this problem is mostly related to adequate turn-taking behaviour in dialogue which exceeds the scope of this work. Within this thesis, we focus on the other central aspects – whether to become proactive and to which extent. These aspects are deemed to be the primary driving factors for improving human-machine cooperation using proactive dialogue. For mitigating the effects of wrongful user interruption, we utilise well-defined timing of proactive actions at the dialogue turn level in this thesis.

The evaluation of proactive system behaviour in related work is heavily one-sided on task-related metrics, such as task efficiency or task success. A user-centered view on proactive system behaviour is widely underrepresented or even non-existent. For this reason, this work adds value by integrating the user for proactive dialogue design. In doing so, we provide a more complete view of the effects of proactive dialogue on human-machine cooperation. Therefore, we also include subjective user metrics for evaluation, e.g. impact on user satisfaction, cognitive load, or trust, for comparing the effects of the designed proactive dialogue strategies on the cooperation.

As trust is an important factor in cooperation and thus essential to consider for the design of CAs, we focus our investigations on the impact of proactive dialogue on the social construct of trust. Oddly, trust in CAs is still an often overlooked topic. Therefore, this work adds value to the understanding of trust in CAs with a focus on proactive dialogue based on the concepts and findings of trust in autonomy.

For deciding on appropriate proactive system behaviour, related work usually implemented some kind of a cost-benefit function. However, the used functions are mostly restricted to task-related or explicit user information, e.g. preferences, and user behaviour, for calculating the costs and benefits. Implicit user information, such as the user's mental state or personality, was widely omitted. We deem this information, and particular perceived user trust of system behaviour, as fundamental for improving cooperation with CAs, Therefore, we include trust in the cost-benefit function for enabling adequate proactive dialogue. This was also ought to enable user-adaptive proactive dialogue.

In related work, often only one proactivity type was defined and used for the complete interaction or scenario-specific proactive behaviour was pre-defined in advance. Thus, proactive behaviour was designed in a rigid way and for quite limited contexts. For improving cooperation, however, a more dynamic and user-centred approach is necessary. Therefore, we aim to render proactive dialogue user-adaptive to provide more flexible strategies for adequate assistance. In the following, we hence review related work regarding techniques and methods for user-centred DSs.

3.2. User-centred Dialogue Systems

Ideally, proactive DSs are developed using a user-centred design approach. In this regard, the ISO 9241-210 Part 210 (ISO, 2019) considering the human-centred design for interactive systems promotes "system design and development that aims to make interactive systems more usable by focusing on the use of the system; applying human factors, ergonomics and usability knowledge, and techniques". The goal is to integrate the user into the design of the system for tailoring the interaction concerning the needs, goals, and preferences of the user. In doing so, a more effective, efficient, and satisfying user experience is envisioned. For achieving this, deep knowledge about users their context, and the task at hand is required. This information needs then to be leveraged by intelligent assistants to adapt the dialogue to the user. The term user-adaptive is closely related and often interchangeable with the term personalisation. According to Fan and Poole (2006), personalisation in digital technologies can be described as "a process that changes the functionality, interface, information access, and content, or distinctiveness of a system to increase its personal relevance to an individual or a category of individuals". For enabling personalisation a user model is inevitable.

A user model contains an individual's or group of individual's characteristics, preferences, interests, and needs and other relevant information for providing adaptation (Kocaballi et al., 2019). There exist two types of how to create a user model. Using implicit personalisation, user information is automatically gathered by analysing previous interactions with the system. Explicit user modelling requires the system to involve the user directly in the process, e.g. by asking questions at the beginning of the interaction. Furthermore, personalisation can be characterised considering for whom the interaction is personalised: a group of people or specific individuals. Similarly, what is personalised can be characterised: the interaction content, information presentation, modality, and functionality of the system itself (Kocaballi et al., 2019). As there exists a wide variety of personalisation possibilities, we restrict to the most relevant aspects of user adaptation concerning the topic of this thesis. Therefore, a more detailed overview of related work in the area of static, i.e. user-dependent, and dynamic, i.e. situation-dependent, adaptive dialogue approaches is presented. Here, static refers to an adaptation approach that remains static throughout the dialogue and only differs on the type of user, while dynamic refers to adaptive approaches that include events or more short-term information, e.g. affective user state, for tailoring the dialogue during ongoing conversation.

3.2.1. Static User Adaptation

Related work on adapting to user-dependent features has primarily relied on user models representing a user's expertise or knowledge, personality, or specific static characteristics and preferences. For adapting to the user's knowledge, the expertise is modelled dependent on the current task domain and categorized into distinct ordinal values for representing the knowledge level. For example, Jokinen and Kanto (2004) modelled the user experience as a scalar value on a 3-point Likert scale. Here, the individual levels were, novice, competent, and expert. The authors integrated the user model in a speech-based e-mail system for providing the dialogue manager with the ability to vary the content and style of the system utterances dependent on the assumed expertise level of the user. When a novice user interacts with the system, the system's responses would be enriched with extra information. Contrary, an expert user would receive concise and pragmatic answers.

Komatani et al. (2005) proposed a more fine-grained user model by making a distinction between the user's skill level of DS usage and their knowledge level about the task domain. Furthermore, they introduced a further variable representing the degree of urgency, as the task domain was to provide bus schedule information. The classification of each of the expertise levels was conducted using decision tree learning. Here, it was differentiated between either high or low skill and knowledge levels. Dependent on these levels, the dialogue content would subsequently be adapted. For example, during interaction with a user having either a low skill or knowledge level, the system's response would be augmented with additional explanations. Additionally, the dialogue initiative was altered depending on the user's skill level. The system interacted with novice users in a system-initiated manner, while it interacted with competent users in a more open-ended manner. Evaluating this user-adaptive behaviour, the authors found that a user's skill level could be increased more rapidly using the proposed user models. Further, the dialogue duration could be decreased for more experienced users by avoiding redundancy.

More recently, Nothdurft (2015) described a more sophisticated approach by modelling knowledge as a structural model depending on the domain. Here, it was differentiated between declarative i.e. the being of things, and procedural knowledge, i.e. know-how that can be applied to perform a task. The user's expertise was modelled as a probability distribution over a 5-point Likert knowledge scale ranging from novice to expert with intermediate levels. Similar to previous work, the expertise level was subsequently used to adapt the type and content of dialogue explanations and whether to provide explanatory messages at all.

Another current hot topic is the adaptation of the dialogue to the user's personality. Typically, the personality in a user model is represented based on the Big 5 model. For detecting the traits there exist several possibilities. Either, a questionnaire is used at the beginning of the interaction (e.g., see Fung et al. (2016)) or data-driven methods are applied. An extensive review of data-driven methods for detecting and adapting to personality features is presented in Ma et al. (2020). For example, personality can be inferred from extracting personal knowledge and possessing a long-term memory over user-related facts and feeding it to a Decoder-Encoder model. Furthermore, personality may be detected using text, audio, visual or multimodal features (Mehta et al., 2019). For adapting the dialogue to the respective user personality, typically the system output's content and wording are altered. This is usually done in such a way, that the system is perceived to have matching personalities (Metze et al., 2011). For example, Ahmad et al. (2020) introduced a personality adaptive conversational agent called Raffi. It used language cues that are specific to a particular personality dimension for mirroring the user's personality. Concerning tutoring systems, Vail and Boyer (2014) made use of adaptive dialogue strategies for adapting to either introverted or extroverted students. Introverted students were provided with additional prompts and encouraging messages to speak more openly about their minds, while extroverts were provided with reflective prompts to encourage discussions with the tutor. The authors presented study results showing that adapting a dialogue to the user's personality can have a positive effect on learning.

Other static adaptation approaches comprise the use of lexical alignment (Oviatt et al., 2004; Linnemann and Jucks, 2018), e.g. using the same words as the user, or adapting to the respective user gender (Liang et al., 2020). Furthermore, user preferences, e.g. which kind of food a user prefers in a restaurant search setting, can be considered for providing personalised and more relevant recommendations (Walker et al., 2004). Due to the limited scope of this thesis, these approaches are not reviewed in detail.

3.2.2. Dynamic User Adaptation

For tailoring the dialogue dynamically during the conversation, there exist various approaches. Among other things, the dialogue strategy can be adapted according to the user's emotions that may change throughout the dialogue (Andre et al., 2004). For example, Gnjatović and Rösner (2008) created an emotion-adaptive system for supporting users while they solve the Tower-of-Hanoi puzzle. Depending on the user's emotional state (negative, neutral, positive) the dialogue strategy was dynamically adapted. When the user was in a negative emotional state, high-intensity support was provided in the form of more informative dialogue content. Contrarily in neutral and positive states, the user was provided with low-intensity support only containing basic information. Nass et al. (2005) observed the pairing of user emotion and car voice emotion which resulted in increased driver safety and better user attitude during driving. However, the authors distinguished only between a positive and negative emotional state during the study. Contrary, Pittermann et al. (2010) presented an approach that incorporates all six basic emotions: anger, boredom, disgust, fear, happiness, and sadness (Ekman, 1993). The emotion recognition was based on extracting the relevant features from the speech signal and using the Rover (Recognizer Output Voting Error Reduction) algorithm for classifying the emotions. The dialogue's emotional wording was then adjusted to match that of the user. In a simulated environment, Papangelis et al. (2012) used an RL approach for optimising the dialogue strategy concerning the user's emotion. Therefore, the authors created a user simulator for sampling 16 different user emotions.

Especially for problem-solving and learning tasks, it is relevant to adapt to the user's fine-grained affective state instead of basic emotions. For example, Liao et al. (2006) proposed a dynamic decision framework to unify affect recognition and user assistance. In their work, they recognised affective states through active probabilistic inference from multimodal sensory data. User assistance was then automatically provided through a decision-making process that evaluates the benefits of keeping the user in productive affective states vs. the costs of performing user assistance. Their work only focused on affective states, specifically stress and fatigue but was not specific to conversational systems. Similarly, Friemel et al. (2018) argued that a real-time intelligent invocation of user assistance can be done via measurements of the cognitive-affective states with neurophysiological tools directly from the human body in real-time. They assumed that negative cognitive-affective states influence the users' behaviour, and therefore the need for assistance. Additionally, they proposed to examine the effects of negative cognitive-affective states on the users' behaviour when they were offered assistance. The cognitive-affective states could be measured via facial expressions with webcams and the user's mental effort via heart rate. Besides these decision-theoretic frameworks, there exist several affect-adaptive conversational systems.

For example, Bui et al. (2009) used statistical dialogue modelling for optimising dialogue strategies based on recognising the user's affective state. They tested their approach via simulation considering the affective state "stress" having five different manifestations from no stress to extreme stress. As a hypothetical use case, they considered route navigation in an unsafe tunnel, where the possibly stressed user needed a route description to coordinate with all other team members. Therefore, the user communicated with the DS to receive this information. As more stressed users tend to make more mistakes, the system was required to adapt its strategy for the dialogue to be successful. Evaluations in the simulated environment showed the advantage of the presented statistical models over hand-crafted models.

Similarly, Callejas et al. (2011) utilised an emotion recognition module for detecting, if a user was angry, bored, or doubtful. This information was subsequently used by the dialogue manager to adapt its strategy. For example, in case the user was doubtful and had alternating behaviour patterns during the dialogue, the system selected a system-directed initiative approach while also adding a help message after each prompt. Evaluation results verified that the adaptive version of the system performed better in terms of dialogue duration. Additionally, users subjectively rated the system better when adapting its behaviour to its affective state.

Litman and Forbes-Riley (2014) described the adaptation of a conversational tutoring system towards the affective states of disengagement and uncertainty in real-time. The classification of these states was learned on a data-corpus based on prosodic, lexical, and contextual features. For this, the authors used two binary classification models. Depending on the respective states the user either provided motivational (disengagement) or reassuring (uncertainty) messages. Comparing the system to a non-adaptive system, the evaluation results showed an increase in satisfaction, motivation, and task success.

Another way of dynamically adapting the dialogue to the user is the selection of appropriate dialogue initiatives and grounding strategies at each dialogue turn for achieving high user satisfaction. Thus, these approaches are considered quality or user-satisfaction-based adaptation approaches. In an early work by Chu-Carroll (2000), a mixed-initiative adaptive movie information system called Mimic was proposed. The system selected the type of initiative dependent on specific interaction cues, e.g. the user utterance contains ambiguous information and the dialogue history. The cues were then used to update probability distributions for selecting the appropriate strategy. A user evaluation of the Mimic system revealed that the adaptive version was able to outperform a non-adaptive version concerning user satisfaction and dialogue efficiency. Furthermore, it was found that Mimic's adaptive behaviour led to better user expectations of the system and resolved dialogue anomalies more efficiently.

Similar to this approach, Litman and Pan (2002) described an adaptive version of the telephone train schedule system Toot. Here, the dialogue strategy was selected dependent on the ASR performance. At each dialogue turn, the ASR confidence level was computed for providing the system with information about the uncertainty of user input. In case the confidence level was low, i.e. the dialogue was problematic, the system adapted its initiative strategy. For example, if the system started with a more user-initiative strategy, the system switched to a conservative system-directed strategy after registering speech recognition errors. The same procedure was used for adapting the grounding strategies. By adapting the dialogue strategy to the ASR performance, the authors could significantly improve the task success of Toot compared to a non-adaptive version.

Schmitt and Ultes (2015) extended this work by introducing the interaction quality (IQ) measure for user satisfaction recognition. IQ is an objective approach to measure the user's satisfaction with the dialogue (Schmitt et al., 2011). For estimating the IQ during an ongoing dialogue at each system-user exchange, interaction parameters (e.g. automatic speech recognition information, last system action) from three dialogue modules (ASR, NLU, DM) as well as temporal features are computed. Temporal features are calculated as means and counts of exchange level parameters on a window level (previous three turns) and dialogue level (up to the current exchange). These parameters are then fed to a SVM or LSTM-based architecture (Rach et al., 2017) in order to generate an estimate of the IQ on a 5-point Likert scale.

It was shown that IQ correlates well with user satisfaction (Ultes et al., 2013). For creating quality-adaptive dialogue, rule-based and statistical approaches were proposed (Ultes, 2015). These were implemented in a hidden-information state dialogue manager that could handle bus schedule information (Heinroth et al., 2010). Here, the user state was extended by including the IQ value. By adapting the grounding strategy and the initiative, the authors could show the high usability of the adaptive strategy approach which outperformed non-adaptive and random strategies. Furthermore, it was shown that IQ could be used for building a reward function for RL (Ultes et al., 2019). In Ultes et al. (2017), the authors described a method for learning a suitable dialogue policy by maximising the IQ using RL. This approach showed to increase task success and outperform hand-crafted strategies.

Besides considering user satisfaction for adapting the interaction, there exist several works on adaptive systems and trust. Trust is particularly important in user-adaptive interaction because adaptive behavior raises several issues related to trust, including controllability, privacy, intrusiveness, breadth of user experience, predictability, and transparency Jameson (2007).

For example, Cramer et al. (2008) studied the effect of transparency on the trust of a user-adaptive recommender system. They found that explanations of the system's decision-making helped the user to build an understanding of the system's functioning, thus having a positive impact on trust.

Nothdurft et al. (2014) continued this approach by including a fine-grained trust model in the user state for dialogue adaptation. Although their work is theoretical, they proposed an explanation framework that automatically augmented the dialogue with a transparency explanation, if trust in the system was endangered. For this, the augmentation process was structured as a POMDP with the sub-bases of trust according to Madsen and Gregor (2000) representing the states and affective states (e.g., confusion, frustration) being the observations. For example, the system could detect whether the user was confused. If so, it inferred the reliability and perceived competence of the system to be as currently low. As a result, the explanation framework would interrupt the current dialogue flow of the DM and trigger an explanation for reacting to this incomprehensible situation.

Similarly, Akash et al. (2020) used a POMDP model to determine the adaptive trust calibration in an automated driving take-over scenario. The authors included the user's trust in the system as well as the mental workload in the user state. Using a reward function designed to calibrate trust, the authors trained a policy for influencing the driver's trust level and workload by controlling the automation transparency level. The transparency level depended on the user's current trust and workload level as well as automation reliability and traffic complexity.

Similar to the topic presented in this thesis is the work by Hammer et al. (2015). In their work, they examined trust-based decision-making for smart and adaptive environments. For this, the authors developed a computational trust model that derived trust from a set of trust-based dimensions: Comfort of Use, Transparency, Controllability, Privacy, Reliability, Security, Credibility, and Seriousness. The relationships between trust and its dimensions were modelled as conditional probabilities in a Bayesian Network. Using a Bayesian Network allowed us to predict a probability distribution over different levels of trust depending on the learned conditional probabilities. Furthermore, the authors distinguished between initial trust and interaction-based trust to incorporate the trust dynamics. The trust dimensions were modelled as hidden variables that could be only observed via user state and social and environmental context. The trust model was then used in a smart office context for deciding on appropriate proactive actions to maintain the system's trustworthiness. For initialising the trust model, the authors gathered data online. In doing so, an accurate mapping between proactive system actions dependent on a specific situation or event and the system's perceived trustworthiness could be figured. The learned policy was then evaluated in a live study. The results showed that the context-adaptive proactive actions could indeed maintain user trust in the system. However, the authors did not examine how accurately the trust model could predict the user's current level of trust.

Closely related to trust is the concept of rapport which is used to describe harmonious relationships both in HHI and HMI. In this regard, Pecune and Marsella (2020) studied an RL-based approach for developing conversational strategies that were ought to achieve both task success and rapport between user and system. For this, the authors made use of a user simulator that included a rapport estimation module (Jain et al., 2018a) and equipped a CA with task-oriented and rapport-building behaviour, e.g. small talk, and self-disclosure. Further, they included the estimated rapport besides task metrics in the reward function for optimising both task and social dialogue policies. Training and testing the CA with the social user simulator showed the usefulness of their approach.

3.2.3. Conclusion and Research Gaps

For both, static and dynamic user adaptation approaches, adequate user modelling is an integral aspect. In related work, several user-specific characteristics were represented in a user model, including user expertise, personality, satisfaction, and affective state. However, few works considered trust for user modelling. Therefore, we intend to add to this line of research by developing a sophisticated user model incorporating a user's perception of the system. This would enrich a system with a simplified TOM for making assumptions about its trustworthiness. For trust-based user modelling in dialogues, we adopt methods of user modelling in recommender systems that rely on both user behaviour and user group-specific information. This information is deemed to be beneficial for representing trust in a user model. For realising the user model, we adopt and extend previous work regarding dynamic user adaptation by including not only system-, and context-related information, but also static user information. The created user model is novel, as it is the first to represent the trust development during an ongoing dialogue. This way, we aim for real-time measurement of an important metric indicating the success of human cooperation. This allows to optimise proactive dialogue strategies regarding their trustworthiness, pushing the boundaries of the state-of-the-art.

For adapting a dialogue, there exist several approaches. Mostly, the content of system messages is adapted or additional messages for encouraging or motivational purposes are provided. Also, the dialogue initiative and grounding behaviour may be adapted. Contrary to related work, we adapt the level of proactive dialogue behaviour by utilising a fine-grained set of proactive dialogue act types. In doing so, we aim to provide a more flexible way of proactive dialogue increasing trust and usability for improving cooperation.

Adaptation behaviour concerning user trust primarily involved transparency behaviour in the form of providing explanations and the modality of system messages. One work considered adapting proactive actions utilising a trust model to maintain a system's trustworthiness. However, this work considered proactive behaviour in smart environments with a low degree of cooperation between system and user, e.g. the system was used to switch on the lights automatically when the user arrived. Further, no proactive dialogues were considered. Therefore, our approach is the first to provide methods for including trust for dialogue adaptation.

Further, related work used Bayesian Networks for learning appropriate mappings between user trust and adequate proactive system action, we applied a sequential decision-making method in the form of a RL-based approach for learning adequate user-adaptive proactive dialogue strategies. A review of related work in the dialogue domain revealed that RL-based approaches are proven to be quite useful for optimising task effective dialogue behaviour. However, optimising user-perceived trust during dialogue for improving cooperation is underrepresented. Even though the work presented by Ultes et al. included user satisfaction in the form of an IQ-value in the reward function, this metric solely considered the functionality of the system and did not reflect the trustworthiness of the DS. Jain et al. (2018b) proposed to include rapport, a concept close to trust, in the reward function for achieving both socially and task-effective dialogue behaviour. However, there no proactive dialogue was considered and the focus was set on the relationship between user and machine and not on cooperation. Therefore, this thesis deals with novel work on adapting the proactive dialogue by including a trust-based user model for achieving both trustworthy and task-effective cooperation between the user and CA.

3.3. Summary

This chapter presented the state-of-the-art regarding proactive HMI and user-centred DSs. We first reviewed various modelling approaches and user perceptions of proactive behaviour in HMI before considering static as well as dynamic user adaptation approaches in DS. During the review process, we identified several research gaps.

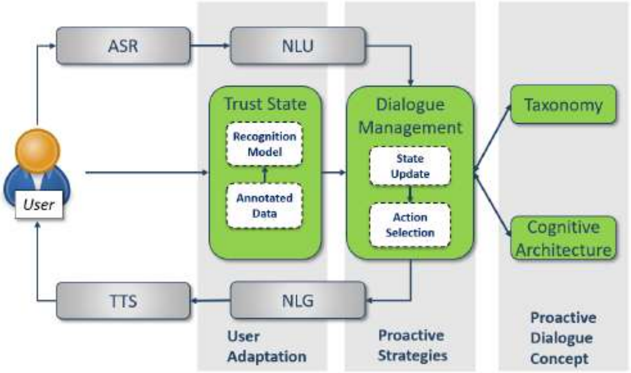


Figure 3.2.: The novelty of our work. All parts of the DS addressed by the novelty of this thesis are displayed in green including the Trust State, Dialogue Management, Taxonomy, and Cognitive Architecture.

First, related work regarding proactivity in the dialogue domain is sparse. However, there exists extensive research on this topic in other areas of HCI and autonomy. This knowledge may be used to transfer the concept of proactivity into the dialogue domain and to allow its structured technical realisation in conversational systems. Further, the evaluation of proactivity in HCI from a user-centered perspective is limited. For the creation of novel proactive dialogue strategies, it is therefore required to observe and evaluate also user-dependent features. This allows gaining in-depth knowledge of proactive behaviour for conversational assistance, e.g. which user characteristics influence the perception of proactivity or the identification of adequate proactivity types for specific user groups. Consequently, this information may be used for dialogue modelling to improve human-machine cooperation. Contrarily to related work, which primarily relied on the WoZ paradigm for testing the effects of proactivity on the user, we aim to evaluate using implementations of realistic prototypes. This way, we not only shed light on the theoretical implications of proactive behaviour but also elucidate the challenges and requirements of its practical realisation.

For the implementation of a more flexible and dynamic approach to proactive dialogue modelling, instead of the rather rigid proactive behaviour in related work, we identified adequate user modelling to be essential. Research on user-centred dialogue systems provides several different approaches for including the user in the DM process. However, none of them sufficiently describes the inclusion of a trust metric for decision-making during dialogue, which we deem important for two reasons: the measurement of the HCT relationship during dialogue would enable a direct way to evaluate the success of cooperation between human and computer. Finally, this provides the possibility to optimise cooperation using ML techniques, which have shown to be effective for user-adaptive dialogue modelling. In Fig. 3.2, we illustrate the novelty of our approach. We model proactive behaviour for DSs by developing a taxonomy of proactive dialogue and describing a cognitive architecture for realising proactive behaviour in CAs. Further, we investigate several proactive dialogue strategies and their relation to the concept of trust and usability to improve cooperation. For this, we include mechanisms for deciding on proactive behaviour in the DM module. Finally, we implement a user-adaptive proactive dialogue model. This includes the user's trust state in the user model for creating trustworthy proactive dialogue strategies with high usability.

Prior to the development of the proactive dialogue model, we conducted two initial experiments for exploring the effects of proactive behaviour on cooperation. These experiments, which are presented in the following chapter, are a first trial of transferring proactive behaviour in the realm of DSs. For this, we provide a simplistic embedding of current approaches to proactive behaviour, i.e. recommendations and notifications, in the dialogue. This was necessary to examine the effect of state-of-the-art proactive behaviour on aspects of cooperation by the means of trust and usability. Further, this allowed us to gain intuition and first insights into how specific user features influence the perception of proactive behaviour. Based on the outcome of the experimental studies, related work, and the observed background, we then derived a proactive dialogue model with a focus on improving human-machine cooperation.

4. Exploratory Analysis of the Effects of Proactive Behaviour on Cooperation

Reviewed literature has shown the importance of the concept of proactive behaviour on both HHI and HCI. The more intelligent and autonomous computer-powered machines become, the more humans expect them to become proactive and take actions unsolicited. However, most literature only considers proactive behaviour from the system's point of view, without extensively studying the user perception of proactive systems. With the goal of this thesis aiming at improving cooperation, it is integral to consider both, proactive interaction design and its effect on the user. To get an intuition of proactive dialogue affecting the cooperation, we needed to examine how state-of-the-art proactive interaction approaches – notifications and recommendations – can be transferred to the dialogue domain and how this influences the user perception. This was aimed to identify relevant user requirements for proactive dialogue modelling with a focus on human-machine cooperation. From a technical perspective, it was essential to embed proactive dialogue in realistic prototypes. In doing so, technical requirements for implementing proactive dialogue systems could also be identified. In summary, the exploratory studies answered the following two questions for establishing user and system requirements of adequate proactive dialogue modelling:

How does proactive dialogue influence usability? For successful cooperation, usability is one factor that can lead to its achievement as a proactive dialogue showing high usability may lead to task-effective cooperation. Therefore, it was necessary to observe how singular constituents of usability relate to proactive behaviour. Further, differences between reactive and proactive dialogue were needed to be identified for distilling user requirements.

Which components of Human-Computer Trust are influenced by proactive dialogue? Another prerequisite of successful cooperation is an adequate trust relationship between system and user. Here, it was particularly interesting to consider whether proactive dialogue affected the HCT and in which way. For example, the question needed to be answered if and how proactive behaviour related to cognitive- and affect-based trust. Congruently to our observations regarding usability, we compared differences in trust between reactive and proactive dialogue to identify user requirements for modeling trustworthy proactive dialogue strategies.

4.1. Effects of Conversational Recommendations on Cooperation

4.1.1. Motivation

In this first experiment, we studied the influence of proactive recommendations during the dialogue on the system's cooperation (perceived usability, trustworthiness) and the overall acceptance of the cooperation itself. Related work considered various aspects of proactive recommendations, however, not form a cooperation-wise perspective. Therefore, we observed whether recommendations affect trust in the system, and in which way usability and acceptance were influenced. For evaluation, a university-based restaurant information system was implemented as an Amazon Alexa application ("Skill") that was able to recommend meals. This domain was chosen as providing suggestions on food and nutrition is a popular topic among recommendation system researchers since the domain offers a large and confusing information space that motivates the need for autonomously-made recommendations. For example, Freyne and Berkovsky (2010) and Elsweiler et al. (2015) considered intelligent food or meal planning applications for a healthy lifestyle. Further, this domain offered to take user-dependent actions as people generally have specific preferences towards different kinds of foods. For identifying user requirements, we implemented several prototypes using different kinds of recommendations. As a baseline, we implemented a system that only provided reactive recommendations. Further, we implemented two kinds of proactive recommendation strategies. These strategies differed in their degree of invasiveness and the level of user control. In the introduction of this thesis, we illustrated controllability and privacy to be one of the major challenges of proactive behaviour. Thus, these aspects should be also considered concerning proactive dialogue modelling for adequate cooperation. In this first exploratory study, it was therefore observed whether there exist differences between proactive system dialogue in which the user has allegedly more control over the recommendations (explicit strategy) and proactive dialogue relying on automatically collected user data (implicit strategy) in which the user might not directly understand the system's reasoning behind the proposed suggestions. An explicit recommendation strategy was based on user preferences that had directly been provided by the user, i.e., a user could put favourite dishes or restaurants on a favourites list. Contrary, the implicit strategy used autonomously gathered information while interacting with the user, i.e., a meal or restaurant was put on the favourites list when the user asked for more details about them. In the following sections, we describe the scenario, the system prototype, and dialogue design, as well as the study setup, results, and discussion.

4.1.2. Scenario

For the creation of a test scenario, we implemented an information retrieval system assisting the user in planning their lunchtime at Ulm University. The system contains various information about the university's different restaurants, Mensa, Cafeteria West, Cafeteria B, Bistro, Burger Bar, West Side Diner.

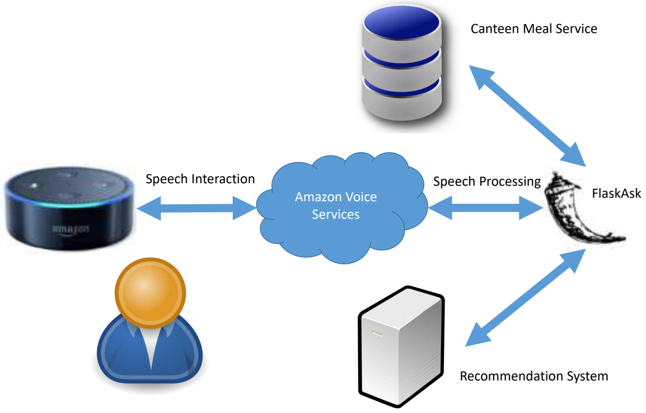


Figure 4.1.: Overview of the combined information retrieval and proactive recommendation system. A user interacted with Amazon's Alexa speech interface. Amazon Voice Services transformed the spoken words into text. Afterward, a FlaskAsk-Python application handled the dialogue management, i.e. retrieved meal or restaurant information from the canteen's database, and selected suggestions from the recommendation system. (Kraus et al., 2020a)

The information about the individual restaurants comprised their meal plans (ca. 70 different kinds of meals), meal-specific information (17 different food categories, price, name, ingredients, vegetarian yes/no, available as menu yes/no), as well as restaurant-specific information (location, opening times, background info). Thus, the domain fulfilled the prerequisite of sufficiently large information space for studying the effects of proactive recommendations. A meal plan for one of the restaurants consisted of several food categories, which were repeated over all weekdays. For example, herbal cream soup and potato soup could be assigned to the food-category "soup". While the categories remained the same over the days of the week, the offered dish of a category changed daily. The user could receive information about the different meals and restaurants by posing spoken requests to the system. The prototype implementation and the dialogue design are explained in the next section.

4.1.3. Prototype Implementation and Dialogue Design

The implementation of the system was based on Amazon's Alexa and its cloud-based speech recognition. For communicating with the Amazon Voice Services, the Python FlaskAsk [1] framework was used. All application data required for the system's functionalities, in particular for proactive meal suggestions, was stored in a database. This database contained all relevant information about the individual restaurants and meal plans for each day. An overview of the system's architecture is depicted in Figure 4.1.

[1] https://github.com/johnwheeler/flask-ask

Common Interaction	**Interaction Including Recommendation**
U: Alexa, start Mensa-Planner.	U: Alexa, start Mensa-Planner.
S: Hi, I am the Mensa-Planner. I assist you finding your favourite meals. How can I help you today?	S: Hi, I am the Mensa-Planner. I assist you finding your favourite meals.
U: What can I eat in the Burger-Bar today?	S: Today, I recommend you a cheeseburger in the Burger-Bar.
S: The Burger-Bar serves cheeseburger, chickenburger, ...	U: Ah okay, nice. Thank you Mensa Planner.
U: How much is a cheeseburger?	S: See you soon!
S: The price for cheeseburger is around 2 Euro.	
U: Is a cheeseburger available in a menu?	
S: Yes, a cheeseburger is available in a menu with french fries.	
U: Ok, put Cheeseburger on my list.	
S: I put Cheeseburger on your favourites list.	

Table 4.1.: Examples dialogues representing the common (*reactive*) interaction flow (S: system; U: user). Left: a typical dialogue between Alexa and user. The green-coloured text represents the functionality of the *explicit* proactive strategy. Right: interaction including a recommendation by the system. The cyan-coloured text represents the typical suggestion utterance for both (*explicit* and *implicit*) proactive strategies. Note that original interactions were conducted in German. (Kraus et al., 2020a)

Generally, the information retrieval system had two main functions: returning a list of meals and returning detailed information about one single meal. Both existed in several variants. A list of meals was returned by Alexa when users had asked for meals in a specific restaurant, meals from a specific category, when they had searched for meals by name, or when they had asked for vegetarian meals. In all cases, users could ask for meal lists of the current day or another weekday within the next seven days: e.g. *"What can I eat in the bistro on Tuesday?"*; *"What are tomorrow's meals in the category pizza?"*; *"Do they serve burgers today?"*.

The corresponding answer of Alexa either contained only a list of meal names when asking for meals in a specific restaurant or a list of meal names and the related name of the restaurant in all other cases. After receiving a list of meals, users could ask for more detailed information about every meal which was on the list before. More precisely, users could request information about restaurants, food categories, descriptions, and pricing information of meals: e.g. *"What do you know about vegetable soup?"*; *"How much is pizza salame?"*; *"Where can I find this?"*. For providing proactive system behaviour, two recommendation variants during dialogue were implemented: *explicit*, and *implicit* proactive recommendation strategy. Both extended the described basic functionalities by collecting user preferences and suggesting appropriate meals to users. Hence, users were provided with personalised active assistance to retrieve information more convenient and time efficient. The *reactive strategy* did not provide recommendations.

The *explicit* proactive recommendation strategy suggested meals by utilising user preferences that were managed by the user directly. Here, personal *favourites* lists were applied that could be directly manipulated. These lists contained preferred restaurants, food categories, and meals. Users were able to explicitly add or remove content by saying, for example, *"Put a cheeseburger into my favourites."* or *"I don't like this anymore."*. The system then recommended one meal to the user depending on these favourite lists. An example of such system behaviour is presented in Table 4.1.

Contrary, the *implicit* proactive recommendation strategy suggested meals by utilising automatically-gathered user preferences. For this, the user's previous behaviour was considered. To illustrate this strategy, consider the examples in Table 4.1. Every time users searched for a meal, a food category, or a restaurant, the search name was added to a corresponding recommendation list with the value of one. In the presented example, *"What can I eat in the Burger-Bar today?"* added "Burger-Bar" to the list of favoured restaurants. When asking for detailed information about a meal, its name was added to the list of favorite meals. Additionally, its category and restaurant were added to respective lists as well. If an entry with the same value already existed, its value increased by one. As the user asked for detailed information about cheeseburgers in our example (*"How much is a cheeseburger?"* and *"Is a cheeseburger available in a menu?"*), the values of "cheeseburger", "Burger-Bar", and "Burger" were increased respectively. The system then recommended meals depending on the highest rated entries of the favourites lists, e.g. *"Today, I recommend you a cheeseburger in the Burger-Bar"*.

4.1.4. Experimental Design

The developed study design consisted of three independent between-subject conditions to which participants were randomly assigned. Conditions were as follows:

Reactive: Participants used the basic version of the implemented information retrieval system that provided no recommendations at all. This was used as a baseline condition.

Proactive (explicit): the system provided suggestions at the beginning of each dialogue based on the previously described *explicit* proactive recommendation strategy, where participants managed their preferences themselves.

Proactive (implicit): the system provided suggestions at the beginning of each dialogue based on the previously described *implicit* proactive recommendation strategy relying on automatic measures of previous user behavior.

Participants

19 German participants (52.6 % female) with an average age of 23.37 ($SD = 4.06$) were recruited and received 10 € in return for their participation. 15 participants were students, while the other 4 were employees of our university. A condition of participation was actually visiting the canteen on a regular basis.

Recommendation Strategy	**Acceptance**	**SRA**	**Likeability**	**Cognitive Demand**	**Satisfaction**	**Habitability**	**Motivation**
	M (SD)	*M (SD)*	*M (SD)*	*M (SD)*	*M (SD)*	*M (SD)*	*M (SD)*
Reactive	4.09 (1.26)	3.50 (.88)	3.74 (1.04)	3.79 (1.22)	4.25 (1.04)	3.67 (.98)	5.28 (.61)
Proactive (explicit)	4.79 (1.66)	3.81 (.73)	4.55 (.94)	4.58 (.88)	5.54 (.70)	4.38 (.93)	6.22 (.54)
Proactive (implicit)	4.30 (.84)	4.17 (.77)	4.67 (.63)	4.64 (.64)	5.36 (.93)	5.21 (.80)	6.19 (.74)

Table 4.2.: Descriptive statistics of the measured dependent variables with reference to the recommendation strategies. Results for cognitive demand are inverted (the higher, the better). SRA implies system response accuracy. (Kraus et al., 2020a)

Experimental Procedure

As a cover story the participants were told that they would test the new Alexa menu assistant of the university for usability. Furthermore, participants were instructed to interact with the Alexa Skill for about five minutes daily over a period of 5 weekdays and to evaluate the application in a questionnaire at the end of the study. For the duration of the evaluation, they were provided with an Alexa Echo Dot on loan. Additionally, participants received detailed online study instructions. These instructions contained example utterances as well as an explicit task description.

First, participants had to make meal queries (at least one) choosing from three request types: query of meals offered by specific restaurants, the query of meals of a specific food category, or an explicit search for meals. Subsequently, they had to make requests (also at least one) for details about specific meals (either all details or particular details, e.g. food price) or restaurants (description, opening times, location). The user could also request help from the system if he or she was unsure about what to say. Furthermore, the system was able to prompt the user again in case of low speech recognition accuracy. After participants repeated the task for 5 days, they received access to the online questionnaire. Upon successful completion, they were then given a reward for participation.

Questionnaires

In our experiment, the acceptance of the cooperation (scale developed by Van Der Laan et al. (1997)), the usability of the system, and trust in the system were assessed as dependent variables. For measuring the usability, the SASSI questionnaire (Hone and Graham, 2000) was employed. Trust was assessed using the German version of the trust in automation scale (Kraus, 2020). Propensity to trust was measured using the questionnaire provided by Merritt et al. (2013). In addition, the participant's motivation to interact with the system was measured using a scale developed by McAuley et al. (1989). All scales were translated into German, as the experiments were conducted using German participants. Additionally, the scales were slightly modified for content and study context.

4.1.5. Results

For data analysis, a one-way analysis of variance (ANOVA) was used for guaranteeing no significant confounding variables and for testing the significance of described reactive and proactive (explicit, implicit) recommendation strategies. To rule out confounding group differences for the study conditions, the participants' experiences with existing speech assistants (Alexa, Google Home, Siri,...) were controlled for. There were no significant group differences ($F(2,16) = 0.001,\ p >> .05$). In addition, participants' ages and genders were similarly distributed in the different experimental groups and no outliers were found in the data set. An overview of the results is presented in Table 4.2.

Effects of Proactive Dialogue on Usability

The data analysis revealed significant differences between conditions for the dependent variables *Satisfaction* ($F(2,16) = 3.65,\ p < .05,\ \eta^2 = 0.31$), *Habitability* ($F(2,16) = 4.81,\ p < .05,\ \eta^2 = 0.38$), and *Motivation* ($F(2,16) = 4.25,\ p < .05,\ \eta^2 = 0.35$). According to Cohen (1988), the limits for the size of the effect are .01 (small effect), .06 (medium effect) and .14 (large effect). According to these rules of thumb, the effect of our ANOVA was considered as large.

For clarifying which conditions differed significantly, post-hoc t-tests were conducted. For *Satisfaction*, the explicit proactive recommendation condition was rated significantly higher than the reactive condition ($t(10) = 2.53,\ p < .05$) The difference between implicit and reactive strategy ($t(11) = 2.03,\ p = .07$) was not significant, however, may become significant with increasing n of participants. *Habitability* was rated significantly higher for the implicit vs. reactive recommendation condition ($t(11) = 3.14,\ p < .01$). Both proactive recommendation strategies were rated significantly higher than the reactive condition for *Motivation* (explicit vs. reactive, ($t(10) = 3.03,\ p < .05$; implicit vs. reactive, ($t(11) = 2.39,\ p < .05$). There were no significant differences between the proactive test conditions.

Effects of Proactive Dialogue on Trust

Further, the trust trajectory throughout the experiment was investigated by measuring the differences between the user's propensity to trust in advance of the experiment and the rated perceived trust after the experiment. Using a paired t-test, we found a tendency for a decrease in trust in the system for the explicit proactive condition ($t(7) = 2.25,\ p = .065$), which may become significant using a larger number of participants. For the other conditions, trust was increased but not significantly. The trust trajectories for each condition are depicted in Fig. 4.2.

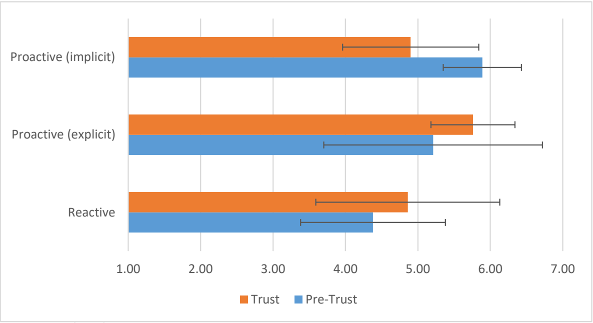


Figure 4.2.: Trust trajectory for each condition measured at two time steps: before the interaction (Pre-Trust), and after the interaction (Trust).

4.1.6. Discussion

The study revealed differences between proactive and reactive conditions. In the following, we discuss the results concerning the two research questions.

How does proactive dialogue influence usability?

Both proactive recommendation strategies received, concerning the explicit strategy significantly, higher ratings for user satisfaction. Hence, a preemptive system led to a satisfying dialogue for information retrieval. Furthermore, both proactive strategies motivated the user significantly more to interact with the system than the reactive strategy. This is a strong indicator that the dialogue incorporating proactive elements was considered more encouraging. Since motivation was measured after 5 days of usage, proactivity seemed to be a factor in increasing user engagement. The implicit proactive recommendation was also rated higher on habitability. According to Hone and Graham (2000), a "habitable system may be defined as one in which there is a good match between the user's conceptual model of the system and the actual system." Consequently, users could build a better model of the system's functionalities, when it implicitly tracked their behaviour for recommendations. Thus, the user was able to build an adequate theory of mind of the system's actions. This implicated that the tracking of user behaviour seemed to work quite well. Also, the explicit proactive recommendation strategy was rated higher for habitability than the reactive strategy. However, the difference was not significant.

Overall, the proactive strategies were perceived as more user-friendly which was backed up by positive tendencies in the measurements of acceptance, cognitive demand, and likeability of the cooperation. This was in line with the results provided by Peng et al. (2019). There, a medium-proactive robot was perceived as more appropriate and helpful in decision-making tasks. However, they used a WoZ set up in contrast to our approach.

Hence, our study suggests that the results may be transferred to real functioning proactive dialogue systems. Although there were no significant differences between the two types of proactivity, some observations were possible. The implicit recommendation strategy was rated higher in system response accuracy and habitability. Thus, it can be cautiously assumed that users found implicit system behaviour to be more understandable as both factors indicated whether the system worked according to the user's intentions and expectations.

The explicit recommendation strategy had higher ratings for user acceptance. This could indicate that users want to be in charge when providing information about their preferences and may not fully trust a system that autonomously collects their data. For observing this claim more in detail, the trust ratings of the strategies need to be investigated.

Which components of Human-Computer Trust are influenced by proactive dialogue?

This study only investigated whether trust, in general, was affected by the proactive interaction and did not consider the sub-bases of trust. Observing the trust trajectory, we found that the implicit recommendation strategy was the only one to lead to a trust decrease. According to Glikson and Woolley (2020), trust in virtual CAs typically is high at the beginning of interaction and decreases over time due to a lack of calibration between system behaviour and its actual level of machine intelligence. However, as the implicit recommendation strategy led to a good understanding of system behaviour, this was not deemed the reason for trust decrease. Continuing our argumentation about a user's preference for explicitly providing the system with data, this may be the true reason for the trust decrease as was also visible in the acceptance ratings. However, this also possibly stemmed from using Alexa Echo Dots for evaluation. Hence, participants could have been biased towards the brand. Although understandability forms a sub-base of cognitive trust, it did not affect a trust increase of the explicit strategy. For this reason, it may be relevant to focus on the other cognitive bases of trust, perceived competence, and reliability. However, no clear effects which would help to determine the trustworthiness of proactive dialogue were found.

Limitations

A disadvantage of the experiment was the quite low number of participants. Using a higher number could have provided more comparable results between the explicit and implicit strategies. Furthermore, the speech recognition errors of Amazon Alexa were troublesome, as several participants of the study reported issues. For example, participant 8 (female, 26) reported that Alexa understood several meals, but some did not. In addition, participant 2 (female, 27) described that Alexa often did not understand the meals, when she had asked for details about a dish. We are aware that a Wizard-of-Oz setup would have prevented such errors. However, the creation of a realistic setup with an experimental duration of 5 days, was deemed impracticable and not expedient in the context of our evaluation.

4.1.7. Conclusion

This experiment dealt with the evaluation and comparison of different types of proactive system behaviour and their effects on the usability and perceived user trust. Therefore, two proactive dialogue strategies that differed regarding data acquisition methods for providing user-adaptive recommendations were implemented in a realistic task scenario of planning university restaurant visits. Here, Amazon's Alexa conversational framework was applied for creating an information retrieval system with recommendation functionality in the restaurant domain. The study results provided evidence, in line with recent research, that proactive behaviour can positively improve the usability of virtual CA when compared to reactive behaviour. Particularly, we found that proactive dialogue has a major influence on usability features, user satisfaction, engagement, and on the creation of a theory of mind of system behaviour. However, users tended to trust proactive system behaviour better when they were more in charge of the machine's intelligence capabilities, i.e. when they explicitly guided the system what their food preferences were. Although no clear results were found in this regard, the findings indicated that controllability and privacy may be also relevant for consideration in proactive dialogue.

4.2. Effects of Notifications and Topic Switching on Cooperation

4.2.1. Motivation

In the second experiment, we studied the influence of proactive notifications and topic switching on the system's perceived trustworthiness, usability, and acceptance of the cooperation. Similarly as described in the previous study, there exist several works on the effect of proactive notifications and topic switching on task effectiveness but not on trust, which is deemed to be the other important factor for successful cooperation. As the observation of general trust in the previous study did not reveal clear results, we considered the effects on the sub-categories of trust, cognitive-based, and affect-based trust in this study. For evaluation in a realistic environment, we implemented a natural language chatbot using the open-source framework Rasa Stack [2]. The proactive strategies of the chatbot were realised in the form of sending push notifications to the user and actively switching topics during dialogue. Furthermore, we implemented relational dialogue strategies in the form of small talk and empathetic reactions to user input. Empathetic reactions and small talk are supposed to deepen relationships and foster trust between agents and users (Bickmore and Cassell, 1999; Brave et al., 2005). In the introduction of this thesis, proactive and empathetic behaviour were both considered to be indicators of a system's conversational intelligence. A comparison between proactive notification as well as topic switching during dialogue and empathetic strategies was thus intended to explore the impact of different conversational intelligence strategies on cooperation. For the study, we also implemented a baseline chatbot (reactive, non-relational) and chatbots

[2] https://rasa.com

demonstrating variations of the designed dialogue behaviour, i.e. proactive, non-relational and reactive, relational. Thus, we could investigate, whether there exists an interaction of proactive and empathetic dialogue on the trustworthiness of the CA.

4.2.2. Scenario

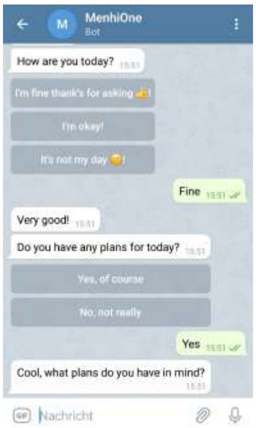
As a use case, we considered a prototypical mental health chatbot for mood and symptom monitoring. Using chatbots in e-health is an emerging topic over the last years and has resulted in various applications ranging from knowledge-based information agents to assist caregivers (Pragst et al., 2015) to digital assistants in an intelligent operating room (Miehle et al., 2017). Also, current research on chatbots in mental health suggests that the psychiatric use of chatbots is favourable, as it promotes self psycho-education and adherence (Vaidyam et al., 2019). The chatbot described in this section was designed in the scope of the EU-funded project *Mental health monitoring through interactive conversations* [3] which deals with researching and developing conversational technologies to promote mental health and assist people with mental ill-health (depression and anxiety) to manage their conditions (Benítez-Guijarro et al., 2020). Furthermore, a study by Benítez-Guijarro et al. (2020) showed that a functional requirement of such a chatbot is configurable proactive behaviour, e.g. in the form of notifications. Therefore, this scenario was an adequate way for testing the influence of proactive dialogue on the user and identifying user requirements.

The main task of the chatbot was to interact with the user for having a daily mood check-in. For the daily check-in, 12 items of the positive and negative affect schedule (PANAS) (Watson et al., 1988) questionnaire were used. This scale consists of several words that describe different feelings and emotions, e.g. interested, guilty, or active. Users indicated to what extent (not at all to extremely) they have this feeling or emotion. Additionally, users had the opportunity to write freely about the experiences of their day and their feelings. However, the study was conducted with participants having no history of mental health treatment due to ethical reasons. For creating a realistic experience, we implemented the chatbot for usage within the Telegram messenger framework. This allowed users to interact with the chatbot using their private phones under non-laboratory conditions. The system and interaction design are described in detail in the following section.

4.2.3. Prototype Implementation and Dialogue Design

The chatbots were implemented using the open-source framework Rasa Stack. This was due to the free availability and privacy issues, as personal data was not shared with external services. Rasa is a framework for creating conversational AI. The framework consists of two modules, one for dialogue control and one for NLU. Rasa Core is a dialogue management system that is designed for using ML to train a dialogue policy instead of a finite-state approach. The chatbot can learn through interactive learning by

[3] ref. no.823907, https://menhir-project.eu



How are you today?

Fine

Very good!

Do you have any plans for today?

Yes

Cool, what plans do you have in mind?

Figure 4.3.: Example dialogue with the RASA chatbot. The system initates the dialogue and then progresses to small-talk with the user. (Kraus et al., 2021a)

utilising so-called stories. A story is a representation of a dialogue between a user and the chatbot, converted into a specific format where user inputs are expressed as corresponding intents. The responses of the chatbot are expressed as corresponding action names. RASA NLU is a statistically-driven NLU service for intent classification, response retrieval and entity extraction.

When RASA receives a message from the user, it attempts to predict the intent and extract the entities present in the message. This part is handled by RASA NLU. Once the user's intention has been identified, the RASA stack performs a specific action. In the example, visualised in Fig. 4.3, the intent of the user's utterance "Fine" would be "express_mood_positive". Then RASA tries to predict what to do next. This decision is made taking into account several factors and is made by the RASA Core unit. In the example, RASA showed an empathetic reaction. It also predicted the next action that the model should perform - to continue with small-talk and to ask users about their current plans. The more sample data RASA has, the more likely it is that the right decision will be made. The model presented was trained with several stories and numerous example utterances for training the NLU.

TELEGRAM was used as platform for the interaction (see Fig. 4.3). RASA offers extensions to be easily implemented in a TELEGRAM chatbot. In addition, TELEGRAM is used widely and can be reached via mobile phone, tablet, and computer, which made it much easier for test participants to use it. A virtual cloud server was used to provide the chatbot. RASA offers the possibility to run an HTTP server that handles requests using a trained RASA model. Since a local server was used, the additional software Nginx (Reese, 2008) was installed to ensure the connection to TELEGRAM. The RASA NLU server was set behind an NGINX reverse proxy, where NGINX handled the secure sockets layer (SSL) for safeguarding sensitive data and then forwarded the data to RASA over hypertext transfer protocol (HTTP). So-called cronjobs were used to send the test persons a daily push message. Under many operating systems there is the so-called CRON-System

(Cron-Daemon), which makes it possible to execute automated tasks (jobs) at special times (Davidovi and Guliani, 2015). To send a message to users, their Telegram user IDs and Bot IDs were required. Then the text of the message and the desired time of the cronjob could be determined.

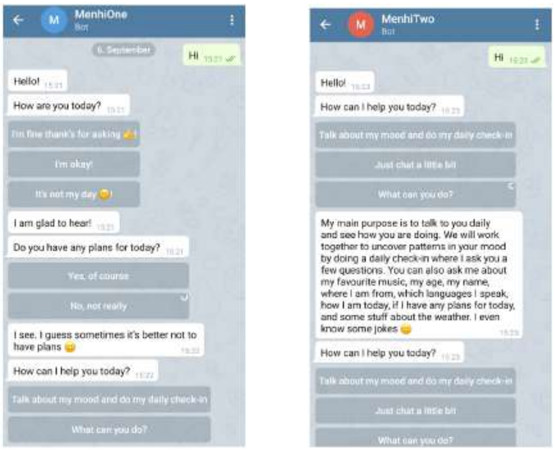
The dialogue with the chatbot was initiated by the user with a simple greeting, which was reciprocated by the system. Afterward, the bot initiates the daily check-in dialogue. For examining the relational and proactive dialogue strategies, we extended the basic functionality of the chatbot's mood-tracking dialogue correspondingly. The role of empathetic and proactive dialogue is explained in the following.

People use a variety of types of social languages, including small talk and empathy, to build collaborative, trusting interpersonal relationships. In particular, the two constructs small talk and empathy showed signs to increase trust by establishing a long-term social-emotional relationship with their users (Bickmore and Picard, 2005). Empathy is the mental process by which a person tries to understand the statements, behaviours, or feelings of another person, from the counterpart's perspective or preconditions. The term "empathy" is not used uniformly in psychology. In the present experiment, the social-psychological meaning of empathy was used (Linden and Hautzinger, 2008). Previous work showed that digital emphatic agents are perceived as more caring, sympathetic, and trustworthy than agents without emphatic abilities (Brave et al., 2005). Above all, effective answers that correspond better to the situation of another than one's own should serve as the main instrument for inducing empathy (Hoffman, 2001).

The relational bot showed different emphatic reactions in different situations. During the daily check-in, for example, the bot repeatedly showed his appreciation and understanding for the user during very personal topics or provided an appropriate reaction to a negative mood on the day of the check-in, e.g *"Thank you, I appreciate you talking to me about this."* or *"I know the questions are not always easy to answer but you are doing great."*.

People use small talk, to establish interpersonal collaborative trusting relationships (Cassell and Bickmore, 2002). Research in the field of conversational agents showed that it is not enough to limit conversations between agents and people to task-oriented topics (Bickmore and Cassell, 1999). The results suggest that small talk supports deepening relationships and building trust between virtual agents and users. Therefore, the developed chatbot was able to deal with topics such as the users' music preferences, personal details, feelings, current plans, and some other topics like the weather. For an example, see the conversation depicted in Fig. 4.3.

For integrating proactive dialogue behaviour, the initiative of the chatbot was manipulated in the form of notifications that intelligently remind users of their daily check-in, as well as active topic switching strategies of the chatbot, e.g. it started small-talk directly and changed topics automatically. A comparison of the different variants is visualised in Fig. 4.4.



Hi

Hello!

How are you today?

I am glad to hear!

Do you have any plans for today?

I see. I guess sometimes it's better not to have plans 😊

How can I help you today?

Hi

Hello!

How can I help you today?

My main purpose is to talk to you daily and see how you are doing. We will work together to uncover patterns in your mood by doing a daily check-in where I ask you a few questions. You can also ask me about my favourite music, my age, my name, where I am from, which languages I speak, how I am today, if I have any plans for today, and some stuff about the weather. I even know some jokes 😊

How can I help you today?

Figure 4.4.: Comparison of the proactive (left) to the reactive dialogue (right). (Kraus et al., 2021a)

4.2.4. Experimental Design

The baseline study was realized in a 2 x 2 between-subject experimental design. The empathetic (relational, non-relational) and the proactive conditions (proactive, reactive) were implemented as two-step factors in four individual chatbots (e.g. relational-proactive, non-relational-proactive, ...). This resulted in four experimental groups in total.

Participants

Participants were recruited at the University. A prerequisite for the participation was the possession of a mobile phone with the messaging program Telegram, as well as a fluent knowledge of the English language. As an incentive, the participants were promised a 10 € amazon voucher. All test persons were informed about the scientific purpose, as well as about the anonymous use of their data. A total of 41 (26 female) people took part in the experiment. However, 5 persons had to be excluded due to insufficient language knowledge (minimum B2). participants were between 19 and 30 years old ($M = 24.83,\ SD = 2.93$). Most of the participants were psychology students or had an academic degree. They were randomly distributed to each study group.

Experimental Procedure

Participants had to interact with one of the chatbots on three consecutive days. As a cover-up, they were told to test a novel chatbot on Telegram. Thereby, their emotional state was checked in the form of a daily check-in. Before, the experiment users had to provide general information, e.g demographics, personality, or experience with chatbots. During the experiment, the dependent variables were collected at two different measuring points (after the first and last interaction with the chatbot). The mean values were used for the evaluation for a more robust result. However, we also evaluated the trajectories of dependent variables to include the dynamic effects of the dialogue strategies. In addition, the participants had the opportunity to note any problems, impressions, or irregularities in the conversations at the end of the two intermediate test questionnaires. A total of three conversations were carried out per respondent, which took place on three consecutive days. Any effects of the empathetic dialogue strategies should thus show their effect.

Questionnaires

Validated psychological scales were used for testing the dependent variables. To rule out confounding variables the participants' technical affinity (Karrer et al., 2009), and the predisposition to trust (Merritt et al., 2013) were recorded before the experiment in combination with demographic data. Further, we included the BFI-10 by Rammstedt et al. (2013) for personality assessment. Trust was measured using the Trust in Automated Systems Scale (Jian et al., 2000). Furthermore, scales for measuring the bases of trust developed by Madsen and Gregor (2000) were used. Usability was studied with the SUS (Brooke, 1996).

4.2.5. Results

For an exploratory data analysis, a multivariate ANOVA was conducted for guaranteeing no significant confounding variables and for testing the significance of the empathetic and proactive strategies. Confounding group differences for the study conditions could be ruled out as we found no significant differences except regarding the proactivity of the chatbot. Participants who interacted with the reactive version of the chatbot had an almost significantly higher technological affinity as compared to the proactive group ($t(34) = -2.01,\ p = .053$). Therefore, this variable was used as a covariate to make up for noisy data when considering the proactive strategies. However, there were no interaction effects between empathetic and proactive strategies (all p-values > 0.05). For further investigations, the effects of the strategies were investigated separately. Therefore, we paired the individual samples and evaluated two study groups in each case: relational vs. non-relational and proactive vs. reactive. The results can be found in Tab. 4.3.

Comparisons between Empathetic Strategies

A notable tendency was found when considering the empathetic strategies, that may become relevant with an increasing number of participants. Personal attachment was rated higher for the non-relational strategy ($F(1, 31) = 3.14,\ p = .086$), but not significantly. For the other dependent variables no significant results were found.

Comparison between Proactive Strategies

Regarding the proactivity of the chatbots, we found two interesting tendencies. First, participants rated the reliability of the proactive chatbot higher compared to the reactive version($F(1, 31) = 3.92,\ p = .057$). Additionally, the understandability of the proactive chatbot was rated higher ($F(1, 31) = 3.26,\ p = .081$).

	Dialogue Strategy			
	Relational	Non-Relational	Proactive	Reactive
Trust	4.63 (.20)	4.69 (.21)	4.89 (.21)	4.41 (.21)
Reliability	4.06 (.27)	4.25 (.29)	4.58 (.29)	3.74 (.29)
Competence	3.62 (.30)	3.63 (.31)	3.84 (.31)	3.42 (.32)
Understand-ability	4.28 (.26)	4.43 (.27)	4.71 (.27)	4.00 (.27)
PA	1.90 (.20)	2.41 (.21)	2.24 (.21)	2.06 (.21)
Faith	2.08 (.23)	2.12 (.25)	2.19 (.25)	2.01 (.25)
Usability	2.63 (.09)	2.51 (.09)	2.61 (.09)	2.52 (.10)

Table 4.3.: Descriptive statistics of the measured dependent variables with reference to the dialogue strategies (means and standard errors). All variables measured on 7-point Likert scales (Kraus et al., 2021a).

Looking into the trajectories of the dependent variables over the course of the experiment, we found several noteworthy results for perceived trust, understandability, personal attachment and acceptance. For both, proactive and reactive version, there was a significant increase in perceived trust for the first interaction as compared to the user's propensity to trust (proactive version $t(17) = -5.97,\ p < .001$; reactive version $t(17) = -2.99,\ p = .008$). The trust trajectory for each condition is depicted in Fig. 4.5. Considering the differences between the measurements for the first interaction and the final evaluation, we found a tendency in increase of understandability ($T_1: \ M = 4.22\ SD = 1.49,\ T_2: \ M = 4.73\ SD = 1.31;\ t(17) = -1.98,\ p = .064$) and acceptance ($T_1 : M = 4.17 SD = 0.43, T_2 : M = 4.29 SD = 0.29, t(17) = -1.95,\ p = .068$) for the proactive version. In addition, personal attachment towards the reactive version showed a tendency to decrease ($T_1\ :\ M\ =\ 2.44\ SD\ =\ 1.15,\ T_2\ :\ \ M\ =\ 1.93\ SD\ = 1.19,\ t(17) = 1.93,\ p = .070$).

Investigating the effects of personality traits on the perception of the proactive and reactive version of the chatbot, we found interesting tendencies regarding the user's level of neuroticism. Using Mann-Whitney-U tests, we found that study participants with high neuroticism had the tendency to rate the interaction with the proactive version higher regarding understandability ($M = 3.80,\ SD = 0.65\ vs.\ M = 5.17,\ SD = 0.40;\ U = 12,\ p = 0.057$) and personal attachment ($M = 1.90,\ SD = 0.41\ vs.\ M = 3.30,\ SD = 0.61;\ U = 12,\ p = 0.057$). However, these results should be considered with caution, as they were not significant. For the other personality traits, no significant differences were found.

4.2.6. Discussion

As the results did not reveal any differences between proactive and reactive system behaviour regarding usability, this discussion solely focuses on our second research question but the observation will be examined in the conclusion of this section.

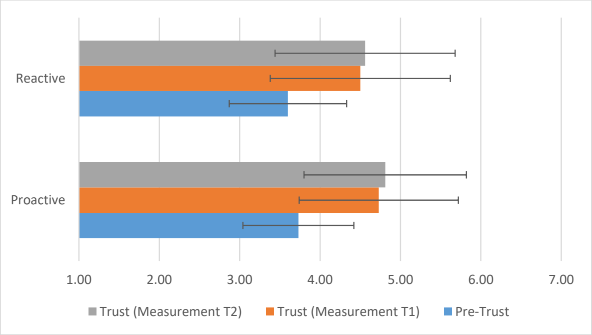


Figure 4.5.: Trust trajectory for the reactive and proactive strategy measured at three time steps: before the interaction (Pre-Trust), during the interaction (T1), and after the interaction (T2).

Which components of Human-Computer Trust are influenced by proactive dialogue?

Although no significant results could be reported, several interesting tendencies that require further investigation were identified. It was found that a proactive chatbot was perceived as more reliable and understandable. Proactive behaviour, like in our case, push notifications and more dialogue control by the system, seemed to have a positive effect on overall trust and perceived competence as well. Thus, the cognitive-based HCT components may be mostly influenced by proactive system actions. This may be also an indicator that the perception of proactive behaviour in HHI could be transferred to some degree into the HMI domain. In organisational management, for example, it was also shown that proactive behaviour can be positively related to perceived competency (Parker et al., 2010). Observing the trajectories of the measured dependent variables during the experiment, we also found that the longer the interaction lasted, the further the understandability and acceptance of proactive system behaviour increased. Contrary, personal attachment towards reactive behaviour decreased during the experiment. The construct of personal attachment to the system used in this study is comprised of: liking, i.e meaning that the user finds using the system agreeable and it suits their taste, as well as loving, i.e. that the user has a strong preference for the system, is partial to using it and has an attachment to it (Madsen and Gregor, 2000). Combining both observations, it seemed that the proactive system was first evaluated according to its cognitive capabilities and more relationship contributing factors of proactivity may become more relevant in longer interactions. Instead, reactive behaviour seemed to get less likable after several trials. This could be a sign that reactive behaviour seems to be less engaging than proactive behaviour, which was also discussed in the first exploratory experiment. Further, we found that a personality trait correlated with the perception of proactive behaviour.

Users with a high degree of neuroticism rated the interaction with the proactive system as more understandable and were more personally attached to it. High neuroticism relates to characteristics such as low-self esteem and a high degree of uncertainty. Proactive behaviour, on the other hand, can be characterised as goal-oriented and typically confident behaviour. Therefore, high neurotic users seemed to welcome the contrast between their personality and the system's behaviour. As previous work (Meurisch et al., 2020) also found correlations between personality and the perception of proactive behaviour, a user's personality may provide clues about the impact of proactive dialogue on the HCT relationship.

Interestingly, we found that proactive dialogue affected the perception of the system regarding trust more than the empathetic strategies. It was even found that participants were personally less attached to the chatbot capable of empathetic dialogue strategies. Hence, the inclusion of small talk and empathetic responses had not the intended effect of forming a trusted bond with the user. The opposite was the case. This seemed rather strange at first sight but could be explained that the interaction dealt with very personal and sensitive content. Therefore, participants seemed to be careful to open themselves to the chatbot. Privacy issues concerning the use of chatbots are an emerging topic (Ischen et al., 2019) and were already stated to be of relevance to consider also for proactive dialogue. Another explanation could be that a kind of uncanny valley (Ciechanowski et al., 2019) was created, and the empathetic behaviour of the agent did not match the participants' expectations of a TELEGRAM chatbot. Further, empathetic behaviour could be even supposed to be not appropriate at the beginning of the interaction respectively the relationship between user and system. Hence, people could be more attached to the non-relational version, as they were more used to and comfortable with such system behaviour during first interactions. Therefore, proactivity showed a tendency to be more relevant for the trust relationship during first interactions as users get to know the system when compared to the empathetic behaviour as another conversational intelligence trait.

Limitations

As a limitation of this work, the rather low usability of the chatbot needs to be addressed. This may have occurred, due to the rigid dialogue capabilities of the chatbots that are centered around the daily check-in dialogue. Hence, people seemed to have gotten bored and annoyed by the system after three days of usage. This needed to be avoided in the studies that followed. Further, in this task domain, rather low cooperation between system and user was required. Therefore, we concluded that in such scenarios proactive behaviour only had a minor effect on trust. In more cooperative task scenarios, it was supposed that proactive system actions would have a more significant effect.

4.2.7. Conclusion

In this experiment, we studied the effects of proactive system behaviour, realised in the form of push notifications and topic switching, on the HCT relationship and sub-based of trust. We developed a chatbot prototype in the mental health domain and equipped it with proactive capabilities. Further, we included empathetic proactive dialogue strategies for comparing the effects of different conversational intelligence strategies on user trust. The results showed tendencies that proactive behaviour affected the HCT relationship positively, while the empathetic strategies had only marginal effects and even hurt building rapport. Considering sub-components of trust, we observed that proactive behaviour primarily seemed to affect the cognitive bases of trust. However, considering different user personalities, we found that proactive behaviour showed a likelihood to positively influenced affect-based trust for users with a high degree of neuroticism. Therefore, user personalities may be also considered when studying the trust effects of proactive behaviour. Regarding usability, no differences were found. Overall, the usability was rated rather low. Hence, we deem that push notifications and topic switching were not sufficiently contributing to the task execution. Therefore, for developing task-effective proactive dialogue strategies, more cooperative contexts are required to be investigated.

4.3. Summary

This chapter presented two exploratory experiments for investigating the effects of state-of-the-art proactive behaviour in CAs – recommendation and notifications as well as topic switching – on the HCT relationship and usability for identifying relevant user and system requirements.

Overall, it was identified that there seems to be generally no “no one size fits all” solution for proactive behaviour in cooperative DS. Both studies revealed that proactive behaviour realised in the form of recommendations, notifications, and topic switching may not necessarily be favoured over reactive system actions. However, observing the results of our first experiments led to the conclusion that defining different levels of proactive dialogue may be a considerable solution. Here, both recommendation strategies showed tendencies to increase a system’s user satisfaction, engagement, likeability, acceptance, and cognitive demand. This may be beneficial for enhancing the usability of CAs that could result in better task effectiveness, one of the constituents of successful cooperation. Regarding the other constituent of successful cooperation – HCT relationship – users only tended to trust proactive recommendations when they had more control over the system behaviour. Therefore, structured proactive dialogue action types seem to be necessary. These may be defined based on the LoA which have been commonly used in related work.

The second experiment investigating the effects of push notifications and topic switching strategies resulted in no clear implications on the system’s trustworthiness and usability. Contrary to the first experiment, however, the system’s proactivity was not as task-related and more considered from the view of conversational intelligence. Thus, it seems to be required for the design of adequate proactive dialogue strategies to enable the system to have more influence on the task itself. In doing so, measurable effects of proactive dialogue on the cooperation should be achieved. An interesting finding of the second experiment was that users tended to evaluate the proactive assistance system first according to its cognitive capabilities. Contrarily, the influence of proactive behaviour on the empathetic relationship between user and system seemed to play a subordinate role at the beginning of the interaction and may become relevant when considering long-term human-system relationships. As a result, we concluded that sub-components of trust, namely the cognitive trust bases competence, reliability, and understandability, need to be investigated during cooperation. Consequently, we hypothesised that cooperation can be improved by realising trustworthy proactive dialogue that illustrates a system's competency, reliability, and understandability.

For the technical realisation, the experiments revealed the necessity to investigate trigger mechanisms of proactive dialogue. During both experiments, proactive behaviour was triggered according to pre-defined rules and remained constant during the interaction. Further, they were not dependent on the current situation or specific user characteristics. However, the results of our second experiment showed that the perception of proactive respective reactive behaviour tended to differ depending on the state of the interaction as well as the user's personality trait neuroticism. We deem this to be one significant factor why the experiment did not provide clear results regarding trust and usability. Contrary, the first experiment showed that triggering proactive recommendations at the beginning of the interaction seemed to be more effective. Thus, it is not enough to implement just some proactive behaviour, but the right kind of proactivity. Consequently, the development and implementation of proactive DM methods are required. For this, different approaches to DM need to be evaluated for their applicability. A user-centred approach may be useful to consider the experimental results and related work showed that proactive behaviour is perceived differently depending on the situation and the user. However, knowledge of when to trigger which kind of proactive behaviour is dependent on these factors is sparse and mostly non-existent.

Therefore, one goal of this work is to elucidate relations between proactive dialogue behaviour and situational and user features for modelling adequate proactive dialogues in order to improve cooperation. To integrate this knowledge during dialogue, however, the cognitive capabilities of a conventional DS need to be enhanced. Thus, for modelling adequate proactive dialogue it needs to be discussed which cognitive features should be added and how a proactive DS can be integrated in such a cognitive architecture. Based on these considerations, we developed a novel user-centered proactive dialogue model which is explained in detail in the next chapter.

5. Development of a User-Centred Proactive Dialogue Model

For enabling a trusted and task-effective cooperation with CAs, a proactive dialogue model that leads to a trustworthy interaction with high usability is pivotal. To model adequate proactive dialogue fulfilling these requirements, several challenges need to be addressed.

Firstly, proactive dialogue needs to be examined from a social, user-centred perspective. The results of the preliminary experiments showed that embedding state-of-the-art proactive behaviour in a dialogue context influences different facets of the user's perceived trust in and usability of a DS. However, it remains unclear which kind of proactive dialogue needs to be triggered in which context and for which kind of user in order to foster trust and usability. A first step towards solving these problems is to review user expectations of proactive dialogue and the impact of proactive behaviour on the user dependent on user-specific and situational characteristics. Therefore, we use the results of the experimental studies and related work in order to distill user requirements towards a proactive DS.

Another challenge of modelling adequate proactive dialogue is the concrete technical realisation and a system's behavioural aspects. An important conclusion of the experimental studies was the necessity to structure proactive dialogue into well-defined action types representing different levels of proactivity. We further identified the necessity of cognitive capabilities and a proactive DM module to be relevant for realising proactive dialogue that enables successful cooperation. Driven by these observations, we review related work and background of DM for formulating behavioural and functional requirements of proactive DS.

Based on the outcomes of the requirement analysis, we provide a taxonomy of proactive dialogue. This taxonomy comprises a definition of assistance behaviour during co-operation, that the proactive CAs presented in this thesis should encompass. Further, we specify proactive dialogue act types that model different LoA. Finally, for enabling proactive dialogue in an assistance context, a cognitive system architecture is explained. This three-layered architecture makes use of various cognitive processes, including natural language processing, planning, reasoning, and decision-making, for realising proactive dialogue behaviour.

5.1. Requirement Analysis for Proactive Dialogue

The requirements for developing accepted and trusted proactive dialogue strategies need to be investigated from two points of view: that of the system and that of the user. For rendering a sound HCI, it is necessary to consider users and their characteristics, preferences, and expectations. This helps to understand how users want to interact with proactive systems and facilitates the interaction design choices for proactive dialogue. Simultaneously, it is required to consider what behavioural and functional abilities a technical system must demonstrate in order to enable proactive dialogue. This allows to point out technical components and features that are relevant for generating proactive behaviour. Further, this helps to mark technical limitations and to identify implementation challenges. Combining these two viewpoints fosters the development of a proactive dialogue model that takes into account the individuality as well as the expectations of users while also providing a valid technical foundation. Both system and user requirements are presented in the following.

5.1.1. User Requirements

User requirements are synthesised from the expectations of users regarding proactive systems and the impact of proactive behaviour on the user. Generally, users are open to technical systems expressing proactive behaviour (Meurisch et al., 2020; Grover et al., 2020; Zhang et al., 2015). Here, it is primarily expected that systems are able to provide a medium-level of proactivity, i.e. using predictive models for generating recommendations (e.g., see Hoffman and Breazeal (2007)). In our experimental studies, we also found that the implemented medium-levels of behaviour (notification, recommendations) lead to high usability (see Section 4.1) and positively influenced cognitive-based trust (Section 4.2), which indicated that the systems acted in accordance with user expectations.

However, Meurisch et al. (2020) also pointed out that these expectations are dependent on the application area and user characteristics. In most areas users want to stay in control. Thus, they initially expect a low level of system proactivity, which may be increased as users get more familiar with the system, e.g. see Glass et al. (2008). A trend towards such expectations was also visible in the preliminary experiment described in Section 4.2. Here, users tended to become less attached to the system that only stayed reactive as they familiarised themselves with the system. Contrary, proactive behaviour showed a tendency to become more understandable and accepted with increasing user familiarisation. As familiarity may be correlated to some degree to trust, we deem that higher trust might result in the user allowing the system to be more proactive. Thus, proactivity may be required to be introduced cautiously to users, i.e. by starting the interaction using a lower degree of proactive dialogue.

In this thesis, also task domains were considered in which rather a low level of proactivity was to be expected at the start of the interactions. We mostly focused on tasks concerning activity assistance, either digital, e.g. planning tasks, or physical, e.g. household tidying or do-it-yourself (DIY) tasks. Especially, for physical activity support, users seem to expect more reactive assistance at first.

Thus, we hypothesise that users generally trust lower-level proactive behaviour more in these domains than a higher degree of proactivity.

Further, the expectations of the proactive abilities of a system are supposed to vary dependent on a user's socio-demographic characteristics and personality traits (Meurisch et al., 2020). For this reason, it is hard to distill relevant requirements for our specific use cases. However, some general presumptions can be made. For example, independent of the task domain users expect proactive systems to be intelligible regarding their decision-making and to meet privacy needs. Particularly, when sensitive user data is used for taking anticipatory actions. This requirement was also visible in the experimental study presented in Section 4.1. There, proactive actions triggered using implicitly collected data were trusted less, as it was unclear to the users which data the system used.

Regarding the impact of proactive behaviour on the user for requirement analysis, related work is quite sparse and there exist only limited works on the effects of proactive dialogue. Thus, standard interaction guidelines may be considered for transferring proactive behaviour into the cooperative dialogue domain, e.g. the cooperative principle (Grice, 1975), guidelines for human-AI interaction (Lieberman, 2009; Amershi et al., 2019) or principles of mixed-initiative interaction design (Horvitz, 1999). Most relevant for our approach are the principles for proactive behaviour (Yorke-Smith et al., 2012) that are based on the mixed-initiative principles: A proactive agent should be *valuable* to the user as it advances his or her interests and tasks. It should be aware of the current situation (*pertinent*) and act according to its abilities and knowledge (*competent*). Moreover, a user should be in control of the assistant and be able to understand its actions (*controllable* and *transparent*). The agent should act unimposing and not interfere with the user's own activities and attention (*deferent* and *unobtrusive*). For adding value to the user, a proactive assistant needs to be aware of current and future needs and opportunities (*anticipatory*) and act in a *safe* way, minimising risks. For implementing proactive dialogue behaviour in realistic systems, these user requirements must be accordingly transformed into appropriate system requirements. Therefore, the system requirements are explained in the following.

5.1.2. System Requirements

System requirements may be split into behavioural, i.e. how should a system express proactive behaviour taking into account the user needs, and functional requirements, i.e. which components or modules are necessary for achieving the desired behaviour. The behavioural requirements largely consider the questions of when, how, and if proactive behaviour should be triggered (Nothdurft et al., 2015b).

If proactive dialogue is necessary depends on the user, the current situation, and the application area. This is mostly related to the user expectations of proactive behaviour as described in the previous section. Further, Nothdurft et al. (2015b) proposed to take into account three factors for deciding whether to become proactive. Here, it was distinguished between the *importance* of proactive actions for the success of the dialogue, the *context* of proactive behaviour, and the *classification accuracy* for the cause of proactive dialogue initiation. The importance of proactive behaviour needs to consider the user's short- and long-term goals and can be framed into a theory of user desires according to Yorke-Smith et al. (2012). This theory allows assessing the expected return of each proactive action with regard to the user's goals. Depending on the effects of proactive action on the safety, utility, and timeliness with respect to user objectives, the importance of proactive behaviour can be determined. For including this principle into a dialogue model, the user's task success is required to be integrated into the dialogue state. For the work described in this thesis, this was realised by including a success score of cooperation into the proactive dialogue model.

Regarding context, it can be differentiated between application- or task-focused proactivity and utility-focused proactivity (Yorke-Smith et al., 2012). Task-focused proactivity aims at providing assistance for specific, well-defined tasks, while utility-focused proactivity is more related to abstract user goals with regard to their interests, e.g. with the aim to reduce workload or to enhance the user's perceived trust in the system. For improving cooperation, proactive dialogue must represent both types. Therefore, this work aims at designing and implementing strategies that enhance utility-focused proactivity (trust) and task-focused proactivity (usability).

Classification accuracy represents a system's confidence measure for the recognition of causes for inducing proactive behaviour, e.g. classification probability of certain user states such as the user's emotions. Therefore, proactive dialogue may be modelled using probabilistic approaches rather than rule-based approaches.

In the following, how proactive behaviour can be expressed is elucidated. There exist several ways for a CA to take proactive actions. Foremost, a requirement of a proactive system is to be able to take actions on several LoA as described in Chapter 4.2. This allows a CA to act on various degrees of proactivity and change their style of interruptions, e.g. be less direct with notifications or be more direct using suggestions. In this regard, Yorke-Smith et al. (2012) proposed a taxonomy of four possible proactive actions specifically designed for a task management task. A system can either act directly, e.g. perform the next step or steps of a shared task, act indirectly, e.g. suggest a user task be delegated to a teammate, collect information, e.g. gather, summarise information relevant to a user, or shared task, or use reminders or requests, e.g. remind of the user's next step in a shared task, ask for feedback or guidance from the user. Further, requirements for user-driven adjustable autonomy may be transferred to proactive dialogue (Maheswaran et al., 2003). These requirements address a system's capability to perform a specific task and to personalise its proactive behaviour. For this, a system needs permission requirements. These define conditions that indicate for which actions a system must obtain authorisation from the user. In addition, consultation requirements address decisions that should be handed to users. Maheswaran et al. (2003) proposed to implement such reasoning using a MDP. Therefore, this work also considered modeling proactive dialogue as an MDP for reflecting a more user-centred approach of DM.

Moreover, proactive behaviour may lead to incomprehensible situations for users because of a mismatch of reality and mental model of system behaviour (Nothdurft et al., 2015b). This requires making use of explanations for justifying system behaviour or rendering its actions more transparent. For example, Grover et al. (2020) believed that providing explanations would have a positive outcome for the cooperative decision-making process. Further, Nothdurft et al. (2013) showed that by providing explanations, proactive behaviour can be explained better which in turn fosters trust in the system. Therefore, we made use of explanations for rendering proactive dialogue more comprehensible, thus eliminating the lack of a mental model as a confounding variable.

The final behavioural requirement of proactive system actions concerns the timing of activities. The timing of proactive actions should adhere to the user requirements of proactive behaviour to be deferent and unobtrusive. This is mostly related to turn-taking problems. These problems can be either considered more from a conversational or from a task-oriented point of view. Turn-taking in spoken conversations is a long-studied research aspect. A comprehensive review can be found for example by Skantze (2021). With regard to this thesis, turn-taking cues that a CA leverages for taking the floor during dialogue are of interest. Here, a CA is required to be able to detect appropriate moments for taking the initiative. For this, a system may use multi-model cues, such as verbal including syntax semantic and pragmatics, prosody (intonation, intensity, duration), as well as breathing, gaze, and gestures. While conversational turn-taking cues not only address the timing of proactive actions but the timing of system utterances in a realistic fashion in general, task-oriented turn-taking cues address well-defined points of dialogue initiation. Here, a system may take the floor either at the task or sub-task level at pre-defined timing thresholds.

Functional requirements consider the technical components and modules that enable a computer to engage in a proactive dialogue. For identifying these requirements, it is useful to consider the requirements for technical assistance systems as, for example, described in Biundo and Wendemuth (2016) or Honold et al. (2014). In the following, we will make references to both works. A first requirement is to have extensive knowledge about the application domain in order to be able to provide assistance at all. This may even require expert knowledge of specific task domains. For example, the provision of adequate assistance in the DIY or home improvement domain requires the system to have conceptual knowledge about the tools and materials and their characteristics that are necessary to solve the task. This knowledge is typically represented in the form of a knowledge base (ontology) that allows ontological reasoning about tasks and required material for task execution. For example, Schiller et al. (2017a) described an ontology for the representation of knowledge for conducting DIY-tasks. Similarly, the prototype implemented for the preliminary experiment described in Section 4.1 made use of a knowledge base for having knowledge about different restaurants and food types which can be used for recommendations. Therefore, knowledge bases are essential for proactive DS for ensuring adequate task-level proactivity.

Further, it is necessary to have procedural task knowledge, i.e. knowledge about the task steps that are necessary to achieve the overall task. For this, a problem-solving component is required. In many cases, a planning module can be used for this task (e.g. see Behnke et al. (2019c)). However, also script-based logic approaches can be applied, e.g. see Miehle et al. (2021). This knowledge can then be used to model the current task state which may be relevant for deciding on proactive system behaviour.

Knowledge and problem-solving components form the foundation for enabling proactive dialogue behaviour in an assistance context. At the core, however, a dialogue component decides how to leverage the information provided by these two modules for conducting proactive dialogue behaviour. For this, the dialogue component makes decisions about how to integrate the user in the planning process and the kind of adequate proactive dialogue behaviour dependent on the user and task context. The interplay between AI planning components and dialogue systems was described, for example, by Nothdurft et al. (2015a). Further, plan-based dialogue systems, e.g. see Cohen and Perrault (1979); Allen et al. (1995), describe how to use planning for cooperative dialogue. For connecting a proactive DM module with such cognitive modules, we hence relied on these previous works as a template in the scope of this thesis.

To personalise proactive dialogue to specific users and their contexts, our preliminary experiments revealed that relying on user models is fundamental. Such user models may use information gathered from a system's sensor, e.g. via speech, video, or physiological sensors, in order to trigger adequate proactive behaviour. In Section 3.2, we have already described several methods on how to include the user for adapting the dialogue. For our approach, we adopted rule-based as well as stochastic methods as trigger mechanisms for enabling user-centred proactive dialogue.

5.1.3. Conclusion

In this section, we presented system and user requirements for integrating proactive behaviour in DS. The observations of user requirements were centred around what expectations users have regarding the level and timing of proactive system actions. Here, we concluded that these expectations are rather unclear due to the varied dependencies on specific user characteristics, the context, and even the particular domain. Therefore, AI interaction guidelines could serve as a foundation for modelling adequate proactive dialogue. Based on these guidelines, the implications of different levels of proactive dialogue on the user and the task success can be investigated for gaining further knowledge on trustworthy and task-effective proactive dialogue design.

Further, we identified several system requirements, both behavioural as well as functional. Behavioural requirements comprised included a system's ability to reason about the importance, the context, and the accuracy of proactive behaviour. Here, we concluded to include a quality measure for deciding on the importance of proactive dialogue, and rely on mechanisms to provide both utility- and task-focused proactive dialogue for increasing the quality of cooperation with CAs. Further, probabilistic methods were identified to be advantageous over rule-based methods for representing a proactive dialogue model's accuracy. In addition, we concluded that the primary requirement on how to express proactive behaviour is to take actions on different LoA. Therefore, we identified the need for a valid decision-making function for selecting an appropriate level of proactive dialogue. Other requirements were the inclusion of explanations for preserving the user's comprehension of the system's proactive actions as well as the usage of well-defined timing strategies for triggering proactive dialogue. Finally, for identifying the functional requirements of a proactive DS, we relied on the characterisation of technical assistance systems. Such systems contain components, like planning, reasoning, knowledge management, and dialogue that allow for implementing proactive dialogue. Here, also personalisation approaches were identified for developing user-centred approaches.

5.2. Taxonomy of Proactive Dialogue

Based on the system and user requirements for CAs to express proactive dialogue behaviour, we developed a taxonomy of proactive dialogue. As we studied proactivity primary in the context of assistance systems, we defined proactivity as the initiation of helpful sub-dialogues during task execution. In this thesis, we focused on a mixed-initiative interaction where a user cooperated with a CA in decision-making and problem-solving tasks. Therefore, we observed proactive dialogue at well-defined points during task execution instead of considering a more turn-taking-oriented approach. In this regard, we introduced a formalisation of proactive dialogue at the task level. Further, we introduced proactive dialogue action types according to different levels of system autonomy and with respect to the guidelines for appropriate proactive behaviour as described in the previous section. First, task-level assistance is explained.

5.2.1. Proactive Assistance on Task-Level

In mixed-initiative user interactions, a user and an autonomous agent, that is able to take actions independently, collaborate for solving tasks (Horvitz, 1999). In the scope of this thesis, these tasks were either decision-making tasks, where the system cooperated with the user in order to select appropriate solutions from a sub-set of possible solutions, or problem-solving tasks, in which a system could take actions in order to facilitate the user's execution of an arbitrary task at hand. Generally, both task types followed a specific task structure. According to the online APA Dictionary of Psychology, task structure can be defined as the "the extent to which there is a clear relationship of means to ends in the performance of a task. In a highly structured task, the procedures required to perform the task successfully are known, whereas, in an unstructured task, there is uncertainty about how to proceed" [1].

For providing task assistance, an autonomous agent needs to track the user's activities and goals while reasoning about the costs and benefits of taking automated actions. Here, proactive dialogue serves for communicating and negotiating a system's decision process for minimizing the risk of system failure and enhance comprehensibility. By this, the agent complies with the requirements described earlier. Typically, the structure of a task can be described as a sequence of task steps s, where at each step a specific problem has to be solved or a decision has to be made. Theoretically, a task can be hierarchically divided into multiple sub-task levels (see Fig. 5.1). For simplification, we considered proactivity always on the most primitive sub-task level. Therefore, most tasks that were used as exemplifying scenarios were structured on two levels: overall abstract task and task step level. Depending on the nature of the task, humans can take different actions.

[1] https://dictionary.apa.org/task-structure

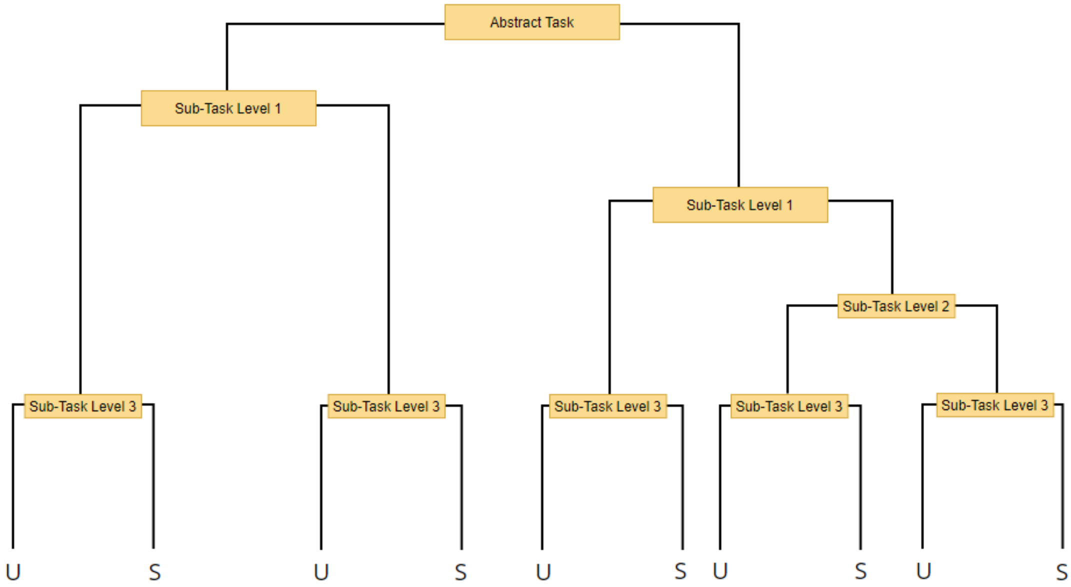


Figure 5.1.: Visualisation of a hierarchical task structure. An abstract task may be decomposed iteratively into several sub-tasks.

For example, in a decision-making scenario, there exist several decision options for each task step, represented as a decision option o_i. Here, humans sequentially have to make decisions d_i until the cooperation process is ended after n task steps. Thus, working on decision-making task can be formulated as follows:

$$(o_1, d_1), (o_2, d_2), (o_3, d_3), ..., (o_n, d_n) \tag{5.1}$$

The formulation of a problem-solving scenario may be described analogously. Considering the task flow, a human is required to decide regarding a specific task, which leads to another task step, where a new decision is required to be made. In a scenario where a human is supported by a CA, both human and computer are able to take actions during the individual task steps. Here, the assistance is provided either in a proactive or a reactive manner. Being reactive, an assistant only helps on explicit requests by a user of the assistance system. In contrast, proactive behaviour implies that the CA suggests or takes over actions on behalf of the user. Therefore, proactive actions can be considered as the initiation of sub-dialogues, where the assistant influences a user's action. Subsequently, a proactive action in a decision-making scenario can be defined as a function of d, noted as $pa(d)$. Under this consideration, the structure of a decision-making task can be updated as follows:

$$(o_1, pa(d), d_1), ..., (o_n, pa(d), d_n) \tag{5.2}$$

Again, the structure for problem-solving tasks is analogous. Based on this task-level taxonomy for providing proactive assistance, the next step was to define proactive dialogue

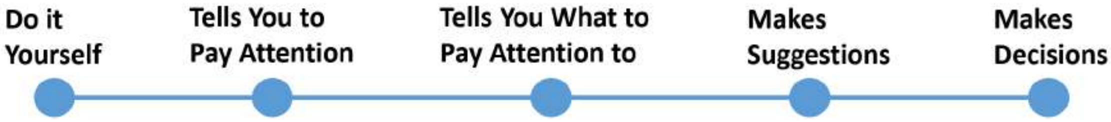

Do it Yourself
Tells You to Pay Attention
Tells You What to Pay Attention to
Makes Suggestions
Makes Decisions

Figure 5.2.: The IP-continuum ranging from zero to full autonomy.

act types. In doing so, different kinds of behaviour could be modelled. The development of these dialogue actions is described in the following.

5.2.2. Definition of Proactive Dialogue Action Types

The proactive dialogue act types were designed following the principles of the IP-continuum developed by Isbell and Pierce (2005). As previously described the continuum describes the way in which a proactive assistant can cooperate in tasks. The authors differentiated between five different levels of proactive assistant behaviour. The IP continuum ranges from zero, i.e., the user acts fully on his own, to to full automation, i.e., the assistant acts fully on behalf of the user. The nuances between these two extremes are alerts, telling the user to pay attention, notifications, telling the user exactly what to pay attention to, and suggestions, providing the user with several decision options. The more proactive a system becomes, the more it takes off control and responsibilities from the user. Hence, the risk of failure also increases, as the possibility that the system might take actions ineptly towards the user's goal without asking for confirmation expands. This may possibly hurt the human-computer relationship (Isbell and Pierce, 2005). Transferring the continuum to application in human-computer dialogue we summarised the second and third points of the continuum (see Fig. 5.2) under the proactive action *Notification*. The proactive actions content was modelled according to our requirement analysis throughout this thesis, e.g. only task-specific information was conveyed that contributed to the user's interests and tasks, or the system was aware of the current situation during task execution and aware of the user's current and future needs. This fulfilled the user requirements of pertinence, competence, and anticipatory behaviour. Further, all prototypes described in this thesis were modelled to be expert systems. Thus, proactive actions were created to minimise risks for the user in a safe manner.

In addition, proactive explanations were added to justify the behaviour of the system to take the initiative. This fulfilled the user requirements of transparency. Besides, justification explanations showed to improve the user's trust in automatic systems (Nothdurft et al., 2014). Therefore, these explanations also ensured the comprehensibility requirement. Based on these considerations, we obtained four levels of proactivity, that were transformed into distinct proactive dialogue act types:

None: This dialogue act type refers to reactive system behaviour and is the lowest level of proactive behaviour. In this condition, users can only explicitly request help from the assistant.

Notification: This dialogue act type represents the most conservative proactive approach. Using this dialogue act, the participant is only notified by the system. In this case, it was up to the user to get assistance or to ignore the system's offer. By applying a notification, the user is in control of the system's proactivity and is able to ignore it. However, this proactive action might shift the user's focus to possible helpful resources and might be perceived as unobtrusive.

Suggestion: Using the aforementioned proactive dialogue act type, the agent directly suggests a solution by also providing a proactive explanation for its decision. Hence, the system takes over some control of the interaction and asks the user to make a choice. This represents a more rigid way of user interruption, but still lets the user in control over the final decision. In a response to the system's proposal, a subject can either confirm or decline the suggestion.

Intervention: In this case the system takes over all responsibilities and performs a particular action in place of the user, also providing a proactive explanation. Utilising this proactive dialogue act type might be perceived as quite obtrusive, but can be helpful if the user has reached a critical level of need for proactivity.

Depending on the use case, these proactive dialogue act types formed the basis for developing proactive dialogue strategies. How the proactive dialogue act types were realised in detail for each prototype, is explained in the sections covering the respective systems.

5.2.3. Conclusion

This section dealt with the definition of a taxonomy of proactive dialogue in CA. Here, we considered proactive behaviour as the ability to proactively initiate sub-dialogues during mixed-initiative interaction between system and users which cooperate on a task-level basis for decision-making and problem-solving. Therefore, we focused on decision-making and problem-solving scenarios for studying our research questions throughout this thesis.

Finally, we defined several proactive dialogue act types that represent different degrees of proactive behaviour according to the IP-continuum for integration into the dialogue domain. Here, we also stressed the importance of explanations for rendering proactive dialogue comprehensible. These dialogue act types were then implemented in various proactive dialogue strategies for providing assistance in decision-making and problem-solving scenarios.

For technically realising proactive dialogue behaviour, we conceptualised a cognitive architecture based on the described system requirements. The individual constituents of the architecture are described in the following.

5.3. Design of a Cognitive Architecture for Proactive Dialogue

For equipping a CA with proactive dialogue behaviour, we identified the system requirement of possessing several cognitive capabilities. Cognitive capabilities may be described

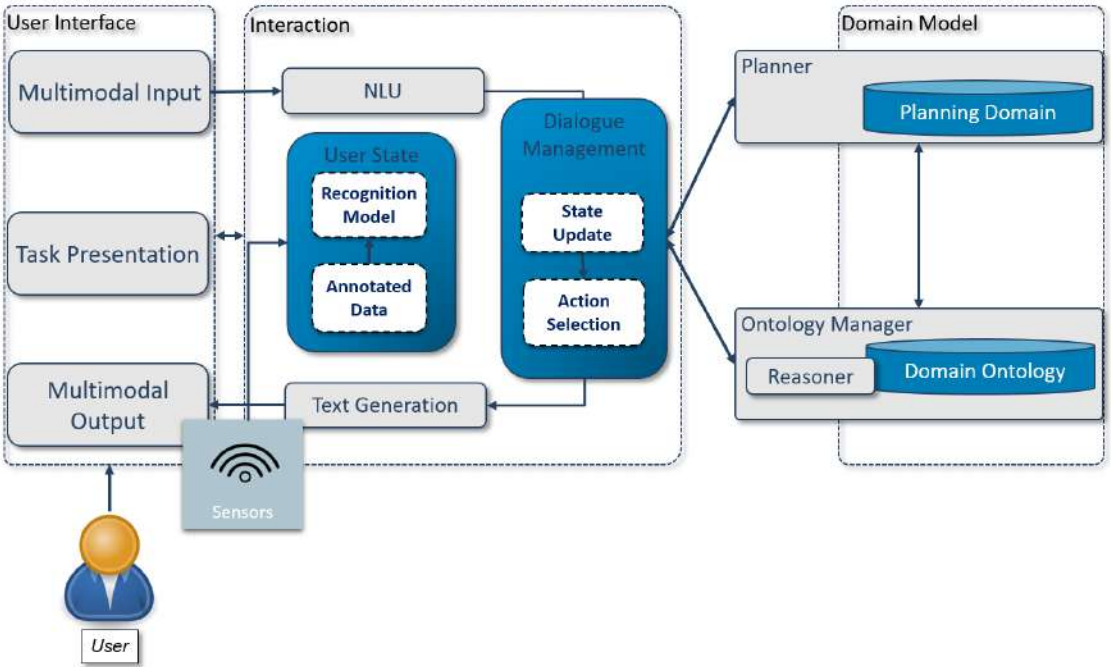


Figure 5.3.: The cognitive architecture that forms the basis for each prototype developed in the scope of this thesis.

as "mental processes involved in the acquisition of knowledge, manipulation of information, and reasoning"(Kiely, 2014). Specific cognitive functions are perception, memory, learning, attention, decision making, and language abilities (Kiely, 2014). For including such capabilities into a technical system, we created a model architecture comprising three major constituents. *Interface* covers the system's perception abilities and presents a system's output using various modalities. *Interaction* describes the system's language abilities, while also containing elements that help the system to learn from a situation and to maintain attention. Finally, *Domain Model* represents a system's memory and decision-making functionalities. The three components are described in the following with a focus on their application in the prototypes presented within the scope of this thesis. The overall cognitive architecture is depicted in Fig. 5.3.

5.3.1. Interface Design

As the front-end of proactive CAs, we used web-based multimodal interfaces. For the experimental prototypes that were developed as virtual CAs, the interface contained a graphical user interface (GUI) representing the cooperation task at hand. All virtual assistants were basically experts in the task domain and had full knowledge about individual task steps and properties. Therefore, the virtual CAs were able to manipulate the GUI as a response to user requests or via proactive actions. Depending on the task type, the task was represented as a set of instructions on how to solve a physical task (see Section 6.3) or as a set of available options for solving decision-making tasks (see Section 6.1, 6.2, 7.2). Further, we implemented a robotic CA prototype, which could directly manipulate the physical world (see Section 6.4). Therefore, a virtual representation of the task was obsolete, and the GUI only served for visualisation of the dialogue between the human and the robot. The reason why the interfaces were developed as web-based applications were to allow the integration of cloud-based language and perception components. In doing so, we could integrate state-of-the-art speech recognition and synthesis modules for all prototypes. Further, this allowed to include sensors for perceiving the user's cognitive-affective state (see Section 6.2), user activities (see Section 6.3) or the user's environment (see Section 6.4). Besides interacting via spoken language, we also allowed the user to interact with virtual and robotic CAs using textual (see Section 6.3 and 6.4) as well as pre-defined answer options (see Section 7.1, 7.2). The interaction modules for a proactive CA are elucidated in the following.

5.3.2. Interaction Design

For extracting the meaning of spoken or written user utterances, we employed NLU modules with varying degrees of complexity. For the prototypes applied in more restricted lab environment studies (see Section 6.1 and 6.2), the usage of simple grammar-based language understanding was sufficient. Here, users only required a limited vocabulary for interacting with the CAs. Contrary, for the more sophisticated prototypes applied in more realistic settings (see Section 6.3 and 6.4), we applied state-of-the-art statistical-based NLU methods by employing RASA's NLU or Microsoft's language understanding intelligent service (LUIS) framework. Note that the prototypes utilising pre-defined answer options did not require any NLU component.

A key part of this thesis, was to develop DM modules that could handle both reactive and proactive dialogue. For this, we applied several DM approaches. Regarding the experiments exploring the effects of proactive dialogue strategies based on the previously described proactive dialogue act types, we implemented simple rule-based DMs (see Section 6.1 and 6.2). For the sophisticated prototypes (see Section 6.3 and 6.4), we made use of agent-based DMs. This was due to wide variety of user input and the ability of the system to take into account environmental and user states for selecting an appropriate dialogue action. For the robotic CA, we extended a RASA-based DM to handle proactive dialogues based on specific events. For implementing a user-centred proactive DS (see Section 7.4), either rule-based or RL-based DM approaches were used. For application of RL-based DM, the dialogue was modelled as an MDP.

All prototypes developed in the scope of this thesis could trigger proactive dialogue dependent on a user state. How these user states were modelled also depended on the complexity and purpose of the application. For the prototype described in Section 6.1, we used a simplistic time measurement approach for estimating the user's state of insecurity. In Section 6.2, the prototype made use of pre-trained models for assuming the user's cognitive-affective state based on facial features that were captured using video input. The prototype described in Section 6.3 applied both pre-trained statistical as well as rule-based models to track the user's current activity state. The robotic prototype (see Section 6.4) modelled the user's context rather than a particular user state. This allowed the robotic CA to act upon specific events. In Section 7.2, we describe the development of a novel model for predicting the user's perceived trust in CAs. Therefore, a dataset was collected and annotated with static as well as dynamic user-, context-, and dialogue-based features. Further, sophisticated statistical methods were applied in order to detect the user's current trust state. This was then used to include a trust measure in the dialogue state for modelling trust-adaptive proactive dialogue.

For producing textual output, all developed prototypes made use of template-based approaches. An exception was the virtual CA described in Section 6.3 which could additionally generate text in a dynamic fashion using ontology verbalisation. The verbaliser was developed by Schiller et al. (2018).

5.3.3. Domain Modelling

The domain models of each prototype contained task-specific information. This included factual as well as procedural task knowledge allowing the virtual and robotic CAs to maintain expert task knowledge. For modelling factual knowledge, we used well-defined knowledge structures. For the prototypes deployed in restricted lab environments (see Sections 6.1 and 6.2, as well as Chapter 7), factual knowledge was represented using javascript object notation (JSON)-based structures. These knowledge representations were modelled rather simplistic as sophisticated reasoning mechanisms were not necessary for these limited study scenarios.

In contrast, the prototypes applied in advanced application scenarios (see Sections 6.3 and 6.4) made use of the W3C web ontology language (OWL) for formally describing and reasoning about complex knowledge. However, these ontologies were not the main objective of this thesis. Therefore, models developed by Schiller et al. (2017a) and Prasad and Ertel (2020) were used. Procedural knowledge was modelled based on hierarchical planning methods, e.g. see Behnke et al. (2018a,b). However, as this was not the main focus of this thesis, we simulated planning for the prototypes used in the lab environment studies. Simulating task plans implied that the order and the content of sequential task steps were pre-defined in advance and were represented using the JSON-format. This also allowed us to model simple relations between individual task steps. The relations were used to measure the task success of the users using numerical scoring models (see Chapter 7).

Solely, for the experiments conducted in the more realistic scenarios, we used full-fledged AI-powered planning algorithms. The robotic CA described in Section 6.4 relied on FlexBE framework (FlexBE, 2018; Schillinger et al., 2016), which utilised a concurrent state machine for realising the robot's action planning. The virtual CA that is explained in Section 6.3 utilised a sophisticated hierarchical task network (HTN) planning approach. This approach was based on a coupled knowledge-based for planning and ontological reasoning (Schiller et al., 2017a). Here, the planning domain comprised atomic actions for executing tasks in specific application domains, e.g. the DIY-domain in our case. These actions contained preconditions under which they may be executed as well as effects on the environment after their execution. The planner, in our case a module called PANDA developed by Behnke et al. (2018b), was able to generate a set of instructions in order to execute a specific task physically. This information was then used by the prototypes DM module for handling task-specific information.

5.3.4. Conclusion

In this section, we presented the design of a cognitive architecture for integrating proactive dialogue into CAs. For designing the interface between the user and proactive system, we utilised web-based multimodal interfaces. These were built in a modular structure for allowing a facilitated use of various services, including self-developed GUIs as well as off-the-shelf speech and other sensory modules. Regarding interaction design, we also utilised a modular architecture for the implementation of proactive dialogue. Here, we made use of various methods for NLU and text generation and proposed various methods for proactive DM. For rendering the interaction user-adaptive, we also included a module for user state recognition. Finally, the cognitive architecture comprised a domain model that contained task-specific information. For domain modelling, we applied simple as well as more advanced third-party approaches for planning and reasoning. These two cognitive capabilities were identified to be inevitable for rendering proactive dialogue in realistic use case scenarios with full-working prototypes.

5.4. Summary

This chapter presented a theoretical proactive dialogue model that formed the foundation of the CA prototypes, proactive dialogue strategies, and experiments described in this thesis.

For forming the proactive dialogue model, we first summarised user and system requirements regarding proactivity in assistance contexts. Examining the user requirements, we found that in task domains that were considered in this work, user expectations rather encompass low-level types of proactivity. Particularly, reactive behaviour seemed to be a user expectation at the beginning of interactions with unfamiliar systems. Therefore, we assumed that reactive behaviour may generally lead to high user-perceived trust ratings in our following experiments. Also, a medium-level of proactivity ought to generally receive higher trust ratings than high levels of proactive dialogue. However, as the expectations may differ depending on the individual user characteristics, we deemed it essential to consider the effects of different levels of proactive dialogue behaviour taking into account user specifics and context.

For designing proactive behaviour, we relied on principles and general guidelines that should foster good interaction design. As some of these principles comprise elements that are known to be related HCT, e.g. competence, control, and transparency, we assumed that adhering to these rules for designing proactive strategies may foster user trust in proactive CAs. We, therefore, hypothesised that adjusting the right level of proactive dialogue at the fitting moment with respect to the specific user type may increase a CA's trustworthiness with regard to perceived competence, reliability, and understandability as observed during preliminary experiments. Thus, an HCT measure seems to be essential as a metric for assessing the social quality of cooperation with a proactive dialogue DS.

Secondly, we summarised system requirements for equipping a technical system with proactive behaviour. Behavioural requirements comprise a system's ability to assess the expected return of each proactive behaviour with regard to the user's goals and desires. Thus, a system requires a mechanism to reason about the quality of their proactive actions and to adjust their behaviour accordingly. We concluded that this may be achieved by developing a module for predicting the quality or utility of proactive dialogue and using an RL-based approach for implementing user-centered proactive dialogue. As proactive behaviour may have a task-focused and utility-focused purpose, the quality measure of proactive dialogue should encompass social (e.g. trust) and task-related (e.g. usability) metrics for achieving the research goal stated in this thesis. For this reason, proactive dialogue strategies not only need to be evaluated regarding their trustworthiness but also their usability.

For designing different levels of proactive dialogue, the LoA known from autonomy research seems beneficial. Further, we argued that the inclusion of explanations is necessary for complying with the design principles of proactive behaviour. For the timing of proactive actions, we restricted the system requirements to be able to act on a task-level basis. Finally, we identified some technical requirements that included the combination of various cognitive processes and an interplay between AI and HCI components for implementing CAs capable of proactive dialogue.

Based on these considerations, we then defined a taxonomy of proactive dialogue. Focusing on task assistance and human-machine cooperation, we defined the cooperation process as a dialogue in which individual task steps can be described as dialogue turns. Here, proactive behaviour was defined as the initiation of sub-dialogues at task-step influencing future user actions. With regard to the LoA, we defined four proactive dialogue act types reflecting different types of autonomous system behaviour. These were developed to be domain-independent and may be utilised in various contexts. Complying with the principles of proactive behaviour, these dialogue act types may be enriched with justification explanations for ensuring transparency and comprehensibility. The leveled proactive dialogue act types formed the foundation for developing proactive dialogue strategies.

Finally, we conceptualised a cognitive architecture for implementing proactive dialogue behaviour into CAs. We presented three layers of the architecture comprising components for interface and interaction design as well as for domain modelling. Further, we outlined the concrete implementations of each layer for the various CA prototypes developed in this thesis.

This chapter provided a solution for closing the research gap on how to transfer the concept of proactivity into the dialogue domain. Further, we provided a solution for the structured realisation of proactive behaviour in DS. For achieving our goal of improving the cooperation between humans and proactive CAs, the consequential next step was to develop adequate proactive dialogue strategies that enhance the trustworthiness and the usability of the system. Therefore, we designed several proactive dialogue strategies based on the described model. These were then evaluated in different user studies regarding their effects on the cooperation focusing on their impact on perceived user trust and usability. As we identified the user's expectations towards proactive behaviours to be highly dynamic, it was required to observe the proactive dialogue strategies in various user states and specific situations. The results of the evaluations may then be used to implement appropriate proactive dialogue strategies for user-centred DM in CAs. In the following, we describe the in-depth design and evaluations of proactive dialogue strategies.

6. Design of User-Centred Proactive Dialogue Strategies and their Effects on Cooperation

In this chapter, we describe the design of four user-centred proactive dialogue strategies and evaluate their effect on human-computer cooperation. Here, we measured the social impact on the cooperation in the form of perceived user trust, as well as task effectiveness measured using subjective and objective usability metrics. Regarding user trust, we focused on measurements of cognitive-based HCT, as our preliminary studies showed tendencies that proactive behaviour foremost influences a system's perceived competence, reliability, and understandability. Due to the highly dynamic nature of user expectations towards proactive behaviour dependent on specific user states and contexts, we designed and evaluated proactive dialogue strategies in four different assistance scenarios using different user- and situation-related trigger mechanisms. Here, we selected scenarios that fulfilled four important criteria for studying proactive assistance. Firstly, the scenario was required to contain decision-making or problem-solving tasks where a cooperation between user and machine was possible. Secondly, the task domain needed to possess a sufficient degree of complexity and difficulty in which assistance, either provided reactively or proactively, was beneficial for task solution. Strongly related to this point is the third criterion which demands the task complexity and difficulty to be easily adjustable for studying the impact of a user's domain and task expertise on the perception of proactive behaviour. Finally, the scenarios were required to contain mechanisms for creating situations in which the user was vulnerable towards the system's decision and trust mattered. In doing so, trust effects of different proactive system behaviour were deemed to be better observable and thus measurable.

For all scenarios, the level of proactivity remained constant throughout the interactions as we wanted to investigate the impact of the individual levels in the different scenarios.

First, we studied the effects of the proactive dialogue strategies on the cooperation depending on the task difficulty of a decision-making task. In recent works, it has been argued that different degrees of task difficulty require distinct types of assistance Qiu et al. (2020); Glas et al. (2008). Therefore, we studied the impact of task difficulty on the perception of proactive dialogue. As a trigger mechanism for proactive dialogue, we evaluated a user insecurity measure.

In the second experiment, we examined the impact of the cognitive-affective user state on the perception of proactive dialogue strategies during cooperation in a learning task. For example, Friemel et al. (2018) proposed to utilise cognitive-affective user states for triggering different types of assistance. As a trigger mechanism, we observed negative user states, i.e. confusion and frustration. For these experiments, two prototypes of virtual CAs were developed. The implementation of these systems was based on the cognitive architecture concept. However, as we conducted the user studies in restricted lab environments to investigate the effect of the different proactive dialogue act types, the cognitive abilities of planning and reasoning were simulated.

The third experiment dealt with investigating the impact of the user's activity on the cooperation with a proactive CA assisting in the user's execution of a physical task. Gaining procedural knowledge through the execution of physical tasks was one of the primary goals in the development of assistive companion systems (Biundo and Wendemuth, 2016). Therefore, we embedded proactive dialogue into a similar scenario. Here, a trigger mechanism based on the user's progress and current activity with a connected electric tool was studied.

Finally, we observed the impact of external events on the cooperation with a proactive dialogue assistant in a household assistance setting. Especially, robot-based assistance in the domestic domain has become an increasingly recognised research topic (e.g. see Pham et al. (2017) or Graf et al. (2004)). Therefore, we implemented proactive behaviour into a robot that was able to physically manipulate objects in its surroundings and execute simple household duties, e.g. tidying up or bringing tasks. As a trigger mechanism, we investigated specific contextual events.

Contrary to the first two experiments, experiments three and four examined proactive dialogue in more realistic application scenarios. For this, we equipped high-fidelity prototypes with proactive dialogue capabilities. High fidelity in this context implied that cognitive abilities, such as planning and reasoning, were not simulated anymore, but provided by sophisticated AI modules. For assisting with the execution of the DIY task, we developed a high-fidelity virtual CA. For assisting in the execution of household tasks, a robotic CA was developed. Due to the complexity of the interaction in the realistic task domains using high-fidelity prototypes, only medium-levels of proactive dialogue were investigated.

The experiments aimed to determine the selection of the appropriate level of proactive dialogue, i.e. proactive dialogue act type, dependent on the specific user type and context information for improving the human-computer cooperation by increasing a system's trustworthiness and usability. By also observing the impact of proactive dialogue in more realistic task scenarios, as per experiments three and four, we also intended to evaluate the portability and validity of our proactive dialogue model. The results of the experiments were then used to implement a user model which finally resulted in the realisation of a user-centred proactive DS. In the following, we provide an in-depth description of the developed prototypes, proactive dialogue strategies, and experiments.

6.1. Effects of Proactive Dialogue Strategies Dependent on Task Difficulty

6.1.1. Motivation

In the first experimental lab study, we investigated the impact of proactive dialogue strategies on cooperation dependent on the task difficulty. For assessing the difficulty of a task, we made use of the ICL value of the task that was explicitly rated by users during evaluation. We hypothesised that dependent on the task difficulty, different proactive levels were more trustworthy than others and show different usability effects. Further, it was tested if there exist general differences between the proactive dialogue act concepts and the HCT relationship. Also, we tested whether user insecurity or hesitation could be used as a trigger mechanism to initiate proactive behaviour. For evaluation, we implemented a prototype that assisted with decision-making by initiating proactive behaviour dependent on recognised user uncertainty. Generally, during decision-making users have to make choices using their knowledge and the available information. Here, we expected the need for proactivity if users had problems with making a profound decision. A sign of arising problematic situations could be user hesitation to select a decision option. To prevent such problematic situations, proactive behaviour may be beneficial. It was assumed that using this trigger mechanism might increase trust and enhance the system's usability. In addition, we evaluated the quality of the study setup using different quality measures. In the following, we describe the use case scenario, the development of the virtual CA prototype, and the design of the proactive dialogue strategies based on the previously described conceptualisations. Further, we provide details about the experiment and present the results. Finally, the outcomes of the experiment are discussed.

6.1.2. Scenario

For developing and evaluating proactive dialogue strategies, a use case in the DIY home improvement domain was chosen. DIY is one of the most popular hobbies around the world. According to a German poll, 11.91 million people aged 14 and older showed particular interest in the topic of home improvement (Allensbach, 2019). Thus, the domain should provide an engaging environment for study participants. However, for novices of craftsmanship and the handling of tools the entrance into the DIY-domain is complicated. For example, beginners do not have task-specific knowledge about materials and tools. Consequently, they lack the technical and practical abilities for planning and conducting DIY projects. This makes it hard for them to decide which methods and tools to use for a specific project without proper instructions. For this reason, novices need trusted and competent assistance to receive appropriate guidance and tutoring. Taking these considerations into account, this scenario was deemed to be particularly suitable for testing the effects of proactive CA behaviour on HCT and usability. Besides, the rich amount of different DIY-projects of varying complexity allowed to easily configure tasks having different degrees of difficulty. Therefore, we developed a virtual CA prototype that was augmented with the ability to express proactive behaviour. In the DIY-scenario the user

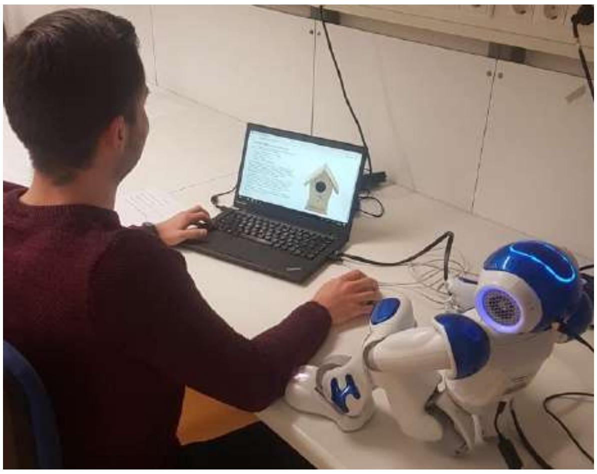
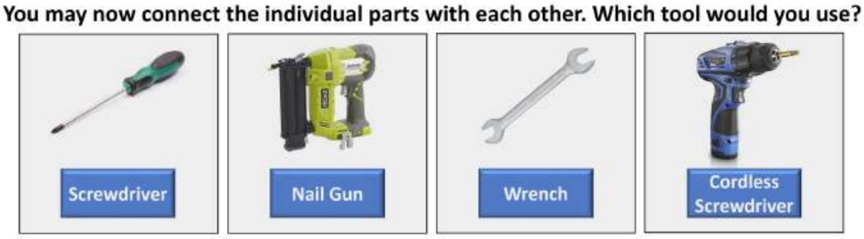


Figure 6.1.: The robotic assistant NAO was placed in front of the task screen right beside the subject (Kraus et al., 2020c). In doing so, the robot should be perceived as a team member for task completion. Additionally, NAO has been seated during the interaction for a better quality of speech recognition.

was accompanied by the CA using different degrees of proactivity. In doing so, the effects of proactive CA behaviour on the HCT and usability could be explored. The user's task in the test scenario was to plan and make decisions on two separate DIY-projects: the building of a wooden nesting box and the assembly of a wall candle holder made from copper tubes. The projects differed in their familiarity with users and might affect their perception of the difficulty of the task. While building a nesting box was ought to be more known to users, a copper-tube wall candle holder was supposed to require a higher degree of imagination from the subjects and could hence be perceived as more difficult. Each project consisted of a predefined set of five sub-tasks. The building of the wooden nesting box comprised the steps "wood cutting", "pre-drill holes", "connecting the parts of the nesting box", "creating an entrance hole", and "process wood". Contrarily, the steps for the wall candle holder were "saw copper tubes", "connect copper tubes", "polishing copper tubes", "pre-drill wall and dwell", and "attach wall candle holder". The construction of the DIY-projects and the different options were based on the extensive internet research on how to perform the projects at hand. For each task, users had to make decisions on how they would perform individual task steps. However, they did not physically work on the DIY-project. They only had to select between different pre-defined approaches or tools which could help to solve a particular task step. The possibilities on how to solve a task step were presented on a task screen implemented as GUI. The order of the task steps was fixed and could not be altered by the user. For each step, four options on how to accomplish the task were presented.

An example of possible options for the sub-task "connect the parts of the nesting box" is depicted in Fig. 6.2. Subjects were told to select the options they considered best. Further, they were informed that a virtual CA would be able to help with decision-making if



You may now connect the individual parts with each other. Which tool would you use?

Figure 6.2.: Screenshot of the interface for the planning task (Kraus et al., 2020c). Users could choose between four different methods for task completion. All options were presented textually and visually. The selection was made either by clicking on the respective button or by confirming NAO's proposal.

required. Additionally, an artificial rewarding system was implemented to better motivate participants to engage in the task. This was also intended to provide a risky environment in which trust would be important. Therefore, options were associated with a rating system based on three fictional categories: quality of product, cost, and time efficiency. Each category was rated between 0 and 10 scoring points. The most common approach to performing a task was awarded the highest scoring (30). Alternative approaches that were functional but more cumbersome or cost-intensive were awarded 0, 10, or 20 depending on their usefulness. In the example depicted in Fig. 6.2, the cordless screwdriver was the best option, while the usage of the nail gun was rated as an inappropriate tool for this task and rated with 0 points. After selection, the score of the chosen approach was presented to the user as direct feedback.

The assistant was designed to be an expert system avoiding the unintended side effects of incompetent system behaviour on perceived user trust and usability. Thus, it would only suggest the most suited options. This allowed us to only consider the effects of the proactive levels for evaluation.

For the experimental setup, study participants worked on the decision-making task using a laptop with a purpose-built GUI. The GUI could be manipulated using mouse clicks. As an external representation of our assistance system constituted a better separation of task and assisting technology, a NAO robot was positioned next to the laptop. This 120 cm high humanoid robot from SOFTBANK ROBOTICS has an integrated speech interface that enables a natural approach to dialogue control. Hence, the setup seemed more realistic and was expected to deliver more significant results. To avoid confusing the user and to achieve a better quality of speech recognition, the robot had a fixed position and deactivated autonomous movements. The experimental setup is illustrated in Fig. 6.1. The user was instructed which phrases can be used, e.g. "*Which option do you recommend in this situation?*". However, if user input was not recognised, the system automatically requested a repeat. Note that the robot could not physically solve any tasks, but only advise users using natural language. Thus, the robot was implemented to be a virtual CA.

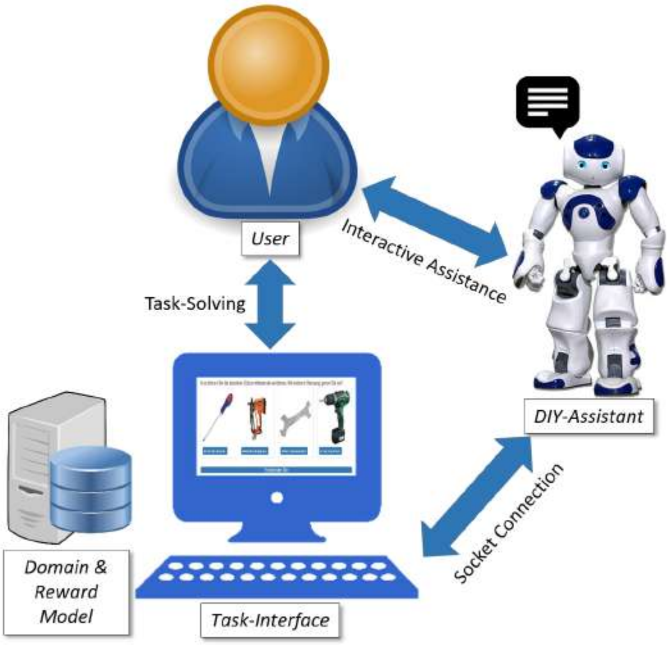


Figure 6.3.: Depiction of the study apparatus.

6.1.3. Prototype Description

For conducting the user study, we implemented a prototypical system consisting of a task interface, a domain and reward model, as well as a virtual CA in the form of a Nao robot. The study apparatus is visualised in Fig. 6.3. The implementation of the prototype was based on the previously described conceptualisation of the cognitive architecture. The user interface presenting task content was implemented as a clickable web application using the JavaScript framework with a Bootstrap plugin for designing the web pages. Fig. 6.2 shows a screenshot of the designed interface. The web page was structured in such a way that the description of the sub-task was presented on top of the screen, whereas the four different options were put in the line below the assignment. Each option was presented with a picture of the tool or approach and the corresponding label. Nao's proactive messages and responses to user requests were provided as spoken utterances using natural language. At each task step, the user could select one option using a mouse click and/or conducting a spoken dialogue with the robot for receiving guidance on decision-making. The domain model contained the content of the individual task steps, options as well as the content of the assistance messages. The corresponding texts and images were pre-defined in advance and stored as hypertext markup language (HTML)-templates. Similarly, the reward model was pre-defined while the association of the scoring points to the respective options was carried out relying on DIY-knowledge from internet research. The associations between scoring points and options were implemented using key-value pairs.

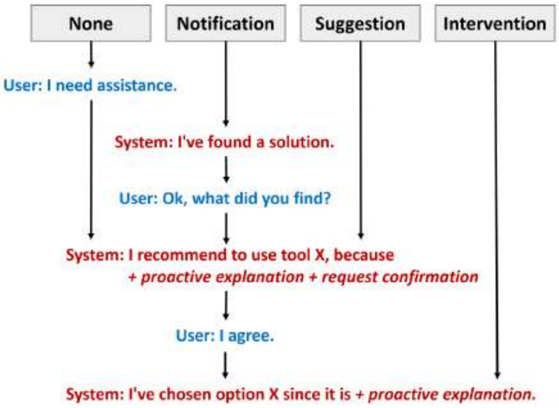

None
Notification
Suggestion
Intervention
User: I need assistance.
System: I've found a solution.
User: Ok, what did you find?
System: I recommend to use tool X, because
+ proactive explanation + request confirmation
User: I agree.
System: I've chosen option X since it is + proactive explanation.

Figure 6.4.: Flowchart, visualising the dialogue content of different levels of proactivity. User utterances are coloured in blue, while system actions are red-coloured (Kraus et al., 2020c).

To obtain knowledge about when and how to provide the proactive messages, we connected the Nao assistant to the web interface using Nao's QiMessaging developing framework. QiMessaging makes use of JavaScript bindings for accessing Nao's speech modules. The bindings provide the class QiSession that connects to the robot and gets proxies to services. After creating a session, Nao's modules (services) can be called using the service() function. This provides a JavaScript proxy to any service. These services are JavaScript objects exposing methods and signals. A service method, e.g. the "AlTextToSpeech" method allowing Nao to provide speech output, is completely asynchronous. This enabled the interaction of Nao to be proactively initiated through timeouts on a web interface. Additionally, we implemented the "ALSpeechRecognition" service method for setting the language and vocabulary of the assistant's speech recognition. We provided a rich vocabulary for ensuring the recognition of multiple paraphrases of the statements the user was allowed to utter. For creating a system response upon the recognised user input, Nao's internal memory "ALMemory" was used. This memory provides callbacks on specific events, e.g. when speech was recognised. The event for speech recognition was subscribed to by our agent to react appropriately to speech commands. For example, in case the user uttered 'Which option do you recommend in this situation?' at the previously described task step "connect the parts of the nesting box", the assistant would provide the suggestion "*The solution with the cordless screwdriver sounds good because it is the most time-efficient way. Should we choose this solution?*". As a response, the user could either accept or decline this offer using speech. This in turn would also trigger a speech event. How the timing and proactive dialogue actions of Nao were designed and implemented is described in detail in the next section.

6.1.4. Design of Proactive Dialogue Strategies

Proactive assistance was modelled according to the conceptualisation of proactive dialogue action types defined in Chapter 5: *None*, *Notification*, *Suggestion*, *Intervention*. Since the scenario was a sequential decision-making task and the user was supposed to select the best option in their opinion per individual task step, the purpose of CA behaviour was to provide helpful information and suggestions for the selection process via natural language. Therefore, we modelled the content of the general proactive action types to fit our use case. The explanations accompanying the proactive messages were generated using scripted templates.

Using the reactive *None* action, the system awaited the user to explicitly ask for suggestions. For example, a user could say "NAO*, help me.*" for receiving assistance with decision-making. As a response, the robot would then suggest the solution with the highest score, which was equivalent to the *Suggestion* action. The more conservative proactive actions *Notification* and *Suggestion* let the user confirm the assistant's proposals and differ only in the degree of directness. While *Notification* allowed users to ignore the system's message and proceed on their own, the *Suggestion* action expected the user to accept or decline the offer. When users reacted to a system's notification, a suggestion was triggered. The *Intervention* action took the responsibility completely out of the user's hands by autonomously choosing an option. Here, the system would utter: "*I have chosen the solution with the cordless screwdriver because it is the most time-efficient*". Simultaneously, the option was selected on the task screen and the user was led to the next task step. Possible dialogue flows of the proactive strategies are depicted in Fig. 6.4. The reason why the *Suggestion* strategy was used, either upon user request or after the user had reacted to an active system notification, was to induce a natural interaction behaviour. If NAO's proposal was rejected by the user, the system did not engage in any further proactive interaction at the present task step.

For triggering the proactive system's actions, we made use of timeouts. This allowed for specifying an elapse of a certain time, after which the robot was taking the initiative. Thus, the timing of the proactive actions was well-defined. The rules upon which we implemented the timing of system actions are explained in the following. Here, we differentiated between two timing strategies:

Fixed timing strategy: This strategy was used as a baseline. For this purpose, we hard-coded the points of time the system took the initiative during the execution of the planning task. Of the five possibilities for taking the initiative, the system proactively acted on the sub-tasks one, four, and five. In doing so, a "quasi"-random proactive system behaviour was simulated. Randomly distributing the timing was omitted to guarantee better comparability among subjects. Technically, NAO took the initiative eight seconds after the respective task screen was loaded. To avoid that subjects could select an option before NAO had behaved proactively, we blocked the selection buttons for this period. As a cover-up, participants were told that we wanted to guarantee that they have read and understood the task.

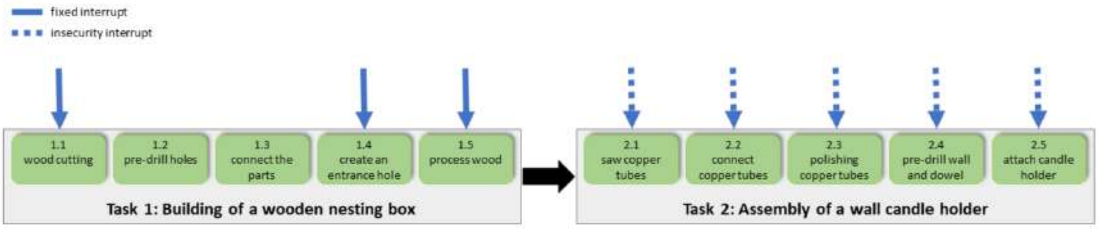


Figure 6.5.: Schematic description of the study procedure (Kraus et al., 2020c). Both planning tasks consisted of five sub-tasks. For the strategy “fixed timing”, the system intervened proactively in the sub-tasks 1, 4, and 5. For the strategy “uncertainty’, the system intervened proactively each time user insecurity was detected. The two strategies were switched depending on the test condition.

Uncertainty-based timing strategy: Here, the robot could take the initiative at each project step, if the subject had not requested help or had not selected an option before a time limit of twelve seconds. We interpreted the four seconds of user inactivity after the selection buttons had been enabled as uncertainty in task performance. As using hesitation as an indicator for uncertainty is extremely user-dependent, this period was chosen as a heuristic measure based on pre-testing.

By testing uncertainty-based against baseline timing, we intended to gain knowledge of whether it was possible to use hesitation as a signal for user uncertainty. If yes, this would allow to include this measure in the user state for initiating proactive dialogue behaviour. Consequently, the usefulness of this metric for triggering adequate proactive dialogue behaviour could be evaluated. In summary, the developed prototype including the different proactive dialogue strategies allowed us to study the impact of proactive dialogue level and trigger mechanism on the cooperation. In the following, we describe the experimental design for studying our research questions concerning trustworthiness and usability.

6.1.5. Experimental Design

In our study setup, a 2x2x4 mixed factorial experimental design was conducted with proactive dialogue action types (none - notification - suggestion - intervention) as between-independent variables. Moreover, task difficulty (low: nesting box - high: wall candle holder), as well as the timing strategies for proactive behaviour (fixed - insecurity-based), were used as within-subject variables. The order of the timing strategies was randomised for each proactive dialogue strategy except for the none condition, which did not require any timing due to reactive behaviour. The order of the tasks was the same for all users. Participants were distributed randomly to each experimental group.

Participants

42 German participants (50 % female) with an average age of 26 ($SD = 4.15$) were recruited and received 10 € as a reward. Most subjects were students (37) majoring either in psychology (27 %) or in computer science (38 %).

Experimental Procedure

After the welcome procedure, participants were provided with first instructions and details about the study. As a cover story they were told that the purpose of the study was to test a decision-making algorithm of the Nao robot and to generally consider problem-solving between humans and robots. Afterward, they had to read and sign the informed consent. In addition, they had to fill out a pre-test questionnaire regarding demographics, their personality, and possible confounding variables. Before the first interaction cycle, they received detailed information about the tasks and the procedure of the study. This included details about the speech capabilities of Nao and about the task to rate the interaction with the robot. Subsequently, the participants had to work on planning the first DIY-project. After completion, they had to fill in a questionnaire to assess the dependent variables and to check the manipulations. The same procedure was repeated for the second task scenario. In addition, the questionnaire provided after the second task also contained an evaluation of the overall perceived user experience with the virtual CA. In conclusion, participants received their reward and were dismissed. A graphical representation of the procedure is depicted in Fig. 6.5.

Questionnaires

In our experiment, we assessed trust and its five bases (competence, reliability, understandability, personal attachment, and faith) in the robot and the participants' cognitive loads during the interaction to evaluate the effects of proactive dialogue behaviour on the cooperation. Furthermore, we measured the user's experience with the system in general for checking the quality of the setup. Each variable was measured with items from established and validated scales. To determine trust towards the robot, the short version of the Trust in Automated Systems Scale (Jian et al., 2000) in German by Kraus (2020) was implemented. Furthermore, scales for measuring the bases of trust developed by Madsen and Gregor (2000) were used. For measuring three types of cognitive loads (extraneous, germane, intrinsic), a questionnaire developed by Klepsch et al. (2017) was included. The user's experience with the system was assessed via the user experience questionnaire (UEQ) developed by Laugwitz et al. (2006). Besides, for personality assessment, the Big-Five-Inventory BFI-10 by Rammstedt et al. (2013) was included. The scales, which were only available in the English language, were translated into German. Besides, all scales were slightly modified for content and study context.

Possible confounding variables were measured using scales of propensity to trust autonomous systems (Merritt et al., 2013), negative attitudes towards robots scale (NARS) (Nomura et al., 2006), as well as self-developed scales for previous experience with speech

DSs and DIY-tasks. All scales were rated on a 7-point Likert-scale from 1 (strongly disagree; word adjective (UEQ)) to 7 (strongly agree; word adjective (UEQ)).

6.1.6. Results

For data analysis, we used t-tests for the manipulation checks, a multivariate analysis of variance (ANOVA) for confounding variables, as well as a mixed ANOVA for testing the significance of the developed proactive dialogue strategies. No significant outliers were found in the data set. Due to the number of samples, a normal distribution could be assumed.

Confounding Variables and Manipulation Check

Confounding group differences for proactive behaviour could be ruled out as the multivariate ANOVA did not reveal any significant differences (all p-values >> .05). The evaluation of the manipulation check confirmed the successful manipulation of proactive dialogue behaviour (all p-values < .05 concerning the non-proactive strategy). However, the manipulation of the timing of proactive behaviour dependent on the subject's uncertainty was not recognised by users (all p-values >> .05). Therefore, we concluded that user uncertainty could not be measured using hesitation. The two tasks differed significantly in their level of difficulty as expected. The conduction of a paired t-test revealed that the intrinsic cognitive load, related to the difficulty of a task, was rated significantly higher for the wall candle than for the nesting box decision-making task ($M = 1.94,\ SD = 1.08$ for nesting box vs. $M = 2.48,\ SD = 1.20$ for wall candle, $t(41) = -3.46,\ p < .01$). Hence, differences between proactive dialogue actions depending on task difficulty could be observed.

User Experience with the Experimental Prototype

In order to ensure the functionality and usefulness of employing Nao as virtual CA, we evaluated the system regarding user experience. In general, the system received positive feedback. Participants rated their interaction partner well understandable, represented by a high value for perspicuity ($M = 5.45,\ SD = 1.18$). Furthermore, the system received good ratings for dependability ($M = 5.42,\ SD = .87$) and efficiency ($M = 5.21,\ SD = .99$). In addition, the interaction with Nao received moderately good ratings for attractiveness ($M = 4.99,\ SD = .83$), novelty ($M = 4.77,\ SD = .92$), and stimulation ($M = 4.80,\ SD = .87$). Thus, the design of the system prototype for assisting in this task domain was successful.

Effects of Proactive Dialogue Strategies on Usability

Regarding usability which was measured using the UEQ's "Efficiency" sub-scale, the *Notification* action showed the highest ratings ($M = 5.55,\ SD = 1.02$) followed by the *Intervention* action ($M = 5.18,\ SD = .92$). The *Suggestion* action showed the lowest

Proactive Action	**Trust**			**Efficiency**		
	Female *M (SD)*	**Male** *M (SD)*	**Overall** *M (SD)*	**Female** *M (SD)*	**Male** *M (SD)*	**Overall** *M (SD)*
None	5.70 (.78)	5.25 (1.05)	5.52 (.87)	5.41 (.92)	4.56 (1.39)	5.08 (1.14)
Notification	6.52 (.45)	6.09 (.67)	6.25 (.61)	6.13 (1.09)	5.21 (.88)	5.55 (1.02)
Suggestion	5.72 (1.05)	5.98 (1.43)	5.82 (1.14)	5.39 (.93)	4.44 (.66)	5.05 (.94)
Intervention	6.27 (.34)	5.21 (1.00)	5.64 (.94)	5.31 (.83)	5.08 (1.04)	5.18 (.92)

Table 6.1.: Descriptive statistics of overall perceived trust and task efficiency regarding the proactive dialogue actions and respective participant gender.

usability scores ($M = 5.05,\ SD = .94$). The *None* action received slightly higher ratings in comparison ($M = 5.08,\ SD = .92$).

Effects of Proactive Dialogue Strategies on Trust

There was a statistically significant interaction between proactive dialogue actions and task difficulty for perceived competence ($F(3, 38) = 8.25,\ p < .001,\ \eta^2 = .39$) and for perceived reliability ($F(3, 38) = 3.95,\ p = .015,\ \eta^2 = .24$). In order to investigate further which groups differed significantly in which task, a series of t-tests with Bonferroni correction was conducted. First, we examined the effects of proactive actions on perceived competence. The *Notification* action was evaluated significantly higher than the *None*, and *Intervention* action for the task *nesting box* ($t(19) = 4.46,\ p < .001$ vs. None; $t(19) = 2.93,\ p = .038$ vs. Intervention). Furthermore, for the more difficult task *wall candle* the *Notification* action was rated higher than the *Intervention* action ($t(19) = 2.90,\ p = .038$).

In the following, results for perceived reliability are presented. For the relatively easier task *nesting box*, the *Notification* action was graded significantly higher than the *None* ($t(19) = 3.03,\ p = .028$ vs. None). For the harder task *wall candle* the *Notification* and *None* action were rated significantly higher than the *Intervention* action (Notification vs. Intervention, $t(19) = 2.96,\ p = .032$; None vs. Intervention, $t(18) = 2.84,\ p = .044$). These results are depicted in Fig. 6.7.

Finally, we investigated significant main effects of proactive dialogue action types. The *Notification* action was evaluated significantly higher than the *Intervention* action for the categories perceived competence ($t(19) = 3.02,\ p = .028$) and perceived reliability ($t(19) = 3.16,\ p = .020$). Additionally, the *Notification* action was rated significantly higher than the *None* action for perceived competence ($t(19) = 2.00,\ p = .036$). No significant results were found for all of the remaining dependent variables. Considering the trust progression throughout the experiment depending on the proactive actions, we investigated the within-subject differences of the trust ratings before the experiment and after each task. Hereby, initial trust was measured using the trust propensity in autonomous systems scale. For testing the significance of the differences, we used paired t-tests. We found a significant trust difference for the *None* action measured after the

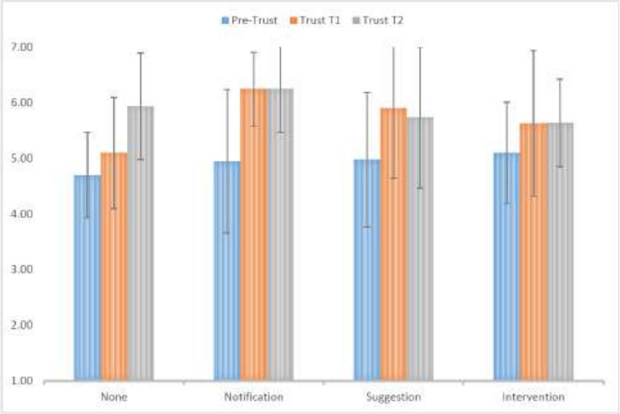


Figure 6.6.: Trust progression throughout the experiment concerning the proactive dialogue actions. Pre-Trust represents predisposition to trust autonomous systems, while Trust T1 and T2 represent the trust measurements after the tasks "nesting box" and "wall candle holder" respectively.

first and second task ($t(9) = -3.00,\ p = .015$). Furthermore, we found significant trust differences between initial trust and trust measured after the first task for the actions *Notification* ($t(10) = -3.95,\ p = .003$) and *Suggestion* ($t(10) = -2.90,\ p = .016$). There was no significant trust progression for the *Intervention* action. The results are depicted in Fig. 6.6.

Interplay between Proactive Dialogue and User Characteristics regarding Trust

Further exploring the data, we found significant gender differences using t-tests on the independent samples. Females rated themselves to be less experienced with DIY ($t(40) = 2.13,\ p = .039$). Additionally, they showed tendencies to be less experienced interacting with CAs ($t(40) = 1.94,\ p = .059$). Considering personality characteristics, females had higher ratings for neuroticism ($t(40) = -3.33,\ p = .002$) and conscientiousness ($t(40) = -2.22,\ p = .032$). Females rated themselves also considerably more open to experiences ($t(40) = -1.83,\ p = .075$).

For observing the effects of the proactive actions depending on the individual gender, we split the data set accordingly and tested for significant differences. Due to the resulting smaller sample size, a normal distribution of the data was not further provided. Therefore, we utilised a Kruskal–Wallis one-way analysis of variance for testing the effects of the different proactive actions. Here, several significant differences were found for the female gender. For the task "nesting box", significant differences were found for reliability ($p = .042$) and competence ($p = .038$). A significant difference was found for the task "wall candle" regarding reliability ($p = .005$). Additionally, we found a significant effect on overall perceived reliability ($p = .020$).

In order to investigate further which groups differed significantly for each task, post-hoc tests using the Dunn-Bonferroni method were conducted. The results showed that the

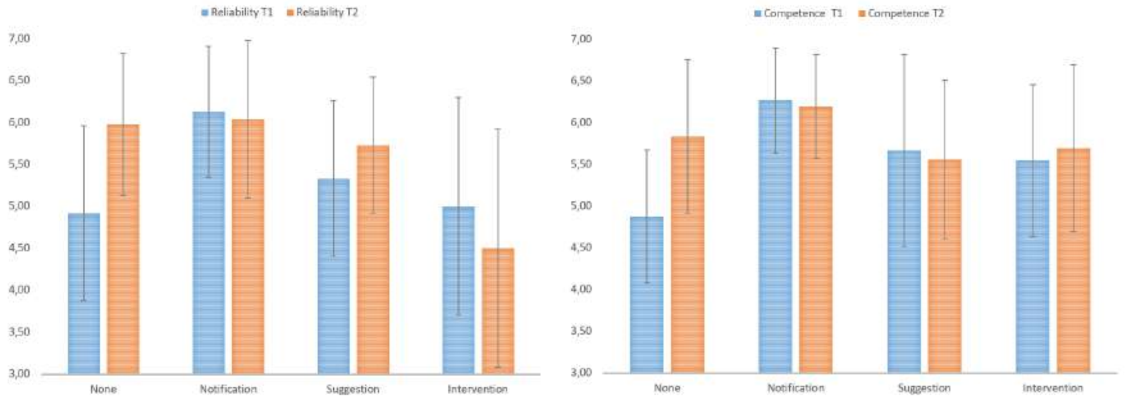


Figure 6.7.: Depiction of the results for perceived reliability (left) and perceived competence (right) depending on the four proactive dialogue strategies (Intervention, None, Notification, Suggestion) and the two tasks (1 = 'nesting box'; 2 = 'wall candle'). Mean values and standard errors are provided.

Notification action was rated higher than the *None* action for reliability ($Z = -2.83$, $p =$ .028 and competence ($Z = -2.88$, $p =$ .024in the task"nesting box". For the task "wall candle", the *None* and *Notification* action were rated higher than the *Intervention* action for reliability ($Z = -3.18$, $p = .009$; $Z = -2.95$, $p = 0.019$). Additionally, we found a tendency that proactive actions had an effect on competence ($p = .055$) and understandability ($p = .059$) for the task "wall candle" and for the UEQ-dimensions novelty ($p = .090$) and dependability ($p = .093$). For the male gender no significant differences were found.

6.1.7. Discussion

The study results verified our hypotheses that altering the degree of proactive system behaviour has a significant impact on the user's trust in the CA. Especially, we discovered interesting insights into the relations between proactive actions and task knowledge/difficulty as well as user characteristics on the perceived competence and reliability of the system. In the following, we discuss the results with a focus on the formulated research questions.

Influence of Proactive Dialogue Level on Usability

Even though there were no significant differences between the measures for perceived usability, it was interesting that the *Notification* action provided the best results ahead of the *Intervention* action, which naturally provided the highest task success due to the system being an expert system for this task domain. This was a further indicator that the *Notification* action let the system be perceived as more competent for the task. *None* and *Suggestion* action received the lowest scores. This was found to be quite plausible, as users had to make decisions completely on their own in the reactive condition. Considering the *Suggestion* action, it seemed that the additional decision to accept or to decline the system's action was perceived to be a factor that decreased usability.

Influence of Proactive Dialogue Level on Trust

For the first, easier perceived task "nesting box", low- and medium-level proactive system actions were particularly trusted more than the reactive condition. This was validated both by the examination of the trust progression analysis and the ANOVA. For this, there exist two possible explanations.

First, users could have been more sure about the decision on appropriate planning steps for this task in comparison to the "wall candle task". Therefore, the proactive actions could have been perceived as a confirmation or reinforcement of their decision-making processes and relieved them in task execution. This in turn could foster trust, as the benefits of proactive actions were higher as compared to the risks of wrongful system advice. Particularly, as the low- and medium-level are more controllable (Isbell and Pierce, 2005).

The relatively low ratings for competence and reliability of the *None* action for the "nesting box"-task could be explained that the ratio between expenses and benefits of system usage was too low, as requesting the system for help was perceived as an unnecessary step and could have been more a distraction. For the second, more difficult task, the *None* action was trusted similarly to the medium-level proactive conditions, as the benefits of requesting system help outweighed the costs of addressing the system. Another explanation could be those study participants perceived low- and medium-level proactive actions to help better in getting familiar with the task and the CA's design and performance than a reactive system that does not actively communicate. Hence, the dynamically learned trust according to Hoff and Bashir (2015) was increased more by proactive actions in the first task, because they initially made the system more transparent. For the second task, the *None* action increased the dynamically learned trust as subjects started communicating more with the system and learned about its benefits.

Among the proactive strategies, the *Notification* action had the most impact on conveying competence and reliability of the system. This particularly held for assistance in the first, easier task. The *Notification* strategy was the most conservative proactive strategy, which offered help more subtly. Hence, study participants always felt in control but were also aware of the system's active assistance. Furthermore, this strategy comprised the most (four) dialogue turns. It seemed that subjects tended to accept proactive system behaviour more when it was possible for them to have natural dialogues.

In line with the findings from our requirement analysis and previous work by Rau et al. (2013), the most autonomous system behaviour, the *Intervention* action, was less trusted than the more conservative strategies. Subjects considered this strategy to be too obtrusive and perceived the system to be imposing. In summary, when being proactive, a system should act more subtle and give the user a feeling of system involvement in the task, i.e. by notifying about or suggesting information. The *Intervention* strategy could be used for really tedious or annoying tasks. Therefore, we considered the *Notification* and *Suggestion* strategies as more trustworthy for the user.

These findings reinforce the results by Peng et al. (2019), who designated medium-level proactivity as the most helpful.

When considering how user characteristics affect the relation of proactive system actions and HCT, we found significant gender differences. The interplay between gender and trust is a common phenomenon in engineering and science (Kraus et al., 2018; Tannenbaum et al., 2019; Law et al., 2020). We found that varying the degree of proactive system behaviour had a particularly significant impact on the female user's cognitive-based trust, reliability, and competence. Female study participants were less experienced with CAs and DIY than male subjects. This suggested the first evidence, that the perception of proactive system behaviour as trustworthy is crucially affected by the user's experience with the task and technology. Furthermore, we found significant differences between the genders regarding the big five personality traits as females rated themselves higher for neuroticism, conscientiousness, and to some degree openness towards new experiences. A high degree of openness to experience relates to curious, innovative, adventurous persons. A high degree of conscientiousness relates to goal-oriented, efficient, disciplined, organised behaviour. Sensitive, insecure individuals have a high degree of neuroticism. Examining the individual personality traits it could be reasoned that proactive behaviour primarily affects innovative, goal-driven, but also more insecure persons. Interestingly, in organisational psychology and management, proactive behaviour is associated with goal-directed activities and innovation (Crant, 2000; Frese and Fay, 2001) relating to the traits of openness and conscientiousness. Seibert et al. (1999) also introduce the "proactive personality". Hence, there could be a correlation between one's tendency for proactive behaviour and the perception of a proactive CA. However, more research on this topic is necessary for providing clear insights and underpin this hypothesis. Nonetheless, taking into consideration the user's personality when developing a proactive CA could be beneficial.

The reason why overall trust in the system did not differ significantly could lie in the short duration of the interaction. These kinds of interactions only influence the cognitive-, and not the affect-based trust. To get significant differences in overall trust, a more long-term human-machine relationship might be necessary. According to Madsen and Gregor (2000), both cognitive- and affect-based trust must be perceived as high to establish an overall trustworthy CA. Further, our investigations of user requirements showed that reactive behaviour may be more expected in this task domain. Thus, reactive behaviour was supposed to receive higher trust ratings. As there was no significant difference, we deemed the proactive dialogue act types to comply with the principles of proactive behaviour which validated their utility.

Influence of the Trigger Mechanism on Cooperation

Manipulation of timing strategy according to the user's uncertainty failed. In consequence, we assumed that a time-dependent measure for uncertainty is insufficient for usage as a trigger-variable of proactive dialogue actions. Arguably, time as initiation-criterion needs to be avoided because there exist too many side factors, which are not necessarily user-related, that could lead to a delay in time.

Limitations

Our work had several limitations. Even though we let subjects interact with an actual autonomous system, the study was still conducted in a controlled environment. In a realistic scenario, a DIY-planning task would be much more unpredictable and unbounded. Additionally, Nao only allows for a limited speech interaction due to its technical constraints. Thus, more sophisticated interfaces may be beneficial for a more realistic evaluation. Furthermore, the timing strategies can only be controlled in an experimental setup and can hardly be transferred to a real case scenario. However, using the user's insecurity as a metric for timing proactivity proved to be unreliable. Therefore, other metrics are required to be identified for measuring user insecurity. Finally, since we kept using the same level of proactivity for a subject while going through a study run, this may have resulted in the perception of a rigid system harming the overall user experience. This approach was necessary to consider the independent effects of the individual proactive dialogue acts on the cooperation though.

6.1.8. Conclusion

The results of this study on the user perception of proactive dialogue action types showed an overall benefit of low- to medium-level proactivity and its relations to the HCT relationship. Furthermore, we discovered an interaction between proactive actions and perceived task difficulty, as well as dependencies between proactive dialogue and certain user characteristics, such as domain experience, technical affinity, and personality properties. The results further showed that the low- to medium-level proactivity was better for establishing an immediate trust relationship. Particularly, the proactive dialogue seemed to be especially relevant for novice users. Further, we found a slight effect of proactive dialogue act types on usability, which stressed the benefits of the *Notification* action. Using a time-based hesitation measure for measuring user uncertainty did not have the intended effect. Thus, there was no difference between the different implemented timing strategies for proactive behaviour. For this reason, we decided to use more sophisticated user state models for triggering the initiation of proactive dialogue behaviour. As related work showed that a user's cognitive-affective state could be useful for enabling proactive assistance, we included the measurements of such in a follow-up prototype. This was then used to investigate the effects of different proactive dialogue strategies dependent on cognitive-affective user states on the cooperation.

6.2. Effects of Proactive Dialogue Strategies Dependent on Cognitive-Affective User States

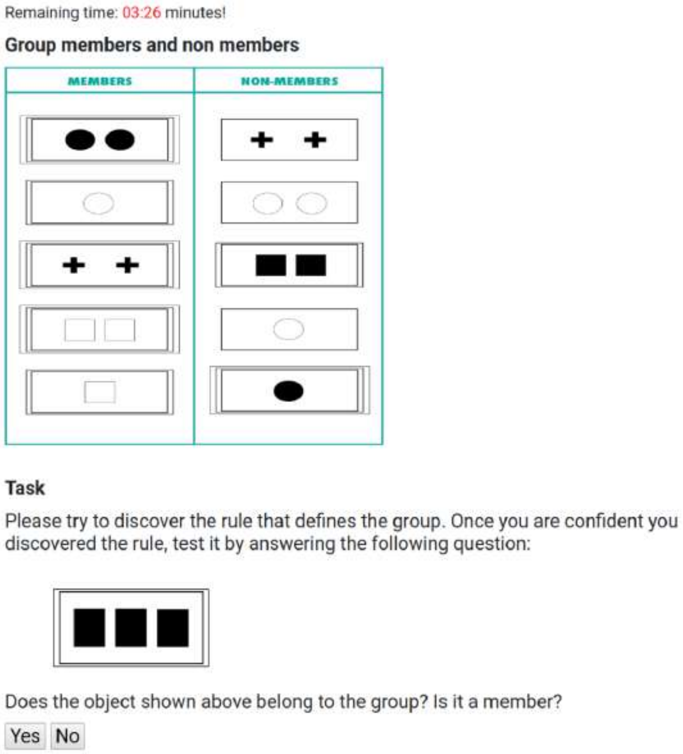
6.2.1. Motivation

In this experiment, we investigated the effects of the modelled proactive dialogue actions on human-machine cooperation similar to the previously described experiment. However, the main research question was to consider, whether it was possible to determine the user's need for assistance with the presence of negative cognitive-affective states during a decision-making task.

Therefore, the user's cognitive-affective states were taken into account as an indicator of an appropriate point in time for the invocation of conversational assistance. For this, it was important to define which cognitive-affective states would be considered. According to the attentional control theory, mainly the user state anxiety pose a need for assistance (Eysenck et al., 2007). However, research has been extended to other negative cognitive-affective states, in particular those related to the interaction with technical systems (Hibbeln et al., 2017). Those negative cognitive-affective states have been described by a negative affective valence which can be evaluated by facial expression analysis or with facial electromyography tools (Ekman, 1993). To be more specific, some of these negative states would be boredom, frustration, and confusion (D'Mello and Graesser, 2011). According to D'Mello and Graesser (2011), negative states such as confusion and frustration are usually associated with mistakes, failure, struggling with problems, or revising plans, while positive ones such as excitement or delight are associated with task completion or making discoveries.

In the following experiment, we focused on the negative states of confusion and frustration for initiating proactive dialogue. Similarly to Friemel et al. (2018), these states were measured using visual cues. For this, a high-resolution camera and the Affectiva (McDuff et al., 2013) software for classifying affective states were used. For measuring the effects of proactive dialogue depending on the user's cognitive-affective state on cooperation, a user study was conducted. Here, study participants performed a concept learning task that involved planning, categorising, and decision making. During the task, they interacted with a Nao robot as virtual CA similar to the previous experiment. It would also provide help either in a reactive or proactive manner. Here, it was also tested whether there exist general differences between the conceptualised proactive dialogue act types and HCT and usability with a focus on the present user state. Therefore, the virtual CA was equipped with different timing strategies. The intervention started either after a random time interval or after the detection of frustration or confusion.

For this experiment, proactive behaviour was applied for assisting users during decision-making for a learning task. Therefore, we also observed the effects of the different proactive dialogue act types on the user's cognitive load depending on the cognitive-affective user state. This was due to the relations between cognitive load and associated learning of a user as described in Chapter 2. In addition, we evaluated the quality of the study setup by investigating user experience measures for ensuring the usefulness of the prototype.



Remaining time: 03:26 minutes!

Group members and non members

Task

Please try to discover the rule that defines the group. Once you are confident you discovered the rule, test it by answering the following question:

Does the object shown above belong to the group? Is it a member?

Figure 6.8.: Depiction of the user interface for solving the conceptual learning task.

In the following, we describe the experiment including the use case scenario, system description, and proactive dialogue strategy design. Moreover, we explain the study setup in detail and report results. These outcomes are then thoroughly discussed.

6.2.2. Scenario

For testing the proactive dialogue strategies, a concept learning scenario was selected in which the user was accompanied by a Nao robot serving as virtual CA. Here, the assistant took the role of a tutoring system (Graesser et al., 2005) that did not physically take action. Using a concept learning task was inspired by the work of Bruner et al. (2017). Their work was based on how humans categorise information by applying a coding system. The participants saw ten objects divided into two columns: Five labeled as members, and five as non-members. The task of the participants was to deduce the correct rule. Instead of explicitly asking the participants for the rule, a new unlabelled object was presented separately. They were then asked to classify it as a member or non-member of the group. The properties that defined each object were as follows: Number of elements: One, two, or three elements. The shape of elements: A square, a cross, or a circle. The number of borders: One, two, or three borders. Filled or not filled elements. Fig. 6.8 illustrates an example of the task presented. This scenario was chosen for two reasons:

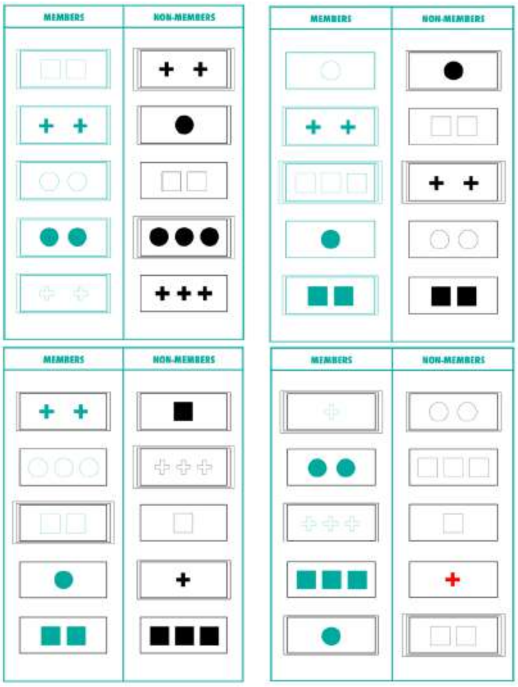


Figure 6.9.: Examples of relational rules between objects.

Firstly, the concept learning task provided a scenario with sufficient complexity where the assistance of a technical system ought to be perceived as useful. Secondly, the rule-based structure of the task allowed to equip the assistant with expert knowledge. This enabled the system to provide helpful contributions to the task. During task completion, a four-minute timer was added for putting the participant under pressure. This aimed at creating a situation of vulnerability in which trust in the assistant was necessary for successful task completion. In the following, the relational rules between objects are described. A visualisation of all possible relations is provided in Fig. 6.9:

And The two properties are connected with an "AND". The depiction of the members and non-members in the top left of Fig. 6.9 exemplifies this rule. In this example, the two properties are two borders and two elements. All members have two borders and two elements in the middle.

Specification The two properties are in a certain relation to each other. The depiction of the members and non-members in the top right of Fig. 6.9 exemplifies this rule. In this example, the number of borders is the same as the number of elements.

Or The two properties are connected with an "OR". The depiction of the members and non-members in the bottom left of Fig. 6.9 exemplifies this rule. All members have two elements or circles.

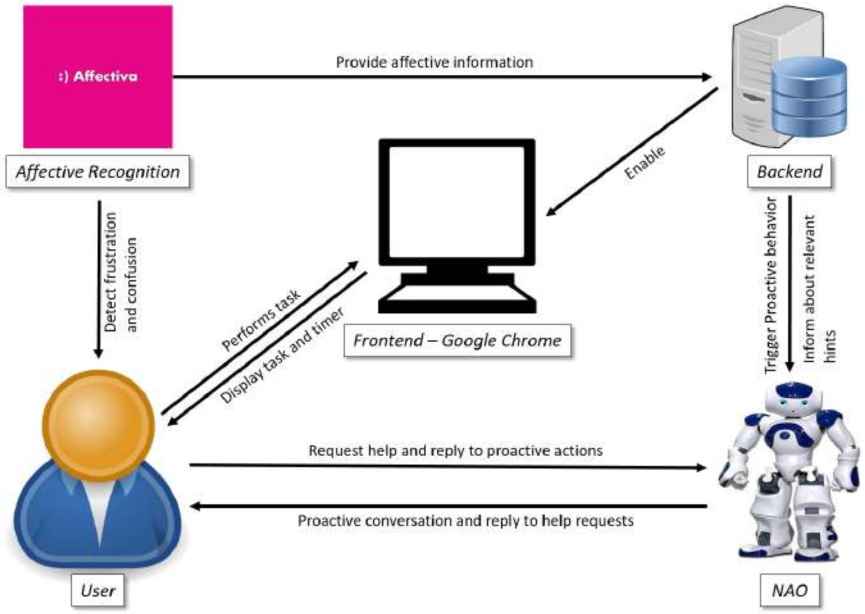

:) Affectiva
Affective Recognition
Provide affective information
Backend
Enable
Detect frustration and confusion
Frontend – Google Chrome
Performs task
Display task and timer
Trigger Proactive behavior
Inform about relevant hints
User
NAO
Request help and reply to proactive actions
Proactive conversation and reply to help requests

Figure 6.10.: System architecture. (Kraus et al., 2022a)

Exclusive Or The two properties are connected with an "EITHER OR". The depiction of the members and non-members in the bottom right of Fig. 6.9 exemplifies this rule. All members either have a cross or a filled element in the middle but not both. The cross in red in the non-member's column refers to an example that has both properties, an element in the middle and a cross, and therefore does not classify as a member.

6.2.3. Prototype Description

The prototype was developed similarly to the previous experiment. A Lenovo Thinkpad laptop computer, equipped with a webcam, was used for the experiment administration and information recording. A Tomcat Apache server version V9 served as the back-end and enabled the communication between the components. The code was implemented as a dynamic web project in Eclipse, based on JavaServer Pages, HTML, and JavaScript. The front-end with the task to be solved was presented to the participants in a Google Chrome web browser. While the participants were performing the task, assistance was provided through the Nao robot. The robot was programmed to listen to user speech and, if requested by the user, to provide hints for solving the task. This behavior was implemented in the JavaScript QiMessaging application program interface (API) by Aldebaran. Nao received information about task-relevant hints through the back-end. Further, the connection to the back-end allowed to trigger proactive dialogue after a random amount of time or after the cognitive-affective states of frustration or confusion had been detected. The random timing strategy was implemented with built-in JavaScript functions. For the detection of cognitive-affective states, the AFFECTIVA JAVASCRIPT API was used (McDuff et al., 2013). Generally, AFFECTIVA analyses spontaneous facial expressions with facial emotion recognition algorithms. These are trained based on a large database of faces from a variety of different countries and morphological groups. Fig. 6.10 shows the interaction between the user and the components on an abstract level.

6.2.4. Design of Proactive Dialogue Strategies

The proactive behavior of the robotic assistant was modelled again according to the previously described conceptualisation. The content of the proactive actions was adjusted to fit the context. In the following, the individual proactive actions are described more in detail:

None: This level was the foundation considered to create the reactive condition which would serve as the baseline. In the reactive condition, the robot reacted to any user help request but did not show any proactive behavior. In this level, the user could ask for help with any word similar to “Help.” or “Give me hints, please.”, but the system would not initiate the help autonomously nor provide solutions. Therefore this level did not require any timing strategy. For example, consider the following dialogue:

U: Nao, help me please.
N: The rule is of type or.
U: Thanks, can you give me another hint?
N: Pay attention to the number of elements, it is part of the rule.
U: Thank you, NAO.
N: You are welcome.

Notification: This strategy was the first level of expressing proactive behavior. It was implemented by informing the user that a hint was available. Users then had the option to say whether they wanted to hear it or not. If the user replied affirmatively, they would receive the hint. If they replied negatively the robot would wish them luck solving the task. For example, consider the following dialogue:

N: Help is available. Do you want me to give you a hint?
U: Yes, please. / No thanks.
N: Focus on the borders. It is relevant for the rule. / Good luck solving the task.
U: Thank you, NAO.
N: You are welcome.

Suggestion: This level was implemented by the robot suggesting a hint to solve the task and the user could decide whether to use the hint or not., e.g. see the following example:

N: A hint is now available. The rule is of type and.
U: Thank you, NAO.
N: You are welcome.

Intervention: This strategy was implemented by the robot saying the answer to the task and simultaneously the correct answer was chosen on the laptop screen. In this level of proactivity, the system took the decisions for the user. For exemplifying this strategy, consider the following dialogue:

N: Help is now available. The rule is: The number of borders is one less than the number of elements.
U: Thank you, NAO.
N: You are welcome.

The timing strategy determined when the system would take the initiative. Depending on when the system would intervene, it could be perceived as helpful, disruptive, or distracting. A strategy with well-defined timing was considered as the baseline. In this condition, three moments were set in each task and a random function was implemented in JAVASCRIPT, to choose one of these three moments at each task. The moments could be thirty seconds, two minutes and a half, and three minutes. Each task lasted 4 minutes so the intervention would occur within this time-lapse at different moments for each task.

For acting upon detected cognitive-affective states an AFFECTIVA-based strategy was used. This strategy would initiate proactive behavior in the detection of confusion or frustration. These two states were detected by using facial action units (AU) based on facial action coding system (FACS) (Ekman, 1993). In general, the FACS allows linking active AUs to the underlying basic emotions of sadness, happiness, surprise, disgust, anger, and fear. Based on this, McDaniel et al. (2007) and Craig et al. (2008) showed that AUs 4 and 7 indicate confusion and AUs 1 and 2 are a manifestation of frustration. AU 4 corresponds to brow lowered and AU 7 to lid tightener. On the other side, AU 1 corresponds to inner brow raise in the face muscles, and AU 2 to outer brow raise. Proactive dialogue was triggered when either frustration or confusion was detected by the AFFECTIVA API (McDuff et al., 2013). The individual steps handled internally by the API were as follows: First, the webcam would provide raw pixel images. Afterward, regions of interest, i.e. pixels of facial information, were extracted using landmark detection. Finally, machine learning regressors predicted activity scores between 0 (no activity) and 1 (high activity) for each AU and returned the results to the caller of the API. The predictors were trained on the AFFECTIVA dataset.

The activity detected by AFFECTIVA API in each of the AUs was compared to a threshold that would trigger the robot. That threshold was defined in a pre-test with three additional participants. In this way, thresholds that seemed to provide a sensible trade-off between precision and recall were determined empirically. The details of this AFFECTIVA implementation will be furtherly explained in the pseudocode in Algorithm 1. In summary, the developed prototype including the different proactive dialogue strategies allowed to study the main research questions of this thesis concerning the human-machine cooperation and the applicability of the trigger mechanism. In the following, we describe the experimental design for studying our research questions.

Algorithm 1: Pseudo-code for the detection of the cognitive-affective states frustration and confusion.

```
subscribe to Face Detection Event triggered by AFFECTIVA API;
set au_activity_tresholds;
initialize expression_detected to False;
while not expression_detected do
    if face_detected then
        compute AU activities;
        # below: Check if confusion was detected
        if $au\_activity\_1 > au\_treshold\_1 \cap au\_activity\_2 > au\_treshold\_2$ then
            trigger proactive behavior in NAO;
            expression_detected = True;
            break;
        end
        # below: Check if frustration was detected
        if $au\_activity\_4 > au\_treshold\_4 \cap au\_activity\_7 > au\_treshold\_7$ then
            trigger proactive behavior in NAO;
            expression_detected = True;
            break;
        end
    end
end
```

6.2.5. Experimental Design

A factorial 2 x 4 mixed design was used for the experiment. The independent variables manipulated were: timing strategies triggering proactive dialogue (randomised timing vs. triggered by the cognitive-affective states confusion/frustration) as within-subject and the levels of proactive dialogue (reactive - notification - suggestion - intervention) as the between-subject factor. Participants were randomly distributed to each of the between-subject factors and were confronted with both timing strategies during the study. To minimise sequence effects, the order of the timing strategies was randomised among participants.

Participants

40 participants were recruited for the study. However, three participants had to be excluded due to not complying with the study guidelines. The average age of the participants was 26 years (Std = 5.14). 37 % of the participants were females, while 63 % percent were males. A high to advanced level of English knowledge was required to perform the study, which was why 72 % percent of the participants had a C-level of English. This corresponds to experts according to the Common European Framework of Reference for Languages. The rest of the participants had advanced knowledge. Participants' English proficiency was self-reported. 60 % of the participants were students and the rest of the participants were employees. For compensation, they received a 5€-Amazon-Voucher.

Experimental Procedure

The participants were welcomed to the study and the initial instructions for the study were presented. The informed consent was given along with any clarification on the adherence to data privacy standards. Participants were told they would participate in a study based on HRI in which decision-making skills and cooperation with the robot would be evaluated. Participants were also informed they would be recorded via video for further analysis of the interaction. Information about the assistance via proactive dialogue strategies using their facial expression analysis for cognitive-affective states and timing was initially omitted to avoid expectancy effects. However, they received details about the speech capabilities of NAO. After the introduction, a base questionnaire was presented including the demographics and possible confounding variables. Afterward, the experiment consisting of ten tasks for each participant was started. For each task, they would see 10 objects like those exemplified in Fig.6.9. Additionally, each task had an upper limit of 4 minutes indicated by a timer on the screen. The ten tasks were divided into two partitions of five tasks. For the first five tasks, subjects were randomly confronted with one of the two timing strategies. After completion, they had to fill in a questionnaire to assess the dependent variables and check the manipulations. The same procedure was then repeated for the other five tasks using the other timing strategy. The whole procedure lasted between 45 minutes to one hour. After the study, the participants were dismissed and additional clarification was provided if requested.

Questionnaires

Trust was measured using the Trust in Automated Systems Scale (Jian et al., 2000). Furthermore, scales for measuring the bases of trust developed by Madsen and Gregor (2000) were used. Acceptance was evaluated by the acceptance scale developed by Van Der Laan et al. (1997). User experience with the system was studied via the short UEQ by Laugwitz et al. (2006). Cognitive load's three types (intrinsic, extraneous, and germane) were measured with the questionnaire developed by Klepsch et al. (2017). Usability was studied with the SUS (Brooke, 1996). Possible confounding variables were taken into account with the Affinity for Technology Scale (Karrer et al., 2009), NARS (Bartneck and Forlizzi, 2004), and the Propensity to Trust Scale (Merritt et al., 2013). Additional demographic information and items related to the experience with DSs were considered. All questionnaires were adapted to a seven-point Likert scale from "Completely disagree." to "Completely agree.", and some were slightly modified to fit the study. Previous experience with DSs was the only non-Likert scale-based item, as it inquired about user experience with different existing DSs. Informed consent forms were used which contained information about the procedure, purpose of the study, data treatment, and confidentiality of the information.

Proactive Action	*Perceived Activeness*	*T-test compared to None-Action*
None	0.36 (1.38)	***
Notify	2.04 (3.39)	p = .296
Suggestion	1.85 (2.80)	p = .296
Intervention	3.17	p = .030

Table 6.2.: Manipulation check of the perceived proactivity of the system. Perceived Activeness was measured as the mean of the difference for the rating scales if users perceived the assistant as active and the scale asking the users if they perceived it as reactive. (Kraus et al., 2022a)

6.2.6. Results

For data analysis, we used t-tests for the manipulation checks, a multivariate ANOVA for confounding variables, as well as a mixed ANOVA for testing the interaction between the different proactive actions and timing strategies. A Bonferroni-Holm correction was applied, where multiple testing was conducted. No significant outliers were found in the data set. Confounding group differences for proactive behavior could be ruled out as the multivariate ANOVA did not reveal any significant differences (all p-values >> .05). The evaluation of the manipulation check confirmed a successful manipulation of proactive dialogue behavior, as the proactive actions were consistently rated higher than reactive behavior for the user-perceived activeness of the system. However, only the difference between the intervention action and reactive behavior was significant. The means and standard deviations along with the p-values are presented in Table 6.2. The manipulation of the triggers of proactive behavior failed, by explicitly requesting whether the user perceived that the system acted when they were confused or frustrated (Affectiva: $M = 3.94$; Baseline: $M = 4.19$; $p = 0.305$). However, as the perception of the level of being confused or frustrated varies from user to user and subtle frustration or confusion could occur during the experiment without the participant consciously noticing, we still conducted the comparisons concerning the different timing strategies.

User Experience with the Experimental Setup

In general, the system received positive feedback. Participants accepted their interaction partner ($M = 5.24,\ SD = 1.11$) and had a good experience, represented by a high UEQ-value ($M = 5.40,\ SD = 0.93$). In addition, the interaction with Nao received moderate ratings for usability ($M = 3.78,\ SD = .39$). Generally, users had high trust in the system ($M = 5.63,\ SD = .78$) as well as their sub-components reliability ($M = 5.21,\ SD = 1.02$), competence ($M = 5.24,\ SD = 1.01$), understandability ($M = 5.67,\ SD = .93$), and faith ($M = 5.25,\ SD = 1.09$). Moderate ratings for personal attachment ($M = 3.98,\ SD = 1.37$) were reported.

Effects of Proactive Dialogue Strategies on Usability

Regarding usability, we only found a statistically significant effect of the timing strategies for the *Intervention* action. Here, the Affectiva-trigger was rated significantly lower than the baseline-trigger for usability ($F(1,8) = 12.34,\ p = .027,\ \eta^2 = .61$). Since the user's cognitive load can be also used for measuring a system's usability to some degree, we consider the effects of GCL depedent on the proactive dialogue strategies. GCL is correlated with a learner's task engagement and task focus (Mayer and Moreno, 2002). Considering the *Intervention* strategy, we found the germane cognitive load to be rated higher for the Affectiva-trigger ($F(1,8) = 4.67,\ p = .063,\ \eta^2 = .37$).

Effects of Proactive Dialogue Strategies on Trust

There was a statistically significant interaction between proactive dialogue actions and the timing strategies for perceived understandability ($F(3,34) = 3.45,\ p = .027,\ \eta^2 = .23$) and a tendency towards an interaction for personal attachment ($F(3,34) = 2.51,\ p = .076,\ \eta^2 = .18$). For investigating the simple main effects of proactive actions and timing strategies, we conducted a one-way, respective repeated measures ANOVA. There were no simple main effects of the proactive actions depending on the timing strategies (all p-values $>>$ .05). However, we found a statistically significant effect of the timing strategies for the *Intervention* action. The Affectiva-trigger was rated significantly lower than the baseline-trigger for perceived understandability ($F(1,8) = 6.40,\ p = .035,\ \eta^2 = .44$). Furthermore, we found a tendency towards faith in the system ($F(1,8) = 3.64,\ p = .093,\ \eta^2 = .31$) being increased by the *Intervention* action.

A significant effect the timing strategy on faith in the system was found ($F(1,34) = 4.46,\ p = .042,\ \eta^2 = .12$). Generally, users had more faith in the CA acting according to the baseline timing condition Additionally, we found a tendency that the Affectiva-trigger resulted in less perceived system competency ($F(1,34) = 3.25,\ p = .080,\ \eta^2 = .09$)

For considering the trust progression throughout the experiment depending on the proactive actions, we investigated the within-subject differences in the trust ratings before and after the experiment. As described in the previous study, initial trust was measured using the predisposed trust in autonomous systems scale. For testing the significance of the differences, we used paired t-tests. Here, we found a significant positive trust development for the *Suggestion* action ($t(9) = -4.28,\ p = .002$). In summary, reactive and proactive behavior had a positive effect on establishing trust, except for the *Intervention* action. The results are depicted in Fig. 6.11. No significant effects were found for user characteristics, e.g. gender or previous experience with the DS were found.

6.2.7. Discussion

According to the results reported in this study, triggering assistance behavior depending on the negative user's cognitive-affective state had an impact on the virtual CA's perception. In the following, we discuss the results concerning the elaborated research questions.

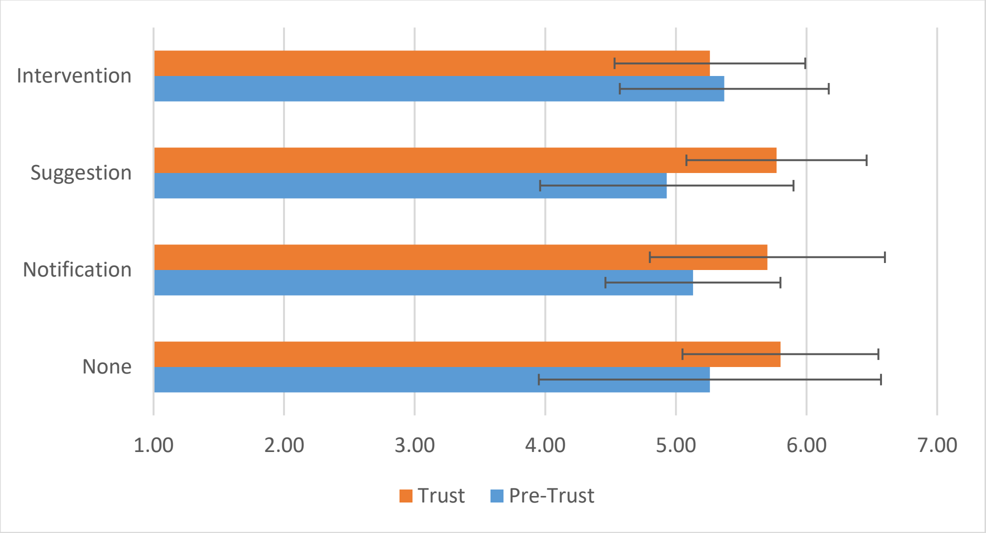


Figure 6.11.: Evolution of the trust development over the course of the experiment with regard to the proactive dialogue actions. (Kraus et al., 2022a)

Proactive Action		**Trust**	**Acceptance**	**Usability**	**UEQ**	**GCL**
		M (SD)	*M (SD)*	*M (SD)*	*M (SD)*	*M (SD)*
None	overall	5.80 (.75)	5.46 (.70)	3.89 (.34)	5.48 (.96)	5.19 (.74)
	Affectiva	5.79 (.74)	5.43 (.73)	3.96 (.36)	5.43 (1.08)	5.33 (.75)
	Random	5.81 (.83)	5.49 (.75)	3.81 (.43)	5.54 (.87)	5.05 (.73)
Notification	overall	5.70 (.90)	5.30 (1.33)	3.76 (.45)	5.54 (1.08)	4.59 (1.40)
	Affectiva	5.65 (1.03)	5.33 (1.53)	3.71 (.88)	5.52 (1.06)	4.61 (1.55)
	Random	5.75 (.87)	5.27 (1.16)	3.81 (.37)	5.56 (1.11)	4.56 (1.25)
Suggestion	overall	5.77 (.69)	5.30 (1.06)	3.90 (.39)	5.37 (.93)	4.99 (.96)
	Affectiva	5.79 (.67)	5.26 (.96)	3.89 (.47)	5.40 (.67)	5.00 (1.05)
	Random	5.75 (.85)	5.33 (1.16)	3.91 (.31)	3.34 (1.00)	4.97 (.84)
Intervention	overall	5.26 (.73)	4.99 (.79)	3.63 (.39)	5.26 (1.04)	4.97 (.77)
	Affectiva	5.11 (1.00)	4.97 (.98)	3.37 (.47)	5.06 (1.19)	5.37 (.72)
	Random	5.41 (.64)	5.02 (0.88)	3.90 (.44)	5.47 (.99)	4.56 (.82)

Table 6.3.: Descriptive statistics of the overall perceived trust and acceptance towards the system, as well as ratings for usability, user experience and germane cognitive load with reference to the proactive dialogue strategies. (Kraus et al., 2022a)

Influence of Proactive Dialogue Level on Usability

For answering this question, we only observe the value for the random time trigger of the "Usability" sub-scale, due to the negative impact of the AFFECTIVA-based timing for this metric (see Table 6.3). Here, higher levels of proactive dialogue contributed better to high usability than lower levels of proactive dialogue, even though the differences were not significant. Not surprisingly, the *Intervention* action was again providing high usability, while the *None* action was less task effective. We already described the reasons for the previous experiment. However, the perceived usability of the medium-level proactive strategies was reversed for this task context. Thus, medium-level strategies might be task-dependent in the context of usability.

As described earlier, the user's GCL may be also related to usability as it correlated with the user's learning during task execution. Surprisingly, the *Intervention* action was the only action to significantly increase a user's rated GCL (see Table 6.3). A high GCL is related to a learner's engagement with the task and their focus on the learning processes (Mayer and Moreno, 2002). Even though being considered less understandable, and usable, and users had less faith in the system's decisions, this action positively contributed to a user's learning when triggered during states of confusion or frustration. We assumed that the low trustworthiness of this proactive dialogue act type could explain the user's increased learning gain. As users did trust the automatic decision of the system less, they may have double-checked the answers of the system more closely and put more thought into the task, which could have resulted in a learning gain. Thus, a competitive CA that presents and selects the correct solution for providing samples to the users when they are frustrated or confused could be beneficial for a user's learning and thus the system's usability.

Influence of Proactive Dialogue Level on Trust

As shown in Table 6.3, low- to medium-levels of proactive dialogue (*None, Notification, Suggestion*) received higher ratings for perceived trust than a high-level of proactivity (*Intervention*). In related work considering proactive behavior for decision-making and problem-solving tasks, a high degree of proactivity also showed to have a decreasing effect on trust (Rau et al., 2013). This is also in line with the study presented in the previous section, where we reported a negative effect on the trust antecedents competence and reliability.

Further, we considered the trust evolution throughout the experiment depending on the proactive dialogue actions (see Fig. 6.11). According to Glikson and Woolley (2020), initial trust starts at a low level and builds over time during interaction with robotic AI. In this regard, we found that the *Reactive*-, and *Notification* strategy showed a tendency but only the *Suggestion* strategy significantly increased the perceived user trust. Thus, these trust trajectories seem to be in line with related work. However, for the *Intervention* strategy a trust decrease was found. This is a strong indicator that the usage of highly autonomous system behaviour needs to be carefully considered when designing proactive systems.

Influence of the Trigger Mechanism on Cooperation

Considering this research question, we found no significant differences for the low- to medium-levels of proactive dialogue concerning the timing strategies. However, high system proactivity showed to have significant negative effects on the CA's perceived understandability and usability when triggered after the detection of user frustration or confusion. An explanation for this could be those study participants were focused on resolving their state of frustration or confusion and considered the system interruption as disruptive or even obstructive. Hence, they could not understand the system's trigger mechanism. Also, a decrease in usability was measured which showed that the wrongful decision also negatively influenced the task effectiveness (see Table 6.3). Baker et al. (2010) stated that frustration and confusion may be a natural aspect of the experience of learning when dealing with difficult material. Furthermore, Graesser et al. (2007) noted that confusion, although being considered a negative state, positively contributes to a user's learning experience. Hence, it could be argued that triggering highly intrusive proactive dialogue during these states needs to be avoided or additional information, e.g context or user features, is necessary for initiation. For these reasons, we concluded that acting upon recognised user confusion or frustration seems to be not a reliable trigger mechanism for determining the need for proactive dialogue behaviour. Especially, since the Affectiva-trigger generally decreased faith in the system and perceived competence of the system.

Limitations

This study also had a few limitations: Like in the previous experiment, we applied a Nao robot for interaction in a controlled environment. Thus, the same limitations as described in the previous experiment can be noted. Furthermore, some challenges that Li and Ji (2005) mentioned could be contemplated when considering cognitive-affective states. The first of the challenges is related to the sensory observations often being imprecise and uncertain. To reduce this imprecision, different modalities are suggested to be considered. This was not done in the current study to create a more natural interaction and avoid being intrusive. Subsequently, for future studies, multi-modal affect detection in non-intrusive ways could be considered.

6.2.8. Conclusion

In summary, we did not find evidence for the usefulness of considering negative cognitive-affective user states as a criterion for initiating proactive assistance behaviour. Therefore, we propose to focus more on the individuality of the users. Modeling the idiosyncrasy of each user would lead to personalised systems by using models that can adapt to the user's characteristics. One example of this is the use of predictive models that consider the user's characteristics (Bull and Kay, 2010). Depending on specific user characteristics, different proactive actions could be more beneficial for creating a trustworthy learning experience. Furthermore, it could be useful to consider more states than confusion and frustration for triggering proactive behaviour. In this experiment, we assumed negative user states to be relevant for triggering proactive assistance. As this assumption did not turn out to be successful, including other states could be more reasonable for assisting. According to D'Mello and Graesser (2011), users cycle through different cognitive states during learning. Hence, a more dynamic approach may be more effective for determining the need for proactive dialogue. Further, we found that triggering an intervening dialogue during the experience of confusion or frustration, was not perceived as trustworthy behaviour. However, this positively led to a learning gain as users might have tended to reflect more on the system's decision. For this reason, it may be beneficial to look more into the relations between trust in a tutoring system and the user's learning gain. Similar to the first study, however, the experiment showed that highly proactive dialogue behaviour was generally less trusted than more reactive behaviour, even though it contributed the best regarding usability. In the following, we shift our focus on implementing proactive behaviour into more sophisticated prototypes for realistic tasks. In doing so, we wanted to reduce the limitations of our previously described study setups and investigate the portability of our model. Realistic task scenarios are less controllable than restricted lab experiments. Thus, they pose more challenges due to more diverse interaction scenarios resulting in unpredictable user behaviour and higher system requirements. For evaluating proactive dialogue regarding the effect on the human-machine cooperation in realistic task scenarios, we developed a virtual and a robotic CA relying on fully modelled cognitive capacities including planning and reasoning.

6.3. Effects of Proactive Dialogue Strategies Dependent on the User Activity

6.3.1. Motivation

In this experiment, proactive dialogue behaviour was embedded in virtual CA combining sophisticated planning, reasoning, and dialogue capabilities. The CA was applied in a complex task scenario, where users interacted with the assistant for receiving guidance for the execution of a real DIY-task using electric tools. An initial, non-proactive version of the CA was developed in collaboration with the use case partner Robert Bosch GmbH during a nationally-funded project by the German Research Foundation. For testing its applicability as virtual CA for helping with DIY tasks, an initial study was conducted. Due to the limited scope of this thesis, we briefly summarise the results of the study. For more information, we refer the reader to the works of Schiller et al. (2017a) and Behnke et al. (2019c). For evaluation, we tested the system against a baseline version of the assistant without any interactive features, i.e. voice commands and question answering were not supported. The baseline provided only static, pre-defined instructions in the form of text and images. The instructions were presented as a slide mimicking the state-of-the-art of current online guides for DIY. During the initial study, the HCT relationship was used as the main evaluation criterion. As the DIY-domain is a complex and delicate application domain, the user is required to accept the system as a trustworthy partner

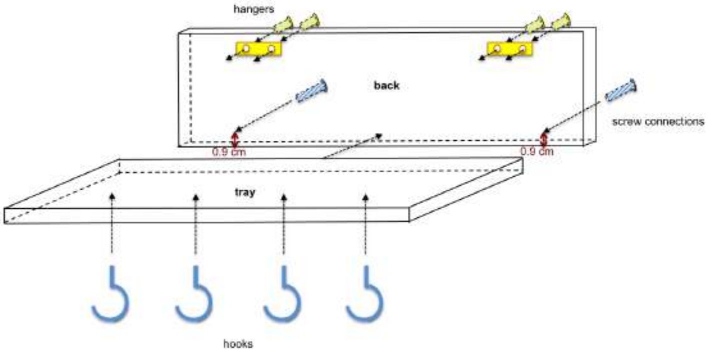
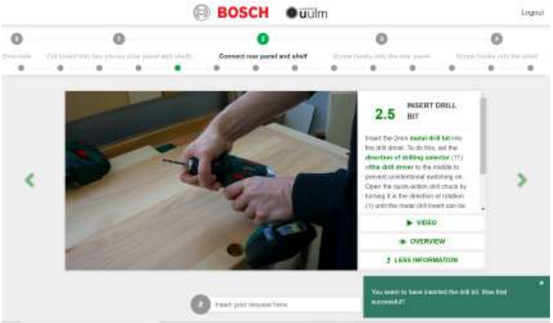


Figure 6.12.: The task of the use case scenario was to build a wooden key rack.

for following its instructions. In this regard, misguidance could even lead to harmful consequences. The results of the initial study showed that the virtual CA received a higher rating of trustworthiness than the baseline version and also significantly reduced the duration for setting up the tools for deployment (Bercher et al., 2021). Thus, the study outcomes validated the usefulness of the virtual CA for application in this task domain.

In this section, we provide an investigation of the inclusion of a medium level of proactive dialogue into the virtual CA by applying *Notification* actions. As a trigger mechanism, we utilised a user activity tracking method. For this, useful intervention points during the plan-based dialogue were identified by tracking the user's progress and current activity. The user's activity was tracked using a connected electric drill, which was developed at the Robert Bosch Company. Based on movement data of the electric drill a classification algorithm was trained that could predict specific actions the user was presently executing using the device. The primary goal of the investigation was to study the impact of proactive dialogue on cooperation depending on the user's activity. Further, it was observed whether the results of the previously described studies using low-fidelity prototypes were transferable and reproducible in realistic task scenarios.

In the following, we describe the scenario in detail. Further, we outline the system components and their interplay. Subsequently, we explain the design of the proactive dialogue strategy, describe the study setup, and present the results. Afterward, the results are discussed concerning our research questions.


BOSCH
uulm
Logout
Overview
Cut board into two pieces (rear panel and shelf)
Connect rear panel and shelf
Screw hooks into the rear panel
Screw hooks into the shelf
2.5
INSERT DRILL BIT
VIDEO
OVERVIEW
LESS INFORMATION
Insert your request here

Insert the 2mm metal drill bit into the drill driver. To do this, set the direction of drilling selector (1) of the drill driver to the middle to prevent unintentional switching on. Open the quick-action drill chuck by turning it in the direction of rotation (1) until the metal drill insert can be

You seem to have inserted the drill bit. Was that successful?

Figure 6.13.: Screenshot of the interface (Kraus et al., 2020b)

6.3.2. Scenario

In the DIY-domain, the virtual CA was intended to assist novice users in the performance of home improvement projects that require knowledge of the use of power tools (electric drills, saws, etc.). For this reason, the system provided support for the user in the form of an instructional dialogue and could provide further background knowledge about materials and tools on request. For a given DIY project, the CA provided its user with step-by-step instructions on how to complete the task successfully. These steps were generated by a planner and adapted to the specific user situation. The assistant presented the individual instructions using text, images, voice, and videos (see Fig. 6.13), which were selected automatically using description logic reasoning. The instructive dialogues were intended to give the user experience in the usage of individual power tools and encourage them to employ such tools in different projects. For demonstrating and evaluating the assistant's capabilities, the specific use case scenario was the construction of a key rack from a wooden plank (as shown in Fig. 6.12) using an electric drill driver and an electric jigsaw while being supported by the artificial assistant. As we limited the number of tools that may be used for this task, the planning module of the assistant created the following abstract plan, i.e. a sequence of actions, for building the key rack: sawing a plank into two boards, connecting the boards, attaching two hangers to the back, and adding four hooks to the tray. Each step of the abstract plan was composited from several sub-tasks. For example, the first task 'sawing a plank into two boards' comprised six steps: 'attach the saw's top unit', 'attach the saw's battery', 'mark the cutting line on the board', 'fixate the board', 'saw the board into two pieces', and 'loosen the fixated board'. Overall, the project consisted of 33 sub-tasks. During the initial study, the virtual CA was named after the founder of the Robert Bosch Company. Thus, the assistant is referred to as Robert in the following.

6.3.3. Prototype Description

Robert comprised three components for providing suitable assistance to novice DIYers according to our previously described conceptualisation of a cognitive architecture for enabling proactive dialogue: *User Interface*, *Interaction*, and a *Domain Model*.

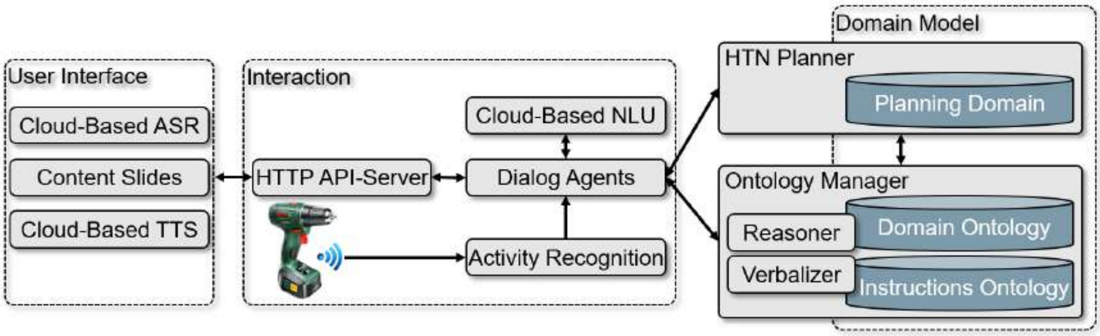


Figure 6.14.: Overview of Robert's architecture. A user interacted with the assistant's interface capable of multimodal input recognition. User input was forwarded to a server-based dialogue manager that mediated the interaction with the HTN planner and the ontology manager. In addition, the system was able to track the user's activity with a connected electric drill for proactive dialogue initiation. (Kraus et al., 2020b)

An overview of the workflow between these components is depicted in Fig. 6.14. All three components shared the same model information. However, each component only stored the information for handling the tasks for which it was best suited. When required, information was transmitted from one component to another. To allow for this interoperability and to ensure the coherent storage of models and information, a specific modelling paradigm was used. The paradigm allowed to store parts of the planning model in a structured way in the ontology (Schiller et al., 2017a). In the following, the three constituents are described.

Interface

Users interacted with Robert using a browser interface based on JavaScript. The interface was implemented as a multimodal GUI using the Vue.js [1] framework. It presented the generated plan in the form of step-by-step instructions. These were presented to the user as a sequence of slides, where each slide corresponds to one plan step. The content of one plan step was provided in the form of a textual and visual (picture, video-on-demand) task description. The interface was capable of processing multi-modal user input (speech, touch, text). Spoken language was transformed into text using Google Chrome's web speech API. For speech activation, a push-to-talk implementation was used. Spoken language system responses were provided to the user in the form of a pop-up modal containing a textual and/or visual description. The written descriptions were synthesised to speech using Chrome's Text-to-Speech API. Proactive messages were presented to the user in the same way. Each user input was forwarded to the modules handling the interaction using HTTP-methods (POST, GET, PUT) and JSON as data format.

[1] https://vuejs.org/

Interaction

For enabling interaction between the user and the assistant, an HTTP API-server served as a broker for exchanging information between the user interface and a modular DS. Using a modular architecture allowed us to easily extend the functionality of the system and maintain its components individually. The DS's purpose was to mediate the interaction with the ontology and the planner. For semantically encoding user input, we made use of statistically-driven approaches. In the first version of the prototype, Microsoft's cloud-based LUIS (Williams et al., 2015) was used. Generally, the natural language understanding approach of LUIS relies on the two trainable concepts of intent and entity. Intent refers to the intention of a user, i.e. the purpose of an utterance, whereas an entity contains meaningful parts of an utterance. For example, when a user communicates the following planning request: "I want to build a key rack" the Intent would be *startPlanning*, while "key rack" would be recognised as a value for the entity named project. For later iterations of the assistant, we switched to Rasa NLU, which used the same concepts, but was easier to maintain and modify due to its availability as an open-source project contrary to LUIS.

For DM, an agent-based approach was implemented (Rao et al., 1995), which is typical for plan-based DM. For each component of the CA there existed a dialogue agent that carried out module-specific tasks. For example, when the system's semantic encoding component recognised a plan-related intention from users, such as receiving instructions for a specified DIY project or the wish to modify the plan according to their preferences, the planner was invoked by its respective dialogue agent. The planner then generated a sequence of actions, i.e. the plan, providing appropriate instructions. The ontology was used to enrich the symbolic description of the actions with textual instructions and media contents for presentation to users.

The ontology-related agent was able to handle requests for information and explanation of specific materials and tools. The system's ontology was able to handle three different kinds of conceptual knowledge-based requests: encyclopedic requests, media requests, and availability requests. Encyclopedic requests concerned the appearance or the purpose of a material or tool, e.g. *"What is a drill-bit?"* or *"What does a drill-bit look like?"*. For receiving an image or a video of a DIY item, a media request was used, e.g. *"Can you show me a video of how to use a drill-bit?"*. To check the availability of certain items, and availability request could be posed, e.g. *"Is there a drill-bit available?"*.

The dialogue-related agent was responsible for meta-dialogues (information confirmation/grounding, proactive behaviour) and for keeping track of the project and dialogue state, stored in an interaction state S. This state S contained information about the project step (*Action*) the user was currently working on, as well as past user input (*UserAct*), and system output (*SystemAct*). The system used this information for adequate response generation. System responses were semantically decoded, i.e. transformed into text, using a combination of templates and using the verbalisation tool developed by Schiller et al. (2017b).

Domain Model

Robert proposed a course of action to its user that if performed would complete the given DIY project. Robert's planning component was responsible for determining this course of action – called a *plan*. For determining a suitable plan, the planner utilised a general description of the DIY setting and the user's project in terms of a *planning model*. This model encompassed formal descriptions of the available tools and materials as well as the actions that could be used to manipulate the environment in a DIY setting, e.g. sawing, drilling, and fixating. The model itself did not pertain to the characteristics of a single (or some) particular problems or projects, but instead represented a general description of the possible activities that could be performed in a DIY setting. This generality allowed Robert's planner to flexibly adapt its plan to the current situation and project of the user. For example, it could come up with other means of making a large hole, if no Forstner bit was available. For formalising the model, we applied the concept of HTN planning (Bercher et al., 2019).

Using HTNs enabled the suitable combination of the planning model and the ontology (Behnke et al., 2015; Schiller et al., 2017a). Hence, information was stored only once and could be handled by a suitable component. Lastly, the hierarchical nature of the description allowed Robert to provide abstract instructions in addition to detailed instructions. This was useful for the case that a user was already familiar with some procedures in the DIY setting (e.g. pre-drilling) and thus did not need to be instructed on how to perform them again. Robert used a SAT-based planner to find optimal (shortest) plans (Behnke et al., 2019b,a, 2018b,a) . More information regarding the planning framework used for the implementation of the here described virtual CA can be found in Behnke et al. (2019c, 2020).

The ontology manager organised Robert's static knowledge specific to the DIY domain. DIY tools and objects (e.g. drills, bits, saw blades, . . .) were organised in an ontology and characterised by properties such as colour, shape, but also technical parameters (e.g. battery voltage) and functionalities (e.g. that a drill driver can serve both as a drill and as a screwdriver).

The ontology also stored suitable configurations for instantiating actions in the domain, e.g. the recommended speed settings for drilling in wood. The ontology's DIY domain model was provided both to the planner and the DM. In addition, the ontology manager organised the instruction elements (texts, images, videos) from which the step-by-step instructions were to be assembled. The instructions were based on the actions and parameters instantiated by planning. For this, logical reasoning (classification) in the ontology was used. The ontology manager was also queried when Robert answered factual questions from the user. In this case, text was generated from the stored descriptions, and media was retrieved. A more thorough overview of the functionalities of the ontology can be found, for example, in Behnke et al. (2019c).

6.3.4. Design of Proactive Dialogue Strategies

According to our conceptualisation of modelling proactive dialogue, we implemented the proactive strategies for appropriately initiating sub-dialogues during the instructional dialogue provided by the CA. We implemented the medium-level of proactive dialogue in the form of a *Notification* action. This level was selected heuristically for being the most appropriate for the scenario and the purpose of this experiment. We developed two proactive dialogue strategies for the updated version of the DIY assistant: a *progress-based* and an *activity-based* strategy.

Following a *progress-based* strategy, proactive system behaviour was triggered at the end of important sub-tasks. These moments marked an important intermediate step during task execution, enabling the user to pause, reflect, and prepare for the next sub-task. For example, "cutting the wooden board into two pieces (rear and shelf)" was the first main task of building the key rack in the evaluation scenario. As the last step of this task, the user was instructed to loosen the screw clamps that are used for fixating the wooden board. There, a reflective meta-dialogue was initiated after the user had spent ten seconds on the instructional slide. The intent of reflective dialogue was for users to understand the processing of the project and to actively inquire about their actions. This was aimed to enrich the users' experiences and to learn (Frølund and Nielsen, 2009). Reflection was shown to have a positive impact on learning (Katz et al., 2007). The timing was chosen heuristically based on the average reading duration of the instruction. An example dialogue of this interaction scenario between the system (**S**) and its user (**U**) is provided in the following:

S: The first step to building a key rack has been taken. Are you satisfied so far?
U: Yes, everything went fine.
S: What did you like about this craft step?
U: I enjoyed using the electric saw. Never thought I could handle it so easily.
S: How did you like the instruction up to here?
U: Nothing to complain right now.
S: Thanks for your response. Would you like to know the steps for the next sub-task?
U: Yes! Thanks!

As can be seen in the example, each reflective meta-dialogue consisted of a maximum of four turns: Questioning the overall satisfaction, asking about an appraisal of the finished craft step and the instruction, and proposing the initiation of the next project step. The interaction followed a static dialogue flow, where the NLU was only applied for confirmations. Here, the system expected affirmations or negations from the user and made use of reprompts otherwise. For the appraisal questions, users could provide arbitrary answers. Questioning the overall satisfaction was intended to boost self-awareness about the finished sub-task. In combination with the specific question about the appraisal of the completed craft step, this was intended to animate the user to appreciate the made progress. Self-appraisal showed positive results on users of tutoring systems regarding learning and task performance (Katz et al., 2000). In that sense, it was supposed to positively affect the perceived assistance of the CA. Questioning the appraisal of the provided instructions was intended to foster the perceived competence and reliability of the system concerning its helpful guidance. This was considered to manifest trust in the assistant as observed in our previous studies regarding the impact of proactive system actions. Afterward, the user was proposed to move on to the next task which then finished the reflective meta-dialogue. In summary, the reflective meta-dialogue served the purpose to get users to talk to the system to establish a trust relationship by providing a careful and competent system appearance. Additionally, it was supposed to reinforce the users' thinking about their capabilities and positively contribute to their task success with the system. It was possible to ignore the assistant's proactive behaviour or to quit the dialogue at each step to continue with the project. The wording of the system's utterances was alternated for each sub-task to increase naturalness. Following an *activity-based* strategy, context-dependent proactive behaviour was triggered using information from a connected drill serving as an external sensor of user behaviour. To proactively assist, sensor data for tracking the user's current activity was used, e.g. drilling or screwing. To collect sensor data, an inertial measurement unit was integrated into a standard cordless drill driver and connected to a Wi-Fi development board. In doing so, gyroscopic, accelerometric, and compass data could be transmitted from the device. Activity classification was provided by a neural network trained with data from 12 participants. This data was collected in a separate experiment. A deep neural network approach was preferred since it is considered state-of-the-art in human activity recognition and yields good results for classifying movement patterns based on sequences of raw sensor inputs (e.g. see Ordóñez and Roggen (2016)).

Classification served to distinguish the following classes of (in-)activity: *off* (machine not moved), *screwing*, *drilling*, *drill change*, *battery change*, *in use* (machine is moved, motor is off), and *other.* Average accuracy of activity classification of better than 0.9 was achieved (in 4-fold cross-validation) in the classifier-training experiment. Additional information was made available in the form of probability distributions, e.g. of the current activity, the activity's operation time and frequency of its occurrence. The CA received new information about the user's activities every 500 ms via TCP-socket connection. For leveraging this information during the interaction, the concept of a machine state as a part of the interaction state S was introduced. While the interaction state stored information about the user's (spoken) input and the dialogue history, the machine state allowed to trigger proactive system actions based on implicit knowledge about the user's tool activity. It consisted of so-called *MachineAct*s. An act had information about the current activity, its operation time, and how often this act had occurred before. The machine state was updated each time a different user activity was tracked with high confidence, i.e. the activity's probability had to be higher than 0.95.

Activity-based proactive system behaviour addressed the active initiation of a reflective meta-dialogue with users to check whether they were performing the project's steps correctly and to provide help in the case of failure. In the initial study evaluating this specific DIY assistant, we discovered that users favoured videos for visualising the task descriptions. For example, subjects explicitly mentioned the "helpful video instructions" or that "... videos have been very helpful". Therefore, the primary content of the proactive dialogue was to offer help in the form of instructional dialogues.

Here, a rule-based decision model was implemented. At each project step, where the connected tool was required to be applied, the CA was able to actively start a dialogue with its user. Depending on the context it was differentiated between three different interaction scenarios: A message was triggered after the user had picked up the machine (*in use*-classification): *"You seem to be working with the connected tool for the first time. Don't worry, I'll guide you through the steps!"* This message was used to foster transparency of the activation of the connected tool and could be triggered only once. To react to possible user insecurity about the current project step, a help request was sent to the user after three minutes of inactivity (*off*-classification). This kind of request was only executed during steps where actions with the connected tool have to be performed: *"I haven't seen any tool activity by you in three minutes. Do you need help?"* In case the user confirmed this question, they were invited to watch a video of the current step.

Finally, the assistant checked whether the user performed the instructed task correctly. An interaction was initiated whenever one of the activities *battery change*, *drilling*, *drill change*, or *screwing* was recognised. The user was then asked whether the current step was going successfully. If the user responded with *no*, further help was offered. After users accepted the help, the system started an explanatory video of the task. Otherwise, the system apologised for the disturbance.

S: I noticed that you were drilling. Was that successful?
U: No, it wasn't.
S: Ok, do you need additional help?
U: Yes, please.
S: A video of this project step could help. I'm going to play it for you. *(Then a video is played.)*

By integrating the previously described proactive behaviour into the virtual CA we aimed to investigate the impact of the designed strategies on the cooperation focusing on the assistant's perceived trustworthiness and its usability. In the following, we describe the experimental design for studying these research questions.

6.3.5. Experimental Design

For comparing the effect of the proactive dialogue in real use case scenario, a between-subject A/B-test was conducted. Here, the proactivity of the CA was used as an independent variable, where an assistant capable of a medium-level of proactive dialogue was compared to a baseline variant without any proactive functionalities. Participants were randomly distributed to both study groups. Dependent variables were selected similarly to the previously described experiments and primarily focused on the HCT relationship and usability.

Participants

33 German subjects were recruited by a professional institute for study consulting MTO. They were evenly assigned to each study group regarding gender and age. A criterion for study participation was to be a novice in DIY projects. However, three subjects had

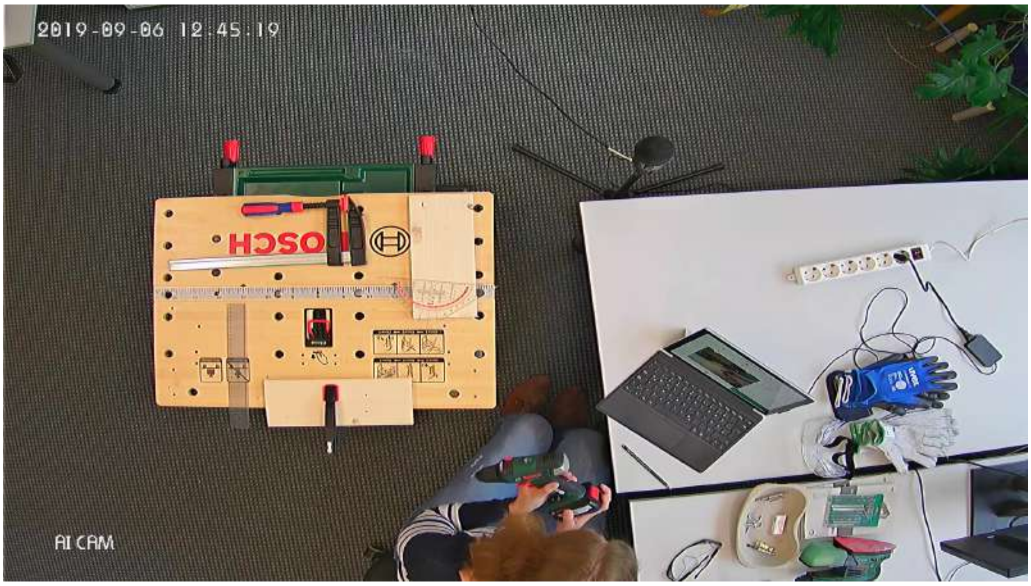


Figure 6.15.: A user attaches the battery of the electric drill driver while being assisted by the CA (Kraus et al., 2020b)

to be excluded from the evaluation, as they were rated as "quite experienced" by study observers in hindsight. In addition, two participants had to be excluded because they did not work according to the study plan. One participant had to be excluded due to a malfunction of the system, and one participant aborted the project. This resulted in data from 26 subjects being considered for evaluation. The average age of subjects was 37.54 ($SD = 14.72$). Study participants had various professional and educational backgrounds. The group size of subjects working with the proactive assistant was 12 (5 male, 7 female), while 14 subjects (5 m, 9 f) worked with the baseline version of the assistant. All subjects received 50 € independently of the study outcome.

Experimental Procedure

After the welcome procedure, the subjects were provided with first instructions and details of the study. They had to read and sign the informed consent and fill out a pre-test questionnaire concerning demographics and possible confounding variables as in previously described experiments. Afterward, they received an interactive tutorial to get familiar with the standard functionalities of the virtual CA, e.g. how to activate the speech recognition, and how to navigate through the user interface. Subsequently, they were asked to build a key rack from a wooden board in cooperation with the DIY assistant. For the duration of the construction, an experiment facilitator was in the same room as the participant for observation but was not allowed to assist. The study was captured on video and audio and streamed to a separate room.

There, study observers took notes about specific events and participant features, e.g. their subjective assessment of participants' DIY-experience levels. After completion, they had to fill in a questionnaire to assess the dependent variables. The total duration of the experiment was between 1.5-2.5 hours. The study setup is depicted in Fig. 6.15.

Questionnaires

In the experiment, the system's perceived trust, acceptance, and user experience with the system were assessed as dependent variables. Regarding trust and acceptance, we used the same questionnaires as for the studies using low-fidelity prototypes. Contrary to these studies, we omitted the UEQ-questionnaire and opted to measure the speech capabilities of the system of the DIY assistant. This was due to the less restricted interaction as opposed to communication with the Nao robot. For this, the SASSI questionnaire (Hone and Graham, 2000) was employed. All scales were translated into German and slightly modified for the study context as in previous studies. All scales were assessed with a 7-point (except where noted with a 5-point) Likert scales ranging from 1 = "totally disagree" to 7 = "totally agree" except for the acceptance assessment which used contrary adjective pairs on a 7-point Likert scale.

6.3.6. Results

For data analysis, a Mann-Whitney-U-Test was calculated to determine whether there were differences in confounding as well as dependent variables. A non-parametric test was chosen, as a rather small sample size was examined. Additionally, a Pearson-Chi-Square test was used for the confounding variables, previous experience in DIY-tools (drilling, sawing,...) and in speech assistants (Siri, Alexa, Google Assistant, ...). These were measured on nominal scales. Confounding group variables for proactive behaviour could be ruled out, as measurements for predisposed trust in autonomous systems (Merritt et al., 2013), technical affinity (Karrer et al., 2009), previous experience in DIY-tools and speech assistants were not significant (all p-values > 0.05). Solely, the Chi-Square test for previous experience with Alexa ($\chi^2 = 7.22,\ p < 0.01$) became significant. However, experience with similar assistants was evenly distributed, hence this result was negligible.

In addition, participants' age and gender were similarly distributed for the different experimental groups and no outliers were found in the data set. An overview of the results is presented in Table 6.4.

User Experience with the Experimental Setup

Overall, the virtual CA received positive reviews regarding its user experience. Users rated the system to be likable ($M = 5.47,\ SD = 1.27$), not cognitive demanding ($M = 5.34,\ SD = 1.01$), and not annoying ($M = 4.70,\ SD = 1.06$). Further, users could learn the handling of the interface easily ($M = 5.03,\ SD = 1.38$) and the system reacted in a fast manner ($M = 5.71,\ SD = 1.25$). However, the virtual CA was only moderately accepted ($M = 3.96,\ SD = .78$).

	Dialogue Strategy					
	Proactive *M (SD)*			**Baseline** *M (SD)*		
	male	female	overall	male	female	overall
Trust*	4.49 (.33)	4.55 (.54)	4.52 (.45)	4.11 (1.01)	4.22 (.58)	4.18 (.72)
Acceptance*	3.71 (.94)	4.44 (.60)	4.14 (.81)	3.67 (.83)	3.88 (.73)	3.80 (.74)
Reliability	4.68 (1.65)	5.60 (.98)	5.22 (1.32)	4.76 (1.82)	5.18 (.82)	5.03 (1.22)
Competence	4.88 (1.40)	5.46 (1.14)	5.22 (1.23)	4.72 (2.11)	4.80 (.81)	4.77 (1.33)
Predictability	5.90 (1.21)	6.07 (1.05)	6.00 (1.07)	5.40 (1.42)	6.17 (.63)	5.89 (1.00)
Personal Attachment	2.28 (1.60)	3.23 (1.60)	2.83 (1.60)	4.56 (2.20)	2.93 (1.20)	3.51 (1.74)
Faith	3.36 (2.17)	4.49 (1.17)	4.02 (1.67)	4.40 (1.64)	4.82 (1.01)	4.67 (1.22)
Satisfaction	4.93 (1.50)	6.32 (.45)	5.74 (1.20)	5.44 (1.84)	5.14 (1.04)	5.25 (1.32)
Cognitive Demand	4.96 (.90)	5.46 (.87)	5.25 (.88)	5.68 (1.21)	5.27 (1.13)	5.41 (1.13)
Annoyance	4.04 (.68)	5.20 (1.25)	4.72 (1.17)	4.92 (.93)	4.56 (1.08)	4.69 (1.01)
Habitability	5.00 (.77)	5.21 (1.64)	5.13 (1.30)	5.00 (1.76)	4.92 (1.41)	4.95 (1.47)
Speed	5.80 (1.10)	6.29 (1.29)	6.08 (1.18)	5.60 (1.52)	5.28 (1.18)	5.39 (1.26)

Table 6.4.: Descriptive statistics of the measured dependent variables with reference to the dialogue strategies. Results for cognitive demand and annoyance are inverted (the higher, the better). *Trust and acceptance measured on 5-point Likert scales. (Kraus et al., 2020b)

Significant gender differences regarding experience in DIY-tools and speech assistants were found. Chi-Squared tests revealed that females had substantially less experience in the usage of two out of four questioned electric tools (percussion drill, $\chi^2 = 4.63,\ p = 0.03$; electric jigsaw, $\chi^2 = 5.10,\ p = 0.02$) than men. Besides, females had less experience in two out of three speech assistants (Google Assistant, $\chi^2 = 6.52,\ p = 0.01$; ALEXA, $\chi^2 = 4.51,\ p = 0.03$). Therefore, gender-based differences regarding proactive meta-dialogue were considered. Females rated the proactive meta-dialogue with the CA significantly higher than interacting with the baseline variant for the categories satisfaction ($U = 8.00,\ Z = -2.50,\ p = 0.01$) and speed ($U = 13.00,\ Z = -2.00,\ p = 0.046$). Furthermore, there was a notable higher rating for acceptance of proactive behaviour by females ($U = 14.50,\ Z = -1.81,\ p = 0.07$).

Effects of Proactive Dialogue Strategies on Usability

The users interacting with the proactive version of ROBERT took an average of $M = 54.67,\ SD = 21.13$ minutes until completion of the project. In contrast, users interacting with the non-proactive version took an average of $M = 56.93,\ SD = 12.78$ minutes until completion. The difference, however, was not significant ($p = 0.560$). Further, we asked the study participants to rate the difficulty of the task before and after the experiment and to rate the quality of their own performance on 5-point Likert scales. Here, we found that users rated their performance higher when using the proactive assistant ($M = 3.67,\ SD = .89$) as compared to the non-reactive version ($M = 3.29,\ SD = 1.07$). Further, users found the task easier after their performance when interacting with proactive CA ($M = -0.17,\ SD = .71$). Contrary, they found the task more difficult after the study when using the non-proactive CA ($M = 0.43,\ SD = 1.02$). None of these differences were significant (all $p >> 0.05$).

Effects of Proactive Dialogue Strategies on Trust

There were no significant between-subject differences for the dependent variables. However, a significant within-subject difference was found regarding the development of trust towards the CA. Therefore, the difference between trust ratings before and after completion of the project was assessed using a Wilcoxon-Signed-Ranks test (initial trust was measured with a propensity towards trust in autonomous systems questionnaire). In comparison to the baseline group, which showed no difference ($Z = -0.39,\ p = 0.71$), the group with the proactive assistant showed a significant higher trust-building ($Z = -2.58,\ p < 0.01$). The results are visualised in Fig. 6.16. Further, there was a visible trend that males evaluated the personal attachment towards the baseline assistant higher than the proactive version of the assistant ($U = 4.00,\ Z = -1.49,\ p = 0.07$). Regarding the age of the participants, no significant differences were found. In general, the proactive behaviour was rated as reliable, competent, and predictable. However, the affect-based trust of the proactive assistant was perceived as rather low. Thus, the overall trust in proactive system behaviour was mediocre. The means and standard deviations can be found in Table 6.4.

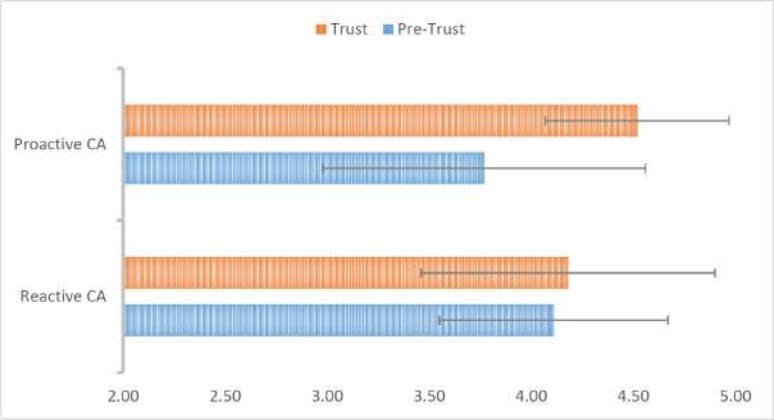


Figure 6.16.: Depiction of the results for the development of trust towards the CA depending on the study condition. Mean values and standard deviations from the user ratings on a 5-point Likert scale are provided. (Kraus et al., 2020b)

6.3.7. Discussion

The study revealed differences between the proactive and the baseline condition. In the following, we discuss the results with a focus on the formulated research questions.

Influence of Proactive Dialogue Level on Trust

It could be shown that the integration of proactive dialogue led to a significantly better establishment of trust between the user and CA as compared to the non-proactive baseline. This result fostered our observations from our previous study on proactivity and validated the application of developed proactive actions in realistic use cases. The specific DIY use case provided a quite intuitive result, as a system that actively engages with the user and tries to participate in the project should be perceived as a more trustworthy assistant. Especially for novices, to which the entry to a new topic can be quite challenging, a more natural and social assistant than a rigid, non-communicative system seems to be beneficial. This manifests our previously discovered effects of proactive dialogue actions on less experienced users.

Even though there were no significant group differences found regarding the overall measurements of trust and its components, a trend can be observed that timely proactivity enhances trust towards assistants. Means for overall trust and cognitive-based trust (competence, reliability, and predictability) were all higher for the proactive condition than for the baseline condition. This is in line with our previous findings of low- to medium-level proactivity to positively affect a system's perceived competence and reliability. For affect-based trust an opposite tendency was observable. Both variables, personal attachment, and faith were rated higher for the baseline condition. This may be explained by noting that users were more familiar with reactive systems. However, the overall score of these variables was rather low for both conditions. For fostering a better affect-based trust, a long-term relationship with the user may be required. Furthermore, the standard deviations were quite high for the affect-based variables. This could be a sign that proactive behaviour has a very individual impact on the user's affect towards assistants.

Furthermore, gender-dependent effects of proactive dialogue were found similar to the study presented in Section 6.1. There was a significant difference between proactive and the baseline condition for satisfaction and speech response accuracy rated by females. Additionally, females tended to accept the proactive assistant more. These gender effects seemed to be due to differences in the experience in DIY-tools and speech assistants between males and females. The more experienced males showed the tendency to prefer working with a non-proactive assistant, as can be seen in the ratings of reliability, personal attachment, satisfaction, and annoyance. This could be explained that men might have felt patronised by the active assistance and found that the help offered was unnecessary, as they could work independently based on their tool knowledge. Contrarily, females, being less experienced, tended to welcome the more communicative guidance by the assistant, as they required a higher level of cooperation.

Overall, the results of this study verified that the developed proactive dialogue model is transferable into realistic use scenarios and can be used for sophisticated dialogue applications. In line with the previous studies, this experiment showed the benefits of a medium-level of proactive system behaviour by improving the perceived reliability and competence. We found that this was especially the case when interacting with novice users, as an active system seemed to reduce the barrier of interacting with a machine.

Influence of Proactive Dialogue Level on Usability

A positive trend of proactive dialogue in CAs was also noticeable considering the usability of the system through the measurements for acceptance and user experience. Means for acceptance, satisfaction and habitability were higher for the proactive condition. This is similar to the results of a Wizard-of-Oz study reported by Peng et al. (2019), where a medium-proactive robot was perceived as more appropriate and helpful in decision-making tasks. However, the differences found in our work were not significant. Nonetheless, the effect of proactive dialogue strategies on the user's acceptance and satisfaction should be further examined, as proactive behaviour showed much potential. Interestingly, there was no difference between the two conditions regarding the rating of annoyance, as proactive intervention can be perceived as obtrusive. Therefore, the possibility to ignore the system-initiated dialogues, i.e. to let the user have control over the interaction, was appropriate in this scenario. As proactive dialogue requires the attention of the user, it was no surprise that the non-proactive variant of the CA was rated better regarding cognitive demand. Another measure that contributed to usability was naturally the subject's task performance of building a wooden key rack.

Although the results considering the user's performance were not significant, we could observe some tendencies. First, the task duration while interacting with the virtual CA capable of proactive dialogue was roundabout two minutes shorter than task completion with the non-proactive assistant.

This can be seen as a positive result, as proactive dialogue usually leads to longer interactions than reactive dialogue, as the system initiates more sub-dialogues. Further, proactive dialogue led to better ratings of the users' performances and even let the task appear easier. Thus, we assumed that proactive dialogue can indeed increase task performance. However, as the differences were not significant in this study, this assumption needs to be verified in future research.

Influence of the Trigger Mechanism on Cooperation

Generally, relying on an activity-based trigger mechanism seemed to positively contribute to the perception of the proactive dialogue during cooperation. For explanation, consider the results depicted in Table 6.4. The best metrics for observing the effect of the trigger mechanism on the cooperation are predictability, cognitive demand, the system's speed, and annoyance. Considering predictability, there was no significant difference between the proactive and baseline version of ROBERT. Consequently, the proactive dialogue seemed to be initiated timely and did not appear to be unpredictable, but at defining moments. In addition, triggering proactive dialogue actions were not perceived as significantly more annoying and did not leverage a user's cognitive load more than reactive behaviour. Hence, we concluded that the utilised trigger mechanism did not lead to obtrusive behaviour and also did not distract the user from their main task of building the key rack. Further, the proactive system seemed to be perceived to accelerate the interaction as measured by the system's subjective speed measurement. Therefore, including contextual features for designing trigger mechanisms for initiating proactive dialogue may enhance the perception of a system's usability, and thus improve the cooperation with the system.

Limitations

Comparing the strengths and weaknesses of the study design, the advantages of the setup were a highly realistic test scenario with a sophisticated CA. A disadvantage was the quite low number of participants. Using a higher number could have provided more significant results between the two conditions. Besides, the quality of the speech synthesis was troublesome as several participants of the study reported the synthetic voice to be unnatural and even annoying. Another problem was that speech had to be activated by "push-to-talk" for technical reasons. Even though users were instructed to use this kind of voice activation in the tutorial, they found this to be cumbersome and annoying.

6.3.8. Conclusion

Comparing two versions of a high-fidelity prototype, one capable of providing medium-level proactive dialogue and one reactive version, showed that the developed proactive dialogue concept is transferable to realistic task scenarios. Here, we observed similar positive effects of *Notification* actions as in our previously described studies. Especially, the perceived competence and reliability of the virtual CA could be increased by including proactive dialogue behaviour.

This effect was also primarily observable in the studies with low-fidelity prototypes. Additionally, also gender-differences concerning the perception of proactive behaviour were detected again. By observing the results, however, it seemed that the reason for this difference again was largely due to differences in domain experience and technical affinity between the genders. These seemed to validate our previous findings that domain experience and technical affinity may be a driving factor in the user's trusting behaviour towards proactive dialogue. Domain experience was also considered a trust-influencing feature concerning autonomous systems (Schaefer et al., 2016). Further, the inclusion of proactive dialogue also led to better user performances and thus usability, even though the differences were only marginal due to the small sample size. Also, the application of the user's progress and activity as a trigger mechanism seemed to be beneficial, as they had no negative effect on the cooperation. Thus, we propose to not only rely on user state information for triggering proactive behaviour but also to focus on including context information.

So far, we considered the effects of virtual CA's proactive dialogue capabilities on the HCT relationship. Virtual CA's are limited in the sense that they cannot physically assist users in the execution of particular tasks. Robotic CA's in turn possess this capability. A robot's actuators can mimic human body motions by simulating muscles and joints. Thus, they can take over physical tasks from humans. Due to the higher degree of anthropomorphism of robotics in comparison to virtual CA, a robot's proactive behaviour might be perceived differently. This may especially hold for perceived trust in the system, as a robot's action may even result in physical harm or property damage. Therefore, we investigate the implementation of our developed proactive dialogue concepts into a robotic CA in the following.

6.4. Effects of Proactive Dialogue Strategies Dependent on External Events

6.4.1. Motivation

Similar to the previous experiment, we investigated the impact of a medium-level proactive dialogue on the cooperation with a high-fidelity CA prototype. However, we applied *Suggestion* actions in a robotic CA depending on external events for this experiment. This proactive dialogue act type was selected because we considered a more free-form conversation in this task scenario in which the interaction was not restricted to the manipulation of a user interface. Therefore, more direct suggestions should ensure a safer interaction. These proactive actions were triggered dependent on the user's context and external events. Thus, we focused more on an environment-based than a user-based trigger mechanism for initiating proactive dialogue in this experiment. Here, the main goal was to study the impact of event-based proactive dialogue strategies on cooperation. Similar to the previously described experiment, we also investigated whether the results of the lab studies using low-fidelity prototypes are transferable and reproducible for robotic CAs.

For this, we implemented a DS for allowing HRI that mediated the interaction between the user and the robot's cognitive modules. These cognitive modules were responsible for robotic action planning and reasoning. Further, a domain ontology contained knowledge about the robot's and user's shared environment. The robotic CA was deployed in a household-assistance task scenario. After having been accepted and applied in an industrial context for quite a time, robotic applications have gradually entered our homes in recent years. This fact is quite visible when observing the current unit sales of robots for domestic tasks (18.6 million) as well as entertainment and leisure (4.6 million) for 2019 (of Robotics, 2021). However, domestic robots are currently mostly used for simple and restricted tasks, such as vacuum cleaning or cutting grass. Due to advancements in AI and robotic technology, it seems plausible that domestic robots can assist with simple household tasks in the future and be able to communicate their actions using natural language. As this may result in a user expectation of proactive robotic CAs, we selected a domestic task domain for our experiments. Further, the domestic domain is a vulnerable user domain as the wrongful robot behaviour may be perceived as an invasion of privacy. Thus, trust is an important aspect of HRI in the domestic domain. The work described in the following was part of the project RobotKoop that was nationally funded by the German Federal Ministry of Education and Research.

In the following, we describe the scenario in detail. Further, we outline the system components and their interplay. Subsequently, we explain the design of the proactive dialogue strategy, describe the study setup, and present the results. Afterward, the results are discussed concerning our main research questions.

6.4.2. Scenario

As robot type, we used a Tiago robot from Pal Robotics[2] for this experiment. The Tiago robot contains various motors, sensors, and a gripper. This enabled the robot to assist in the household by executing give- and take-actions. Furthermore, it was implemented with the capability of anomaly detection. This implies, that it inspected its surroundings and was able to detect deviations from a learned standard (tidy) state. Therefore, it could detect whether its environment was in an untidy state and automatically started a clean-up. The robot used in the scope of this experiment was called Kurt. For evaluating the robot's proactive capabilities, we tested Kurt in a small user study. For conducting the study in a realistic setup, a lab environment was furnished with a couch, couch table, closet, and dining table. Furthermore, this "living room" was equipped with typical household items. In doing so, the environment was intended to simulate a typical living room in a domestic environment. In this environment, the user interacted with Kurt in three different scripted scenarios.

First, Kurt provided helpful information about its sensory system and functionalities through an introductory dialogue. The purpose was to get users with different degrees of experience with robots on the same level. This was intended to ensure the comparability of the results.

[2] https://pal-robotics.com

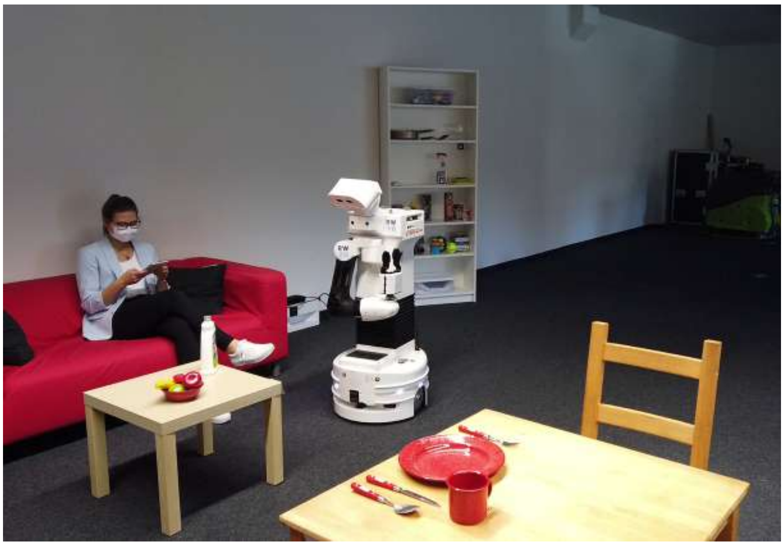


Figure 6.17.: Depiction of a study session with a subject. The user can choose between allowing the robot to remove the bottle of bleach or ignoring it. (Kraus et al., 2022c)

In the next step, KURT asked for permission to inspect the domestic environment and to memorise the position of the furniture and different categories of objects (e.g. kitchen utensils, fruits, cutlery). After inspection, the robot recognised an anomaly, i.e. a deviation from the original tidy state, which the robot had learned by observing its surroundings. According to our script, a bottle of bleach that had been placed on the couch table were such an anomaly. To resolve this "untidy" state, the robot proactively approached the user by suggesting to take the object back to the closet where the other detergents were stored. If the user affirmed the routine, KURT proceeded with the task. Otherwise, the robot left the scene untouched. A picture of this sequence is shown in Fig. 6.17.

The last scenario dealt with the system's ability to offer and negotiate alternatives. Here, the user initiated the interaction by asking for an item (*"Bring me some crackers."*). However, KURT did not find the requested item in its object database. Consequently, the robot returned to the user and announced that the object was not in stock (*"I am sorry, there are no crackers available."*). Subsequently, the assistant began to suggest alternatives based on a list of related items that had a sufficient conceptual overlap with the initially requested object in the robot's world model database, e.g. *"Would you like crisps instead?"*. If the user agreed, the robot proceeded with the routine and handed over the alternative. If rejected, the robot left the user alone. After this, the use case scenario ended.

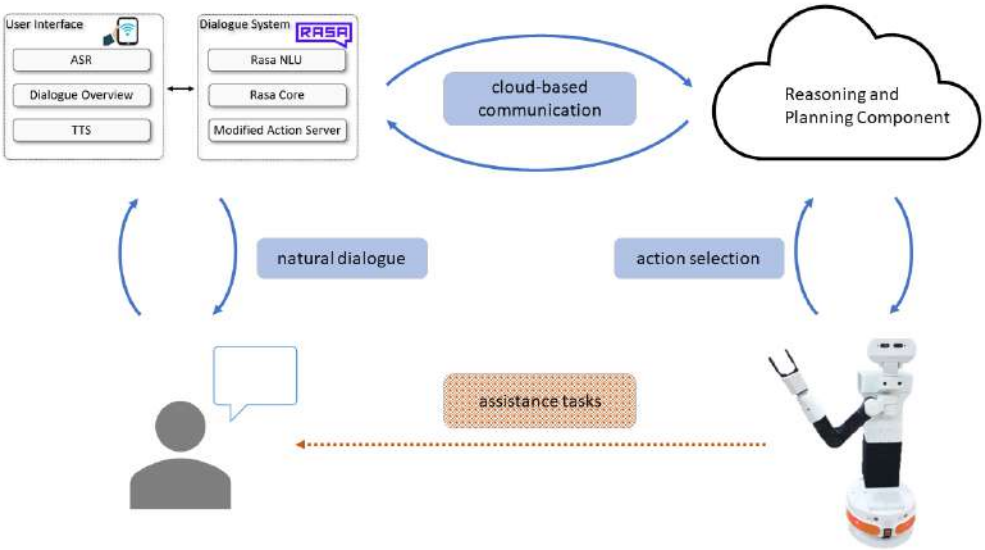


Figure 6.18.: Depiction of the overall architecture.

6.4.3. Prototype Description

For developing the robotic CA, we adhered to the cognitive architecture for enabling proactive dialogue. The main components of KURT were the physical device in the form of a TIAGO robot, the cognitive robot capabilities in the form of a planning and reasoning framework, and a spoken DS that was based on the RASA [3] dialogue framework. Generally, the TIAGO robot is a customisable robot with a configurable height of 110-145 cm and an arm payload of 3 kg (without end-effector), running on an Ubuntu operating system. Furthermore, the robot has an RGB depth camera and stereo microphone speakers. The model that was used in the experiment included a gripper arm for fetch-and-carry tasks. We developed an Android application as a user interface that allowed users to interact with the spoken DS. The overall architecture of the system is depicted in Fig. 6.18. In the following, we present the individual components.

User Interface

The robot's microphones and internal speech recognition modules were prone to communication errors due to background noise of robot movements and varying distances between user and robot during the experiment. Therefore, the user interacted with KURT through an Android-based user interface that could access Google's speech recognition and synthesis engines for handling spoken language. In doing so, a more stable speech recognition quality could be ensured. The user interface was implemented as an Android application using the KOTLIN programming language [4]. Besides handling the system's

[3] https://rasa.com/

[4] https://developer.android.com/kotlin

speech input and output capabilities, the interface could also handle written input. Further, the interface presented the dialogue history of the current conversation up to the current turn. This was intended to help users with their situational awareness, as task execution by the robot could take up to a few minutes. Information between the user interface and the interaction components was exchanged via HTTP messages using the JSON-data format. The interface continuously scanned for system actions to be able to handle proactive dialogue behaviour.

Interaction

The interaction components were based on the open-source ML framework RASA[5]. The framework consists of the RASA NLU framework and RASA Core, which can be used for DM. The RASA NLU framework was used for semantic encoding of user utterances in the same way as presented in the previous experiment. RASA Core as the decision framework selected the next system action depending on the given user input. The procedure of RASA Core has been already explained in Section 4.2. However, adjustments had to be made for handling proactive dialogue based on external events. These adjustments included the definition of external intent concepts for NLU. External intents are pre-defined concepts that tag proactive system actions coming from an external source. These are then treated in the same way as user intents by the RASA Core. Another adjustment was to extend the framework with an additional action server for allowing proactive robot-initiated dialogue. This server checked for incoming messages from KURT and contained a rule-based logic for taking action based on external events. Messages between RASA Core and the robot's cognitive capabilities were exchanged using a specifically defined MQTT protocol [6]. MQTT is a widely used Internet-of-things protocol and allows an easy implementation of an asynchronous and bi-directional interaction that was required to enable proactive dialogue. For an illustration of the working method of the specified RASA model for enabling proactive dialogue, consider the following example. The interaction between robotic and interaction modules via MQTT was conducted by both parties subscribing to a so-called topic. Both interaction and robotic modules were able to publish on topics and an MQTT-broker forwarded the message to the entities that had subscribed to this topic. If an external event occurred, e.g. the robot sighted bleach on the couch table and thus reasoned the environment to be in an untidy state, the robot published this event on the specific topic. Subsequently, The action server received this message and used a rule-based mechanism for determining whether this contextual information required the initiation of proactive dialogue. When the need for proactive dialogue was determined, the action server triggered an external intent for further processing at RASA Core. Depending on the external intent, RASA Core then initiated a proactive message which was then provided to the user via the interface. How the domain was modelled by KURT's cognitive modules is described in the following.

[5] https://rasa.com/

[6] https://mqtt.org/

Domain Model

To detect the current state of the environment, it was important to know the tidy state. However, in the experimental domestic environment, this state was not unique, as the objects could have multiple correct locations. Therefore, KURT recorded the location of each object and calculated probability clusters for each object. The size of the clusters corresponded to the probability of objects being located at a specific place. This information was utilised for bring and place tasks. Furniture was detected by using QR-codes, however, for detecting the household objects, the learned shape of the objects was used. This learned information along with a small part of general world knowledge, such as *Coke* is a *Drink*, was used to form the knowledge graph of the robot. The knowledge graph was stored using a relational object ontology relying on the database framework ARANGODB (ArangoDB, 2021). All environmental entities were characterised by three conceptual categories in the robot knowledge base, *locations*, *objects*, and *properties*. Each of these categories had its hierarchy of specific classes and instances called nodes. Furthermore, the nodes were connected with edges that represented their relationships. The knowledge graph was used for inference reasoning. Misplaced objects were detected by consulting the knowledge base. Further, KURT could offer alternatives using the knowledge of the conceptual categories. For example, if the user asked for crackers and they were not available in the database, KURT offered chips as an alternative as the object "chips" was listed in the same category, "snacks". For more information about the robot's knowledge and reasoning capabilities, we refer the reader to Prasad and Ertel (2020).

KURT's action planning behaviour was implemented using the FLEXBE framework (FlexBE, 2018; Schillinger et al., 2016). A concurrent state machine was built that handled a particular task and also allowed user interrupts during task execution. To perform pick and place actions, the object had to be grasped, which potentially was surrounded by other objects. Depending upon the shape of an object, Kurt simulated multiple motion plans and executed the first successful plan that would not result in collisions.

6.4.4. Design of Proactive Dialogue Strategies

We defined proactive dialogue in the form of a *Suggestion* action for two scenarios. The first scenario for proactive robot behaviour was the detection of anomalous situations and their associated objects. For this, KURT exploited knowledge gained from learning the environmental state. To resolve a disordered (or *untidy*) state, a clean-up sequence was required to be initiated. To avoid confusion and misunderstandings, the robot first needed to proactively communicate this to the user. This allowed them to intervene if necessary. For example, a newspaper on the table could be a deviation from the original state, but it may not be desirable to remove it. In any case, the user's decision was then added to the robot's knowledge base, which empowered KURT to resolve a similar situation on its own in the future.

For our use case, the robot detected a deviation from the "normal" environmental state by sighting a bleach bin located on the couch table. This would induce the following proactive dialogue between KURT (**K**) and its user (**U**).

(...KURT *detects bleach bin on the couch table.*)

K: I have noticed a bleach bin on the couch table. I do not think that it belongs there. Should I put it back in the closet?

U: Yes, please!

K: Ok, I will put it back.

Furthermore, proactivity was deemed to be necessary for situations where a user's goal could not be fulfilled and alternatives were required to be offered. This was also intended to help alleviate the already mentioned problem of confusion and misunderstanding. For illustration consider the following interaction scenario, in which the user had the intention to eat a snack:

U: Kurt, bring me some crackers!

K: Alright, I will do that.

(...KURT *detects that the item the user wished for is not available*)

K: I am sorry, there are no crackers available. Would you like crisps instead?

U: Yes, please!

K: Ok, I will get you some.

Using these proactive dialogue strategies, we aimed to answer our research questions regarding cooperation. In the following, we describe the experimental design for studying these questions.

6.4.5. Experimental Design

We tested the described proactive version of KURT in a realistic task scenario. Due to the complexity of the study setup, we restrained from testing against a non-proactive version.

Participants

For the study, 17 participants were recruited and received 20 € in return for their participation. However, five had to be excluded due to the malfunctioning of the system. Hence, data from 12 subjects (34 % male) with an average age of 27.17 (SD = 10.71) were used for evaluation. 16 subjects were students of which 66 % were enrolled in a psychology major. Even though subjects had low experience (M = 2.92; SD = 1.68 measured on a 7-point Likert scale) and few prior knowledge (M = 3.17; SD = 1.90) regarding robots, they had no negative attitudes towards the interaction with a robotic system (M = 2.58; SD = 1.06 measured using the NARS (Nomura et al., 2006) scale). Furthermore, subjects had moderately high predisposition to trust autonomous systems (Merritt et al., 2013) (M = 5.15; SD = 0.98) as well as affinity towards technical systems (Karrer et al., 2009) (M = 5.39; SD = 0.78). All participants had to sign a declaration of consent to join the study and were assured of the confidentiality of their data.

Experimental Procedure

The experimental procedure was as follows: First, subjects were briefed about details of the data survey, e.g. duration (30 minutes) and purpose of the survey.

Subjects were told that they would take the role of a person in the household of the future. They would have purchased the household robot called Kurt recently. After the introduction, they had to fill out a pre-test questionnaire. Here, participants had to rate their predisposed trust in autonomous systems, NARS, and provide general information, e.g. age, gender, and major. Subsequently, the participants were introduced to the robot and received a note that described the scenario. They were told that the robot would introduce itself and that they would be required to ask for a snack after the robot has observed its surroundings. This served the purpose of standardising the study setup for all participants. Furthermore, subjects were also informed that Kurt was operated via tablet through which they could communicate with the robot using natural language. Afterward, the interaction with Kurt started. Upon completion, they had to fill in a questionnaire to assess the robot's proactive dialogue behaviour. Finally, the participants received their reward and were dismissed.

Questionnaires

The research questions were evaluated using the same questionnaires as used in the previous experiments. In the present study, we measured the robot's trustworthiness (Kraus, 2020), reliability, competence, usability, and acceptance. Contrary to the previous studies, we used the AttrakDiff questionnaire (Hassenzahl et al., 2003) for evaluating the usability of proactive dialogue. All scales were rated on a 7-point Likert scale from 1 (strongly disagree; word adjective (acceptance)) to 7 (strongly agree; word adjective (acceptance)), except usability which was rated on an 11-Point Likert scale (word adjective).

6.4.6. Results

The results of the study are presented in Table 6.5. In the following, we distinguish between individual result categories as we did for the other experiments.

Effects of Proactive Dialogue Strategies on Usability

The robot received generally positive ratings for the usability categories including the AttrakDiff and acceptance questionnaire. Observing the individual items of the used acceptance scale, we particularly found the interaction with the robot to be "good" ($M = 6.08$; $SD = 0.79$ 7-point Likert Scale), "non-annoying" ($M = 6.17$; $SD = 0.84$), and "pleasant" ($M = 6.00$; $SD = 0.85$). Furthermore, considering the AttrakDiff questionnaire, we found the proactive robot to be perceived "sympathetic" ($M = 9.08$; $SD = 1.68$ 11-point Likert Scale) and "innovative" ($M = 9.50$; $SD = 1.57$). Thus, we deemed Kurt to be well accepted amongst the study participants and showed tendencies towards high usability.

Effects of Proactive Dialogue Strategies on Trust

Considering the trust-related measures, the robot's proactive dialogue strategy received positive ratings for overall perceived trust, as well as reliability and competence (see Table

Dimension	*Score*
Trust	5.76 (0.67)
Reliability	5.10 (1.16)
Competence	5.42 (1.18)
Acceptance	5.85 (0.73)
AttrakDiff	8.83 (1.11)

Table 6.5.: Mean scores with standard deviations in brackets. (Kraus et al., 2022c)

6.5). Additionally, a significant within-subject difference was found regarding the development of trust towards the robot. Therefore, the differences between trust ratings before and after completion of the scenarios were assessed using a Wilcoxon-Signed-Ranks test (initial trust was measured with a propensity towards trust in autonomous systems questionnaire). The proactive robot could significantly increase trust during the experiment ($Z = -2.04,\ p = 0.041$).

6.4.7. Discussion

In the following, we discuss the results of the experiment regarding the influence of the robot's proactive dialogue strategies on HCT and usability. Further, we describe the observed impact of the trigger mechanism on cooperation.

Influence of Proactive Dialogue Level on Trust

Similar to the preceding experiments, we found that a robotic CA expressing a medium-level proactive behaviour significantly increased trust in the system throughout the experiment. Generally, Kurt's proactivity led to high perceived trust in the application. Congruently to our previous observations regarding the trust effects of the *Suggestion* action, this proactive dialogue act type seems to positively influence the robotic CA's cognitive trust bases – competency and reliability. Due to similar effects of medium-level proactive behaviour on trust and its related concepts in a full-scale robotic CA, we deemed the results of the more restricted lab studies with virtual CA's transferable to realistic use case scenarios.

Influence of Proactive Dialogue Level on Usability

The experiment also showed a positive impact of proactive dialogue on the system's usability. In this regard, the interaction with the assistant was rated as good and pleasant, and let the robot even be perceived as highly sympathetic and innovative. Further, the robot achieved to fulfill the required tasks for all scenarios showing high usability for this specific task domain.

Influence of the Trigger Mechanism on Cooperation

Using the same argumentation as for the previous experiment, we deem the trigger mechanism utilised for the robotic CA to have a positive effect on the cooperation because the interaction was highly perceived as not annoying. Thus, the initiation of proactive seemed to be conducted in a natural, non-intrusive fashion fostering an adequate proactive dialogue between human and robot.

Limitations

A limitation of this study was the rather small amount of experimental subjects. This was due to the complexity and expensiveness of the study setup. Further, more insights could have been gathered comparing different versions of the robotic CA being able to express distinct levels of proactivity. However, one purpose of this study was to measure the portability of the results found using the low-fidelity prototypes to realistic task scenarios and more sophisticated prototypes. Therefore, this study set up deemed to be appropriate for studying this aspect. Besides, comparing different prototype versions would have also further increased the study's complexity. For this reason, we conducted an interactive video study with Kurt, where the robot was able to express different types of proactive dialogue. This study is described in the next chapter. Finally, we found that the robotic planning framework required considerable computational effort and accordingly was not able to operate in a time frame most users would expect. Therefore, this could have negatively influenced the user's perception of the robot.

6.4.8. Conclusion

This experiment showed the benefits of using external events as a trigger mechanism for proactive behaviour and provided evidence for the portability of a medium-level of proactive dialogue to the domain of robotic CAs. Here, proactive behaviour was enabled by including modules for reasoning, planning, and dialogue capability in a Tiago robot. In a small experiment considering a realistic task scenario, where users collaborated with the robot on tidy-up and fetch-and-carry tasks, it was shown that a medium-level of proactivity provoked similar effects on users as found in previous experiments. In summary, the proactive robot was perceived as competent and reliable in the domestic task scenario, which led to high ratings of trust towards the robot. Further, the developed proactive dialogue strategy showed tendencies to positively contribute to the system's usability by expressing unobtrusive and natural proactive behaviour. Thus, the usage of external events for triggering proactive dialogue seemed to benefit human-machine cooperation.

6.5. Summary

In this chapter, we contributed four experiments that provided novel insights into the effects of proactive dialogue strategies on human-machine cooperation. For this, we considered the impact of the individual proactive dialogue act types on a system's trustworthiness and usability dependent on the task context, user characteristics, specific user states as well as activities, and external events that were supposed to require proactive assistance. In doing so, we aimed to understand proactive dialogue behaviour from a social and task-related perspective during cooperation. This understanding should be then used to implement a user-centred proactive DS for improving the cooperation. The first two experiments were conducted in restricted laboratory settings using rather simplistic prototypes, due to the complexity of the experimental designs and to enable standardisation.

The first experiment revealed different effects of proactive dialogue act types on HCT dependent on task difficulty and several user characteristics. Due to the effects of task difficulty, other task properties, e.g. task complexity, may also play a role in the trust perception of proactive dialogue. Considering user properties, we found differences in the trust perception of the dialogue act types for the user's level of domain expertise and technical affinity. For novice users with low technical affinity, a medium-level of proactive behaviour during the first interactions might be beneficial for building rapport and increasing cognitive trust in the system's abilities. Further, we also found differences in several personality traits. Here, we found that similar to the pilot study presented in Section 4.2, neuroticism may affect the perception of proactive dialogue. Further, we found differences in the traits of conscientiousness, and openness.

The second study provided insights that using only one specific user state (cognitive-affective user state) may be insufficient for deciding whether to become proactive. This observation was similar to the first experiment where the trigger mechanism was the user's insecurity level. Therefore, we concluded to include more information about different types of user information for developing a more adequate trigger mechanism for these decisions. Generally, the results of both experiments showed that the perception of the dialogue act types at the extremes of the continuum of our proactive dialogue act concept, namely the reactive *None* and the most proactive *Intervention* action, are highly affected by the context and the user. Therefore, we assumed that the decision of whether the system should express reactive or highly proactive behaviour largely depends on the social expectations regarding the proactivity of individual users given a specific situation. Contrary, the medium-level proactive behaviour in the form of the *Notification* and *Suggestion* action, generally led to a trust increase, and a system expressing such behaviour was considered reliable and competent.

For this reason, we deemed a medium-level proactive dialogue useful for the HCT relationship with being only slightly influenced by context and the individual user. Similarly to Isbell and Pierce (2005) and Yorke-Smith et al. (2012), the problem of selecting an adequate level of proactive dialogue can be considered as a cost-benefit function concerning a system's trustworthiness.

For medium-level proactive dialogue, the benefits generally outweigh the costs. For reactive or highly proactive dialogue behaviour the benefit of successful action implementation, as well as the cost of failure, are high. If a user expects a system to take over actions from the user (*Intervention* action), then the benefit is high. In case such behaviour is not expected, the costs are high. Also, Isbell and Pierce describe this relationship as considering highly proactive user interfaces. However, they claim that reactive behaviour has neither benefits nor costs.

Based on our results, we argue that reactive system behaviour may have similarly high benefits and costs as highly proactive behaviour. For example, when a user expects a system to become proactive for a given situation but it stays reactive, then this has a high cost as the user might question the system's competence and reliability which could damage the HCT relationship. Considering the usability of proactive dialogue act types during the first two experiments, we found that a high level of proactivity was generally perceived to be more task effective, wherein reactive system behaviour received the lowest scores for task effectiveness. Considering the medium-level of proactivity, we could make no clear statement. However, the *Notification* action showed tendencies to increase usability dependent on the task context.

For the third and fourth experiments, we switched to high-fidelity prototypes for investigating user-centred proactive dialogue strategies in more realistic task contexts. Further, we studied the influence of more contextual trigger mechanisms – user activity and external events – on cooperation.

The third study provided evidence that the approach is transferable into virtual CAs for application in realistic task scenarios. Here, the results emphasized our previous findings that proactive dialogue largely affects cognitive-based trust. Similar to our previous experiments, we found the proactive dialogue to have a low impact on affect-based trust. Further, the study stressed the influence of a user's domain expertise and technical affinity regarding the trustworthiness of proactive dialogue. Differentiating between different levels of domain expertise even revealed marginal differences between proactive and reactive behaviour for personal attachment. Considering the influence of proactive dialogue on the system's usability, we found positive tendencies of proactive dialogue on increasing task efficiency and task success. This reinforces our previous results of the *Notification* action increasing task effectiveness. Also, the usage of the user's activity as a trigger mechanism provided promising results in improving the cooperation by fostering a system's usability.

The fourth study aimed to test the applicability of proactive dialogue in robotic CAs. Here, the results further validated the generalisability of our concept and provided further evidence on the usefulness of the cognitive architecture for enabling proactive dialogue. Additionally, it stressed the importance of reasoning and planning for the adequate realisation of proactive behaviour. Here, we also found a positive impact of utilising external events as a trigger mechanism for cooperation. Generally, we found the usage of more context-related than user-specific information as a trigger mechanism for proactive behaviour highly useful when considering the third and fourth experiments. Thus, we concluded a combination of context- and user-specific information to be suitable for deciding whether to become proactive and to which extent.

In the next chapter, we fused the gained knowledge from our experimental studies to implement a user-adaptive proactive DS for improving the cooperation CAs. For this, we realised a user model for the inclusion of a trust measurement as a trigger mechanism for proactive dialogue behaviour in order to improve trustworthiness. In addition, we also included task-related features in the user model for also improving the usability of the system. For DM, we utilised statistical and rule-based approaches relying on study results from the examination of the different proactive dialogue strategies in this chapter. For realising a user-centred DS, adaptation mechanisms to the specific users and situations were required. Here, the question that needed to be answered was, how to adapt proactive dialogue behaviour and to how to identify a metric that may be used as decision criteria to adapt to. In doing so, the next chapter concludes the development process of a novel proactive dialogue model by implementing a sophisticated DM module that allows trustworthy and task-effective cooperation with CAs.

7. Improving Cooperation by Implementing User-Centred Proactive Dialogue Strategies

The overall goal of this thesis was the development of trustworthy CAs by implementing adequate proactive behaviour for improving human-machine cooperation from a social and a task-oriented point of view. During the design process of user-centred proactive dialogue strategies presented in the previous chapter, we gained an understanding of the effects of proactive dialogue on cooperation. From a social perspective, we showed that an appropriately selected proactive dialogue level may increase a system's trust by rendering it more reliable and competent. From the viewpoint of considering a system's usability, we found that the choice of trigger mechanisms for initiating proactive behaviour has an important influence on the user's task effectiveness. Further, we identified several implications of specific user characteristics and task context on the perception of proactive dialogue regarding cooperation.

The gained knowledge was exploited for the implementation of user-centred proactive DM. A central aspect in this regard was to provide the system with a mechanism that adapts the level of proactivity to the specific user and the current situation during cooperation. As a result of our previous experiments, measurements of a system's trustworthiness and usability seemed to form adequate decision criteria for adapting the proactive dialogue to improve cooperation. Usability measurements in the form of user satisfaction (Litman and Silliman, 2004) and interaction quality (Ultes et al., 2015) have already shown to be adequate adaptation criteria for improving human-computer dialogue. However, a trustworthiness measurement has yet to be shown to be successfully integrated for dialogue adaptation. Therefore, we first needed to evaluate trustworthiness as adaptation criteria for user-centred proactive dialogue. Subsequently, methods to adequately model trust during dialogue needed to be implemented.

For this, several steps were necessary. The first step towards this goal was to create a user model for taking into account a proactive system's perceived trustworthiness during an ongoing interaction. However, in current dialogue literature, there does not exist interaction data representing proactive system behaviour and the system's perceived trustworthiness. Hence, a trust-based proactive dialogue corpus had to be created in advance.

For generating the corpus, data was collected online with 308 users who had to interact with an artificial advisor agent in a serious proactive dialogue game. The data was annotated with objective features, e.g, task duration and success, as well as subjectively self-reported features, e.g. user's age, gender, and personality, for capturing the interplay between proactive behaviour as well as situational and user-dependent characteristics. For reflecting the trust relationship, interactions in the corpus were labeled with self-reported measures on the system's trustworthiness and its related concepts.

Based on the annotated data corpus a user model was implemented. The model incorporated user-, system-, and context-dependent features and can be used for a live prediction of the user's trust in the proactive actions of a virtual CA during an ongoing interaction. For predicting the user's trust level, three machine-learning algorithms were trained and tested on the corpus for comparing their applicability for the given task.

Finally, we implemented a user-centred proactive dialogue model including trustworthiness and usability measures and compared two different adaptation strategies: a *rule-based* and an *RL-based* method. For evaluation, we developed a user simulator based on the collected corpus data. In the following, we describe the steps for implementing user-centred proactive dialogue strategies.

7.1. Evaluation of Trustworthiness as Adaptation Criterion for User-Centred Proactive Dialogue

7.1.1. Motivation

Generally, humans tend to personify and associate human traits with machines (e.g., see Nass et al. (1994)). Thus, people have certain social expectations regarding interactions with such, similar to interacting with a fellow human being. In case, there exists a mismatch between a CA's impression of intelligence and its actual behaviour, a so-called *expectation gap* (Kwon et al., 2016) may be created. As a result, a loss of trust may occur.

In the introduction of this thesis, we explained that proactive behaviour is integral for CA's to be perceived as intelligent. Thus, proactive behaviour that fulfills the user's social expectation should result in a formation of trust and otherwise should result in a trust decrease. Therefore, trust may be an adequate measure to determine whether proactive behaviour is socially expected or not and may be used for selecting adequate proactive dialogue strategies to improve cooperation. Thus, we utilise this relationship to evaluate trust as adaptation criteria for user-centred proactive dialogue.

For evaluation, we equipped a household assistance robot with proactive behaviour that adapted to the user's social expectations and present a user study showing the effects on perceived user trust. The domestic domain and a robotic CA were selected, as social expectations particularly influence the interaction with such when they are applied in more social settings, e.g. as a household assistance robot (Edwards et al., 2016, 2019). Further, the domestic domain fulfilled the requirements of an adequate test scenario for studying the impact of proactive as described in the previous chapter.

To evaluate trust as an adequate evaluation metric, we compared an expectation-driven proactive dialogue strategy to four static strategies based on our conceptualisation of proactive dialogue act types (None, Notification, Suggestion, Intervention). Here, we deemed our assumption to be successful in the case that the trust relationship is enhanced by the expectation-driven strategy.

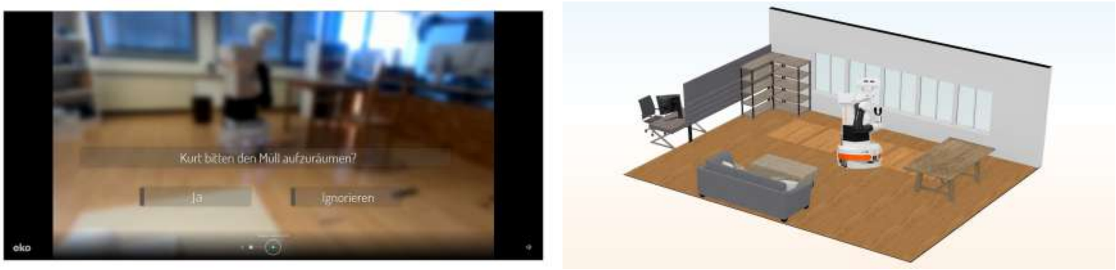


Figure 7.1.: Left: Depiction of the decision screen during the interactive video. The user may choose between asking the user to collect the garbage or ignore the robot. Right: Setup of the video recording in the simulated domestic environment. (Kraus et al., 2022e)

To produce a large sample size and to strengthen the standardisation of the study design, data were collected online using an interactive video method. Using this method, study participants were able to interact with the robot while watching a video. At certain moments, subjects were able to explicitly make decisions that directly influenced the robot's behavior and the further course of the experiment. In preparation for the study, the corresponding videos had been created with a manually operated robot that assisted in six typical domestic assistance scenarios. For evaluation, study participants rated the robot's trustworthiness, as well as whether they complied with the robot's proactive actions. Further, we evaluated the user experience with the study setup.

In the following, we present the approach, the design of the proactive dialogue strategies, and the experimental design. Further, we provide and discuss the results of the evaluation.

7.1.2. Approach

For evaluating trust as an adaptation criterion for user-adaptive proactive dialogue, a collaborative task scenario in the domestic domain was set up. As previously mentioned, related research showed that the domestic domain is particularly suitable for measuring trust in a robotic CA (de Graaf et al., 2019). One's own home is a place of intimacy and vulnerability, where forming a bond of trust is inevitable for allowing a robot to autonomously perform actions. Thus, robotic systems need to understand the user's intentions in such environments and adapt to socially adequate criteria, e.g. via engaging in a proactive dialogue.

In the presented use case, a user collaboratively interacted with a robot to solve typical tasks occurring in the domestic domain, e.g managing groceries or tidying up. As a household assistant, a Tiago robot was utilised for the high-fidelity prototype study. The Tiago robot, called Kurt again in the following, was embedded in a lab environment that was furnished with a couch, couch table, closet, and dining table. In doing so, the environment should resemble a typical living room and simulate a domestic environment. A schematic drawing of the lab is depicted in Fig. 7.1. In the experiment, the individual study subject took the role of the user and could control their actions.

Further, the user applied the thinking aloud method for keeping the study subject in the loop of the user's intentions. The user interacted with Kurt in six different scenarios. In all but two scenarios the robot could interact proactively with the user. The robot's proactive dialogue behaviour was modelled according to our proactive dialogue concepts. In the following, the distinct task scenarios are explained.

Scenario 1: Robot Introduction The purpose of this scenario was for the users to familiarise themselves with the system. Here, the assistant provided helpful information about its sensory system and functionalities through an introductory dialogue. This allowed novice users to obtain an overview of the features of the system. The interaction started with the robot greeting the user. Here, the proactive behaviour of the robot was not manipulated, as the robot's interaction purpose was only to present itself and not to assist in any task.

Scenario 2: Groceries Management This scenario was intended to provide the user a first experience of the assistance functionalities of Kurt for a simple household task. At the end of Scenario 1, the user thought aloud about going groceries shopping. After returning, the user put their groceries on the table and was welcomed back by the assistant. Depending on the configured proactivity level, different strategies were used for offering support in putting the groceries away.

Scenario 3: Bring Task I The purpose of the third scenario was to make the user aware of the robot's fetch-and-carry capability. Here, no robot proactivity was required. In this scenario, the user rested on the couch and developed an appetite for a snack. While the robot navigated through the room, the user could select from a list of options, e.g. *"Get me some chips!"*, and instruct the robot to perform the task. Subsequently, the robot fetched the snack and handed it over to the user. Another fetch-and-carry task was initiated by the user in scenario 5.

Scenario 4: Tidy Up I Here, the user decided to read a newspaper at the couch table. The user's point of view is depicted in Figure 7.2. After a while, an incoming phone call (simulated by cell phone noises) caused the user to leave the table. In the meanwhile, Kurt approached the table and noticed the newspaper. Analogously to scenario 2, the robot selected one of the proactive strategies for offering assistance. However, in this scenario, the user thought aloud about not having finished reading yet and only needed to interrupt the activity due to the distraction. Hence, the dialogue strategies applying a higher level of proactivity were deemed inappropriate at this point. This scenario aimed to get feedback on how subjects perceive unwanted help from Kurt.

Scenario 5: Bring Task II Similar to scenario 3, this scenario dealt with a fetch-and-carry task. Here, the user thought aloud of being thirsty and asked Kurt for a soft drink. The robot confirmed the task and went away for fetching the drink. However, it returned shortly afterward and reported that the desired beverage was not in stock (*"I'm sorry.*

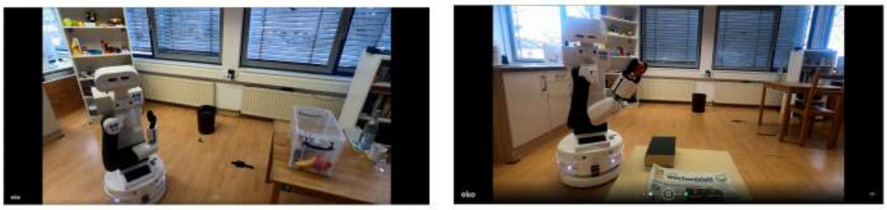


Figure 7.2.: Screenshots of the interactive videos. Left: Groceries Management (Scenario 2). Right: Bring Task I (Scenario 3). (Kraus et al., 2022e)

Coke is not available"). Subsequently, the robot acted according to one of the proactive strategies. In the reactive condition, the user had to ask explicitly for an alternate drink. In the proactive conditions, KURT notified about or recommends alternatives, or directly told the user that it would get an alternative drink instead. The purpose of this scenario was to let subjects experience the robot's behaviour acting upon unexpected events.

Scenario 6: Tidy Up II Contrary to scenario 4, in which high proactive behaviour was supposed to be inappropriate for the given situation, this scenario was intended to favour proactive robot actions. Here, the user left an empty bottle on the table. After KURT had approached the table, it noticed the bottle. Depending on the proactive configuration of the robot, it could offer to throw away the bottle in the already described ways. Generally, users were expected to want a robot action in this context and to request or let the robot perform the task.

We did not utilise an actual "system" prototype for the experiment, but combined an WoZ approach (Kelley, 1984) with an interactive video method. Using this method, study participants were able to interact with the robot while watching a video. This allowed the study to be conducted online and to produce a large sample size for strengthening the standardisation of the study design. Further, this was intended to reduce the study's complexity. The description of the combined approach can be divided into two parts: the creation of the video material and the development of the interactive videos.

Creation of Video-Material

The video recordings were based on a screenplay that comprised different interaction scenarios. The screenplay featured two protagonists: the user and the household assistance robot called KURT. The recordings were shot from the first-person perspective. In doing so, a more realistic experience regarding the HRI could be created where the viewer empathised better with the main character. As the actor was a male, a male voice was used for this character. For shooting the videos the protagonist held a GoPro HERO8 camera in his hand. This camera allowed video production in high definition with a resolution of 2704 x 1520 pixels and a frame rate of 50 fps. In addition, the sound was recorded at a sampling rate of 48 kHz with stereo microphones. For facilitating the control of the camera, i.e. starting and stoppage of filming, a "director" remotely controlled the GoPro using a smartphone. The director was able to watch the camera's footage on the smartphone screen. This further allowed us to correct the camera settings in case of unfavorable perspectives.

The WoZ-paradigm was realised by a human operator controlling the movements of the robot, the gripper, and triggering the robot's speech output at well-defined moments. The robot's utterances were scripted in the screenplay. The appropriate moments for triggering the robot's speech were pre-defined in the script and the same for each proactive configuration of the robot. Depending on the proactive level, KURT used slightly different wording.

For each scenario where the robot could engage in a proactive conversation, video snippets of different proactive behavior were created. The whole video creation process lasted approximately seven hours and served as the foundation for developing the interactive videos. How the videos were provided with interactivity is described in the following.

Development of Interactive Videos

After recording the video material, data processing was carried out. This included sorting and editing the files, and occasionally filtering out background noise. As a result, the human-robot dialogue was segmented into separate video segments with a duration of 10 - 30 seconds. Subsequently, the toolkit EKO [1] was employed to create an interactive movie. The basic structures of these movies were similar to a decision tree. In our videos, each dialogue step ended with a system question. The user could then select an answer from a list of options. While the options menu was displayed, the video was stopped and blurred. A picture presenting the options menu is provided in Fig. 7.1. Depending on the user's selection, the appropriate follow-up video was displayed. During the interactive movie, it was possible to repeat the entire conversation as well as individual steps. In the next section, the design of the proactive dialogue strategies is explained in detail.

7.1.3. Design of Proactive Dialogue Strategies

In the domestic task domain, a household assistant robot was able to perform tasks using different levels of autonomy on the spectre provided by Sheridan and Verplank (1978). For communication of these degrees, our conceptualisation of proactive dialogue act types was applied. For each proactive dialogue act type, a separate video was created for the respective use case scenarios. In the following, the individual proactive actions are described more in detail:

None This strategy implied reactive robot behavior and constituted the lowest level of autonomy. In this condition, users could only explicitly request help from the robotic CA. For the presented use case scenarios, this was implemented as a "wait and see" behavior expressed by KURT. In scenarios 2, 4, and 6, the robot positioned itself near the household objects (groceries, newspaper, and empty bottle) and awaited the user to act. In scenario 5, the robot simply said that the beverage would be out

[1] https://studio.eko.com/

of stock. In all scenarios, the user could ignore the robot or ask for assistance. For example, consider the following dialogue:

*(*Kurt *positions itself in front of the empty bottle)*
U: Hey Kurt, can you put away the empty bottle?
K: Yes, I will put the empty bottle away for you.
*(*Kurt *starts to grab the empty bottle)*

Notification This strategy was the least intrusive proactive approach. Here, the robot verbally notified the user to shift their focus on the current situation. Afterward, it was left to the user to ask the robot for assistance or to ignore the notification. For the use case scenarios 2, 4, and 6 the robot positioned itself again close to the household objects but instead of waiting for a user action, the robot would notify about its detection. In scenario 5, the robot would explicitly tell the user which beverage was still available. In all scenarios, the user could ignore the robot or ask the robot for assistance. For example, consider the following dialogue:

*(*Kurt *positions itself in front of the empty bottle)*
K: There seems to be garbage on the table.
U: Hey Kurt, can you put away the empty bottle?
K: Yes, I will put the empty bottle away for you.
*(*Kurt *starts to grab the empty bottle)*

Suggestion Using the aforementioned strategy, the robot directly proposed an action the robot could take on behalf of the user. Thus, Kurt took more initiative in the interaction and presented an option. In response to the robot's proposal, the user could either confirm or decline the offer. In all use case scenarios, where proactive robot behavior was applicable the system positioned itself either close to the object or the user and proposed action, e.g. see the following example:

*(*Kurt *positions itself in front of the empty bottle)*
K: There seems to be garbage on the table. Should I put it away for you?
U: Yes, Kurt. I allow it.
K: Ok, I will put the empty bottle away for you.
*(*Kurt *starts to grab the empty bottle)*

Intervention Following this strategy, Kurt executed a particular action in place of the user. In all use case scenarios where proactive behaviour was applicable, the robot explicitly conducted either tidy-up tasks or provided the user with an alternative beverage. However, users were able to stop the robot verbally during execution. For exemplifying this strategy, consider the following dialogue:

*(*Kurt *positions itself in front of the empty bottle)*
K: There seems to be garbage on the table. I am going to put it away for you.
*(*Kurt *starts to grab the empty bottle)*
(User lets the robot take over)

Static proactive behavior was realised by providing the user with the same proactive action, i.e. only *None*, *Notification*, ..., throughout all scenarios. To act upon the user's social expectations of the robot's behaviour, we created an adaptive strategy that varied the proactive actions for each scenario dependent on social guidelines.

User-adaptive Strategy For adapting KURT's proactive behavior to the user's social expectations, a hand-crafted strategy was created. The strategy was designed for choosing the most suitable proactive action for the respective use case scenario. For making the decisions on which proactive actions to use at which moment, we adhered to the guidelines of "social etiquette" in the design of human-automation interaction by Sheridan and Parasuraman (2005), the theory of proactivity by Yorke-Smith et al. (2012), and our considerations. An example of good etiquette in human-automation interaction is to act in such a way that serves the present purpose and is not interrupting but patient (Sheridan and Parasuraman, 2005). The theory of proactivity comprises the theory of user desires ("assess the situated value of each potential agent action in terms of the user's objectives"), theory of helpfulness ("agent's reasoning to determine what actions would (most) aid the user now and in the future"), and the theory of safe actions ("bounds on what an agent is allowed to do when performing tasks proactively"). Based on this, we selected a proactive action for each scenario that was supposed to match the subjects' expectations. For scenario 2, the suggestion action was deemed to be the most socially appropriate. People might have a certain preferred arrangement for groceries, so there was a need for more control of the human in this situation. Further, this scenario described the first assistance context in the study and the subject was not yet familiar with KURT's actions. Therefore, suggestion behaviour was implemented for avoiding imposing behavior and being perceived as more polite. For scenario 4, reactive behaviour was implemented. Here, proactive behaviour may not be expected by users as they were only distracted from the task. For scenario 5, a notification action was selected. Here, directly offering a specific alternate drink was deemed inappropriate as the robot did not know the user's preferences. Thus, only notifying the user that there exist alternatives was implemented. For the final scenario, the intervention action was implemented, as a robot that is autonomously able to dispose of the waste was deemed to be socially expected. In the video-based experiment, we compared this expectation-based strategy to the four static proactive dialogue strategies. The experimental setup is explained in the following section.

In summary, the design of the expectation-driven proactive dialogue strategy allowed us to study whether trust represented an adequate decision criterion for adapting proactive dialogue strategies. In the following, we describe the experimental design of the investigation.

7.1.4. Experimental Design

The study setup followed a mixed-factorial experimental design. Here, the proactive dialogue strategies (none - notification - suggestion - intervention - adaptive) were evaluated

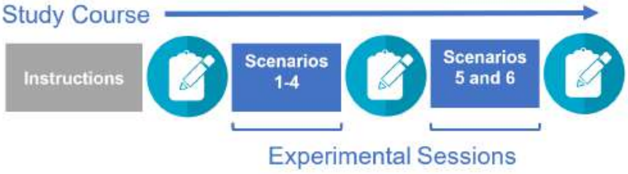


Figure 7.3.: Procedure of experiment. After each experimental session, dependent variables were assessed. At the beginning, study participants received instructions about the study and filled out a pre-questionnaire concerning their demographics, etc. (Kraus et al., 2022e)

to be independent between-subject variables. Study participants were evenly distributed among these five groups. In our experiment, we assessed trust and its five bases (competence, reliability, understandability, personal attachment, faith) towards the robot as well as the participants' cognitive workload during the interaction to evaluate the effects of proactive dialogue behavior. Furthermore, we measured the user's experience with the robot. These measures were collected twice during the experiment for each study participant. Users answered the questionnaire after scenario 4 and after the final scenario. The reason for this was to measure the immediate impact of the respective robot behaviour that was either contrasting (e.g. high level of proactivity in scenario 4) or in favour (e.g., high level of proactivity in scenario 6) of the social expectation. In addition, we assessed the compliance rates, i.e. how often subjects agreed with the robot's decisions, as an objective measurement of the user's trust.

Participants

Data collection was conducted using the German clickworker platform [2]. Eligibility conditions required users to be aged between 18 and 65, to be a native speakers of German, and to watch the interactive videos on a desktop computer for compatibility reasons. In total, 200 participants were recruited. However, some participants were excluded due to violations of the study instructions and technical errors. As a result, 163 subjects (34 % female) with an average age of 41 ($SD = 12.04$) were considered for evaluation. Participation was compensated with a monetary reward of 3.50 €.

Experimental Procedure

In advance of the experiment, users were briefed about details of the data survey, e.g. duration (20 minutes) and purpose of the survey, and had to give signed consent. Furthermore, participants were informed that concentration checks were included in the ratings to avoid misuse. For this reason, also the videos could not be skipped. After the introduction, subjects had to fill out a pre-test questionnaire comprising demographics and confounding variables. Subsequently, the participants had to watch the interactive videos

[2] www.clickworker.de

for scenarios 1 through 4. After completion, they filled in a questionnaire to assess the dependent variables and to check the manipulations. The same procedure was repeated for the last two scenarios. In conclusion, participants received their clickworker code for compensation and were dismissed. The experimental procedure is depicted in Fig. 7.3.

Questionnaires

Each dependent variable was measured with items from established and validated psychological scales. To determine trust towards the robot, a short version of the Trust in Automated Systems Scale (Jian et al., 2000) in German by Kraus (2020) was implemented. Furthermore, scales for measuring the bases of trust developed by Madsen and Gregor (2000) were used. The user's experience with the system was assessed using a short version of the user experience questionnaire (UEQ s) developed by Laugwitz et al. (2006). For measuring three types of cognitive loads (extraneous, germane, intrinsic), a questionnaire developed by Klepsch et al. (2017) was included. Besides, for personality assessment, the Big-Five-Inventory BFI-10 by Rammstedt et al. (2013) was included. The scales, which were only available in the English language, were translated into German. Besides, some scales were slightly modified for content and study context. For example, we clarified that participants rate the interaction with "the robot", as the original questionnaires make use of the neutral term "system" in the scale statements.

Possible confounding variables were measured using scales of predisposed trust in autonomous systems (Merritt et al., 2013), NARS (Nomura et al., 2006), as well as self-developed questions for previous experience with spoken dialog systems and the users' responsibility for household tasks. In doing so, we wanted to detect user-dependent biases for any study group. All scales were rated on a 7-point Likert scale from 1 (strongly disagree; word adjective (UEQ)) to 7 (strongly agree; word adjective (UEQ)).

7.1.5. Results

For data analysis, a multivariate ANOVA for confounding variables and the manipulation checks, as well as a mixed ANOVA for the independent variables at different time steps were used. No significant outliers were found in the data set.

Confounding group differences for proactive behaviour could be ruled out as the multivariate ANOVA did not reveal any significant differences (all p-values >> .05) except for the users' responsibility for household tasks ($F(4, 158) = 3.48$, $p = .009$). However, the Bonferroni-Holm corrected post-hoc t-tests were not significant. For this reason, responsibility for household tasks was not specifically considered for analysis.

The evaluation of the manipulation check confirmed the successful manipulation of proactive dialogue behavior (all p-values $< .001$ concerning the non-proactive strategy). Regarding the manipulation of the robot's adaptivity, all strategies were rated as adaptive: *Adaptive* ($M = 5.29$, $SD = 1.02$), *Intervention* ($M = 5.42$, $SD = 1.20$), Suggestion ($M = 5.92$, $SD = .82$), Notification, ($M = 5.53$, $SD = 1.03$), None ($M = 5.05$, $SD = 1.41$). Therefore, we conclude that study participants perceived the robot's ability to adjust its functions and vocabulary to different tasks as adaptivity.

Proactive Strategy	**Trust**	**Competence**	**Reliability**	**Understandability**
	M (SD)	*M (SD)*	*M (SD)*	*M (SD)*
None	0.15 (1.20)	0.11 (1.08)	0.45 (1.00)	0.62 (0.95)
Notification	0.12 (1.27)	0.19 (1.07)	0.34 (1.29)	0.55 (1.20)
Suggestion	0.11 (0.93)	0.12 (.93)	0.21 (0.85)	0.68 (0.67)
Intervention	-0.22 (1.00)	-0.13 (1.02)	0.02 (0.88)	0.38 (0.88)
Adaptive	0.43 (0.7)	0.34 (0.76)	0.42 (0.69)	0.65 (0.79)

Table 7.1.: Descriptive statistics of perceived trust, competence, reliability, understandability in the system with reference to proactive dialogue strategy. Values are taken from the final evaluation after the last scenario. Trust and its subbases were baseline-corrected according to measurement of predisposed trust in each group. The means for each group: $None = 4.97, Notification = 5.10, Suggestion = 5.26, Intervention = 5.01, Adaptive = 4.83$. (Kraus et al., 2022e)

Hence, the manipulation of the robot's proactive dialogue strategy to different situations was only implicitly perceivable.

As the feeling of trust is quite individual and is dependent on several factors, e.g. attitudes of a person or previous experiences, trust measurements should be baseline-corrected concerning a subject's propensity to trust (Merritt et al., 2013). For allowing such a correction, the correlations between a user's propensity to trust and all trust-related concepts needed to be considered. Using Spearman's ρ, we found strong correlations (Cohen, 1988) of a subject's propensity to trust and the measurements of trust towards the robot ($\rho = 0.55, p < .001$), perceived competence ($\rho = 0.59, p < .001$), reliability ($\rho = 0.61, p < .001$), and understandability ($\rho = 0.59, p < .001$). Furthermore, we found moderate relationships with the measurements of faith and personal attachment (both $\rho = 0.49, p < .001$). However, it only seemed reasonable to consider only the strong correlations for the baseline correction. Hence, the correction was conducted by subtracting the value of a participant's propensity to trust from the values of perceived trust, competency, and reliability. For a clearer description of the results, the evaluation was split regarding the user experience with the study setup, as well as the effects of the different proactive dialogue strategies on trust, and user experience.

User Experience with the Experimental Design

Generally, the experimental setup received favourable ratings. The hedonic quality of the interaction with the robotic CA was rated positively with an average rating of $M = 5.16,\ SD = 1.16$. Also the pragmatic quality of the interaction was well received $M = 5.22,\ SD = 1.15$. Considering the individual questionnaire items, the interaction with the robotic CA was particularly rated as "easy" ($M = 5.45,\ SD = 1.14$), "clear" ($M = 5.28,\ SD = 1.21$), and "leading edge" ($M = 5.45,\ SD = 1.31$).

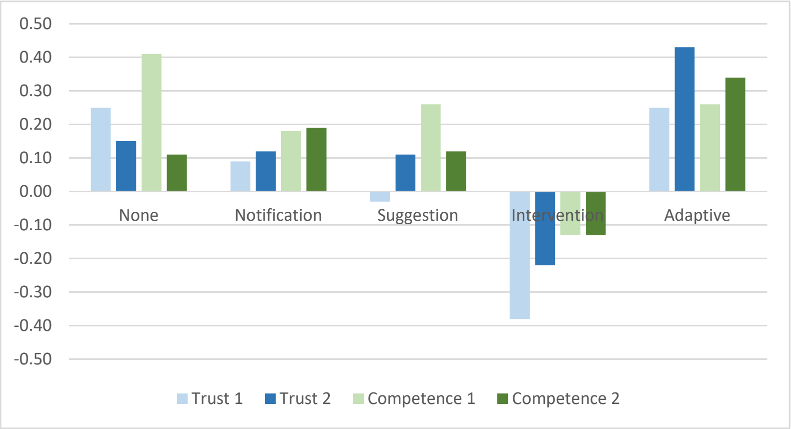


Figure 7.4.: Trust and competence development in the robot's actions during the experiment with respect to the proactive strategy. All values are baseline corrected. The indices "1" and "2" represent the times of measurements: "1" = after scenario 4 and "2" after scenario 6. Indications of standard deviations were omitted for clarity reasons. (Kraus et al., 2022e)

Effects of Proactive Dialogue Strategies on Human-Computer Trust

The mixed ANOVA showed a tendency towards interaction effects for perceived trust ($F(4, 158) = 2.21,\ p = .070$) and competency ($F(4, 158) = 2.01,\ p = .096$) depending on the measurement timing. For further evaluation, the simple main effects of the proactive strategy and the timing of measurements were investigated. Using Welch's ANOVA, a significant influence of the level of proactive dialogue on trust was found for both measurements ($F(4, 158) = 2.64,\ p = .040$ for t_1, $F(4, 158) = 2.54,\ p = .047$ for t_2). However, Bonferroni-Holm corrected post-hoc tests, revealed no significant differences between the proactive strategies. For examining the influence of the degree of proactive behaviour between and after the experiment, paired t-tests were applied. Here, significantly increased trust ratings between the two measurements were found for the *Adaptive*- ($t(27) = 2.20,\ p = .036$) and the *Intervention* strategy ($t(40) = 2.27,\ p = .029$). The perceived competence in the robot significantly decreased for the *None* strategy ($t(30) = -2.73,\ p = .011$). The understandability of the *Intervention* strategy increased significantly ($t(40) = 2.51,\ p = .016$). The results for each trust-related variables with respect to the proactive strategy after the final evaluation are depicted in Table 7.1, where the baseline corrected values for trust, competence, reliability and understandability are presented. The temporal differences of the proactive strategies on trust and competence are visualised in Fig. 7.4. Here, also the baseline-corrected values are shown which were measured after scenario 4 and after scenario 6.

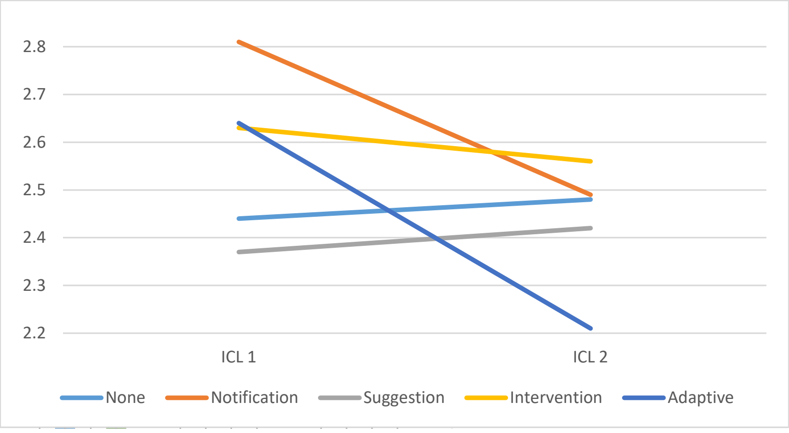


Figure 7.5.: Development of the ICL over the course of the experiment with respect to the proactive dialogue strategies. The indices “1” and “2” represent the times of measurements: “1” = after scenario 4 and “2” after scenario 6. Indication of standard deviations were omitted for clarity reasons.

Effects of Proactive Dialogue Strategies on User Experience and Cognitive Load

The mixed ANOVA revealed a statistically significant interaction between proactive dialogue strategies and the different measurement times for the measured ICL ($F(4, 158) = 2.99,\ p = .020$). Utilising the *Adaptive* strategy resulted in an attenuation of the ICL ($t(27) = -3.29,\ p = .003$). Additionally, the ICL was decreased by the *Notification-strategy* ($t(36) = -2.55,\ p = .015$). The results regarding the within-subject factor are visualised in Fig. 7.5 for the ICL. The mixed ANOVA revealed a tendency towards interaction effects for the proactive dialogue strategies measured during and after the experiment for pragmatic ($F(4, 158) = 2.28,\ p = .063$) and hedonic quality ($F(4, 158) = 2.05,\ p = .090$) of the user experience ($F(3, 38) = 3.95,\ p < .05$). Using Welch’s ANOVA, there were no significant regarding the influence of the proactive strategies (all $p > .010$). However, several significant results were found while investigating the influence of the degree of proactive behaviour dependent on the measurement timing. Utilising the *Adaptive* strategy resulted in an increased rating for hedonic quality ($t(27) = 2.79,\ p = .010$). The pragmatic quality was increased by the *Intervention* strategy ($t(40) = 2.80,\ p = .008$) and *Suggestion-strategy* ($t(25) = 2.31,\ p = .029$). The results regarding the within-subject factor are visualised in Fig. 7.6.

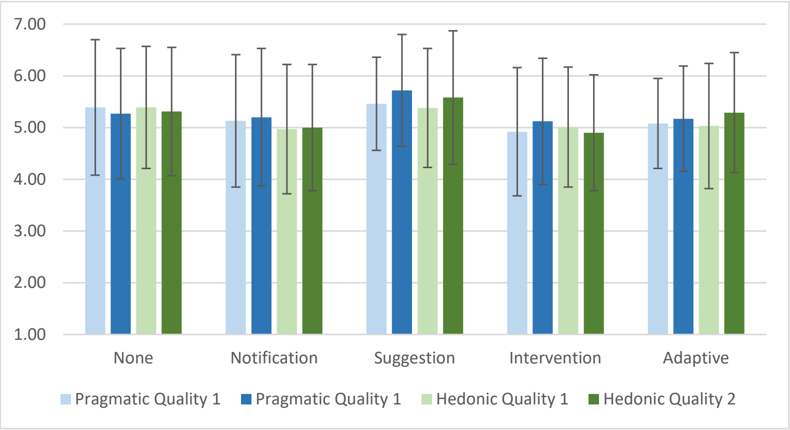


Figure 7.6.: Descriptive results of the user experience ratings over the course of the experiment with respect to the proactive dialogue strategies. The indices “1” and “2” represent the times of measurements: “1” = after scenario 4 and “2” after scenario 6.

Proactive Strategy	**Scenario 2**	**Scenario 4**	**Scenario 5**	**Scenario 6**	**Mean**
None	87 %	65 %	74 %	81 %	77 %
Notification	97 %	41%	87 %	81 %	77 %
Suggestion	96 %	23 %	50 %	77 %	62%
Intervention	88 %	20 %	63 %	88 %	65%
Adaptive	93 %	54 %	93 %	93 %	83 %

Table 7.2.: Compliance rates with the robot's actions dependent on the scenario and the proactive strategy. (Kraus et al., 2022e)

7.1.6. Discussion

In the following, we discuss the results under consideration whether trust formed an adequate decision criterion for proactive dialogue adaptation. In addition, we describe limitations of the study design.

Trust as Adaptation Criterion for Proactive Dialogue

The results suggested that trust may be indeed useful for the selection of an appropriate level of proactive dialogue during cooperation. This was supported by the significant increase of trust in the expectation-driven proactive dialogue strategy throughout the experiment (see Fig. 7.4). Furthermore, perceived competence increased the most as compared to the static strategies, whereas the *Adaptive* strategy also yielded the highest scores for overall trust and competence (see Table 7.1). Besides, the *Adaptive* strategy showed high values for reliability and understandability. Thus, including social expectations for choosing the level of proactive dialogue increased the robotic CA's trustworthiness.

However, this primarily held considering cognition-based trust (competence, reliability, understandability) as there were no findings in this regard for affect-based trust. This is congruent with our previous experiments. A reason for this may be that we only considered short-term interactions with the robot where only the system's functional capabilities were the centre of attention. Related work showed that the adaptation of the level of autonomy, either explicitly by the user, e.g. see Sanders et al. (2011), or the task difficulty, e.g. see de Visser and Parasuraman (2011), similarly helped to foster cognition-based trust in a robot's autonomous behaviour. Also, in Section 6.1, we showed that using different proactive dialogue strategies dependent on the task difficulty could increase user cognition-based trust. Therefore, only cognitive-based trust should be considered as an adaptation criterion for proactive dialogue.

The main driving factor for the trust increase by the *Adaptive* strategy concerning cognition-based trust seemed to be the avoidance of communication errors. These were prevented by changing the communication behaviour according to the user's social expectations. For example, the inappropriate use of the *None* and *Intervention* strategy, produced communication errors that negatively influenced the perceived trust towards the robot (see Fig. 7.4).

Similar results concerning the negative influence of communication errors on trust were shown by Wang et al. (2010). Using a constant medium-level of proactivity (Notification-, Suggestion-strategy) seemed to mitigate this effect, as there occurred no notable drop in the user's trust. Thus, we propose to carefully consider the use of reactive and fully proactive dialogue strategies dependent on social expectations.

Other evidence that the robot's assistance was objectively trusted the most using the *Adaptive-strategy*, was provided by the compliance rates with the robot's actions (see Table 7.2). Surprisingly, in Scenario 4, where users were not supposed to accept help from the system, they requested robot assistance when it expressed a low level of proactivity (None: 65%; Notification: 41 %). When the robot expressed higher proactivity, users tended to decline the offer or even stopped the robot in execution (Suggestion: 23 %; Intervention: 20%). This could be a sign that users are more to change their intention if they have more control over the interaction and the system acts more in the background as opposed to imposing itself. However, this needs to be investigated in different studies.

Considering the user experience with the robot, the *Adaptive strategy* increased the hedonic quality of the robot throughout the interaction. Hedonic quality is mostly non-goal-oriented and related to concepts of stimulation and novelty (Laugwitz et al., 2006). Thus, an adaptive robot seemed to increase the social aspects of cooperation. However, there were no differences in the absolute values of the measurement of hedonic quality and pragmatic quality. The *Suggestion* and *Intervention* strategies improved the pragmatic quality between the two measurement times. Pragmatic quality is a goal-oriented concept that is related to the ease of use of a system or its efficiency. A reason for the increase in the pragmatic quality of these strategies seemed to be that they were perceived as less goal-directed in scenario 4. As previously mentioned, users' expected the robot to not intervene actively during this task. Thus, the robot using these strategies could rebuild its pragmatic quality after acting in favor of the users' expectations. Therefore, we concluded that trust as an adaptation criterion may only improve the social aspects of the cooperation, but may not necessarily enhance the usability of a system for task effectiveness.

Another interesting finding was that the *Adaptive* strategy could significantly decrease the measured ICL. The ICL is usually associated with the difficulty of the experimental task. In this experiment, the user's task was to interact with the robot. Thus, ICL was related to the inherent difficulty or complexity of the task. As a result, the interaction with the adaptive robot was perceived as less complex and exhaustive. However, this finding was not surprising as the system always acted following the user's expectations. Furthermore, the *Notification* strategy led to a decrease in the experienced ICL. This could be a reason why this strategy was more robust for contextual changes, as the interaction was perceived as less complex. Regarding the ECL and GCL no significant differences were observed. Based on these observations, we concluded that task difficulty or the complexity of the task may be considered for dialogue adaptation as well besides a HCT measurement.

Limitations

A limitation of this experiment was that no face-to-face interaction with the robot took place. Interacting with a robot in person may reveal further insights and more significant effects as compared to a video study. However, Babel et al. (2021) provide support for the validity of online study findings for robot evaluations as compared to lab studies. By conducting an interactive video study, the validity of our study should be further increased, as users were more actively integrated into the experiment. Despite utilising an online study design. Furthermore, only a short-term interaction consisting of six dialogue turns was investigated. Including more dialogue steps would most likely show more effects regarding the dynamics of proactive dialogue.

7.1.7. Conclusion

In this experiment, we investigated whether trust may be used as an adaptation criterion for the selection of adequate proactive dialogue strategies. For this, a hand-crafted expectation-driven proactive dialogue strategy was created and evaluated regarding its influence on trust in a domestic use case scenario. The adaptive strategy was compared to four static proactive levels (None, Notification, Suggestion, Intervention). For evaluation, we employed an interactive video method to collect data. Using this method, study participants were able to interact with the robot while watching a video. At certain moments, subjects were able to explicitly make decisions that directly influence the robot's behavior and the further course of the experiment. The results showed that trust may be a useful adaptation criterion for user-adaptive proactive dialogue to improve cooperation. Here, we found that especially cognition-based trust features may be used for adapting the proactive dialogue. Regarding the influence on the cooperation, we concluded that deciding on an adequate level of proactive dialogue based on such a trust measurement may solely improve the social aspects of the cooperation but not necessarily its usability. Therefore, usability features should also be integrated for dialogue adaptation. Further, we reinforced the results from previous findings that task difficulty or complexity had a major influence on the effect of proactive dialogue regarding usability. To use a trust metric as an adaptation criterion during proactive dialogue, we first needed to model trust during cooperation to finally recognise and measure trust. The development and implementation of such a recognition module are presented in the following section.

7.2. Implementation and Evaluation of a Trust Recognition Module

As extensively described in Section 3.2, a (spoken) dialogue can be adapted to various features. For example, the decision on which dialogue action to choose can be based on solely performance-based or usability-based criteria, such as task success (Young et al., 2013; Lemon, 2008). Further, also more user-related information may be taken into account, e.g. perceived user satisfaction with the system (Litman and Pan, 2002) or interaction quality (Schmitt et al., 2011). The latter was used by Ultes et al. (2015); Ultes (2019) for creating user-adaptive dialogues in an information retrieval domain. There, it was shown that adaptive dialogue led to higher task performance and increased the quality of the dialogue. Although being labeled as user-adaptive, the dialogues were rather adapted to the performance of the spoken DS modules and did not include individual user characteristics. Besides, user adaptations were conducted for a rather soft assistance task, i.e information retrieval, in which the system was not required to become proactive. In more complex task scenarios, e.g. decision-making, proactive behaviour as a machine intelligence trait showed to be socially expected for CAs. However, the experiments regarding different proactive dialogue strategies stressed that the decision when to become proactive and to which extent largely depended on the user and context. Further, this decision highly benefited or harmed the HCT relationship, in case proactivity was provided adequately respectively wrongfully. Therefore, not only usability-based measures but also the HCT relationship seemed to be useful for deciding on appropriate proactive CA behaviour. As previous trust-related work revealed various human, machine, and context-related factors influencing the trust relationship (Parasuraman and Riley, 1997; Muir, 1994; Lee and See, 2004; Hoff and Bashir, 2015), an extensive set of information needs to be taken into account for modelling trust in proactive mixed-initiative interaction.

Although there exists a variety of data corpora (e.g. DSCT (Williams et al., 2014), MultiWoz (Budzianowski et al., 2018)) for conventional dialogue modeling, none of them are sufficient for modeling proactive dialogue. The main reason for this is that proactive behaviour is simply not included in such corpora or highly underrepresented (Balaraman and Magnini, 2020). Further, existing data corpora were also not sufficiently annotated using trust-related features. Therefore, a novel data corpus for this purpose was created in the scope of this thesis. For this, a low-fidelity CA prototype for personal advising was developed incorporating a proactive dialogue model. The CA was then used for collecting personal and dialogue data in a serious gaming scenario. This resulted in a trust-annotated data corpus containing interactions with the proactive assistance system. The corpus was then used for developing a user model for predicting the perceived trustworthiness of a proactive system. A depiction of this user modelling process is presented in Fig. 7.7. The goal here was to accurately model and predict trust using user-, system-, and context-related features during mixed-initiative dialogue. In the following, the data collection method, details about the corpus, and the methods for predicting the HCT relationship are described.

7.2.1. Data Collection Method

For the human-to-machine data collection, a low-fidelity prototypical proactive dialogue assistant was implemented based on our developed proactive dialogue model. The assistant was embedded in a mixed-initiative serious games environment. Serious games are "games used for purposes other than mere entertainment" (Susi et al., 2007), and are intended to lever "the power of computer games to captivate and engage end-users for a specific purpose, such as to develop new knowledge and skills" (Corti, 2006). Two properties of serious gaming are particularly beneficial for the approach of data acquisition

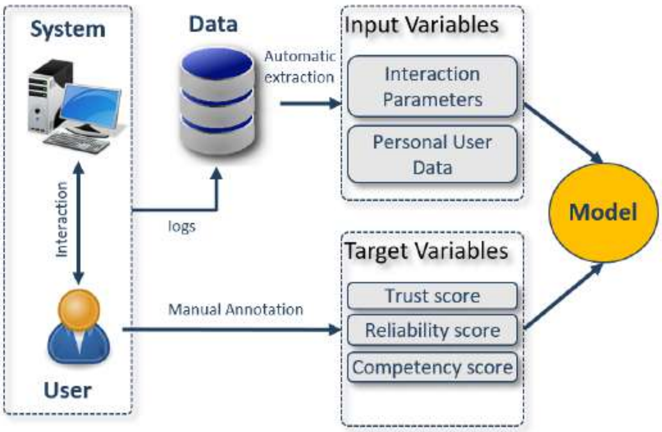
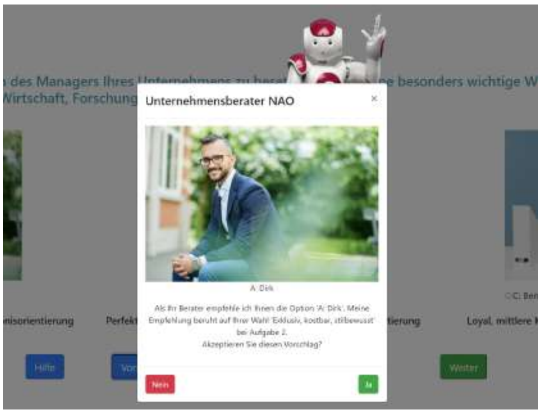


Figure 7.7.: Illustration of the development process for modelling user trust.

presented in this work: First, serious games are highly motivating for users and foster engagement and intrinsic motivation (Abt, 1970). Engaged users are required for taking the game and the assistant's actions seriously. In doing so, an environment of risk and vulnerability was intended to be created. Thus, trust in the system could be developed or destroyed depending on the agent's actions in such an environment. Secondly, testing and evaluating policies (or in our case dialogue strategies) in the real world is too expensive and cumbersome. For this reason, serious games provide a simulated reality based on reduced-scale models for allowing problem-solving (Abt, 1970). Hence, such games enabled the evaluation of the consequences of alternative dialogue policies on the HCT in different situations, promoting the development of data-driven adaptive strategies.

The data acquisition itself was conducted online with crowd workers, that interacted with the system and provided annotations regarding their perception of the system at defined time steps. Objective features were automatically collected by the system itself and combined with user annotations to be written into a database. In the following sections, a detailed description of the serious game scenario and an overview of the prototypical system is provided. A role-playing game was selected as a scenario, in which a user took the role of the CEO of a high-tech company that develops, produces, and sells electrically powered cars. The user's goal was to successfully manage the company by executing strategic actions to maximise profits. In doing so, users had to make step-by-step decisions and plan undertakings in the interest of the company, such as location planning or personnel management. Individual decisions had consequences and affected the success of the management.

The game was designed as a turn-based decision-making task, in which the system sequentially presented a task step and the available choices, whereas the user could take different actions and cooperatively solve the task with a CA. Hence, the task structure of the game resembled that of a system-directed dialogue in which both dialogue participants took turns, i.e. the system took an action providing task step relevant information, upon the user took an action solving the respective task step. The game ended after a total of



Als Ihr Berater empfehle ich Ihnen die Option 'A: Dirk'. Meine Empfehlung beruht auf Ihrer Wahl 'Exklusiv, kostbar, stilbewusst' bei Aufgabe 2.
Akzeptieren Sie diesen Vorschlag?

Figure 7.8.: Illustration of the proactive assistant and its suggestion during the CEO-game. (Kraus et al., 2021c)

12 task steps. The order of the tasks was fixed and could not be altered by the user. For each step, several options from which the user had to select were presented. The number of options changed from task to task ranging from a minimum of three to a maximum of five options. The purpose of this was to vary the complexity of each task to influence the user's perceived task difficulty, as this showed to affect the perceived trustworthiness of proactive behaviour in our previous experiments.

At each task step, users could execute four actions: select an option without system assistance and continue with the game, explicitly ask the CA for a suggestion, or ask for help. By asking for help, general information about the game is provided, e.g. which previous decisions need to be considered at the current task step.

By asking the assistant for a suggestion, appropriate advice was provided. The user could either accept or decline the system's proposal. When an answer option had been selected, the user could continue with the game.

The success of the user's decision-making was measured by attaching numeric scores to the individual selection options. This allowed previous decisions to directly influence the value of future actions. Consider the following example: User Alice is currently required to decide on task "Research", where a plausible research direction concerning the built-up company needs to be chosen. This task is influenced by Alice's selections in previous tasks "Management" and "Banking". Depending on the combination of selections in respective tasks, one of the four options (Hydrogen Drive, Autonomous Driving, Battery Research, Climate Neutral Production) would yield the most points, whereas in the worst-case scenario a user would yield zero points minimum. The concept of a game score should create a vulnerable but also engaging environment for the user. Thus, performance was used as an incentive for the users to take the game seriously. The game score was based on an artificial scoring model, particularly developed for this application.

During the game, the user was supported with decision-making by a CA. The agent was introduced to the user as an AI-driven virtual assistant called Nao, who would provide business advice. An anthropomorphised assistant was chosen to form a clearer separation

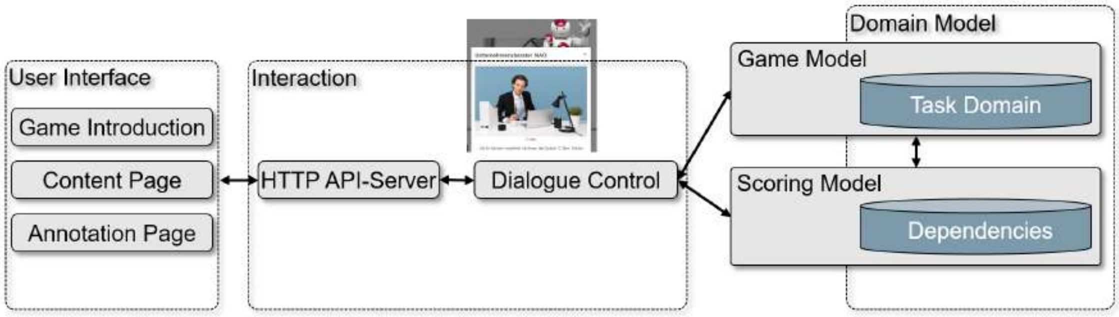


Figure 7.9.: Overview of the system's architecture for creating a data collection environment. (Kraus et al., 2022b)

of task and assistance technology compared to using simple pop-up messages. For this, the assistant was presented as a picture of the famous humanoid robot Nao from Softbank Robotics. Nao could either provide suggestions actively or reactively. A depiction of the CA and its proactive text messages is presented in Fig. 7.8.

Nao was designed to be an expert system avoiding the unintended side effects of incompetent system behaviour on its trustworthiness. This allowed us to only consider the effects of the proactive levels on the HCT. For selecting the best option per task step, the assistant made use of a simple reasoning mechanism by knowing past user selections and accordingly querying the game's scoring model. Further, proactive explanations were added to justify the behaviour of the system to take the initiative. For creating the explaining messages, a template-based approach was used. The relation between the best option and previous user selections that led to finding this option was exploited to include information about past user behaviour in the explanation. This information was transformed into natural language and wrapped into a predefined sentence template. For example, "As your adviser, I recommend option A. My recommendation is based on your choice of B in Task C, whose characteristics of D best fit our concept".

7.2.2. System Design

For creating the virtual CA for the CEO game, we relied on the proposed cognitive architecture for proactive dialogue. The CEO game was implemented as a servlet application based on a client-server model. On the client-side, a user played the game and interacted with the proactive assistant using a clickable GUI. On the server-side, a dialogue control logic received user input from the GUI and provided task-related content to the interface by accessing a domain model. A JSON-based database served as the domain model for the application and contained the complete game content and structure as well as the scoring model. This entity was used to simulate the planning and reasoning components that would be used in high-fidelity prototypes Information between GUI and dialogue control was exchanged using HTTP client requests and Javascript forms. The system's architecture is visualised in Fig. 7.9. The individual constituents are described in the following.

User Interface

The GUI was created as an HTML/Javascript-based web page. The web page's content was created dynamically on the fly for each task retrieving content from the database through the dialog control. In general, the tasks were presented on the GUI using the title and number of the current task, a task description, and the different task options (name, image). Users could interact with the GUI using its action buttons (help, suggest, continue). The action buttons were blocked for 20 seconds for providing the user with enough time to read the information about the task and to guarantee that the user received the system's proactive messages which were triggered after the same amount of time. In this way, the CA did not interfere with the user's process of getting familiar with the task description.

Interaction

The dialogue control logic was implemented as an HTTP web server. It was responsible for controlling the (proactive) interaction with the user and stored information of a game session while interacting with a game-specific database for retrieving relevant domain information. For receiving information from the GUI and to read/write data to a database, the server made use of the typical life-cycle methods *init*, *doPost*, and *doGet*. The mode of operation of each function is described in the following:

init: This method was only called once at the initialisation of the web page and used for retrieving the game content and the scoring model from the database.

doPost: This function was called at the start of the game and iteratively after the user had clicked the 'continue' button to progress with the game. At the game's beginning, this method initialised all parameters and randomised the timing of the proactive dialogue strategies. Additionally, it provided the GUI with the task content and used methods for automatically determining the best option depending on the user's past behaviour and created the corresponding explanation. This behaviour was then repeated for each task. After the user clicked the "continue"-button on the GUI's game web page, a JavaScript form containing objective user-related data was passed to this function. Subsequently, the function stored the data as an attribute of the current game session in the JSON-format. Besides the method redirected the user to the rating web page after every three task steps for letting the user rate the CA's trustworthiness.

doGet: For storing the user's self-reported user data, this method was called after the user had clicked the "continue" button on the rating web page. Equally to clicking this button on the game web page, a form was conveyed to this function which in turn stored the data as JSON in the session attributes. After the fourth and final user rating, this function collected the objective and self-reported user data. Both data types were stored in a JSON file. The information was then written to a corpus database and the user was redirected to the game's ending web page.

Domain Model

The database contained models for the game content and structure as well as for the scoring model, both defined in the JSON format. The game model consisted of a sequence of task steps comprising relevant task information. Each option possessed the option's name, supplementary information, and a file path to a depiction of the option. In correspondence to the game model, the scoring model was also constructed as a sequence of task steps. However, the individual task steps comprised information about the influence of previous decisions on the options of the current task step. This concept was called "Dependencies". If one of the options was positively influenced by a previous selection it was valued with a score of 10, otherwise, it received a score of 0. This composition allowed the dialogue control to determine the best selection of the current task step concerning previous choices, which was then used as content of the proactive or reactive assistance.

7.2.3. Implementation of Proactive Dialogue Behaviour

Potential proactive behaviour by Nao was triggered after 20 seconds into the task step to avoid disturbance of the user in getting familiar with the specific problem. Transferring the framework of proactive dialogue act types into this particular use case scenario, NAO possessed the following four levels of proactivity:

None: Only reactive behaviour was expressed. In case the user requested a suggestion by the system, assistance was provided as if the *Suggestion* action was triggered. For example, according to the "Research" task example, the system would respond to Alice: "As your adviser, I recommend option 'Autonomous Driving'. My recommendation is based on your choices in Task 'Management' and 'Banking', whose characteristics best fit our concept. Should we take this action?".

Notification: Here, the system would utter: "As your adviser, I have a suggestion.". After this prompt, the user could either ignore the message or proceed by selecting the "Learn more" button.

Suggestion: The system provided a suggestion: "As your adviser, I recommend option 'Autonomous Driving". My recommendation is based on your choices in Task 'Management' and 'Banking', whose characteristics best fit our concept. Should we take this action?". As a response, the user could either decline or accept the suggestion.

Intervention: The system took the action out of the hand of the user: 'As your adviser, I have chosen the option 'Autonomous Driving'. My decision is based on your choices in Task 'Management' and 'Banking', whose characteristics best fit our concept". Afterward, the game proceeded to the next task step.

The dialogue flows of the different actions are depicted in Fig. 7.10. For gathering a sufficient amount of data, the proactive dialogues were initiated at random task steps using a restricted randomising policy for ensuring naturalness. The policy restricted proactivity to occur only on four out of twelve task steps, as too frequent system interventions

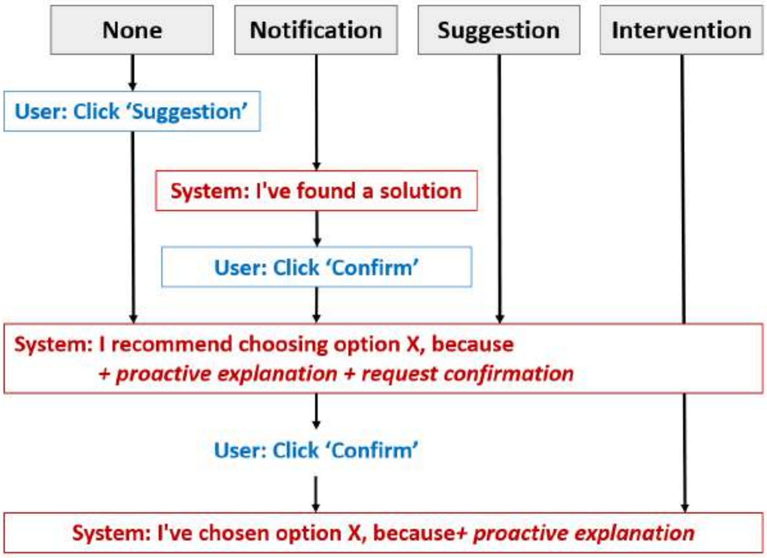

None
Notification
Suggestion
Intervention
User: Click 'Suggestion'
System: I've found a solution
User: Click 'Confirm'
System: I recommend choosing option X, because + proactive explanation + request confirmation
User: Click 'Confirm'
System: I've chosen option X, because+ proactive explanation

Figure 7.10.: Flowchart visualising the dialogue content of different levels of proactivity. User utterances are coloured in blue, while system actions are red-coloured. (Kraus et al., 2021c)

were deemed unrealistic and annoying in a counseling scenario. Further, proactivity was restricted to occur only once within every three task steps for facilitating the annotation process. The annotation process is described in detail in the next section.

7.2.4. Annotation Process

The annotation process was divided into two phases. First, users provided anonymised personal information by answering a questionnaire.

Secondly, for collecting data on the perception of the proactive dialogue assistant, users were instructed to rate their experience with the system, e.g. trust, competence, and user satisfaction, after every three tasks. By not measuring trust at each step, it was supposed to prevent survey fatigue of the users and preserve the participants' cognitive loads.

The CA was designed to become proactive at one task step during each segment of three tasks. This was intended to capture the effect of one specific proactive level at a time. The specific task step and proactive dialogue act type were selected randomly by the system using a uniform distribution.

During the decision-making users were unaware of the consequences of the respective option selection, nor did they know about the expertise of the CA. This method was intended to ensure the vulnerability of the user towards the assistant and let the user self-explore the abilities and usefulness of the system.

For helping the users to rate their experience with the assistant, the outcome of their choices was presented in the form of a game score after every three steps. The user experience ratings were then attached to the previous three exchanges. For obtaining a

User-Dependent Features	*Context-Dependent Features*	*Target Features*
Age	Proactive Dialogue Strategy	Trust
Gender	Task Difficulty	Competence
Technical Affinity (Karrer et al., 2009)	Task Complexity	Reliability
Trust Propensity (Merritt et al., 2013)	Task Duration	Understandability
Domain Expertise	User Selection	Acceptance
Big 5 personality traits (Rammstedt et al., 2013)	Suggestion Request	Annoyance
	Help Request	User Satisfaction

Table 7.3.: Overview of the collected features using the described data collection method. (Kraus et al., 2022b)

sufficiently large distribution of different proactive actions across all steps, a high number of user interactions was required. Objective annotations, like task success, duration, or the user's actions taken, were captured by the system itself at each task step.

7.2.5. Corpus Information

In this section, the method for data collection is described in detail. Information about corpus-formatting and associated parameters as well as a summary of collected data is presented.

Corpus-Formatting and -Parameterisation

A corpus for the creation of user-adaptive proactive dialogue was generated by collecting data from user interactions with the described CA during the CEO game. The data collection was centred around features known to be affecting the user's trust in the CA. Characteristic features from previous work on trust in automation and HCI in general were selected. Based on literature research, 10 user-dependent and 7 context-dependent features were considered for predicting the user's trust. An overview of the features is described in Table 7.3. User-dependent features were collected using a questionnaire before the user had started the game and therefore remained static throughout playing the game. In the questionnaire, users could state their age by providing a numeric value. Three options (female, male, diverse) were possible for expressing the gender. Personality information was collected using the BFI-10 scale developed by Rammstedt et al. (2013). Affinity towards technical systems was assessed using the TA-EG-scale comprising six statements designed by Karrer et al. (2009). For measuring domain expertise, we developed our questionnaire consisting of three items for checking the user's experience in management. Further, propensity towards trust in autonomous systems using the scale by Merritt et al. (2013) was measured to gain information about the user's initial trust. All scales were measured on 5-point Likert scales.

Context-dependent features were collected for each of the twelve task steps. Proactive actions were annotated at each step in the format None, Notification, Suggestion, or Intervention. Perceived task difficulty was self-reported by the user on segment-level, i.e. after three task steps, using a 5-point Likert scale ranging from 1="very low" to 5="very high". Task complexity denoted the number of options of a specific task step and ranged from three to five. User selection indicated the number of points a user received for his or her decision at a task step. The minimum points a user could receive is zero, while it was possible to gather a maximum of 40 points for one decision. Task duration was measured in seconds for each task step. When the user triggered a suggestion or a help request for a certain task step, either a 1 (= action triggered) were annotated. Otherwise, a 0 (=action not triggered) was noted.

The target variable trust was measured on a 5-point Likert scale after each segment of the game for the reasons described in the previous section. The scale ranged from 1="very low" to 5="very high". The annotated trust value was also applied to the previous three task steps. We deemed trust to stay invariant during this time frame as only one proactive action was triggered. Additionally, the user's perceived competence, predictability, and reliability to represent the user's cognition-based trust (Madsen and Gregor, 2000) were annotated. This was due to our previous study identifying cognition-based trust to be a useful adaptation criterion for adapting the dialogue.

Data Summary

Data collection was conducted using the German clickworker platform. Eligibility conditions required users to be aged between 18 and 60, to be a native speaker of German, and to play the game on a desktop computer for compatibility reasons. In total 320 participants were recruited. However, twelve had to be excluded due to violation of instructional terms and technical errors resulting in a final number of 308 users for data collection. In advance of the start of the game, users were briefed about details of the data survey, e.g. duration (20 minutes) and purpose of the survey. Further, participants were informed that concentration checks were included in the ratings to take the game and the evaluation seriously. When users did not pass the checks they did not receive their reward. Participation was compensated with a monetary reward of 3 €. Further, the actions buttons were blocked for 20 seconds to avoid users clicking through the tasks. The details of the corpus are depicted in Table 7.4. Overall, the agent was rated generally as trustworthy with 52 % of the system-user exchanges being labeled with "High" or "Very High". Consequently, the expert assistance system was able to provide adequate help as was expected per design. More interestingly, even though the assistant always provided a correct suggestion, still 11 % of the system exchanges were rated with below neutral trustworthiness. This could be explained either by inappropriate proactive system behaviour or by a user's general low tendency to trust a technical system irrespective of its capabilities. However, the tendency to use the agent for help was evident by considering the number of suggestion clicks. Requests for the system's suggestion messages were used in 43 % of the dialogue. Hence, this may be more related to the random dialogue strategy of the system.

#Dialogues	308
#System-User Exchanges	3696
Avg. Dialogue Duration in seconds	492 s ± 191
Avg. Duration System-User Exchange in seconds	41 s ± 16
Avg. Perceived Task Difficulty	2.6 ± 0.6
Avg. #Help Clicks	0.6 ± 1.8
Avg. #Suggestion Clicks	5.2 ± 3.3
Avg. #Total Points	154 ± 28 /210
#Proactive-None	2523
#Proactive-Notification	364
#Proactive-Suggestion	419
#Proactive-Intervention	390
Avg. User Age in years	37 y ± 11
#Male	194
#Female	113
#Other	1
Avg. Technical Affinity	4.0 ± 0.5
Avg. Experience Management	2.9 ± 1.0
Avg. Propensity to Trust	3.5 ± 0.7
#Trust-Very Low	69
#Trust-Low	336
#Trust-Neutral	1242
#Trust-High	1707
#Trust-Very High	342

Table 7.4.: Descriptive statistics of the generated corpus. Counts are symbolised with the prefix #. (Kraus et al., 2021c)

Requests for help regarding the principle of the game were used rarely (5 % per dialogue). This indicated a clear and understandable design of the developed dialogue game.

For evaluating the usefulness of the collected data, we investigated whether there existed the same correlations between user- and system-related factors in the corpus as found in related work. In line with related work, the user characteristics that correlated the most with trust were the user's propensity to trust a technical system ($r = 0.32,\ p < .001$) and technical affinity ($r = 0.23,\ p < .001$), e.g. see Merritt et al. (2013) and Kraus (2020). While the user's age ($r = -.005,\ p = 0.76$) and gender ($F = 0.84,\ p = 0.36$) did not generally correlate with trust and its related concepts (also mentioned in Hoff and Bashir (2015)), the domain expertise of an individual user showed a significant relationship ($r = 0.13,\ p < .001$). In contrast to related work (Sanchez et al., 2014), where a higher domain expertise related to a lower trust in the technical system, a positive correlation was found in this corpus. Further, significant correlations between the user's personality and the HCT were discovered. In line with related work, a positive correlation between extraversion (Evans and Revelle, 2008) ($r = 0.12,\ p < .001$), agreeableness ($r = 0.09,\ p < .001$), and conscientiousness ($r = 0.16,\ p < .001$) (Chien et al., 2016) and trust was found. Additionally, a negative correlation between neuroticism and trust was found ($r = -0.10,\ p < .001$). This was also indicated by a previous study of Evans and Revelle (2008). As the corpus was annotated with only one trust value, including more items may contribute to the robustness of the target variable. Thus, we tested the monotonicity between trust and respectively competence ($r = -0.73,\ p < .001$), reliability ($r = -0.70,\ p < .001$), and predictability ($r = -0.21,\ p < .001$). The results showed that a combination of the variables should have more predictive power, especially since cognition-based trust was identified to be highly influenced by the levels of proactive dialogue.

The proactive actions did not differ significantly regarding their influence on the system's perceived trustworthiness ($F = 0.98,\ p = 0.40$). Consequently, there seems to exist no "one size fits all" solution to designing proactive dialogue strategies. However, this could be expected as we randomly triggered the proactive actions without taking into account user features or context. Later in this thesis, the personal and context information gathered was used for creating a user simulator (see Section 7.3). The simulator was intended to train appropriate proactive dialogue strategies.

Furthermore, the results highlighted the importance of the system's perceived competence, reliability, and understandability for the HCT relationship in short-term interactions with proactive agents as they showed comparable correlations to features that contribute to trust. Therefore, we deemed the modeling of a user's cognition-based trust to be relevant for adequately predicting the user's perceived trust in a virtual CA.

For predicting the user's perceived trust in the system at each task step, feature engineering and several ML algorithms were applied to the presented corpus. In the following, we describe in detail the steps taken for adequately modelling and predicting trust based on this corpus.

Features	**Trust**	**Competence**	**Reliability**	**Understandability**
Trust Propensity	$r = 0.32$ $p < .001$	$r = 0.30$ $p < .001$	$r = 0.29$ $p < .001$	$r = 0.05$ $p = .002$
Technical Affinity	$r = 0.23$ $p < .001$	$r = 0.20$ $p < .001$	$r = 0.22$ $p < .001$	$r = 0.10$ $p < .001$
Age	$r = -.005$ $p = 0.76$	$r = 0.02$ $p = 0.19$	$r = -0.03$ $p = 0.12$	$r = 0.02$ $p = 0.30$
Gender*	$F = 0.84$ $p = 0.36$	$F = 1.38$ $p = 0.24$	$F = 0.15$ $p = 0.70$	$F = 4.35$ $p = 0.04$
Domain Experience	$r = 0.13$ $p < .001$	$r = 0.08$ $p < .001$	$r = 0.08$ $p < .001$	$r = 0.25$ $p < .001$
Neuroticism	$r = -0.10$ $p < .001$	$r = -0.12$ $p < .001$	$r = -0.10$ $p < .001$	$r = -0.05$ $p = .002$
Agreeableness	$r = 0.09$ $p < .001$	$r = 0.13$ $p < .001$	$r = 0.11$ $p < .001$	$r = -0.06$ $p < .001$
Conscientiousness	$r = 0.16$ $p < .001$	$r = 0.17$ $p < .001$	$r = 0.16$ $p < .001$	$r = -0.02$ $p = 0.28$
Extraversion	$r = 0.12$ $p < .001$	$r = 0.14$ $p < .001$	$r = 0.12$ $p < .001$	$r = -0.03$ $p = 0.06$
Openness	$r = 0.03$ $p = 0.07$	$r = 0.02$ $p = 0.17$	$r = 0.04$ $p = 0.02$	$r = 0.04$ $p = .008$
Proactivity*	$F = 0.98$ $p = 0.40$	$F = 0.37$ $p = 0.77$	$F = 0.62$ $p = 0.60$	$F = 0.19$ $p = 0.91$

Table 7.5.: Correlation between the corpus features and the trust-related target variables. Correlations measured using spearman's r, except where noted. * indicates usage of a one-way ANOVA for comparing the effect of categorical values on the target variables.

7.2.6. Predicting Trust for Proactive Dialogue Adaptation

Analysis of the corpus revealed several outliers concerning the duration of an exchange. For example, some exchanges lasted for over eight minutes, while the average duration of a system-user exchange was 41 seconds. A reason for this was that the proactive dialogue game was performed using a web browser. Therefore, users could interrupt the game at any time to resume later on. Hence, such outliers indicated user disengagement from the crowdsourcing task. This possibly negatively influenced the user's trust annotation and created noisy data. Therefore, exchanges with a duration of over two minutes as well as the following exchanges of the particular dialogue game were discarded. In doing so, only coherent user engagement was aimed to be represented in the data. This resulted in 3161 usable exchanges for online trust prediction. As features for the prediction, we selected all corpus parameters. This was grounded on the findings of the corpus analysis and the preceding considerations in related work.

The corpus parameters comprised numerical as well as categorical values. For representing all parameters as a numerical input vector, categorical features were encoded in a one-hot vector. As the numerical values were either measured on ordinal scales (e.g. trust propensity) or metric scales (e.g. age), these values were standardised for comparability using z-transformation. Encoded categorical and standardised numerical parameters were then concatenated to a feature vector totalling 27 entries for each exchange.

As trust is a dynamic variable (Hoff and Bashir, 2015) and depends on previous interaction with the proactive agent, each feature vector was enriched with temporal information. Therefore, the non-static, i.e. interaction parameters, were used to artificially create temporal features. This was conducted by taking means from the turn-based information for a window of the last three system-user-exchanges and the complete dialogue up to the current exchange (see Fig. 7.11). This approach is similar to the modelling of temporal information for IQ estimation (Ultes et al., 2015).

The total size of the feature vector containing personal user parameters PUP (12 entries), interaction parameters IAP (15 entries), and temporal interaction parameters TIP (30 entries) amounted for 57 features.

For considering the effects on short-term trust during the action, we calculated a combined target variable as the rounded mean of the labels for trust, competence, reliability, and predictability. This value ranged between 1 and 5 as a result of using a Likert scale. As we modelled trust and its related concepts on a discrete, ordinal scale, the prediction problem was formulated as a multi-class classification task. The target classes were the distinct trust values. By combining the variables the prediction was also supposed to be more robust to outliers, i.e. noise. However, this also contributed to the skewness of the distribution of the labels towards labels 3 and 4, while the other labels were annotated significantly less. For facilitating training on an imbalanced data set, class weights were balanced before classifier training. Three machine-learning approaches were compared for solving the prediction task; an SVM, XGB, and a GRU-based RNN approach. The SVM was trained using all parameter sets $\{PUP,\ IAP,\ TIP\}$. For training the XGB, input variables were the same features from the described groups as for the SVM. Contrary to these static approaches, where temporal features were required to be modelled by hand,

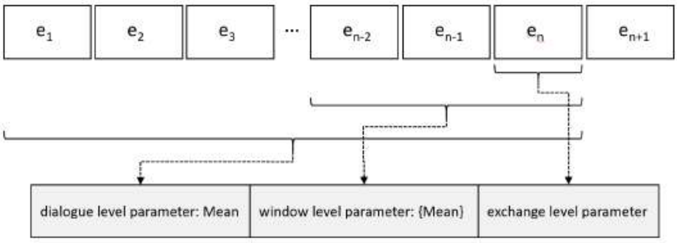


Figure 7.11.: Modelling of temporal information in the interaction parameters used as input for trust prediction.(Kraus et al., 2021c)

also a sequential approach using a RNN was implemented. In doing so, temporal information could be learned automatically instead of being provided manually (Rach et al., 2017; Ultes, 2019). Therefore, only the exchange level parameters were used as input for the network $\{IAP\}$. For allowing trust estimation using an GRU approach, the exchange level parameters of a particular time step t were required to be included in a sequence. Therefore, the classification problem was transformed in such a way that the trust value at time t was estimated for the corresponding sub-dialogue sequence consisting of all exchanges from the beginning up to t. For guaranteeing a consistent sequence length of twelve, which is congruent to the number of total dialogue steps of the game, future information beyond time step t was encoded using zero vector padding.

Experiments and Results

For comparison, all classifiers were trained and tested on the previously described corpus. The SVM was implemented using the scikit-learn SVC library based on LIBSVM (Chang and Lin, 2011). The deep neural net model was implemented with Keras (Chollet, 2015), while the XGB model was implemented using the XGBoost 1.3.3 library.

As evaluation measures, the F_1-score, UAR, linearly weighted Cohen's κ, and Spearman's ρ were used. For deciding on an adequate proactive action based on the user's trust level, it was crucial to also consider the extreme cases, i.e. a trust level of 1 and 5. As these classes were underrepresented in the corpus, we optimised the classifiers regarding F_1-score and UAR for classifying all labels as correctly as possible and not only the majority classes.

As trust was measured on an ordinal scale, the distance between a wrong prediction and the real class was important, particularly given a real-life application. Therefore, the number of guesses in which the classification was wrong only by one class, e.g. an instant of trust 4 classified as trust 5 or vice versa, was computed (Rach et al., 2017; Ultes, 2019). Therefore, the extended accuracy metric eA was used. Due to the novelty of our approach, the evaluation baseline was a random prediction ($F_1 = 0.2$, UAR $= 0.2$, $\kappa = 0.0$).

Experiments were conducted using a dialogue-wise and a classical stratified 10-fold cross-validation setup. First, a dialogue-wise setup was employed to optimise the hyperparameters of each classifier on 10 folds of disjoint sets of dialogues.

		$F1$	UAR	κ	ρ	eA
SVM	All	**0.415**	0.512	**0.224**	0.293	0.867
SVM	PUP	0.332	0.506	0.182	**0.294**	0.852
SVM	IP	0.256	0.388	0.143	0.208	0.809
XGB	All	0.377	**0.529**	0.212	0.279	0.854
XGB	PUP	0.316	0.510	0.178	0.262	0.821
XGB	IP	0.269	0.438	0.139	0.192	0.800
GRU	All	0.381	0.479	0.198	0.282	**0.875**
GRU	PUP	0.273	0.472	0.143	0.228	0.803
GRU	IP	0.211	0.436	0.096	0.140	0.772

Table 7.6.: Dialogue-wise Cross-Validation.(Kraus et al., 2021c)

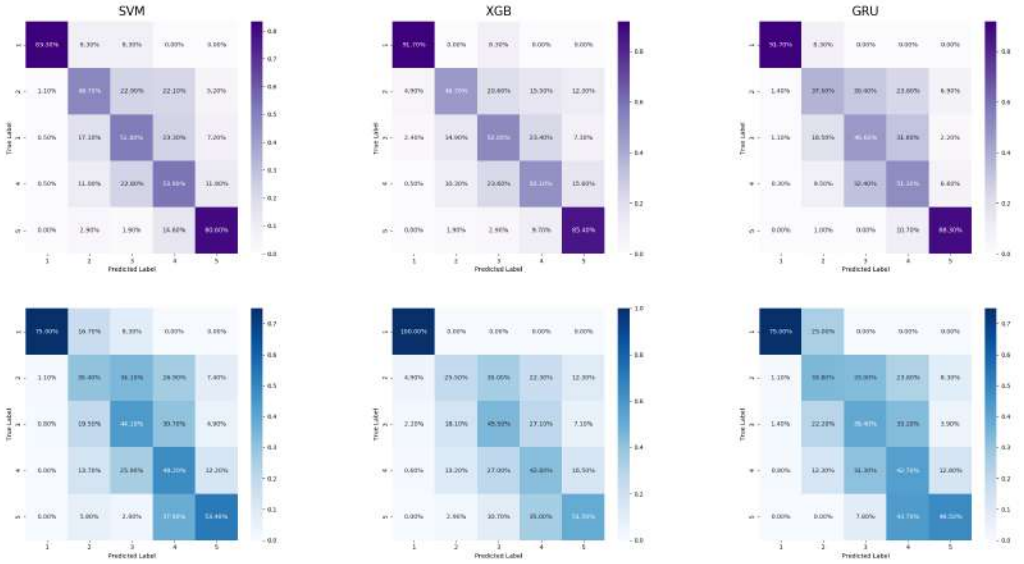


Figure 7.12.: Confusion matrices for each classifier. The top row shows the classification results using the classical cross-validation setup, while the bottom row shows the results using the dialogue-wise setup. The percentages of the class-wise recognition results are colourised.(Kraus et al., 2021c)

	F1	*UAR*	κ	ρ	*eA*
SVM	**0.533**	0.654	**0.363**	**0.426**	0.895
XGB	0.435	**0.660**	0.342	0.399	0.879
GRU	0.465	0.633	0.296	0.379	**0.906**

Table 7.7.: Classical Cross-Validation. (Kraus et al., 2021c)

In doing so, the classifiers were tested on completely new dialogues as opposed to potential overlapping sub-dialogues in the training and test sets using the classical approach. Hence, the classifiers' performances were deemed to be better generalisable. A stratified approach was used due to imbalanced trust labels. Hyper-parameter optimisation was conducted using a combination of grid and heuristic search. The optimal model of the SVM was constructed with a radial basis function as kernel, a regularisation parameter $C = 1$, and a scaled γ. The optimal XGB model was trained against mlogloss using $n = 1000$ estimators with a maximum depth of 3 and a learning rate of 0.01. The GRU-model was trained against cross-entropy loss using the Adam optimiser (Kingma and Ba, 2014) with a learning rate of 0.0001 and a mini-batch size of 16, run for a total of 100 epochs. The GRUs consisted of 60 neurons and were each followed by a dropout layer. The results are presented in Table 7.6.

To get an understanding of the impact of solely personal user information and (temporal) interaction parameters on the classification performance, evaluation results considering only these parameters as input variables are also provided. However, the results of the dialogue-wise setup were too pessimistic, as a scarce amount of dialogues (< 300) was used. Furthermore, the minority classes were underrepresented in the training and test sets due to imbalanced data, complicating the training procedure. Therefore, the optimised models on the dialogue-wise setup were also evaluated using the classical cross-validation setup. These results are visualised in Table 7.7.

7.2.7. Discussion

In the following, we discuss the results of the experimentation for developing a trust estimator using different machine-learning predictors. Here, we emphasise whether trust can be predicted accurately using the proposed model. Further, the limitations of the data collection method and the trust model are explained.

Predicting Trust using Statistical Models based on Context and User-Related Features

The results showed that each classifier performed well on the given prediction task, clearly outperforming the random baseline. The SVM model performed best overall by achieving an F_1-score of 0.533, κ of 0.363 and a ρ of 0.426. XGB provided the best result for UAR with a value of 0.660, while the GRU network showed the best accuracy with an *eA* of 0.906. An explanation for the advantage of the SVM over the other approaches could be that deep learning approaches and boosted trees typically require large data to perform well (LeCun et al., 2015). However, the used data set was quite scarce which favoured the usage of an SVM. Overall, all classifiers outperformed a random baseline, which validates the applicability of the developed user model for predicting trust during interaction with a proactive dialogue agent. Furthermore, the presented approach allowed us to predict well the two extremes of the combined trust scale, i.e. 1 and 5, with an accuracy of over 80 % in the classical cross-validation setup. These two trust levels were supposed to be especially useful when considering trust for developing proactive dialogue strategies, as they were the best indicators of why a specific proactive strategy failed respectively succeeded.

A further finding of the experiments was that personal user information had more impact on the trust prediction than the interaction parameters. This was not surprising, as the personal user information features could be assigned to two (dispositional and situational) of the three trust layers according to Hoff and Bashir (2015). Therefore, these features represented the user's tendency to trust in general. Contrary, the interaction parameters represented the third layer – learned trust in the system – which is subject to change during interaction depending on the system's performance and design. Hence, these features measured the subtle changes in the user's trust in the proactive agent. However, combining both features showed the best results as we expected.

As all classifiers achieved an eA of at least 0.879, i.e. over 88 % of the classifier's guesses were usable, the proposed trust prediction model may be useful for application in real-life scenarios. For example, Ultes (2019) used an BiLSTM-based IQ predictor with an eA of 94 % for modelling the reward function of an RL approach for training user-adapted dialogue policies. Similarly, the presented trust prediction model could be used to develop proactive dialogue strategies. This can be used to design trustworthy CAs in various application scenarios where users collaborate with a system on a certain task, e.g. decision-making or recommendation systems. For implementing a proactive DS based on a user trust model, several user-, and situation-specific features needed to be acquired. Some user features are hard to obtain in real-world applications but may be collected implicitly, e.g. personality data, or by asking the user explicitly for the information. In doing so, it may be possible to investigate different aspects of proactive behaviour and their effect on the HCT relationship.

Limitations

A limitation of the presented data collection method was that only the assistant itself used natural language for communication, while the user interacted via actions buttons that trigger predefined utterances. Allowing users to interact with the interface using text or even speech input would create another great possibility to capture relevant features (lexical, linguistic, etc.) for predicting the effect of the proactive actions. However, a more complex communication channel would also add noise and increase the possibility of failures, which are independent of the actions and only related to the system's performance regarding speech recognition and understanding. A more restricted input channel was beneficial for establishing safe communication between assistant and user. Another drawback was that a perfect system endowed with expert knowledge was used. In a real-world scenario, a system that always provides the best counseling is unrealistic. However, as the proactive behaviour was randomised and limited to a reasonable frequency, a certain naturalness was added to the agent, as absent system activity could have been perceived as unknowing behaviour. In future work, an error model could be included in the system to simulate not ideal counseling. Here, it may be interesting to study whether proactive behaviour remedies a low system performance. A limitation of training the classifiers was that they were trained on a relatively scarce data set. Training with more data could provide more generalisable results.

7.2.8. Conclusion

For implementing a trust recognition module, several individual processing steps were necessary. First, we described a method for creating a corpus of proactive dialogue using a human-to-machine approach. Therefore, an autonomous assistant embedded in a serious game scenario was developed and implemented as a web service. Data from 308 dialogues were collected via crowdsourcing and annotated with several user-dependent, context-dependent, as well as several target variables, whereas the focus was set on the HCT relationship. Analysis of the corpus revealed the usefulness of the collected data and the necessity to consider proactive actions in combination with user characteristics and personality when developing trustworthy strategies. Using the rich feature pool of this corpus allowed us to develop a novel user model for predicting trust in interactions with a proactive CA. Therefore, trust parameters were categorised into user-, system-, and context-dependent features. This allowed us to predict trust online during interaction with a virtual CA using the developed set of proactive dialogue action types. For predicting trust, three classification algorithms (SVM, XGB, GRU) were trained and tested on the proactive dialogue corpus. The experimental results showed that the model was well-suitable for predicting trust in proactive dialogue. Each applied classifier proved to be useful for the classification task, showing reasonable recall and accuracy. However, an SVM-based model provided the most well-rounded performance. For this reason, we used this model for recognising the user's trust during the interaction. Further, the collected data allowed us to develop a socially-aware user simulator, that was used to create a train- and test environment for developing proactive strategies. This environment enabled the exploration of the effect of different proactive dialogue strategies on the HCT relationship inexpensively and efficiently. The development of the user simulator is described in the following.

7.3. Implementation of a Trust-Aware User Simulator for User-Centred Proactive Dialogue Modelling

For allowing the implementation of a user-centred proactive dialogue model, a corpus-based user simulator was developed. Here, the main objective was to replicate realistic user characteristics, task, and trusting behaviour for training and testing various dialogue policies. Task behaviour implied the actions taken by users on a sub-task level basis and their effects on task duration and the game score. Trusting behaviour implied the user's perceived trust level per task step. The simulation relied on relevant personal and dialogue data gathered from the previously described corpus collection. In doing so, socio-demographic features (age, gender), personality traits, and other user-specific information could be simulated. Further, this enabled us to reproduce user behaviour as a reaction to proactive system actions. Both, simulated user personal information and behaviour, were then used to estimate the current trustworthiness of system behaviour. This in turn allowed the integration of trust in the dialogue state and into a reward function for creating trust-adaptive proactive dialogue strategies. In the following, the architecture of the developed user simulator and an evaluation regarding its realism and usefulness for application is presented.

7.3.1. User Simulator Architecture

User simulation was based on two components: a *user model* and a *user dialogue manager*. The user model contained all the necessary information for modelling distinct user types whose specific task and trust behaviours were imitated. The user dialogue manager was designed as a rule-based agent that triggered various behaviours dependent on the proactive CA's actions and the current task context. First, the user model is described in detail.

User Model

For creating distinct user types, the corpus' user-dependent information was used: age, gender, technical affinity, the propensity to trust, domain expertise, and the Big 5 personality traits. In the first step, random distributions for these variables were calculated. Except for gender, which was randomised based on the gender's likelihood of occurrence in the corpus, all other variables were randomised using truncated Gaussian distributions. Truncated normal distributions were necessary, because technical affinity, the propensity to trust, domain expertise, and personality traits were rated on 5-point Likert scales. Also, a user's age was limited in the data collection process due to the study restrictions only allowing participants between the age of 18 and 60. Our definition of a user's task behaviour comprised the selection of options, which was represented as the game score, help requests about the game, and suggestion requests towards the CA.

While all features of the created user types were used for trust estimation, we only deemed three variables relevant for the specific user's task behaviour: domain expertise, the propensity to trust, and technical affinity. Domain expertise was deemed relevant for task behaviour as it would influence their decision-making. A novice would probably ask more for recommendations than a more experienced user. Propensity to trust was used because we assumed that a user's reactions to proactive behaviour would be dependent on their attitude to trust an autonomous system. For example, a low propensity to trust may lead to rejections of the CA's offers or not asking the system for assistance. Similarly, a user's technical affinity was supposed to influence the decision-making process in collaboration with an autonomous technical system.

For simplifying the selection of specific task behaviour, these three user traits were transformed into binary values. When a generated trait value was above or equal to the threshold of 3 on the 5-point Likert scale, it was represented with a value of “1”. Otherwise, it was represented as a “0”. The purpose of this transformation was to reduce the rule space for simulating more profound user behaviour.

Finally, user-specific task behaviour and the CA’s actions affect the duration and the perceived difficulty of a specific task step. Both task-related variables were randomised also using truncated Gaussian distributions, as the duration of a task step was always greater than 20 seconds and perceived task difficulty was measured on a 5-point Likert scale. The user’s task behaviour was based on a pre-defined rule set. This was generated by a user dialogue manager. The user’s DM process is described in the next section.

User Dialogue Manager

For generating task behaviour, two different approaches were used: *complexity-based* and *task-step-based*. The serious dialogue game consisted of 12 task steps with varying complexities. Complexity in this context meant the number of options from which a user had to select one for decision-making. As previously mentioned, the number of options ranged between three and five options per task in a sequentially repetitive order, i.e. $3, 4, 5, 3, 4, 5$. Using the complexity-based method, the user’s task behaviour was simulated depending on the CA’s action and the complexity of a task step. For example, if the current proactive dialogue act type was *Notification* and the current task step had a complexity level of 3, the simulator would use the corpus data distributions for these specific cases for generating task behaviour.

Contrarily, the task-step-based method incorporated information from a certain task step and the CA’s action. For example, if the current proactive dialogue act type was *Notification* and the user was working on the seventh task step, the simulator would use the corpus’ data distributions for these specific cases for generating task behaviour.

An advantage of including task complexity in DM was that user behaviour could be generated in a more generalised way, and was not dependent on the particular task steps. However, the advantage of the task-step-based method was that sequential dependencies between the task steps could be modelled better. Thus, there existed a certain trade-off between both variants. In the following, both approaches are described in more detail.

For both approaches, user behaviour was simulated by generating values for the game score, whether a user initiated a suggestion or help request, along with the corresponding duration of the task step and perceived task difficulty. The probabilities for each specific user behaviour were based on structured data sets depending on the user model and the current dialogue situation.

First, the overall data set was sorted concerning the occurrences of user behaviour dependent on the relevant user traits. Therefore, the user traits domain experience, propensity to trust, and technical affinity were represented as tuples of three binary values, i.e. “000” to “111”. For example, “000” represented low domain experience, low propensity to trust, and low technical affinity.

Algorithm 2: Pseudo-code for simulating user behaviour dependent on the task-step-based approach.

```
generate user_traits;
load step_based_data;
step_number = 0;
game_ended = False;
while not game_ended do
    receiveproactive_act  step_number + +;
    associatecomplexitywithstepnumber  if step_number == 12 then
        game_ended = True;
    end
    The relevant user traits domain experience, propensity to trust, and technical affinity
     are represented as tuples of three binary values;
    if relevant_user_traits == 000 then
        if step_number == 1 then
            if system_action == None then
                trait_data, step_data = get_data_for_None_1_000;
                if trait_data < 10 then
                    trait_data = fallback
                end
                if step_data < 10 then
                    step_data = fallback
                end
                sugg_request = generate_sugg_request(trait_data, step_data);
                help_request = generate_help_request(trait_data, step_data);
                if sugg_request == Falseandhelp_request == False then
                    duration_data, difficulty_data = get_data_for_None_1_000_False_False;
                    if duration_data < 10 then
                        duration_data = fallback
                    end
                    if difficulty_data < 10 then
                        difficulty_data = fallback
                    end
                    duration = generate_duration(trait_data, step_data, duration_data);
                    difficulty = generate_difficulty(trait_data, step_data, difficulty_data);
                end
                if sugg_request == Falseandhelp_request == True then
                    ...
                end
                ...
                points =
                  generate_points(step_number, trait_data, step_data, sugg_request, help_request);
            end
        end
    end
end
```

Contrary, “111” represented high domain experience, high propensity to trust, and high technical affinity. Afterward, the data set was structured for the individual approach.

According to the complexity-based method, the already categorised user-dependent data was first summarised based on the tasks of the same complexity. For example, user behaviour occurrences from task steps 1, 3, 4, and 7 were summarised, as they all had three options the user could select from. Next, the processed data was summarised according to the used types of assistant proactivity, i.e. *None*, *Notification*, *Suggestion*, and *Intervention*. As the last step, the resulting data was then structured according to the occurrences of help and suggestion requests. These were also represented as binary values. For example, occurrences of both help and suggestion requests were classified as “11”, whereas non-occurrences of both features were labeled as “00”. The task-step-based approach used the same method. However, categorised user-dependent data was here summarised based on the respective task step and not based on the complexity of the task. For example, user behaviour for the first, second, third, etc. task steps was summarised.

Each approach also used fallback data sets in case the occurrences of specific parameters did not exceed a specific threshold. If there was not enough data for a specific user trait, i.e. occurrences for a trait were below 10, then this parameter was omitted and the means and standard deviations or counts of all user traits were used for calculating the probabilities for user behaviour generation. This was also done if the classification depending on the complexity, the step number, and help as well as suggestion requests resulted in a too low number of data for probability generation.

The simulation process using the task-step-based approach is depicted in Algorithm 2. The algorithm for the complexity-based approach was structured analogously. Here, complexity-based data distributions were loaded instead of task-step-based distributions. Further, the if-statement did not consider the specific task step number, but the complexity level. In the following, the algorithmic process is described. First, a user type with specific traits was generated and the approach-specific structured data sets were loaded. Afterward, the dialogue game was initialised and the task steps (1-12) with the respective complexities (3,4,5) were iterated. For each task step, values for help and suggestion request, duration, perceived difficulty, and the achieved game score were calculated depending on the user type and CA’s action. For this, the relevant traits’ categories were queried, e.g. “000”, and the context was determined, i.e. proactive system action and complexity or task step number. Here, it was checked whether the fallback threshold had been exceeded. If this was not the case, the personality traits would be neglected and overall means were used for probability calculation. Afterward, it was simulated whether a help and/or suggestion request would be set. Again, a fallback check for the respective request types was conducted. Depending on the specific case the perceived difficulty, task step duration, and the achieved game score were simulated.

	Complexity-based *M (SD)*		**Task-step-based** *M (SD)*	
	KL	MSE	KL	MSE
Game Score	0.369 (.185)	73.19 (64.6)	0.354 (.166)	70.94 (64.7)
Duration	0.261 (.064)	1722 (844)	0.244 (.079)	1530 (104e1)
Difficulty	0.145 (.011)	1.909 (.155)	0.149 (.008)	1.887 (.217)
Help Request	0.029 (.009)	0.088 (.028)	0.031 (.011)	0.097 (.035)
Suggestion Request	0.084 (.006)	0.352 (.025)	0.082 (.010)	0.337 (.034)
Overall	0.178 (.151)	359.5 (780)	0.172 (.142)	320.6 (765)

Table 7.8.: Descriptive statistics of the KL distances and MSEs for each user simulator type with regard to the measures of game score, duration, help and suggestions request, and perceived difficulty.

7.3.2. Experiments and Results

For deciding which approach to use for training and testing trust-adaptive proactive dialogue strategies, we conducted an evaluation. The evaluation aimed to determine the degree of realism of each user simulation approach and to select the approach that was the closest to realistic user behaviour as seen in the data. For this, we simulated user behaviour using both approaches based on the user types and CA actions that occurred during the data collection process with real users. We then compared the simulated user behaviour with the actual behaviour. For comparison, we used the Kullback-Leibler (KL) (Kullback and Leibler, 1951) distance between distributions of behaviour generated by the user simulator and real users. KL distance is a measure of how one probability distribution Q is different from a second, reference probability distribution P:

$$D_{KL}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \tag{7.1}$$

where the distances range from 0, i.e. distributions are equal, to 1, i.e distributions are completely different. The lower the distance between the distributions, the more realistic is the respective user simulator. For evaluation, we calculated the distances of distributions between the complexity-, respectively task-step-based approach and actual behaviour for each task step. In Table 7.8, the overall mean distances for each task step as well as the individual distances for the game score, duration, help and suggestion request, and perceived difficulty are listed.

7.3.3. Discussion

The results showed that both user simulator types had a comparable performance. There were no significant differences for all measured features (all p-values $p > 0.05$). As can be seen from Table 7.8, the task-step-based approach produced a slightly more realistic behaviour than the complexity-based method. This was mostly due to simulating more realistic game scores and durations of specific task steps. The reason why the task-step approach generated slightly more realistic values may be that the averages of distinct task steps were used to create the distributions. Using the complexity-based method, averages values over four task steps of the same complexity were applied for distribution creation. Help and suggestion requests were simulated almost identically to the observed user behaviour during the data collection. Due to the slightly more realistic results, we selected the task-step-based approach creating the train- and test-environment for the implementation of user-centred proactive dialogue.

7.3.4. Conclusion

Based on the created data corpus, we developed two different types of user simulators. a complexity- and a task-step-based approach. Both types comprised a user model that maintained the distinct user data distributions, either dependent on the complexity levels or the respective turn number of the specific task steps. A rule-based user dialogue manager then generated user behaviour in the form of a game score, the task step duration, perceived task step difficulty, and whether a help or suggestion request occurred. This generation was based on the specific simulated user type and the proactive dialogue act type. As the task-step-based approach generated more realistic user behaviour, this type of user simulator was employed for training and testing *rule-* as well as *RL-based* proactive dialogue strategies. The implementation and evaluation of these strategies are presented in the following.

7.4. Implementation and Evaluation of Trust-Adaptive Proactive Dialogue Strategies

The overall goal of this thesis was to improve the cooperation with CA by utilising user-centred proactive dialogue modeling. For achieving this goal, a CA needs to express task-oriented and also socially effective proactive behaviour. Whether proactive behaviour is socially effective can be determined by estimating the user's perceived trust in the system: inappropriate system proactivity would result in lower trust than adequate behaviour. However, as seen in our previous studies, the development of trustworthy proactive dialogue strategies is quite user- as well as context-dependent. Therefore, we made use of our previously described trust estimation module for integrating trust as a user state in dialogue processing. This allowed the development of dialogue strategies or policies that take into account the user's perceived trust state for socially appropriate decision-making. By also including the current dialogue context and usability measures for action selection, we aimed to improve the proactive dialogue model concerning task success, efficiency, and thus fostering enhanced cooperation with the CA. In the following, we present a *rule-based* and a *RL-based* approach for enabling such user-centred proactive dialogue strategies. Using the *rule-based* approach, an explicit adaptation mechanism was implemented, while the *RL-based* approach used the estimated trust value in the dialogue state as well as in a reward function for implicitly learning an adaptation mechanism.

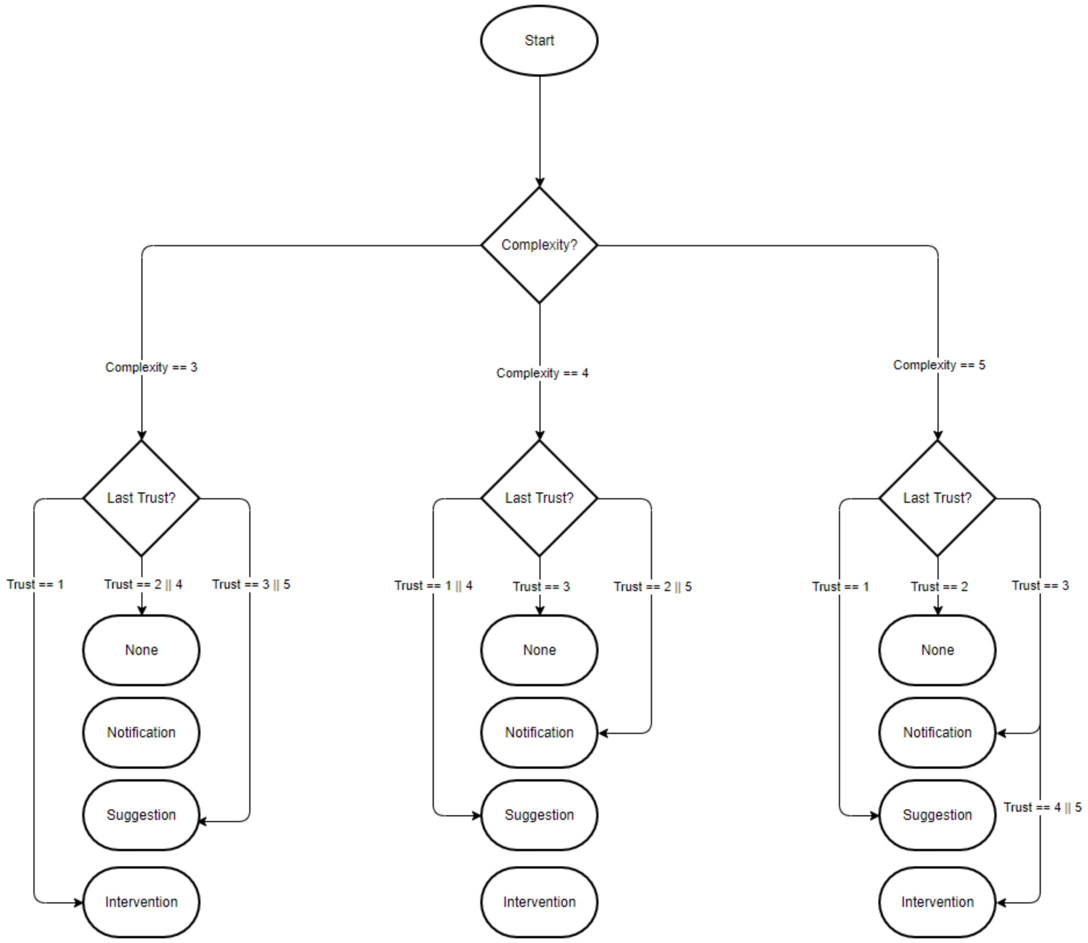


Figure 7.13.: Illustration of the proactive assistant and its suggestion during the serious dialogue game.

7.4.1. Rule-based Strategy

For adapting the dialogue to users and their current situation, adequate adaptation rules are required. Here, we developed rules for proactive action selection dependent on the present dialogue state. The dialogue state for rule-based action selection comprised the complexity level of the current task step and the trust value of the last task step. Task complexity was deemed to influence the need for proactive behaviour from a usability perspective. For example, based on our previous results, a task with high complexity was ought to require a higher degree of system proactivity than tasks with lower complexity. Further, we deemed the last task step's trust value to indicate whether a user wished for proactive system behaviour which influenced the system's trustworthiness. For example, if the last proactive action was perceived as inappropriate by the user and resulted in a low trust value, the system may need to switch its strategy and act according to a lower level of proactivity.

However, hand-crafting rules for these two adaptation criteria were not trivial. Taking into account the four possible proactive system actions (*None*, *Notification*, *Suggestion*, *Intervention*), three different levels of task complexity, and five possible trust values, the amount of valid rules was $4^{15} = 1.073.741.824$. Thus, we used a randomised approach for finding an appropriate rule-based strategy. The approach was inspired by the grid search method for hyper parameter tuning for ANNs. The rule set for selecting actions was modelled using if-else statements. First, we checked the current complexity of the task step, i.e. 3, 4, 5, and then selected one action dependent on the last trust value. In summary, this resulted in 15 rules for selecting a proactive action. For finding appropriate rules concerning the user's perceived trust in the CA, actions were randomly selected for these 15 rules and evaluated with 500 simulated users. To generate a broad set of different rule-based strategies, we applied 100 random action initialisation. Each rule-based strategy was tested on the same set of simulated users to guarantee comparability. Considering the different evaluations, we selected the strategy which resulted in the highest mean value for perceived user trust, by also taking into account the adequate task success and efficiency of this strategy. The *rule-based* strategy that was used for comparisons with the static, random, and the *RL-based* strategies is visualised in Fig. 7.13. Due to the difficulty of finding adequate hand-crafted rules for this problem, we made use of ML techniques to let the CA explore effective strategies during training with the user simulator. This approach is described in the following.

7.4.2. Reinforcement Learning Strategy

RL allows an agent to learn strategies for solving complex problems by maximising a reward. This reward is a feedback signal from the agent's environment for determining the goodness of agent behaviour. To apply a RL-based approach for the adaptation of proactive dialogue behaviour, it was required to model the interaction between the user simulator and the CA as an MDP. Thus, dialogue states, actions, and rewards needed to be defined. Further, the states had to be modelled according to the Markov property, so that the evolution of the dialogue only depended on the present state and not the

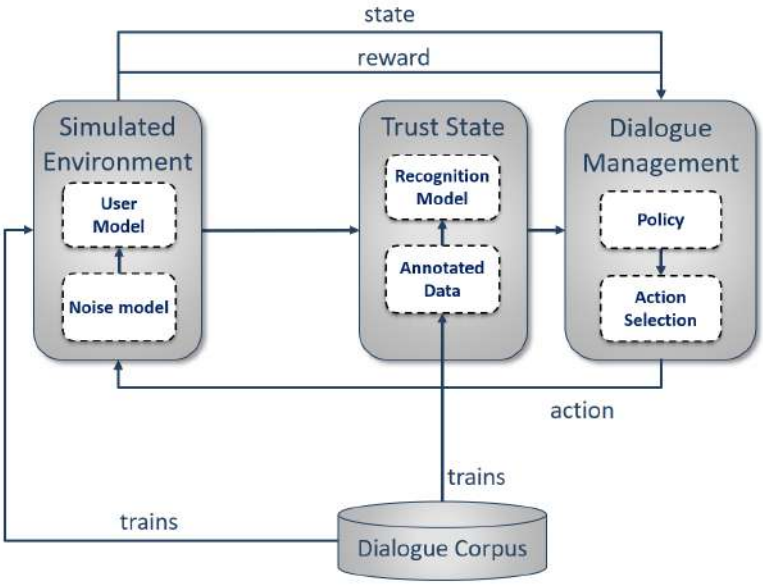


Figure 7.14.: Illustration of the RL approach for trust-adaptive DM. (Kraus et al., 2022d)

history. The RL approach is depicted in Fig. 7.14. For modelling the dialogue state, we included the current task step s_{step} and the level of task complexity $s_{complexity}$. These states represented the agent's static knowledge of the task, as the dialogue game consisted of 12 task steps and three recurring levels of complexity.

Dynamic knowledge was represented by integrating the last known estimated user trust value s_{trust}, as well as the task success $s_{success}$ and the duration $s_{duration}$ of the last task step. These values changed dynamically depending on the context of the interaction. Trust (1-5) and task success (0-40) could respectively take five different values, while duration was measured in seconds and could take 100 different values ranging from a minimum of 20 seconds to 120 seconds. Consequently, the state space

$$S = \{s_{step}, s_{trust}, s_{complexity}, s_{success}, s_{duration}\} \tag{7.2}$$

had a dimension of 90000 different states. For modelling the action space

$$A = \{a_{none}, a_{notification}, a_{suggestion}, a_{intervention}\} \tag{7.3}$$

we relied on our taxonomy of proactive dialogue acts: *None* a_{none}, *Notification* $a_{notification}$, *Suggestion* $a_{suggestion}$ and *Intervention* $a_{intervention}$. The reward function was modelled in order to promote usability-focused and trustworthy proactive behaviour. As trustworthy proactive dialogue behaviour does not necessarily imply task effective or efficient behaviour and vice versa, several aspects were taken into account for designing the reward function. To enable both trustworthy, successful, and efficient proactive dialogue behaviour, the reward was modelled as the sum of the rewards for estimated trust, task success, and task duration

$$r_t = r_{trust} + r_{success} + r_{duration} \tag{7.4}$$

In the following, we present the design of the individual reward functions. The reward function based on the trust rating for each task step was modelled in such a way that high levels of trust, i.e. trust levels 4 and 5, were rewarded with a positive numerical value, while low levels of trust, i.e. 1 and 2, were punished with a negative numerical value. A medium level of trust received zero rewards.

This resulted in the following reward function for training trustworthy proactive behaviour.

$$r_{trust} = \begin{cases} 20 & \text{, if } s_{trust} = 5 \\ 10 & \text{, if } s_{trust} = 4 \\ 0 & \text{, if } s_{trust} = 3 \\ -10 & \text{, if } s_{trust} = 2 \\ -20 & \text{, if } s_{trust} = 1 \end{cases} \tag{7.5}$$

For modelling the reward function based on task step success, we first measured the game score mean for each individual task step taken from the data collection. This was then used as decision criterion for assigning a specific numerical reward. Above average task success resulted in the highest reward regarding task success. The reward value was reduced step-wise for task success scores that were equal to the mean, were below it, or were equal to the minimum. In conclusion, the reward function concerning task success was defined as follows

$$r_{success} = \begin{cases} 15 & \text{, if } s_{success} > mean \\ 10 & \text{, if } s_{success} = mean \\ 5 & \text{, if } s_{success} < mean \\ 0 & \text{, if } s_{success} = min \end{cases} \tag{7.6}$$

Similarly, we measured the mean duration of each task step for modelling the reward function based on task duration. Here, we defined the reward function for promoting task efficient behaviour, i.e. the duration of a specific task step was below or equal to the mean duration, and not rewarding actions that resulted in a task duration that was above the average. Thus, this culminated in the following reward function

$$r_{duration} = \begin{cases} 10 & \text{, if } s_{duration} \leq mean \\ 0 & \text{, if } s_{duration} > mean \end{cases} \tag{7.7}$$

Note that we weighted the individual reward functions regarding their importance. Foremost, we wanted to achieve trustworthy behaviour. Therefore high trust received the highest possible reward amongst all functions, whereas low trustworthy behaviour even resulted in negative numerical scores. For fostering high usability, we also rewarded successful and efficient actions. However, unsuccessful and not efficient behaviour did not receive a negative reward for balancing the training more to benefit trustworthy dialogue actions.

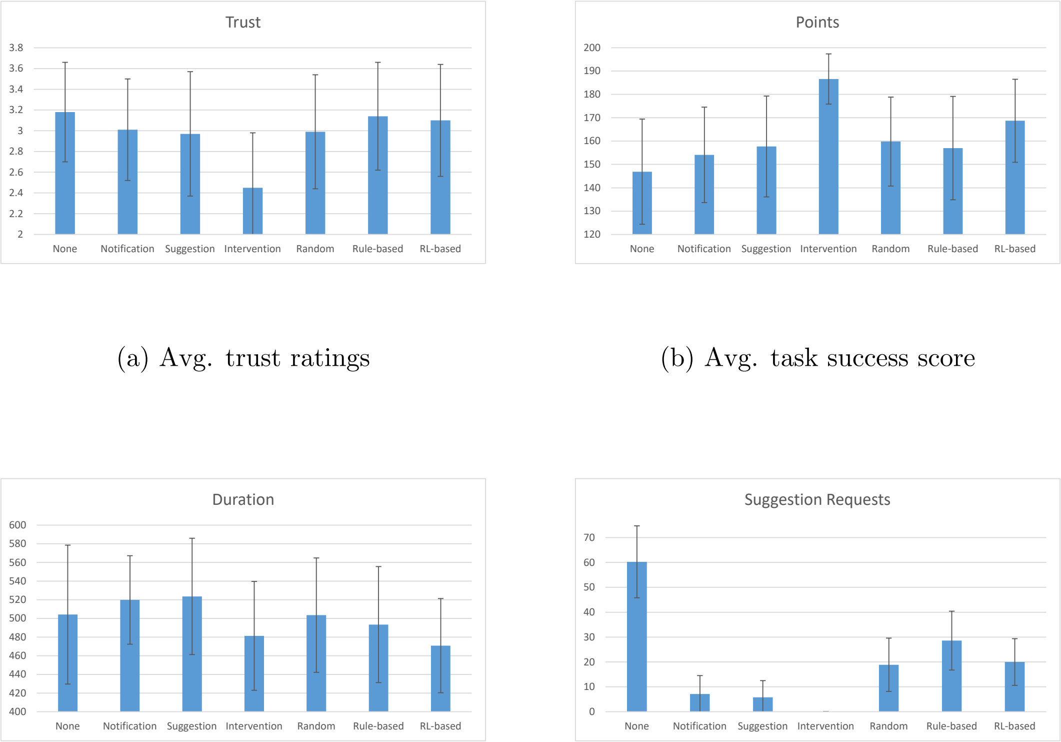
Using the presented reward function, the CA's proactive dialogue policy was then trained in numerous interactions with simulated users. As the state space was quite large with $\approx$ 90.000 possible states, conventional model-free Q-leaning was not feasible. For this reason, we implemented a DQN (Mnih et al., 2015) approach with a stacked

MLP for function approximation. Using a DQN is sample efficient and can handle discrete state and action spaces, as used in this thesis. For implementing the DQN, we utilised the stable-baseline implementation [3]. The architecture of the DQN consisted of two MLP-layers with 256 neurons, an input layer sized in the dimension of the state space, and an output player for producing the Q-values of the dialogue actions. For creating the output a softmax layer is used. Using heuristic search, we applied the following hyper parameters to the DQN. For discounting future rewards, we set $\gamma = 0.99$. Further, we trained the network using the RMSProp-algorithm with the ADAM optimiser (Kingma and Ba, 2014), a learning rate of 0.00005, and a mini-batch size of 64. The replay buffer had a size of 100000 samples. The target network was updated every 500 time steps. In addition, we used ϵ-greedy for training the behaviour policy. Here, ϵ annealed linearly from 1 to 0.1 over 15 % of the training sample and was fixed at 0.1 thereafter. The DQN was trained on a total of 300000 training samples (task steps) or 25000 dialogue games with different simulated users. For speeding up the training process, we normalised the state space using min-max scaling for transforming the state values to range between 0 and 1. The trained *RL-based* strategy was then evaluated against the rule-based, random, and static proactive dialogue strategies. Here, we primarily studied our main two research questions on how these trust-adaptive proactive dialogue strategies influence the system's trustworthiness and usability. Further, we compared the performances of the different approaches to trust-adaptive proactive dialogue. The evaluation and the results are presented in the following.

7.4.3. Experiments and Results

For studying the effects of integrating trust and context into the dialogue state (*rule-based* strategy) as well as the reward function (*RL-based* approach) to implement trust-adaptive proactive dialogue behaviour, we conducted an empirical evaluation with simulated users. For comparison, we tested the trust-adaptive dialogue strategies against the four static baseline strategies, i.e only one proactive dialogue act type (*None*, *Notification*, *Suggestion*, *Intervention*) was used by the CA throughout the dialogue games. Further, we tested against a random baseline strategy that selected the proactive dialogue act type randomly for each task step. For evaluation, we simulated 500 dialogue games per strategy. For each dialogue game, a different user type was simulated. However, the set of simulated users was kept constant for each strategy to ensure comparability of the results. The number of dialogue games was selected to produce normally distributed data sets, that allow the usage of parametric statistical significance tests. As evaluation metrics, we used the average overall trust ratings, overall task success score, and overall task duration. Further, we observed the percentage of suggestion requests per strategy. The results of the evaluation are depicted in Fig. 7.15. Fig. 7.15a shows the average trust ratings. Fig. 7.15b shows the average task success score. Fig. 7.15c shows the average task duration. Fig. 7.15d shows the average of suggestion requests by the simulated users.

[3] https://stable-baselines.readthedocs.io/en/master/modules/dqn.html



(a) Avg. trust ratings

(b) Avg. task success score

(c) Avg. task duration

(d) Avg. suggestion requests

Figure 7.15.: The averages and standard deviations of the evaluation metrics with respect to the proactive dialogue strategies.

Proactive Strategy	**None**	**Notification**	**Suggestion**	**Intervention**	**Random**	**Rule-based**	**RL-based**
None	1.000	.000	.000	.000	.000	1.000	.315
Notification	.000	1.000	1.000	.000	1.000	.002	.174
Suggestion	.000	1.000	1.000	.000	1.000	.000	.002
Intervention	.000	.000	.000	1.000	.000	.000	.000
Random	.000	1.000	1.000	.000	1.000	.000	.025
Rule-based	1.000	.002	.000	.000	.000	1.000	1.000
RL-based	.315	.174	.002	.000	.025	1.000	1.000

Table 7.9.: Comparison of the significance of differences between each proactive dialogue strategy regarding trust.

Significance tests for the differences between the strategies were conducted using t-tests with Bonferroni correction regarding multiple testing. For readability, we indicated the p-values for significant differences in separate tables, except for average suggestion requests for which only two non-significant differences were found. The p-values for differences regarding trust are presented in Table 7.9. The p-values for differences regarding task duration are presented in Table 7.10. Finally, the p-values for differences regarding task success are presented in Table 7.11. In the following, the results of the evaluation between the strategies regarding trust are described.

Effects of Proactive Dialogue Strategies on Trust

Here, the *None* strategy produced the highest trust ratings, followed by *rule-based* and *RL-based* strategies. However, differences between *None* strategy and these two strategies were not significant. Trust ratings declined with an increased level of system proactivity. However, differences between the medium-levels of proactivity were not significant. Trust-adaptive strategies were rated significantly higher for trust than medium-levels of proactivity. An exception were the trust ratings between *RL-based* and *Notification* strategy. There was a drastically decrease of trust for the *Intervention* strategy. Here, there were measured significant differences to all other strategies. The trust ratings for the *Random* strategy were between the ratings for *Notification* and *Suggestion* strategy. Both trust-adaptive strategies were rated significant higher for trust than the *Random* strategy. Next, the results regarding average task duration are described.

Effects of Proactive Dialogue Strategies on Task Duration

The *RL-based* strategy followed by the *Intervention* strategy led to the fastest game completion. The difference between those strategies was not significant. However, there were measured significant differences of both strategies to all other strategies. A further observation was that the task duration was increased step-wise by applying the *rule-based* and *None* strategy. The *Random* strategy resulted in a task duration almost equal to the *None* strategy. Differences between those strategies were non-significant.

Proactive Strategy	**None**	**Notification**	**Suggestion**	**Intervention**	**Random**	**Rule-based**	**RL-based**
None	1.000	.001	.000	.000	1.000	.098	.000
Notification	.001	1.000	1.000	.000	.000	.000	.000
Suggestion	.000	1.000	1.000	.000	.000	.000	.000
Intervention	.000	.000	.000	1.000	.000	.033	.113
Random	1.000	.000	.000	.000	1.000	.156	.000
Rule-based	.098	.000	.000	.033	.156	1.000	.000
RL-based	.000	.000	.000	.113	.000	.000	1.000

Table 7.10.: Comparison of the significance of differences between each proactive dialogue strategy regarding task duration.

Proactive Strategy	**None**	**Notification**	**Suggestion**	**Intervention**	**Random**	**Rule-based**	**RL-based**
None	1.000	.000	.000	.000	.000	.000	.000
Notification	.000	1.000	.076	.000	.000	.401	.000
Suggestion	.000	.076	1.000	.000	1.000	1.000	.000
Intervention	.000	.000	.000	1.000	.000	.000	.000
Random	.000	.000	1.000	.000	1.000	.563	.000
Rule-based	.000	.401	1.000	.000	.563	1.000	.000
RL-based	.000	.000	.000	.000	.000	.000	1.000

Table 7.11.: Comparison of the significance of differences between each proactive dialogue strategy regarding task success.

Finally, the strategies with the longest task duration were *Notification*- followed by the *Suggestion* strategy. Here, the differences were not significant.

Effects of Proactive Dialogue Strategies on Task Success

The *None* strategy produced the lowest task success score measured in game points. Here, the measurements were significant lower than those of all other strategies. The strategies with the next higher task success were the *Notification* and *Suggestion* strategy. The *rule-based* strategy produced similar task success scores than the *Suggestion* strategy. Further, the *Random-* strategy led to a slightly higher task success than these two strategies. The differences between the strategies of a medium-level of proactivity and the *rule-based* strategy were not significant. Also the *Random* strategy was only rated significantly higher than the *Notification* strategy regarding task success. The *RL-based* strategy produced significantly higher task success than all previously mentioned strategies. Only applying the *Intervention* strategy resulted in a significantly higher task success.

In the following, some observations from the percentages of suggestion requests for each strategy are described. Naturally, the *None* strategy resulted in the highest amount of suggestion requests, followed by the *rule-based* strategy. The *rule-based* strategy was

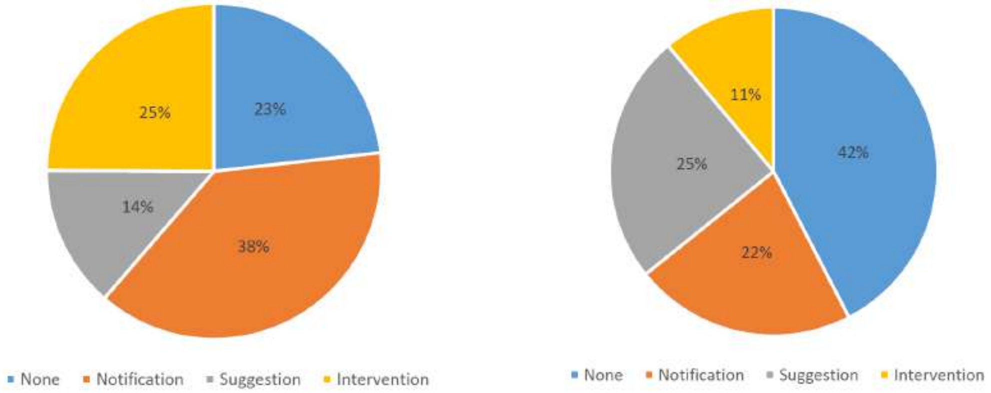


Figure 7.16.: Left: Distribution of proactive dialogue act types using an RL-based approach. Right: Distribution of proactive dialogue act types with regard to rule-based trust adaptation.

followed by the *Random* and *RL*-based approach. They almost had the same amount of suggestion requests. Thus, the difference was not significant. *Notification* and *Suggestion* strategy led to significantly lower suggestions requests. Both strategies had almost the same request frequency (no significant difference). The *Intervention* strategy resulted in no suggestion requests.

Comparison of the Trust-Adaptive Proactive Dialogue Strategies

In the following, we continue to solely compare the trust-adaptive dialogue strategies. In Fig. 7.16, the distributions of proactive dialogue act types with regard to these strategies are visualised. Using the *rule-based* strategy, the most frequent proactive dialogue act type was the *None* action with 42 %. *Notification* and *Suggestion* action were almost evenly distributed. The least frequent occurring proactive dialogue act type was the *Intervention* action with 11 %. Considering the *RL-based* strategy, the *Notification* action was the most frequent used act type with 38 %. The *Intervention* and *None* action were the second most used act types with almost the same distribution. The least used act type was the *Suggestion* action with 14 %. Table 7.12 describes which proactive dialogue act type was used how often at different task step complexities per trust-adaptive strategy. For both strategies, the *None* action type was most frequently used in task steps with a complexity of 1, i.e. three options. Using the *rule-based* strategy, the *Notification* act type was most frequently used for complexity 3 tasks (five different options). Using the *RL-based* strategy, the *Notification* act type was most frequently used for complexity 2 tasks (four options). For both strategies, the *Suggestion* act type was most frequently used for complexity 1 tasks. Using the *rule-based* strategy, the *Intervention* act type was most frequently used for complexity 3 tasks. Contrary, the *Intervention* act type was most frequently used for complexity 2 tasks using the *RL-based* strategy. For complexity 1 tasks, the most frequent used proactive dialogue act type, was the *None* action for both strategies.

Proactive DialAct	**Rule-based**			**RL-based**		
	Complexity 1	Complexity 2	Complexity 3	Complexity 1	Complexity 2	Complexity 3
None	1195	818	531	1097	117	203
Notification	0	509	802	110	1152	1064
Suggestion	804	673	1	793	34	11
Intervention	1	0	666	0	797	723

Table 7.12.: Description of the counts of proactive dialogue act types dependent on the task step complexity and the trust-adaptive proactive dialogue strategy.

Additionally, we compared the distributions of proactive dialogue act types dependent on the task complexity type. The distributions between all dialogue act types were similar for complexity 1 tasks. For complexity 2 tasks, the most frequent used proactive dialogue act type was the *None* action for the *rule-based* strategy and the *Notification* action for the *RL-based* strategy. For complexity 3 tasks, the most frequently used proactive dialogue act type was the *Notification* action for both strategies.

Further, it could be observed that for both trust-adaptive strategies, *None* and *Suggestion* actions were rather used at task steps with lower complexity. Meanwhile, *Notification* and *Intervention* actions were applied at task steps with a higher degree of complexity. In addition, the *RL-based* strategy showed to make a clearer use of the *None* and *Suggestion* actions. While the *rule-based* strategy applied both actions more evenly at tasks with complexities of 2 and 3, the *RL-based* strategy, applied those actions primarily for complexity 1 task steps. Moreover, the *RL-based* strategy applied the *Intervention* strategy almost evenly for task steps with complexities 2 and 3. Contrary, the *rule-based* strategy almost entirely used the *Intervention* action for complexity 3 tasks.

Understanding the Behaviour of the RL-based Strategy

In the following, we focus our investigation solely on the *RL-based* strategy, in order to understand the reasons for its learned behaviour. Therefore, the system's selection of proactive dialogue acts dependent on its last known trust value was observed. The results are presented in Table 7.13. We further analysed the dialogue act selection dependent on the game score of the last task step. These results are shown in Table 7.14.

Regarding trust, it could be observed that only two times a trust value of $Trust = 1$ was estimated. For the task step after this estimation, the system always chose a *Notification* action. After the system observed an estimated trust value of $Trust = 2$, also a *Notification* action was selected the most. The action was chosen 64 % of the time, while the selection of the other actions was almost evenly distributed. As a response to an estimated trust score of $Trust = 3$, again the *Notification* action was selected the most. Here, the action was however used only 41 % of the time, followed by the *None* action with 27 %. After estimating a trust value $Trust = 4$, the *Intervention* action was the most frequently selected action with an occurrence 37 %.

Proactive DialAct	**Trust = 1**	**Trust = 2**	**Trust = 3**	**Trust = 4**	**Trust = 5**
None	0	169	669	517	62
Notification	2	897	1000	408	19
Suggestion	0	158	288	359	32
Intervention	0	184	481	742	13

Table 7.13.: Description of the counts of proactive dialogue act types for the RL-based strategy dependent on the last observed trust value.

Proactive DialAct	**Points = 0**	**Points = 10**	**Points = 20**	**Points = 30**	**Points = 40**
None	8	377	409	401	198
Notification	355	1808	115	8	0
Suggestion	130	139	90	42	0
Intervention	81	865	474	0	0

Table 7.14.: Description of the counts of proactive dialogue act types for the RL-based strategy dependent on the last observed game score in points.

For all three previously described trust values, the *Suggestion* action was selected the least frequently. A trust value of $Trust = 5$ was not frequently observed by the system like a trust value $Trust = 1$. Here, predominantly the *None* action was selected at 49 %. In the following, the relation between the selected proactive dialogue act dependent on the user's last game score is investigated. After a user received zero points, the system selected a *Notification* action the most. Here, a medium level of proactivity was predominant with an occurrence of 85 %. A *None* action was selected only at 1 % of the time. After a user received 10 points, the system also selected a *Notification* action the most at 57 %. Also the *Intervention* action was selected frequently at 27 %. After the user received a game score of 20 points, the system mostly selected a *Intervention*- (44%) and *None* action (38 %). A *None* action was the most selected proactive dialogue act after the user received a game score of 30 points (89%) and 40 points (100 %).

7.4.4. Discussion

First, we discuss the results of the static proactive dialogue strategies for the user simulation study and put them into context with our previous experiments. As indicated by Fig. 7.15a and Table 7.9, the user's perceived trust (measured as a combination of the cognition-based trust ratings) steadily decreased with an increasing level of proactive dialogue. Except for the medium-level proactive dialogue strategies, the differences between the static strategies were significant. These results were congruent to our experiments with human subjects and suggested that the user simulator results are close to reality. Considering the task success scores of each static strategy, the inverted effect was observable (see Fig. 7.15b and Table 7.11). The higher the level of proactivity, the higher the task success. Again, only the differences between the medium-level strategies were not significant. These results, however, were to be expected, as the virtual CA was designed to be an expert system having complete domain knowledge. Therefore, the *Intervention* strategy led to optimal task success, as the CA consistently selected the best option. More interesting seemed to be the fact that trust was not positively correlated to task success, even though complete system autonomy led to the highest task success rates. This was also observable for the medium-level proactive dialogue strategies that showed significantly higher trust than the *Intervention* strategy, but simulated users did not always comply with the assistant's notifications or suggestions, as indicated by the lower task success scores. This is a strong indicator that high-performing systems are not necessarily more trustworthy, but also the way a system communicates its decision processes seems to be important.

Further, the concept of trust and usability may not be handled independently. For this, a compromise between high-performing but also trustworthy behaviour needs to be found. Looking at the average task success results provided in Fig. 7.15c and Table 7.10, the *Intervention* strategy was naturally the most time-efficient as it did not negotiate the decision with users. For this reason, the medium-level strategies had the highest average task duration as these strategies provoked the longest dialogues. The *None* strategy led to the shortest task duration behind the *Intervention* strategy, as the CA stayed reactive and users were not necessarily required to interact with the system. However, for roughly 80 % of the task steps users requested suggestions from the reactive version of the CA (see Fig. 7.15d). This indicated that help by the CA was welcomed by the simulated users. Overall, the results seem to be in line with our previous experiments.

In the following, we focus on the effects of the trust-adaptive adaptive strategies concerning the main research questions of this thesis.

Influence of Trust-Adaptive Proactive Dialogue on Trust and Usability

For this investigation, we looked into the performance and trust metrics of both *rule-* and *RL*-based strategy in comparison with the other strategies. Regarding trust, both strategies led to significantly higher trust than random, high- and medium-level strategies. They were only rated slightly lower for trust than the *None* strategy albeit not significantly. Note that the reactive strategy resulted in the optimal trust values (*RL-based* strategy only learning to increase trust resulted in *None* strategy). For this reason, both trust-adaptive strategies achieved near-optimal values regarding trust. Regarding task success, the rule-based strategy performed at the same level as the random, and medium-level strategies. The *RL-based* strategy outperformed all other strategies except the *Intervention* strategy which achieved optimal results regarding task success. For task duration the results are similar for the *RL-based* strategy. Here, also the *rule-based* strategy performed well resulting in significantly shorter interactions than medium-level proactivity and notably shorter interactions than the *None* strategy. Considering the task efficiency as a comparison between task success and task duration, it was observable that the trust-adaptive strategies than the other strategies except for the *Intervention* strategy. However, it must be noted that regarding task efficiency, the *rule-based* strategy performed at a similar rate to the random strategy. Taking the trust ratings into account, it became clear that the trust-adaptive strategies achieved the best compromise for providing socially and task-effective proactive dialogue employing high trustworthiness and usability. Therefore, we deemed trust to be an adequate metric for designing proactive dialogue strategies for improving the cooperation with CAs.

Performance of the RL-Based User-Centred Proactive Dialogue Strategy

The *RL-based* strategy outperformed the hand-crafted strategy regarding task efficiency while simultaneously achieving comparable trust ratings. Therefore, it may be interesting to look more closely into the decisions the *RL-based* strategy made in comparison to the *rule-based* strategy. This may help to find reasons, why this strategy performed that well for this task and provide further insights into the user perception of proactive dialogue behaviour. The first interesting finding was that increased usage of *Notification* and *Intervention* action possibly led to a higher task efficiency of the *RL-based* strategy without losing the user's trust. Similar to our findings in Section 6.1 and 6.2, this indicated that both reactive and highly proactive system behaviour needs to be applied carefully, but can lead to enhanced trustworthiness and usability when applied properly. Further, it was shown again that a medium-level of proactivity (primarily a *Notification* action) seemed beneficial for providing trustworthy but also successful assistance.

Next, it may be useful to consider the frequency at which each proactive dialogue act type was selected depending on the task complexity level. Here, the most notable difference between the strategies was that the *RL-based* strategies more distinctively made use of *None* and *Suggestion* actions. The *RL-based* strategy almost predominantly used *None* and *Suggestion* actions for tasks with a complexity level of 1. *Notification* and *Intervention* actions were almost evenly distributed for tasks with a complexity level of 2 or 3, but rarely (or not at all used) for tasks with a complexity level of 1. Combined, this possibly resulted in high task efficiency. Observing both strategies, the usage of *None* and *Suggestion* actions at task steps with lower complexity while using *Notification* and *Intervention* actions at task steps with a higher degree of complexity seemed to positively contribute to the high trust ratings of these strategies.

Finally, it may be useful to consider the *RL-based* strategy proactive dialogue act type selection dependent on the system's lastly obtained trust value or game score. Under consideration of the user's last perceived trust in the system, it could be observed that the *Notification* action was primarily used after the system had observed low or medium trust values. The higher the lastly observed trust value, the more one of the other proactive dialogue act types was selected. However, after observing a positive trust value of 4, the system predominantly used an *Intervention* action. Interestingly, in the case of the highest trust values, the system switched into a reactive mode. Thus, it can be stated that for developing socially and task-effective proactive dialogue, notifications should be considered if the system estimates its trustworthiness to be low to medium, while suggestions and highly proactive behaviour may be applied at higher trust levels.

However, when trust is estimated as the highest, it seems to be more useful to stay reactive and wait for the user's action to not harm the relationship between the user and CA.

Similar behaviour was also observable considering the *RL-based* strategy's proactive action selection dependent on the last game score. Here, the virtual CA primarily used a medium level of proactivity after the user was unsuccessful and received zero points. This is a clear sign of the system taking control into its own "hands" to provide helpful behaviour, but still letting the user control the final decision in order not to lose trust. For low task success (points = 10), the system primarily selected a *Notification* action but also often selected an *Intervention* action. As the *Intervention* strategy was primarily used for trust values $trust = 4$, whereas the *Notification* action was used for lower trust values, it may be concluded that after observing low success by the user, the system only triggered highly proactive behaviour if it also observed a high trust value. After the system observed high task success by the user, it gradually almost exclusively selected a *None* action. Here, it could be reasoned that the virtual CA opted to take more control into the hands of the user after detecting successful user behaviour. This may be due to the success of the task not being endangered and reciprocal trust was deemed to be established.

Limitations

The limitations of these experiments were similar to those of the data collection. First, using more features for modelling the states and the proactive actions could lead to more general results. However, as we measured similar effects as compared to our experiments with human subjects, the results of the study should be sufficiently validated. Nonetheless, experiments of real users interacting with a *RL-based* virtual CA are necessary for final validation. Further, we tested the strategies in a simplified task domain for decision-making. Decision-making in real-life is far more complex. Therefore, the here presented approaches need to be transferred into more realistic use case scenarios to make more profound claims. However, due to the quality of the results and the overlaps of this study with previous experiments, we deemed the findings to be reproducible in realistic settings. Further, the study results may provide guidelines for designing proactive dialogue behaviour in realistic task scenarios.

7.4.5. Conclusion

We developed two trust-adaptive proactive dialogue strategies to improve the cooperation with CAs by achieving a socially and task-effective dialogue. A *rule-based* method utilised the current trust estimate and task complexity for proactive dialogue action selection. Further, an *RL-based* method was trained using both trust estimate and usability measures for providing adequate proactive dialogue behaviour. For evaluating the approaches, we compared them with static and random proactive dialogue strategies.

Including trust in the dialogue model for enabling trust-adaptive dialogue proved to be successful in creating trustworthy CAs with high usability. Both methods achieved the best compromise of contributing to task completion effectively but also acting in a trustworthy manner. Particularly, the *RL-based* trust-adaptive proactive dialogue strategy was evaluated to be superior to all other strategies. Thus, these results showed that the main research goal of this thesis could be successfully achieved.

Further, examining the behaviour of the *RL-based* strategy allowed us to provide insights on the utility of a proactive dialogue action dependent on the user and the context. In several cases, our previous findings were emphasised by observing the behaviour of the *RL-based* strategy. For example, the *Notification* actions were primarily used by the *RL-based* strategy for achieving the desired behaviour. However, also new findings could be discovered. Generally, the proactive agent seemed to learn a quite human-like strategy. The system adjusted its behavior depending on current the level of the HCT relationship. Recognising a rather low trust level, the system primarily expressed a low- to medium-level of proactivity. However, the more it recognised that the user trusts its abilities, the more it expressed higher levels of proactivity and decided on its own. An exception was the highest level of the HCT-relationship. Here, the system mostly stayed reactive. Therefore, it may be concluded that the system did not want to risk the damage of the HCT relationship by becoming proactive when the user had the highest trust in the system. For a medium leveled trust relationship, the benefits of becoming proactive seemed to outweigh the costs of damaging the relationship. Further, the system seemed to recognise the need for proactive behaviour if users were less successful at task execution. The more successful the user got, the less the system interferes in the user's decision making.. This behaviour strongly resembled the behaviour of human assistants that act more actively if they recognise that the person they help trusts them and requires help due to low task success. In summary, the inclusion of trust in the reward function seemed to have equipped the proactive CA not only with the ability to provide task effective assistance but also more human-likeness which is reflected in the social effectiveness.

7.5. Summary

This chapter presented the implementation of a user-centred proactive dialogue model for achieving our main research goal of improving the cooperation with CAs that act both socially and task effectively by the means of trustworthiness and usability.

First, we presented a study showing that the concept of trust represents a valid adaptation criterion for proactive dialogue. Here, we found that especially cognition-based trust features may be used for adapting the proactive dialogue. However, this may only increase a CA's trustworthiness during cooperation and not its usability. Therefore, a combination of trustworthiness and usability/performance-related measures was proposed to be included for improving the cooperation using user-adaptive proactive dialogue strategies.

In a second step, we contributed a novel trust recognition module for measuring a user's trust in the system's actions online during a dialogue. For developing the trust module, we fused knowledge gained from our experimental studies and related work regarding HCT. The development of the trust module comprised several steps. First, we created a data corpus containing proactive system behaviour and trust annotations.

Subsequently, we developed a novel dialogue-based trust model. Finally, we used the trust model for implementing the trust recognition module using machine-learning predictors. Evaluation of different predictors revealed that an SVM approach was best suited for estimating trust during an ongoing dialogue. This allowed to include trust into the dialogue model for generating trust-adaptive proactive dialogue strategies. For testing and training trust-adaptive proactive dialogue strategies, we developed a novel user simulator that was able to simulate different user types and user-specific task behaviour. This information could then be used by a proactive CA to measure its trustworthiness concerning the simulated user. Evaluation of the user simulator showed realistic behaviour.

Finally, we developed a *rule-based* and an *RL-based* trust-adaptive proactive dialogue strategy that was trained in interaction with the user simulator. We subsequently tested the strategies and their influence on cooperation by applying user simulator evaluation. The results proved the success of our approach in achieving the main goal of this thesis of developing a user-centred proactive dialogue model for rendering CAs trustworthy and improving the cooperation from a social as well as from a task-oriented perspective.

8. Conclusion and Future Directions

In this thesis, we presented work on improving human-machine cooperation by taking a user-centred approach in proactive dialogue modelling for developing trusted and task-effective CAs. Despite providing intelligent functionalities, current assistance systems lack the ability to provide trustworthy highly usable proactive conversation strategies due to a mismatch between user expectations and actual system behaviour. For overcoming this "gulf" and thus improve cooperation, we divided the problem into several sub-problems:

1. Proactive dialogue modelling for human-machine cooperation
2. Design of user-centred proactive dialogue strategies and their effects on cooperation
3. Implementation and evaluation of user-centred proactive dialogue strategies for trustworthy CAs with high usability

For providing a proactive dialogue model in cooperation contexts, we first conducted two experimental pilot studies embedding state-of-the-art proactive behaviour in the form of recommendations and notifications into the dialogue domain. These studies provided a foundation for the user perception of proactive dialogue on a system's usability and trustworthiness. In combination with related work, the results of these studies allowed to distill user and system requirements for enabling proactive dialogue in CAs. Based on these requirements, we formulated the human-machine cooperation process as a dialogue problem and defined four novel proactive dialogue act types representing different levels of system autonomy. Additionally, we presented a novel cognitive system architecture, combining AI and HCI components, for implementing proactive DM in assistance systems.

We advanced the state-of-the-art understanding of the effect of the proactive dialogue model on cooperation by developing four novel approaches to user-centered proactive dialogue design and implementing them into prototypical CAs. In two laboratory experiments, we were able to reveal significant relations between proactive dialogue act types and the HCT relationship dependent on specific user characteristics, task properties, and cognitive-affective user states. Further, we found that proactive dialogue mainly had an impact on the user's cognitive-based trust (perceived competence, reliability, understandability). In addition, a high level of proactive dialogue showed tendencies to increase a system's usability in comparison to reactive behaviour, however, with reversed trust effects. For determining the need for proactive dialogue behaviour, we found evidence that the usage of singular user states seems to be insufficient for this task. In two user studies in realistic task scenarios using high-fidelity proactive system prototypes, we showed that our findings from the laboratory experiments are transferable and gained several novel insights.

The user studies revealed positive results of a medium-level of proactive dialogue for inexperienced users with low technical affinity. In addition, including more context-related features as a trigger mechanism for proactive decision-making proved to be useful for improving the cooperation and may be enhanced by including various user states. The gained knowledge from these experimental studies considering proactive dialogue design was then used to implement a user-centered proactive DS

The implementation of a user-centered DS provided several individual contributions to advancing the state-of-the-art. First, we identified trust to be an adequate metric for the assessment of whether proactive dialogue meets social expectations and may be used for the adaptation of proactive dialogue. For including trust in the proactive dialogue model, we contributed a novel user model that allows the prediction of the user's trust level during an ongoing dialogue. The evaluation of the model provided promising results for utilising a trust metric as dialogue adaptation criteria. Finally, the implementation of a trust-adaptive proactive DM module was achieved to enable trusted and task-effective proactive behaviour. Particularly, a novel approach including trust and task metrics in a reward function for RL-based DM proved to be beneficial for improving human-machine cooperation.

Overall, we consider technical systems with the ability to recognise and measure social aspects, such as trust, during interaction with humans to be fundamental for providing adequate proactive assistance. Equipping CA with these capabilities is an important step towards the development of more human-like and true cooperation partners. While the major findings and conclusions of this thesis were outlined so far, a more detailed description of the contributions of our work is presented in the following.

8.1. Thesis Contributions

The presented work on user-centred proactive dialogue modelling for trustworthy CAs provided several contributions that advance the state-of-the art of proactive DM. In the following, we summarise our contributions grouped into *theoretical*, *practical*, and *experimental* contributions.

8.1.1. Theoretical

We contributed a theoretical proactive dialogue model building upon the state-of-the-art of proactive interaction design. This included modelling the mixed-initiative cooperation process between humans and machines as a dialogue problem, where we defined proactive dialogue as the initiation of sub-dialogues at turn-level influencing future user actions (Kraus et al., 2021b). In addition, we introduced four different levels of proactive dialogue act types based on autonomy research (Kraus et al., 2020c). Further, a novel cognitive architecture for proactive DM was developed (Kraus et al., 2019, 2020b). This architecture comprised modules for planning, reasoning, and dialogue for allowing proactive behaviour during task assistance. The dialogue module utilised a user model for deciding whether to become proactive and to which extent.

For evaluating proactive dialogue, we developed two novel evaluation frameworks. The first introduced the concept of a serious dialogue game for enabling data collection and for testing different proactive dialogue strategies in cooperation contexts (Kraus et al., 2020c, 2022b, 2021c). The second framework utilised an interactive video method for allowing users to conduct a dialogue with a CA while watching a video. At a certain moment during the video, users could take actions that directly influenced the CA's behavior and the further interaction (Kraus et al., 2022e).

Moreover, we developed four novel user-centred proactive dialogue strategies that utilised different kinds of user state and context recognition. One strategy utilised the user's state of insecurity for proactive dialogue (Kraus et al., 2020c, 2021b). In addition, we developed a proactive dialogue design based on the user's cognitive-affective state (Kraus et al., 2022a), and two context-related strategies based on the detection of contextual events (Kraus et al., 2022c) and user activity (Kraus et al., 2020b).

A trust model was developed for integrating trust as metric for user-adaptive proactive DM (Kraus et al., 2021c). In this regard, we also introduced a human-to-machine data collection setup (Kraus et al., 2022b) and a trust recognition approach based on methods for ML (Kraus et al., 2021c).

Finally, we developed two trust-adaptive proactive dialogue strategies: a rule-based and a RL-based method (Kraus et al., 2022d). For the RL-based method, we modelled the dialogue as an MDP and developed a novel socially-aware user simulator for training purposes. This allowed us to optimise a proactive strategy taking into account task-related metrics and information about the user's trust. In this regard, we developed a reward modelling strategy for creating effective user-adaptive dialogue strategies from both a social (trustworthy) and a task-oriented (usability) perspective.

8.1.2. Practical

For allowing experimentation, the previously described theoretical contributions were implemented into a range of prototypes. For our pilot studies, we implemented two DM prototypes into existing dialogue frameworks. A recommendation functionality was implemented into the Amazon Alexa framework (Kraus et al., 2020a). Utilising the Rasa dialogue framework, we implemented proactive behaviour in the form of push notifications and topic switching behaviour into a mental health assistant (Kraus et al., 2021a).

For the design of user-centered proactive dialogue strategies and for measuring their effects on cooperation, we implemented two low- as well as two high-fidelity DM prototypes. The low-fidelity prototypes implemented rule-based DM for controlling a Nao robots proactive behaviour. One version utilised the user's insecurity state measured as a specific duration of user inactivity for managing the dialogue (Kraus et al., 2020c, 2021b), while the other utilised the user's cognitive-affective state which was measured using an off-the-shelf recognition module for controlling the flow of the dialogue (Kraus et al., 2022a).

The high-fidelity prototypes implemented context-related strategies utilising an agent-based approach. Here, several dialogue agents handled the interplay between the dialogue manager and AI planning as well as reasoning modules. We implemented the agent-based approach in a virtual CA (Kraus et al., 2019, 2020b) and in a robotic CA (Kraus et al., 2022c)

For including the concept of trust in proactive DM, we implemented a web-based data collection setup and created a trust-annotated proactive dialogue corpus (Kraus et al., 2022b). Further, we implemented a trust recognition module using ML frameworks (Kraus et al., 2021c). Additionally, ML frameworks were applied for implementing an RL-based proactive dialogue manager Kraus et al. (2022d). The manager was trained with a purpose-built user simulator.

8.1.3. Experimental

Based on the presented theories and practical implementations, we conducted user simulator and real-world experiments, both in the laboratory (Kraus et al., 2020c, 2022a,e, 2021b, 2022d) and in realistic environments (Kraus et al., 2020a,b, 2021a, 2022c).

For building an intuition of the user perception of proactive dialogue on the human-machine cooperation and to distill user requirements, we conducted two pilot experiments in the wild. Here, users interacted with the respective prototype either using their smartphone messenger service (Kraus et al., 2021a) or via the Amazon Alexa device (Kraus et al., 2020a).

For contributing to the understanding of the effect of user-centered proactive dialogue design on the cooperation, we conducted two laboratory experiments (Kraus et al., 2020c, 2021b, 2022a, 2020b, 2022c). Here, we considered the impact of the individual proactive dialogue act types on a system's trustworthiness and usability dependent on the task context, user characteristics, specific user states as well as activities, and external events that were supposed to require proactive assistance. We found significant relations between proactive level and specific user characteristics, namely personality traits, technical affinity, domain expertise (Kraus et al., 2021b). Furthermore, how difficult a task is perceived by the user seemed to have an effect on the trustworthiness of proactive dialogue actions (Kraus et al., 2020c). Additionally, the effect of different levels of proactivity on the user experience and trust depending on the user's cognitive-affective states was studied during a learning task (Kraus et al., 2022a). Here, we found the proactive dialogue to have an impact on the learning process and identified negative cognitive-affective user states (confusion, frustration) to be insufficient for detecting the user's need for assistance. Considering the usability of proactive dialogue act types during the first two experiments, we found that a high level of proactivity was generally perceived to be more task effective, wherein reactive system behaviour received the lowest scores for task effectiveness. Considering the medium-level of proactivity, we could make no clear statement. However, notification showed tendencies to increase usability dependent on the task context.

Findings of the impact of proactive dialogue design on the cooperation could be confirmed in realistic task scenarios using sophisticated virtual (Kraus et al., 2020b) and robotic CAs (Kraus et al., 2022c). Here, the results showed that proactive dialogue was able to build an adequate level of user trust, particularly in interaction with technical and domain novices. Considering the outcomes of the experiments in more realistic task scenarios, we found the usage of more context-related than user-specific information as a trigger mechanism for proactive behaviour highly useful. Consequently, we concluded a combination of context- and user-specific information to be suitable for deciding whether to become proactive and to which extent.

Further, we conducted an experiment on the application of trust as a measure for proactive dialogue adaptation using an interactive video method (Kraus et al., 2022e). Here, the main finding was that trust is indeed a reasonable metric for measuring the match between social expectations and proactive dialogue behaviour and thus be used as an adaptation criterion. Regarding the influence on the cooperation, we concluded that deciding on an adequate level of proactive dialogue based on such a trust measurement may however solely improve the social aspects of the cooperation but not necessarily its usability. Therefore, usability features should also be integrated for dialogue adaptation.

Subsequently, we conducted experiments for testing the utility of a trust-based model for accurately predicting the user's perceived trust in the system. The results showed that the model is well-suitable for predicting trust in proactive dialogue. Each applied classifier proved to be useful for the classification task, showing reasonable recall and accuracy (Kraus et al., 2021c).

Our final user simulation experiments showed promising results indicating that the inclusion of trust for user-centered proactive DM can be used for improving the cooperation with CAs by rendering a DS trustworthy and task effectively. Particularly, an RL-based trust-adaptive proactive dialogue approach achieved high usability and adequate user trust in the system.

8.2. Future Directions

This work presented positive results for improving human-machine cooperation utilising user-centred proactive dialogue models. However, there is a long road with many obstacles ahead until proactive dialogue can be used to its fullest potential in CAs. In the following, we, therefore, address some open questions and limitations of our work.

Advanced Turn-Taking: The decision when to initiate proactive actions was handled on a (sub-)task-level basis using well-defined points of time. In order to allow more natural and flexible turn-taking for proactive conversation a less rigid approach needs to be taken. For example, user cues, e.g. utterances, pauses, etc. could be processed incrementally to increase the quality of predicting the appropriate timing of proactive actions. Here, computational models for turn-taking (Schlangen, 2006) and incremental dialogue processing (Schlangen and Skantze, 2011) could be used as a foundation. Furthermore, this could help to observe proactive system behaviour in a more conversational than an assistance context, e.g. investigate proactive actions for collaborative utterance construction. Furthermore, the timing of proactive dialogue could be based on multimodal cues e.g. gaze of the user (Huang and Mutlu, 2016), or verbal cues (Seon et al., 2012).

System Errors: Another important aspect of HMI is errors in cooperation and responding to them. While in this thesis expert systems were used for evaluation standardisation, erroneous system behaviour needs to be considered for transferring proactive dialogue models to real-world applications. Therefore, repair strategies and the impact of wrongful proactive actions on the user need to be studied. The development of repair strategies for CA has emerged to become a current hot topic in research, e.g. see Cuadra et al. (2021) and Candello and Pinhanez (2018). However, for being able to repair, a CA must first recognise that it has made an error that requires correction. This is a quite difficult problem for which the complexity of the DM module needs to be increased. For this, sophisticated reasoning and decision-making mechanisms are required to characterise the type of error and to elaborate adequate actions. A promising approach that may allow computational systems to self-detect erroneous behaviour seems to be an inference from a user's behavioural and social signals in the face of errors, e.g. see Kontogiorgos et al. (2020).

Modelling Trust in Conversation: In this work, a trust model for dyadic conversations was presented. The computational model was based on several factors known to influence a proactive system's trustworthiness, including system-, task-, and user-related features. However, due to the highly multi-faceted nature of trust, it was only possible to include a subset of factors in the user model. For creating a more general concept of conversational trust, additional trust-related factors need to be represented in the user model. Here, affective computing (e.g. see Picard (2000)) and especially the interpretation of a user's social signals (Wagner et al., 2013) can help to develop more accurate models. Besides, the measurement of trust is still an open topic. Current measurements are based on Likert scales, however other forms of representations need to be explored. Additionally, the application of deep learning architectures, e.g. transformer embeddings (e.g. see Chiang et al. (2020)), may be beneficial for improving the trust prediction. This may help to provide a trust-based user model which is applicable in real-life scenarios.

Augmentation with Argumentation: As future assistants are expected to operate in more and more complex tasks, providing explanations for system behaviour might make the user aware of its internal processes. However, it might not help users to accept the system's actions as users could be unsatisfied with the explanations, due to the system using the wrong arguments. Furthermore, users might want to debate about the system's actions before they accept them. Therefore, augmenting a proactive dialogue with argumentative strategies might be an adequate way of convincing users to trust a CA actions and become a more versatile helper. The development of argumentative dialogue systems has received wide recognition in the last decade (Rosenfeld and Kraus, 2016; Rach et al., 2018). Especially with regard to AI-based human-machine teaming in which the user is supposed to follow the judgment/recommendation of an AI system (e.g. see Chesñevar et al. (2009)). Here, argumentation serves as the basis to explain the reasoning of the internal processes of a recommendation system. Such "black box" explanations of internal system reasoning are known under the term explainable AI which could boost a system's trustworthiness(e.g. see Adadi and Berrada (2018)). Furthermore, Weld and Bansal (2019) argue to augment an interactive system's capability to provide explanations by allowing users to ask further questions. The authors identified a number of follow-up question types that a system should answer, e.g. questions about a basis for decision-making, queries about the sensitivity of the system, or questions for further details. As a response, an intelligent system could engage in dialogue with the user presenting its arguments for justifying its decisions. Such explanatory approaches enriched with argumentation can also be beneficial for research on proactive dialogue systems. However, how to combine argumentation and proactivity is still an open quest.

Multi-Party Dialogue: In this thesis, we assumed cooperation during task execution between two participants: a human and a proactive CA. In reality, however, often groups of people or teams collaborate for solving complex tasks or making decisions as a group. Here, the natural language interaction takes place in the form of a multi-party dialogue (Traum and Rickel, 2002; Branigan, 2006). Applying a CA in multi-party dialogues poses several new challenges for proactive dialogue design. For example, the CA needs to detect the needs of individuals in the group and the ensemble itself for contributing to the cooperation. Thus, it is required to be aware of not only individual user states and contexts, but also specific group dynamics for deciding whether to become proactive. Further, the target of the impact of proactive behaviour is no longer only one user, but a subset of users or even the whole group. Therefore, multiple users are affected by a proactive action which may even affect the relationships between the users themselves. Considering such challenges, the interesting question is whether the results of this thesis may be transferred to the domain of multi-party dialogue. For application in this domain, we deem the prediction of social group dynamics, similar to the inclusion of trust in this thesis, as a promising approach for determining the need for proactive behaviour. Further, the proactive dialogue acts types need to be extended in order to receive the attention of the addressee, which is not as easy as in dyadic interaction. In this regard, several barge-in techniques may be relevant to consider (Wagner et al., 2021). In summary, proactivity in multi-party dialogue provides a challenging new research topic, for which the findings in this work may provide a solid foundation.

A. Questionnaires

Nr.	Items	Sub-Scales
1.	I see myself as someone who is reserved.	Extraversion
2.	I see myself as someone who is generally trusting.	Agreeableness
3.	I see myself as someone who tends to be lazy.	Conscientiousness
4.	I see myself as someone who is relaxed, handles stress well.	Neuroticism
5.	I see myself as someone who has few artistic interests.	Openness
6.	I see myself as someone who is outgoing, sociable.	Extraversion
7.	I see myself as someone who tends to find fault with others.	Agreeableness
8.	I see myself as someone who does a thorough job.	Conscientiousness
9.	I see myself as someone who gets nervous easily.	Neuroticism
10.	I see myself as someone who has an active imagination.	Openness

Table A.1.: The BFI-10 items and the respective sub-scales according to Rammstedt et al. (2013).

Nr.	Items	Sub-Scales
1.	I enjoyed doing this activity very much.	Interest
2.	This activity was fun to do.	Interest
3.	I thought this was a boring activity.	Interest
4.	This activity did not hold my attention at all.	Interest
5.	I would describe this activity as very interesting.	Interest
6.	I thought this activity was quite enjoyable.	Interest
7.	While I was doing this activity, I was thinking about how much I enjoyed it.	Interest
8.	I think I am pretty good at this activity.	Perceived Competence
9.	After working at this activity for awhile, I felt pretty competent.	Perceived Competence
10.	I am satisfied with my performance at this task.	Perceived Competence
11.	I was pretty skilled at this activity.	Perceived Competence
12.	This was an activity that I couldn't do very well.	Perceived Competence
13.	I put a lot of effort into this.	Effort
14.	I didn't try very hard to do well at this activity.	Effort
15.	I tried very hard on this activity.	Effort
16.	It was important to me to do well at this task.	Effort
17.	I didn't put much energy into this.	Effort

Table A.2.: The Intrinsic Motivation Inventory (IMI) items and the respective sub-scales according to McAuley et al. (1989).

Nr.	Items	Sub-Scales
1.	For this task, many things needed to be kept in mind simultaneously.	ICL
2.	This task was very complex.	ICL
3.	made an effort, not only to understand several details, but to understand the overall context.	GCL
4.	My point while dealing with the task was to understand everything correct.	GCL
5.	The learning task consisted of elements supporting my comprehension of the task.	GCL
6.	During this task, it was exhausting to find the important information.	ECL
7.	The design of this task was very inconvenient for learning.	ECL
8.	During this task, it was difficult to recognize and link the crucial information..	ECL

Table A.3.: The Cognitive Load Survey items and the respective sub-scales according to Klepsch et al. (2017).

Nr.	Items	Sub-Scales
1.	I enjoy trying out a technical system.	Enjoyment
2.	I know most of the functions of the technical systems I own.	Competence
3.	It is easy for me to learn how to operate of a technical system.	Competence
4.	Technical systems make my everyday life easier.	Positive Attitude
5.	Technical systems make many things more cumbersome.	Negative Attitude
6.	Technical systems cause stress.	Negative Attitude

Table A.4.: The technological affinity scale items and the respective sub-scales adopted from Karrer et al. (2009).

Nr.	Items	Sub-Scales
1.	I would feel uneasy if I was given a job where I had to use robots.	Situations and Interactions with Robots
2.	The word “robot” means nothing to me.	Situations and Interactions with Robots
3.	I would feel nervous operating a robot in front of other people.	Situations and Interactions with Robots
4.	I would hate the idea that robots or artificial intelligences were making judgements about things.	Situations and Interactions with Robots
5.	I would feel very nervous just standing in front of a robot.	Situations and Interactions with Robots
6.	I would feel paranoid talking with a robot.	Situations and Interactions with Robots

Table A.5.: The NARS-items and the respective sub-scales according to Nomura et al. (2006).

Nr.	Items	Sub-Scales
1.	I usually trust machines until there is a reason not to.	-
2.	For the most part, I distrust machines.	-
3.	In general, I would rely on a machine to assist me.	-
4.	My tendency to trust machines is high.	-
5.	. It is easy for me to trust machines to do their job.	-
6.	I am likely to trust a machine even when I have little knowledge about it.	-

Table A.6.: The Propensity to Trust Scale items according to Merritt et al. (2013).

Nr.	Items	Sub-Scales
1.	Useful-Useless	Usefulness
2.	Pleasant-Unpleasant	Satisfying
3.	Bad-Good	Usefulness
4.	Nice-Annoying	Satisfying
5.	Effective-Superfluous	Usefulness
6.	Irritating-Likeable	Satisfying
7.	Assisting-Worthless	Usefulness
8.	Undesirable-Desirable	Satisfying
9.	Raising Alertness-Sleep-inducing.	Usefulness

Table A.7.: The Acceptance Scale items and the respective sub-scales according to Van Der Laan et al. (1997).

Nr.	Items	Sub-Scales
1.	The system is deceptive.	-
2.	The system behaves in an underhanded manner	-
3.	I am suspicious of the system's intent, action, or outputs	-
4.	I am wary of the system.	-
5.	The system's actions will have a harmful or injurious outcome.	-
6.	I am confident in the system.	-
7.	The system provides security	-
8.	The system has integrity	-
9.	The system is dependable.	-
10.	The system is reliable.	-
11.	I can trust the system.	-
12.	I am familiar with the system.	-

Table A.8.: The Trust in Automated Systems Survey items according to Jian et al. (2000).

Nr.	Items	Sub-Scales
1.	The system is deceptive.	-
2.	I am suspicious of the system's intent, action, or outputs	-
3.	I am wary of the system.	-
4.	The system's actions will have a harmful or injurious outcome.	-
5.	I am confident in the system.	-
6.	The system is reliable.	-
7.	I can trust the system.	-

Table A.9.: The Short Learned Trust in Automation Scale items according to Kraus (2020).

Nr.	Items	Sub-Scales
1.	The system always provides the advice I require to make my decision.	Perceived Reliability
2.	The system performs reliably.	Perceived Reliability
3.	The system responds the same way under the same conditions at different times.	Perceived Reliability
4.	can rely on the system to function properly.	Perceived Reliability
5.	The system analyzes problems consistently.	Perceived Reliability
6.	The system uses appropriate methods to reach decisions.	Perceived Technical Competence
7.	The system has sound knowledge about this type of problem built into it.	Perceived Technical Competence
8.	The advice the system produces is as good as that which a highly competent person could produce.	Perceived Technical Competence
9.	The system correctly uses the information I enter.	Perceived Technical Competence
10.	The system makes use of all the knowledge and information available to it to produce its solution to the problem.	Perceived Technical Competence
11.	I know what will happen the next time I use the system because I understand how it behaves.	Perceived Understandability
12.	I understand how the system will assist me with decisions I have to make.	Perceived Understandability
13.	Although I may not know exactly how the system works, I know how to use it to make decisions about the problem.	Perceived Understandability
14.	It is easy to follow what the system does.	Perceived Understandability
15.	I recognize what I should do to get the advice I need from the system the next time I use it.	Perceived Understandability
16.	I believe advice from the system even when I don't know for certain that it is correct.	Faith
17.	When I am uncertain about a decision I believe the system rather than myself.	Faith
18.	If I am not sure about a decision, I have faith that the system will provide the best solution.	Faith
19.	When the system gives unusual advice I am confident that the advice is correct.	Faith
20.	Even if I have no reason to expect the system will be able to solve a difficult problem, I still feel certain that it will.	Faith
21.	I would feel a sense of loss if the system was unavailable and I could no longer use it.	Personal Attachment
22.	I feel a sense of attachment to using the system.	Personal Attachment
23.	I find the system suitable to my style of decision making.	Personal Attachment
24.	I like using the system for decision making.	Personal Attachment
25.	I have a personal preference for making decisions with the system.	Personal Attachment

Table A.10.: The Human-Computer Trust Scale items and the respective sub-scales according to Madsen and Gregor (2000).

Nr.	Items	Sub-Scales
1.	I think that I would like to use this system frequently.	-
2.	I found the system unnecessarily complex.	-
3.	I thought the system was easy to use.	-
4.	I think that I would need the support of a technical person to be able to use this system.	-
5.	I found the various functions in this system were well integrated.	-
6.	I thought there was too much inconsistency in this system.	-
7.	I would imagine that most people would learn to use this system very quickly.	-
8.	I found the system very cumbersome to use.	-
9.	I felt very confident using the system.	-
10.	I needed to learn a lot of things before I could get going with this system.	-

Table A.11.: The SUS items according to Brooke (1996).

Nr.	Items	Sub-Scales
1.	annoying-enjoyable	Attractiveness
2.	bad-good	Attractiveness
3.	unlikeable-pleasing	Attractiveness
4.	unpleasant-pleasant	Attractiveness
5.	unattractive-attractive	Attractiveness
6.	unfriendly-friendly	Attractiveness
7.	slow-fast	Efficiency
8.	inefficient-efficient.	Efficiency
9.	impractical-practical.	Efficiency
10.	cluttered-organized	Efficiency
11.	not understandable - understandable	Perspicuity
12.	difficult to learn - easy to learn	Perspicuity
13.	complicated - easy	Perspicuity
14.	confusing - clear	Perspicuity
15.	unpredictable-predictable	Dependability
16.	obstructive-supportive	Dependability
17.	not secure-secure	Dependability
18.	does not meet expectations-meets expectations	Dependability
19.	inferior-valuable	Stimulation
20.	boring-exciting	Stimulation
21.	not interesting-interesting	Stimulation
22.	demotivating-motivating	Stimulation
23.	dull-creative	Novelty
24.	conventional-inventive	Novelty
25.	usual-leading edge	Novelty
26.	conservative-innovative	Novelty

Table A.12.: The UEQ items and the respective sub-scales according to Laugwitz et al. (2006).

Nr.	**Items**	**Sub-Scales**
1.	obstructive-supportive	Pragmatic Quality
2.	complicated - easy	Pragmatic Quality
3.	inefficient-efficient.	Pragmatic Quality
4.	confusing - clear	Pragmatic Quality
5.	boring-exciting	Hedonic Quality
6.	not interesting-interesting	Hedonic Quality
7.	conventional-inventive	Hedonic Quality
8.	usual-leading edge	Hedonic Quality

Table A.13.: The UEQ-S items and the respective sub-scales according to Laugwitz et al. (2006).

Nr.	Items	Sub-Scales
1.	The system is accurate.	System Response Accuracy
2.	The system is unreliable.	System Response Accuracy
3.	The interaction with the system is unpredictable.	System Response Accuracy
4.	The system didn't always do what I wanted.	System Response Accuracy
5.	The system didn't always do what I expected.	System Response Accuracy
6.	The system is dependable.	System Response Accuracy
7.	The system makes few errors.	System Response Accuracy
8.	The interaction with the system is consistent.	System Response Accuracy
9.	The interaction with the system is efficient.	System Response Accuracy
10.	The system is useful.	Likeability
11.	The system is pleasant.	Likeability
12.	The system is friendly.	Likeability
13.	I was able to recover easily from errors.	Likeability
14.	I enjoyed using the system	Likeability
15.	It is clear how to speak to the system.	Likeability
16.	It is easy to learn to use the system.	Likeability
17.	I would use this system.	Likeability
18.	I felt in control of the interaction with the system.	Likeability
19.	I felt confident using the system.	Cognitive Demand
20.	I felt tense using the system.	Cognitive Demand
21.	I felt calm using the system.	Cognitive Demand
22.	A high level of concentration is required when using the system.	Cognitive Demand
23.	The system is easy to use	Cognitive Demand
24.	The interaction with the system is repetitive.	Annoyance
25.	The interaction with the system is boring.	Annoyance
26.	The interaction with the system is irritating.	Annoyance
27.	The interaction with the system is frustrating.	Annoyance
28.	I sometimes wondered if I was using the right word.	Habitability
29.	I always knew what to say to the system.	Habitability
30.	I was not always sure what the system was doing.	Habitability
31.	It is easy to lose track of where you are in an interaction with the system.	Habitability
32.	The interaction with the system is fast.	Speed
33.	The system responds too slowly.	Speed

Table A.14.: The SASSI items and the respective sub-scales according to Hone and Graham (2000).

Bibliography

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1.

Abt, C. C. (1970). Serious games. new york: Viking, 1970, 176 pp., $5.95, l.c. 79-83234. *American Behavioral Scientist*, 14(1):129–129.

Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160.

Adamczyk, P. D. and Bailey, B. P. (2004). If not now, when?: the effects of interruption at different moments within task execution. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 271–278. ACM.

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., et al. (2020). Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Ahmad, R., Siemon, D., Fernau, D., and Robra-Bissantz, S. (2020). Introducing" raffi": A personality adaptive conversational agent. In *PACIS*, page 28.

Ajenaghughrure, I. B., Sousa, S. C., Kosunen, I. J., and Lamas, D. (2019). Predictive model to assess user trust: a psycho-physiological approach. In *Proceedings of the 10th Indian Conference on Human-Computer Interaction*, pages 1–10.

Akash, K., Jain, N., and Misu, T. (2020). Toward adaptive trust calibration for level 2 driving automation. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, pages 538–547.

Allen, J., André, E., Cohen, P. R., Hakkani-Tür, D., Kaplan, R., Lemon, O., and Traum, D. (2019). Challenge discussion: advancing multimodal dialogue. In *The Handbook of Multimodal-Multisensor Interfaces: Language Processing, Software, Commercialization, and Emerging Directions-Volume 3*, pages 191–217.

Allen, J., Blaylock, N., and Ferguson, G. (2002). A problem solving model for collaborative agents. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, pages 774–781.

Allen, J. F. and Perrault, C. R. (1980). Analyzing intention in utterances. *Artificial intelligence*, 15(3):143–178.

Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., Light, M., Martin, N., Miller, B., Poesio, M., et al. (1995). The trains project: A case study in building a conversational planning agent. *Journal of Experimental & Theoretical Artificial Intelligence*, 7(1):7–48.

Allensbach, I. (2019). Allensbacher markt- und werbetraeger-analyse - awa 2019, zitiert nach de.statista.com. Retrieved on 07.04.2020 at 18.37.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Alves, T., Natálio, J., Henriques-Calado, J., and Gama, S. (2020). Incorporating personality in user interface design: A review. *Personality and Individual Differences*, 155:109709.

Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P. N., Inkpen, K., et al. (2019). Guidelines for human-ai interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–13.

Anderson, E., Fannin, T., and Nelson, B. (2018). Levels of aviation autonomy. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pages 1–8. IEEE.

André, E. and Pelachaud, C. (2010). Interacting with embodied conversational agents. In *Speech technology*, pages 123–149. Springer.

Andre, E., Rehm, M., Minker, W., and Bühler, D. (2004). Endowing spoken language dialogue systems with emotional intelligence. In *Tutorial and Research Workshop on Affective Dialogue Systems*, pages 178–187. Springer.

ArangoDB (2021). Arangodb - graph and beyond. `https://www.arangodb.com/`. Last Accessed: 2021-11-30.

Babel, F., Kraus, J., Hock, P., Asenbauer, H., and Baumann, M. (2021). Investigating the validity of online robot evaluations: Comparison of findings from an one-sample online and laboratory study. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, pages 116–120.

Baddeley, A. (1994). The magical number seven: Still magic after all these years?

Baker, C., Saxe, R., and Tenenbaum, J. (2011). Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, volume 33.

Baker, R. S., D'Mello, S. K., Rodrigo, M. M. T., and Graesser, A. C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, 68(4):223–241.

Balaraman, V. and Magnini, B. (2020). Proactive systems and influenceable users: Simulating proactivity in task-oriented dialogues. In *Proceedings of the 24th Workshop on the Semantics and Pragmatics of Dialogue-Full Papers, Virually at Brandeis, Waltham, New Jersey, July. SEMDIAL.*

Baraglia, J., Cakmak, M., Nagai, Y., Rao, R., and Asada, M. (2016). Initiative in robot assistance during collaborative task execution. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, pages 67–74. IEEE Press.

Baron-Cohen, S., Leslie, A. M., and Frith, U. (1985). Does the autistic child have a "theory of mind"? *Cognition*, 21(1):37–46.

Bartneck, C. and Forlizzi, J. (2004). A design-centred framework for social human-robot interaction. In *RO-MAN 2004. 13th IEEE international workshop on robot and human interactive communication (IEEE Catalog No. 04TH8759)*, pages 591–594. IEEE.

Bateman, T. S. and Crant, J. M. (1993). The proactive component of organizational behavior: A measure and correlates. *Journal of organizational behavior*, 14(2):103–118.

Batrinca, L., Lepri, B., Mana, N., and Pianesi, F. (2012). Multimodal recognition of personality traits in human-computer collaborative tasks. In *Proceedings of the 14th ACM international conference on multimodal interaction*, pages 39–46.

Beer, J. M., Fisk, A. D., and Rogers, W. A. (2014). Toward a framework for levels of robot autonomy in human-robot interaction. *Journal of human-robot interaction*, 3(2):74–99.

Behnke, G., Bercher, P., Kraus, M., Schiller, M., Mickeleit, K., Häge, T., Dorna, M., Dambier, M., Minker, W., Glimm, B., and Biundo, S. (2020). New developments for Robert – Assisting novice users even better in DIY projects. In *Proc. of the 30th Int. Conf. on Autom. Plan. and Sched. (ICAPS 2020).*

Behnke, G., Höller, D., and Biundo, S. (2018a). totsat-totally-ordered hierarchical planning through sat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Behnke, G., Höller, D., and Biundo, S. (2018b). Tracking branches in trees – A propositional encoding for solving partially-ordered HTN planning problems. In *Proc. of the 30th Int. Conf. on Tools with Art. Int. (ICTAI 2018).*

Behnke, G., Höller, D., and Biundo, S. (2019a). Bringing order to chaos – A compact representation of partial order in SAT-based HTN planning. In *Proc. of the 33rd AAAI Conf. on AI (AAAI 2019).*

Behnke, G., Höller, D., and Biundo, S. (2019b). Finding optimal solutions in HTN planning – A SAT-based approach. In *Proc. of the 28th Int. Joint Conf. on AI (IJCAI 2019).*

Behnke, G., Ponomaryov, D., Schiller, M., Bercher, P., Nothdurft, F., Glimm, B., and Biundo, S. (2015). Coherence across components in cognitive systems – One ontology to rule them all. In *Proc. of the 24th Int. Joint Conf. on AI (IJCAI 2015).*

Behnke, G., Schiller, M., Kraus, M., Bercher, P., Schmautz, M., Dorna, M., Dambier, M., Minker, W., Glimm, B., and Biundo, S. (2019c). Alice in DIY wonderland or: Instructing novice users on how to use tools in DIY projects. *AI Communications*, (Preprint):1–27.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Benítez-Guijarro, A., Bond, R., Booth, F., Callejas, Z., Ennis, E., Esposito, A., Kraus, M., McConvey, G., McTear, M., Mulvenna, M., et al. (2020). Co-creating requirements and assessing end-user acceptability of a voice-based chatbot to support mental health: A thematic analysis of a living lab workshop. In *Conversational Dialogue Systems for the Next Decade*, pages 201–212. Springer.

Bercher, P., Alford, R., and Höller, D. (2019). A survey on hierarchical planning – One abstract idea, many concrete realizations. In *Proc. of the 28th Int. Joint Conf. on AI (IJCAI 2019).*

Bercher, P., Behnke, G., Kraus, M., Schiller, M., Manstetten, D., Dambier, M., Dorna, M., Minker, W., Glimm, B., and Biundo, S. (2021). Do it yourself, but not alone: Companion-technology for home improvement—bringing a planning-based interactive diy assistant to life. *KI-Künstliche Intelligenz*, 35(3):367–375.

Bercher, P., Biundo, S., Geier, T., Hoernle, T., Nothdurft, F., Richter, F., and Schattenberg, B. (2014). Plan, repair, execute, explain—how planning helps to assemble your home theater. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 24.

Bickmore, T. and Cassell, J. (1999). Small talk and conversational storytelling in embodied conversational interface agents. In *AAAI fall symposium on narrative intelligence*, pages 87–92.

Bickmore, T. W. and Picard, R. W. (2005). Establishing and maintaining long-term human-computer relationships. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(2):293–327.

Biondi, F., Alvarez, I., and Jeong, K.-A. (2019). Human–vehicle cooperation in automated driving: A multidisciplinary review and appraisal. *International Journal of Human–Computer Interaction*, 35(11):932–946.

Bishop, C. M. et al. (1995). *Neural networks for pattern recognition.* Oxford university press.

Biundo, S. and Wendemuth, A. (2016). Companion-technology for cognitive technical systems. *KI-Künstliche Intelligenz*, 30(1):71–75.

Black, A. W. and Eskenazi, M. (2009). The spoken dialogue challenge. In *Proceedings of the SIGDIAL 2009 Conference*, pages 337–340.

Bohus, D. and Rudnicky, A. I. (2009). The ravenclaw dialog management framework: Architecture and systems. *Computer Speech & Language*, 23(3):332–361.

Bordes, A., Boureau, Y.-L., and Weston, J. (2016). Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

Branigan, H. (2006). Perspectives on multi-party dialogue. *Research on Language and Computation*, 4(2):153–177.

Bratman, M. E., Israel, D. J., and Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational intelligence*, 4(3):349–355.

Braun, M., Mainz, A., Chadowitz, R., Pfleging, B., and Alt, F. (2019). At your service: Designing voice assistant personalities to improve automotive user interfaces. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–11.

Brave, S., Nass, C., and Hutchinson, K. (2005). Computers that care: investigating the effects of orientation of emotion exhibited by an embodied computer agent. *International journal of human-computer studies*, 62(2):161–178.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification and regression trees*. Routledge.

Brooke, J. (1996). Sus: a "quick and dirty'usability. *Usability evaluation in industry*, page 189.

Bruner, J. S., Goodnow, J. J., and Austin, G. A. (2017). *A study of thinking*. Routledge.

Budzianowski, P., Wen, T.-H., Tseng, B.-H., Casanueva, I., Ultes, S., Ramadan, O., and Gašić, M. (2018). Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Bui, T. H., Poel, M., Nijholt, A., and Zwiers, J. (2009). A tractable hybrid ddn–pomdp approach to affective dialogue modeling for probabilistic frame-based dialogue systems. *Natural Language Engineering*, 15(2):273–307.

Bull, S. and Kay, J. (2010). Open learner models. In *Advances in intelligent tutoring systems*, pages 301–322. Springer.

Byrne, E. A. and Parasuraman, R. (1996). Psychophysiology and adaptive automation. *Biological psychology*, 42(3):249–268.

Cai, W. and Chen, L. (2020). Predicting user intents and satisfaction with dialogue-based conversational recommendations. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pages 33–42.

Callejas, Z., Griol, D., and López-Cózar, R. (2011). Predicting user mental states in spoken dialogue systems. *EURASIP Journal on Advances in Signal Processing*, 2011(1):6.

Candello, H. and Pinhanez, C. (2018). Recovering from dialogue failures using multiple agents in wealth management advice. In *Studies in conversational UX design*, pages 139–157. Springer.

Card, S. K. (1981). The model human processor: A model for making engineering calculations of human performance. In *Proceedings of the Human Factors Society Annual Meeting*, volume 25, pages 301–305. SAGE Publications Sage CA: Los Angeles, CA.

Carruthers, P. and Smith, P. K. (1996). *Theories of theories of mind.* Cambridge university press.

Cassell, J. and Bickmore, T. (2002). Negotiated collusion: Modeling social language and its relationship effects in intelligent agents. *User Modeling and User-Adapted Interaction.*

Cattell, R. B., Eber, H. W., and Tatsuoka, M. M. (1970). *Handbook for the sixteen personality factor questionnaire (16 PF): In clinical, educational, industrial, and research psychology, for use with all forms of the test.* Institute for Personality and Ability Testing.

Cha, N., Kim, A., Park, C. Y., Kang, S., Park, M., Lee, J.-G., Lee, S., and Lee, U. (2020). Hello there! is now a good time to talk? opportune moments for proactive interactions with smart speakers. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–28.

Chandler, P. and Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and instruction*, 8(4):293–332.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27.

Chaves, A. P. and Gerosa, M. A. (2018). Single or multiple conversational agents? an interactional coherence comparison. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Chaves, A. P. and Gerosa, M. A. (2021). How should my chatbot interact? a survey on social characteristics in human–chatbot interaction design. *International Journal of Human–Computer Interaction*, 37(8):729–758.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Chesñevar, C., Maguitman, A. G., and González, M. P. (2009). Empowering recommendation technologies through argumentation. In *Argumentation in artificial intelligence*, pages 403–422. Springer.

Chiang, T.-R., Huang, C.-W., Su, S.-Y., and Chen, Y.-N. (2020). Learning multi-level information for dialogue response selection by highway recurrent transformer. *Computer Speech & Language*, 63:101073.

Chien, S.-Y., Sycara, K., Liu, J.-S., and Kumru, A. (2016). Relation between trust attitudes toward automation, hofstede's cultural dimensions, and big five personality traits. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 60, pages 841–845. SAGE Publications Sage CA: Los Angeles, CA.

Chollet, F. e. a. (2015). Keras. `https://keras.io`.

Christakopoulou, K., Beutel, A., Li, R., Jain, S., and Chi, E. H. (2018). Q&r: A two-stage approach toward interactive recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 139–148. ACM.

Christakopoulou, K., Radlinski, F., and Hofmann, K. (2016). Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 815–824. ACM.

Chu-Carroll, J. (2000). Evaluating automatic dialogue strategy adaptation for a spoken dialogue system. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics.*

Chu-Carroll, J. and Brown, M. K. (1998). An evidential model for tracking initiative in collaborative dialogue interactions. In *Computational Models of Mixed-Initiative Interaction*, pages 49–87. Springer.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555.*

Ciechanowski, L., Przegalinska, A., Magnuski, M., and Gloor, P. (2019). In the shades of the uncanny valley: An experimental study of human–chatbot interaction. *Future Generation Computer Systems*, 92:539–548.

Clark, H. H. (1996). *Using language.* Cambridge university press.

Clark, L., Pantidi, N., Cooney, O., Doyle, P., Garaialde, D., Edwards, J., Spillane, B., Gilmartin, E., Murad, C., Munteanu, C., et al. (2019). What makes a good conversation? challenges in designing truly conversational agents. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Cohen, J. (1988). Statistical power analysis jbr the behavioral. *Sciences. Hillsdale (NJ): Lawrence Erlbaum Associates*, pages 18–74.

Cohen, P. R. and Levesque, H. J. (1990). Intention is choice with commitment. *Artificial intelligence*, 42(2-3):213–261.

Cohen, P. R. and Perrault, C. R. (1979). Elements of a plan-based theory of speech acts. *Cognitive science*, 3(3):177–212.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167.

Core, M. G. and Allen, J. (1997). Coding dialogs with the damsl annotation scheme. In *AAAI fall symposium on communicative action in humans and machines*, volume 56, pages 28–35. Boston, MA.

Corneanu, C. A., Simón, M. O., Cohn, J. F., and Guerrero, S. E. (2016). Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1548–1568.

Corti, K. (2006). Games-based learning; a serious business application. *Informe de PixelLearning*, 34(6):1–20.

Costa, P. T. and McCrae, R. R. (1989). *NEO PI/FFI manual supplement for use with the NEO Personality Inventory and the NEO Five-Factor Inventory.* Psychological Assessment Resources.

Craig, S. D., D'Mello, S., Witherspoon, A., and Graesser, A. (2008). Emote aloud during learning with autotutor: Applying the facial action coding system to cognitive–affective states during learning. *Cognition and Emotion*, 22(5):777–788.

Cramer, H., Evers, V., Ramlal, S., Van Someren, M., Rutledge, L., Stash, N., Aroyo, L., and Wielinga, B. (2008). The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-adapted interaction*, 18(5):455.

Cramer, H., Kemper, N., Amin, A., and Evers, V. (2009). The effects of robot touch and proactive behaviour on perceptions of human-robot interactions. In *2009 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 275–276. IEEE.

Crant, J. M. (2000). Proactive behavior in organizations. *Journal of management*, 26(3):435–462.

Cuadra, A., Li, S., Lee, H., Cho, J., and Ju, W. (2021). My bad! repairing intelligent voice assistant errors improves interaction. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–24.

Cuzzolin, F., Morelli, A., Cirstea, B., and Sahakian, B. J. (2020). Knowing me, knowing you: theory of mind in ai. *Psychological medicine*, 50(7):1057–1061.

Davidovi, Š. and Guliani, K. (2015). Reliable cron across the planet. *Queue*, 13(3):30–39.

Davis, F. D., Bagozzi, R. P., and Warshaw, P. R. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management science*, 35(8):982–1003.

de Graaf, M. M., Ben Allouch, S., and Van Dijk, J. A. (2019). Why would i use this in my home? a model of domestic social robot acceptance. *Human–Computer Interaction*, 34(2):115–173.

de Visser, E. and Parasuraman, R. (2011). Adaptive aiding of human-robot teaming: Effects of imperfect automation on performance, trust, and workload. *Journal of Cognitive Engineering and Decision Making*, 5(2):209–231.

Deutsch, M. (1960). The effect of motivational orientation upon trust and suspicion. *Human relations*, 13(2):123–139.

DeYoung, C., Gray, J., Corr, P., and Matthews, G. (2009). The cambridge handbook of personality psychology.

Dimoka, A., Davis, F. D., Gupta, A., Pavlou, P. A., Banker, R. D., Dennis, A. R., Ischebeck, A., Müller-Putz, G., Benbasat, I., Gefen, D., et al. (2012). On the use of neurophysiological tools in is research: Developing a research agenda for neurois. *MIS quarterly*, pages 679–702.

Dingler, T., Tag, B., Lehrer, S., and Schmidt, A. (2018). Reading scheduler: proactive recommendations to help users cope with their daily reading volume. In *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia*, pages 239–244.

D'Mello, S. and Graesser, A. (2011). The half-life of cognitive-affective states during complex learning. *Cognition & Emotion*, 25(7):1299–1308.

D'Mello, S. K., Craig, S. D., Sullins, J., and Graesser, A. C. (2006). Predicting affective states expressed through an emote-aloud procedure from autotutor's mixed-initiative dialogue. *International Journal of Artificial Intelligence in Education*, 16(1):3–28.

Du, M., Liu, N., and Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77.

Duijvelshoff, W. (2017). Use-cases and ethics of chatbots on plek: a social intranet for organizations. In *Workshop on chatbots and artificial intelligence*.

Dušek, O., Novikova, J., and Rieser, V. (2020). Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156.

Eckert, W., Levin, E., and Pieraccini, R. (1997). User modeling for spoken dialogue system evaluation. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 80–87. IEEE.

Edwards, A., Edwards, C., Westerman, D., and Spence, P. R. (2019). Initial expectations, interactions, and beyond with social robots. *Computers in Human Behavior*, 90:308–314.

Edwards, C., Edwards, A., Spence, P. R., and Westerman, D. (2016). Initial interaction expectations with robots: Testing the human-to-human interaction script. *Communication Studies*, 67(2):227–238.

Ekman, P. (1993). Facial expression and emotion. *American psychologist*, 48(4):384.

Elsweiler, D., Harvey, M., Ludwig, B., and Said, A. (2015). Bringing the" healthy" into food recommenders. In *DMRS*, pages 33–36.

Endsley, M. R. and Kiris, E. O. (1995). The out-of-the-loop performance problem and level of control in automation. *Human factors*, 37(2):381–394.

Evans, A. M. and Revelle, W. (2008). Survey and behavioral measurements of interpersonal trust. *Journal of Research in Personality*, 42(6):1585–1593.

Eysenck, H. J. (1963). Biological basis of personality. *Nature*, 199(4898):1031–1034.

Eysenck, H. J. (1966). Personality and experimental psychology. *Bulletin of the British Psychological Society.*

Eysenck, H. J. and Eysenck, S. B. G. (1975). *Manual of the Eysenck Personality Questionnaire (junior & adult).* Hodder and Stoughton Educational.

Eysenck, M. W., Derakshan, N., Santos, R., and Calvo, M. G. (2007). Anxiety and cognitive performance: attentional control theory. *Emotion*, 7(2):336.

Fan, H. and Poole, M. S. (2006). What is personalization? perspectives on the design and implementation of personalization in information systems. *Journal of Organizational Computing and Electronic Commerce*, 16(3-4):179–202.

Faulring, A., Myers, B., Mohnkern, K., Schmerl, B., Steinfeld, A., Zimmerman, J., Smailagic, A., Hansen, J., and Siewiorek, D. (2010). Agent-assisted task management that reduces email overload. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 61–70.

Ferraz de Abreu, J., Santos, R., Silva, T., Marques, T., and Cardoso, B. (2019). Proactivity: The next step in voice assistants for the tv ecosystem. In *Iberoamerican Conference on Applications and Usability of Interactive TV*, pages 103–116. Springer.

Flemisch, F., Kelsch, J., Löper, C., Schieben, A., Schindler, J., and Heesen, M. (2008). Cooperative control and active interfaces for vehicle assistance and automation.

FlexBE (2018). Ros flexbe. `http://wiki.ros.org/flexbe`. Last Accessed: 2021-11-30.

Forbes-Riley, K. and Litman, D. (2006). Modelling user satisfaction and student learning in a spoken dialogue tutoring system with generic, tutoring, and user affect parameters. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 264–271.

Frese, M. and Fay, D. (2001). 4. personal initiative: An active performance concept for work in the 21st century. *Research in organizational behavior*, 23:133–187.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer.

Freyne, J. and Berkovsky, S. (2010). Intelligent food planning: personalized recipe recommendation. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 321–324.

Friemel, C., Morana, S., Pfeiffer, J., and Maedche, A. (2018). On the role of users' cognitive-affective states for user assistance invocation. In *Information Systems and Neuroscience*, pages 37–46. Springer.

Frølund, L. and Nielsen, J. (2009). The reflective meta-dialogue in psychodynamic supervision. *Nordic Psychology*, 61(4):85–105.

Fung, P., Dey, A., Siddique, F. B., Lin, R., Yang, Y., Bertero, D., Wan, Y., Chan, R. H. Y., and Wu, C.-S. (2016). Zara: a virtual interactive dialogue system incorporating emotion, sentiment and personality recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 278–281.

Galescu, L., Teng, C. M., Allen, J., and Perera, I. (2018). Cogent: A generic dialogue system shell based on a collaborative problem solving model. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 400–409.

Garlan, D. and Schmerl, B. (2007). The radar architecture for personal cognitive assistance. *International Journal of Software Engineering and Knowledge Engineering*, 17(02):171–190.

Gasper, K. and Clore, G. L. (2000). Do you have to pay attention to your feelings to be influenced by them? *Personality and Social Psychology Bulletin*, 26(6):698–711.

Ginzburg, J. et al. (1996). Dynamics and the semantics of dialogue. *Logic, language and computation*, 1:221–237.

Glas, D. F., Kanda, T., Ishiguro, H., and Hagita, N. (2008). Simultaneous teleoperation of multiple social robots. In *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 311–318. IEEE.

Glass, A., McGuinness, D. L., and Wolverton, M. (2008). Toward establishing trust in adaptive agents. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 227–236.

Glikson, E. and Woolley, A. W. (2020). Human trust in artificial intelligence: Review of empirical research. *Academy of Management Annals*, 14(2):627–660.

Gnjatović, M. and Rösner, D. (2008). Adaptive dialogue management in the nimitek prototype system. In *International Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-Based Systems*, pages 14–25. Springer.

Gordon, R. M. (1986). Folk psychology as simulation. *Mind & language*, 1(2):158–171.

Graesser, A., D'Mello, S., Chipman, P., King, B., and McDANIEL, B. (2007). Exploring relationships between affect and learning with autotutor. In *Proc Int Conf AIED*.

Graesser, A. C., Chipman, P., Haynes, B. C., and Olney, A. (2005). Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618.

Graesser, A. C., VanLehn, K., Rosé, C. P., Jordan, P. W., and Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI magazine*, 22(4):39–39.

Graf, B., Hans, M., and Schraft, R. D. (2004). Care-o-bot ii—development of a next generation robotic home assistant. *Autonomous robots*, 16(2):193–205.

Grant, A. M. and Ashford, S. J. (2008). The dynamics of proactivity at work. *Research in organizational behavior*, 28:3–34.

Green, M. (1986). A survey of three dialogue models. *ACM Transactions on Graphics (TOG)*, 5(3):244–275.

Gregor, S. and Benbasat, I. (1999). Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS quarterly*, pages 497–530.

Grice, H. P. (1975). Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Grosinger, J., Pecora, F., and Saffiotti, A. (2016). Making robots proactive through equilibrium maintenance. In *IJCAI*, pages 3375–3381.

Grover, S., Sengupta, S., Chakraborti, T., Mishra, A. P., and Kambhampati, S. (2020). Radar: automated task planning for proactive decision support. *Human–Computer Interaction*, 35(5-6):387–412.

Gulati, S., Sousa, S., and Lamas, D. (2019). Design, development and evaluation of a human-computer trust scale. *Behaviour & Information Technology*, 38(10):1004–1015.

Guo, D., Tang, D., Duan, N., Zhou, M., and Yin, J. (2018). Dialog-to-action: Conversational question answering over a large-scale knowledge base. In *Advances in Neural Information Processing Systems*, pages 2942–2951.

Guo, Y. and Yang, X. J. (2020). Modeling and predicting trust dynamics in human–robot teaming: A bayesian inference approach. *International Journal of Social Robotics*, pages 1–11.

Hammer, S., Wißner, M., and André, E. (2015). Trust-based decision-making for smart and adaptive environments. *User Modeling and User-Adapted Interaction*, 25(3):267–293.

Hancock, P. A., Billings, D. R., Schaefer, K. E., Chen, J. Y., De Visser, E. J., and Parasuraman, R. (2011). A meta-analysis of factors affecting trust in human-robot interaction. *Human factors*, 53(5):517–527.

Hartmann, B., Mancini, M., and Pelachaud, C. (2005). Implementing expressive gesture synthesis for embodied conversational agents. In *International Gesture Workshop*, pages 188–199. Springer.

Hassenzahl, M., Burmester, M., and Koller, F. (2003). Attrakdiff: Ein fragebogen zur messung wahrgenommener hedonischer und pragmatischer qualität. In *Mensch & computer 2003*, pages 187–196. Springer.

Hastie, H. W., Prasad, R., and Walker, M. A. (2002). Automatic evaluation: Using a date dialogue act tagger for user satisfaction and task completion prediction. In *LREC*. Citeseer.

Heinroth, T., Denich, D., and Schmitt, A. (2010). Owlspeak-adaptive spoken dialogue within intelligent environments. In *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 666–671. IEEE.

Henderson, M., Thomson, B., and Williams, J. D. (2014). The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 263–272.

Hibbeln, M. T., Jenkins, J. L., Schneider, C., Valacich, J., and Weinmann, M. (2017). How is your user feeling? inferring emotion through human-computer interaction devices. *Mis Quarterly*, 41(1):1–21.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hoff, K. A. and Bashir, M. (2015). Trust in automation: Integrating empirical evidence on factors that influence trust. *Human factors*, 57(3):407–434.

Hoffman, G. and Breazeal, C. (2007). Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 1–8.

Hoffman, M. L. (2001). *Empathy and moral development: Implications for caring and justice*. Cambridge University Press.

Hone, K. S. and Graham, R. (2000). Towards a tool for the subjective assessment of speech system interfaces (sassi). *Natural Language Engineering*, 6(3-4):287–303.

Honold, F., Bercher, P., Richter, F., Nothdurft, F., Geier, T., Barth, R., Hörnle, T., Schüssel, F., Reuter, S., Rau, M., et al. (2014). Companion-technology: towards user- and situation-adaptive functionality of technical systems. In *2014 International Conference on Intelligent Environments*, pages 378–381. IEEE.

Horvitz, E. (1998). Lumiere project: Bayesian reasoning for automated assistance. *Decision Theory & Adaptive Systems Group, Microsoft Research. Microsoft Corp. Redmond, WA*.

Horvitz, E. (1999). Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166. ACM.

Huang, C.-M. and Mutlu, B. (2016). Anticipatory robot control for efficient human-robot collaboration. In *The eleventh ACM/IEEE international conference on human robot interaction*, pages 83–90. IEEE Press.

Hudlicka, E. (2003). To feel or not to feel: The role of affect in human–computer interaction. *International journal of human-computer studies*, 59(1-2):1–32.

Hunt, A. J. and Black, A. W. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 373–376. IEEE.

Ikemoto, Y., Asawavetvutt, V., Kuwabara, K., and Huang, H.-H. (2019). Tuning a conversation strategy for interactive recommendations in a chatbot setting. *Journal of Information and Telecommunication*, 3(2):180–195.

Iqbal, S. T. and Bailey, B. P. (2008). Effects of intelligent notification management on users and their tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 93–102.

Isbell, C. L. and Pierce, J. S. (2005). An IP continuum for adaptive interface design. In *Proc. of HCI International*.

Ischen, C., Araujo, T., Voorveld, H., van Noort, G., and Smit, E. (2019). Privacy concerns in chatbot interactions. In *International Workshop on Chatbot Research and Design*, pages 34–48. Springer.

ISO (2019). Iso 9241-210:2019 ergonomics of human-system interaction — part 210: Human-centred design for interactive systems. `https://www.iso.org/standard/77520.html`.

Jain, A., Pecune, F., Matsuyama, Y., and Cassell, J. (2018a). A user simulator architecture for socially-aware conversational agents. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 133–140.

Jain, M., Kumar, P., Kota, R., and Patel, S. N. (2018b). Evaluating and informing the design of chatbots. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pages 895–906.

Jameson, A. (2007). Adaptive interfaces and agents. In *The human-computer interaction handbook*, pages 459–484. CRC Press.

Jenkins, J. L., Anderson, B. B., Vance, A., Kirwan, C. B., and Eargle, D. (2016). More harm than good? how messages that interrupt can make us vulnerable. *Information Systems Research*, 27(4):880–896.

Jian, J.-Y., Bisantz, A. M., and Drury, C. G. (2000). Foundations for an empirically determined scale of trust in automated systems. *International Journal of Cognitive Ergonomics*, 4(1):53–71.

John, O. P., Srivastava, S., et al. (1999). The big-five trait taxonomy: History, measurement, and theoretical perspectives.

Jokinen, K. and Kanto, K. (2004). User expertise modeling and adaptivity in a speech-based e-mail system. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 87–94.

Juang, B. H. and Rabiner, L. R. (1991). Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272.

Jurafsky, D. and Martin, H. (2020). Speech and language processing (draft). 2017. *URL: https://web. stanford. edu/˜ jurafsky/slp3.*

Jurcıcek, F., Keizer, S., Gašic, M., Mairesse, F., Thomson, B., Yu, K., and Young, S. (2011). Real user evaluation of spoken dialogue systems using amazon mechanical turk. In *Proceedings of INTERSPEECH*, volume 11.

Kaber, D. B., Riley, J. M., Tan, K.-W., and Endsley, M. R. (2001). On the design of adaptive automation for complex systems. *International Journal of Cognitive Ergonomics*, 5(1):37–57.

Karrer, K., Glaser, C., Clemens, C., and Bruder, C. (2009). Technikaffinität erfassen–der fragebogen ta-eg. *Der Mensch im Mittelpunkt technischer Systeme*, 8:196–201.

Kato, Y., Kanda, T., and Ishiguro, H. (2015). May i help you?: Design of human-like polite approaching behavior. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 35–42. ACM.

Katz, S., Connelly, J., and Wilson, C. (2007). Out of the lab and into the classroom: An evaluation of reflective dialogue in andes. *Frontiers in Artificial Intelligence and Applications*, 158:425.

Katz, S., O'Donnell, G., and Kay, H. (2000). An approach to analyzing the role and structure of reflective dialogue. *International Journal of Artificial Intelligence in Education*, 11:320–343.

Kecman, V. (2005). Support vector machines–an introduction. In *Support vector machines: theory and applications*, pages 1–47. Springer.

Kelley, J. F. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41.

Khalid, H., Liew, W., Helander, M., and Loo, C. (2016). Prediction of trust in scripted dialogs using neuro-fuzzy method. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 1558–1562. IEEE.

Kiely, K. M. (2014). *Cognitive Function*, pages 974–978. Springer Netherlands, Dordrecht.

Kim, A., Choi, W., Park, J., Kim, K., and Lee, U. (2019). Predicting opportune moments for in-vehicle proactive speech services. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pages 101–104.

Kim, Y., Lee, H., and Provost, E. M. (2013). Deep learning for robust feature generation in audiovisual emotion recognition. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 3687–3691. IEEE.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv e-prints*, pages arXiv–1412.

Klepsch, M., Schmitz, F., and Seufert, T. (2017). Development and validation of two instruments measuring intrinsic, extraneous, and germane cognitive load. *Frontiers in psychology*, 8:1997.

Kocaballi, A. B., Berkovsky, S., Quiroz, J. C., Laranjo, L., Tong, H. L., Rezazadegan, D., Briatore, A., and Coiera, E. (2019). The personalization of conversational agents in health care: systematic review. *Journal of medical Internet research*, 21(11):e15360.

Komatani, K., Ueno, S., Kawahara, T., and Okuno, H. G. (2005). User modeling in spoken dialogue systems to generate flexible guidance. *User Modeling and User-Adapted Interaction*, 15(1):169–183.

Kontogiorgos, D., Pereira, A., Sahindal, B., van Waveren, S., and Gustafson, J. (2020). Behavioural responses to robot conversational failures. In *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 53–62. IEEE.

Kraus, J. M. (2020). *Psychological processes in the formation and calibration of trust in automation.* PhD thesis, Universität Ulm.

Kraus, M., Betancourt, D., and Minker, W. (2022a). On using cognitive-affective user states for proactive human-computer dialogue. In *Under Review*, page XXX. XXX.

Kraus, M., Fischbach, F., Jansen, P., and Minker, W. (2020a). A comparison of explicit and implicit proactive dialogue strategies for conversational recommendation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 429–435.

Kraus, M., Kraus, J., Baumann, M., and Minker, W. (2018). Effects of gender stereotypes on trust and likability in spoken human-robot interaction. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).*

Kraus, M., Schiller, M., Behnke, G., Bercher, P., Biundo, S., Glimm, B., and Minker, W. (2019). A multimodal dialogue framework for cloud-based companion systems. In *9th International Workshop on Spoken Dialogue System Technology*, pages 405–410. Springer.

Kraus, M., Schiller, M., Behnke, G., Bercher, P., Dorna, M., Dambier, M., Glimm, B., Biundo, S., and Minker, W. (2020b). "was that successful?" on integrating proactive meta-dialogue in a diy-assistant using multimodal cues. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, page 585–594, New York, NY, USA. Association for Computing Machinery.

Kraus, M., Seldschopf, P., and Minker, W. (2021a). Towards the development of a trustworthy chatbot for mental health applications. In *International Conference on Multimedia Modeling*, pages 354–366. Springer.

Kraus, M., Wagner, N., Callejas, Z., and Minker, W. (2021b). The role of trust in proactive conversational assistants. *IEEE Access*, 9:112821–112836.

Kraus, M., Wagner, N., and Minker, W. (2020c). Effects of proactive dialogue strategies on human-computer trust. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '20, page 107–116, New York, NY, USA. Association for Computing Machinery.

Kraus, M., Wagner, N., and Minker, W. (2021c). Modelling and predicting trust for developing proactive dialogue strategies in mixed-initiative interaction. In *Proceedings of the 2021 International Conference on Multimodal Interaction*, pages 131–140.

Kraus, M., Wagner, N., and Minker, W. (2022b). Prodial – an annotated proactive dialogue act corpus for conversational assistants using crowdsourcing. In *Proceedings of the 13th International Conference on Language Resources and Evaluation (LREC 2022).*

Kraus, M., Wagner, N., Minker, W., Agrawal, A., Schmidt, A., Krishna Prasad, P., and Ertel, W. (2022c). Kurt: A household assistance robot capable of proactive dialogue. In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, pages 855–859.

Kraus, M., Wagner, N., Riekenbrauck, R., and Minker, W. (2022d). Improving human-machine cooperation using trust-adaptive proactive dialogue modelling. In *Under review COLING*.

Kraus, M., Wagner, N., Untereiner, N., and Minker, W. (2022e). Including social expectations for trustworthy proactive human-robot dialogue. In *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '22)*.

Kreyssig, F., Casanueva, I., Budzianowski, P., and Gasic, M. (2018). Neural user simulation for corpus-based policy optimisation of spoken dialogue systems. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 60–69.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Kwon, M., Jung, M. F., and Knepper, R. A. (2016). Human expectations of social robots. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 463–464. IEEE.

Laban, G., George, J.-N., Morrison, V., and Cross, E. S. (2021). Tell me more! assessing interactions with social robots from speech. *Paladyn, Journal of Behavioral Robotics*, 12(1):136–159.

L'Abbate, M., Thiel, U., and Kamps, T. (2005). Can proactive behavior turn chatterbots into conversational agents? In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 173–179. IEEE.

Larsson, S. and Traum, D. R. (2000). Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural language engineering*, 6(3-4):323–340.

Laugwitz, B., Schrepp, M., and Held, T. (2006). Konstruktion eines fragebogens zur messung der user experience von softwareprodukten. *Mensch und Computer 2006: Mensch und Computer im Strukturwandel*, pages 125–134.

Law, T., Chita-Tegmark, M., and Scheutz, M. (2020). The interplay between emotional intelligence, trust, and gender in human–robot interaction. *International Journal of Social Robotics*, pages 1–13.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lee, J. D. and Moray, N. (1994). Trust, self-confidence, and operators' adaptation to automation. *International journal of human-computer studies*, 40(1):153–184.

Lee, J. D. and See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human factors*, 46(1):50–80.

Lee, J. J., Knox, B., Baumann, J., Breazeal, C., and DeSteno, D. (2013). Computationally modeling interpersonal trust. *Frontiers in psychology*, 4:893.

Lee, S. and Eskenazi, M. (2012). An unsupervised approach to user simulation: toward self-improving dialog systems. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 50–59.

Lemon, O. (2008). Adaptive natural language generation in dialogue using reinforcement learning. In *LONDIAL 2008 the 12th Workshop on the Semantics and Pragmatics of Dialogue*, page 149.

Levin, E. and Pieraccini, R. (1997). A stochastic model of computer-human interaction for learning dialogue strategies. In *Fifth European Conference on Speech Communication and Technology*.

Levin, E., Pieraccini, R., and Eckert, W. (2000). A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on speech and audio processing*, 8(1):11–23.

Lewis, C. (1982). *Using the" thinking-aloud" method in cognitive interface design*. IBM TJ Watson Research Center Yorktown Heights, NY.

Li, X. and Ji, Q. (2005). Active affective state detection and user assistance with dynamic bayesian networks. *IEEE transactions on systems, man, and cybernetics-part a: systems and humans*, 35(1):93–105.

Liang, K., Chau, A., Li, Y., Lu, X., Yu, D., Zhou, M., Jain, I., Davidson, S., Arnold, J., Nguyen, M., et al. (2020). Gunrock 2.0: A user adaptive social conversational system. *arXiv preprint arXiv:2011.08906*.

Liao, Q. V., Davis, M., Geyer, W., Muller, M., and Shami, N. S. (2016). What can you do? studying social-agent orientation and agent proactive interactions with an agent for employees. In *Proceedings of the 2016 acm conference on designing interactive systems*, pages 264–275.

Liao, W., Zhang, W., Zhu, Z., Ji, Q., and Gray, W. D. (2006). Toward a decision-theoretic framework for affect recognition and user assistance. *International Journal of Human-Computer Studies*, 64(9):847–873.

Lieberman, H. (2009). User interface goals, ai opportunities. *AI Magazine*, 30(4):16–16.

Linden, G., Smith, B., and York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80.

Linden, M. and Hautzinger, M. (2008). *Verhaltenstherapiemanual*, volume 8. Springer.

Lindsay, P. H. and Norman, D. A. (1972). Human information processing: An introduction to psychology.

Linnemann, G. A. and Jucks, R. (2018). ‘can i trust the spoken dialogue system because it uses the same words as i do?’—influence of lexically aligned spoken dialogue systems on trustworthiness and user satisfaction. *Interacting with Computers*, 30(3):173–186.

Litman, D. and Forbes-Riley, K. (2014). Evaluating a spoken dialogue system that detects and adapts to user affective states. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 181–185.

Litman, D. and Silliman, S. (2004). Itspoke: An intelligent tutoring spoken dialogue system. In *Demonstration papers at HLT-NAACL 2004*, pages 5–8.

Litman, D. J. and Pan, S. (2002). Designing and evaluating an adaptive spoken dialogue system. *User Modeling and User-Adapted Interaction*, 12(2-3):111–137.

Liu, P., Glas, D. F., Kanda, T., and Ishiguro, H. (2018). Learning proactive behavior for interactive social robots. *Autonomous Robots*, 42(5):1067–1085.

López-Cózar, R., De la Torre, A., Segura, J. C., and Rubio, A. J. (2003). Assessment of dialogue systems by means of a new simulation technique. *Speech Communication*, 40(3):387–407.

Lopez-Tovar, H., Charalambous, A., and Dowell, J. (2015). Managing smartphone interruptions through adaptive modes and modulation of notifications. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pages 296–299.

Lou, P., Liu, Q., Zhou, Z., Wang, H., and Sun, S. X. (2012). Multi-agent-based proactive–reactive scheduling for a job shop. *The International Journal of Advanced Manufacturing Technology*, 59(1):311–324.

Lowe, R., Pow, N., Serban, I., and Pineau, J. (2015). The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*.

Lowe, R., Pow, N., Serban, I. V., Charlin, L., Liu, C.-W., and Pineau, J. (2017). Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue & Discourse*, 8(1):31–65.

Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. (2015). Recommender system application developments: a survey. *Decision Support Systems*, 74:12–32.

Luger, E. and Sellen, A. (2016). ” like having a really bad pa” the gulf between user expectation and experience of conversational agents. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 5286–5297.

Ma, Y., Nguyen, K. L., Xing, F. Z., and Cambria, E. (2020). A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70.

Madhavan, P. and Wiegmann, D. A. (2007). Similarities and differences between human–human and human–automation trust: an integrative review. *Theoretical Issues in Ergonomics Science*, 8(4):277–301.

Madsen, M. and Gregor, S. (2000). Measuring human-computer trust. In *11th australasian conference on information systems*, volume 53, pages 6–8. Citeseer.

Maheswaran, R. T., Tambe, M., Varakantham, P., and Myers, K. (2003). Adjustable autonomy challenges in personal assistant agents: A position paper. In *International Workshop on Computational Autonomy*, pages 187–194. Springer.

Mairesse, F. and Walker, M. (2006). Automatic recognition of personality in conversation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 85–88.

Mairesse, F. and Walker, M. A. (2010). Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction*, 20(3):227–278.

Malle, B. F. and Ullman, D. (2021). A multidimensional conception and measure of human-robot trust. In *Trust in Human-Robot Interaction*, pages 3–25. Elsevier.

Mandler, G. (1984). *Mind and body: Psychology of emotion and stress.* WW Norton & Company Incorporated.

Margolis, E., Samuels, R., and Stich, S. P. (2012). *The Oxford handbook of philosophy of cognitive science.* Oxford University Press.

Mayer, R. C., Davis, J. H., and Schoorman, F. D. (1995). An integrative model of organizational trust. *Academy of management review*, 20(3):709–734.

Mayer, R. E. and Moreno, R. (2002). Aids to computer-based multimedia learning. *Learning and instruction*, 12(1):107–119.

McAuley, E., Duncan, T., and Tammen, V. V. (1989). Psychometric properties of the intrinsic motivation inventory in a competitive sport setting: A confirmatory factor analysis. *Research quarterly for exercise and sport*, 60(1):48–58.

McCrae, R. R. and Costa Jr, P. T. (2008). The five-factor theory of personality.

McCrae, R. R. and John, O. P. (1992). An introduction to the five-factor model and its applications. *Journal of personality*, 60(2):175–215.

McDaniel, B., D'Mello, S., King, B., Chipman, P., Tapp, K., and Graesser, A. (2007). Facial features for affective state detection in learning environments. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 29.

McDuff, D., Kaliouby, R., Senechal, T., Amr, M., Cohn, J., and Picard, R. (2013). Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 881–888.

McFarlane, D. C. and Latorella, K. A. (2002). The scope and importance of human interruption in human-computer interaction design. *Human-Computer Interaction*, 17(1):1–61.

McTear, M. F. (2004). *Spoken dialogue technology: toward the conversational user interface*. Springer Science & Business Media.

McTear, M. F. (2020). Conversational ai: Dialogue systems, conversational agents, and chatbots. *Synthesis Lectures on Human Language Technologies*, 13(3):1–251.

Mehta, Y., Majumder, N., Gelbukh, A., and Cambria, E. (2019). Recent trends in deep learning based personality detection. *Artificial Intelligence Review*, pages 1–27.

Merritt, S. M., Heimbaugh, H., LaChapell, J., and Lee, D. (2013). I trust it, but I don't know why: Effects of implicit attitudes toward automation on trust in an automated system. *Human factors*, 55(3):520–534.

Merritt, S. M. and Ilgen, D. R. (2008). Not all trust is created equal: Dispositional and history-based trust in human-automation interactions. *Human factors*, 50(2):194–210.

Metze, F., Black, A., and Polzehl, T. (2011). A review of personality in voice-based man machine interaction. In *International Conference on Human-Computer Interaction*, pages 358–367. Springer.

Meurisch, C., Ionescu, M.-D., Schmidt, B., and Mühlhäuser, M. (2017). Reference model of next-generation digital personal assistant: integrating proactive behavior. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 149–152.

Meurisch, C., Mihale-Wilson, C. A., Hawlitschek, A., Giger, F., Müller, F., Hinz, O., and Mühlhäuser, M. (2020). Exploring user expectations of proactive ai systems. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(4):1–22.

Miehle, J., Ostler, D., Gerstenlauer, N., and Minker, W. (2017). The next step: intelligent digital assistance for clinical operating rooms. *Innovative surgical sciences*, 2(3):159–161.

Miehle, J., Wieluch, S., Minker, W., and Ultes, S. (2021). Decide or delegate: How script knowledge based conversational assistants should act in inconclusive situations. In *Adjunct Proceedings of the 2021 International Conference on Distributed Computing and Networking*, pages 69–73.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Mitra, S. and Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3):311–324.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Möller, S. (2004). A new itu-t recommendation on the evaluation of telephone-based spoken dialogue systems. Citeseer.

Mori, M., MacDorman, K. F., and Kageki, N. (2012). The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine*, 19(2):98–100.

Morrison, E. W. and Phelps, C. C. (1999). Taking charge at work: Extrarole efforts to initiate workplace change. *Academy of management Journal*, 42(4):403–419.

Morrissey, K. and Kirakowski, J. (2013). 'realness' in chatbots: establishing quantifiable criteria. In *International conference on human-computer interaction*, pages 87–96. Springer.

Muir, B. M. (1987). Trust between humans and machines, and the design of decision aids. *International journal of man-machine studies*, 27(5-6):527–539.

Muir, B. M. (1994). Trust in automation: Part i. theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics*, 37(11):1905–1922.

Muir, B. M. and Moray, N. (1996). Trust in automation. part ii. experimental studies of trust and human intervention in a process control simulation. *Ergonomics*, 39(3):429–460.

Nass, C., Jonsson, I.-M., Harris, H., Reaves, B., Endo, J., Brave, S., and Takayama, L. (2005). Improving automotive safety by pairing driver emotion and car voice emotion. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1973–1976.

Nass, C. and Lee, K. M. (2001). Does computer-synthesized speech manifest personality? experimental tests of recognition, similarity-attraction, and consistency-attraction. *Journal of experimental psychology: applied*, 7(3):171.

Nass, C. and Moon, Y. (2000). Machines and mindlessness: Social responses to computers. *Journal of social issues*, 56(1):81–103.

Nass, C., Steuer, J., and Tauber, E. R. (1994). Computers are social actors. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 72–78.

Nass, C. and Yen, C. (2012). *The man who lied to his laptop: What we can learn about ourselves from our machines.* Penguin.

Nielsen, M. A. (2015). *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA.

Nomura, T., Kanda, T., and Suzuki, T. (2006). Experimental investigation into influence of negative attitudes toward robots on human–robot interaction. *Ai & Society*, 20(2):138–150.

Norman, W. T. (1963). Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality ratings. *The journal of abnormal and social psychology*, 66(6):574.

Nothdurft, F., Behnke, G., Bercher, P., Biundo, S., and Minker, W. (2015a). The interplay of user-centered dialog systems and ai planning. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 344–353.

Nothdurft, F., Bertrand, G., Lang, H., and Minker, W. (2012). Adaptive explanation architecture for maintaining human-computer trust. In *2012 IEEE 36th Annual Computer Software and Applications Conference*, pages 176–184. IEEE.

Nothdurft, F., Heinroth, T., and Minker, W. (2013). The impact of explanation dialogues on human-computer trust. In *International Conference on Human-Computer Interaction*, pages 59–67. Springer.

Nothdurft, F., Richter, F., and Minker, W. (2014). Probabilistic human-computer trust handling. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 51–59.

Nothdurft, F., Ultes, S., and Minker, W. (2015b). Finding appropriate interaction strategies for proactive dialogue systems-an open quest. In *Proceedings of the 2nd European and the 5th Nordic Symposium on Multimodal Communication, August 6-8, 2014, Tartu, Estonia*, number 110, pages 73–80. Linköping University Electronic Press.

Nothdurft, F. M. (2015). *User-and Situation-adaptive Explanations in Dialogue Systems.* PhD thesis, Universität Ulm.

of Robotics, I. F. (2021). 31-million-robots-helping-in-households-worldwide-by-2019. `https://ifr.org/ifr-press-releases/news/31-million-robots-helping-in-households-worldwide-by-2019`. Accessed: 2021-08-27.

Oh, A. H. and Rudnicky, A. I. (2002). Stochastic natural language generation for spoken dialog systems. *Computer Speech & Language*, 16(3-4):387–407.

O'Leary, D. E. (2019). Google's duplex: Pretending to be human. *Intelligent Systems in Accounting, Finance and Management*, 26(1):46–53.

Ordóñez, F. J. and Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115.

Oviatt, S., Darves, C., and Coulston, R. (2004). Toward adaptive conversational interfaces: Modeling speech convergence with animated personas. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 11(3):300–328.

P.851, I.-T. R. (2003). Subjective quality evaluation of telephone services based on spoken dialogue systems. *International Telecomm. Union, Geneva.*

Paas, F., Van Gog, T., and Sweller, J. (2010). Cognitive load theory: New conceptualizations, specifications, and integrated research perspectives. *Educational psychology review*, 22(2):115–121.

Paas, F. G. (1992). Training strategies for attaining transfer of problem-solving skill in statistics: a cognitive-load approach. *Journal of educational psychology*, 84(4):429.

Pandey, A. K., Ali, M., and Alami, R. (2013). Towards a task-aware proactive sociable robot based on multi-state perspective-taking. *International Journal of Social Robotics*, 5(2):215–236.

Papangelis, A., Karkaletsis, V., and Makedon, F. (2012). Online complex action learning and user state estimation for adaptive dialogue systems. In *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, volume 1, pages 642–649. IEEE.

Parasuraman, R. and Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human factors*, 39(2):230–253.

Parasuraman, R., Sheridan, T. B., and Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 30(3):286–297.

Park, S. Y., Moore, D. J., and Sirkin, D. (2020). What a driver wants: User preferences in semi-autonomous vehicle decision-making. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Parker, S. K., Bindl, U. K., and Strauss, K. (2010). Making things happen: A model of proactive motivation. *Journal of management*, 36(4):827–856.

Parker, S. K., Williams, H. M., and Turner, N. (2006). Modeling the antecedents of proactive behavior at work. *Journal of applied psychology*, 91(3):636.

Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer.

Pecune, F. and Marsella, S. (2020). A framework to co-optimize task and social dialogue policies using reinforcement learning. In *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents*, pages 1–8.

Pelachaud, C. and Poggi, I. (2002). Subtleties of facial expressions in embodied agents. *The Journal of Visualization and Computer Animation*, 13(5):301–312.

Peng, Z., Kwon, Y., Lu, J., Wu, Z., and Ma, X. (2019). Design and evaluation of service robot's proactivity in decision-making support process. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 98. ACM.

Pham, T., Hayashi, K., Becker-Asano, C., Lacher, S., and Mizuuchi, I. (2017). Evaluating the usability and users' acceptance of a kitchen assistant robot in household environment. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 987–992. IEEE.

Picard, R. W. (2000). *Affective computing.* MIT press.

Pietquin, O. (2005). *A framework for unsupervised learning of dialogue strategies.* Presses univ. de Louvain.

Pietquin, O. and Hastie, H. (2013). A survey on metrics for the evaluation of user simulations. *The knowledge engineering review*, 28(1):59–73.

Pittermann, J. and Pittermann, A. (2006). Integrating emotion recognition into an adaptive spoken language dialogue system. In *2006 2nd IET International Conference on Intelligent Environments-IE 06*, volume 1, pages 197–202. IET.

Pittermann, J., Pittermann, A., and Minker, W. (2010). Emotion recognition and adaptation in spoken dialogue systems. *International Journal of Speech Technology*, 13(1):49–60.

Portela, M. and Granell-Canut, C. (2017). A new friend in our smartphone? observing interactions with chatbots in the search of emotional engagement. In *Proceedings of the XVIII International Conference on Human Computer Interaction*, pages 1–7.

Poushneh, A. (2021). Humanizing voice assistant: The impact of voice assistant personality on consumers' attitudes and behaviors. *Journal of Retailing and Consumer Services*, 58:102283.

Pragst, L., Ultes, S., Kraus, M., and Minker, W. (2015). Adaptive dialogue management in the kristina project for multicultural health care applications. *Proceedings of the 19thWorkshop on the Semantics and Pragmatics of Dialogue (SEMDIAL)*, pages 202–203.

Prasad, P. K. and Ertel, W. (2020). Knowledge acquisition and reasoning systems for service robots: A short review of the state of the art. In *2020 5th International Conference on Robotics and Automation Engineering (ICRAE)*, pages 36–45.

Prenger, R., Valle, R., and Catanzaro, B. (2019). Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE.

Qiu, S., Gadiraju, U., and Bozzon, A. (2020). Improving worker engagement through conversational microtask crowdsourcing. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. A., and Botvinick, M. (2018). Machine theory of mind. In *International conference on machine learning*, pages 4218–4227. PMLR.

Rach, N., Minker, W., and Ultes, S. (2017). Interaction quality estimation using long short-term memories. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 164–169.

Rach, N., Weber, K., Pragst, L., André, E., Minker, W., and Ultes, S. (2018). Eva: A multimodal argumentative dialogue system. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ICMI '18, page 551–552, New York, NY, USA. Association for Computing Machinery.

Rammstedt, B., Kemper, C., Klein, M. C., Beierlein, C., and Kovaleva, A. (2013). Eine kurze skala zur messung der fünf dimensionen der persönlichkeit: Big-five-inventory-10 (bfi-10). *Methoden, Daten, Analysen (mda)*, 7(2):233–249.

Rao, A. S., Georgeff, M. P., et al. (1995). Bdi agents: from theory to practice. In *ICMAS*, volume 95, pages 312–319.

Rau, P.-L. P., Li, Y., and Liu, J. (2013). Effects of a social robot's autonomy and group orientation on human decision-making. *Advances in Human-Computer Interaction*, 2013:11.

Reese, W. (2008). Nginx: The high-performance web server and reverse proxy. *Linux J.*, 2008(173).

Reiter, E. and Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press.

Rempel, J. K., Holmes, J. G., and Zanna, M. P. (1985). Trust in close relationships. *Journal of personality and social psychology*, 49(1):95.

Rheu, M., Shin, J. Y., Peng, W., and Huh-Yoo, J. (2021). Systematic review: Trust-building factors and implications for conversational agent design. *International Journal of Human–Computer Interaction*, 37(1):81–96.

Riegelsberger, J., Sasse, M. A., and McCarthy, J. D. (2003). Shiny happy people building trust? photos on e-commerce websites and consumer trust. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 121–128.

Rieser, V., Lemon, O., and Keizer, S. (2014). Natural language generation as incremental planning under uncertainty: Adaptive information presentation for statistical dialogue systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(5):979–994.

Ritter, A., Cherry, C., and Dolan, W. B. (2010). Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180.

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Shuster, K., Smith, E. M., et al. (2020). Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Rook, L., Sabic, A., and Zanker, M. (2020). Engagement in proactive recommendations. *Journal of Intelligent Information Systems*, 54(1):79–100.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Rosenfeld, A. and Kraus, S. (2016). Strategical argumentative agent for human persuasion. In *Proceedings of the Twenty-Second European Conference on Artificial Intelligence*, pages 320–328.

Rotter, J. B. (1980). Interpersonal trust, trustworthiness, and gullibility. *American psychologist*, 35(1):1.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Russell, J. A. (2003). Core affect and the psychological construction of emotion. *Psychological review*, 110(1):145.

Sadek, M. D., Bretier, P., and Panaget, F. (1997). Artimis: Natural dialogue meets rational agency. *IJCAI (2)*, 1030:1035.

Sanchez, J., Rogers, W. A., Fisk, A. D., and Rovira, E. (2014). Understanding reliance on automation: effects of error type, error distribution, age and experience. *Theoretical issues in ergonomics science*, 15(2):134–160.

Sanders, T., Oleson, K. E., Billings, D. R., Chen, J. Y., and Hancock, P. A. (2011). A model of human-robot trust: Theoretical model development. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 55, pages 1432–1436. SAGE Publications Sage CA: Los Angeles, CA.

Sankaran, S. and Markopoulos, P. (2021). " it's like a puppet master": User perceptions of personal autonomy when interacting with intelligent technologies. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, pages 108–118.

Sarikaya, R. (2017). The technology behind personal digital assistants: An overview of the system architecture and key components. *IEEE Signal Processing Magazine*, 34(1):67–81.

Sarikaya, R., Hinton, G. E., and Deoras, A. (2014). Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):778–784.

Scerbo, M. (1996). Theoretical perspectives on adaptive automation. automation and human performance: Theory and applications (a 98-12010 01-54).

Schaefer, K. E., Chen, J. Y., Szalma, J. L., and Hancock, P. A. (2016). A meta-analysis of factors influencing the development of trust in automation: Implications for understanding autonomy in future systems. *Human factors*, 58(3):377–400.

Schatzmann, J., Weilhammer, K., Stuttle, M., and Young, S. (2006). A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review*, 21(2):97–126.

Schiller, M., Behnke, G., Bercher, P., Kraus, M., Dorna, M., Richter, F., Biundo, S., Glimm, B., and Minker, W. (2018). Evaluating knowledge-based assistance for DIY. In Dachselt, R. and Weber, G., editors, *Proc. of MCI Works. "Digital Companion"*.

Schiller, M., Behnke, G., Schmautz, M., Bercher, P., Kraus, M., Dorna, M., Minker, W., Glimm, B., and Biundo, S. (2017a). A paradigm for coupling procedural and conceptual knowledge in companion systems. In *Proc. of the 2nd Int. Conf. on Companion Tech. (ICCT 2017)*.

Schiller, M. R., Schiller, F., and Glimm, B. (2017b). Testing the adequacy of automated explanations of el subsumptions. *Description Logics*, 1879.

Schillinger, P., Kohlbrecher, S., and von Stryk, O. (2016). Human-robot collaborative high-level control with application to rescue robotics. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2796–2802. IEEE.

Schlangen, D. (2006). From reaction to prediction: Experiments with computational models of turn-taking. *Proceedings of Interspeech 2006, Panel on Prosody of Dialogue Acts and Turn-Taking.*

Schlangen, D. and Skantze, G. (2011). A general, abstract model of incremental dialogue processing. *Dialogue & Discourse*, 2(1):83–111.

Schmidt, M. and Braunger, P. (2018). A survey on different means of personalized dialog output for an adaptive personal assistant. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*, pages 75–81. ACM.

Schmidt, M., Minker, W., and Werner, S. (2020). User acceptance of proactive voice assistant behavior. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2020*, pages 18–25.

Schmitt, A., Schatz, B., and Minker, W. (2011). Modeling and predicting quality in spoken human-computer interaction. In *Proceedings of the SIGDIAL 2011 Conference*, pages 173–184.

Schmitt, A. and Ultes, S. (2015). Interaction quality: assessing the quality of ongoing spoken dialog interaction by experts—and how it relates to user satisfaction. *Speech Communication*, 74:12–36.

Schneider, K. P. (1988). *Small talk: Analyzing phatic discourse*, volume 1. Hitzeroth.

Schuller, B., Batliner, A., Steidl, S., and Seppi, D. (2011). Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge. *Speech communication*, 53(9-10):1062–1087.

Schuller, B., Müller, R., Eyben, F., Gast, J., Hörnler, B., Wöllmer, M., Rigoll, G., Höthker, A., and Konosu, H. (2009). Being bored? recognising natural interest by extensive audiovisual integration for real-life application. *Image and Vision Computing*, 27(12):1760–1774.

Schuurmans, J. and Frasincar, F. (2019). Intent classification for dialogue utterances. *IEEE Intelligent Systems*, 35(1):82–88.

Scissors, L. E., Gill, A. J., Geraghty, K., and Gergle, D. (2009). In cmc we trust: The role of similarity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 527–536, New York, NY, USA. Association for Computing Machinery.

Sciuto, A., Saini, A., Forlizzi, J., and Hong, J. I. (2018). " hey alexa, what's up?" a mixed-methods studies of in-home conversational agent usage. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pages 857–868.

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press.

Segal, A., Gal, K., Kamar, E., Horvitz, E., and Miller, G. (2018). Optimizing interventions via offline policy evaluation: Studies in citizen science. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Seibert, S. E., Crant, J. M., and Kraimer, M. L. (1999). Proactive personality and career success. *Journal of applied psychology*, 84(3):416.

Seon, C.-N., Kim, H., and Seo, J. (2012). A statistical prediction model of speakers' intentions using multi-level features in a goal-oriented dialog system. *Pattern Recognition Letters*, 33(10):1397–1404.

Shah, C. (2018). Information fostering-being proactive with information seeking and retrieval: Perspective paper. In *Proceedings of the 2018 conference on human information interaction & retrieval*, pages 62–71.

Shah, P., Hakkani-Tür, D., Tür, G., Rastogi, A., Bapna, A., Nayak, N., and Heck, L. (2018). Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Sheridan, T. B. and Parasuraman, R. (2005). Human-automation interaction. *Reviews of human factors and ergonomics*, 1(1):89–129.

Sheridan, T. B. and Verplank, W. L. (1978). Human and computer control of undersea teleoperators. Technical report, Massachusetts Inst of Tech Cambridge Man-Machine Systems Lab.

Simpson, J. A. (2007). Psychological foundations of trust. *Current directions in psychological science*, 16(5):264–268.

Skantze, G. (2007). *Error handling in spoken dialogue systems-managing uncertainty, grounding and miscommunication*. Gabriel Skantze.

Skantze, G. (2021). Turn-taking in conversational systems and human-robot interaction: a review. *Computer Speech & Language*, 67:101178.

Sørmo, F. and Cassens, J. (2004). Explanation goals in case-based reasoning. In *Proceedings of the ECCBR*, pages 165–174.

Spearman, C. (1904). "general intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292.

Spitzer, R. L., Cohen, J., Fleiss, J. L., and Endicott, J. (1967). Quantification of agreement in psychiatric diagnosis: A new approach. *Archives of General Psychiatry*, 17(1):83–87.

Stachl, C., Au, Q., Schoedel, R., Gosling, S. D., Harari, G. M., Buschek, D., Völkel, S. T., Schuwerk, T., Oldemeier, M., Ullmann, T., Hussmann, H., Bischl, B., and Bühner, M. (2020). Predicting personality from patterns of behavior collected with smartphones. *Proceedings of the National Academy of Sciences*, 117(30):17680–17687.

Steinfeld, A., Bennett, S. R., Cunningham, K., Lahut, M., Quinones, P.-A., Wexler, D., Siewiorek, D., Hayes, J., Cohen, P., Fitzgerald, J., et al. (2007a). Evaluation of an integrated multi-task machine learning system with humans in the loop. In *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems*, pages 168–174.

Steinfeld, A., Quinones, P.-A., Zimmerman, J., Bennett, S. R., and Siewiorek, D. (2007b). Survey measures for evaluation of cognitive assistants. In *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems*, pages 175–179.

Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Science & Business Media.

Su, P.-H., Vandyke, D., Gasic, M., Kim, D., Mrksic, N., Wen, T.-H., and Young, S. (2015). Learning from real users: Rating dialogue success with neural networks for reinforcement learning in spoken dialogue systems. *arXiv preprint arXiv:1508.03386*.

Susi, T., Johannesson, M., and Backlund, P. (2007). Serious games: An overview.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2):257–285.

Sweller, J. (2010). Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational psychology review*, 22(2):123–138.

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103.

Tannenbaum, C., Ellis, R. P., Eyssel, F., Zou, J., and Schiebinger, L. (2019). Sex and gender analysis improves science and engineering. *Nature*, 575(7781):137–146.

Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge university press.

Times (2010). 50 worst inventions of all time. `http://content.time.com/time/specials/packages/article/0,28804,1991915_1991909_1991755,00.html`. Accessed: 2021-12-09.

Torre, I., Goslin, J., White, L., and Zanatto, D. (2018). Trust in artificial voices: A" congruency effect" of first impressions and behavioural experience. In *Proceedings of the Technology, Mind, and Society*, pages 1–6.

Traum, D. and Rickel, J. (2002). Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, pages 766–773.

Traum, D. R. and Hinkelman, E. A. (1992). Conversation acts in task-oriented spoken dialogue. *Computational intelligence*, 8(3):575–599.

Tseng, S. and Fogg, B. (1999). Credibility and computing technology. *Communications of the ACM*, 42(5):39–44.

Tupes, E. C. and Christal, R. E. (1961). Recurrent personality factors based on trait ratings. Technical report, Personnel Research Lab Lackland AFB TX.

Tur, G., Celikyilmaz, A., He, X., Hakkani-Tür, D., and Deng, L. (2018). Deep learning in conversational language understanding. In *Deep learning in natural language processing*, pages 23–48. Springer.

Tur, G. and De Mori, R. (2011). *Spoken language understanding: Systems for extracting semantic information from speech.* John Wiley & Sons.

Ultes, S. (2015). *User-centred adaptive spoken dialogue modelling.* PhD thesis, Universität Ulm.

Ultes, S. (2019). Improving interaction quality estimation with bilstms and the impact on dialogue policy learning. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 11–20.

Ultes, S., Budzianowski, P., Casanueva, I., Mrksic, N., Rojas-Barahona, L. M., Su, P.-H., Wen, T.-H., Gasic, M., and Young, S. J. (2017). Domain-independent user satisfaction reward estimation for dialogue policy learning. In *INTERSPEECH*, pages 1721–1725.

Ultes, S., Kraus, M., Schmitt, A., and Minker, W. (2015). Quality-adaptive spoken dialogue initiative selection and implications on reward modelling. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 374–383.

Ultes, S., Miehle, J., and Minker, W. (2019). On the applicability of a user satisfaction-based reward for dialogue policy learning. In *Advanced Social Interaction with Agents*, pages 211–217. Springer.

Ultes, S., Schmitt, A., and Minker, W. (2013). On quality ratings for spoken dialogue systems–experts vs. users. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 569–578.

Vaidyam, A. N., Wisniewski, H., Halamka, J. D., Kashavan, M. S., and Torous, J. B. (2019). Chatbots and conversational agents in mental health: a review of the psychiatric landscape. *The Canadian Journal of Psychiatry*, 64(7):456–464.

Vail, A. and Boyer, K. (2014). Adapting to personality over time: examining the effectiveness of dialogue policy progressions in task-oriented interaction. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 41–50.

Van Der Laan, J. D., Heino, A., and De Waard, D. (1997). A simple procedure for the assessment of acceptance of advanced transport telematics. *Transportation Research Part C: Emerging Technologies*, 5(1):1–10.

Vapnik, V. (2000). *The Nature of Statistical Learning Theory*, volume 8, pages 1–15.

Völkel, S. T., Schödel, R., Buschek, D., Stachl, C., Winterhalter, V., Bühner, M., and Hussmann, H. (2020). Developing a personality model for speech-based conversational agents using the psycholexical approach. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14.

Wagner, J., Lingenfelser, F., Baur, T., Damian, I., Kistler, F., and André, E. (2013). The social signal interpretation (ssi) framework: multimodal signal processing and recognition in real-time. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 831–834.

Wagner, N., Kraus, M., Rach, N., and Minker, W. (2021). How to address humans: System barge-in in multi-user hri. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction: 10th International Workshop on Spoken Dialogue Systems*, pages 147–152. Springer Singapore.

Walch, M., Sieber, T., Hock, P., Baumann, M., and Weber, M. (2016). Towards cooperative driving: involving the driver in an autonomous vehicle's decision making. In *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 261–268.

Walker, M. A., Whittaker, S. J., Stent, A., Maloor, P., Moore, J., Johnston, M., and Vasireddy, G. (2004). Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*, 28(5):811–840.

Wang, L., Rau, P.-L. P., Evers, V., Robinson, B. K., and Hinds, P. (2010). When in rome: the role of culture & context in adherence to robot recommendations. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 359–366. IEEE.

Watson, D., Clark, L. A., and Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: the panas scales. *Journal of personality and social psychology*, 54(6):1063.

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Weld, D. S. and Bansal, G. (2019). The challenge of crafting intelligible intelligence. *Communications of the ACM*, 62(6):70–79.

Wen, T.-H., Gasic, M., Mrksic, N., Su, P.-H., Vandyke, D., and Young, S. (2015). Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Wen, T.-H., Vandyke, D., Mrksic, N., Gasic, M., Rojas-Barahona, L. M., Su, P.-H., Ultes, S., and Young, S. (2016). A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.

Weston, J., Bordes, A., Chopra, S., Rush, A. M., van Merriënboer, B., Joulin, A., and Mikolov, T. (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Williams, J. D., Henderson, M., Raux, A., Thomson, B., Black, A., and Ramachandran, D. (2014). The dialog state tracking challenge series. *AI Magazine*, 35(4):121–124.

Williams, J. D., Kamal, E., Ashour, M., Amr, H., Miller, J., and Zweig, G. (2015). Fast and easy language understanding for dialog systems with microsoft language understanding intelligent service (luis). In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 159–161.

Williams, J. D. and Young, S. (2007). Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Wu, W., Guo, Z., Zhou, X., Wu, H., Zhang, X., Lian, R., and Wang, H. (2019). Proactive human-machine conversation with explicit conversation goals. *arXiv preprint arXiv:1906.05572*.

Yorke-Smith, N., Saadati, S., Myers, K. L., and Morley, D. N. (2012). The design of a proactive personal agent for task management. *International Journal on Artificial Intelligence Tools*, 21(01):1250004.

Yoshida, W., Dolan, R. J., and Friston, K. J. (2008). Game theory of mind. *PLoS computational biology*, 4(12):e1000254.

Yoshino, K. and Kawahara, T. (2015). News navigation system based on proactive dialogue strategy. In *Natural Language Dialog Systems and Intelligent Assistants*, pages 15–25. Springer.

Young, S., Gašić, M., Thomson, B., and Williams, J. D. (2013). Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Yu, D. and Deng, L. (2016). *Automatic Speech Recognition.* Springer.

Zamora, J. (2017). I'm sorry, dave, i'm afraid i can't do that: Chatbot perception and expectations. In *Proceedings of the 5th international conference on human agent interaction*, pages 253–260.

Zhang, Y., Narayanan, V., Chakraborti, T., and Kambhampati, S. (2015). A human factors analysis of proactive support in human-robot teaming. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3586–3593. IEEE.

Zheng, R. Z. (2018). *Cognitive Load Measurement and Application.* Routledge New York.

Zhou, J., Zhu, H., Kim, M., and Cummings, M. L. (2019). The impact of different levels of autonomy and training on operators' drone control strategies. *ACM Transactions on Human-Robot Interaction (THRI)*, 8(4):1–15.

Zonca, J., Folsø, A., and Sciutti, A. (2021). The role of reciprocity in human-robot social influence. *Iscience*, 24(12):103424.