Adaptive Cognitive Interaction Systems

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Felix Putze
aus Köln

Tag der mündlichen Prüfung:	31.10.2014
Erste Gutachterin:	Prof. Dr.-Ing. Tanja Schultz
Zweiter Gutachter:	Prof. Dr. rer.nat. Joachim Funke

Deutsche Zusammenfassung

Durch die wachsende Verbreitung von Bildschirmarbeitsplätzen, intelligenten Telefonen, und multimodalen Tablet-PCs wird die optimale Gestaltung der Mensch-Computer-Interaktion immer wichtiger. Im Laufe des letzten Jahrzehnts ergaben sich entscheidende Verbesserungen durch die Implementierung natürlicher Ein- und Ausgabemodalitäten, wie Gestensteuerung, automatische Spracherkennung oder Sprachsynthese. Durch diese natürlichen Schnittstellen steigen auch die Erwartungen der Benutzer an die Systeme. Die „Media Equation“ [RN96] belegt, dass Menschen Erfahrungen aus der Mensch-Mensch-Interaktion auf die Mensch-Computer-Interaktion übertragen. Menschen beobachten den affektiven und kognitiven Zustand ihres Interaktionspartners und passen ihr Interaktionsverhalten entsprechend an. Ein wesentliche Konsequenz der "Media Equation" ist, dass Benutzer solches adaptives Verhalten auch von ihren technischen Systemen erwarten. In dieser Arbeit beschäftigen wir uns damit, die dafür notwendigen Fähigkeiten für ein technisches System zu implementieren.

Ein adaptives, kognitives Interaktionssystem erfasst und modelliert den Zustand des Benutzers und reagiert auf diesen Zustand durch eine Anpassung des Interaktionsverhaltens. Ein adaptives, kognitives Interaktionssystem besteht aus drei Komponenten: Dem adaptiven Interaktionsmanager, dem empirischen kognitiven Modell und dem komputationalen Modell. Ein adaptiver Interaktionsmanager führt die Kommunikation mit dem Benutzer durch und greift dazu auf die Informationen über dessen Zustand zurück. An diesen Zustand passt der Interaktionsmanager das Interaktionsverhalten an. Um die Informationen über den Benutzerzustand zu erhalten, greift der Interaktionsmanager auf zwei verschiedene Arten von kognitiven Modellen zurück. Das empirische kognitive Modell zeichnet Sensordaten des Benutzers auf und verwendet dazu Methoden des maschinellen Lernens, um aus den Daten Benutzerzustände zu erkennen. Ein komputationales kognitives Modell repräsentiert komplexe kognitive Prozesse und prädiziert das aus diesen resultierende Verhalten. Diese beiden Ansätze zur kognitiven Modellierung können parallel eingesetzt werden, um sich gegenseitig zu ergänzen (da sie unterschiedliche Aspekte desselben Zustands abbilden) und einander zu beeinflussen (um den Einfluss eines Zustands auf einen anderen zu modellieren). In dieser Arbeit leisten wir substantielle Beiträge zum Fortschritt aller drei Komponenten. Weiterhin zeigen wir, wie die drei Komponenten gemeinsam in Ende-zu-Ende Interaktionssystemen eingesetzt werden können, um eine signifikante objektive und subjektive Verbesserung der Mensch-Computer-Interaktion zu erzielen.

In Bereich der empirischen kognitiven Modellierung konzentrieren wir uns auf die Zustände „mentale Auslastung“ und „Verwirrung“. Beide Zustände spielen in der Mensch-Computer-Interaktion eine große Rolle, da sie – wie im Falle mentaler Auslastung – direkten Einfluss auf das Verhalten und die kognitive Leistungsfähigkeit des Benutzers haben oder – wie im Falle der Verwirrung – Aufschluss über den Verlauf der Interaktion liefern. Um diese beiden Zustände zu erkennen, beschreiben wir die Entwicklung und Evaluierung eines personenunabhängigen Modells mentaler Auslastung, basierend auf dem Elektroenzephalogramm (EEG) zur Gehirnaktivitätsmessung und anderen physiologischen Signalen. Dieses System wird auf einem außergewöhnlich großen Datensatz ausgewertet. Dieser enthält Daten von über 150 Versuchspersonen, die verschiedene kognitive Aufgaben bearbeiten. Weiterhin entwickeln wir das erste empirische kognitive Modell, dass verschiedene Modalitäten kombiniert, um die Verwendung verschiedener Wahrnehmungsmodalitäten zu erkennen. Außerdem stellen wir ein personen-adaptives Modell zur EEG-basierten Erkennung des Benutzerzustands „Verwirrung“ vor. Diese personen-unabhängigen und personen-adaptiven empirischen kognitiven Modelle sind besonders für den Einsatz im Kontext der Mensch-Computer-Interaktion geeignet. Sie liefern die erkannten Benutzerzustände an die anderen Komponenten weiter, die Modelle und Interaktionsverhalten an die geschätzen Zustände anpassen.

Im Bereich der komputationalen kognitiven Modellierung entwickeln wir ein modulares Gedächtnismodell zur Repräsentation von dynamischen Assoziationsprozessen im Gedächtnis für die Verwendung in Interaktionssystemen. Wir zeigen mit zwei verschiedenen Ansätzen, wie ein solches komputationales Modell an verschiedene Niveaus mentaler Auslastung angepasst werden kann. Das Niveau mentaler Auslastung kann über ein entsprechendes empirisches kognitives Modell erfasst werden. Dadurch steigt die Vorhersagegenauigkeit bei wechselnder Auslastung gegenüber einem nicht-adaptiven Modell signifikant. Am Beispiel einer mehrschrittigen, assoziativen Lernaufgabe zeigen weiterhin, dass komputationale und empirische Modellierung kombiniert werden können, um Benutzerzustände zu erfassen, die keines der Modellierungsparadigmen für sich allein erkennen kann.

Außerdem entwickeln wir den leichtgewichtigen, adaptiven Interaktionsmanager AIM. Mit dessen Hilfe implementieren wir mehrere adaptive, kognitive Interaktionssysteme für verschiedene Einsatzzwecke. AIM erhält die Schätzungen der kognitiven Modelle über den Benutzerzustand, um sein Interaktionsverhalten an diese Zustände anzupassen. Die Interaktionssysteme werden in Benutzerstudien evaluiert, um messbare objektive und subjektive Usability-Verbesserungen adaptiver kognitiver Interaktionssysteme (gegenüber nicht-adaptiven Systemen) zu dokumentieren. Dazu gehört die Auswertung eines Ende-zu-Ende-Systems, bei dem wir zeigen, dass eine adaptive Strategie zur Informations-Präsentation signifikante Verbesserungen gegenüber einer nicht-adaptiven Strategie erzielen kann, sowohl bezüglich objektiver als auch subjektiver Qualitätsmetriken. Wir untersuchen weiterhin, wie sich verschiedene Interaktionsstrategien auf die subjektiv empfundene Intrusivität auswirken. Eine selbst-korrigierende gestenbasierte Benutzerschnittstelle reagiert auf Verwirrung des Benutzers nach Fehlinterpretationen von Benutzereingaben. Eine Benutzerstudie vergleicht verschiedene Korrekturstrategien hinsichtlich Genauigkeit und Korrekturkosten. Zuletzt beschreiben wir die Entwicklung einer kognitiven Benutzersimulation für sprachbasierte Interaktionssysteme, die statistische Methoden mit komputationalen kognitiven Modellen verbindet.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

This chapter serves as a motivation for the present dissertation and as an introduction to the relevant aspects of adaptive cognitive interaction systems. We motivate the need for adaptive cognitive interaction systems and present our proposed architecture, consisting of three main components: Empirical cognitive model, computational cognitive model and interaction manager. Finally, the structure and contributions of this thesis are presented.

1.1 Motivation

As computers and complex technical devices become more present in our daily life and work environment, we are constantly facing Human-Computer Interaction (HCI) situations. This development mandates that HCI is as robust, efficient and satisfactory as possible. Since the 1980s, researchers in HCI systematically strive to design user interfaces which fulfill those requirements. Traditional user interfaces which use a mouse or keyboard as input device are established, but those devices lead to artificial and inefficient means of communication. Interfaces only based on those devices are oblique to the large number of signals emitted by the user which voluntarily or involuntarily send a lot of additional information about the user to the computer.

Since the advent of mobile smartphones, camera-equipped entertainment consoles and wearable computing devices, user interfaces based on gesture and speech modalities are used on a regular basis. Such interfaces allow their users to employ intuitive and efficient means of communication. Gestures and speech can transmit information with a very high transfer rate and do not need any training for using them efficiently. As novel input techniques become more common, people get used to interact with seemingly intelligent devices which use the same modalities humans are using when communicating with other persons. One major side effect of this development is that users unconsciously and inevitably develop the expectation that HCI follows the same explicit and implicit social rules and principles as there are established for human-human interaction. Byron Reeves and Clifford Nass describe this phenomenon as the *Media Equation* and validated its claim in numerous studies [RN96]. However, current interfaces are not yet prepared to fulfill those expectations, because besides input modalities, other aspects of this “human-like” HCI are still not present.

A major gap between user expectations and the State-of-the-Art of HCI is the fact that most systems are completely oblique to the situation or state the user is in. For example, a user who interacts with a system while performing a secondary task (e.g. talking to another person, driving, etc.) will show completely different behavior than a user who completely focuses on the interaction: Splitting cognitive resources between two tasks may lead to missed information or a reduced memory span. A considerate human communication partner will pay attention to cues which signal the other person’s inner state and react appropriately by adjusting his or her interaction behavior. In the example of high workload caused by dual tasking, a considerate partner will avoid or delay non-critical communication to prevent information overload. This process of noting the state of the interaction partner and adapting interaction behavior accordingly is called empathy [Ick93].

It is the goal of this thesis to enable technical systems to replicate this behavior to the best extend possible. To develop empathy-like capabilities for technical systems requires three components: The system needs to observe the user to detect his or her user states. The system needs to model those user states and predict their impact on the interaction. Finally, the system needs to adapt its interaction behavior to the detected user states. The expected benefit of this approach is to make HCI more robust, more efficient and more satisfactory compared to non-adaptive systems.

In the remainder of this chapter, we define the term user state, describe the components of an adaptive cognitive interaction system in detail, outline the contributions of this thesis to the field and present the structure of the thesis.

1.2 User States

In this section, we motivate the need of adaptive system behavior by the Media Equation. We define the term “user state” and distinguish it from the term “user trait”. Then, we give examples of user states and discuss their relevance in HCI applications. This will help us to chose which user states to concentrate on in this thesis.

1.2.1 Media Equation & Adaptive Behavior

According to the “Media Equation” [RN96], humans tend to react to and interact with complex machines in ways that are similar to behavior shown in the interaction with other humans. This phenomenon has been investigated in a large number of user studies and was shown to be a general behavioral pattern that is hard to suppress. For example, [NJH^{+}05] presented a user study of a “virtual passenger” in a car. Experimental results indicated that both subjective and objective criteria (such as driving quality) improved when the interaction system adapted its voice characteristics to the driver’s emotional state. The study demonstrated that a single fixed emotional stance of the virtual passenger is suboptimal. Instead, the system has to continuously follow and match the changing emotional state of the user. [NB05] convincingly showed that the Media Equation is especially valid for systems that use input or output modalities which imitate natural means of communication: Humans are “wired for speech” as speech is the primary way of natural communication. Therefore, speech-based systems (also called “voice-enabled”) are likely to trigger strong effects as predicted by the Media Equation. For example, [NL01] investigated whether similarity attraction of a human towards another human with similar personality (an effect which is long known in psychology) transfers to a human who is interacting with a technical system. The authors manipulated the personality of a voice-enabled system by adjusting specific speech attributes. Their experiment showed that users preferred the system which matched their own personality. The Media Equation also relates to workload induced by multi-tasking during HCI: [MJ05] and [Cha09] showed that the effect of an inconsiderate human passenger is much more detrimental to driving performance than a conversation with a considerate passenger. The latter flexibly reacts to traffic situations which increase workload, for example by pausing the conversation or alerting the driver. The impact of operation of a mobile phone on driving performance is comparable to the effect of an inconsiderate passenger. The Media Equation therefore suggests that users expect their systems to behave like a considerate passenger to avoid this negative effect.

1.2.2 User States vs. User Traits

The Media Equation mandates that an interaction system must flexibly react to user states if it aims at offering an intuitive and efficient user experience. We define a *user state* as a dynamic attribute of the user that may change over the course of an interaction. A user state refers to the cognitive or affective condition of the user which influences the user's behavior and performance during the interaction. Examples for user states are the mental workload level, the emotional state, fatigue, etc. In contrast to a user state, a *user trait* is a characteristic of the user that we consider to be stable during the lifetime of the interaction system. Examples for user traits are gender, age [WBG05, MEB$^+$09] or personality [BLMP12]. While it is clear that customization for user traits is useful when building systems which are optimized for a specific user (for example, Nass' example of personality-based similarity attraction [NL01]), in this thesis, we will concentrate on systems which adapt to user states: Adaptation to dynamically changing user states is more challenging for HCI as systems can be customized for each stable user trait once and than load that profile when the user is identified. Additionally, we can treat adaptation to persistent user traits as special case of a one-time adaptation to a potentially changing user state.

Of course, there is a vast number of potential user states which determine a user's behavior and performance in a cognitive task. Building a comprehensive model which represents all potential states is currently beyond the scope of existing architectures as this would imply developing a model of human-like complexity. We therefore look for a small selection of user states for investigating the concept of adaptive cognitive interaction systems in this thesis. For this purpose of selecting user states, we define a list of five criteria:

- The user state occurs frequently in typical HCI scenarios
- The user state strongly influences user behavior or the outcome of the interaction

- There is potential for an adaptive system to react to the detected user state
- It is feasible to systematically collect data containing different manifestations of the user state
- The choice of user state creates new research opportunities

1.2.3 User State: Emotion

One of the most actively researched user states is *emotion*. Emotions modulate cognitive processes like memory [Chr92] or problem solving strategies [SWF05] and therefore impact HCI. By coining the term “Affective Computing” [Pic00], Rosalind Picard started the research on computers which were able to detect the emotional state of their users and react appropriately, potentially by synthesizing emotion themselves. Affective computing applications like tutoring systems or entertainment-centered systems are very successful and demonstrate the feasibility of this approach. However, full-blown emotions are often rare in real-world HCI scenarios and often unrelated to the interaction task itself. The only regular exception to this is anger, induced by undesired system behavior [BvBE$^+$09]. However, when the emotional state is unrelated to the interaction, the system is limited in its possibilities to adapt, besides mirroring the emotion or showing sympathy. Additionally, emotion elicitation for data collections is a challenging task [CB07] and ground truth is unreliable [DCDM$^+$05, Cow09, MML$^+$09]. Those limitations reduce the applicability of the affective computing approach to few HCI scenarios. In the relevant domains (i.e. tutoring, entertainment), affective computing has already been extensively explored, while other promising user states did not receive the same attention. For those reasons, we abstain from taking emotion into account as user state and focus on others which we deem to be more promising as a target user state to make an impact on HCI in general.

1.2.4 User State: Workload

Another important user state is (mental) *workload*. [HS88] defines workload as “the perceived relationship between the amount of mental processing capability or resources and the amount required by the task”. A user who interacts with a system while operating a secondary task (e.g. talking to another person, driving, etc.) will show completely different behavior than a user who can fully focus on the interaction task. This is because increased workload may result in compensatory behavior [RH97] and cognitive performance degradation [HD01]. Ignorance of a user's workload level by the system may cause information overload and low user satisfaction. Due to the increasing presence of mobile applications (for example on smartphones), variable workload levels become omnipresent in HCI. This means that variable workload levels and resulting differences in cognitive performance are a property of most current HCI scenarios. One important example where such multitasking situations are omnipresent is the operation of interactive systems in the car. Navigation systems, entertainment systems, smartphones and other devices have become a quasi-standard of most current cars. Distraction while driving is one of the major causes of accidents on the road [LBCK04]. An adaptive cognitive interaction system which is able to detect states of high workload and adapt the interaction appropriately (e.g. by delaying non-critical information, changing its style of information presentation, etc.) would increase both safety and usability in such scenarios.

In experiments, different levels of workload can be systematically induced by alternating between single-tasking or multitasking or by varying task difficulty. While existing research has shown the general feasibility of detecting the workload level of a human from sensor signals, little is known about 1) the feasibility of workload recognition in more realistic HCI scenarios and 2) the possibilities of an interaction system to adapt to changes in workload level.

For those reasons, we chose to concentrate on workload as one of the central user states in this thesis.

1.2.5 User State: Confusion

Another user state we are looking at in this thesis is the state of *confusion.* We define confusion as the reaction to erroneous behavior exhibited by the system which was not expected by the user. For example this can occur, if the user gives input to the system to select an item from a menu using a potentially error-prone input modality like speech or gestures. Even for well-trained statistical gesture and conversational speech recognizers, error rates still are in the double-digits [SLY11]; therefore, erroneous and unexpected feedback to user input are common for such recognizers. As the input modalities speech and gestures find their way into many new technical systems, such recognition errors and the resulting user state of confusion will occur frequently in many HCI applications. For conducting experiments on confusion detection, this user state can be provoked by systematically introducing erroneous system feedback in an HCI task.

The user state of confusion is highly relevant for the interaction as undesired behavior of the system forces the user to enter an error correction sub-interaction [SNG$^+$02] which is time-consuming and distracts from the original task. Automatic confusion detection would enable systems to proactively recover from recognition input errors, for example by reprompting the user. This would reduce the effort for error detection and error recovery for the user. Because of the frequency of recognition errors and the potential benefits for detecting the resulting user state of confusion (benefits which are currently untapped in the research community), we include this user state as one of the regarded states in this thesis.

1.2.6 User State: Memory

Other relevant user states are more complex and cannot be defined by the absence of presence of a certain condition. One example for such a user state is the configuration of a user's memory. Memory determines which information the user can readily access, which information was already forgotten (even if already given earlier by the system) and which information is currently relevant to the user. As most tasks in HCI comprise a memory component (e.g. to remember the interaction discourse, or to give appropriate input to the system), modeling the state of the user's memory and the capabilities and limitations of memory retrieval are important to accurately predict user behavior. Psychologically sound models which predict behavior and performance of human memory exist, but have not been researched thoroughly in the HCI context. Accurate representation of human memory is especially important when we cannot assume a perfect memory, as most current interaction systems implicitly do: Cognitive performance (and memory performance in particular) of a person is variable and depends on the person's workload level. This interplay indicates that user states need to be modeled in context of each other, not in isolation. For this reason, we investigate a memory model that is able to predict memory performance for different workload levels.

1.3 Adaptive Cognitive Interaction Systems

In this section, we define the term adaptive cognitive interaction system an introduce its three main components. An *adaptive cognitive interaction system* is an interaction system (i.e. a technical system which receives input by the user and generates output for the user in real-time) that models user states, detects user states automatically and reacts to them appropriately. An adaptive cognitive interaction system consists of three main components, see Figure 1.1: An empirical cognitive model, a computational cognitive model and an adaptive interaction manager. These components are intertwined and share information about the user.

The first component is an *empirical cognitive model* which observes the user to detect user states. In a bottom-up fashion, an empirical cognitive model uses sensors (cameras, microphones, or physiological sensors) to collect data from the user. The model processes this data to extract meaningful features, abd then applies statistical machine learning methods to automatically classify the data regarding the manifestation of different user states.

Those empirical bottom-up models are complemented by top-down *computational cognitive models* which represent the state of the user's cognitive processes to predict user behavior and performance. This is necessary because not all user states can be inferred from noisy sensor data. A computational cognitive model formalizes psychological knowledge about human cognition in a form that allows to make predictions about human cognition, e.g. on the user's behavior and performance in a certain task.

Both types of models, empirical and computational cognitive models, exchange information to adjust their respective predictions. This exchange can be used to model a user state with information from both knowledge sources or to represent the influence of one user state on the other.

The information from both the empirical and computational cognitive model are sent to the *interaction manager*. In general, the interaction manager engages the user in interaction and contains all information to generate meaningful responses and queries. In an adaptive cognitive interaction system, this component also uses information on the user state to adapt to the user by modifying its behavior.

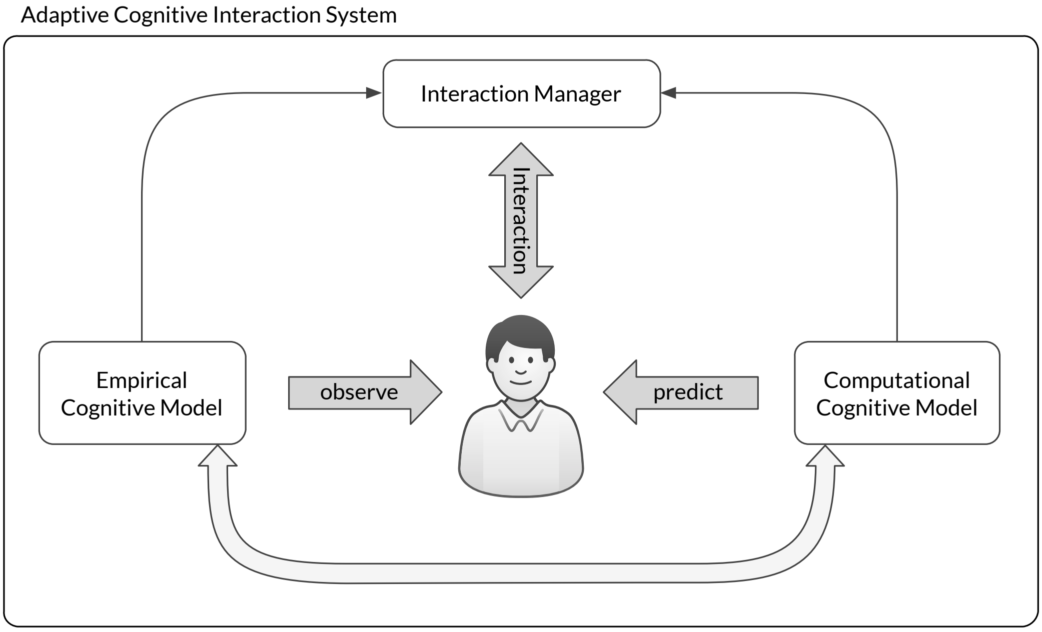

Adaptive Cognitive Interaction System
Interaction Manager
Interaction
Empirical Cognitive Model
observe
predict
Computational Cognitive Model

Figure 1.1 – General architecture of an adaptive cognitive interaction system.

1.4 Contributions

In this section, we will give an overview over the scientific contributions to the field of adaptive cognitive interaction systems we present in this thesis. In this thesis, we contribute several novel findings and methods to all three central components of an end-to-end adaptive cognitive interaction system.

Considering the empirical cognitive modeling, we contribute the development and evaluation of models for different user states. We concentrate on systems which address challenges occurring in realistic HCI scenarios: the requirement to deal with artifacts caused by user movement and the environment; the need to reduce setup time of the model for new users by exploiting existing data from other people; and the feasibility of transferring an empirical cognitive model between different scenarios. We show the versatility of empirical cognitive modeling by regarding three different user states: We present systems based on Electroencephalography (EEG) and other physiological signals to recognize a person's workload level and to detect a state of confusion. We describe the implementation of a multimodal **person-independent workload recognition system**, which is evaluated on an exceptionally large – in the context of EEG data collections for HCI

– data corpus with more than 150 subjects performing multiple tasks. Our investigation on modality recognition demonstrates the **first hybrid passive Brain Computer Interface** using EEG and functional Near Infrared Spectroscopy (fNIRS). To demonstrate the versatility of empirical cognitive modeling, we contribute an EEG-based **person-adaptive system for confusion detection** for which we also evaluate the potential for task transfer.

Considering computational cognitive models, we contribute the development and evaluation of such models especially for the HCI context. For this goal, we addressed the need for real-time model tracing (i.e. the ability to dynamically model the changing cognitive state of a human during the execution of a task); the accommodation of multiple workload levels by computational cognitive models to improve the prediction of performance under different user states; the combination of empirical and computational cognitive modeling. More concretely, our individual contributions are: The development and evaluation of a generic **memory modeling component for interaction systems**, which model dynamic memory associations on large-scale databases. Furthermore, we contribute – to our knowledge for the first time – the description, analysis and comparison of two approaches to **model the impact of workload level on performance** in cognitive tasks. We also show findings on how the **combination of empirical cognitive modeling and computational cognitive modeling** yields information which is not accessible to a system which only resorts to one of those two complimentary approaches.

In the field of adaptive systems, our contributions focused on the goal of exploiting the user state predictions of the cognitive to achieve a measurable benefit for the user of an interaction system. One of our main contributions is that we do not only look at the detection and modeling of user states (as the majority of publications in the research community does), but that we develop multiple end-to-end interaction systems in a common framework and present extensive evaluations of both objective and subjective performance measures. For this purpose, we contribute the implementation of the light-weight **adaptive interaction manager AIM** which focuses on flexible adaptation mechanisms. We use the AIM in three user studies to demonstrate the benefits and challenges of adaptive systems. Those studies contribute **findings on both objective and subjective usability effects** of adaptive user interfaces and compare different adaptation strategies. Our final contribution in this field is the development of a **cognitive user simulation** which combines the benefits of statistical user simulation techniques with computational cognitive models to generate plausible user and system utterances.

1.5 Structure of this Thesis

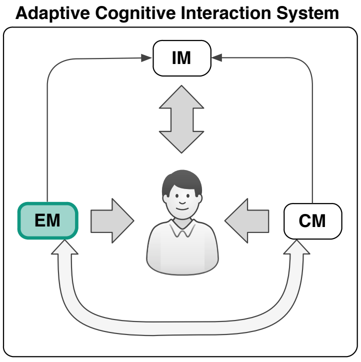
This thesis is composed of three main chapters. These chapters describe in-depth the three main components of an adaptive cognitive interaction system. Each chapter provides related work, including the necessary methodological fundamentals, and a detailed description of the corresponding contributions. Chapter 2 deals with empirical cognitive models to recognize workload level, workload type and the state of confusion from physiological signals. Chapter 3 deals with computational cognitive models and how they can be designed for the application in interaction systems. Chapter 4 focuses on the development of a framework for adaptive interaction management and its application in several evaluation studies. The chapter describes three systems which investigate different aspects of workload- and confusion-adaptive interaction systems. Chapter 5 concludes the thesis by summarizing the main results and proposes steps for future work.

Chapter 2

Empirical Cognitive Modeling

In this chapter, we introduce the concept of empirical cognitive modeling to detect user states from sensor data. We start by discussing the related work in this field. We then present three examples of empirical cognitive models to detect the user states workload level, workload type and confusion. We describe the employed methods and present thorough evaluations for all three examples.

2.1 Introduction



Empirical cognitive models are bottom-up models which recognize the manifestation of a certain user state (e.g. workload level) or its absence or presence (e.g. is the user confused?). Empirical cognitive models are usually implemented as statistical classifiers working on features derived from sensor data. Most of the systems presented in this chapter use signals which capture information on brain activity, which is the most direct source of information on the user's cognitive processes. Systems which use such signals to classify certain cognitive user states or imagined user commands are usually called Brain-Computer Interfaces (BCIs). Throughout this chapter, we often use the terms (statistical) classifier and BCI to refer to implementations of empirical cognitive models because these terms are frequently used in the research community.

To explore different types of user states and associated challenges, we present empirical cognitive models to detect three different user states. In Section 1.2.2, we already motivated why we concentrate on the user states workload and confusion. In this chapter, section 2.3 presents a system which uses EEG and other physiological signals to discriminate different levels of mental workload. We present a person-independent evaluation on a large data corpus with more than 150 participants. Section 2.4 describes the development of a hybrid empirical cognitive model which combines EEG and fNIRS signals to detect and discriminate different perceptual modalities. Finally, Section 2.5 describes a person-adapted model which uses EEG for the detection of the user state confusion.

2.2 Related Work

In this section, we discuss related work on systems which classify user states based on signals recorded from the user. We start with an introduction of the methodological fundamentals of this field, ranging from signal type selection to the development of the models themselves. Afterwards, we review the State-of-the-Art of the detection of two user states, workload and confusion. We discuss the strengths and weaknesses of the existing literature and describe our contributions to this field.

2.2.1 Signal Types for Empirical Cognitive Modeling

To build an empirical cognitive model, we first need to specify the signal types which are used to supply the model with data. Selection of signal types influences the accuracy of the model as well as the suitability of the system for specific applications. User states as defined in Section 1.2.2 may be extracted from many different signal types. Here, we start by categorizing systems for user state detection by the employed signal types into three groups, following [CD10]: audio-based, video-based and physiology-based systems. For each category of signal types, we discuss its advantages and drawbacks.

Audio-based Systems

Audio-based systems are capable of extracting emotions [ZTL$^+$07], fatigue [GFW$^+$06], workload [KCM10] or other user states from speech. Since 2009, the Interspeech conference hosts yearly contests on the detection of user states and traits from speech signals [SSB09, SSB$^+$10, SSB$^+$11, SSB$^+$12, SSB$^+$13]. Speech based systems can exploit information on linguistic content by extracting counts of words and word sequences etc. and also on paralinguistic cues, using features like autocorrelation, pitch, jitter, shimmer, and frequency attributes. Sensors for recording audio data (i.e. microphones) are cheap, small and non-intrusive. There exists a number of studies which demonstrate the feasibility of extracting various user states from audio signals in natural use cases [ZPRH09].

The major limitation of audio-based empirical cognitive models is their restriction to situations where a user is speaking. On the one hand, if the user is operating an interface which is not voice-enabled, such situations are rare. On the other hand, if the user is operating a voice-enabled interface, the user's speech will be a response to system prompts. However, adaptation to user states which the system is only able to detect when the interaction is already ongoing may be too late in many cases. In other situations, no usable speech of the user will be available because environmental noise suppresses the speaker's voice in the audio stream.

Video-based Systems

Video-based systems are another promising venue of research on user state detection. Such models are able to extract information on the user state from fine-scale recognition of mimics or large-scale recognition of body posture and gestures (or both). For example, systems to measure vigilance from high-resolution, high-frequency video recordings of the eye have a long tradition [JY02] and are already deployed as commercial car accessories. From facial video recordings, facial action units (FAUs) can be extracted [GE11]. Each FAU encodes contraction or relaxation of one or more muscles in the face. FAUs can be used for emotion recognition [VKA$^+$11], or for the detection frustration and boredom [CDWG08]. [SCL$^+$11] used video-based analysis of posture and body movement to predict engagement of children involved in human-robot interaction. Video recordings can also act as contact-free physiological sensors: The Cardiocam [PMP10] is able to estimate the heart rate of a person at a distance, using an off-the-shelf webcam by applying blind source separation techniques to the color channels of the camera. While those examples have shown that video based approaches for user state recognition are remarkably successful, they are limited in practice by camera positioning and the dependency on lighting conditions. For those reasons, video data is hard to acquire in many mobile scenarios. Video analysis is also limited to user states that generate reliable and visible reactions in facial expression, gestures or body language.

Physiology-based Systems

The third category of signal types for empirical cognitive modeling covers signals which are measured by physiological sensors. This broad definition summarizes a large group of measurements of specific bodily functions correlated to user states. Examples of such measurements are:

- **Blood volume pressure** (BVP). Can be measured by a photoplethysmograph which is placed at a finger or at the mastoid of the participant. Typical features which can be extracted from BVP are heart rate and heart rate variability.
- **Electrodermal activity** (EDA). EDA measures electrical conductance of a person's skin. Conductance varies with moisture of the skin, which is influenced by sweat. EDA is measured with an ohmmeter and the application of a small current between two points on the surface of the skin (other ways of measurement are possible). A typical sensor placement is on the palm of the hand, where many sweat glands are present.
- **Respiration** (RESP). Respiratory activity can be recorded with (pressure based or electro-magnetism based) respiration belts around the chest of a person. From those signals, features like frequency and depth of breathing can be extracted.

Sensors to record such signals are very cheap and mobile due to their small size. There exist devices which incorporate such sensors in items of daily use and make it possible to record physiological data in an non-intrusive way. Examples include sensors integrated in clothing [SSO11], a computer mouse [ADL99] or a steering wheel [CL07] in the car. However, the sensitivity and specificity we can expect from those signals is limited. While those signals correlate with user states such as stress, they are also influenced by physical activity, pressure level, temperature and many other factors unrelated to the user's state. This means that the validity of the relation between the captured physiological parameter and user state is often weak [Fai09]. In Chapter 2.3, we will see that we can reliably use features derived from physiological signals for some classification tasks regarding the user state. However, we will see that not all user states can be detected reliably using such signals. As a consequence, we will mostly concentrate on sensors which record brain activity (or correlates thereof) in this thesis. An advantage of sensors of brain activity is that their signals are more directly related to the cognitive processes while parameters like BVP, EDA or respiration are only indirectly moderated by the peripheral nervous system. Sensors for brain activity usually provide multi-channel recordings which allow for artifact correction of individual channels and spatial localization of activity in the brain.

Human-computer-interfaces which operate on signals correlated to brain activity are often called Brain-Computer Interfaces (BCIs) in the literature. Traditional BCIs target handicapped users (e.g. locked-in patients) which are enabled by a BCI to control a user interface with simple selection commands [WBM$^+$02] or letter spelling [KSM$^+$08]. However, in the last decade, the term *passive BCI* was coined [ZK11]. Passive BCIs are targeted to a more general audience and are designed to detect and react to certain user states. By far the most common sensor technology for BCIs is Electroencephalography (EEG). EEG directly measures electrical activity of neurons in the brain – namely, excitatory postsynaptic potentials – at the surface of the scalp. Signals are not captured from single neurons but from large clusters of synchronously firing pyramid cells in the cortex of the human brain. EEG is traditionally measured using an electrode cap with 16 to up to 256 electrodes. To place electrodes reproducibly on relevant regions of the scalp, we follow the 10/20 standard or its extension, the 10/10 standard, for a higher number of electrodes. See Figure 2.1) which defines electrode positions relative to the anatomy of the skull [Jas58].

EEG technology is affordable, mobile and offers a high temporal resolution. A drawback of EEG is the low spatial resolution as it measures accumulated activity of large areas of the cortex and of deeper brain regions. Another challenge is that EEG signals are characterized by a small amplitude in the range of microvolts which makes them susceptible to a large number of artifacts. Artifact sources are technical, mechanical, or physiological in nature. In some scenarios, those artifacts may actually contain useful information to classify the current user state (e.g. to recognize certain facial expressions from muscular artifacts in the EEG signal [HPS11]). In most cases however, handling artifacts is critical to recover the actual EEG signal for classification. BCI literature has proposed a number of methods to deal with such arti-

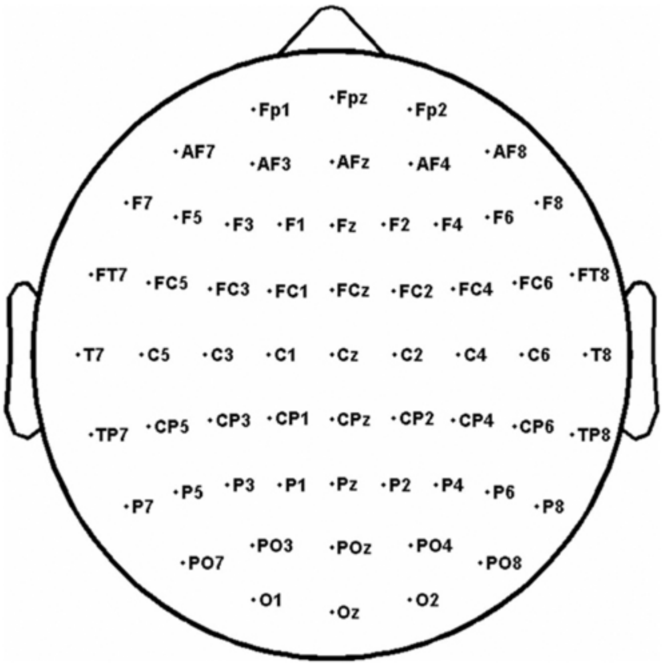


Figure 2.1 – Arangement of EEG electrodes according to the 10-10 positioning system [PBZ11].

facts, ranging from methods for detection and removal [NWR10, MJBB11], to methods of isolating and filtering of artifacts [JMW$^+$00, SKZ$^+$07].

An alternative or complement to EEG is functional Near Infrared Spectroscopy (fNIRS). fNIRS captures the hemodynamic response to cognitive activity by exploiting the fact that oxygenated and de-oxygenated blood absorbs different proportions of light of different wavelengths in the near-infrared spectrum. As active brain regions consume more oxygen than inactive ones, the ratio of oxygenated and de-oxygenated blood correlates with brain activity. For fNIRS recordings, a set of optodes is placed on the surface of the participant's head. The optodes function as light sources and detectors.

On the one hand, due to both the origin of the signal (hemodynamic response for fNIRS vs. electrical process for EEG), EEG has a much higher temporal resolution than fNIRS. On the other hand, fNIRS potentially has a higher spatial resolution (compared to an EEG recording for which the number of electrodes is similar to the number of optodes in the fNIRS recording) and is less prone to certain types of artifacts, such as those from muscular or ocular sources, see [SGC$^+$09]. The different origin of both signal types creates the potential for the combination of both signal types. A model which extracts information from both EEG and fNIRS may be more accurate compared to a system which uses only one signal type as each signal type may capture different cognitive processes to complement the other one.

In contrast to audio and video sensors, physiological sensors and especially those to record brain activity, require direct contact to the user's body. Additionally, some of them, like classical EEG caps, require a cumbersome setup procedure prior to application. This is a serious limitation for the acceptance of interaction systems relying on such sensors. However, due to the fast-paced development of high-quality electrodes and miniaturization, producers of gaming accessory and traditional EEG equipment are in a continuous process of deploying smaller and more convenient devices which work wireless, without electrode gel and with fixed electrode setups. Those recording devices make BCI a possibility for the application in real-world scenarios, especially if we can demonstrate a significantly measurable benefit for the user.

2.2.2 Building Blocks of an Empirical Cognitive Model

Technically, the task of an empirical cognitive model can be formulated as a classification problem: Choosing from a finite set of values, the model has to classify the manifestation of the user state from a given signal segment. In this thesis, we will be mostly looking at binary classification. In theory, it would be desirable to train regression models of user states which predict a user state on a continuous scale (e.g. workload level). However, the granularity even for subjective assessments of such user states is limited to discrete levels [RDMP04], which severely limits the usefulness of regression models.

Empirical cognitive models can be categorized by several criteria. One of those criteria is the differentiation between models which are trained from labeled training data (supervised classifiers) and those which use clustering methods on unlabeled data (unsupervised methods). In this thesis, we concentrate on supervised methods as those allow us to make use of knowledge about the class distribution in the training data. Another important criterion is whether the model processes stimulus-locked data, i.e. data which is segmented from a data stream with a fixed temporal distance to a known event, or performs continuous classification which does not rely on such information. In this thesis, we will look at both types of models. Furthermore, we can discriminate empirical cognitive models between person-dependent, person-independent and person-adapted models. Those groups of models describe the type of training data which is used to build the model. A person-dependent model is trained only on data of the current user. A person-independent model is trained on data from multiple persons and can be applied to previously unseen users (it may require some amount of unlabeled calibration data). A person-adapted model is between those extremes as it combines person-independent training data with labeled data of the current user (but in general requires less person-specific training material than a purely person-dependent model). There are other criteria to categorize empirical cognitive models, for example type of employed features, type of classifier, etc. Those criteria will be discussed in the following sections.

The major steps of building an empirical cognitive model are 1) signal preprocessing, 2) feature extraction and 3) training of the statistical classifier. For training and evaluation of the model, we further need to 4) provide labeled data and to 5) define evaluation metrics. The remainder of this section provides the necessary fundamentals on those five central steps for building an empirical cognitive model.

Preprocessing

Preprocessing manipulates the incoming digital signal (i.e. after A/D transformation and sampling). The main purpose of the preprocessing step is to increase the signal-to-noise ratio, for example by removing the influence of artifacts on the signal.

Typical preprocessing steps for EDA, BVP, RESP signals are drift removal by highpass filtering normalization and smoothing by median filtering or by lowpass frequency filtering. Signal drifts can for example be caused by warming of sensors or skin. Smoothing has the goal of removing high-frequency components of the signal which are not caused by the measured physiological construct but by artifacts, for example movement of the sensor on the skin.

Careful signal preprocessing is especially relevant for signals of brain activity. To handle the impact of artifacts on the EEG signal and therefore on the resulting features and classification results, a large number of methods has been proposed. One group of methods tries to identify artifacts, for example by inspecting the signal for certain non-EEG characteristics. The other group of methods tries to remove the influence of the artifacts on the signal. One very important tool for the latter group is the Independent Component Analysis (ICA). Because of the importance of this method for many EEG applications, we will now have a more detailed look at it: Independent component analysis (ICA) is a statistical blind source separation approach [JH91]. Its goal is to decompose a set of mixed input signals into a statistically independent set of unmixed source signals (called components). For EEG preprocessing, these components may represent localized brain activity or a signal portion generated from an artifact source. Such artifact components can be filtered or removed to clean the remaining signal from artifacts.

ICA transformation from signal space to component space is calculated as a linear transformation applied to each data sample. More formally, let x_i $(i = 1 \ldots, n)$ be a set of n observed (EEG) channels mixed from the n sources $s_1, \ldots, s_n$. It is assumed that the observations x_i are a linear combination of the sources:

$$x_i = a_{i1}s_1 + a_{i2}s_2 + \ldots + a_{in}s_n, \; \forall\, i = 1, \ldots, n.$$

This can be written using vector and matrix notation:

$$x = \mathbf{A}s \tag{2.1}$$

where $\mathbf{A}$ is called the mixing matrix. The only parameter known in Equation 2.1 is the observation vector x, while $\mathbf{A}$ and s need to be estimated. The main assumption of ICA is that the components s_i are statistically independent. Further it is assumed that the independent components are not normally distributed (with at most one exception). Under these conditions, we can estimate the mixing matrix $\mathbf{A}$: The Central Limit Theorem says that the distribution of a sum of independent random variables converges towards a Gaussian distribution. Therefore, the key to estimate the matrix $\mathbf{A}$ is to maximize the non-Gaussianity of the components s in Equation 2.1 and to minimize mutual information between them to yield statistically independent components [HO00]. This general ICA framework is implemented in form of several different algorithms (for example BinICA, Sobi or AMICA), which differ in their approaches for measuring the non-Gaussianity of a distribution. For artifact removal, usually one or several components are identified as contaminated and then filtered or removed. The inverse transformation is then applied to convert the signal back to a artifact-cleaned EEG signal.

Feature Extraction

The goal of the feature extraction step is to find a compact and generalizing representation of signal characteristics which allow the classification of the signal regarding the given classes. In most cases, the continuous signal stream is cut into overlapping windows. One feature vector is calculated for each

Band Name	Range
θ	4–8 Hz
α	8–13 Hz
β	13–30 Hz
γ	>30 Hz

Table 2.1 – Traditional frequency bands in EEG.

window. Window size and degree of overlap are tuning parameters of an empirical cognitive model: While long windows and high overlap provide more data for reliable feature estimation, short windows and low overlap allow fast reaction to changes in user state.

Feature extraction for physiological signals can take place in time domain and frequency domain. Time-domain features are most useful when we need to recognize a temporally localized reaction to a well-defined event or (sensory, cognitive or motor) stimulus. For EEG, such reactions are called Event Related Potentials (ERPs). [BLT^{+}11] gives an overview on methods for extracting time-domain features from the EEG signal with spatial-temporal filters. Those in essence perform downsampling of the signal at specific channels using non-uniform sampling intervals. Similar methods also apply to other physiological signals.

Features from the frequency domain provide another form of signal representation. Many physiological signal characteristics can be expressed in the frequency domain, for example heart rate, respiration frequency, etc. For measuring mental load or activity from EEG, the power distribution of the frequency spectrum is a popular feature. This is motivated by the very early observation that the power in different frequency bands responds differently to different levels of cognitive activity. For example, increasing relative power in the frequency range of 8 Hz to 13 Hz is correlated with increasing cognitive load [vWSG84]. Table 2.1 shows the traditional definition of five frequency bands measured in EEG. While those bands allow very general conclusions on cognitive activity (e.g. resting state vs. cognitive activity), features for classification of cognitive states usually require a finer resolution to represent the highly individual frequency power distributions [KRDP98] and more complex user states (e.g. to determine the type of cognitive activity). For a more detailed description of (EEG) frequency features, [KM11] compares a number of feature characterizations which use different algorithms and representations to extract frequency information.

Data Labeling

In this thesis, we concentrate on supervised statistical classifiers. Training of such a classifier requires to provide labeled training feature vectors. The ground truth for labeling the classes of the training data must be provided externally. There are different approaches to provide labels for user states. Depending on the user state and the style of data collection, this can be done a-priori (i.e. before the data collection) or post-hoc (i.e. after the data collection). We will discuss the different labeling approaches for the example of the user state workload. For this user state, a-priori labels can be generated from different task difficulty levels which can be assumed to correlate to the resulting workload level. Post-hoc labels for the user state of workload can be generated using single- or multidimensional subjective workload assessment questionnaires [RDMP04, Pau08]. One instrument for subjective workload assessment which is used throughout this thesis is the NASA Task Load Index (TLX) [HS88]. The TLX is a six-dimensional questionnaire which defines workload as the individually weighted sum of the scales for mental demand, physical demand, time pressure, performance, required effort and frustration. Each dimension is measured on a 20-point scale. We often resort to the RTLX variant [BBH89] which uses an unweighted sum of the individual, which removes the time consuming process to determine the weights.

Classification

As the final step for building an empirical cognitive model, a statistical classifier is trained. While there exists a large variety of available classifiers (for example Artificial Neural Networks, Gaussian Classifiers or Bayesian methods), we concentrate on the two classifiers which are the most frequently used classifiers for BCIs: Linear Discriminant Analysis (LDA) and Support Vector Machines (SVMs). Both offer good generalization capabilities and are fast to compute. LDA (we follow the characterization of [DHS12]) is a linear feature transformation technique which maximizes between-class scatter while simultaneously minimizing within-class scatter on the training data. A transformation is defined as a linear function $x \mapsto w^T x$. LDA chooses the weight vector w which maximizes the discriminability of classes in the training data. To this end, we first define the *scatter* $\tilde{s}_i^2$ as a measure of the spread of a sample Y_i with label i and mean feature vector m_i:

$$\tilde{s}_i^2 = \sum_{y \in Y_i} (w^T y - w^T m_i)^2 \qquad (2.2)$$

Using this definition, we calculate the *within-class scatter* $\tilde{s}_1^2 + \tilde{s}_2^2$ yielding a measure of the samples' joint spread. Having this, we can define the Fisher linear discriminant function as the vector w that maximizes the criterion function

$$J(w) = \frac{|w^T m_1 - w^T m_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}, \quad |w| = 1. \tag{2.3}$$

This criterion maximizes the distance between the projected sample means, while minimizing the joint within-class scatter by varying the direction of w. This can be solved as a generalized Eigenvalue problem. The LDA projects the data in a one-dimensional space[1] and we can easily train a linear discrimination function as a threshold in the center between the projected classes of training data.

A Support Vector Machine (SVM) [CV95] is a linear binary statistical classifier (extensible to more classes [WWo99]) that is optimized during training to maximize the margin of the training samples to the separating hyperplane. Given a set of sample-label pairs (x_i, y_i), $i = 1, \ldots, n$ with $x_i \in \mathbb{R}^p$ and $y \in \{-1, 1\}^n$ the SVM hyperplane can be identified by solving the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \tag{2.4}$$

$$\text{such that:} \quad \begin{array}{rcl} y_i \cdot (w^T x_i + b) & \geq & 1 - \xi_i, \\ \xi_i & \geq & 0. \end{array} .$$

The normal vector w and the offset b specify the separating hyperplane and the slack variable ξ_i measures the allowed degree of misclassification of x_i. The training examples which are at the minimum distance to the separating hyperplane are called support vectors. Many real-world classification problems are not linearly separable; in these cases, the SVM is often combined with a kernel function Φ which transforms the original feature space to a new space of higher dimensionality, in which linear separability is possible. There are several types of kernel functions, such as linear, polynomial, sigmoidal, or radial basis functions.

When using multiple signal types or multiple types of features, it is often beneficial to combine those different information sources in one fusion classifier. [DK12] shows in a meta-study on affect recognition from different signal types that fusion has a positive effect on recognition accuracy compared to using classifiers which only employ one signal type. This effect is modest in

[1] In this thesis, we only give the definition of a 2-class LDA. In general, the LDA space has at most $k-1$ non-singular dimensions, where k is the number of classes.

size but consistent and statistically significant. There exist many techniques for early fusion (on a signal or feature level) and late fusion (on a decision level). In this thesis, we employ weighted majority voting as decision level technique. This means that one classifier is trained per signal type and classifiers vote on the final result, whereby each vote is scaled with a weight between 0 and 1. The advantages of this decision-level approach (compared to early fusion on feature level) are little dependency on temporal alignment of the involved signal types and the possibility to combine signal types of different reliability by assigning appropriate weights.

Classifier Evaluation Metrics

When assessing the quality of an empirical cognitive model on labeled test data, we can resort to several evaluation metrics. The most common metric is *accuracy*. Accuracy is the ratio of the number of correctly classified samples from the test data set to the number of all samples in the test data set. When regarding test data with unbalanced class distribution or when interested in the ability of the model to detect a specific class c, we can instead calculate *precision* and *recall*. Equation 2.5 gives the definition of both metrics, where TP denotes the number of true positives (i.e. test samples which are correctly classified as c), FP the number of false positives (i.e. test samples incorrectly classified as c) and FN the number of false negatives (i.e. test samples incorrectly not classified as c) for the regarded class. Finally, the *F-Score* is the harmonic mean of precision and recall.

$$\begin{aligned} precision &= \frac{TP}{TP + FP} \\ recall &= \frac{TP}{TP + FN} \end{aligned} \tag{2.5}$$

2.2.3 Workload Recognition

The previous subsection has introduced the fundamentals of empirical cognitive modeling. In this subsection (and the following one), we will turn our attention to related work on models for a specific user state; this subsection focuses on the user state “workload”.

Mental workload is known to influence a large number of physiological parameters – for example heart rate, respiration frequency, or brain activity – which can be directly or indirectly measured and classified. In the following, we will summarize the State-of-the-Art in form of a number of successful evaluations of such systems to establish a common base of methods for empirical workload modeling.

For example, [HP05] used several physiological signals, namely electrocardiogram, electromyogram, skin conductance, and respiration, to reliably classify stress for 5-minute segments of real-world driving of 24 participants, categorized in three different difficulty levels, defined by road characteristics. The authors also performed an analysis of continuous, short-time recognition of workload on 1-second windows. [LJB13] used completely non-invasive sensors like microphones, camera and CAN-Bus information to assess the workload of a driver in a realistic driving situation. The authors showed that certain multi-tasking situations can be discriminated very reliably from pure driving situations using their set of sensors, while multitasking conditions which did induce only little visible or audible behavior changes are more difficult to detect.

[LR11] used EEG to assess driver's workload. The authors used a simulated driving task by employing the Lane Change Task (LCT). While driving, participants were operating a secondary n-back memory task. Both driving task and secondary task were available in two difficulty settings and all combinations were recorded, leading to nine different conditions which were labeled according to the subjective workload assessment instrument NASA TLX. The authors showed significant effects in EEG power spectrum, e.g. power attenuation in the α-band with increased working memory load. They also noted interaction effects on the power spectrum by the two tasks and also noted differences between different types of tasks, e.g. the memory-loading n-back and the vision- and motion-loading LCT task. [Mur05] recognized different workload levels in a continuous matching task where workload is controlled by levels of task difficulty. Evaluation of the NASA TLX and the different reaction times validated that those difficulty levels actually correspond to different workload levels. The authors used wavelets to estimate spectograms of each block and used θ, α and β band power and latency of peak power in each band as features for classification. [BLR$^+$05] used the Aegis simulator and generated realistic combat scenarios to generate five levels of difficulty assigned to certain events and operations. EEG band power features were used to classify four different workload levels on epochs of one second length. The authors showed that they could identify events dichotomized in a high and low workload group with near perfect accuracy. [DWM13] recorded EEG data from 34 participants in a driving simulator. They used stages of different driving demand levels to induce different levels of workload. Common Spatial Filters (CSPs), which maximize the difference in variance between two given classes of signals, are applied to signals filtered with band pass filters at different frequencies to extract participant-specific spatial patterns for feature extraction. The authors also discussed the fact that not all differences in the signals may be caused by actual brain activity but by task-correlated EMG-artifacts. [WHW^{+}12] controls workload levels using different levels of difficulty of the Multi-Attribute Task battery. The authors concentrated on cross-participant classification. They employed a hierarchical Bayesian model trained on data from multiple participants for classification and power in the classical frequency bands as features. They achieved a recognition accuracy of 80% for person-independent classification on a corpus of eight participants. The authors of [BHvE^{+}12] collected EEG data from 35 participants performing an n-back memory task with different levels of working memory load. Apart from frequency-based features, they argued that time-domain features which capture Event Related Potentials (ERPs) like the P300 also change when workload increases. They showed that both types of signal characteristics can be used to predict the workload level and that their combination leads to the best classification results, especially when only little data is available. Changes in P300 characteristics are also exploited by [AP08]. Here, the authors assessed workload induced by a first-person computer game, modulated by game difficulty. To achieve the stimulus-lock required for ERP analysis, a parallel single-stimulus oddball paradigm was used and the P300 response was evaluated as workload index as high workload due to the primary task leads to increased latency and reduced peak amplitude.

As most presented systems rely on some variation of frequency extraction but may differ in the details of preprocessing, spatial filtering or feature calculation, [KM11] reports the results of a comparison of eleven different workload classification BCIs using different features. Data is employed from eight participants performing the Multi-Attribute Task battery, collected in multiple sessions spread across one month. The authors compare feature extraction methods using direct frequency estimates and methods extracting frequency from spatially filtered estimated brain sources.

Recently, fNIRS was established as an alternative input modality for BCIs. While fNIRS provides a lower temporal resolution compared to EEG, it can potentially provide a higher spatial resolution [BPM^{+}11]. The authors of [SZH^{+}08] placed fNIRS optodes on the forehead to measure concentration changes of oxyhemoglobin and deoxyhemoglobin in the prefrontal cortex during memory tasks. Using nearest neighbor classification, they achieved better-than-chance accuracy for all three participants to discriminate between three different levels of workload. Similarly, the authors of [BIA^{+}11] discriminated different workload levels for a complex Warship Commander Task, for which task difficulty was manipulated to create different levels of workload. They record fNIRS from 16 optodes at the dorsolateral prefrontal cortex. While they did not perform single-trial classification, they saw significant differences in oxygenation between low and high workload conditions. There was a significant difference in signal responses to different difficulty settings for expert and novice users, which was mirrored by the behavioral data. [HHF$^+$14] showed that it is possible to classify different levels of n-back difficulty corresponding to different levels of mental workload on a single trials for prefrontal fNIRS signals with an accuracy of up to 78%.

[DK12] performed an extensive meta-study to investigate the impact of fusion methods to improve recognition of user states by combining signals from several signal types. The authors showed that across a large number of studies, fusion leads to a significant improvement in recognition performance, although the effect size is often only modest (they find an improvement of 8.12% from the best individual classifier to the fusion classifier, averaged across all studies). Their results also indicate that the performance of the single-best individual modality has a strong impact on the fusion performance. The term hybrid BCI generally describes a combination of several individual BCI systems. A sequential hybrid BCI allows the first system to act as a “brain switch” [PAB$^+$10], while a simultaneous hybrid BCI system usually combines entirely different brain signals to improve the result of each individual signal modality. The first simultaneous hybrid BCI that was based on simultaneous measures of fNIRS and EEG was proposed by [FMS$^+$12] for classification of motor imagery and motor execution recordings. The authors reported an improvement in recognition accuracy by combining both signal types.[HCG$^+$09] combined EEG and fNIRS data for workload estimation in a counting task and see better results for fNIRS in comparison to frequency based EEG-features. In contrast, [CBE12] presented results from a similar study but showed worse results for the fNIRS features. As those few studies on the combination of EEG and fNIRS present contradictory results, we see that the synergistic potential between both signal types and their applicability to specific classification tasks is still largely unknown.

To summarize, we note that there exists a large body of research on the detection of workload from various physiological sensors, with a focus on sensors which capture brain activity. However, the task of detecting workload is far from solved. Most of the presented evaluations used highly controlled setups, which limits the validity of the results for realistic, uncontrolled scenarios. Evaluation is usually performed on small corpora, which limits the generalizability of the results. Additionally, most research which uses signals of brain activity to recognize workload uses person-dependent systems, which limits the applicability of such systems in situations where a short setup time is required. Finally, nearly all research treats workload as a uniform construct and does not differentiate types of workload, for example induced by tasks which use different input signal types (e.g. vision or hearing). In this chapter, we will address those limitations.

2.2.4 Confusion Detection

In this subsection, we review related work on the empirical cognitive models for the user state confusion. Especially, we are interested in confusion which results from erroneous system behavior caused by recognition errors of user input. There exists a number of systems which make use of confidence scores to estimate the presence of recognition errors [GHW08, SSYH12]. However, when statistical models are unreliable and generate incorrect results, it is unreasonable to expect a very reliable confidence estimate. For example, Vu et al. [VKS10] showed that confidence scores in ASR correlate well to recognition performance for well-trained models but confidence reliability starts to deteriorate for models which are trained on small data corpora or data which does not match the testing data. This indicates that in order to provide self-correcting behavior for a user interface, we need additional information sources on the presence of an error besides confidence scores. One promising candidate in this regard is the detection of Error Potentials (ErrPs) from EEG. An ErrP is a characteristic pattern of brain activity which is triggered by the perception of unexpected feedback of another agent (e.g. a technical system) resulting from erroneous interpretation of the person's activity (e.g. user giving input to the system).

The analysis and recognition of ErrPs for improving both BCI-based and other HCI applications already has some history. [SWMP00] did one of the first investigations of error potentials in the context of (BCI-based) HCI. In their study, four participants performed at least 160 trials of a cursor control task and the authors reveal differences in grand averages of data following correctly classified and misclassified trials. They also noted that EEG data immediately following the completion of a cognitive task (i.e. usually when the feedback is presented) contained systematic eye blink artifacts as participants often suppress blinking until the concentration phase is over. The authors used artifact correction and rejection methods to reduce the influence of those artifacts on their analysis. [FJ08] detected ErrPs from EEG data recorded during the operation of a simulated BCI for spatial control with a predefined error rate of 20%. Using temporal features from electrodes Fz and FCz without correction for ocular artifacts, calculated from 750 trials, they achieved a correct classification rate of 0.82 for both classes (trial with or without ErrP) and were also able to maintain this accuracy when transferring between sessions of the same participant from different days. [ICM$^+$12] described an approach to reduce the number of required calibration trials by transferring data of the same participant from one ErrP task to another. While the authors were able to achieve impressive improvements in calibration time and classification accuracy (from 0.69 to 0.74 accuracy for one condition by using 200 transferred trials), they also see unstable results for some conditions (i.e. for which the adaptation decreased recognition performance). [VS12] propose an ErrP recognition system based on a consumer-level EEG device. They perform person-dependent single trial classification, using a test set of 80 trials of a cognitive task and achieve a recognition accuracy of about 0.7. They also show that already a non-perfect recognition rate between 0.65 and 0.8 is good enough to enhance an interactive system for spatial item selection.

Most of the work on ErrP detection is rooted in the aim of improving BCI performance. As an exception, describes the recognition of ErrPs for the application in general HCI scenarios. They develop a gesture recognizer which performs online learning from trials which were identified as erroneous by an ErrP classifier. Their Bayesian classifier is based on temporal features derived from electrodes at positions Fz and FCz. For each participant, the system uses more than 2700 trials, or over two hours, of data. The reported precision and recall are not very high (0.65 resp. 0.62), but nevertheless impressive due to the realistic, unconstrained task.

To summarize, we see the that it is feasible to develop EEG-based empirical cognitive models for the user state confusion. However, there is a lack of research on models which are explicitly designed for realistic HCI applications. To overcome this lack, we see two main challenges: First, we need to look for ways to reduce the required setup time for the user, for example by providing person-adapted models. Second, we need to evaluate models on realistic data which is recorded not only in pure BCI scenarios. One example of such realistic scenario would target the detection of ErrPs in response to gesture recognition errors. In this chapter, we will address both challenges.

2.3 Person-Independent Workload Recognition

In this section, we pursue the goal of building an empirical cognitive model for the user state *workload*[2]. We implement this model in the form of a person-independent classifier that is able to discriminate two levels of workload induced by a selection and combination of different tasks. We record EEG as well as other physiological signals to investigate the contributions of different signal types to classification accuracy. We also aim for a validation of the potential of workload recognition by providing a reliable estimate of recognition accuracy. For this purpose, we perform the analysis on a large data corpus of more than 150 participants performing a variety of cognitive tasks.

While workload recognition from physiological data is not new per se (see Section 2.2.3), this section contributes to the research community the development of a person-independent workload model (in contrast to the strong focus on person-dependent models in the literature). Furthermore, this model is evaluated on a data corpus which exceeds existing evaluations by far in terms of number of participants, number of evaluated tasks and number of classification conditions (e.g. controlled vs. uncontrolled recording environment, task engagement vs. task count vs. task difficulty).

2.3.1 Tasks and Recording Environments

For training and evaluation of the workload classifier, we collected data sets of 152 participants. The data collection took place during an extensive biosignal collection study 'CogniFit'[3] at Karlsruhe Institute of Technology (KIT). The CogniFit study was a large data collection with the purpose of investigating the relationship between physical activity and cognitive performance. For more details on the CogniFit study, refer to [KBB^{+}12]. Each participant was tested on three days. During one test day, participants performed a number of cognitive tasks while physiological data (EEG, BVP, EDA, RESP) was recorded. For the purpose of the following evaluation, we look at the

[2] The experiments and results in this section are partially based on the diploma thesis of Jan-Philip Jarvis which was co-supervised by the author of this thesis.

[3] We owe special thanks to Janina Krell and Klaus Bös of KIT and Alexander Stahn of *Charité Berlin* for collaborating in participant recruitment and organization of the study. CogniFit was partially funded by a Young Investigator Group grant to Alexander Stahn.

recorded data from this day. Using the performed cognitive tasks, we defined a number of classification conditions of different types. Each classification condition consists of a low workload and a high workload class. The three types of classification conditions are: Task engagement (i.e. discriminating a relaxed state from the state of cognitive activity induced by a task; abbrev. TE), task count (i.e. discriminating a single-task condition from a multi-task condition; abbrev. TC) and task difficulty (i.e. discriminating an easy task from a difficult task; abbrev. TD). Additionally, we also look at two different recording environments: Controlled recordings of a physically inactive participants placed in front of a desktop computer and uncontrolled recordings in an ecologically more valid scenario (i.e. increased relevance for real-world HCI applications) in which the participant moved during recording.

Cognitive Tasks

To increase validity of the evaluation, we cover a number of different cognitive tasks during our recordings. In the following, we briefly explain the employed task paradigms.

In the *Switching task* (SWT) [Mon03], participants were asked to respond to a sequence of trials. In each trial, the participants were presented a numeric stimulus from the set $[1, \ldots, 4, 6, \ldots, 9]$. The stimulus was padded either by a solid or dashed line. If presented with a dashed line, the participants were supposed to determine whether the stimulus was higher or lower than five. If presented with a solid line, the task was to check if the stimulus was even or odd. Participants gave responses using a computer keyboard. The Switching task was conducted in two difficulty levels: For the easy variant, only trials with solid line padding were presented. For the difficult variant, solid and dashed line padding were presented. For both difficulty levels, each trial was presented for 200 ms, with an inter-trial interval of 2000 ms.

The *Lane Change Task* (LCT) [Mat03] was executed in a driving simulator. Here, participants were asked to drive down a three-lane highway at a speed of up to 120 km/h. At regular intervals signs on both roadsides indicated a request for a lane change (see left part of Figure 2.2). The participants were asked to perform the lane changes as they would do in real life and travel at a comfortable and safe speed. The metric to measure LCT performance is the mean distance to the center of the optimal track (the *mean track deviation*).

The *Visual Search Task* (VST) was chosen following [Kuh05]. It was originally implemented to act as an abstraction of operating of a graphical in-car

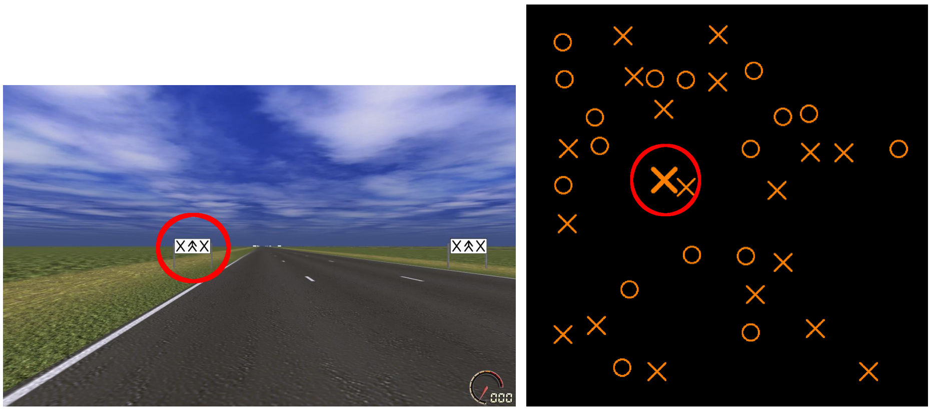


Figure 2.2 – Example of the Lane Change Task (left) and the Visual Search Task (right).

user interface, e.g. for navigation. We used the implementation of the VST provided together with the LCT with modified parameters. The participant was presented a set of symbols (crosses, circles and squares) on a display and was asked to identify and locate a target symbol (see right part of Figure 2.2). The target symbol differs from the distraction symbols in line thickness and symbol size. We defined two levels of difficulty by varying the relative size of target symbol to distraction symbols and the number of fields that can be selected as target symbol location. The participants controlled the task using a number pad keyboard. A new screen was presented whenever participants confirmed a decision on the current screen by pressing a button. The participants' task performance in the VST was measured using the percentage of correctly identified target symbols.

The final cognitive task which we employed is the (auditory) *n-Back* task (ANB). The n-Back (in our implementation) is a working memory task which uses acoustic stimuli. The participants listened to a prerecorded series of numbers in the range of one to ten and were asked to identify pairs of identical numbers placed exactly n positions apart. The task difficulty was controlled by n. We defined two difficulty levels: $n = 1$ for the easy level, $n = 3$ for the difficult level. Independently of the difficulty level, inter-trial distance was at 2000 ms. In this task, the participants responded using a number pad keyboard. The participants' task performance for the ANB was measured using the number of correctly identified targets, independent of the response time.

Recording Environments

The four tasks SWT, LCT, VST and ANB were distributed across two different recording environments. In the controlled recording environment, the participant sat on a chair on a desk in front of a 20" computer monitor. The participant remained motionless apart from manual operation of a keyboard. In this environment, we recorded the SWT (in both difficulty settings) and a relaxing phase (REL). In the uncontrolled recording environment, the participant was seated in a realistic driving simulator, see Figure 2.3. We based our driving simulator on a real car and kept the interior fully intact and functional to provide a realistic in-car feeling. The car is surrounded by a projection wall, covering the view of the windshield and the side windows. The simulator features acoustic feedback via engine sound and environmental surround sound as well as haptic feedback in the seat (via tactile transducers) and steering wheel (via Force-Feedback). In the uncontrolled recording environment, we recorded another relaxing phase as well as the LCT, either without any additional task ("LCT baseline") or combined with either the VST or the ANB task ("LCT+VST" and "LCT+ANB"). As there are two difficulty levels for VST and ANB, this resulted in five runs of the LCT in total. For task response in the driving simulator, a numeric keyboard was strapped to the participant's left thigh. We expected the recorded biosignals in the uncontrolled environment to contain an increased amount of artifacts compared to the controlled recording environment, e.g. caused by movement of the participant. As most real-life HCI scenarios take place in uncontrolled environments, it is important to investigate whether we are still able to correctly discriminate different workload levels in such environments.

Corpus Summary

The SWT in the controlled recording environment was performed twice, with 128 trials for the easy variant and 256 trials for the difficult variant. All tasks in the uncontrolled environment were performed for a duration of 60 seconds for each difficulty level. Relaxing phases lasted for 60 seconds in both recording environments. Additionally, for each task, the participant was allowed a free training period to get comfortable with it. The duration of the training period was variable until participants were comfortable with the operation of the task. Training involved both training of the individual tasks as well as the regarded task combinations. Task order was counterbalanced across participants to avoid systematic temporal effects on the recorded biosignals.

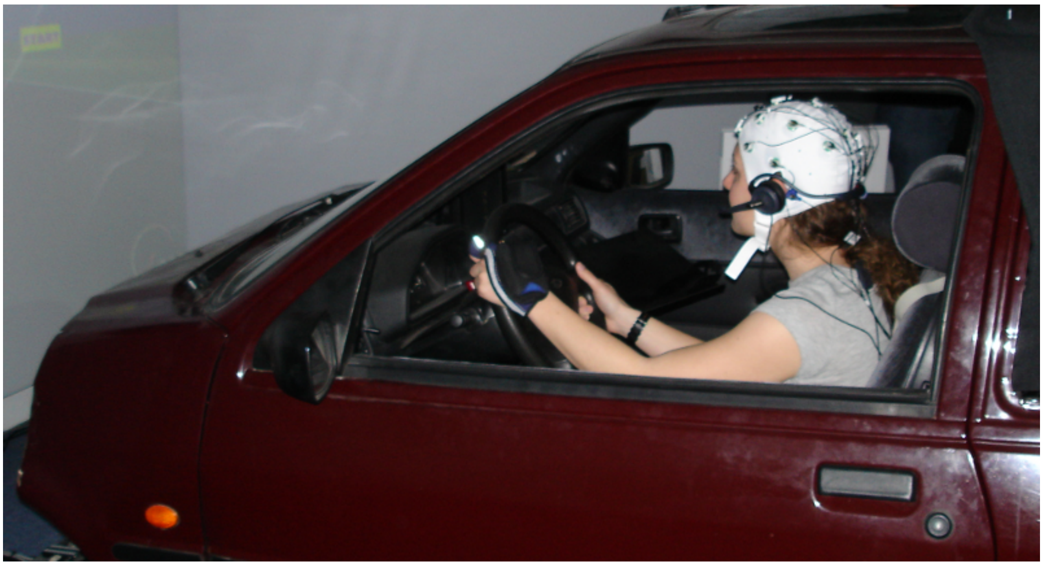


Figure 2.3 – Driving Simulator with biosignal recording setup.

Task(s)	Controlled?
Relax	both
easy SWT	yes
difficult SWT	yes
LCT	no
LCT + easy ANB	no
LCT + difficult ANB	no
LCT + easy VST	no
LCT + difficult VST	no

Table 2.2 – Recorded combinations of cognitive tasks in the CogniFit corpus.

To summarize, the subset of the CogniFit corpus which we use in this section consists of two relaxed recordings (one in each recording environment), one recording of each difficulty level of the SWT, one baseline recording of the LCT, and two recordings each of the VST and the ANB task in combination with the LCT. Table 2.2 summarizes the different combinations of cognitive tasks which were recorded for the CogniFit corpus. In total, 20 minutes of recorded biosignal data for each participant were available. While this corpus composition provided a variety of different task combinations and task difficulty levels, it also created the challenge of data sparseness for each experimental condition. We addressed this challenge by building person-

Id	low workload task					high workload task				
	REL	SWT	LCT	VST	ANB	REL	SWT	LCT	VST	ANB
C_TE	x						diff.			
C_TD		easy					diff.			
U_TE	x							x		diff.
U_TC_ANB			x					x		diff.
U_TC_VST			x					x	diff.	
U_TD_ANB			x		easy			x		diff.
U_TD_VST			x	easy				x	diff.	

Table 2.3 – Binary classification conditions, using data from controlled (u) and uncontrolled (c) recording environments. We differentiate task engagement (TE), task count (TC) and task difficulty (TD) to define different workload levels. The columns list the tasks involved in the definition of the conditions.

independent empirical models. Such a model is able to exploit data from many different persons, as are provided by the CogniFit corpus.

Classification Conditions

Using this set of tasks and recording environments, we define the binary *classification conditions* listed in Table 2.3. Each condition C_TE to U_TD_VST defines a low workload class and a high workload class. In the following subsections, we will implement and evaluate empirical cognitive models to discriminate low from high workload classes for all conditions. We expect the C_TE condition (REL vs. difficult SWT in the controlled recording environment) to be easy to discriminate as cognitive inactivity is characterized for example by typical resting rhythms in the EEG. For the U_TC_ANB and U_TC_VST conditions, we are required to differentiate single-tasking from multi-tasking situations for participants which are always engaged in a cognitive task. The additional coordination effort for multi-tasking compared to single tasking is known to correspond to characteristic neural activation patterns [SSMvC02, DVP+13] which we will exploit to differentiate both conditions. For the C_TD, U_TD_ANB and U_TD_VST conditions, we have to discriminate two structurally identical task setups for which only the difficulty of primary or secondary task varies. We expect that those are the most challenging conditions for person-independent classification as the difference between low and high workload class is not in the presence or absence of a distinct cognitive process, but only gradually in the level of resource demand, which is highly variable between different persons [RDMP04].

2.3.2 Physiological Data Acquisition

To build the empirical cognitive workload model, we record data from a number of different physiological sensors. An active EEG cap (Brain Products' *ActiCap*) was used to measure brain activity. The EEG-cap contains 16 active electrodes placed at the following positions according to the international 10-20-positioning system: FP1, FP2, F3, F4, F7, F8, T3, T4, C3, Cz, C4, P3, Pz, O1 and O2 with reference to the left mastoid. EEG data was recorded at a sampling rate of 256 Hz.

Peripheral blood volume pressure was measured using a photoplethysmograph to measure blood volume pressure (BVP) placed on the participant's left index finger and recorded at a sampling rate of 256 Hz. Electrodermal activity (EDA) was recorded at a sampling rate of 256 Hz and measured by a skin conductivity sensor placed on the palm of the participant's left hand. EDA and BVP sensor were both placed in a glove transmitting the signals to the recording system via BlueTooth. Respiration (RESP) was measured using a respiration belt at a sampling rate of 512 Hz. The belt was placed below the participant's thorax and worn above all layers of clothing.

We used the BiosignalsStudio [HPA$^+$10] (developed at the Cognitive Systems Lab) to record the various data streams in a simultaneous fashion.

2.3.3 Biosignal Labeling & Behavioral Data

For supervised classification, it is necessary to assign a label to each data segment. The label of a data segment corresponds to the workload level in the segment. This section discusses several possibilities to define labels and looks at relevant behavioral data to investigate the consistency of resulting labels when using different approaches.

Defining labels for user states is ambiguous as many inner states cannot always reliably elicited and observed from the outside. There are three ways to define such labels: Using a-priori labels depending on the condition under which the data segment was recorded, using task performance as an objective post-hoc indicator for workload or using subjective post-hoc self-assessment of the participants from a questionnaire like the NASA-TLX [HS88] (see Section 2.2.2). Potentially, those methods can lead to different labels, for example when a variation of task difficulty does not lead to a correlated variation in task performance because participants are able to compensate for the increased difficulty. In this section, we will compare the different labeling methods and related behavioral data for the CogniFit corpus. This involves an analysis of behavioral data to asses the impact of workload level on human performance in the tasks. Furthermore, we investigate the relationship between the metrics for workload level and performance which are relevant for label definition. We do this to learn whether different labeling methods would lead to substantially different outcomes.

The raw task load index (RTLX) is an instrument (i.e. a questionnaire) for subjective workload assessment. It is the average of the six different TLX scales, mapped to a real number in the interval $[0, 1]$. Zero is the lowest and one the highest measure of perceived workload level assessed by a questionnaire. The participants were presented an RTLX questionnaire sheet after each task in the uncontrolled environment. Therefore, a total of five questionnaires per participant was available for evaluation. Table 2.4 presents the resulting RTLX scores. We see significant differences between the LCT baseline and LCT plus both easy secondary tasks. Additionally, there are also significant differences in RTLX score between the LCT plus easy secondary task and the corresponding LCT plus difficult secondary task. p-values were corrected for multiple hypotheses testing using the Bonferroni-Holm method ($p < 0.001$ for all listed comparisons). This validates the choice of the classification conditions as *low* and *high* workload tasks.

Task	**RTLX**
LCT baseline	15.91 ($\sigma = 0.11$)
LCT + easy ANB	21.09 ($\sigma = 0.13$)
LCT + diff. ANB	46.01 ($\sigma = 0.12$)
LCT + easy VST	31.48 ($\sigma = 0.13$)
LCT + diff. VST	45.96 ($\sigma = 0.12$)

Table 2.4 – Mean and standard deviation of RTLX results for different tasks.

After inspecting subjective workload assessment, we look at task performance as objective metric of task difficulty. As most conditions in the uncontrolled recording setup where dual-tasks, we have to look at the performance of both LCT and the respective secondary tasks. To assess the impact of secondary task presentation on driving performance, we performed a one-way ANOVA on the participants' LCT performance. In Table 2.5, we present the average LCT performance for all tasks in the uncontrolled recording environment. The differences between the means are very small and consequently, the ANOVA fails to detect significant differences between them. This indicates that participants do not suffer in their main task performance when executing a parallel task but compensate either with additional effort or with reduced attention for the secondary task.

Task	LCT Track Deviation
LCT baseline	13.5 ($\sigma = 3.3$)
LCT + easy ANB	12.37 ($\sigma = 2.4$)
LCT + diff. ANB	16.48 ($\sigma = 4.2$)
LCT + easy VST	14.46 ($\sigma = 2.5$)
LCT + diff. VST	14.95 ($\sigma = 4.7$)

Table 2.5 – Mean and standard deviation of LCT error rate for different combinations of cognitive tasks.

In Table 2.6, we show average performance for the secondary tasks in the uncontrolled recording environment. For the ANB, we measured task performance as average hit rate for the target stimuli. For the VST, we measured task performance as hit rate with respect to the zone the target symbol was in. Task performance significantly decreased in the difficult tasks of both kinds, compared to the corresponding easy tasks. This shows that task difficulty has a substantial impact on secondary task performance.

Task	Hit Rate
LCT + easy ANB	92.36 ($\sigma = 0.10$)
LCT + diff. ANB	82.51 ($\sigma = 0.07$)
LCT + easy VST	95.10 ($\sigma = 5.46$)
LCT + diff. VST	82.93 ($\sigma = 17.41$)

Table 2.6 – Mean and standard deviation of secondary task performance for different combinations of cognitive tasks.

To investigate how we should assign class labels to the different recordings, we performed a prestudy with 13 participants. This study confirmed that the results of the different labeling methods are strongly correlated, see Table 2.7: Task difficulty, secondary task performance and subjective workload assessment all exhibit pairwise correlation of 0.82 or higher. This result implies that using any of those measures to generate labels would yield very similar results (in contrast, LCT performance would lead to very different labels, as it is not correlated to any of the three measures with more than 0.5). This observation encourages us to proceed with a-priori labels (i.e. derived from task count or task difficulty) as those are easy to generate and do not need to be dichotomized for binary classification.

	TLX	**ERR**	**LEV**	μ_{LCT}	σ_{LCT}
TLX	1.00	0.83	0.82	0.45	0.43
ERR	0.83	1.00	0.94	0.54	0.46
LEV	0.82	0.94	1.00	0.43	0.32
μ_{LCT}	0.45	0.54	0.43	1.00	0.80
σ_{LCT}	0.43	0.46	0.32	0.80	1.00

Table 2.7 – Correlation coefficients of potential ground truth scores for VST: RTLX, Secondary Task Error (ERR), Task Difficulty Level (LEV), mean (μ_{TLX}) and std. dev. (σ_{TLX}) of deviation from ideal route.

2.3.4 Workload Classifier Setup

The design of the empirical workload model follows the general processing chain described in section 2.2.2. For each signal type, signals were preprocessed, we calculated features and trained an LDA classifier. To combine information from different signal types, we applied weighted decision fusion. In the following, we describe the different components of the classification system in more detail.

Preprocessing

The main goal of our preprocessing is to increase the signal-to-noise ratio. As signal characteristics are different for each signal type (EDA, BVP, RESP, EEG), we treated each one individually: For EDA, the signal is filtered using a 25 sample median filter to remove white noise artifacts. Additionally, a lowpass filter with a cutoff frequency of 1 Hz was applied to the signal. Finally the linear trend of the signal was removed. For RESP, a rectangular finite impuls response highpass filter with a cutoff frequency of 1 Hz was applied to the respiration signal. Adjacent zero crossings of the filtered signal's slope were then used to extract respiration cycles. The BVP signal was not preprocessed as the employed feature extraction itself is robust to artifacts.

For the EEG signal, a preprocessing step based on Independent Component Analysis (ICA) was applied to remove eye movement artifacts from the raw signal. The usage of ICA for the purpose of reducing eye movement artifacts in EEG signals was proposed in [JMW$^+$00]. As eye movement artifacts occur highly correlated in all electrodes near the frontal lobe, ICA was expected to return a decomposition containing these artifacts in one single component. We use the spectral energy for the different ICA components for automatic detection of an eye movement component: The components were filtered using a band-pass filter with cut off frequencies 1–6 Hz. Choosing the component with the highest energy in this band lead to a robust estimate of the eye movement component. The filtered signal was obtained by applying a high-pass filter with a cutoff frequency of 20 Hz to the eye movement component and calculating the inverse ICA transform of the modified components.

Feature Extraction

After preprocessing, we extracted features for each signal type. To retrieve a semi-continuous measure of the user's workload, we extracted features from sliding windows. In general, the window size governs the time resolution of the classification system. A small window size yields a high temporal resolution but only few samples are contained in each time window to estimate the features. In this work, we aim to reduce the window size as far as possible without affecting classification accuracy. The different physiological processes influencing the physiological signals used in this study require the usage of individual window sizes for each signal type. Therefore, an alignment is necessary for synchronous windowing. We chose to align the windows by the last sample. The window shift governs the decision output frequency of the classifier. We chose a window shift of 0.25 s, yielding a decision frequency of 4 Hz.

We will know look at the concrete features for every signal type. The features derived from the BVP, EDA and RESP signal were chosen following [HP05], which showed their suitability for stress recognition. For the BVP signal, we derive a total number of five features in time and frequency domain from a window of 6 s length:

- mean heart rate (PPG_MEAN),
- heart beat standard deviation (PPG_STD),
- bandpower 0 – 0.08 Hz (PPG_BDPWR_1),
- bandpower 0.08 – 0.15 Hz (PPG_BDPWR_2) and
- bandpower 0-15 – 0.5 Hz (PPG_BDPWR_2)

For each window, the mean heart rate was extracted using a peak detection approach: First, we calculated the Power Spectral Density (PSD) of the signal using Welch's method [Wel67] to determine the mean heart rate frequency. The signal was then bandpass filtered using a rectangular bandpass filter centered at the mean heart rate frequency and using cutoff frequencies of mean heart rate ± 0.2 Hz which corresponds to ± 12 heart beats per minute. A peak detection on the filtered signal was then used to identify the signal peaks. The standard deviation of the time between two heart beats is then calculated from the peak positions and is also used as a feature. The bandpowers for the features PPG_BDPWR_1, PPG_BDPWR_2 and PPG_BDPWR_3 are calculated using a n-point discrete Fourier transform.

For EDA, we extract a total number of six features from a window of 4 s length:

- signal mean (EDA_MEAN),
- standard deviation of EDA signal (EDA_STD),
- number of startles (EDA_ST_NO),
- sum of startle amplitudes (EDA_ST_SUM),
- sum of startle duration (EDA_ST_DUR) and
- area under startles (EDA_ST_AREA).

Startles are sudden rises in the signal amplitude that are automatically detected inspecting the signals slope using a thresholding approach. Startles are associated with reactions to emotional or unexpected stimuli [VSL88].

We extract the following seven features from the RESP signal from a window of 10s length:

- mean respiration rate (RESP_MEAN),
- mean respiration depth (RESP_DEPTH),
- signal energy in the frequency band 0–0.1 Hz (RESP_BP_1),
- signal energy in the frequency band 0.1–0.2 Hz (RESP_BP_2),
- signal energy in the frequency band 0.2–0.3 Hz (RESP_BP_3),
- signal energy in the frequency band 0.3–0.4 Hz (RESP_BP_4).

For feature extraction from the EEG signal, we employed two different methods for frequency analysis, one based on Fourier transformation and one based on Wavelet transformation. For the short time Fourier transform (EEG_B) approach, the signal's PSD was calculated on each window using Welch's method [Wel67]. The resulting coefficients were logarithmized. To smooth the resulting spectrum, we applied a Hamming function to each window. We received 83 frequency coefficients between 4 and 45 Hz for each electrode. We then reduced the dimensionality of the feature space by averaging over three adjacent frequency bins. To generate the final feature vector, we concatenated the features for all electrodes.

For implementation of the EEG Discrete Wavelet Transformation (EEG_W) feature extraction method, we employed Daubechies' order three wavelets (DB3). This approach serves for a good comparability to the traditional frequency bands (see Section 2.2.2), as the pseudo frequencies of the wavelet levels two to five roughly correspond to the $\gamma-$, $\beta-$, $\alpha-$ and $\theta-$ frequency bands [AZD03].

Feature Normalization

To account for inter-personal differences in physiological responses to different workload conditions, we normalized the features for each participant. For different use cases, we compared two different normalization modes. First, we evaluated feature space normalization using unsupervised person-wise z-normalization (mean subtraction and division by standard deviation) in every feature space dimension (INT_NORM) on both training and test data. However, this approach is not suited for many real-world applications, as the complete test data is used for calculation of normalization statistics. This data is not generally available to an online recognizer, for example at the start of a session. Therefore, we also used an alternative normalization scheme calculating mean and standard deviation for each feature and user from a previously recorded calibration data set not used in training or testing (EXT_NORM). This approach can be applied in real-time applications by using bootstrapping of normalization statistics from unlabeled calibration data at the beginning of a session. However, it is important to note that mean and standard deviation estimates – and therefore the z-normalization result itself – are dependent on class distributions, i.e. class distribution in the calibration data must be similar to the distribution in the testing data. In our evaluation, classes were always balanced and this requirement was therefore fulfilled.

Temporal Smoothing

In most real-world applications, we can expect the frequency with which the users' workload level changes to be lower than 4 Hz, which results from a window shift of 0.25 s. Therefore, we applied temporal smoothing to reduce the frequency of workload estimate outputs to increase robustness in turn:

We applied a running majority vote filter to the decision output of each classifier prior to fusion. An n-point running majority vote filter provides a smoothing on the classification-results yielded by a classification system: Let $\omega_1, \ldots, \omega_k$ be the classification outcome (i.e. a list of classes) of a classifier for $k > n$ adjacent windows and n an even filter length. Let $count(\Omega, c)$ denote the counting function which returns the number of occurrences of a class label c in list Ω. We then obtain the filtered classification outcome $\tilde{\omega}_{1+\frac{n}{2}}, \ldots, \tilde{\omega}_{k-\frac{n}{2}}$ by letting:

$$\tilde{\omega}_i = \omega_j, \text{ where } j = \arg\max_c count([\omega_1, \ldots, \omega_k], c)$$

In this work, we applied an eight-point running majority vote filter to the classification outcomes of all individual classifiers prior to decision fusion.

2.3.5 Evaluation of Classification Performance

One central goal of this section was the extensive validation of the person-independent empirical workload model. Consequently, the evaluation of the classifiers consists of several parts: The first three parts deal with the classification performance of the model for three different categories of classification conditions: detection of task engagement, detection of task count and detection of task difficulty. We performed these evaluations by using leave-one-participant-out cross-validation on the first 75 sessions of the data corpus. In the fourth part of the evaluation, we repeated the analyses from the first three parts, but trained on the whole 75 first sessions and used the remaining sessions of the corpus as a one-time testing set. This is followed by the fifth part of the evaluation, which focuses on task transfer, i.e. the evaluation of a model which was trained on one classification condition on a different classification condition. In the final part of the evaluation, we investigated the effect of temporal ordering on classification accuracy to rule out than systematic temporal effects influenced classification accuracy.

As evaluation metric, we report classification accuracy in all parts of the evaluation. Note that for each condition, the low workload and high workload class are balanced, i.e. baseline accuracy of a trivial classifier is 50%.

Detection of Task Engagement

In Table 2.8, we see classification accuracy for the different signal types as well as the decision fusion (DEC) for classification conditions C_TE and U_TE

	EEG_B	EEG_W	BVP	EDA	RESP	DEC
C_TE (INT_NORM)	92.5	88.79	83.81	89.93	81.81	**95.31**
C_TE (EXT_NORM)	85.96	85.0	79.05	86.97	81.92	**93.46**
U_TE (INT_NORM)	76.94	64.28	84.0	71.54	75.99	**85.02**
U_TE (EXT_NORM)	82.95	64.90	86.93	77.97	83.03	**91.72**

Table 2.8 – 2-class classification accuracy using session wise leave-one-participant-out cross-validation on the C_TE and U_TE classification tasks (relax vs. task engagement) using INT_NORM and EXT_NORM normalization.

which aim for the discrimination of task engagement vs. relax in both the controlled and uncontrolled environment. We observe that while the controlled recording environment provided higher classification accuracy than the uncontrolled recording environment (95.3% vs. 85.02% for the DEC classifier using INT_NORM), task engagement can be reliably detected in both environments. Furthermore, we see that in general, classifiers for all signal types were capable of discriminating low vs. high workload. The fusion classifier was superior to all individual classifiers. This supports the idea of combining several signal types for robust classification. Finally, we observe that the application of EXT_NORM had no adverse effect on classification accuracy. On the contrary, in this evaluation, the average result using EXT_NORM actually outperforms INT_NORM. One possible reason is the larger variability of the EXT_NORM calibration data, leading to a more robust estimation of normalization parameters.

Detection of Task Count

The classification tasks U_TC_ANB and U_TC_VST defined low and high workload classes using single-task and dual-task conditions rather than using a relaxed state and task execution. Table 2.9 summarizes the average classification accuracy. Multitasking is more difficult to detect from physiological parameters than task engagement, because for the U_TC_ANB and U_TC_VST classification task, the participants are always cognitively active. Consequently, fusion classification rates drop to 70.89% and 76.89% for INT_NORM and respectively 66.48% and 69.15% for EXT_NORM. We note that in contrast to the previous evaluation, the fusion system does not outperform all individual classifiers. This is because the EEG features are now contributing most to the overall result, while the other physiological features hardly surpass the random baseline in terms of accuracy. This indicates that EEG is more suited for classification tasks with more subtle differences between

	EEG_B	EEG_W	BVP	EDA	RESP	DEC
U_TC_ANB (INT_NORM)	70.42	**77.29**	47.93	58.10	55.76	70.89
U_TC_ANB (EXT_NORM)	61.34	**69.29**	50.2	63.09	48.11	66.48
U_TC_VST (INT_NORM)	76.88	72.17	50.43	56.88	67.07	**76.89**
U_TC_VST (EXT_NORM)	**75.7**	65.4	52.41	57.18	53.06	69.15

Table 2.9 – 2-class classification accuracy using session wise leave-one-participant-out cross validation on the U_TC_ANB and U_TC_VST classification tasks (single-task vs. multi-task) using INT_NORM and EXT_NORM normalization.

conditions. Between the two different feature types for EEG, there is no clearly superior approach. Therefore, decision fusion is still beneficial – even if outperformed by a single EEG classifier in three of four cases – as it reduces the risk of choosing a suboptimal EEG feature for a given classification condition.

The conclusions we draw from the results of the first two evaluation steps is as follows: Simple physiological signals like BVP, EDA and RESP are good choices for simple classification tasks (e.g. relax vs. task engagement), when differences in signal characteristics are large. When the two classes become more similar (e.g. single- vs. multi-tasking), performance of physiological signals deteriorates. In contrast, the performance for the EEG based classifiers was only a bit lower for the U_TC_ANB and U_TC_VST conditions compared to the task engagement conditions, but stayed relatively stable and provided classification accuracy still far beyond the baseline. This indicates that a person-independent workload classification from EEG is feasible and more versatile compared to other physiological signals.

Detection of Task Difficulty

Table 2.10 shows results for the most difficult classification tasks, i.e. detecting workload level induced by varying task difficulty for a single-tasks (C_TD) and two dual-tasks (U_TD_ANB, U_TD_VST). Here, task count was identical for both classes, only the difficulty of the tasks varied. For this category of classification conditions, we only employed INT_NORM, as the large individual differences required calibration data which optimally matched the testing condition. As expected, accuracy was lower compared to the other classification tasks. Still, the models significantly outperformed ($p < 0.001$ for all three cases) the random baseline for all three conditions with the de-

	EEG_B	EEG_W	BVP	EDA	RESP	DEC
C_TD	71.08	69.91	62.92	61.56	62.02	**76.19**
U_TD_ANB	61.68	**62.38**	53.26	40.72	54.62	59.95
U_TD_VST	52.53	54.13	56.12	60.51	59.24	**64.34**

Table 2.10 – 2-class classification accuracy using session wise leave-one-participant-out cross validation on the C_TD, U_TD_ANB and U_TD_VST classification tasks (task difficulty) using INT_NORM normalization.

	EEG_B	EEG_W	BVP	EDA	RESP	DEC
U_TE	81.18	75.76	86.97	66.83	83.33	**93.41**
U_TC_ANB	63.49	**66.96**	51.50	58.20	49.58	65.86
U_TC_VST	**73.99**	68.63	48.49	53.97	53.104	66.02

Table 2.11 – 2-class classification accuracy obtained on a test set for different classification conditions using the EXT_NORM normalization method.

cision fusion classifier. As expected, classification accuracy was highest for the controlled condition C_TD. For C_TD and U_TD_ANB, the best individual classifier was based on EEG features. For U_TD_ANB, EEG did not outperform the other signal types. Overall, this part of the evaluation shows that person-independent classification was able to contribute to a discrimination between different workload levels. This holds even if workload is not induced by the number of processed tasks but by task difficulty.

Evaluation on Test Set

The previous results were obtained using session-wise cross validation on the first 75 sessions of the data corpus. In Table 2.11, we present the classification accuracy values yielded by our classification system trained on these 75 sessions and evaluated on the test set consisting of the remaining 77 sessions of the CogniFit corpus. This test set was never presented to the algorithm or used by the developer during the development and tuning process of the classification systems. Although the classification accuracies of the individual classifiers differ slightly from the values obtained during cross validation, the accuracy scores obtained by decision fusion are within 3% absolute range of the results obtained by cross-validation for all investigated classification conditions (and even slightly better for U_TC_VST). These results indicate that the system generalizes well across participants.

	EEG_B	EEG_W	BVP	EDA	RESP	DEC
U_TC_ANB on U_TC_VST	53.17	**66.89**	47.46	56.87	49.20	55.61
U_TC_VST on U_TC_ANB	58.17	**69.47**	48.89	53.78	47.93	61.64

Table 2.12 – 2-class classification accuracy obtained on test set for task transfer between U_TC_ANB and U_TC_VST using the EXT_NORM normalization method.

Evaluation of Task Transfer

In the introduction, we argued that multi-tasking vs. single-tasking (i.e. the task count) could be identified from general cognitive processes of coordination between multiple tasks, captured by EEG. To evaluate this claim, we tested our person-independent models for task transfer. For this purpose, we modified the previous evaluation on the test set in the following way: A model was trained on the training set of one multi-tasking condition and evaluated on the test set of the other multi-tasking condition. For example, in "U_TC_ANB on U_TC_VST", the classifier was trained on data from the U_TC_ANB task and evaluated on data from the U_TC_VST task. We expected a drop in classification accuracy compared to evaluations with matching training and test conditions, as we transfer between an auditory memory task and a visual attention task, i.e. it is not possible to transfer knowledge on the required cognitive resources. Table 2.12 summarizes the results. We see that while the model lost some performance due to missing task-specific information, we were still able to achieve classification accuracy close to 70% from the EEG data. We see that the Wavelet-based features outperform the STFT features which seem to capture more of the task specific feature characteristics. Also note that the classification of the fusion system is not competitive in this evaluation. This is because fusion weights depend highly on the classification condition, as seen in the previous evaluations. For task transfer, weights were estimated on a condition which does not match the condition of the test data. This lead to weights which do not reflect the accuracy of the individual classifiers on the test data reliably.

Effects of Task Order

We know from other investigations of EEG analysis that ordering of tasks potentially has a strong effect on the resulting EEG data [PWCS09]. Task order during data collection can impact the participants' inner state and thus may negatively impact the validity of the features and the corresponding ground truth from which the model is trained. Therefore, we analyzed the impact of the task order in the uncontrolled data collection scenario on the physiological measures used for feature extraction.

To assess the impact of task order on the recorded signals, we conducted an evaluation of a model which used training data from the U_TE setup (REL vs. LCT + diff. ANB). Instead of workload-dependent class labels, we used the index of the respective experiment in the task order as class label. This means: data collected during the REL data was labeled with class label ONE if REL was conducted first for the regarded participant. If LCT + diff. ANB was conducted first, the REL data was labeled with class label TWO. The opposite holds for the class labels assigned to the data recorded during the LCT + diff. ANB. As stated in Section 2.3.1, we adopted four different task order schemes in the uncontrolled data collection scenario. In two of the task orders, the LCT + diff. ANB was presented before the REL and in two task orders the LCT + diff. ANB was presented after REL. Therefore, we can assume the training data class distributions to be balanced for each fold of a session-wise leave-one-out cross-validation on the first 75 sessions of the corpus. When evaluating this classification condition, none of the classifiers was able to accurately discriminate the sessions' task orders significantly better than chance ($p > 0.05$ for all classifiers in one-sample t-test against the random baseline). This indicates that task order had no systematic impact on the physiological measures used for this work.

2.3.6 Discussion

In this section, we described the implementation and evaluation of a person-independent empirical cognitive model to discriminate workload levels defined by task engagement, task count and task difficulty. We used a large data corpus which exceeds the size of all known corpora on physiological data for workload recognition. We investigated task transfer, reproduced the cross validation results on an unseen test set and validated the results concerning physiological plausibility and task order. The results showed the generalizability of the learned models. To our best knowledge, all those results contribute valuable novel findings to the research community.

Still, there are some open questions which remain for future research: While we employed different cognitive tasks during the evaluation and also evaluated in uncontrolled recording environments, the used tasks are still not very naturalistic. Recordings which emerge from more realistic (HCI) tasks are more heterogeneous in workload distribution over time. Switching to more ecologically valid tasks to induce workload would therefore add new challenges to the recognition task, as simple temporal fusion will not suffice anymore to model the temporal patterns of such tasks. Another limitation is that the method which we applied for transferring classification models between participants and tasks only relies on normalization to reduce the difference between recordings. In the future, we suggest to use more sophisticated methods of transfer learning [PY10], for example transferring knowledge of feature representations between setups, to identify a generalizing joint feature space.

2.4 Workload Modality Recognition

In this section, we describe the design and evaluation of an empirical cognitive model that can detect or discriminate perceptual processes for different modalities from measures of brain activity[4]. We investigate how reliably a hybrid BCI using synchronous Electroencephalography (EEG) and functional Near Infrared Spectroscopy (fNIRS) signals can perform such classification tasks. We describe an experimental setup in which natural visual and auditory stimuli are presented in isolation and in parallel to the participant of which both EEG and fNIRS data is recorded. On a corpus of 12 recorded sessions, we trained empirical cognitive models using features from one or both signal types to differentiate and detect the different perceptual modalities.

Multimodality refers to both the possibility to operate a system using multiple input modalities and to the ability of a system to present information using multiple output modalities. For example, a system can either present information on a screen using text, images and videos or it can present the same information acoustically by using speech synthesis and sounds. However, such a system has to select an output modality for each given situation. One important aspect to consider is the user's workload level. If the workload level is too high, it may negatively influence task performance and user satisfaction. The workload level of the user also depends on the output modality of the system. Which output modality imposes the smaller workload on the user depends mainly on the concurrently executed cognitive tasks. Examples for such concurrent tasks are: a spoken conversation with another person, watching television or virtually any other engagement with a perceptual component. Especially in dynamic and mobile application scenarios, users of a

[4] This section is partially based on the results of the diploma thesis of Sebastian Hesslinger which was co-supervised by the author of this thesis.

system are frequently exposed to external stimuli from other devices, people or their general environment.

According to the multiple resource theory of [Wic08], the impact of a dual task on the workload level depends on the type of cognitive resources which are required by both tasks. If the overlap is large, the limited resources have to be shared between both tasks. Consequently, overall workload will increase compared to a pair of tasks with less overlap, even if the total individual task load is identical. For example, [YRM$^+$12] showed a study in which they combined a primary driving task with an auditory or visual task. They showed that the difference in the performance level of the driving task depends on the modality of the secondary task: According to their results, secondary visual tasks impacted the driving task much more than secondary auditory tasks, even if individual workload of the auditory tasks was slightly higher than of the visual tasks. Neural evidence from a study [KMSM13] of participants switching between bimodal and unimodal processing has also indicated that cognitive resources for visual and auditory processing should be modeled separately. Most basic visual processing takes place in the visual cortex of the human brain, located in the occipital lobe, while auditory stimuli are processed in the auditory cortex located in the temporal lobes. Both brain areas might be captured by non-invasive EEG or fNIRS sensors. This clear localization of important modality-specific processing at accessible sites in the cortex hints at the feasibility of separating both types of processing modes.

These observations on multimodal stimulus processing imply that the choice of output modality should consider the user's cognitive processes. It is possible to model the resource demands of cognitive tasks induced by the system itself (see [CTN09]). For example, we know that presenting information using speech synthesis requires auditory perceptual resources while presenting information using a graphical display will require visual perceptual resources. However, modeling independent cognitive tasks (i.e. not induced by the system) is impossible in an open-world scenario where the number of potential distractions is virtually unlimited. Therefore, we have to employ sensors to infer which cognitive resources are occupied. To some degree, perceptual load can be estimated from context information gathered using sensors like microphones or cameras. However, if, for example, the user wears earmuffs or head phones, acoustic sensors cannot reliably relate acoustic scene events to processes of auditory perception. Therefore, we need a more direct method to estimate those mental states. An empirical cognitive model can detect or discriminate perceptual processes for different modalities directly from measures of brain activity and is therefore a strong candidate to reliably discriminate and detect modality-specific perceptual processes. In this section, we investigate how reliably a hybrid BCI using synchronous Electroencephalography (EEG) and functional Near Infrared Spectroscopy (fNIRS) signals can perform such classification tasks.

This section contributes a number of substantial findings to the field of empirical cognitive modeling for HCI: We train and evaluate two types of classifiers: First, we look at classifiers which which discriminate between predominantly visual and predominantly auditory perceptual activity. Second, we look at classifiers which were able to detect visual or auditory activity (e.g. discriminate modality-specific activation from other or no activation) independently of each other. The latter is ecologically important as many real-life tasks demand both visual and auditory resources. We show that both types of classifiers achieved a very high accuracy both in a person-dependent and person-independent setup. We investigate the potential of combining different feature types derived from different signals to achieve a more robust and accurate recognition result.

2.4.1 Participants

12 healthy young adults (6 male, 6 female), between 21 and 30 years old (mean age 23.6 ± 2.6 years) without any known history of neurological disorders participated in this study. All of them have normal or corrected-to-normal visual acuity, normal auditory acuity, and were paid for their participation. The experimental protocol was approved by the local ethical committee of National University of Singapore, and performed in accordance with the policy of the Declaration of Helsinki. Written informed consent was obtained from all participants and the nature of the study was fully explained prior to the start of the study. All participants had previous experience with BCI operation or EEG/fNIRS recordings.

2.4.2 Experimental procedure

Participants were seated in a sound-attenuated room with a distance of approximately one metre from a widescreen monitor (24” BenQ XL2420T LED Monitor, 120Hz, 1920x1080), which was equipped with two loudspeakers on both sides (DELL AX210 Stereo Speaker). During the experiment, participants were presented with movie and audio clips, i.e. silent movies (no sound; `VIS`), audiobooks (no video; `AUD`), and movies with both video and audio (MIX). We have chosen natural, complex stimuli in contrast to more controlled, artificially generated stimuli to keep participants engaged with the materials and to achieve a realistic setup.

Besides any stimulus material, the screen always showed a fixation cross. Participants were given the task to look at the cross at all times to avoid an accumulation of artifacts. When there was no video shown, e.g. during audio clips and during rest periods, the screen pictured the fixation cross on a dark gray background. In addition to the auditory, visual and audiovisual trials, there were IDLE trials. During IDLE, we showed a dark gray screen with a fixation cross in the same way as during the rest period between different stimuli. Therefore, participants were not be able to distinguish this trial type from the rest period. In contrast to the rest periods, IDLE trials did not follow immediately after a segment of stimulus processing and can therefore be assumed to be free of fading cognitive activity. IDLE trials were assumed to not contain any systematic processing of stimuli. While participants received other visual or auditory stimulations from the environment during IDLE trials, those stimulations were not task relevant and of lesser intensity compared to the prepared stimuli. In contrast to AUD, VIS and MIX trials, there was no additional resting period after IDLE trials. In the following, we use the term *trial type* to discriminate AUD, VIS, MIX, and IDLE trials.

The entire recording, which had a total duration of nearly one hour, consisted of five blocks. Figure 2.4 gives an overview of the block design. The first block consisted of three continuous clips (60s audio, 60s video, 60s audio&video with a break of 20s between each of them. This block had a fixed duration of 3 minutes 40 seconds. The remaining four blocks had random durations of approximately 13 minutes each. The blocks 2–5 followed a design with random stimulus durations of 12.5s ± 2.5s (uniformly distributed) and rest periods of 20s ± 5s (uniformly distributed). The stimulus order of different modalities was randomized within each block. However, there was no two consecutive stimuli of the same modality. Figure 2.5 shows an example of four consecutive trials in the experiment. Counted over all blocks, there were 30 trials of each category AUD, VIS, MIX and IDLE.

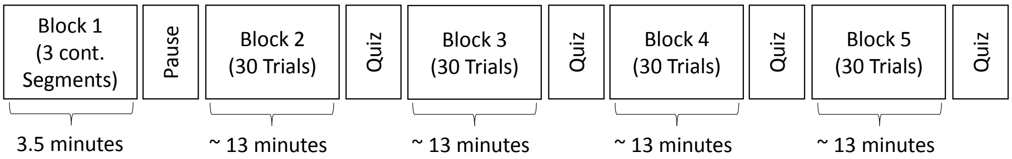


Figure 2.4 – Block design of the experimental setup.

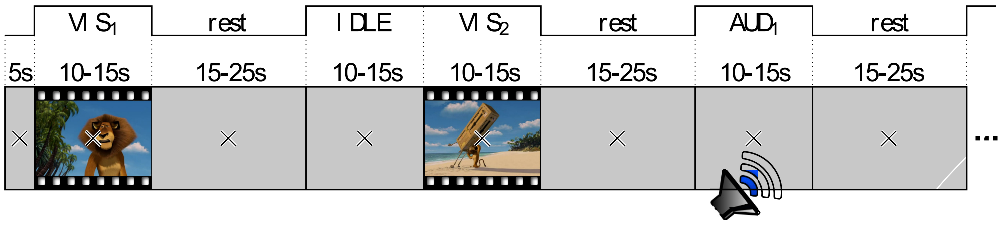


Figure 2.5 – Example of four consecutive trials with all perceptual modalities.

The stimuli of one modality in one block formed a coherent story. During the experiment, subjects were instructed to memorize as much of these stories (`AUD`/`VIS`/`MIX` story) as possible. In order to ensure that participants paid attention to the task, they filled out a set of multiple choice questions (one for each story) after each block. This included questions on contents, e.g. "what happens after. . . ?", as well as general questions, such as "how many different voices appeared?" or "what was the color of . . . ?". According to their answers, all participants paid attention throughout the entire experiment. For the the auditory trial type, participants achieved an averaged correct answer rate of 85%, whereas for the visual trial type there is a correct answer rate of 82%.

2.4.3 Data acquisition

To capture fNIRS, a frequency-domain oximeter (Imagent, ISS, Inc., Champaign, IL, USA) was used for optical brain imaging. Frequency-modulated near-infrared light from laser diodes (690 nm and 830 nm, 110 MHz) was conducted to the participants head with 64 optical source fibers (32 for each wavelength), pairwise co-localized in light source bundles. A rigid custom-made head-mount system (montage) was used to hold the source and detector fibers to cover three different areas on the head: one for the occipital cortex (vision-related area) and one on each side of the temporal lobe (audition-related area). The multi-distance approach as described in [WWC$^+$03, JHFB06] was applied in order to create overlapping light channels. Figure 2.6 shows the arrangement of sources and detectors in three probes (one at the occipital cortex and two at the temporal lobe). For each probe, two columns of detectors were placed between two rows of sources each to the left and the right, at source-detector distances of 1.7 cm to 2.5 cm. See Figure 2.6(a) for the placement of the probes and Figure 2.6(b) for the arrangement of the sources and detectors. After separating source-detector pairs of different probes into three distinct areas, there were a total of 120 channels on the visual probe and 110 channels on each auditory probe. Thus, there was a total number of $n_c = 340$ channels for each wavelength. The sampling frequency used was 19.5 Hz. This comparably low sampling rate (when compared to EEG) results from temporal multiplexing of channels. Because of the nature of the fNIRS signal (which is determined by a mechanical process), this sampling rate is sufficient.

EEG was simultaneously recorded with an asalab ANT neuro amplifier and digitized with a sampling rate of 256 Hz. The custom-made head-mount system, used for fNIRS recording, enabled us to place the following 10 Ag/AgCl electrodes according to the standard 10-20 system: Fz, Cz, Pz, Oz, O1, O2, FT7, FT8, TP7, TP8. M1 and M2 were used as reference.

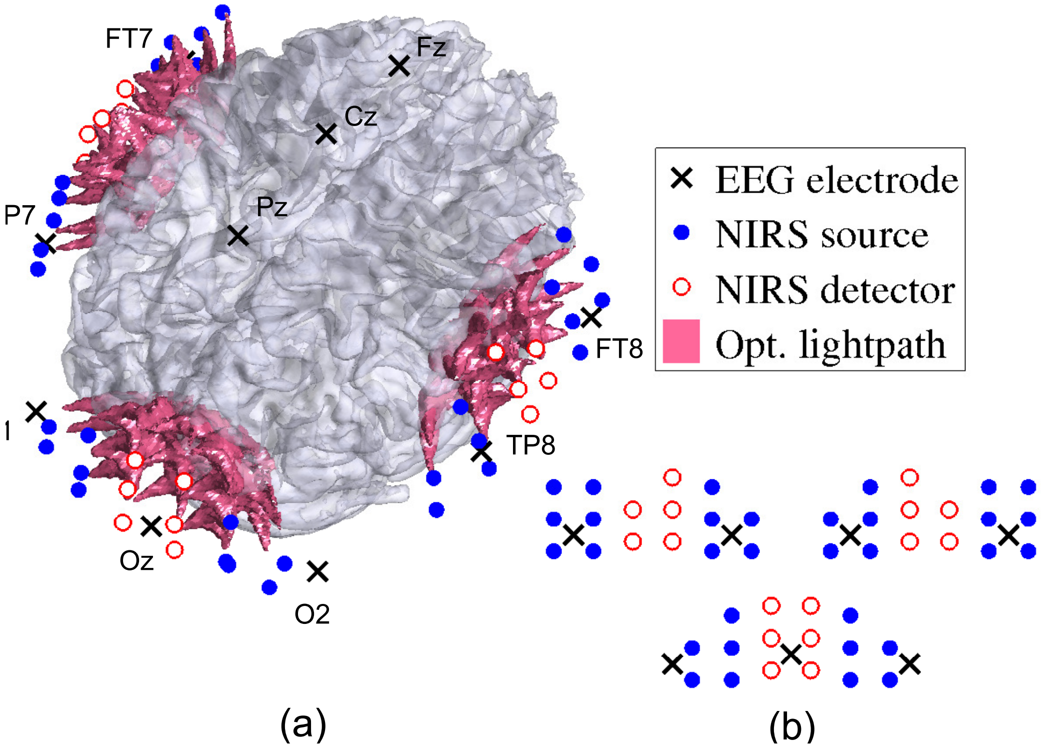


Figure 2.6 – Locations of EEG electrodes, fNIRS optrodes, and their corresponding optical lightpath. The arrangement of fNIRS sources and detectors is shown projected on the brain in subfigure (a) and as unwrapped schematic in subfigure (b) for the two auditory probes (top left and right) and the visual probe (bottom).

After the montage was positioned, the locations of fNIRS optrodes, EEG electrodes, as well as the nasion, pre-auricular points and 123 random scalp coordinates were digitized with Visor (ANT BV) and ASA 4.5 3D digitizer.

Using each participant's structural MRI, these digitized points were then coregistered, following [WMFG08], in order to have all participants' data in a common space, independent of individual brain geometry.

2.4.4 Preprocessing

The preprocessing of both fNIRS and EEG data was performed offline. Optical data included an AC, a DC, and a phase component; however, only the AC intensities were used in this study. Data from each AC channel was normalized by dividing by mean, pulse-corrected following [GC95], median filtered with a filter length of 8s, and downsampled from 19.5Hz to 1Hz. The downsampled optical density changes $\Delta\mathrm{OD}_c$ were converted to changes in concentration of oxyhemoglobin (HbO) and deoxyhemoglobin (HbR) using the modified Beer-Lambert law (MBLL) [SF04]:

$$\Delta\mathrm{OD}_c^\lambda = L_{i,j}^\lambda \mathrm{DPF}^\lambda(\epsilon_{\mathrm{HbO}}^\lambda \Delta[\mathrm{HbO}]_c + \epsilon_{\mathrm{HbR}}^\lambda \Delta[\mathrm{HbR}]_c). \tag{2.6}$$

$L_{i,j}^\lambda$ is the distance traveled by the light from source i to detector j; DPF^λ is the differential path-length factor, and $\epsilon_{\mathrm{HbO}}^\lambda$, $\epsilon_{\mathrm{HbR}}^\lambda$ are the wavelength-dependent extinction coefficients of HbO and HbR, respectively. The particular quantities within this study were based on standard parameters in the HOMER2 package, which was used for the conversion process from optical density to HbO and HbR values[HDFB09]. That is, $\mathrm{DPF}^\lambda = 6.0$ for both $\lambda_1 = 830\mathrm{nm}$ and $\lambda_2 = 690\mathrm{nm}$. Values of molar extinction coefficients ϵ^λ were taken from [Pra98]. Based on equation 2.6, hemoglobin changes were estimated by the following least-squares solution:

$$\begin{bmatrix}\Delta[\mathrm{HbO}]_c \\ \Delta[\mathrm{HbR}]_c\end{bmatrix} = (\mathbf{E}^T\mathbf{E})^{-1}\mathbf{E}^T \begin{bmatrix}\Delta\mathrm{OD}_c^{\lambda_1}/L_{i,j}^{\lambda_1}\mathrm{DPF}^{\lambda_1} \\ \Delta\mathrm{OD}_c^{\lambda_2}/L_{i,j}^{\lambda_2}\mathrm{DPF}^{\lambda_2}\end{bmatrix}, \tag{2.7}$$

where

$$\mathbf{E} = \begin{bmatrix}\epsilon_{\mathrm{HbO}}^{\lambda_1} & \epsilon_{\mathrm{HbR}}^{\lambda_1} \\ \epsilon_{\mathrm{HbO}}^{\lambda_2} & \epsilon_{\mathrm{HbR}}^{\lambda_2}\end{bmatrix}. \tag{2.8}$$

Finally, common average referencing (CAR) was applied to the converted data in order to reduce noise and artifacts that are common in all n_c channels ([AYG12]). Thereby, the mean of all channels is substracted from each individual channel c. It is performed on both $\Delta[\mathrm{HbO}]$ and $\Delta[\mathrm{HbR}]$:

$$\begin{aligned}\Delta\overline{[\mathrm{HbO}]}_c(t) &= \Delta[\mathrm{HbO}]_c(t) - \frac{1}{n_c}\sum_{i=1}^{n_c}\Delta[\mathrm{HbO}]_i(t),\\ \Delta\overline{[\mathrm{HbR}]}_c(t) &= \Delta[\mathrm{HbR}]_c(t) - \frac{1}{n_c}\sum_{i=1}^{n_c}\Delta[\mathrm{HbR}]_i(t).\end{aligned} \tag{2.9}$$

EEG data were preprocessed with EEGLAB 2013a [DM04]. First the data was bandpass filtered in the range of 0.5-48Hz using a FIR filter of standard filter order 6 ($= \frac{3}{\text{low cutoff}} \cdot$sampling rate). Then, ocular artifacts were rejected using Independent Component Analysis (ICA) as proposed by [JMW$^+$00]. In this process, all 10 channels were converted to 10 independent components. One component of each participant was manually rejected based on prefrontal eye blink artifacts. Finally, a pre-stimulus mean of 100ms was substracted from all stimulus-locked data epochs.

2.4.5 Grand Averages

We will now look at averaged EEG and fNIRS signals for the different classes to learn about systematic differences between classes. Figure 2.7 shows the haemodynamic response function (HRF) for selected channels averaged over all 12 participants for auditory stimuli (`AUD`, blue), visual stimuli (`VIS`, red), and no stimuli (`IDLE`, black). The stimulus locked data trials (blocks 2-5) are epoched by extracting the first 10s of each stimulus, and a 2s prestimulus baseline was substracted from each channel. There was a clear peak in the HRF in response to a `VIS` stimulus on channels from the occipital cortex (channels 141 and 311 in the Figure) and a return to baseline after the stimulus is over after $12.5s$. Both effects are absent for an `AUD` stimulus. Conversely, the channels from the auditory cortex in the temporal lobe (channels 30 and 133 in the Figure) react much stronger to an auditory than to a visual stimulus.

Figure 2.8 shows the first second of EEG ERP waveforms of trial types `AUD` (blue), `VIS` (red), and `IDLE` (black), averaged over all 12 participants. As expected, it shows distinct pattern for auditory and visual stimuli when comparing electrodes at the visual cortex (Oz, O1) with electrodes at more frontal positions (Fz, FT7). Note that the auditory cortex cannot be accessed by surface EEG. Regarding the frequency domain, it is widely known that frequency responses can be used to identify cognitive processes [vWSG84].

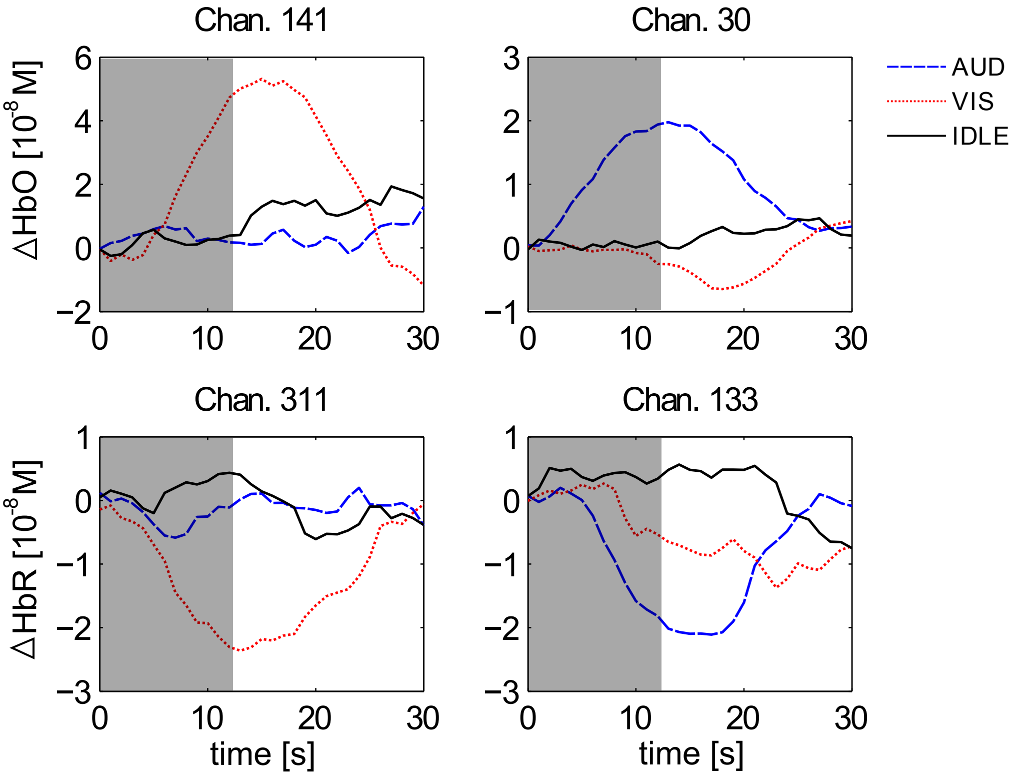


Figure 2.7 – Grand averaged HRFs of HbO (top) and HbR (bottom) for visual (left) and auditory (right) channels. Depicted are averages for the classes AUD (blue), VIS (red), and IDLE (black).

Figure 2.9 shows EEG power spectral densities on a logarithmic scale at electrodes at prefrontal cortex (Fz), occipital cortex (Cz) and auditory cortex (FT7). The figure indicates strong differences in frequency power distribution between classes. The peak in the alpha band (8-13 Hz) for the AUD trial type is expected, but unusually pronounced. We attribute this to the fact that the VIS stimuli are richer than the AUD stimuli as the visual stimulus material often contains multiple focal points of interest (e.g. occurring characters) at once. The difference between VIS and AUD trials does also not only involve perceptual processes but also other aspects of cognition, as they differ in content, processing codes and other parameters. On the one hand, this difference is partially specific to the scenario we employed. On the other hand, we argue that this difference between visual and auditory information processing pertains for most natural situations. We will further investigate the impact of this issue in Section 2.4.7.

To summarize, given the observed differences in EEG and fNIRS signals between classes, we expect to be able to successfully discriminate workload classifiers on a single-trial basis.

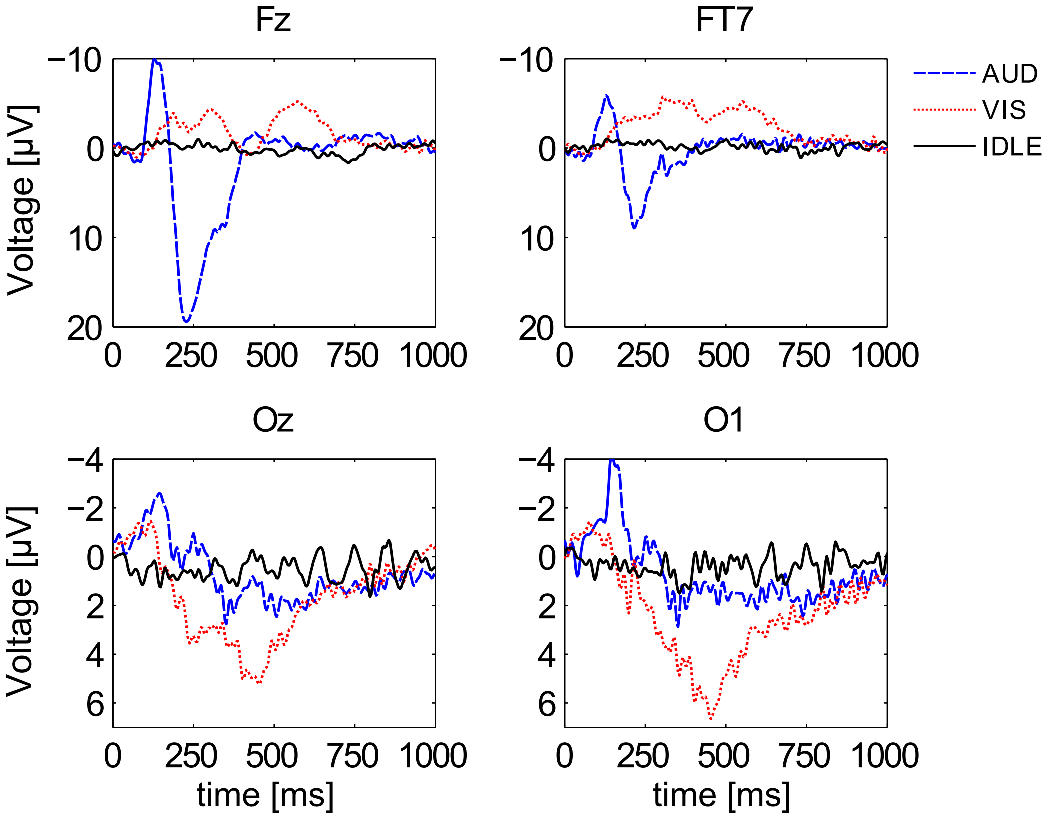


Figure 2.8 – Grand averaged ERPs at 4 different electrode positions (frontal (top) and occipital (bottom)). Depicted are averages over all 12 participants for the trial types `AUD` (blue), `VIS` (red), and `IDLE` (black).

2.4.6 Classification

In this study, we first aimed to discriminate auditory from visual perception processes. Second, we wanted to detect auditory or visual processes, i.e. distinguish modality-specific activity from no activity or other activity.

To examine the expected benefits of combining the fNIRS and EEG signals, we first explored two individual classifiers for each signal domain, before we examined their combination by estimating a meta classifier. The two individual fNIRS classifiers were based on the evoked deflection from baseline HbO (`HbO` classifier) and HbR (`HbR` classifier). The EEG classifiers were based

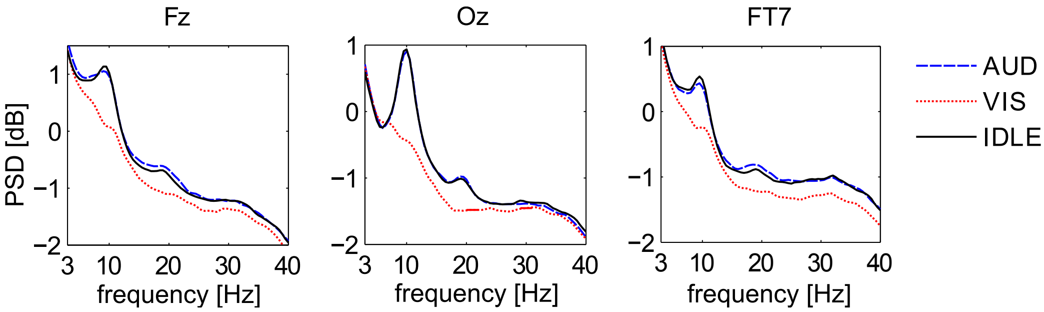


Figure 2.9 – Power Spectral Densities of EEG signals at Fz, Cz, FT7 for three different trial types `AUD` (blue), `VIS` (red), and `IDLE` (black). PSD is averaged over all 12 participants.

on induced band power changes (`POW` classifier) and the downsampled ERP waveform (`ERP` classifier).

fNIRS features: Assuming an idealized haemodynamic stimulus response, i.e. a rise in HbO (`HbO` features) and a decrease in HbR (`HbR` features), stimulus-locked fNIRS features were extracted by taking the mean of the first $\frac{w}{2}$ samples (i.e. $t_{opt} - \frac{w}{2}, \dots, t_{opt}$) subtracted from the mean of the following $\frac{w}{2}$ samples (i.e. $t_{opt}, \dots, t_{opt} + \frac{w}{2}$) in all channels c of each trial, similar to [LCW11]. Equation 2.10 illustrates how the features were calculated.

$$\begin{aligned} f_c^{\mathtt{HbO}} &= \frac{2}{w}\left(\sum_{t=t_{opt}}^{t_{opt}+\frac{w}{2}} \Delta\overline{[\mathrm{HbO}]}_c(t) - \sum_{t=t_{opt}-\frac{w}{2}}^{t_{opt}} \Delta\overline{[\mathrm{HbO}]}_c(t)\right) \\ f_c^{\mathtt{HbR}} &= \frac{2}{w}\left(\sum_{t=t_{opt}}^{t_{opt}+\frac{w}{2}} \Delta\overline{[\mathrm{HbR}]}_c(t) - \sum_{t=t_{opt}-\frac{w}{2}}^{t_{opt}} \Delta\overline{[\mathrm{HbR}]}_c(t)\right) \end{aligned} \tag{2.10}$$

EEG features: For `POW`, the entire 10 seconds of all 10 channels were transformed to the spectral domain using Welch's method, and every 1 Hz frequency bin in the range of 3-40Hz was concatenated to a 38-dimensional feature vector per channel. `ERP` features were always based on the first second (onset) of each trial. First, we applied a median filter ($k_{med} = 5 \approx 0.02$s) to the ERP waveform, followed by a moving average filter ($k_{avg} = 13 \approx 0.05$s). A final downsampling of the resulting waveform ($k_{down} = k_{avg}$) produced a 20-dimensional feature vector for each channel.

In the end, all features, i.e. `HbO`, `HbR`, `POW`, and `ERP`, were standardized to zero mean and unit standard deviation (z-normalization).

Based upon these four different feature types, four individual classifiers were trained. Then, a fifth meta-classifier was created based on the probability values for each of these individual classifier's prediction outcome. This META classifier was based on the weighted decision values $\overline{p}_m$ of each of the four individual classifiers, mapped to a probability distribution of classes. Each element in $\overline{p}_m$ characterized a probability for one particular trial type, predicted by the m-th classifier (with m being the index to the individual classifiers). The combined decision values of ourMETA classifier were given by

$$\overline{p}^{\text{meta}} = \frac{1}{4} \sum_{m=1}^{4} \overline{p}_m \cdot w_m. \tag{2.11}$$

In the following, $C = \{1, \ldots, k\}$ includes the indices of all classes, i.e. $k = |C|$ indicates the number of classes. Then, the class $c_{\text{best}} \in C$, which has the highest prediction among all four individual classifiers, was selected by the META classifier.

$$c_{\text{best}} = \arg\max_{c \in C} \ p_c^{\text{meta}}, \quad \text{where} \ \overline{p}^{\text{meta}} = \begin{bmatrix} p_1^{\text{meta}} \\ \vdots \\ p_c^{\text{meta}} \\ \vdots \\ p_k^{\text{meta}} \end{bmatrix} \tag{2.12}$$

The weights w_m were estimated based on the classification accuracy for a development set. Classification accuracy values higher than baseline (pure chance) were linearly scaled between 0 and 1, while those below baseline were discarded ($w_m = 0$). Afterwards, the weight vector $\overline{w} = [w_1, w_2, w_3, w_4]^T$ was divided by its 1-norm in order to sum all of its elements to 1.

$$\left[0, \frac{1}{k}\right] \mapsto 0, \quad \text{and} \quad \left[\frac{1}{k}, 1\right] \mapsto [0, 1] \ni w_m \tag{2.13}$$

For three classifiers (HbO, HbR, and POW), a linear discriminant analysis (LDA) classifier was employed, while a linear support vector machine (SVM) was used for the ERP classifier (using the LibSVM implementation by [CL11]). We did this because we expected the first three feature sets to be more discriminative as they are less prone to inter-trial variability (caused by outliers) and intra-trial variability (caused by the complex temporal patterns of an ERP).

2.4.7 Results

In this section, we present the evaluation results for the proposed hybrid BCI for a number of binary classification tasks. We call each classification task a *condition*. We regard two types of condition: First, a condition to discriminate visual from auditory activity (assuming that each classified trial belongs exactly to one of the two classes `VIS` or `AUD`). Second, multiple conditions to detect one type of perceptual activity, independently of other perceptual or cognitive activity (e.g. also for trials which do not contain any perceptual activity or both visual and auditory activity).

In the person-dependent case, we applied leave-one-trial-out cross-validation (resulting in 60 folds for 60 trials per subject). To estimate parameters of feature extraction and classification (t_{opt} and w from Equation 2.10 for each fold, fusion weights w_m from Equation 2.13), we performed another nested 10-fold cross-validation (i.e. 54 trials for training and 6 trials for evaluation in each fold) for the train set of each fold. The averaged accuracy in the inner cross-validation is used for parameter selection in the outer cross-validation. This procedure avoided overfitting of the parameters to the training data. In the subject-independent case, we performed leave-one-person-out cross-validation, resulting in a training set of 660 trials and a test set of 60 trials per fold.

Person-Depdendent Classification

Tables 2.13 summarizes the classification accuracy for all conditions for both the person-dependent evaluation. The first row is a discriminative task in which the 2-class classifier learned to separate visual and auditory perceptual activity. We see that for all four individual classifiers, a reliable classification is possible. EEG-based features outperform fNIRS features (`HbO`: 79.4% vs. `POW`: 93.6%). The fusion of all four classifiers (`META`) yields the best performance, significantly better (paired, one-sided t-test, $\alpha = 0.05$) than the best individual classifier by a difference of 4.4% absolute. This is in line with the results of [DK12], who found modest but consistent improvements by combining different modalities for the classification of inner states. Figure 2.10 shows a detailed breakdown of classification results over all participants for the example of `AUD` vs. `VIS`. We see that for every participant, classification performance for every feature type was above the chance level of 50% and the performance of `META` was above 80% for all participants.

	HbO	HbR	POW	ERP	META
AUD vs. VIS	79.4	74.3	93.6	93.3	**97.8***
AUD vs. IDLE	80.0	74.7	71.9	91.4	**95.6***
VIS vs. IDLE	83.8	78.1	90.7	81.9	**96.4***
allAUD vs. nonAUD	67.2	62.8	69.7	85.9	**89.0***
allVIS vs. nonVIS	68.5	64.7	91.5	81.9	**94.8***

Table 2.13 – Stimulus-locked classification accuracies (in %) for *person-dependent* classification. An asterisk in the META column indicates a significant improvement ($\alpha = 0.05$) over the best corresponding individual feature type.

	HbO	HbR	POW	ERP	META
AUD vs. VIS	70.3	65.7	84.3	90.4	**94.6***
AUD vs. IDLE	64.0	61.9	66.1	84.2	**86.9***
VIS vs. IDLE	72.2	69.0	82.5	75.3	**89.9***
allAUD vs. nonAUD	60.6	58.8	41.7	**85.6**	84.7
allVIS vs. nonVIS	62.7	62.0	84.2	73.1	**86.7**

Table 2.14 – Stimulus-locked classification accuracies (in %) for *person-independent* classification. An asterisk in the META column indicates a significant improvement ($\alpha = 0.05$) over the best corresponding individual feature type.

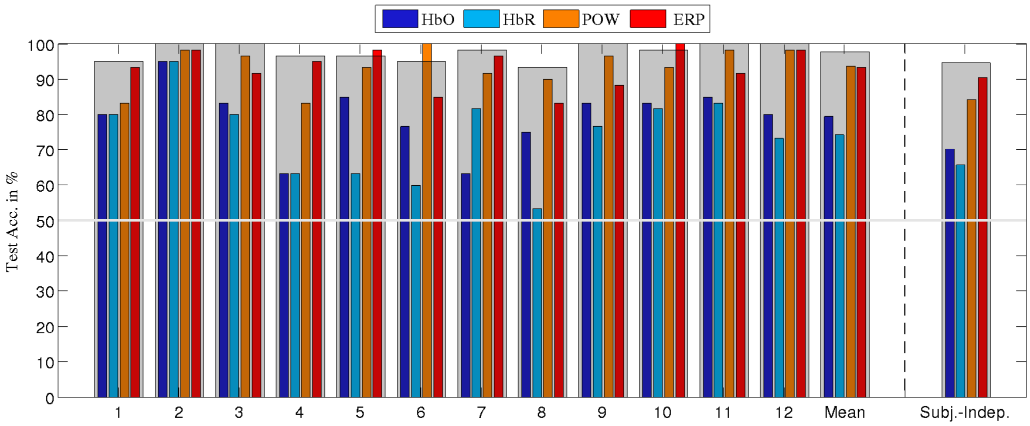


Figure 2.10 – Stimulus-locked classification accuracies of AUD vs. VIS for person-dependent, as well as for person-independent classification. Recognition rates of the META classifier are indicated by a grey rectangular overlay over the individual classifiers' bars.

Person-Independent Classification

In the next step, we evaluated person-independent classification on the same conditions as for the person-dependent evaluation. The results for this evaluation are summarized in Table 2.14. Averaged over all conditions, classification accuracy degrades by 6.5% compared to the person-dependent results. This is mostly due to a higher signal variance caused by individual differences. Still, we managed to achieve robust results for all conditions, i.e. person-independent discrimination of visual and auditory processes is feasible. The remaining analyses will be reported for the person-independent systems only as those are more preferable for HCI.

trained on...	evaluated on...	HbO	HbR	POW	ERP	META
AUD vs. IDLE	MIX	67%	63%	47%	88%	88%
VIS vs. IDLE	MIX	69%	68%	69%	84%	77%
AUD vs. IDLE	VIS	66%	66%	52%	48%	48%
VIS vs. IDLE	AUD	59%	61%	49%	50%	48%

Table 2.15 – Subject-independent classification results of classifiers for AUD vs. IDLE and VIS vs. IDLE, evaluated on different trials from outside the respective training set.

Detection of Perceptual Modality

The AUD vs. VIS condition denotes a discrimininination task, i.e. to classify a given stimulus as either auditory or visual. However, for an HCI application, auditory and visual perception are not mutually exclusive, i.e. they may either occur at the same time or be both inactive in idle situations. We therefore need to define conditions which train a detector for specific perceptual activity, independently of the presence or absence of other perceptual activity. Our first approach towards such a detector for either auditory or visual perceptual activity is to define the AUD vs. IDLE and the VIS vs. IDLE conditions. A classifier trained on these conditions should be able to identify neural activity induced by the specific perceptual modality. In Table 2.13, we see that those conditions can be discriminated with high accuracy of 95.6% (person-independent) and 96.4% (person-dependent), respectively. To test whether this neural activity can still be detected in the presence of other perceptual processes, we evaluate the classifiers trained on those conditions also on MIX trials, which combine perception of auditory and visual stimuli. The top two rows of Table 2.15 summarize the results for the person-independent case and show that perceptual activity is correctly classified in most cases by the META classifier.

For the AUD vs. IDLE and the VIS vs. IDLE conditions, it is not clear if a detector trained on them has actually detected visual or auditory activities or rather general cognitive activity which was present in both the AUD or VIS trials, but not in the IDLE trials. To analyze this possibility, we trained a classifier on the AUD vs. IDLE condition and evaluated it on VIS trials (and accordingly for VIS vs. IDLE on AUD trials). We present the results in the bottom two rows of Table 2.15. The classifiers yield very inconsistent results and 'detect' the modality-specific activity in non-matching trials in nearly half of the cases. They do not exceed the chance level of 50%. To train a classifier which is more sensitive for the modality-specific neural characteristics, we need to include non-IDLE trials in the training data as negative examples. For this purpose, we define the condition allAUD vs. nonAUD, where the allAUD class is defined as allAUD = {AUD, MIX} and the nonAD is defined as nonAUD = {IDLE, VIS}. allAUD contains all data with auditory processing (but potentially not exclusively), while nonAUD contains all data without auditory processing (but potentially with other activity). The condition allVIS vs. nonVIS is defined respectively. The bottom two rows of Tables 2.13 and 2.14 document that the trained classifiers can robustly detect modality-specific perceptual activity for both conditions. In the former case of the AUD vs. IDLE and the VIS vs. IDLE conditions, we showed that the classifier only learned to separate general activity from a resting state. If this was also the case for the allAUD vs. nonAUD and the allVIS vs. nonVIS conditions, we would expect a classification accuracy of 75% or less (for example, in the allVIS vs. nonVIS condition, we would expect 100% accuracy for the VIS, MIX and IDLE trials, and 0% accuracy for the AUD trials). This baseline is outperformed by our classifiers for detection. This result indicates that we were indeed able to detect specific perceptual activity, even in the presence of other perceptual processes. This result shows that the new detectors did not only learn to separate general activity from a resting state (as did the detectors defined earlier). If that would have been the case, we would have seen a classification accuracy of 75% or less: For example, if we make this assumption in the allVIS vs. nonVIS condition, we would expect 100% accuracy for the VIS, MIX and IDLE trials, and 0% accuracy for the AUD trials, which would be incorrectly classified as they contain general activity but none which is specific to visual processing. This baseline of 75% is outperformed by our classifiers for detection. This result indicates that we were indeed able to detect specific perceptual activity, even in the presence of other perceptual processes. For additional evidence, we look at how often the original labels (AUD, VIS, IDLE, MIX) were classified correctly in the two new detection setups by the META classifier. The results are summarized in Table 2.16 as a confusion matrix. We see that all classes are correctly classified in more than 75% of all cases, indicating that we detected the modality-specific characteristics in contrast to general cognitive activity.

	AUD	VIS	IDLE	MIX
allAUD vs. nonAUD	91.1	85.3	85.0	77.2
allVIS vs. nonVIS	81.9	94.2	82.2	88.3

Table 2.16 – Person-independent classification accuracy (in %) of the META classifier for the allAUD vs. nonAUD and the allVIS vs. nonVIS conditions, broken down by original labels.

Comparison of Signal Types

The results in Tables 2.13 and 2.14 indicate that fusion (META) was useful to achieve a high classification accuracy. There was a remarkable difference between the results achieved by the classifiers using fNIRS features and by classifiers using EEG features. This holds across all investigated conditions and for both person-dependent and person-independent classification. We suspect that the advantage of the META classifier was mostly due to the combination of the two EEG based classifiers. We investigated this question by comparing two intermediate fusion classifiers. Those combined only the two fNIRS features (fNIRS-META) or the two EEG features (EEG-META), respectively. The results are given in Figure 2.11. The results show that for the majority of the conditions, the EEG-META classifier performed as good as or even better than the overall META classifier. In contrast, the fNIRS features contributed significantly to the classification accuracy for the conditions AUD vs. IDLE and VIS vs. IDLE. To exclude that the difference was due to the specific fNIRS feature under-performing in this evaluation, we repeated the analysis with other fNIRS features (average amplitude, value of largest amplitude increase or decrease). We did not achieve improvements compared to the evaluated feature. Overall, we see that fNIRS-based features were outperformed by the combination of EEG based features on the investigated task but could still contribute to a high recognition accuracy in some of the cases.

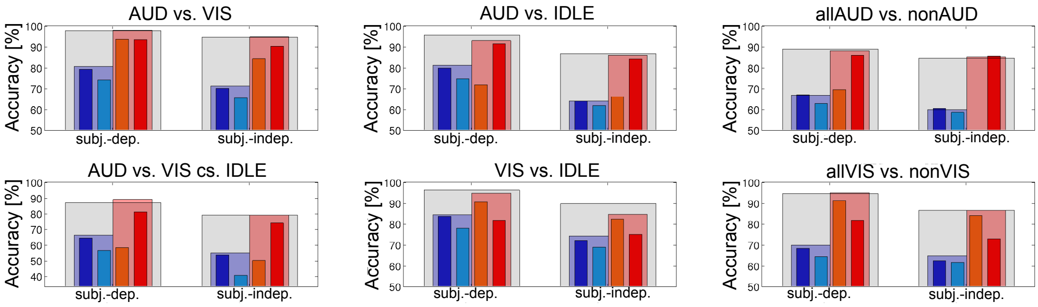


Figure 2.11 – `fNIRS-META` (red) vs. `EEG-META` (blue) evaluated for both person-dependent and person-independent classification for different conditions.

Impact of EEG Frequency Band Selection

There are however some caveats to the dominance of EEG features. First, the `ERP` classifier is the only one of the four classifiers which uses features that highly dependent on temporal alignment to the stimulus onset. Therefore, it is not suited for continuous classification. Second, concerning the `POW` classifier, we see in Figure 2.9 a large difference in alpha power between `VIS` on the one hand and `AUD` and `IDLE` on the other hand. As we cannot completely rule out that this effect is caused at least in parts by the experimental design or participant selection (e.g. trained participants which can unusually quickly enter a resting state when no stimulus is presented), we need to verify that the discrimination ability of the `POW` classifier does not solely depend on differences in alpha power. For that purpose, we repeated the evaluation of `AUD` vs. `VIS` with different sets of filters, of which some excluded the alpha band completely. Results are summarized in Figure 2.12. We see that as expected, feature sets including the alpha band performed best. Accuracy dropped by a maximum of 9.4% relative when removing the alpha band (for the participant dependent evaluation from 1-40Hz to 13-40Hz). This indicates the upper frequency bands still contain useful discriminating information, but brings the EEG based results closer to the results of the fNIRS-based features.

2.4.8 Discussion

The results from the previous section indicate that both the discrimination and detection of modality-specific perceptual processes in the brain is feasible. This holds for both the person-dependent as well as a person-independent case with high recognition accuracy. We see that the fusion of

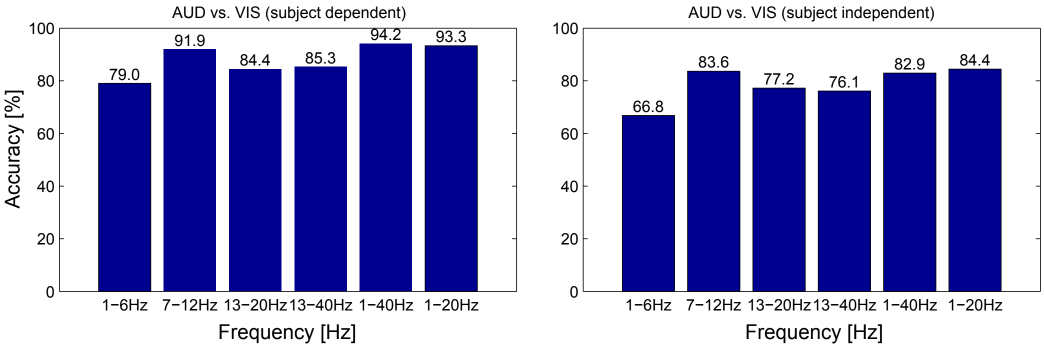


Figure 2.12 – Classification accuracy for different filter boundaries for the POW feature set, evaluated for both person-dependent (left) and person-independent (right) classification of AUD vs. VIS.

multiple features from different signal types led to significant improvement in classification accuracy. However, in general fNIRS-based features were outperformed by features based on the EEG signal. One difference between fNIRS and EEG signals is the fact that the fNIRS signals may still contain artifacts which we did not account for (like we did for the EEG signal by ICA). Artifact removal techniques for fNIRS have been applied with some success in other research on fNIRS BCIs [MD12]. Another difference is that the coverage of fNIRS optodes was limited mainly to the sensory areas, but the EEG measures may include activity generated from other brain regions, such as the frontal-parietal network. Activities in these regions may be reflecting higher cognitive processes triggered by the different modalities, other than purely perceptual ones. It may be worthwhile to extend the fNIRS condition to include those regions as well. Still, we already saw that fNIRS features can contribute significantly to certain classification tasks.

While evaluation on stimulus-locked data allows a very controlled evaluation process and is supported by very high accuracy, this condition is not very realistic for most HCI applications. In many cases, stimuli will continue over longer periods of time. Features like the ERP feature explicitly model the onset of a perceptual process but will not provide useful information for ongoing processes.

Following the general guidelines of [Fai09], we identify one limitation in validity of the present study is the fact that there may be other confounding variables that can explain the differences in the observed neurological responses to the stimuli of different modalities. Participants were following the same task for all types of stimuli; still, factors like different memory load or increased need for attention management due to multiple parallel stimuli for visual trials may contribute to the separability of the classes. We address this partially by identifying the expected effects, for example in Figure 2.7 comparing fNIRS signals from visual and auditory cortex. Also the fact that detection of both visual and auditory processing worked on `MIX` trials shows that the learned patterns were not only present in the dedicated data segments but were to some extend generalizable. Still, we require additional experiments with different tasks and other conditions to reveal whether it is possible to train a fully generalizable detector and discriminator for perceptual processes.

2.5 Recognition of Confusion

The last two previous sections dealt with empirical cognitive models for workload. As motivated in Section 1.2.2, another important user state relevant for HCI applications is confusion. We define confusion as the cognitive state resulting from erroneous system feedback presented to the user. This situation typically occurs when the user's input is not correctly recognized. Incorrect recognition of user input is a regular event for interaction systems which use automatic speech recognition, active BCIs for computer control or gesture recognizers. Users do not directly perceive that their input is incorrectly recognized. Instead, they indirectly notice erroneous system behavior when the system feedback to their input (e.g. the selection of a menu entry, the execution of a certain function, etc.) does not match their expectations. As a result, the users are confused by such system behavior.

In this section, we describe the development of an empirical cognitive model for the user state confusion, using EEG. While EEG-based person-dependent models for confusion detection already exist (see Section 2.2.4), this work contributes to the research community the development of a person-adapted classifier for confusion, trained and evaluated on a simple calibration task of simulated BCI operation. Another contribution to the research community is that we show that this classifier transfers to data recorded in a realistic, gesture-based HCI scenario.

2.5.1 Error Potentials & Confusion

We base our implementation of a model for confusion on the detection of error potentials in the EEG signal. An (interaction) error potential (ErrP) is a type of Event Related Potential, i.e. a characteristic pattern of brain activity triggered by a specific type of stimulus. An ErrP occurs after an erroneous behavior of an observed agent, which can be another person or a technical system. In HCI, an ErrP is triggered by the perception of unexpected and erroneous system feedback to the user's input. The concept of an ErrP is related to the concept of Error Related Negativity/Positivity (*ERN*/*Pe*): The ERN and Pe are a low-latency reactions to an error of the acting person him- or herself. A variant of the ERN, the feedback-related Error Negativity (fERN) is a reaction to external feedback signaling an error of the person him- or herself. In contrast to both ERN and fERN, an interaction error potential (ErrP) is related to errors of another agent. One major difference between the different types of error signals manifests in the latency of the resulting signals. While an ERN occurs 80 ms after an error of the person, a typical ErrP can be measured at front-central electrode positions and occurs in a window of about 150 ms to 600 ms after the error of the observed agent. This difference in latency is because an ErrP can only occur after an external stimulus has been perceived, while an ERN results from internal processes only. The most pronounced components of an ErrP are a negative peak around 250 ms and a positive peak around 350 ms [FJ08]. The exact contour and latency of an ErrP may vary with tasks and individual participants [ICM^{+}12]. Figure 2.15 shows a typical ErrP pattern as difference between brain activity following error-free (noErrP) and erroneous feedback (ErrP).

The task of modeling the user state "confusion" from ErrPs in the EEG signal can be defined as a two-class classification problem: A given EEG data segment either contains an error potential following erroneous feedback (i.e. it belongs to the *ErrP* class) or not (i.e. it belongs to the *noErrP* class). One fact we can exploit for the classification of error potentials is that in many situations, it is clear at which point in time exactly we can expect an error potential to occur. This is for example the case for the common scenario of graphical system output where feedback to user's input occurs instantaneously. This is beneficial for the ErrP detection, because it reduces the number of segments we need to inspect for ErrPs (and thus decreases the false alarm rate). Another benefit is that a tight alignment of the inspected data segment to the stimulus which potentially triggers an ErrP reduces the variability of signal patterns in the analyzed data segments.

2.5.2 ErrP Elicitation Paradigms

To collect data for training and evaluation of an empirical confusion model, we need a procedure to elicit the user state confusion. We differentiate two paradigms to induce confusion in an HCI context. In both cases, we record data during a live interaction session of a user with a system, during which erroneous feedback to user's input occurs.

In the first elicitation paradigm, we manipulate the occurring errors with a Wizard-of-Oz setup: The user is not actually operating the system by his or her input commands, but the outcome of each input command is predefined (i.e. whether the system generates correct or erroneous feedback to a user command). We call this paradigm *Wizard-ErrP*. In the second paradigm, the user is operating a fully functional automatic system, which elicits ErrPs because of actual input recognition errors. We call this paradigm *Automatic-ErrP*.

The advantage of the Wizard-ErrP paradigm is that we have full control over the balance between ErrP and noErrP trials for every participant. Another benefit of Wizard-ErrP is that it can be applied even when a fully automatic input recognizer is not available (e.g. during early stages of development). However, we may experience label noise (i.e. trials which are not labeled correctly) with Wizard-ErrP if the user notices the missing relation between his or her performance and the task outcome. We can also not exploit information from the user's original input for the detection or handling of confusion in the Wizard-ErrP paradigm, as the input is not actually related to feedback of the system. The Automatic-ErrP paradigm, i.e. the unconstrained operation of a fully functional user interface, does circumvent those problems: Whenever the input recognizer misinterprets the user's input, we can label the following EEG segment as belonging to the ErrP class, without the risk of adding label noise[5]. The Automatic-ErrP paradigm generates more natural reactions to erroneous feedback. When using this paradigm, it is possible to exploit information from the user's input which lead to an error, for example to check whether the user's input was untypical or whether the confidence of the input classifier was low. A drawback of the Automatic-ErrP is that it requires a functional user interface. If the error rate of the input recognizer is too high, too low, or too variable for different users, the generated ErrP corpus will be very unbalanced regarding ErrP and noErrP class labels.

In the following, we will use data recorded from both paradigms. In both cases, it is important to not mix error potentials occurring after a simulated or real system error with error potentials occurring due to a mistake of the user. Therefore, the task must be designed in way that user errors do not occur or are clearly distinguishable from system errors. As we are investigating EEG

[5] This assumes that erroneous feedback is always recognized as such by the user. If not, label noise will rise for both elicitation paradigms.

signals prone to various ocular and muscular artifacts, we should also design experiments in a way that minimize the impact of systematic, class-specific artifacts.

2.5.3 Experiment Design

To collect data for our research on person-adapted ErrP classification, we conducted two studies with different experiment setups.

For the first study (which we call *BCI-Error*), we designed a simple experiment which also functions as a calibration task for new applications. This experiment follows the Wizard-ErrP paradigm to elicit confusion. In order to evoke ErrPs generated by unexpected behavior of an interactive system, we developed a BCI mockup similar to the one presented in [FJ08]. A screen showed two numbers in one line. Participants where told to operate a BCI to select the larger of the two numbers. For this purpose, participants were asked to choose two mental images to represent the concepts 'left' and 'right' and they reproduced those images to select the target number. Numbers were presented at a distance of 1 cm such that participants had both numbers in the visual focus without producing ocular artifacts. Participants pressed a button when they concentrated on the command for the number they wanted to select. Then, they were presented with a predefined feedback (a circle around the selected number), corresponding to an simulated error rate of 30%. Before feedback, we inserted a pause of one second for motor activation to decay. Participants were given the impression that they were operating a working BCI but were also made aware of the fact that due to signal noise, they should expect a certain number of errors. The task was chosen to be simple enough to make sure that for every trial, the participant actually expected the correct response, i.e. we can regard each simulated error to actually induce an ErrP. After the experiments, participants were debriefed about the true nature of the experiment and the operated BCI.

In the second experiment (which we call *Gest-Error*), errors were not simulated but resulted from the recognition error of an automatic system (i.e. we followed the Automatic-ErrP paradigm). Here, participants were operating an automatic gesture recognizer which detected a number of pointing gestures to select one out of six options displayed on a screen. The option which was recognized by the gesture recognizer as selected was highlighted visually to give the user feedback to their input. The timing of the task was such that feedback was presented when the participant was in a resting position, i.e. not generating motion artifacts. Feedback was presented in form of a pictogram symbolizing the selected option. It was presented on a fixed position (to avoid systematic eye movement artifacts) as a large, high-contrast overlay. Participants were instructed to pay attention to the feedback with the motivation to improve gesture recognition accuracy. This procedure was chosen to increase the likelihood and effect size of an ErrP. We deliberately did not tune the gesture recognizer for maximum recognition accuracy (average of 61%), to achieve an acceptable balance of ErrP/noErrP classes. The system thus generated erroneous feedback with an average probability of 39%. The recording setup for the Gest-Error experiment is described in more detail in Section 4.6, as it was also part of a user study on a self-correcting gesture interface.

2.5.4 Data Collection

EEG was recorded at 500 Hz using a BrainVision actiCHamp system with 32 active electrodes, of which 23 were used for evaluation[6]: Fz, F3, F7, FC5, FC1, C3, T7, CP5, CP1, P3, P7, O1, O2, P4, CP6, CP2, Cz, C4, T8, FC6, FC2, F4, F8. Impedance was kept below $16k\Omega$ for all electrodes. Pz was used as reference electrode and an additional light sensor attached to the stimulus presentation screen was used for synchronization.

Using this setup, we recorded data from 20 participants (one session each) in the BCI-Error study. Participants were university students from several different departments and had no previous experience with BCIs. The data set was balanced for gender. Participants completed a varying number of trials but never less than 150 plus a small number of training trials. We decided for this experiment to keep the total recording time around five to ten minutes. On average, the pure task time, which varies as the task is self-paced, was less than nine minutes, i.e. it took participants less than four seconds for one trial. With this short duration, the task can also function as a calibration task to collect ErrP data of a person for adapting an existing confusion model to that person.

For the Gest-Error experiment, we collected data from 11 participants (one session each), using the same recording setup as for the BCI-Error experiment. All participants were university students (4 female, 7 male). For each participant, we collected 72 or 144 trials. The average gesture recognition accuracy (i.e. the relative frequency of the noErrP class) was 61%. Table 2.17

[6] As different electrode montages were used, not all of them were available for all sessions

summarizes the most important information about the two collected data sets.

	Experiment I	**Experiment II**
Name	BCI-Error	BCI-Gest
Paradigm of ErrP elicitation	Wizard-ErrP	Automatic-ErrP
Number of participants	20	11
Num of trials per participant	150–200	72–144
Relative freq. of ErrP class	30%	39%

Table 2.17 – Summary of our ErrP experiments and the collected data sets.

2.5.5 ErrP Classifier

In this subsection, we will describe the components of our person-adapted ErrP classifier. We will use the same classifier for data from both, the BCI-Error and the Gest-Error data set.

To extract the classification trials, segments of 500 ms succeeding the feedback stimulus were extracted. We assigned labels *ErrP* and *noErrP* based on whether correct or incorrect feedback was given. Using stimulus-locked data extraction is not a limitation for ErrP classification, as in many application scenarios, the event which triggers an ErrP can be clearly located in time (e.g. the display of a new screen on a graphical user interface). For preprocessing, data was re-referenced to a common average. Each trial was then detrended and normalized by subtracting the mean of 200 ms before stimulus. To extract features, the data of the channels Fz and Cz was subsampled, averaging over a window of 50 ms length with a shift of 25 ms. Features from both channels were concatenated, which resulted in a feature vector with 28 dimensions[7].

Most ErrP classification systems in the literature refrain from the use of artifact filters (e.g. [FJ08, VS12]) or do not see improvements when using them (e.g. [SBK$^+$12]). We propose the moderate use of artifact removal techniques when dealing with EEG data from different participants, as individual eye-movement artifacts hamper the classifier's ability to extract generalizable EEG patterns. To remove the most critical artifacts, we performed an Independent Component Analysis (ICA) using the AMICA [PMDR08] algorithm.

[7] Note that while features where only calculated from two electrodes, the remaining electrodes where still used for artifact removal.

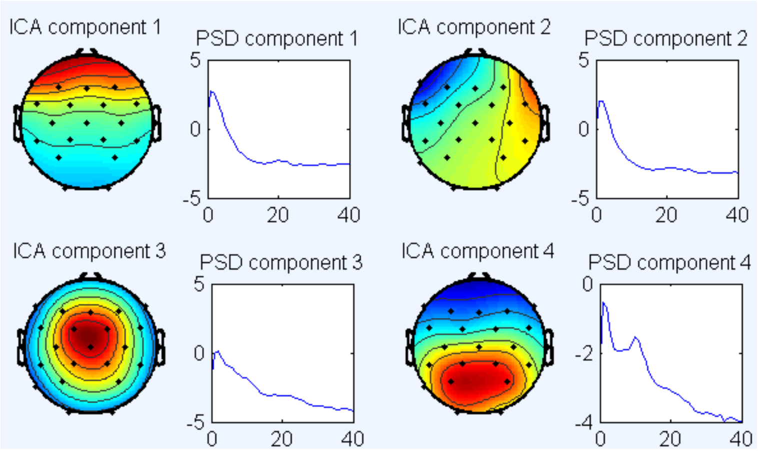


Figure 2.13 – Topographic maps of the removed ICA components. Components 1 and 2 are dominated by occular artifacts. Components 3 and 4 mainly contain brain activity.

Two components with strong frontal activity and very smooth power spectra were automatically removed. Which components were removed was selected by minimizing cosine distance of the corresponding spatial filters to a prototype filter generated from another data set of a similar ErrP task. ICA was computed on all ErrP trials from all available training sessions. This method allowed us to perform fast and fully automatic artifact correction. All computationally expensive steps can be performed offline before classification and all remaining operations for feature extraction and classification are of low complexity, which makes this approach feasible for online classification.

Figure 2.13 shows the topographic map of the removed components (components 1 and 2 in the figure). For comparison, we also show two components which mainly contain brain activity (components 3 and 4 in the figure). The effect of the component removal is notable: Figure 2.14 and 2.15 show Grand Averages of the ErrP and the noErrP class at Fz calculated before and after artifact removal. We see that the averaged signal after artifact removal clearly resembles a characteristic ErrP pattern as published in [SBK$^+$12] or [FJ08] while the original pattern before correction contains strong artifacts superimposed on the brain activity. We will later see how this difference influenced classification performance.

For classification, we looked at both person-dependent and person-adapted approaches. As classification model, a Support Vector Machine with Radial Basis Function Kernel was employed, using the SVMLight implementation of

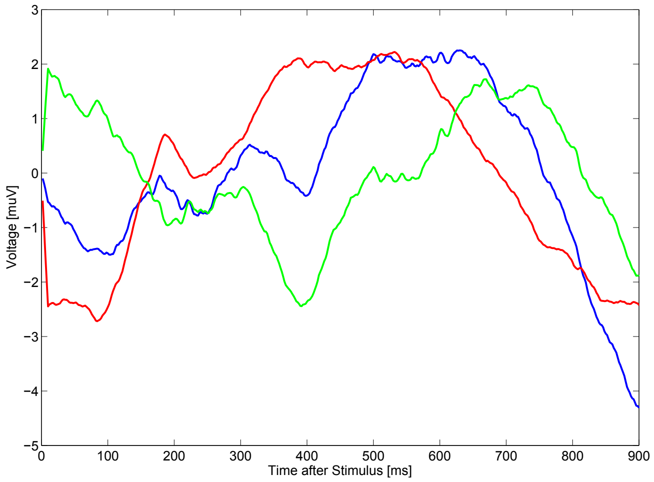


Figure 2.14 – Grand average (red: noErrP, blue: ErrP, green: error-minus-correct) at Fz (raw data)

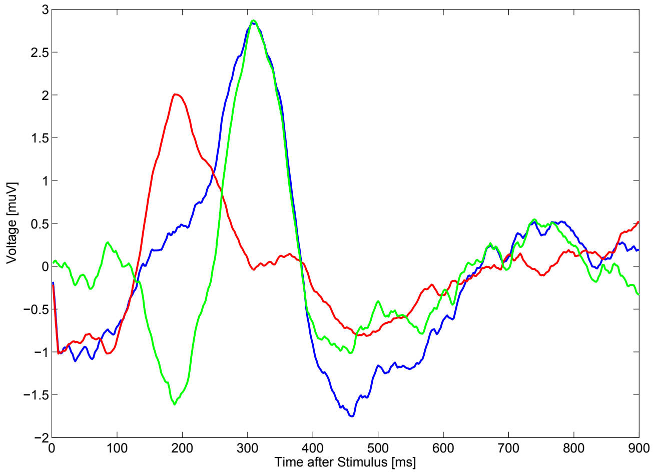


Figure 2.15 – Grand average (red: noErrP, blue: ErrP, green: error-minus-correct) at Fz (after ICA filtering)

the Shogun Toolbox[8]. Kernel parameters were fixed at $C = 1$ and $\gamma = 5$ ($\gamma = 25$ for the person-dependent system, to account for smaller training corpus size). Training data was balanced (i.e. the noErrP class was undersampled by randomly removing trials) to not bias the classifier towards the majority class. The training data for the model always comprised calibration trials of the test participant. For the presented evaluation (see Subsection 2.5.6),

[8] http://www.shogun-toolbox.org

those calibration trials were a random subset of the available session of the test participant.

A person-adapted classifier tries to reduce the number of required calibration trials (compared to a person-dependent classifier) per participant by combining calibration trials by the test participant with training data from other participants. While providing additional training data is beneficial in general, simply adding all available sessions to the training data set can compromise classification accuracy. This is because some of those sessions might contain data of sub-optimal recording quality or data from participants with very different ErrP characteristics compared to the test participant. To identify the most relevant training sessions for the given testing session, we calculated the Bhattacharyya distance [Kai67] between the available calibration features of the test participant and features from each training session. For training, we then only used the 20% closest sessions plus the provided calibration data of the test person. Whenever data sets from different participants were combined, data for each participant was z-normalized individually to make feature vectors commensurable.

2.5.6 Evaluation

In this section, we will evaluate different configurations of the developed ErrP classifier on the two collected data corpora, BCI-Error and Gest-Error.

We first evaluated the classifier on the BCI-Error corpus. Evaluation was performed in a leave-one-participant-out cross-validation. As calibration data was selected randomly, performance metrics for each configuration are averaged across ten evaluation runs. The reported quality metric is the F-score for the ErrP class if not noted otherwise.

We compare results for four different training configurations of the classifier: `BASE` was the baseline person-adapted configuration, for which all available training sessions were concatenated together with the available calibration data of the test participant. `ICA` used the same training data as `BASE` but additionally used the ICA-based artifact correction. `DIST` was a person-adapted configuration which selected training sessions in addition to the calibration data as described in the previous section. `PD` was a person-dependent configuration trained only on the available calibration data without data from other participants. `DIST` and `PD` both apply ICA. Table 2.18 summarizes the different classifier configurations.

Configuration	ICA?	Use of Calibration & Training Data
`PD`	yes	only calibration
`BASE`	no	calibration + all training
`ICA`	yes	calibration + all training
`DIST`	yes	calibration + training similar to calibration

Table 2.18 – Person-dependent and person adapted ErrP classifier setups. The term "training data" refers to data which is from participants other than the test participant.

Effect of Artifact Removal and Calibration Set Size

Figure 2.16 shows classification performance over the number of provided calibration data. The first observation we make is that that removal of ocular artifacts consistently has a positive effect (an improvement of up to 7.7% relative of `ICA` compared to `BASE`) across all conditions. This indicates that the classifier does not rely on such artifacts for achieving a high accuracy. In the contrary, removal of such artifact actually helps the classifier to extract generalizing EEG patterns.

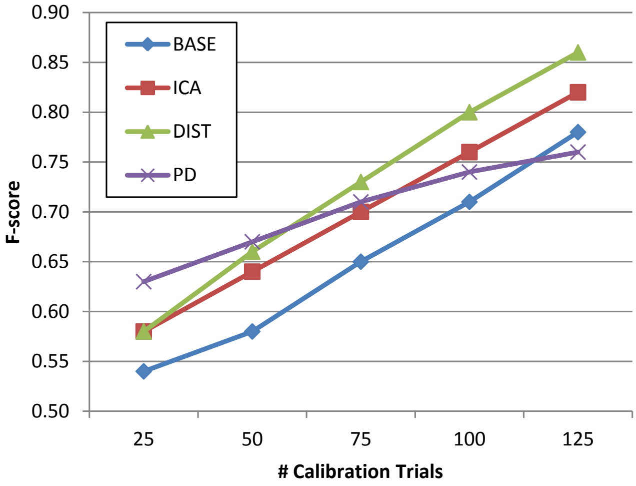


Figure 2.16 – F-scores for different modes and calibration set sizes.

Furthermore, we see that adding more calibration trials from the test participant consistently improved recognition accuracy. What is the best method of exploiting available training data and calibration data depended on the amount of calibration data available: If very little training data was available, the system performed best with a completely person-dependent training set consisting of only the calibration data (`PD`). We postulate that this happened because when the calibration data set is very small, complementing it with a much larger number of trials which do not all fit the current test participant had a detrimental effect on the classification performance. For a small calibration set, we were not able to reliably sort out the relevant data using the `DIST` configuration.

When the number of calibration trials increased, the superiority of the `PD` configuration disappeared: While the performance of the `PD` mode improved with a larger calibration set as expected, the slope of the curve is smaller than the ones for `ICA` configuration and `DIST` configuration, which both improved drastically as more calibration data became available. With 75 calibration trials, the `DIST` configuration already lead to better classification performance than `PD`. The `DIST` configuration was also up to 5.2% relative better than `ICA` configuration, from an already very good starting point[9]. This shows that the selection of suitable training sessions improved performance compared to the naive person-adapted method which uses all available training sessions. Paired t-tests on the results of the individual evaluation folds showeds that performance differences between `ICA` and `DIST` for calibration set sizes of 75, 100 and 125 were significant at a level of $\alpha = 0.05$. The same result holds for the difference between `ICA` and `BASE`.

Analysis of Training Data Selection

To investigate whether features from other participants actually could be transferred to the test participant, we calculated an all-pair evaluation where for each possible pair of test participant and training participant, we build a classifier from data of the latter and evaluate it on data of the former. When maximizing F-score for each testing session across all such classifiers, we achieved an average performance of 0.64 which is on par with a person-dependent system with 25 calibration trials (which is a fair comparison, as in the all-pair evaluation, the training set always consisted of only one participant). When relating classification performance in this all-pair evaluation with the calculated Bhattacharyya distances between the respective testing and training feature sets, we got a modest but significant negative Pearson correlation ($r = -0.36$, $p < 0.001$), i.e. a lower distance to the training data implied a higher classification performance. Both results show that selection of training data is both required and possible.

[9] A post-hoc optimization run indicates that those differences do not depend on the chosen SVM parameters.

Precision–Recall Trade-off

For the best system using 125 calibration trials in the DIST mode, we achieved an F-score of 0.86, which corresponds to a recognition accuracy of 0.92, a precision of 0.86 and a recall of 0.88. Those values indicate a high robustness and a good balance between false positives and false negatives. If desired for a certain application (e.g. if false positives are considered very disruptive), precision can be increased to up 0.96 by relaxing the balancing of the training data to a ratio of 1 : 1.5. Doing so resulted in a drop of recall to 0.76.

Across all participants, the standard deviation for precision and recall was always below 0.09, i.e. while there were participants for which the system performed better than for others, even the worst performance is still in the range which was deemed useful for Human Computer Interaction by [VS12].

Task Transfer

To understand how the ErrP classifier generalized to more realistic HCI applications, we investigated how the model trained on the BCI-Error corpus transferred to data from a realistic HCI scenario, recorded as part of the Gest-Error corpus. As [ICM$^+$12] showed, transfer between different tasks may result in performance degradation as characteristics of the ErrP patterns, for example onset latency, change. We wanted to investigate whether the procedure of building person-adapted models also worked when combined with task transfer. For this purpose, we evaluated the DIST configuration of the ErrP classifier with all sessions from the BCI-Error corpus forming the available training data (from which sessions were selected) and the sessions from the Gest-Error corpus forming the testing and calibration data. As the Gest-Error corpus contained fewer trials than the BCI-Error corpus, we could only spare 60 calibration trials. Still, we achieved an accuracy of 78.3% (significantly above the baseline of 61%), corresponding to an F-score of 0.69, a precision of 0.72 and a recall of 0.67. This result shows that recognition of ErrPs in the presented real-life gesture scenario is feasible with limited calibration time.

2.5.7 Discussion

In this section, we showed that person-adapted classification of ErrPs is feasible. Artifact correction and training data selection by minimizing distance to calibration data from the test participant proved to be key factors for robust classification. We further showed that person-adapted ErrP classification possible even when transferring models between tasks. This result indicates that the model can be transferred to other error-prone input modalities (e.g. automatic speech recognition), as long as the system provides an immediate feedback mechanism. This result is a novel contribution to the research community. We conclude that EEG-based classification of ErrPs can be used to provide an empirical confusion model with high accuracy for the application in adaptive cognitive interaction systems.

One limitation of the described approach is that it is relying on time-locked evaluation of EEG data relative to feedback presentation. This works well for situations were the system can give instantaneous feedback in an unambiguous way, for example global feedback on a graphical user interface. Time-locked evaluation becomes more challenging when feedback is given more locally (e.g. in a small window instead of a global overlay), is not immediately obviously erroneous (e.g. because correct and incorrect feedback are very similar) or is extended across longer periods of time (e.g. because feedback is given by synthesized speech). In such cases, the ErrP classification would have to rely on additional information sources (e.g. eye tracking to determine when a presented feedback was perceived) or become less relying on temporal alignment (e.g. by using methods in [MRM13]).

2.6 Discussion

In this chapter, we presented empirical cognitive models for the detection of three different user states: workload level, workload type and confusion. For each user state, we provided a model using physiological sensors and performed extensive evaluations to validate the ability of the models to detect the user states in different conditions. While the investigated user states had different characteristics, there are some strong similarities between the different models: In all three sections of this chapter, we looked at person-independent or person-adapted empirical cognitive models to reduce the setup time of the model compared to a person-dependent approach. Additionally, we performed data recordings in uncontrolled environments (e.g. in the car or while performing gestures), and with realistic stimulus material (pictures, videos). Both, reduced setup time and realistic scenarios, are important steps to transfer empirical cognitive models from a laboratory setting to real HCI applications. Given that we applied the generic processing chain of empirical cognitive modeling to three different user states successfully, we assume that the approach can be extended to many additional user states for which neural or physiological correlates exist.

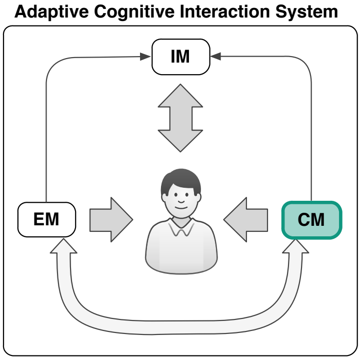
In the following chapters, we will employ the developed empirical models for two main use cases: First, information from an empirical cognitive model will be used to adapt a computational cognitive model for the prediction of human behavior and performance at different workload levels. We will look at different variants of this application in Chapter 3. Second, the information of the models will be sent to the interaction manager to adapt the behavior of an interaction system according to the detected user states. Will look at such applications in Chapter 4. In all three sections of this chapter, we performed binary classification with accuracy scores significantly above the random baseline for nearly all classification conditions, but still far from error-free user state detection. In the next two chapters, we will investigate whether the achieved accuracy is high enough to provide a measurable benefit for improving prediction of a computational cognitive model and for improving user experience of an interaction system.

Chapter 3

Computational Cognitive Modeling

In this chapter, we introduce the concept of computational cognitive modeling for cognitive interaction systems. Cognitive models simulate cognitive processes and predict performance as well as behavior of a user. We investigate computational cognitive models for memory and reinforcement learning. We analyze how these models can be combined with empirical cognitive models to adapt the prediction to different workload levels and to detect user states which can only be modeled by the interplay of both model types.

3.1 Introduction



In the previous chapter, we introduced empirical cognitive models to estimate internal user states from sensor data. Still, there are limits to the complexity of empirical models, measured in the number of classes which they can discriminate. [KCVP07] showed experimental results and gave theoretical reasons indicating that BCI-based empirical cognitive models cannot reliably discriminate more than 5-7 classes. Causes for this limit are the noisy signals and the limited spatial resolution of non-invasive methods to capture brain activity. This limited number of classes restricts the potential of adaptive behavior for an interaction system which is used for complex, dynamic tasks. Additionally, models based on surface EEG or fNIRS can only measure cortical activity. This imposes fundamental limits on the selection of cognitive processes which can be predicted using empirical cognitive models. This is because activity in several relevant brain areas (e.g. the amygdala, associated with memory, decision-making, and generation of emotional responses) cannot be captured by those sensors. However, the user states which are important for HCI are not limited to the user states which can be captured by empirical cognitive models. For example, the state of the user's memory is relevant for the interaction system: The system should account for the limitations of memory, as well as for association processes and memory dynamics. However, most State-of-the-Art interaction systems simply assume human memory to be an unlimited, static and unconnected data storage.

In this chapter, we propose to complement empirical cognitive models with computational cognitive models which are dedicated to a detailed, validated representation of cognitive processes in a psychologically plausible manner. One promising application of computational cognitive models in HCI is the representation of the user's memory. This is especially relevant for interactions in which the system conveys a lot of information from large domains. Examples for such interaction scenarios are virtual in-car tourguides, personal technical companions or humanoid robots. A computational memory model represents the associations between different pieces of information which are triggered in the user's mind by the system and external stimuli. The computational model may also model dynamics of memory, i.e. increasing and decreasing activation of memory content.

There are two potential ways to employ a computational cognitive model in the context of interaction systems: First, a computational cognitive model can be used to simulate a user who interacts with a system, for purpose of automated evaluation and training. Second, a computational cognitive model can be used during an HCI session to provide a real-time prediction of the user's cognitive state. While the first way (user simulation) has already been established in the research community, the second way (real-time prediction) has been investigated very little. For the application of a computational model for real-time prediction in realistic interaction systems, we identified three major challenges: (1) the need for real-time model tracing (i.e. the ability to dynamically model the changing cognitive state of a human during the execution of a task) and the integration of large-scale databases of knowledge; (2) the accommodation of multiple workload levels to improve the prediction of performance under different user states; (3) the combination of empirical and computational cognitive modeling. This chapter will provide contributions to these three challenges.

The chapter contains five main sections: Section 3.2 introduces the relevant fundamentals of computational cognitive modeling, with a focus on ACT-R and its declarative memory module. This section also gives an overview of the related research regarding cognitive user simulation and adaptive computational modeling. Section 3.3 describes a memory modeling component for use to dynamically model memory activation during interaction. Section 3.4 investigates the adaptation of the memory model to different workload levels by explicitly modifying model parameters and present a detailed evaluation. Section 3.5 shows an alternative implicit approach for workload adaptation using dummy models. Finally, Section 3.6 presents how empirical and computational cognitive models predict the presence of learning situations in an associative learning task.

3.2 Related Work

This section discusses the relevant State-of-the-Art in computational cognitive modeling. It starts with a short introduction to computational cognitive modeling and cognitive architectures in general. The next two subsections present related work on the two application areas of computational models, cognitive user simulation and real-time prediction of cognitive states. Afterwards, we turn our attention to the ACT-R architecture and introduce the relevant modeling concepts: Reinforcement Learning, Threaded Cognition, knowledge representation, and memory modeling. As the latter is of high importance for our own contributions, we also discuss limitations of the ACT-R memory model and the alternative approach to (associative) memory in the LTM^C approach. As a computational model in an HCI context needs to connect to other components, we also discuss approaches to interface ACT-R. Finally, we look at the State-of-the-Art regarding adaptive computational modeling.

3.2.1 Computational Cognitive Modeling & Cognitive Architectures

Modeling processes of human cognition by computational cognitive models is one of the main methods in cognitive science. This research area has emerged in the 1950s [Mil03], pioneered by researchers like Alan Newell [NSS59], Alan Turing [Tur50], Marvin Minsky [Min61], Noam Chomsky [Cho59] and many others as a new approach to understand human behavior and performance. Cognitive science provided an alternative to Behaviorism which was the dominant research agenda since the beginning of the 20th century. Behaviorism focused on identifying mappings from stimuli to observable behavior and how those mappings emerged by conditioning. In contrast, cognitive science strives for an understanding of the internal information processing and the corresponding internal representations which ultimately lead from input stimuli to such behavior. Ever since, cognitive science has developed into a mature multidisciplinary endeavor, influenced by psychology, neuroscience, computer science and other areas.

While the initial research on computational modeling focused on specialized models for specific aspects of cognition, Newell's call for a "unified theory of cognition" [New94] stipulated the development of comprehensive cognitive architectures. A cognitive architecture defines the fundamental, irreducible primitives of cognition and perception which can then be used to specify models for concrete tasks. Over the years, many ambitious and successful architectures have been proposed. While each architecture by definition aims at a comprehensive modeling of human cognition, each architecture has its strengths and weaknesses. For example, the SOAR architecture [LNR87] concentrates on planning and problem solving. The PSI architecture [Dör01] concentrates on motivation and emotion, and CO-JACK [ERB+08] on modulation of cognition through various influences. The ACT-R architecture [ABB+04] focuses on memory, perceptual-motor capabilities and the accurate prediction of execution time and accuracy of the modeled cognitive tasks.

3.2.2 Cognitive User Simulation

Cognitive user simulation [RY01] is one of the major use cases of computational cognitive modeling in the HCI context. It is a way of simulating user behavior in a cognitively plausible way. By replacing a real human user with a corresponding simulation, it is possible to evaluate an interaction system cheaply by generating a potentially unlimited number of user-system interactions. This allows to test early prototypes for which evaluation with real users would be expensive and time-consuming. As a sound cognitive user simulation behaves similarly to the average user (also in regards of its limits of cognition), those simulated interactions can be used to identify design flaws and bottlenecks in these prototypes. A cognitive user simulation consists of a computational cognitive model of the interaction task, a model of the task itself and of facilities for accessing the real or simulated interface, i.e. for sending system input and retrieving system output. User behavior is then simulated by running the model.

One of the first practical implementations of a cognitive user simulation was presented in [AHR07]. The authors used a cognitive user model to simulate expert operation of a cell phone menu in ACT-R and compared it to a model based on Fitts's law [Fit54] (a successful rule of thumb for estimating time to select a target (i.e. a menu item) from spatial distance to the target and target size) for predicting response time. For interaction with the environment, the authors connect the user model with a simulator of the cell phone to send visual input to the model and to receive the key strokes from the motor module. The authors showed that the computational cognitive model outperformed the traditional non-cognitive model in prediction accuracy. [CP10] used a similar methodology to evaluate interfaces for smart homes for users with cognitive disabilities. The authors showed that for their use case of a contextual assistant, the ACT-R model yielded the most accurate prediction of user behavior, compared to a GOMS model (another, simpler cognitive architecture) and one based on Fitts's law. The DISTRACT-R model [SZBM05] used the ACT-R mechanism of threaded cognition to combine a complex model of car driving [Sal06] with adjustable models of interfaces for in-car systems like radio, and phone. The software is able to model the impact of system operation on driving performance for different interface configurations. It can therefore be used for rapid prototype evaluation. DISTRACT-R provides a graphical interface to select model parameters and evaluate the simulation results.

Apart from employing computational models for the simulation of user behavior, computational cognitive models also have been used in the design of interaction systems to create plausible intelligent virtual agents (IVAs). Those agents are designed to emulate human behavior to represent a system with which a user can naturally interact and converse. [BLK10] developed an IVA for large-scale computer games with an episodic memory. Episodes are represented in memory as nodes in a tree structure. The episodic memory supported a forgetting mechanism which removed "unimportant" episodes.

Importance of an episode is determined by its age and emotional salience. In a similar fashion, [LAH$^+$09] implements a memory model based on the LTM^C extension of the ACT-R memory model to personalize interaction. For this purpose, information about the user which the agent learns during one interaction is stored in memory and retrieved again during subsequent interactions.

3.2.3 Computational Cognitive Modeling in Interaction Systems

Simulation of cognitive processes is also relevant for an interaction system at runtime. There are a number of systems and studies which acknowledge that the user's memory is relevant for the design of interaction systems. For example, it is accepted that the design of a system has to account for the limitations of working memory. [KA07] dealed with cognitive tutoring systems which employed strategies for reducing memory load by removing irrelevant information or by visualizing the discourse structure. [WGM$^+$09] investigated the influence of different strategies for information presentation on working memory and compared the trade-off between shorter utterances at the cost of more complex discourse structures. [JSW$^+$99] modeled in a Bayesian network several factors of human cognition which have an impact on dialog system performance, including memory capacity limitations.

One type of interaction systems for which user's memory processes are very important is spoken dialog systems, as those deal with information exchange and grounding processes for large domains. Most State-of-the-Art spoken dialog systems acknowledge the difference between the discourse model of the system and the set of beliefs of the user. For example, many systems have a notation of grounding to model presence or absence of a common ground and can thus model potential discrepancies between the system's perspective and the state of the user's mind [PH00]. However, once information is assumed to be correctly processed, most systems cannot handle the user's dynamic memory processes, i.e. they do not cover the activation of new concepts by association or their removal by forgetting or concept drift. Lieberman et al. [LFDE05] used a large associative database as an additional information source for an automatic speech recognizer and showed how incorporating knowledge of human associations can improve the results of statistical models. However, this approach does not explicitly model cognitive processes.

3.2.4 The cognitive architecture ACT-R

This section will give on overview of the cognitive architecture ACT-R. ACT-R (which stands for Adaptive Control of Thought - Rational [ABB$^+$04]) concentrates on memory, perceptual-motor capabilities (inherited from EPIC [KM97]) and the accurate prediction of execution time and accuracy modeled at a fine-grain level. Those strengths make it well-suited for the analysis of cognitive tasks which emerge during HCI. ACT-R is a hybrid architecture that uses both symbolic and sub-symbolic computation to model cognitive processes. While the symbolic parts of the architecture (e.g. production rules) represent aspects of higher-level cognition, its subsymbolic aspects mainly represent automated, low-level aspects of cognition. This hybrid approach relates the concept of two distinct systems of cognition [Slo96], *system one* for low-level cognition of similarity and temporal relations and *system two* for logical high-level cognition formulated in rules.

ACT-R is a modular architecture which is composed of several building blocks which represent functionally encapsulated aspects of human cognition (see Figure 3.1). Among others, there exist modules for (declarative and procedural) memory, for perception and motor execution. Each module has one or more buffers which are used to exchange information between modules, to send requests to modules and to receive output from modules. There exists evidence for a high-level mapping of ACT-R modules to certain specialized brain regions [And07]. ACT-R is implemented in the programming language LISP and is currently freely available in version 6[1].

The Pattern Matching component is the central execution unit of ACT-R. It retrieves production rules from the Procedural Memory and selects exactly one for execution (this restriction to one rule is called the “serial bottleneck” [ABB$^+$04]). A production rule consists of preconditions and bindings. The preconditions determine whether a rule is eligible for execution. Preconditions refer to the content of buffers or the status of modules (busy or free). To chose between multiple eligible rules, ACT-R evaluates the utility of each rule and performs a probabilistic selection preferring rules with high utility. The higher the utility of a rule, the higher the estimated probability of achieving the corresponding goal. Using the Pattern Matcher, a single production rule is retrieved and executed whenever the state of the model is updated. Execution of a production rule takes exactly 50 ms.

Utility, which is one of the sub-symbolic aspects of the architecture, is not assigned manually but learned from experience by the model. For this pur-

[1] http://http://act-r.psy.cmu.edu/

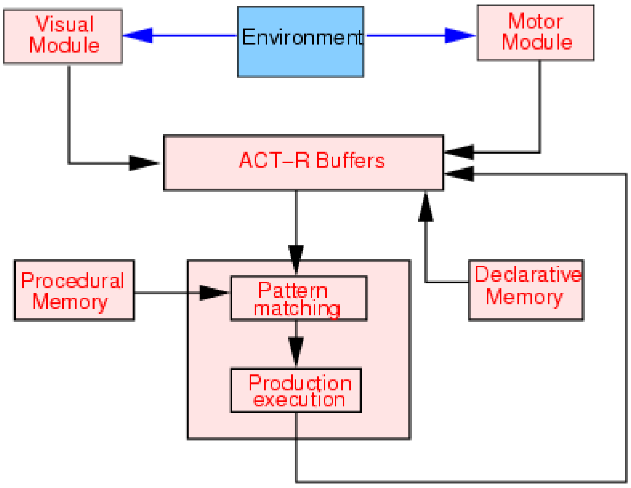


Figure 3.1 – Overall architecture of ACT-R with all core modules (from the official ACT-R website, http://act-r.psy.cmu.edu/about) and the associated brain areas (in parentheses).

pose, ACT-R implements an approach based on Reinforcement Learning (RL) [FA08].

3.2.5 Reinforcement Learning

RL is a paradigm of learning from observation and experience. RL has been shown to allow both, the prediction of human behavior in repeated learning from experience [FA06, SG12] as well as the learning of complex behavior by artificial systems like robots [KP12]. The fundamental task of RL can be modeled as a Markov Decision Process (MDP) which is observed by the learning agent. The agent reacts to the observed state of its environment by taking actions which lead to a state transition of the MDP and the return of a reward. The goal of the agent is to learn a strategy which maximizes the cumulative reward over time. More formally, an MDP for RL is defined as a tuple $M = (S, s, A, t, r)$, which consists of a set of states S, an initial state s, a set of actions A, a transition function $t : S \times A \to S$, and a reward function $r : S \times A \to \mathbb{R}$. The goal of the agent is to learn a strategy $\pi : S \to A$ that maximizes the discounted cumulative reward $\sum_{i=0}^{\infty} \gamma^i r(s_i, \pi(s_i))$ for a state-action sequence $s_0 = s, s_{i+1} = t(s_i, \pi(s_i))$ generated by iteratively applying π to select actions. γ is the discounting factor, which models the importance of initial rewards over future rewards.

There are several fundamental paradigms to implement RL algorithms, from dynamic programming to Monte Carlo approaches [SB98]. In this thesis, we concentrated on Temporal Difference Learning (TD) methods, which are also the paradigm underlying the ACT-R utility learning. The basic principle of TD is that the agent explores the state-action space to learn an optimal strategy by observing state transitions and rewards. One of the central algorithms of TD is Q-Learning [Wat89]. It uses the exploration to learn an approximation $\hat{Q}(s,a)$ of $Q(s,a)$, which is the expected discounted cumulative reward when taking action a in state s. $Q(s,a)$ can be recursively defined as $Q(s,a) = r(s,a) + \gamma \max_{a'} Q(\pi(s), a')$. After each step taking action a in state s, the evaluation of that state-action pair $\hat{Q}(s,a)$ is updated based on the observed reward $r(s,a)$ and the (tentative) evaluation of the following state $t(s,a)$. At the beginning of exploration, $Q(s,a)$ is randomly initialized with small positive values. This process is repeated until $\hat{Q}(s,a)$ converges for all state-action-pairs. Once the agent has learned $\hat{Q}(s,a)$, the exploration stops. The final strategy is to take action a which maximizes $\hat{Q}(s,a)$ for a given state s. [SB98] calls this approach “bootstrapping”, as tentative (and initially unreliable) estimates of the evaluation function are used to update other estimates. One major advantage of TD methods is that they are model-free, i.e. they do not require a-priori knowledge of r and t.

3.2.6 Threaded Cognition

Threaded cognition is an approach to model multi-tasking in ACT-R [ST08]. While the serial bottleneck of ACT-R (see Section 3.2.4) enforces the execution of exactly one production rule at a time, threaded cognition allows quasi-parallel execution of tasks by maintaining parallel goals, one for each task. A scheduling mechanism selects one production rule from the union of all rule sets in the shared procedural module. The tasks compete for exclusive cognitive resources (i.e. modules) which are not available when another task requires this resource to be processed. Tasks which require otherwise busy resources have to be delayed. Threaded cognition has been validated with a number of dual-tasking paradigms. One famous example is the Schumacher dual task [SSG$^+$01]. For this dual task, humans are able to achieve perfect time sharing between a visual-manual task and a auditory-verbal task. This is possible because the demand for the cognitive resource shared by both tasks (declarative memory) was drastically reduced by learning during training. [ST08] could show that an ACT-R model of the Schumacher task with threaded cognition is able to reproduce this behavior.

Originally, threaded cognition was provided as an extension to ACT-R and is now integrated in the ACT-R version 6.

3.2.7 Knowledge Representation & Memory

In this section, we will briefly describe how declarative knowledge (i.e. conscious, factual information) can be represented and how declarative memory can be modeled.

Declarative knowledge is represented in ACT-R in the form of chunks. A chunk, similar to the frame as defined by [Min74], consists of a name, a type, and attributes. Attributes are either atomic (e.g. an integer) or a reference to other chunks. Chunk types are organized in an ontology to represent type generalization and specialization. Chunks are the basic unit for communication between modules in ACT-R as they are transferred between buffers. Preconditions and bindings of production rules usually refer to and manipulate chunks. Chunks also form the content of the declarative memory module.

Memory is one of the most important aspects of cognition, as it is involved in nearly any non-trivial cognitive task. For example, working memory is strongly correlated [CKE03] to the concept of general intelligence [Spe27]. For this reason, memory modeling has a long tradition in cognitive psychology and a variety of models are available. For many theories of memory, models are implemented in cognitive architectures like ACT-R [ABB$^+$04]. However, there exists no common ground among modeling experts on how memory processes should be represented formally to account for all known memory phenomena and for findings from neuroscience. For example, there is ongoing debate on whether there is a fundamental separation between short-time and long-time storage [AS68] or not [Cow93], on whether forgetting is based on interference [OK06] or decay [Cow93], and whether memory is composed of specialized sub-systems (e.g. for information from certain perceptual modalities [Bad92]) or not. [MS99] presents a large variety of very different models of working memory.

In ACT-R, memory is divided into the declarative module and the procedural module. The declarative module represents semantic and episodic memory. As this comprises the content of interactions between user and system, we concentrate on declarative module in this work. We chose this model as the basis for our computational cognitive memory model as it is validated in numerous studies which show that it accounts for a large number of important memory phenomena [ABLM98, AR99, LDR00]. It also provides a concrete mathematical formalism which is suited for implementation within a component for memory modeling in interaction systems. Declarative memory is presented as a unitary construct with no explicit distinction between long and short-time memory. Forgetting is mainly implemented as decay of activation. However a limitation in activation spreading may also be interpreted as a displacement mechanism [ARL96]. Requests to the declarative module are formulated in form of partially filled chunks which are matched against the stored chunks. The module associates an activation value to each chunk in order to define "active" information units. Activation is calculated from base level activation, spreading activation and noise. It determines retrieval probability and retrieval latency. In Section 3.2.8, we will continue with a more in-depth discussion of the relevant aspects of the memory model in ACT-R.

3.2.8 Memory Modeling in ACT-R

In this section, we look at the details of the ACT-R model of declarative memory. The ACT-R declarative module maintains a set of chunks that represent the current declarative knowledge. Chunks are inserted into the declarative module, when new information is learned, and they are retrieved from it, when information is requested from memory. Probability of retrieval and retrieval time depend on the activation value of the chunk. Activation increases if a chunk (or a related chunk, see below) is stimulated (or "encoded"). This stimulation can happen externally, for example when the concept represented by this chunk is mentioned by the system during interaction, or internally, for example as the result of an internal thought process. Activation of a chunk decays over time if it is not stimulated.

$$A_i = B_i + S_i + N_i \tag{3.1}$$

Activation A_i of a chunk i in ACT-R is calculated as the sum of three components: base level activation B_i, spreading activation C_i and noise N_i, see Equation 3.1. Base level activation models the influence of recency and frequency of stimulations on activation. Equation 3.2 shows the formula to calculate *base level activation* B_i of a chunk i that was stimulated n times. Activation depends on the current time t_c and the time t_k of the all stimulations $1 \ldots k$. The decay parameter d describes the rate of decay over time

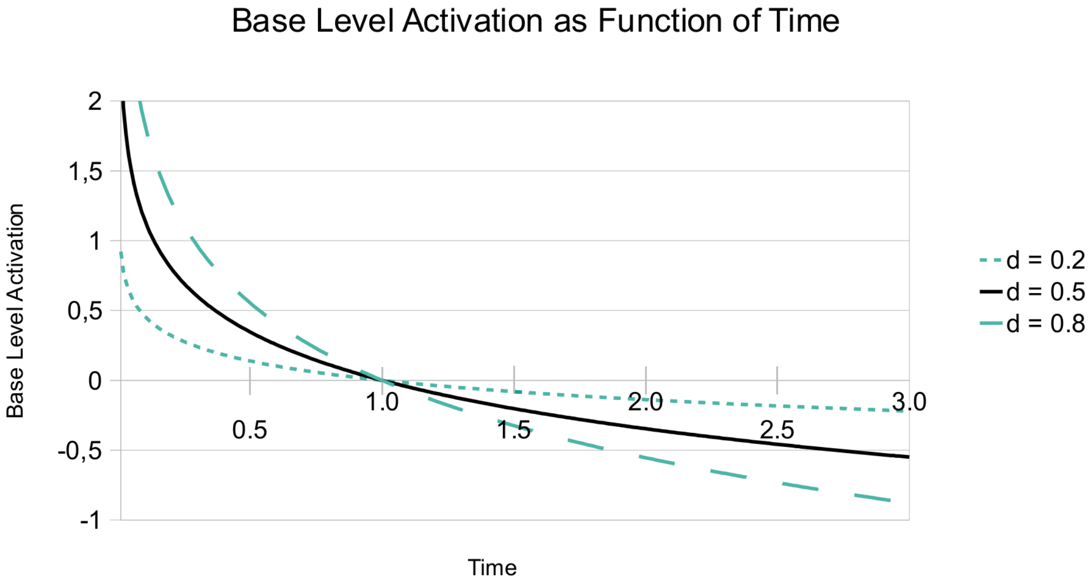


Figure 3.2 – Base level activation over time for different decay factors d.

(which simulates forgetting). The default value for d in ACT-R is 0.5.

$$B_i = \ln(\sum_{k=1}^{n} (t_c - t_k)^{-d}) \tag{3.2}$$

Figure 3.2 illustrates the base level activation over time of a chunk that has been encoded at $t = 0$, using three different values for the decay parameter. As it can be seen in the figure, a higher value of d causes a faster drop of base level activation and is therefore associated with faster forgetting.

Another source of activation is the *spreading activation*, which is computed as in Equation 3.3, where W_j reflects the attentional weight of chunk j at the current point in time and S_{ji} represents the association strength between chunk i and element j.

$$C_i = \sum_j W_j S_{ji} \tag{3.3}$$

W_j is usually set to $1/x$ with x being the number of activation sources. S_{ji} is usually set to $S - ln(\text{fan}_j)$ with fan_j being the number of facts that are associated to element j. The parameter S is often set to a value of 2 which has emerged as reasonable value.

The third source of activation is the *noise activation* N_i. It is modeled as a random variable following a logistic distribution with mean value 0 and variance σ^2.

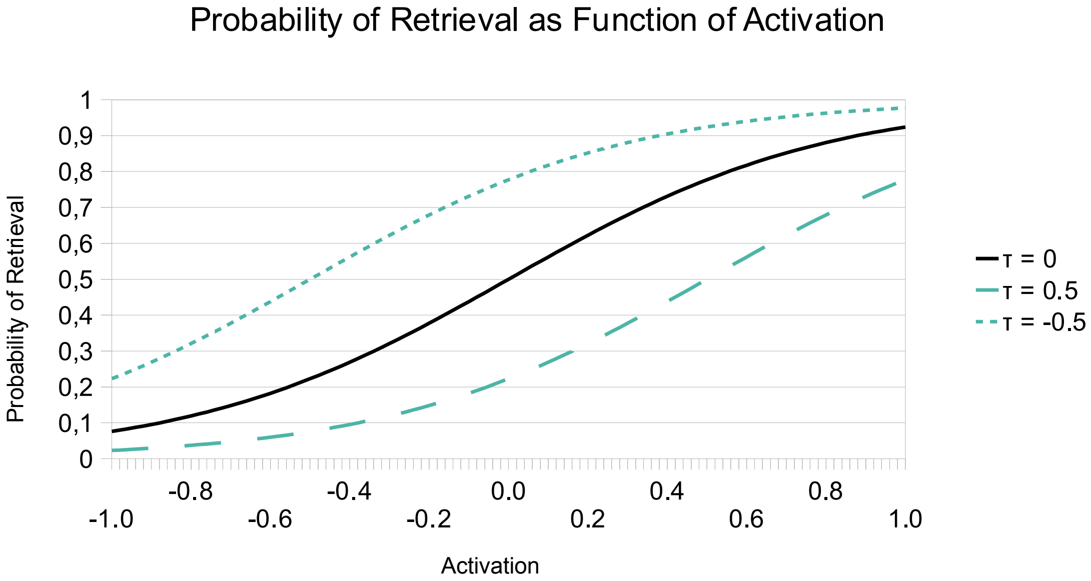


Figure 3.3 – Probability of retrieval as function of activation. The probability of retrieval is illustrated for three different values of τ. The parameter s is alway set to 0.4 in this example.

Chunks are retrieved with a probability which depends on their activation. Equation 3.4 shows how the *probability of retrieval* is computed[2]:

$$P_i = \frac{1}{1 + e^{-(A_i - \tau)/s}} \tag{3.4}$$

Chunks can only be retrieved successfully if their activation value is greater than a specific threshold value τ. The parameter s controls the sensitivity of the retrieval probability against varying activation values and is by default set to 0.4. It also influences the variance σ^2 of the noise activation. Figure 3.3 illustrates equation 3.4 for three different values of τ and with $s = 0.4$.

If a chunk is retrieved successfully, the ACT-R declarative module provides the *latency of retrieval* which describes the duration of the retrieval process (i.e. the time between request and retrieval). It is computed by using equation 3.5 where A_i is the activation value of chunk i and F is a latency factor.

$$T_i = F \cdot e^{-A_i} \tag{3.5}$$

[2] Note that some variables in this section (A, s) are also used in the context of RL in Section 3.2.5. We decided to retain this double usage as the variable identifiers are firmly established in the literature. The semantic of a variable will always be clear from the context.

Equation 3.5 is only applied if a chunk is retrieved successfully. In case no chunk is retrieved, the *failure latency* is returned, which is the time it takes to detect the failure. According to the ACT-R Tutorial[3], it can be computed by using equation 3.6:

$$T_i = F \cdot e^{-\tau} \tag{3.6}$$

If encoding and responding are involved (e.g. when reading a word, then trying to remember this word and finally pushing a button to indicate that the word is known), the overall recognition time (measured as delay between beginning of the perception and the actual response) can be computed by adding an additional parameter I to the retrieval latency as defined in equation 3.5 and to the failure latency as defined in Equation 3.6, respectively. This parameter I is the intercept time and reflects the time needed to encode the item and to perform a response. The *recognition time* can then be computed as in Equation 3.7.

$$\text{recognition time} = I + T_i \tag{3.7}$$

3.2.9 Limitations of the ACT-R Memory Model

Though it has been used in many different applications, the ACT-R memory model is not without problems. A very important limitation is that in order to make use of the declarative knowledge, one has to know the exact structure of the knowledge defined in the chunk types [SBB06]. Without knowledge of the semantic of its attributes, the information stored in a chunk is meaningless. This makes it difficult to use the chunk system with very large knowledge bases since one would need to define and keep track of a lot of different chunk types. Another problem is the lack of generalization in the procedural knowledge. The ability to follow associations in ACT-R would need to be modeled with a set of production rules. However, since production rules are specific to their corresponding chunk types, they can not be used on a knowledge base that uses different chunk types. This makes it necessary to re-implement a different set of production rules corresponding to each knowledge base. These limitations of the ACT-R declarative memory model also hinder its application for interaction systems. It would be required to enter all available knowledge – which can be very diverse for a real-world

[3] http://act-r.psy.cmu.edu/actr6/reference-manual.pdf

application – as chunks in the declarative module, which also requires the definition of the associated chunk types and the corresponding ontology. The limitation on association processes and partial matching make it difficult to use the ACT-R memory model to represent human memory during tasks of handling.

When we pursue the goal to employ a memory model for a conversational general-purpose interaction system, it is important to provide a large number of common-knowledge chunks to the model to cover all potentially relevant concepts. Both formally (requiring the modeler to manually provide a lot of structural information) and by its software design (which does not provide efficient data structures), the ACT-R declarative module is not prepared to handle large data sets. There exist few publications on the integration of a larger knowledge base to the ACT-R declarative module. WN-Lexical [Emo06] is an example for a replacement of the original ACT-R declarative memory using WordNet, a large lexical database. [DM10] externalizes the declarative module of ACT-R by employing a relational database. This technically enables the system to handle much larger number of chunks compared to the original implementation. Methodologically, it replaces the original implicitly defined association network of chunks by a large semantic network. [DLS10] uses this approach for the cognitive architecture SOAR and demonstrates performance improvements by two orders of magnitude.

3.2.10 LTM^C

LTM^C was developed to address the issues of the ACT-R memory system. LTM stands for “Long Term Memory”, the C represents Casimir, a cognitive architecture which concentrates on spatial cognition [Bar09]. LTM^C can be used as an extension or a replacement to the ACT-R declarative memory module. While its name explicitly refers to long-term memory, “LTM-Buffers approach follows the view that working memory is not a separate memory store, but rather that it is highly-activated portions of LTM”[SL07].

In LTM^C, the memory is stored as nodes and directed connections between them, forming a network. Every node has a name that identifies which entity it represents. The edges between nodes do not have types assigned. This means that the edges themselves do not stand for relations. Instead, the relations are represented by nodes, too. The only exception to this is the “IsA” relation (also called subsumption) - it is directly represented by edges between concepts and used to build an ontology that includes the relation nodes: Every specific relation is subsumed under a node for its relation type (this architecture is the reason why the subsumption relation cannot be represented by nodes - it would lead to infinite regress). For an example, see Figure 3.5 in Section 3.3.1. LTM^C offers a great deal of flexibility in knowledge modeling: The are no predefined types, new relation types can be added at any time and relation nodes can even have relations themselves.

The spreading activation in LTM^C works essentially the same as in ACT-R. Nodes get activated when a retrieval request is made and they spread part of their activation along their links. The spreading process is stopped when the activation falls under a threshold. Because of the graph model of LTM^C, spreading corresponds to a simple graph traversal, starting at the activated nodes. After spreading has stopped, another process takes place to choose the set of nodes to be retrieved. Only nodes with an activation higher than the average activation of nodes in the network are considered. Out of these nodes, the connected subnet with the highest total activation is selected to be retrieved. Finally, in order to be usable by the rest of the ACT-R modules, this subnet has to be converted into a chunk. This is achieved by using a mapping that has to be defined with the chunk types of ACT-R to specify how the nodes in LTM^C relate to the slots of the chunk type [SL07].

3.2.11 Interfacing ACT-R

ACT-R provides an Application Programming Interface (API) which can be accessed by programs written in LISP to model a specific task. For this purpose, one has to provide a full set of production rules encoding the processing strategy, a definition of chunk types and chunks, a model of the task, and the relevant aspects of the environment in form of a LISP program. While this means that ACT-R models and extensions can use the full potential of a comprehensive functional programming language, it also means that the implementation of ACT-R models is a cumbersome task. It requires to connect all dependent components (i.e. an interaction manager) to the ACT-R environment. Additionally, developers need to provide a full model even if they are only interested in a subset of functions of the architecture. Several approaches to remedy this problem have been explored: ACT-R can be extended with interfaces to programs written in more modern programming languages ([MS99] and see also Section 3.5). However, this still leads to large overhead in software design and process communication. There exist alternative implementations of ACT-R in Java[4] and Python[5]. However, those

[4] http://cog.cs.drexel.edu/act-r

[5] https://sites.google.com/site/pythonactr

are naturally lacking the improvements of the latest official release and are less well supported with extensions and models. Finally, there are software toolkits available [AHR07], which automatically create ACT-R models from a more abstract representation, for example in the form of GOMS (Goals, Operators, Methods, and Selection rules) models. The downside of such toolkits is that they are usually limited to a very restricted domain and do not provide help for general modeling tasks. For those reasons extract and encapsulate certain aspects of the architecture and create re-usable building blocks which can be directly applied in interaction systems. This approach was for example used in [LDG12]. In this publication, the authors show an example of successfully extracting certain mechanisms of ACT-R and using them within another model or application. In their case, they develop the Instance Based Learning Theory (IBLT) of dynamic decision making, which models learning from experience in repeated choice tasks. Each observation is stored as an instance in memory and at decision time, instances similar to the current one are retrieved from memory using the base level activation. Extracting the memory model from ACT-R provides a validated model to ACT-R to IBLT and reduces modeling effort, necessary technical knowledge and increases generalizability compared to employing the complete ACT-R architecture.

3.2.12 Adaptive Cognitive Modeling

The ACT-R memory model accounts for the retrieval probability and latency for a memorized item given how often and how recent it was presented to the agent. However, this memory model was designed to model average human performance with no distracting tasks and a low workload level. This implicit assumption holds for the whole ACT-R theory. There are few publications available which deal with cognitive modeling of non-standard conditions. In [CLC06], the authors model the impact of arousal on memory performance. This is done by introducing additional parameters to the formulas for calculating activation of a memory item: The decay parameter d is scaled depending on the arousal level to represent an inverted u-shaped curve. [RRS06] discuss several theories of modeling stress in a cognitive architecture. The authors propose the technique of overlays which modify the basic mechanisms of the architecture to model the effect of certain cognitive conditions. The authors propose several variants of such overlays, ranging from simple parameter adjustments to the addition of a “worry” task to consume cognitive resources. They suggest different effects of stress: perceptual tunneling, cognitive tunneling, and decreased attention overlays. The authors do not provide a detailed evaluation of their proposed approaches. [GRG+11] proposes a generalized mechanism to develop models which predict performance under sleep deprivation. A modification of utility calculation results in higher probability of executing no rule, reflecting the effect of microlapses which are typical for this user state.

Other cognitive architectures besides ACT-R implement the overlay mechanism already in their core design. The CO-JACK architecture [ERB+08] in its core concepts incorporates modulation of cognitive processes to model variability due to physiological factors and affect. The PSI architecture [DSS99, Bac09] is a neural network based architecture, specialized on modeling factors which influence decision making and planning of a virtual agent. For example, PSI has a notion of motivation and emotion, which both directly and indirectly influence cognitive processes and the resulting behavior. Both are not represented as isolated modules, but as an intrinsic aspect of cognition. One central mechanism is the concept of modulators. The cognitive state of the agent is determined by the agent's urges, which are grouped in three categories: physiological, cognitive, and social. Those urges on the one hand determine the motives towards which the agent plans its actions. On the other hand, they influence parameters of the central planning algorithms. For example, the selection threshold determines the readiness of the agent to switch its current planning goal. It depends on the urgency of the current planning goal and the agent's competence for the current task. From this internal state, emotions emerge which the agent potentially can communicate to other agents. Closer to the original ACT-R theory is ACT-RΦ. This hybrid architecture combines ACT-R with the HumMod model of physiology and affect [Dan13]. Modeled affective states and physiological conditions influence the simulated cognition by modifying utility values of production rules. The authors demonstrate their approach by modeling the impact of thirst on decision making in an ultimatum game.

Conclusion

Literature review has shown that computational cognitive modeling is a feasible technique for prediction of user behavior and performance in the HCI context. Most cited work in this area is on offline user simulation for the evaluation of interfaces. However, there are a few limitations of computational cognitive models which hamper their application in adaptive cognitive interaction systems. The ACT-R memory model is not designed to flexibly handle associations in large information networks. The LTM^C extension provides a promising alternative but needs to be extended to account for memory dynamics and to import large-scale databases. To support real-time model tracing, computational models which adapt to user states like workload are necessary. While there exist some studies which demonstrate the general feasibility of modulating simulated cognition, only a small number of user states has been covered and the proposed models often lack evaluation. Finally, we saw that there is little work on the combination of empirical and computational models.

3.3 Dynamic Large-Scale Memory Model

Interaction systems have matured to a point where they are routinely employed in static and controllable scenarios, such as virtual call-center agents. However, they still lack flexibility and robustness in dynamic scenarios involving spoken interaction between system and user. Examples for such scenarios which require a large flexibility are human-robot interaction, in-car systems or portable companion technology. In such scenarios, a major challenge is the fact that it is hard to estimate "what is on the user's mind": The discourse of the interaction between system and user may shift between topics due to evoked associations in the user's mind; external stimuli may cause sudden changes of attentional focus, while other discourse items may fade out and eventually be forgotten by the user. These effects become particularly important when the interaction becomes less task-driven and more conversational, as envisioned for natural interaction systems. Another challenge in verbal human-computer interaction is the ambiguity of natural language. While humans are able to resolve it by referring to a shared context, computer systems mostly lack this ability. Providing knowledge about human association mechanisms is one step towards enabling systems to understand the underlying semantic context of an interaction.

In most State-of-the-Art interaction systems, such a model of human memory is usually implemented implicitly in the form of a discourse model or dialog state. However, human memory is imperfect, context-dependent, nondeterministic and limited in capacity. Ignoring these properties will lead to imprecise prediction of human behavior in relevant cognitive tasks and in interaction situations. One example of behavior resulting from those properties of human memory is the "Moses Illusion" [EM81]: Participants of a study were asked questions like "How many animals of each kind did Moses take onto the Ark?". Even though most Christians know that it was Noah and not Moses who built the ark, many give the answer “two”, which is wrong in terms of deductive reasoning but plausible in terms of association. While this example was constructed specifically to study such “fuzzy associations” [EM81], similar situations can occur in the HCI context (for example, a user of a virtual tourguide asking for information on the Neuschwanstein Castle in Munich). Observations like this underline the importance of using a cognitively sound and validated model of memory.

Besides accurate modeling of human memory and associations, another important aspect of a computational cognitive model for interaction systems is its interface to the other components of the interaction system. A mature cognitive architecture like ACT-R provides a powerful model of human cognition, but is also cumbersome to integrate into an existing interaction system. Therefore, we follow the philosophy of [LDG12]: We extract the mechanisms of declarative modeling and encapsulate them in a re-usable memory model that can be plugged into interaction systems as part of the user model or user simulation.

The main contribution to the scientific community of this section is three-fold: First, the development of a flexible, stand-alone memory model which supports flexible modeling of associations. Second, the addition of the capability to access large knowledge-bases to populate the memory model. Third, the implementation of memory dynamics which support phenomena like concept drift, which occur frequently in interaction situations.

3.3.1 Flexible, Stand-alone Memory Model

In this section, we introduce the Dynamic Memory Module (DMM), a stand-alone associative memory framework, which provides a way to model dynamic association processes of the user’s memory[6]. DMM is able to determine the most likely associations of a human for a given memory configuration and a set of new stimuli. It can provide this information to an interaction system or to speech processing components so that these systems can resolve ambiguities or determine the user concerns. Since content and context of an interaction change over time, modeling dynamics is a crucial part of DMM: Different associations are activated as new items come into focus, integrated with previously active items while items which are not active gradually fade out. As a user modeling component in an interaction system, DMM can

[6] This section is based in parts on the study theses of Robert Pröpper and Florian Krupicka which were supervised by the author of this thesis.

identify topics which are most relevant to the user in the current context without an explicit request. Another very important area of application of the DMM is user simulation to automatically create training and evaluation scenarios for dialog systems: Here, the model can be used in a generative fashion to simulate plausible associations for a situation and derive consistent speech acts and utterances of a user. Other possible applications for DMM are the enhancement of speech processing systems and translation systems by helping to resolve ambiguities - a result obtained with purely statistical methods is more likely to be correct if it is also part of the associative context in DMM. Our DMM is based on the LTM^C extension of the ACT-R memory model as described in Section 3.2.8. LTM^C is an improvement over the ACT-R declarative memory model in terms of flexibility and association modeling. However, it is not explicitly designed to reflect memory dynamics resulting from a sequence of memory stimulations over time, for example caused by concept drift during an interaction. We extend the model for the use in interaction systems such that it represents memory dynamics and enables the import of large-scale common knowledge databases.

Knowledge Structure

To describe the structure of the knowledge representation, we will use the terms *concept* and *association*. A concept is an object of common sense knowledge. It can be a physical object, an attribute, an activity or an abstract idea. Associations are links between concepts. An example for an association is the statement “The KIT is located in Karlsruhe”, where “KIT” and “Karlsruhe” are concepts and “is located in” is the association between them. Both concepts and associations are represented as equivalent memory items in the DMM.

For knowledge representation, we have adopted the graph-based of LTM$^{\mathrm{C}}$ described in [SBB06]. There are two basic types of nodes in the knowledge graph: concept nodes and association nodes. Edges between nodes generally have no other meaning than describing a general relationship between two nodes. The statement “The KIT is located in Karlsruhe” would be encoded in three nodes as shown in figure 3.4. The advantage of modeling associations as nodes and not as edges in the graph is twofold: First, this approach allows associations between more than two concepts without introducing multi-edges to the network structure. Second, treating associations as nodes allows to also stimulate and query them with the same methods which are used for concepts.

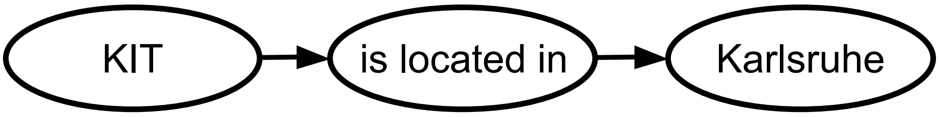


Figure 3.4 – Example of an association in DMM. Two concepts and an associations of the type “is located in”

DMM is able to import data from different large-scale knowledge sources to populate its network with nodes and edges. This enables the system to represent common knowledge to cover many of the concepts which are relevant in general purpose interaction systems that deal with a large variety of domains. Currently, DMM supports the import of the ConceptNet [SH13] database and of OpenCyc [MCWD06]. ConceptNet is a large semantic graph accumulated from different knowledge sources, including ontologies handcrafted by language experts as well as crowdsourced data from serious games. OpenCyc was manually designed as a common knowledge database for natural language understanding applications. In most cases, the ConceptNet data fits better with the associative scenario in DMM. This is not surprising since in the original method of data collection for ConceptNet, users were asked about associations between concepts (see [LS04]). In contrast, OpenCyc is mainly an ontology – there are few associations that are not of the type “is a”. Many associations in OpenCyc represent internal metadata (e.g. “Wn_20_synset_Germany_noun_1”) which cannot be easily mapped to a concept or association. However, there are areas of knowledge (e.g. specific people) which are not covered by ConceptNet and for which OpenCyc is a better choice.

Figure 3.5 illustrates some of the differences between OpenCyc and ConceptNet. It shows a small but representative subset from the neighborhood of the node “Germany” in both OpenCyc and ConceptNet (the full neighborhoods contain 150 and 70 nodes, respectively). The concepts and associations in OpenCyc are very formal, many of them describing generalizations and specializations of concepts (“broaderTerm”, ”instanceOf”). Some associations are very specific and specialized (“CertainDistantCountriesWithInterests...”). In comparison, the associations of ConceptNet are more colloquial (“good beer”), of greater variety in the types of association (“PartOf”, “hasA”, etc.) and more general (“person AtLocation Germany”). Some basic information (“europe” vs. “EuropeanUnion”, “country” vs. “WesternEuropeanCountry”) is contained in both graphs.

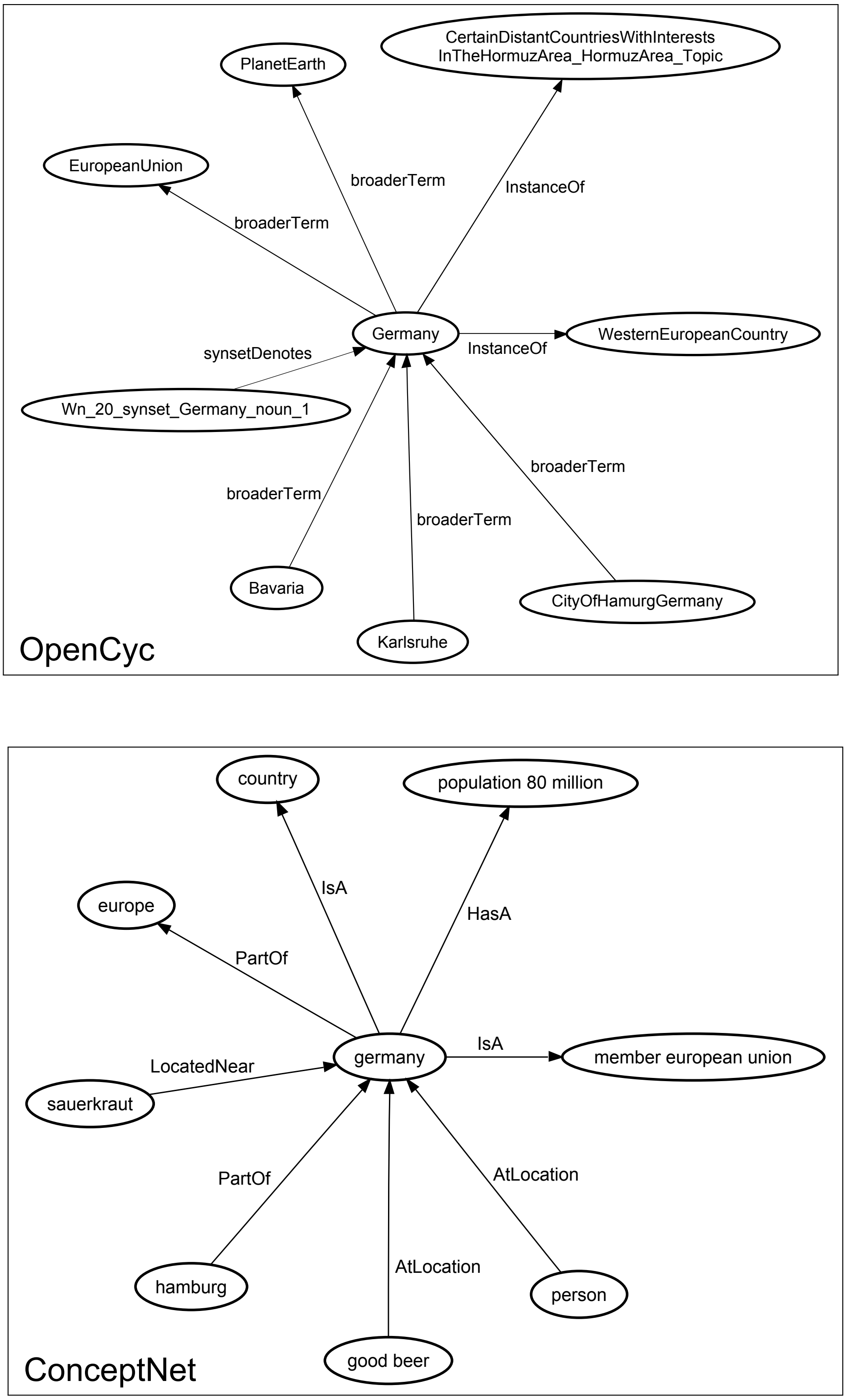


Figure 3.5 – The node "Germany" and some of its neighbors in OpenCyc (top) and ConceptNet (bottom). The names of nodes are shown as they appear in their respective database [Pro11].

Memory Dynamics

Each node in DMM has an activation value. This value determines the likelihood of a node to be retrieved if requested. It also serves as an indicator for the amount of time it takes for the node to be retrieved from memory. Activation of a node can increase in two ways: By an external stimulus or through spreading activation, i.e. the propagation of activation from activated nodes to associated nodes. A typical example for an external stimulus is the system mentioning an item during an interaction. However, a stimulus does not always have to be verbal. An object coming into view could provide a stimulus (to include such stimulus in the model, the occurrence of the stimulus needs to be detected by the model, for example by proximity estimatiob from GPS). Also, previous knowledge or events can influence node activation.

The model behind DMM is building on the approach of [SBB06] for the extension of the ACT-R declarative memory model. Calculation of base level activation and noise is identical to the ACT-R theory presented in Section 3.2.8. Spreading activation is the main mechanism used in DMM to trigger associations. It is implemented as a depth first traversal of the graph, starting with the set of externally stimulated nodes: each node n spreads part of its activation to every node linked to n, which in turn spread part of their received activation. The total amount of spreading activation $\mathrm{ta}(n)$ a node n receives from its predecessors $P(n)$ is calculated by adding the partial spreading activation $r(n, n_{\mathrm{pred}})$ received from each predecessor n_{pred}:

$$\mathrm{r}(n, n_{\mathrm{pred}}) = \frac{\mathrm{ta}(n_{\mathrm{pred}}) * f_{\mathrm{damp}}}{N(n_{\mathrm{pred}})} \tag{3.8}$$

$$\mathrm{ta}(n) = \sum_{n_{\mathrm{pred}} \in \mathrm{P(n)}} \mathrm{r}(n, n_{\mathrm{pred}}) \tag{3.9}$$

$N(n_{\mathrm{pred}})$ in Equation 3.8 denotes the number of neighbors of node n_{pred}. Spreading stops once the amount of activation to be spread from a node falls below a threshold τ_{spread}. This is necessary to keep the model computable, but it also follows the all-or-nothing principle in the human nervous system: neurons only transmit a signal if their received signal strength is above a certain threshold.

The free parameter f_{damp} can be used to restrain the activation spreading. It can be thought of as a measure of creativity in free association - higher values result in more associations with less direct links to the stimulated input. The amount of activation which a node spreads is reciprocal to the number of its neighbors. This models the fan effect [AR99], which describes the strength of associations.

The original LTM^C model has no notion of temporal dynamics regarding spreading activation. This is a serious limitation when modeling human memory during HCI. Over the course of one interaction, new items will be stimulated while the activation of old items will fade. In order to keep activation values realistic over time, we introduce a decay mechanism for spreading activation. We use the entire activation history of a node (resulting from stimulation and spreading) to calculate its activation at a given time. The total dynamic spreading activation $\mathrm{A}_n(t_{\text{current}})$ of a node n at the time t_{current} is given by the equation:

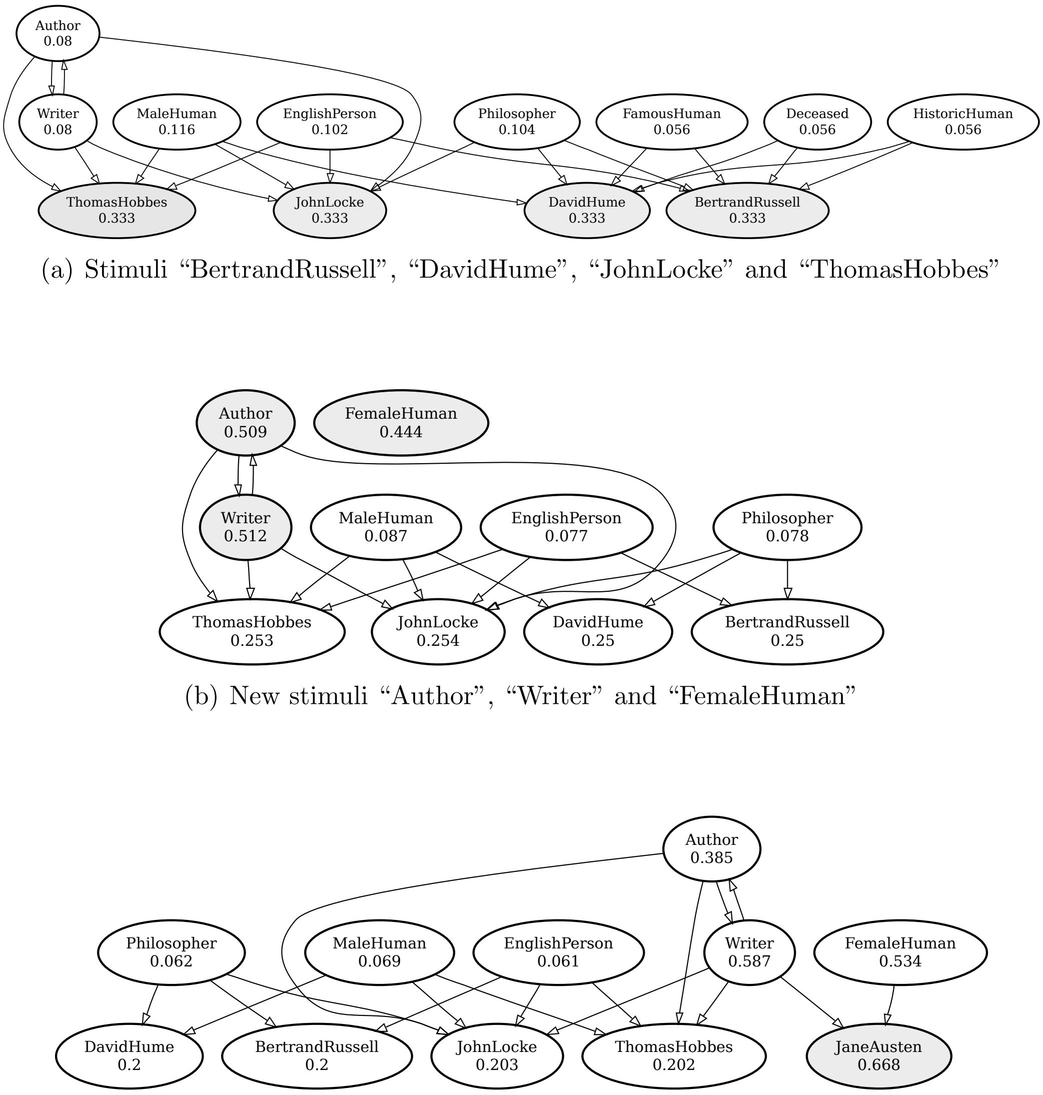
$$\mathrm{A}_n(t_{\text{current}}) = \sum_{t \in \mathrm{H}_n} (z(t_{\text{current}} - t) * \mathrm{AH}_n(t)) + \mathrm{ta}(n) \tag{3.10}$$

H_n is a set of timestamps at which the node n was active (i.e. is part of the connected component of the graph with the highest overall activation). $\mathrm{AH}_n(t)$ ("activation history") returns the total spreading activation value of the node n at the time t. This equation is not recursive – the activation history contains only total spreading activation values ($\mathrm{ta}(n)$) from different points in time, not the dynamic spreading activation. $z(x)$ is defined as:

$$\begin{aligned} z(x) &= 1, && \text{for } x < 0 \\ z(x) &= \tfrac{1}{x+1}, && \text{for } x \geq 0 \end{aligned} \tag{3.11}$$

$z(x)$ decreases almost linearly for small values of x (i.e. t is close to t_{current}) and asymptotically approaches 0 for large values of x. Because it is multiplied with the activation history, this means that the total activation of items which are not recently stimulated will drop fast, ensuring that newly stimulated items have a higher activation. Items that have not been stimulated for a while will have a very small activation but are still distinguishable from items that have never been active. Figure 3.6 shows an example of how activation evolves over three spreading iterations.

Retrieval probability (which predicts the likelihood that an item can be retrieved successfully) and retrieval latency (the response time to a memory request) of an item are calculated using the original formula from ACT-R, see Section 3.2.8.



(a) Stimuli “BertrandRussell”, “DavidHume”, “JohnLocke” and “ThomasHobbes”

(b) New stimuli “Author”, “Writer” and “FemaleHuman”

(c) New stimulus “JaneAusten”

Figure 3.6 – Results of three consecutive stimulations. The numbers indicate the resulting spreading activation value. This could be part of a conversation about English authors. At first, four specific people were stimulated (colored gray). Next, three categories were stimulated, one of which (“FemaleHuman”) did not fit for any of the previous stimuli. The final stimulus is another specific person that is part of the new category. The effects of introducing new activation through stimulation is counteracted by the decay of the activation history. In this example, OpenCyc was used. All associations are of the type “IsA” [Pro11].

3.3.2 Implementation

DMM is implemented as a Java library, so it can be used by Java applications and as a standalone server accessible remotely by any application, e.g. speech processing systems or a dialog managers. It is designed to accommodate a variety of different implementations for its internal network representation. The central interfaces are `Network` and `Session`.

A `Network` object represents the knowledge graph. There are currently importers for OpenCyc and ConceptNet, but a `Network` can also be manipulated directly through the API, e.g. to create a customized, application-specific graph. Two different implementations of `Network` can be used: `POJONetwork` provides a fast implementation where all data is stored as Java objects in memory for platforms with sufficient main memory (around 1 Gigabyte for OpenCyc or ConceptNet). With `HGDBNetwork`, a slower implementation with a much smaller memory footprint based on the HypergraphDB [Ior10] database is available.

The `Session` interface provides a mutable view on a `Network`. It mainly contains activation histories for all nodes. Multiple `Session` objects can operate on the same `Network` object simultaneously, e.g. to maintain different hypotheses of the memory state. An implementation of the `Session` interface contains the dynamic processes that operate on the knowledge stored in the knowledge graph. For example, the algorithm we have described in section 3.3.1 is contained in an implementation of `Session` named `SessionSpread`. Other `Session` implementations are also available.

3.3.3 DMM Evaluation

To evaluate the DMM for interaction applications, we look at two aspects of its performance. First, we perform a quantitative evaluation of a user study comparing human associations to the associations generated by DMM. Second, we simulate a conversation of two instances of DMMs to qualitatively evaluate their dynamic behavior over time.

Evaluation of Associations

DMM is supposed to work as a model of the human associative process. If DMM performs optimally, it should provide associations identical to human associations. Therefore, we compare in our evaluation the DMM results to human associations. For this purpose, we conducted a questionnaire on associations to certain stimulus words and compared the answers to the result of a single spreading process. The ConceptNet database was used in all evaluations. The entire database was loaded[7], leading to 320000 nodes and 480000 links. ConceptNet was chosen over OpenCyc because of its closer relation to common human associations, which is what we are evaluating here.

To compare the associations made by DMM using the ConceptNet database to those of humans, we developed a questionnaire and asked 20 people to fill it out. All participants were students or employees of the KIT between 20 and 30 years of age and all participated in the same week. None of them was an English native speaker. The questionnaire included five sets of three related stimuli and participants were asked to write down their first two associations for each set. We then activated the same sets of stimulus words as concepts in DMM in order to compare the results with the answers of the participants. The presented stimuli were:
a) go restaurant, fork, diminish own hunger
b) tennis, soccer, volleyball
c) germany, france, spain
d) hamster, dog, cat
e) pen, work, desk

To evaluate if our spreading activation implementation using the ConceptNet data returns plausible results, we checked if the most frequent answers by humands were reflected by the nodes with the highest activation. The results are listed in Table 3.1. It shows in the first column the different sets of stimuli and in the second column all associations that were given by more than one person, ordered by frequency. For each association, it also contains the number of participants which reported this association. Column three contains its activation rank in the results of DMM ("-" indicates that the concept was not activated at all or that its activation was negligible). If the DMM output contained a different, but semantically very similar concept (e.g. "hungry" vs. "hunger") with a high rank, we also included it. We expected that the ordering of human associations by number of people corresponded roughly to the ordering of DMM associations by activation rank.

Overall, the results are very encouraging. Especially for the top entries of each set, we see a large overlap between the result of DMM and the user replies: Each association shared by at least a third of the participants was also highly activated in DMM. In most cases, the first items on both ordered

[7] ConceptNet 5, downloaded from http://conceptnet5.media.mit.edu/downloads

Stimuli	**Association (# people)**	**DMM Rank**
go restaurant fork diminish own hunger	**eat** (7)	**1**
	food (6)	**6 (eat food 2)**
	plate (3)	**4**
	hunger (2)	- (hungry 10)
tennis soccer volleyball	**sport** (9)	**1**
	ball (5)	**15 (ball sport 5)**
	basketball (3)	**-**
	team (2)	- (team sport 4)
	television (2)	-
	play (2)	- (play volleyball 8)
germany france spain	**europe** (14)	**2**
	country (8)	**1**
	italy (5)	**-**
	language (3)	**-**
	greece (2)	-
	holiday (2)	-
hamster dog cat	**pet** (12)	**1**
	animal (5)	**4**
	rabbit (2)	-
	mouse (2)	- (rat 2)
pen work desk	**write** (6)	**1**
	office (5)	**3**
	paper (5)	**2**
	university (4)	**-**
	computer (3)	**6**
	school (2)	10
	chaos (2)	-
	money (2)	-

Table 3.1 – Comparison of human associations and DMM output. The names of stimuli and associations are shown as they appear in the ConceptNet database. The maximum total number of associations for each set of three stimuli is 40 (2 associations for 20 participants).

lists were the same. There are a few interesting observations: Some of the associations of the participants did not fit with all of the stimuli – they seem to be associated with only one or two of the stimuli items. This behavior is also reflected by the DMM output (e.g. "team" / "team sport" does not fit very well for "tennis"), which indicates that DMM also reproduced this unexpected behavior. We also note that not all top associations by humans are generalizations of the given stimuli (e.g. "writing" or "paper" for the last set), which means that an ontology (i.e. a knowledge graph which only consists of "isA" relations) is not enough to cover all relevant associations. As we used the ConceptNet database (which in contrast to the OpenCyc database contains many associations besides "isA' relations) to populate DMM, this effect could also be reproduced.

The vast majority of associations mentioned by participants consisted of only one word, while DMM concepts sometimes include multiple words (this is especially obvious with the *sports* stimuli). When presented a stimulus consisting of multiple concepts of the same type, the participants often associated another example of the type (like "basketball" for *sports* or "italy" for *countries*) - these types of associations do not seem to be represented well in DMM, as they require at least two spreading steps (e.g. from "germany" to "europe" to "italy", as no direct link exists).

There are some external factors that can influence a study like this. Among the most important ones is language. Since we used the English version of the ConceptNet database (there are versions in other languages available, but they are much smaller), we conducted the study in English. However, none of the participants were English native speakers. Depending on the individual language proficiency of the participants, this could affect the results – especially if one has to translate the stimuli, associate in one's native language and then translate back the associations.

Another consideration is the order of the items. The order of the sets and inside of the sets depicted in Table 3.1 is the same as in the questionnaire. One participant had the association "food" for the *pets* stimuli, which could be attributed to priming by the earlier *eating* related stimuli. The order in which the three stimuli of each set were given could influence the resulting associations as well. The order in all questionnaires was the same. The DMM queries did not incorporate order considerations: Each set of stimuli was given synchronous and without previous activation from other sets because the dynamic aspects of `SessionSpread` have a much stronger impact on the results than the more subtle effects described here.

Overall, we can conclude that DMM provides a reasonable prediction of human associations using its network built from a large-scale semantic knowledge base. It should be noted that this evaluation of DMM associations is not a realistic representation of associations in a complex HCI scenario, which may contain a temporal ordering of stimuli, embedding of stimuli in a conversational context or multiple intertwined categories of concepts. However, it still validates the general relation between human associations and the DMM prediction thereof.

Evaluation of Conversations

In order to evaluate the dynamic behavior of DMM, we tested it in an evolving context. We decided to simulate a "conversation" since this is a common example for a possible usage of DMM for HCI applications. To test if the course of the conversation remains realistic when all associations in a dialog are generated by DMM, we used two instances of DMMs which communicated with each other.

Initially, we stimulated both instances with the same concepts to create a common ground. To initiate a conversation, we randomly chose one activated concept that was not mentioned before. This represented the concept which one instance wanted to add to the conversation. To ensure that no irrelevant items were selected, we only chose from the five items with the highest activation. The probability of a concept to be selected was proportional to its activation. The concept was selected from one instance and activated in the other, where spreading was calculated. By iterating this process for both DMM instances, a simple dialog on a concept level was generated. Examples of such dialogs on a concept level are shown in Tables 3.2 and 3.3. These conversations could have started with a discussion on fruits, leading to the stimulation of the concepts "apple" and "orange". The left column contains the concepts that were selected from the first instance and stimulated in the second, the right column shows the concepts that were selected from the second instance. Table 3.2 shows a conversation going back and forth without outside intervention. We see that each topic of the conversation is associated with concepts issued by both collocutors (e.g. "lemonade" associated to "lime" from instance 1 and "drink" from instance 2). While the most recent concepts have the strongest influence on topic selection, the more distant discourse is still relevant to the interaction (e.g. "drink" influenced by the initial "juice"). In Table 3.3, we introduced new external stimuli, which were activated in both DMM instances, halfway through the dialog. The new stimuli “tropic” and “island” could for example be caused by passing a holiday advertisement during the conversation, leading to a topic shift. The rest of the conversation is influenced by the newly introduced topic. Note while the initial topics for both conversations were identical, the stochastic nature of the selection process lead to different interactions already before the introduction of the new external stimuli.

We can see that the majority of the concepts mentioned in the conversation are related to the originally stimulated items, i.e. we maintain a coherent conversation. Still, we see a gradual shift of topic. This is especially true for the second example, where the new stimuli strongly influence the course of the interaction. Note that the new items do not simply override the old ones. The spreading process supports items which are associated to both the old and the new concepts, e.g. “tropical fruit”. These examples indicate that the DMM is indeed able to generate coherent interactions over a period of time, including the handling of topic drift. It has to be noted that the quality of the conversation depends on the quality and quantity of relevant nodes in the knowledge base. For certain domain specific parts of an application, it will be necessary to manually extend the existing entries, since generic State-of-the-Art databases might not be rich enough.

3.3.4 Discussion

In this section, we described the design and implementation of the dynamic memory model (DMM) for application in interaction systems. The DMM is based on the validated theory of ACT-R and LTM^c and extended by the ability to handle memory dynamics and to populate the model from large semantic databases. The evaluation showed that the model is able to predict associations close to those produced by humans and showed that the model also maintains plausible associations during a simulated interaction with potential topic shifts. The presented model as well as the conducted validation experiments from an HCI perspective are novel contributions to the research community.

We note that the evaluation does not yet cover all aspects of associations during HCI. For example, the simulated interactions take place on a semantic level, which avoids challenges of automatic language understanding, which is a large research area on its own [DMBHT$^+$08], which provides tools to extract meaning from text. Additionally, both instances of the simulated interactions operated on the same semantic network. This design avoids grounding challenges which occur situations in which two agents (humans

Table 3.2 – DMM conversation

Instance 1		Instance 2
Initial stimuli: apple, orange		
juice		
	↘	
		lemon
	↙	
lemon butter		
	↘	
		fruit
	↙	
sour		
	↘	
		store
	↙	
bitter		
	↘	
		sugar
	↙	
lime		
	↘	
		drink
	↙	
lemonade		
	↘	
		refrigerator
	↙	
seed		
	↘	
		citrus fruit
	↙	
pineapple		
	↘	
		yellow
	↙	
bergamot		
		yellow fruit
	↘	
pear		
	↙	
		sweet

Table 3.3 – Topic switch

Instance 1		Instance 2
Initial stimuli: apple, orange		
juice		
	↘	
		tangerine
	↙	
citrus fruit		
	↘	
		lime
	↙	
like orange		
	↘	
		eat orange
	↙	
peel orange		
	↘	
		outside
	↙	
sweet		
	↘	
		bergamot
	↙	
New stimuli: tropic, island		
ficus		
	↘	
		pineapple
	↙	
tropical fruit		
	↘	
		person
	↙	
peel first		
	↘	
		banana
	↙	
yellow		
	↘	
		mango
	↙	
salsa		
	↘	
		fun

or systems) operate on different knowledge bases, for example when user and system have different levels of knowledge about the topic of a conversation. As the employed database of the DMM is not hard-wired but can be exchanged flexibly, it would be possible to explore such situations.

3.4 Overlay-based Modeling of Workload

In the previous section, we introduced DMM, a dynamic, large-scale memory model. The ACT-R model which provided the theoretical foundation for DMM was designed to match average performance of average humans who fully concentrate on one given task. However, human memory performance depends to a large extend on the current workload level: A higher workload level – for example induced when humans perform several tasks at the same time – influences information retrieval, activation decay, etc. When applying a standard computational cognitive model with default parameters to predict performance in a cognitive task under high workload, the prediction of memory performance will be overly optimistic. This is because such a model does not account for the effects of workload on human cognition. When using a standard model to predict human behavior during interaction, this imprecision will lead to wrong assumptions about what past information (e.g. from the system or the context) the user might be able to recall. In the following two sections, we will present two approaches to model the impact of workload on cognitive performance. For each approach, we present a user study for validation.

In this section, we start with the approach of direct manipulation of the DMM parameters to explicitly represent different workload levels[8]. In Section 3.5, we present an alternative implicit approach using dummy models. To our best knowledge, the systematic implementation and validation of different approaches to represent the impact of workload on cognition in computational models is a substantial novel contribution to the research community.

[8] This section is based in parts on the Bachelor thesis of Lucas Bechberger which was supervised by the author of this thesis.

3.4.1 Approach: Explicit Representation of Workload Levels as DMM parameters

In this subsection, we introduce our approach of explicit representation of workload levels in the DMM. First, we introduce the general idea of workload-dependent parameter sets. Second, we introduce the task for which we evaluate the approach and specify the memory model internals. Third, we explain how the memory model is populated with data from the Wordnet database. Fourth, we describe how the workload-dependent parameter sets are optimized by a genetic algorithm.

Workload-Dependent Parameter Sets

One approach to model the impact of high workload on cognition is its direct, explicit representation in the DMM (as described in Section 3.3) by adding new degrees of freedom to the memory model. Of the DMM described in Section 3.3, we regard the following parameters: decay d, retrieval threshold τ, latency factor F, spreading potential P_{spread}, spreading threshold τ_{spread}, and retrieval sensitivity s. Since F depends on τ [ABB$^+$04], and τ_{spread} depends on P_{spread} as well as $B(i)$, five free parameters remain to model performance differences between workload levels. For this purpose, we follow the overlay idea of [RVRAS06] to represent cognitive modulators: To enable the model to exhibit different memory performance under different workload conditions, we extend the model to provide different parameter sets, one for each workload level. Each workload-dependent parameter set comprises the five parameters of the DMM listed above. When a certain workload level is detected (e.g. by using an empirical workload model), the associated parameter set is selected, which influences retrieval probability and latency of the affected memory items.

While the implementation of this approach is straight-forward, determining the parameter sets and evaluating their impact on performance is challenging: The memory model is complex and contains non-linear dependencies of activation on previous stimulations of the target item and of associated items. Additionally, the model predicts multiple dependent variables (e.g. retrieval probability and retrieval latency) which may have to adapt differently to changes in workload level. Therefore, we provide a detailed evaluation of the impact of workload on memory items in different situations: This concerns the recency and frequency of stimulation (i.e. how often and how long ago was the item stimulated?) as well as the relevance of spreading (i.e. does the item appear in the context of semantically similar items?). For all those different situations, we analyze and quantize to what extend it is possible to model changes of memory performance using workload-dependent parameter sets. We used optimization based a genetic algorithm to determine the optimal parameter sets for each workload level. This is done by fitting the parameters for each set to behavioral human data from the respective workload level.

The proposed approach makes little assumptions on the relation of workload level and memory performance. An alternative to a finite number of parameter sets would have been to use a continuous workload parameter $w \in [0, 1]$ to represent gradual shifts in workload level and a mapping from w to the model parameters. However, prior to this study, there was no knowledge available about which free parameters are influenced (linearly or non-linearly) by different workload modes. Thus, the resulting model would either by oversimplified or not falsifiable given a corpus with a limited number of discrete workload levels. Furthermore, sensitivity analysis of existing workload measurements is usually performed using discrete task difficulty levels [RDMP04], i.e. it is unclear to what extend a continuous workload scale could be measured reliably. For those reasons, we decided to follow the described approach of distinct parameter sets.

Memory Model Specification

In this section, we introduce the task for which we evaluated the approach and specify the memory model internals. We evaluated the explicit representation of workload levels using a verbal recognition task, in which participants were presented a list of words to remember ("encoding phase"). Afterwards, participants saw a list of target and distractor words and had to indicate for each word whether it was on the learning list ("recognition phase"). This task was designed in analogy to the Hopkins Verbal Learning Test (HVLT) [Bra91]. To induce different levels of workload, two different versions of the switching task [Mon03] were used, an easy and a difficult one. To disentangle effects of distraction during recognition from the impact of workload on the encoding, the switching task was only activated during encoding.

For this task, we tried to predict different behavioral performance metrics by employing the adaptive memory model as outlined in the previous subsection. When new items were learned, they were stimulated in the memory model, i.e. they received and spread activation. During recognition, this activation was used to calculate the retrieval probability of the queried item. If above the retrieval threshold, the item was reported as known by the model (i.e. the model predicted that the simulated participant would indicate the item as a target word). Otherwise, it was reported as unknown (i.e. the model predicted that the simulated participant would indicate the item as a distractor word). The memory model we used in this evaluation was identical to the DMM introduced in Section 3.3. The ACT-R equation for base level activation leaves the unit of time as a degree of freedom for the modeler. For the DMM, we measured times t_k and t_c in minutes since the start of the experiment. To provide all memory elements with a reasonable initial activation value, all elements were assumed to be encoded once at time $t = -60$ min, i.e. at a time out of the scope of the experiment. This prevented the activation value from becoming negative infinity when no stimulation was presented, which is implausible in our scenario. Since base level activation can be negative, but only positive values should be spread through the memory item network, a *spreading potential* P_{spread} is added to the base level activation and this sum is multiplied with the factor 10. If the resulting value $10 \cdot (B(i) + P_{spread})$ is higher than τ_{spread}, it will be spread amongst the memory element network. To reduce the number of free parameters, like in LTM^C, the spreading threshold is set relatively to the initial activation to be spread.

For calculating the base level activation $B(i)$ of a memory item, the three workload parameter sets interact: For each summand $(t_c - t_k)^{-d}$ in Equation 3.2, the decay parameter $d = d_k$ is taken from the workload parameter set which corresponds to the workload level at encoding time t_k. For example, if a memory item was encoded at t_0 when the workload level was Low, and at t_1 when the workload level was Medium, then the base level activation is computed as $B(i) = \ln((t_c - t_0)^{-d_{Low}} + (t_c - t_1)^{-d_{Medium}})$ with d_{Low} and d_{Medium} being the decay parameters of parameter sets Low and Medium, respectively. This approach allows us to model variable workload conditions during encoding of different stimulations. Spreading activation, retrieval probability and response time are calculated independently of the other parameter sets, as they do not have a temporal component. They are still calculated depending on the current workload level.

Memory Model Population

This section describes how the memory content of the DMM was defined to create an appropriate memory representation for the described task: The Wordnet database [Mil95] was used to populate the memory model as described in Section 3.3. WordNet is a lexical database for the English language which has been developed at Princeton University. WordNet contains a large number of nouns, verbs, adjectives and adverbs, all grouped into sets of synonyms called *synsets*. For the scope of this section, only nouns were considered. Synsets are connected with other synsets via semantic pointers. The relations between noun synsets include hypernymy/hyponymy ("is-a" relation), meronymy/holonymy ("has-a" relation) and antonymy ("opposite-of" relation). The words the model operated on are German nouns as we conducted the experiment with German native speakers. Therefore, all terms were translated and each term was associated to exactly one synset to avoid ambiguities caused by translations from or to words with multiple meanings.

Genetic Optimization

In order to optimize the model parameters to fit the different workload levels, we used a genetic algorithm. We maintained one population of size 100 of possible parameter configurations for each workload mode. Each parameter configuration consisted of values for decay d, intercept time I, retrieval threshold τ, spreading potential P_{spread}, and retrieval sensitivity s. The fitness value, which is required to determine the surviving configurations in each iteration, was obtained by performing a simulation with this parameter configuration and by comparing the simulation results with the results of the human participants. The final fitness value was then computed by adding up the negative relative errors for the three dependent variables response time, hit rate and false alarm rate. Response time is the time between the display of a word in the recognition phase and the corresponding response by the participant. Hit rate is the relative frequency of correct responses to target words. False alarm rate is the relative frequency of incorrect responses to distractor words.

To generate a new population from a given one during one iteration, the genetic algorithm mated two randomly chosen parameter configurations and mutated individual parameter configurations. Mating of two parameter configurations was implemented by selecting a random subset of parameters from one parent and combining them with the missing parameters taken from the other one. The resulting configuration was then added to the population. Mutation of a parameter configuration was implemented by selecting a random subset of parameters from a parent and randomly varying the values of the selected parameters. The resulting parameter configuration was then added to the population. The genetic algorithm terminated after a 1000 iterations.

3.4.2 Experiment Design

In this subsection, we describe the experiment which was conducted to validate the proposed approach. The experiment consisted of a verbal learning task which is modeled by the DMM and a secondary task to generate three different workload levels. First, we describe the verbal learning task. Second, we present the structure of one block of this task. Third, we introduce the secondary task to induce different workload levels. Fourth, we describe the overall structure of the conducted experiment and finally, we present information on the collected data corpus.

Verbal Learning Task

In the following, we present details about the concrete implementation of the verbal learning task. To validate the representation of workload level by variable parameter sets, we conducted an experiment in which participants were asked to perform the following verbal memory task: During a learning phase, a list of five or eight nouns was presented one at a time on a computer screen (1.5 seconds per word with a short break of 0.5 seconds between words). The participants were asked to memorize these words. Immediately after each learning phase, a suppression task was started: Participants were asked to count down in threes from a random three-digit number presented on the screen (e.g. starting from 328, participants had to count down: 328, 325, 322, 319, . . .). This suppression task was adopted from [FM00] and had the purpose of reducing recency effects during the subsequent recognition phase. After 20 seconds of counting down, participants were alerted by an audio signal, indicating the start of the recognition phase: A list of eight nouns was presented on the screen one by one and the participants had to decide for each word whether they had learned it before in one of the previous learning phases. If they recognized the word, they had to press “Y”, otherwise they had to press “N”. Participants were told to respond as fast and as accurately as possible. After pressing “Y” or “N”, the next word appeared on the screen. Both response time and response accuracy were recorded. Half of the presented words were target words, which had been learned before, the other half were distractor words. Participants were not told how many target words there would be in a recognition phase. Each word (on both, the target and distractor lists) was queried only once during the experiment to avoid unintended learning effects during the recognition phase. The three phases (learning – suppression task – recognition) were repeated six times, forming one *block*. A block was executed with a consistent workload level which only switched between blocks. After each recognition phase within a block, there was a break of ten seconds.

Block Structure

Each block used a word list which contained 54 words (24 target words, 24 distractor words and 6 filler words that were learned but never retrieved[9]). The first four learning phases of a block consisted of eight words, the last two learning phases of a block consisted of five words (note that some 24 unique target words were presented twice). Participants were informed that no retrieval was required across blocks, i.e. words learned in block i were not queried in block j (for $i \neq j$). However, retrievals were required across different phases of the same block. All words used in this experiment were German nouns with a mean length of 5.82 letters (SD: 2.99, range: 2–12) and a mean number of syllables of 1.89 (SD: 0.48, range: 1–4). Half of the 24 words learned during a block were presented for learning only once, half of the words were presented in two subsequent learning phases (this is the "reinforcement" property of a word). Figure 3.7 shows the structure of the described word recognition task.

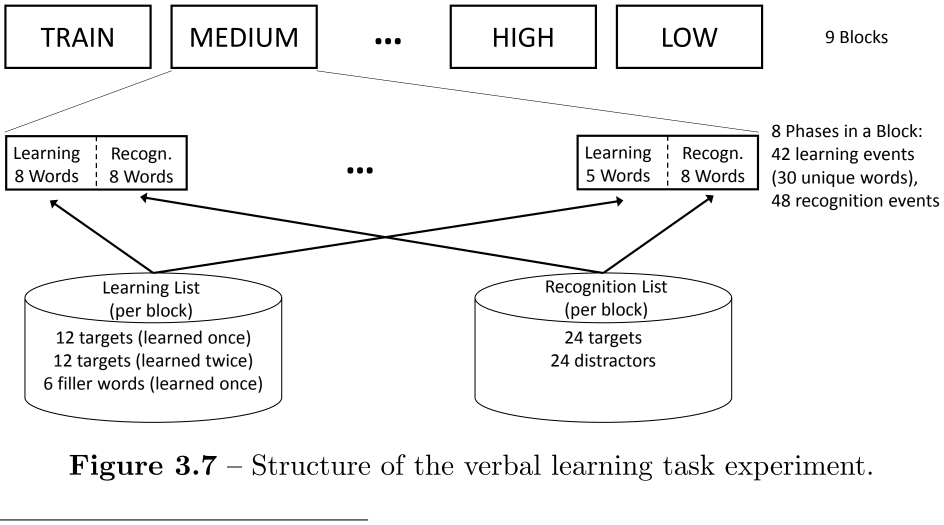


Figure 3.7 – Structure of the verbal learning task experiment.

[9] This prevented an elimination strategy to identify distractor words when all target words were already queried.

Half of the words learned during a block were presented in the recognition phase right after they had been learned, half of the words were queried with a delay of one recognition phase ("gap" property). Half of the 48 words retrieved in each block were so called "cluster words": There were four clusters per block and each cluster consisted of six semantically related words (e.g. cat, dog, cow, horse, lion, tiger; this is the "cluster" property). Half of the words of each cluster were used as target words, half of them were used as distractor words. The clusters and their words were taken from [BM69]. Words which do not belong to a cluster are called singleton words. We call each combination of "cluster", "gap" and "reinforcement" property a *cell*. The three properties "cluster" (cluster words vs. singleton words), "reinforcement" (word learned once or twice) and "gap" (immediate recognition vs. delayed recognition) were counterbalanced, so that all cells (e.g. cluster words that were reinforced twice and retrieved without a gap) were of equal size.

Secondary Switching Task

To induce different workload levels, two secondary tasks based on the Switching task paradigm were used, similar to those mentioned in [Mon03]: In the easy variant, a randomly chosen sound file, containing a digit, was played concurrently with each word appearing on the screen during a learning phase. Participants were asked to report verbally for each digit, whether it was "large" or "small". A digit was considered to be "large", if it was greater than or equal to 5, and "small" otherwise. Participants were asked to report their answer as accurately as possible before the next word was presented on the screen and the next sound file was played. In the difficult variant of the Switching task, again a randomly chosen sound file containing a digit was played concurrently with each word appearing on the screen during a learning phase. Participants were asked to report verbally alternating whether the digit was "large" or "small" and whether the digit was "odd" or "even", starting with the former. For example for the digit sequence "three", "eight", "nine", "one", the correct responses would have been "small", "even", "large", "odd". Again, participants were asked to report their answer as accurately as possible before the next word was presented on the screen and the next sound file was played. Participants were told that the verbal learning task and the Switching task were of equal importance. Figure 3.8 illustrates how the difficult Switching task and the verbal learning task were performed concurrently.

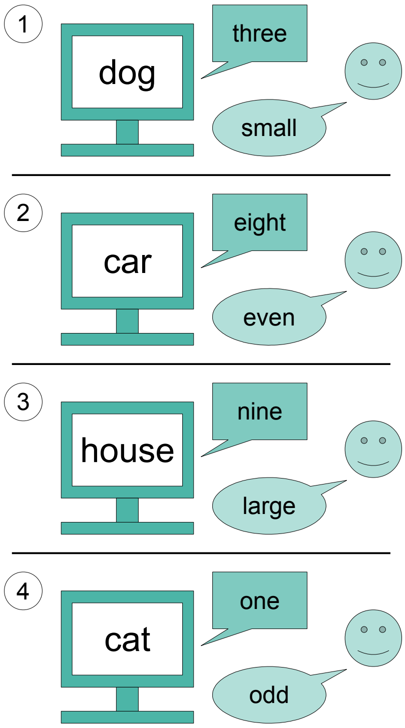


Figure 3.8 – Concurrent execution of both the verbal learning task (words to memorize presented on the screen) and the difficult Switching task (digits to classify presented acoustically by playing sound files) [Bec12].

Both secondary tasks were designed to avoid modality conflicts with the verbal learning task in perception and response: In the verbal learning task, stimuli were presented visually and responses were made manually, whereas in both Switching tasks, stimuli were presented acoustically and responses were made verbally. Since the verbal learning task operated on words, both secondary tasks were designed to operate mainly on digits to avoid conflicts in verbal memory. However, since both verbal learning task and secondary tasks require cognitive resources, cognitive workload should rise when the secondary task was executed. An audio recording was made in order to analyze the verbal responses in the secondary task post-experimentally.

In this experiment, three workload levels were distinguished:

- In workload level Low, participants performed solely the verbal learning task.

- In workload level MEDIUM, participants performed the verbal learning task and the easy Switching task concurrently with equal importance assigned to both tasks.
- In workload level HIGH, participants performed the verbal learning task and the difficult Switching task concurrently with equal importance assigned to both tasks.

Ordering of Blocks

There were nine blocks in total: three training blocks at the beginning (one of each workload level) and six test blocks (two of each workload level). Training blocks consisted of four learning and recognition phases. Three of the learning phases consisted of five words and the last one consisted of three words. Only data from the test blocks was analyzed. Between two subsequent test blocks there was a break of 90 seconds. For the both, training and test blocks, the order of underlying word lists and order of workload levels was counterbalanced across participants. The order of workload levels was randomized but constrained by the following two restrictions: Two subsequent blocks had to be of different workload levels and both the first three and the last three of the test blocks each had to cover all three workload levels.

Collected Data Corpus

The experiment took about 65 minutes and was conducted with 24 participants (mean age: 20.75 years, SD: 3.37, range: 15 – 29). 17 of them were male (mean age: 20.24 years, SD: 2.88, range: 15 – 28), 7 participants were female (mean age: 22.0, SD: 4.32, range: 16 – 29). The majority of 17 participants were students, 4 of them were trainees and 3 were young professionals. Per participant, for each cell (e.g. cluster words that were reinforced twice and retrieved without a gap) and workload level, a total number of 6 data points was recorded per participant. The experiment was run on a MacBook Pro 13" (Intel Core 2 Duo 2.26 GHz, 2 GB RAM) using the PsychoPy framework [Pei07].

3.4.3 Evaluation Results

In this subsection, we present, analyze and interpret the results of the genetic optimization to determine the workload-dependent parameter sets on the collected experimental data. The number of free parameters of the full DMM is large compared to the size of the available data (while we recorded six blocks per participants with 54 words each, words are distributed across eight cells and three workload levels). Therefore, we did not optimize all parameters in one monolithic optimization run, but rather by using a two-step approach: First, we optimize the model without the spreading mechanism. Second, we fix the parameters from step one and add the spreading mechanism. This approach ensures the interpretability of the resulting parameter configurations since only few parameters change during each evaluation step.

We start with basic properties of the DMM derived only from base level activation and than extend this initial model to the full model with all parameters, including spreading activation: In the first step, response time and hit rate are evaluated in dependency of the properties reinforcement and gap, considering only singleton target words. We investigate whether the obtained parameter configurations are able to reproduce the differences in human performance induced by workload. Second, the cluster property will be evaluated by introducing the spreading mechanism (and the corresponding model parameters) and by considering the false alarm rate as performance metric.

Another consequence of the data sparseness is that we train person-independent parameter sets. This is desirable for three reasons: First, the ratio between trial count and parameter count is higher than for person-dependent parameter sets. Second, person-independent parameters can – if we can prove their feasibility – be transferred to unseen participants for which no calibration data is available. Third, a stable parameter set allows a more profound analysis of model plausibility. A disadvantage of this design choice is that for the estimation of individual performance in combination with an empirical workload model, predictions of the person-independent memory model will be less accurate than predictions of a person-dependent model. In Section 3.5, we will look at the alternative of using models which use individualized parameters.

Evaluation of Reinforcement and Gap Property

In this first evaluation step, the model was optimized for reproducing the effects of the reinforcement property and the gap property on task performance for different workload levels. In this part of the evaluation, spreading was deactivated and we only considered singleton target words, to reduce the number of parameters for optimization. The optimization algorithm was run for minimizing the distance of predicted and empirically measured given response time (time between presentation of recognition query and response) and hit rate (relative frequency of correct answers for target words) values. False alarm rate (relative frequency of wrong answers for distractor words) was ignored for this evaluation step because only target words were used. Both, response time and hit rate, were assigned equal priority.

The intercept time I was set to 0.680 s which has emerged as reasonable value in several preliminary tests to represent the average human time for perception and manual response. Intercept time was kept constant across workload levels since it is interpreted as time needed for perception and motor reaction and should therefore not be influenced by cognitive workload. Furthermore, the spreading potential P_{spread} was not optimized since spreading was disabled. Hence, only the decay d, retrieval threshold τ and retrieval sensitivity s were left for optimization. Table 3.4 shows the resulting values after optimization as well as the corresponding fitness values of these parameter sets. The fitness value represents the negated sum of relative errors (response time, hit rate and false alarm rate, see 3.4.1), i.e. the fitness value measures the goodness of fit of the DMM (with the parameters which were determined during optimization) to the experimental data. The precision of the prediction can therefore be quantified be assessing the fitness values: As denoted in Table 3.4, the three parameter sets have a fitness value of -0.068, -0.081 and -0.109, respectively, when evaluated against empirical data from their corresponding workload mode. This means that the average relative prediction error is around three to six percent (relative to human performance) which can be considered a good fit against the experimental data.

By taking a closer look to the plausibility of the parameters, we made the following observations:

- The decay parameter d increases from 0.500 for Low, over 0.505 for Medium up to 0.798 for High. This seems plausible, because a higher value of d results in faster forgetting. Since the hit rate declines under higher workload, faster forgetting under higher workload seems to be a reasonable explanation.

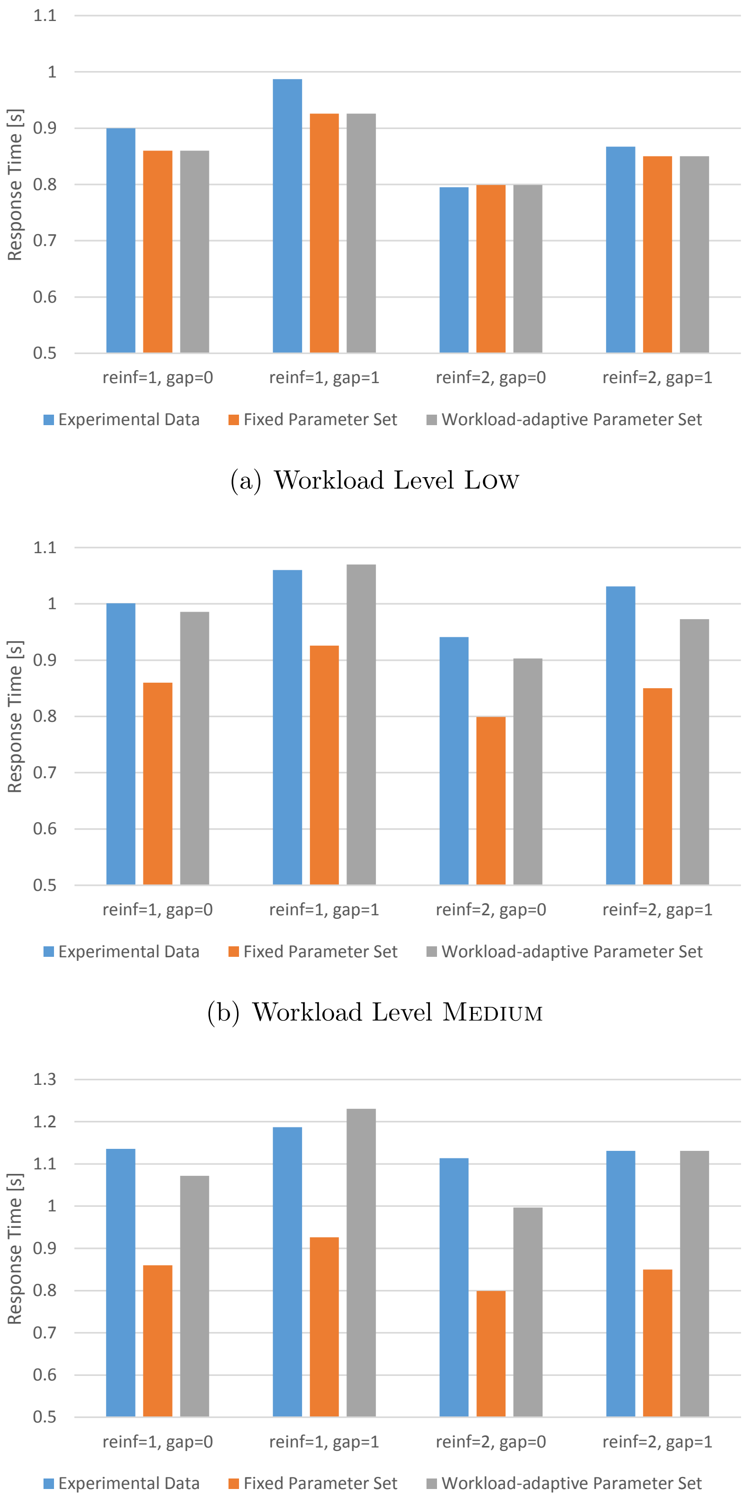
Parameter Configurations For Reinforcement-Gap-Optimization				
Workload Level	d	τ	s	Fitness Value
Low	0.500[†]	-0.337	0.352	-0.068
Medium	0.505	0.031	0.749	-0.081
High	0.798	0.218	0.965	-0.109

Table 3.4 – Parameter configurations for the three workload levels as yielded by the optimization algorithm, and their corresponding fitness values. [†]Note that the decay parameter d was set to 0.5 for workload level Low a priori (since this is the standard value of the ACT-R implementation), and was not optimized.

- The retrieval threshold τ also increases: from -0.337 for Low, over 0.031 for Medium to 0.218 for High. This also seems reasonable: τ determines a threshold value for the activation of a memory item. If activation of a memory item is exactly τ, retrieval probability equals 0.5. Therefore, a higher threshold τ means that items need a higher activation for being retrieved. This appears to be another plausible explanation for the decreasing performance under higher workload.

- The retrieval sensitivity s does also increase: from 0.352 for Low, over 0.749 for Medium to 0.965 for High. This is in line with the previous findings: Since s influences the sensitivity of the retrieval probability function, a value of s that is close to zero will cause the retrieval probability to form a smooth transition between low and high probabilities, whereas a value of s that is close to one will result in a rather sharp transition. When interpreted this way, the parameter values listed above indicate that for higher workload, an increasing activation does not substantially increase retrieval probability until activation approaches τ. The value of 0.352 for workload level Low seems to be plausible since values around 0.4 have emerged as reasonable values for this parameter in the ACT-R community.

Figures 3.9 and 3.10 compare the results obtained in the experiment with the predictions made by the non-adaptive as well as the adaptive model. The analysis differentiate the words regarding the reinforcement and gap properties to study whether the effect of both properties is modeled accurately. The non-adaptive model in this analysis uses the optimized parameters for the workload level Low. In contrast, the adaptive model uses workload-dependent parameter sets. When concentrating on the workload level Low,

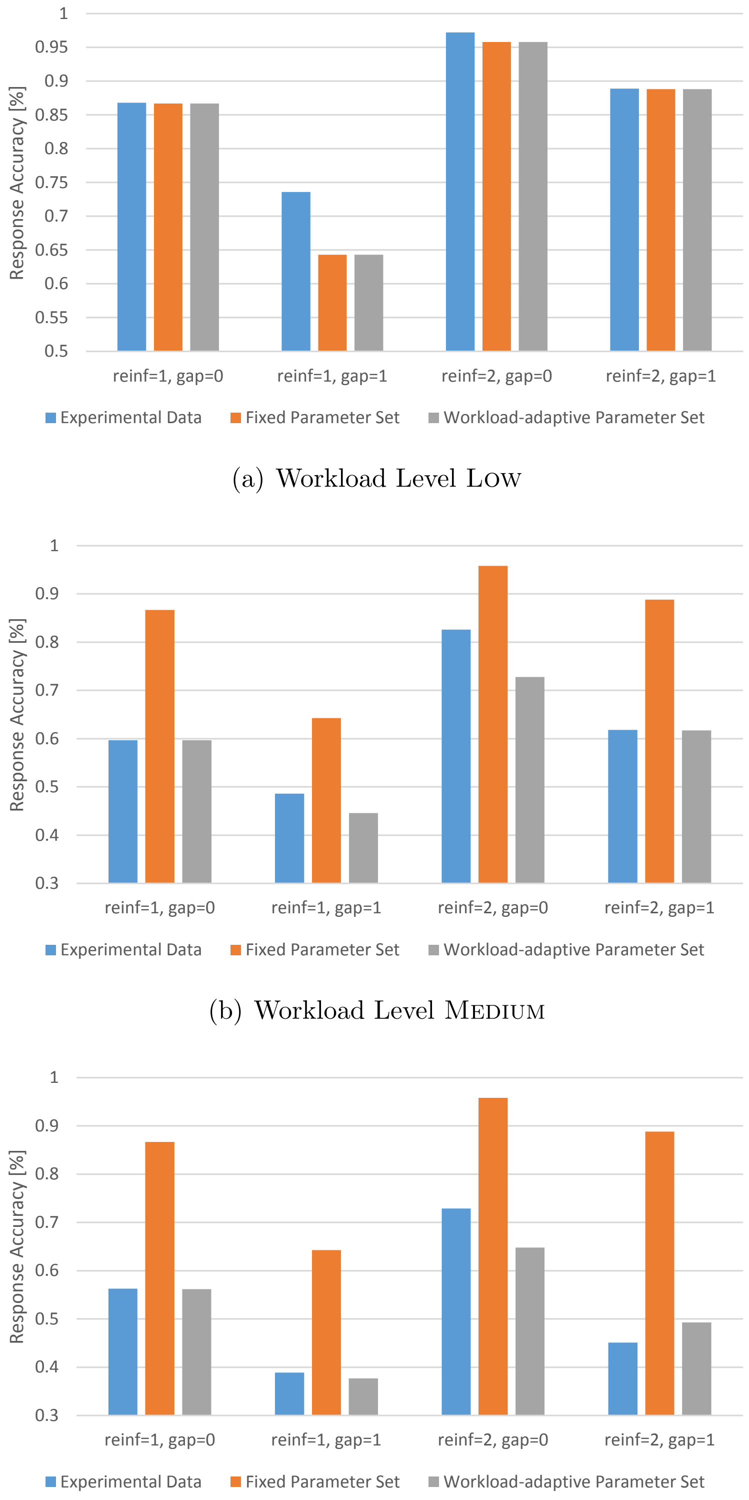


(a) Workload Level Low

(b) Workload Level Medium

(c) Workload Level High

Figure 3.9 – Average response time for different workload levels and word properties: 1 or 2 presentations (“reinf=1” / “reinf=2”), query within the same block as the presentation (“gap=0”) or with a gap of at least one block (“gap=1”). Besides the empirical data, we present the predictions of a model with fixed parameter set (optimized for Low) and of the adaptive model which uses workload-dependent parameter sets.



(a) Workload Level Low

(b) Workload Level Medium

(c) Workload Level High

Figure 3.10 – Average hit rate for different workload levels and stimulus properties: 1 or 2 presentations (“reinf=1” / “reinf=2”), query within the same block as the presentation (“gap=0”) or with a gap of at least one block (“gap=1”). Besides the empirical data, we present the predictions of a model with fixed parameter set (optimized for Low) and of the adaptive model which uses workload-dependent parameter sets.

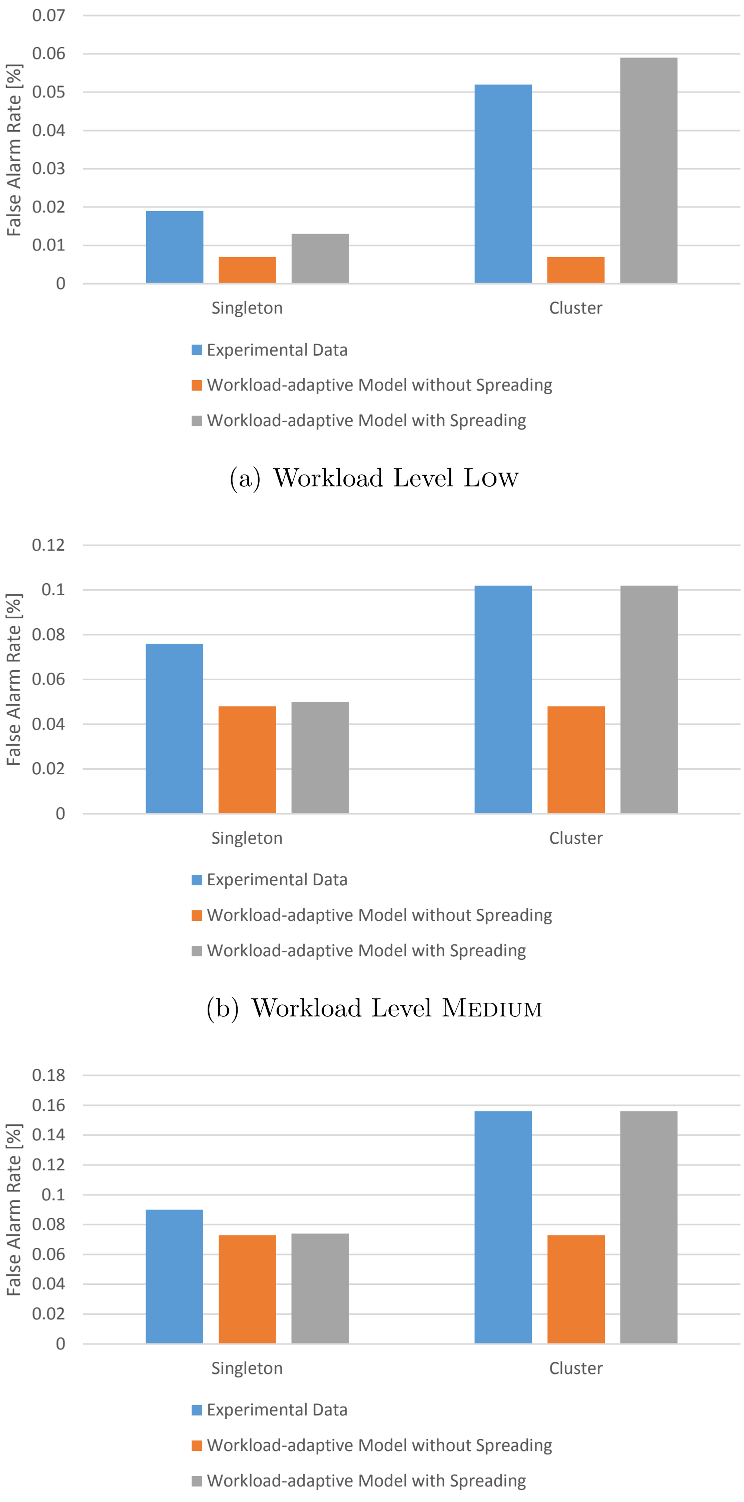
the models predict well the effects of both properties[10]: Response time of words retrieved with a gap is predicted to be higher and their hit rate is predicted to be lower than of words retrieved without a gap. Response time of words reinforced twice is predicted to be lower and their hit rate is predicted to be higher than of words reinforced only once. While the non-adaptive model predicts the human performance well for the workload level LOW, it is overly optimistic for the workload levels MEDIUM and HIGH. The model with workload-dependent parameter sets on the other hand reliably predicts the detrimental of rising workload on task performance. This is also reflected by the achieved fitness values, which correspond to the relative errors of the model, see again Table 3.4. If we use the parameter set for workload level LOW (i.e. the model that was optimized to data without secondary task), the fitness value when applied to the workload modes MEDIUM and HIGH was computed as -0.491 for workload level MEDIUM and as -0.868 for workload level HIGH, respectively. The relatively small average relative error of the workload-dependent parameter sets in contrast to the comparatively large relative error of the LOW parameter set when transferred to the workload modes MEDIUM and HIGH indicates that the design decision of modeling different workload modes as independent parameter sets was reasonable.

Evaluation of Cluster Property

In the previous part of the evaluation, two important aspects of the model were not regarded: False alarm rate and the difference between singleton and cluster words. Due to the spreading mechanism, we expect to see a difference between singleton words and cluster words, especially in false alarm rate. As a first naive attempt of predicting human performance for words with and without the cluster property, the parameter sets obtained in the first step of the evaluation (i.e. from Table 3.4) were used on data with both cluster and singleton words. Spreading was still deactivated. As expected, the prediction performance was relatively poor (with fitness values of -0.743 for LOW, -0.448 for MEDIUM and -0.360 for HIGH), as spreading is the designated mechanism to model the differences between those two categories of words.

To achieve a better fit, the model was then optimized again with activated spreading. Doing so introduced a new parameter, the spreading potential P_{spread}. The parameters d, τ, and s determined in previous evaluation step were kept fixed and only P_{spread} was optimized. Resulting parameter configu-

[10] Performance for adaptive and non-adaptive model are identical at workload level LOW, as the non-adaptive model uses parameters optimized for this workload level



(a) Workload Level Low

(b) Workload Level Medium

(c) Workload Level High

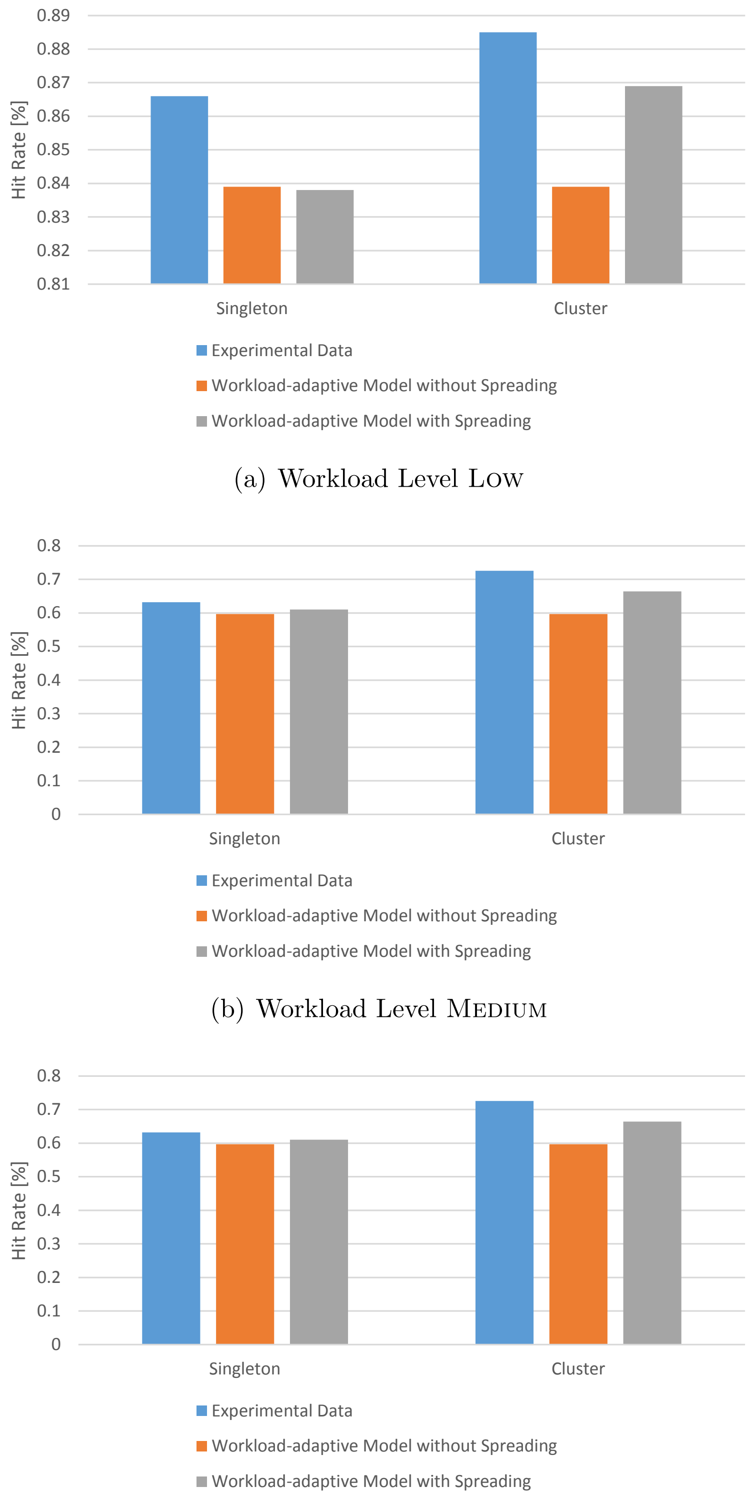
Figure 3.11 – Average false alarm rate for different workload levels and the “cluster” property: (“Singleton” / “Cluster”). We compare the experimental data to predictions of the adaptive model without spreading as well as to the adaptive model with spreading.

Parameter Configurations of Cluster-Optimization					
Workload Level	$d^\dagger$	$\tau^\dagger$	$s^\dagger$	$\boldsymbol{P_{spread}}$	**Fitness Value**
Low	0.500	-0.337	0.352	**2.012**	**-0.219**
Medium	0.505	0.031	0.749	**2.550**	**-0.171**
High	0.798	0.218	0.965	**2.983**	**-0.156**

Table 3.5 – Parameter configurations for the three workload modes as yielded by the optimization algorithm when optimizing for singleton and cluster words, and their corresponding fitness values. †Parameter values d, τ and s were taken from the previous evaluation step, and were therefore kept fixed.

rations are given in Table 3.5. Figure 3.11 compares the human performance (measured by false alarm rate) with the predicted performance of the adaptive model with and without activated spreading (we already know from the first step of the evaluation that the non-adaptive model will not predict the empirical data accurately). The human experimental data shows that there is a remarkable difference between singleton and cluster words in false alarm rate. Furthermore, we see that only the model with activated spreading was capable of predicting this effect of the cluster property. Resulting fitness values for the model with spreading were -0.219 for Low, -0.171 for Medium and -0.156 for High. This is a substantial improvement compared to the fitness values of the model without spreading, which cannot predict the difference between singleton and cluster words. When analyzing the parameter configurations given in Table 3.5, an interesting tendency regarding the spreading potential P_{spread} can be observed: It rises from 2.012 for Low, over 2.550 for Medium to 2.983 for High. This effect can be interpreted as spreading becoming more important under higher workload, a hypothesis which – given the good fit of the model – is also supported by the behavioral data from the participants.

We also analyzed the predictions made for the performance metric hit rate in dependency of the cluster property. The results of this analysis are presented in Figure 3.12. In the experimental data, we observed a slightly increased hit rate for cluster words compared to singleton words. This effect is predicted reasonably well by the model with activated spreading, as indicated by the fitness values. This indicates that the model is generally capable of predicting cluster effects on hit rate.



(a) Workload Level Low

(b) Workload Level Medium

(c) Workload Level High

Figure 3.12 – Average hit rate for different workload levels and the “cluster” property: (“Singleton” / “Cluster”). We compare the experimental data to predictions of the adaptive model without spreading as well as to the adaptive model with spreading.

3.4.4 Discussion

By using a genetic optimization algorithm in step-by-step approach, we optimized and evaluated the DMM with workload-dependent parameter sets. In summary, our central findings are:

- Modeling different workload modes by using different parameter sets yields good results. The workload-dependent model outperforms the one-fits-all model for workload levels other than Low.
- The effects of reinforcement and gap can be reproduced in all workload levels.
- The effects of the cluster condition can be reproduced by using the workload-adaptive model with activated spreading mechanism.

Given those results, we can conclude that it is possible to adapt a memory model to different workload settings. The experimentally found parameter modifications to adapt the model to these different workload settings are plausible for all parameters: The decay parameter d rises with increasing workload, which leads to a faster decline in base level activation and therefore to faster forgetting. Also the retrieval threshold τ rises with increasing workload. This means that memory elements need a higher activation value to be retrieved from memory. The retrieval sensitivity s, which controls the slope of the retrieval probability curve, also grows with increasing workload. This can be interpreted as decision making becoming more binary under high workload. Also the spreading potential P_{spread} rises with increasing workload, which suggests that spreading and therefore learning words as clusters become more important under high workload.

For the first time, this work has shown that the DMM (which is based on the established ACT-R memory model) can be extended by workload-dependent parameter sets to reflect the changes in human memory performance caused by changing workload levels. This approach was validated for different categories of memory items (gap, reinforcement and cluster property) and performance metrics (hit rate, response time, false alarm rate). This implies that the extended model can be used to accurately predict memory performance not only during uninterrupted, distraction-free task execution, but also in high workload situations. The necessary prediction of workload level can be provided by an empirical cognitive model as described in Section 2.3.

3.5 Dummy Model based Workload Modeling

In the previous section, we have shown that we can manipulate the parameters in a computational cognitive model explicitly to express the effect of different workload levels during the execution of a cognitive task. The downside of this approach is that manipulation of parameters only models those effects which were explicitly taken care of (e.g. the model in Section 3.4 only models the impact of workload on memory, not on manual execution) and that the manipulation of core model parameters bears the risk of compromising model validity (e.g. most validation studies of the ACT-R memory model used the proposed default value for the decay parameter $d = 0.5$, while the adaptive model presented in Section 3.4 used different values for d). In this section, we propose an alternative approach which does not explicitly reflect the impact of workload on specific modeling components. Instead, the implicit approach represents the effect of high workload implicitly by adding a dummy task which consumes cognitive resources[11].

An additional limitation of the previous approach which we also address in this section is the fact that in Section 3.4, we assumed perfect information on workload level to be present, which an empirical workload model cannot provide from noisy sensor data. In this section, we therefore analyze the impact of classification error of an empirical cognitive model on the performance of the computational cognitive model. Finally, we acknowledge the fact that individual differences between different humans play a large role when modeling the impact of internal states on cognitive processes. Therefore, we investigate how individualized models (which were tuned to a low-workload condition) are capable of adjusting to different workload levels.

In the following, we will describe the integration of an EEG-based empirical workload model into a cognitive ACT-R model to predict the impact of different workload levels on human behavior and task performance. The workload level during execution of a main task is manipulated using two different secondary tasks. The changing workload levels are recognized from the recorded signals by an empirical cognitive model and used to activate or deactivate dummy models to run in parallel to the main task model. This influences the predicted performance to better match human behavior.

[11] This section is based in parts on the diploma thesis of Robert Pröpper which was supervised by the author of this thesis.

Our proposed architecture consists of four components: an empirical workload model, a main task model, a dummy model and a switching strategy. The interaction between those components is displayed in Figure 3.13: During the execution of an experiment, the empirical workload model continuously evaluates features extracted from the recorded EEG signal. It yields a workload estimate on a scale from 0 to 1 which is propagated to the ACT-R model. The ACT-R model then decides based on the workload estimate to either activate or deactivate a dummy module. This dummy module represents a separate task thread in ACT-R (using the Threaded Cognition mechanism, see Section 3.2.6) which is an abstract model of cognitive activity caused by a secondary task. Executing the dummy model in parallel to the main task model will cause cognitive resources to be occupied for the actual task model, potentially resulting in reduced task accuracy or increased response time.

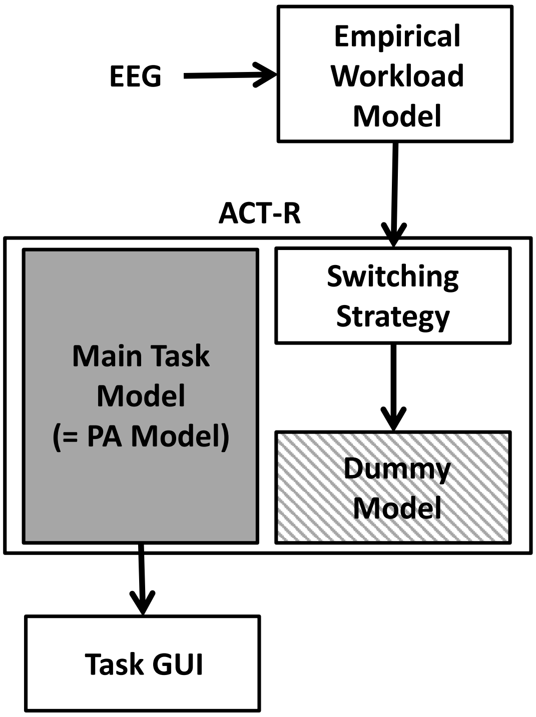


Figure 3.13 – System setup for workload adaptation of ACT-R based on a dummy model.

In the following sections, we will explain the details of the dummy-model approach and validate the approach using data from two user studies. To our best knowledge, this is the first implementation and systematic evaluation of an ACT-R model which is able to predict behavior and performance under variable workload levels.

3.5.1 Computational and Empirical Models

This subsection describes the components required for the dummy model approach to represent the effect of workload on behavior and performance: Main Task Model, Empirical Workload Model, Dummy Model & Switching Strategy.

Main Task Model

The employed main task model is a regular ACT-R model of the main task. In this section, our example main task is the Paired Associate (PA) task. The PA is a task of learning and recalling the pairing of two items – a stimulus and a response. First, a list of such pairings is learned by a participant. Then, a sequence of stimuli is presented and the participant is asked to give the associated response. Associative learning is an important aspect of intelligence and required by many cognitive tasks in an HCI context (e.g. learning the correct input command in response to certain system outputs). Additionally, our implementation of the PA represents the operation of a GUI (visual stimuli on a screen and manual responses on a keyboard). For those two reasons, the PA task is a representative abstraction of many relevant tasks in real life.

For the ACT-R model, we used a slightly modified version of the PA model provided as part of the ACT-R distribution. As the only modification to the original model, we included one additional request to the declarative module to represent the learned non-trivial mapping of numbers to keys on a numerical keypad (i.e. the participants had to remember which keys were associated with which responses in the task). Input and output to the model were directly provided from and to the GUI which participants used during the experiments.

The goal of this section is the prediction of cognitive performance and behavior under different workload levels for individual sessions. This use case implies that we need to individualize modeling parameters for each participant, as we are predicting performance in one concrete situation, not on average. Therefore, we determined optimal model parameters for each participant. To avoid overfitting of the models to limited training data, we restricted our optimization to sequential adjustment of only two parameters of the declarative model. We do this by adjusting first the τ parameter (the retrieval threshold) to optimally match the retrieval accuracy of a participant and then adjust the parameter F (which determines retrieval latency) to optimally match the response time. τ was optimized in the range of $[1.0, 2.0]$ and F was optimized in the range of $[0.2, 0.8]$. We chose those two parameters because τ and F are sufficient to manipulate the prediction of response accuracy and response time, respectively. The necessary statistics for optimization of those parameters were estimated on data without secondary task for each individual participant. We performed optimization as an exhaustive search on data which was not used for the subsequent evaluation in Section 3.5.4. The `LOW` bars of Figures 3.16, 3.17, 3.18, and 3.19 show a breakdown of PA response accuracy and response time by individual participant for two different data sets. In these figures, we see that the inter-participant performance variance was large. As the model parameter determine the predicted performance, only individual parameter sets will be able to adequately model these different performance levels.

To have the ACT-R model interact with the task GUI, which was implemented in Python, we implemented the `Hello Python` module. Similarly to the existing `Hello Java` module [Büt10], this component allows interfacing Python applications with ACT-R models. `Hello Python` consists of two components, an ACT-R extension and a Python module. Both components communicate via TCP/IP to control Python applications from ACT-R.

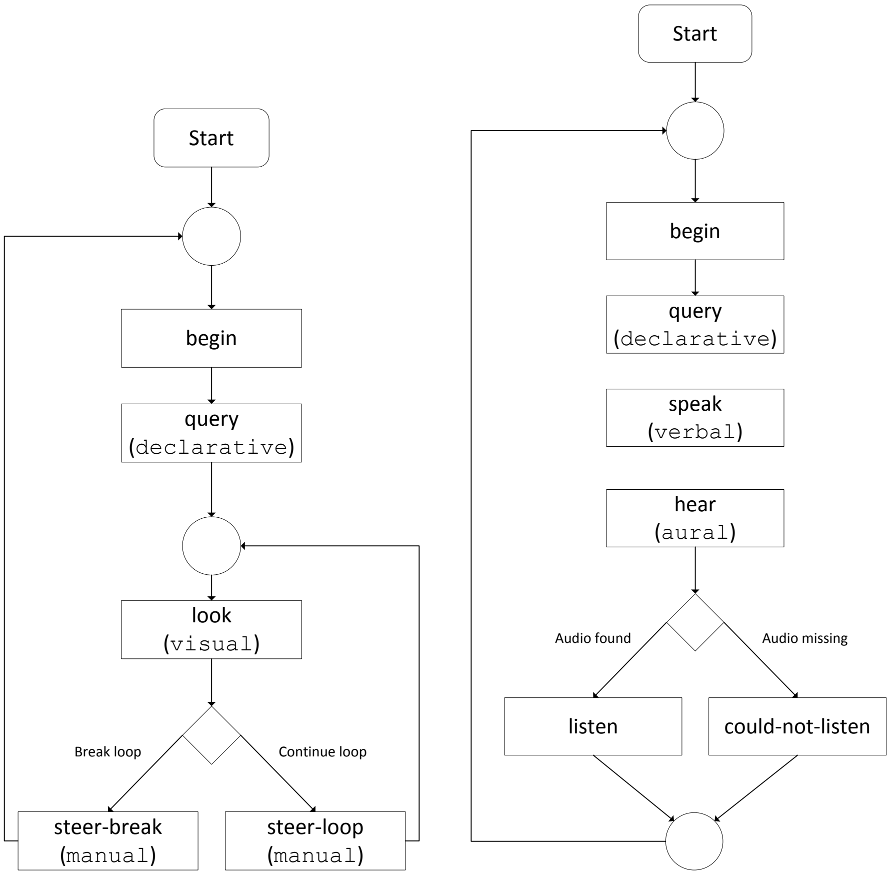
3.5.2 Empirical Workload Model

To provide an empirical workload model, we employed the model of Section 2.3 in a person-dependent variant, using frequency features calculated on windows of 2 s length with an overlap of 1.5 s. For data acquisition, we applied an active EEG-cap (BrainProducts actiCap) to measure the participants' brain activity using 16 electrodes placed at positions FP1, FP2, F3, Fz, F4, F7, F8, T3, T4, C3, Cz, C4, P3, P4, Pz, and Oz according to the international 10-20 system [Jas58] with reference to the left mastoid. The impedance of each electrode was kept below 20 kΩ during all sessions. Amplification and A/D-conversion was performed on a 16 channel Vario-Port biosignals recording system by Becker Meditec using a sampling rate of 256 Hz.

Dummy Model & Switching Strategy

The dummy model is an ACT-R model which runs in parallel to the main task model (using the Threaded Cognition mechanism). It abstractly models the cognitive processes involved in the secondary task. In contrast to the main task model, the dummy model is not a detailed model of a valid human solution strategy of the secondary task. Instead, it contains a sequence of requests to ACT-R modules associated to the task, for example the declarative module or the visual module. This sequence is repeated while the model is running. It is possible to activate and deactivate the model at runtime. In deactivated mode, the model does not perform any module requests. In activated mode, the repeated module requests cause exclusive or limited cognitive resources to be temporarily blocked for the main task model, potentially resulting in longer response times or even task failures. The activation of the dummy model is performed on the basis of the empirical model, i.e. when high workload is detected. This reflects the degradation of human performance caused by multi-tasking which we already observed in Sections 2.3 and 3.4.

As different secondary tasks may have different characteristics of resource usage, we implemented different dummy models corresponding to different types of secondary tasks. For this work, we chose two different paradigms of secondary tasks to explore the possibilities of the presented approach: The Sternberg memory task [S^+66] and the Lane Change Task (LCT) [Mat03]. The Sternberg task generates heavy memory load and is therefore expected to interfere with the memory demands of the Paired Associate main task. We employed a purely acoustical version of the task where the stimuli were read to the participant and responses were given verbally. Therefore, we did not expect interference with the visual input and motor output of the PA task. The corresponding dummy module `Sternberg-Dummy` performs periodic requests to the aural, the verbal and the declarative module. The LCT on the other hand is a driving task executed in a driving simulator (see Section 2.3.1). It requires the participant to change lanes on a three-lane highway as indicated by road signs. The memory load of this task is low, however input and output modalities interfere with the PA task. The corresponding dummy module `LCT-Dummy` performs periodic requests to the visual and the motor module. Figure 3.14 shows flow charts which describe the sequence of operations performed by the two dummy models. Each block corresponds to one production rule occupying a corresponding ACT-R module (e.g. “steer” occupies the manual module). Note that corresponding real ACT-R models would be much more complex, especially for the LCT: Considering how complex even a model of lane and distance keeping is [Sal06], we save much modeling effort by introducing the dummy model. We achieve this reduction in effort because the dummy model concentrates on its main goal which is to give an abstract representation of the cognitive resources required for the execution of the secondary task.



(a) Lane Change Task

(b) Sternberg task

Figure 3.14 – Flowcharts of two ACT-R dummy models `LCT-Dummy` (left) and `Sternberg-Dummy` (right). Each box corresponds to one ACT-R rule. In parentheses we give the ACT-R modules which are involved in the execution of the corresponding rule.

Note that cognitive resources are not permanently occupied by the secondary tasks, as those also contain pause segments. For this reason, the dummy model is not processed all the time during high workload phases. Instead, the model is randomly activated during those phases with a certain probability. This probability is 50% for the `Sternberg-Dummy` and 25% for the `LCT-Dummy`. Those numbers correspond to the ratio of task processing to pause segments for those two secondary tasks.

3.5.3 Experimental Setup

For evaluation of the dummy-model approach, we recorded two data sets with the same main task (PA) and different secondary tasks (Sternberg, LCT). In both data sets, participants performed the PA task multiple times with and without secondary task. Each condition (e.g. with and without secondary task) was repeated twice for the LCT data set and four times for the Sternberg data set. Additionally, there was a training session for each condition which was not recorded.

The LCT data set was recorded in the driving simulator. For the Paired Associate task, we showed sequences of words for learning and query on a 7“ computer screen in the driver's cockpit. All presented words were in German language with four letters and associated with numbers from 1 to 9. A learning phase consisted of 16 words. During the query phase, each word was queried three times in randomized order during the query phase. The response window was 4 s, correct answers were always shown afterwards for 2 s. Subjects gave their response using a numeric keypad strapped to their right leg.

The LCT task was performed using a force-feedback steering wheel and a gas and brake pedal for controlling a virtual car on a large projection screen. Participants were instructed to drive at a constant speed of 180 km/s and had the task to follow lane changing instructions given visually on road signs which appeared at fixed intervals. One run of the LCT task this setup lasted for five minutes.

For the Sternberg data set, recording was performed on a standard desktop computer and screen, on which the PA task was performed. For the Sternberg task, sequences of five short phonetic strings (e.g. ”omo”) without semantic meaning were read to the participant during a learning phase. In a subsequent query phase, target strings from the learning phase had to be discriminated from distractors. Responses were given verbally by the participant in a subsequent response window of 3 s. In total, one query phase contained 20 phonetic strings. One run of the Sternberg task consisted of four pairs of training and query phases and lasted five minutes.

Using this setup, we recorded a total of nine sessions in the Sternberg data set and nine sessions in the LCT data set (one session per subject). All 18 participants were university students with a mean age of 23 ($\sigma = 2.6$). Overall, 6 of the participants were female, 12 were male. All participants gave their written consent to their participation in the study.

3.5.4 Evaluation of the Dummy Model Approach

The evaluation of the dummy model approach consists of six parts: First, we investigate whether the PA runs with and without secondary task result in different performance for the PA task and whether they were subjectively perceived as different in workload. Second, we compare the two secondary tasks regarding their impact on the participants' performance in the PA task. Third, we evaluate the individualized non-adapted model (i.e. without dummy model) for its ability to predict human performance for the PA runs with and without secondary task. Fourth, we repeat this analysis with an oracle-adapted model (i.e. with dummy model activated by perfect workload recognition). Fifth, we evaluate the results of the EEG-based empirical workload model. Sixth, we evaluate the EEG-adapted model (i.e. with dummy model activated by the empirical workload model).

Evaluation of Workload Assessment

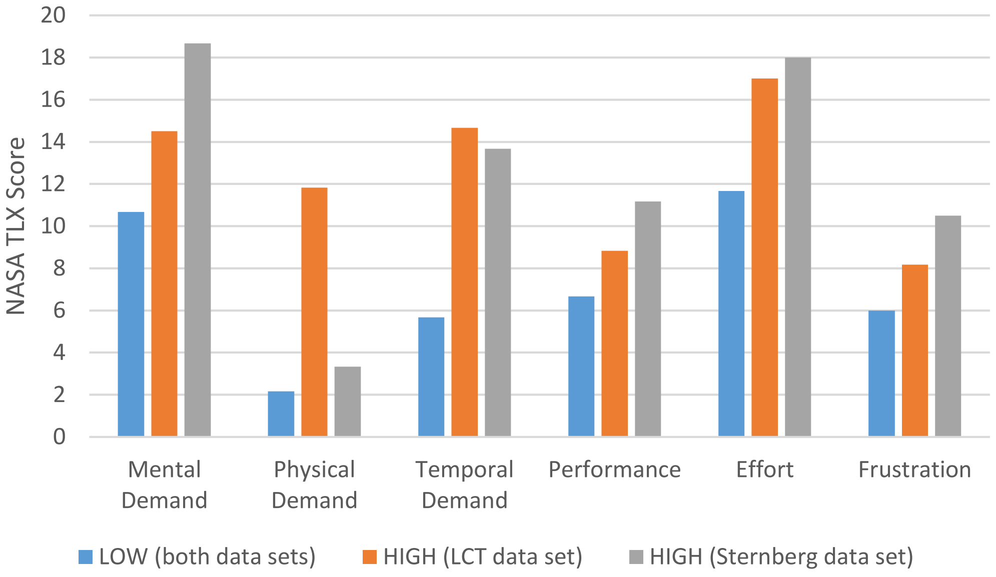


Figure 3.15 – Raw TLX ratings for different data sets and conditions. "Solo" denotes recordings without secondary tasks.

We used the NASA-TLX questionnaire [HS88] to assess the experience of our participants with the different tasks in our experiment. After every PA run, the participants were handed a German version of the NASA-TLX questionnaire. We only used the raw questionnaire scores instead of the weighted ones, as suggested by [BBH89]. Figure 3.15 shows the mean TLX ratings for our experiments. The results fit to our expectations of the task difficulty and the performance of our test participants: In all categories, the Paired Associate task without secondary task was scored lowest (a lower score generally corresponds to an easier task). In most questions, the combination of PA with LCT (PA+LCT) was scored lower than the combination of PA with Sternberg (PA+SB). Two categories do not follow this general trend: Physical demand was ranked much higher for the PA+LCT than both other task conditions. This is easily understandable given that driving a car poses a much larger physical challenge than pressing keys on a keypad or speaking. Also, our participants felt that the temporal demand of the Lane Change Task was slightly higher than the temporal demand of the Sternberg task. This coincides with the fact that LCT only influences response time and not response accuracy of the PA task, while the Sternberg task influences both (see next subsection). Overall, the evaluation of the NASA-TLX justifies the notion of low and high workload conditions. Therefore, we will sometimes refer to the condition PA without secondary task as `LOW` and the combination of PA with either `LCT` or `Sternberg` secondary task as `HIGH` condition.

Comparison of Secondary Tasks Impact on Performance

Table 3.6 summarizes the performance of the participants on both data sets for `LOW` and `HIGH` workload conditions in the PA task. We see that response time and response accuracy rise significantly ($p < 0.01$ for all data sets and performance metrics) for both data sets when an additional secondary task is processed. A non-adaptive model which is designed to predict such performance metrics assuming full concentration on the main task will therefore greatly overestimate performance of human participants when a secondary task is actually present. By inspecting the standard deviations, we also see that the individual differences are large, stressing the need for model individualization to perform real-time prediction.

Moreover, there is a difference between the two data sets in the quality of performance impact: While the performance differences between `LOW` and `HIGH` workload conditions are statistically significant on average, the measured effect size is not the same for both data sets. We have a strong impact on the accuracy metric only for the `Sternberg` data set. For the `LCT` data set, only reaction time shows a strong degradation for the `HIGH` condition compared the corresponding `LOW` condition. This observation is consistent with the types of distraction which are caused by the two secondary tasks: The Sternberg task incurs strong working memory load and therefore makes the declarative module a bottleneck. This behavior harms both PA accuracy and PA response time. On the other hand, the LCT task only marginally influences PA accuracy, as it does not occupy the declarative module. Instead, occupying visual and manual module leads to delays in stimulus processing and response execution, therefore to an increase in response time.

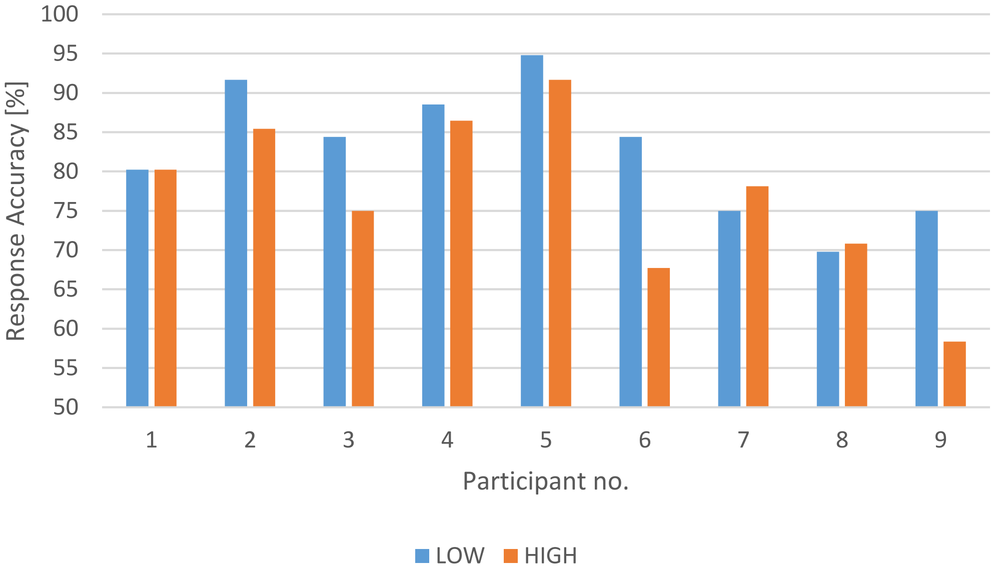


Figure 3.16 – Breakdown of individual average response accuracy for the `LCT` data set.

There is another difference between both data sets, regarding the consistency of performance degradation between `LOW` and `HIGH` condition. This can be observed when looking at the breakdown of performance metrics by individual participants in Figures 3.16, 3.17, 3.18 and 3.19: For the `Sternberg` data set, both accuracy and response time suffered from the addition of a secondary task for every single participant. In contrast, this was only the case for six out of nine participants for the `LCT` data set, because for three participants response accuracy did not decrease (or even increased) in the `HIGH` workload condition. Additionally, the standard deviation of the differences between `LOW` and `HIGH` is much higher for the `LCT` data set compared to the `Sternberg` data set (6.7 vs. 5.6 for accuracy and 0.44 vs. 0.28 for response time). We explain this by the fact that the time window for responding to stimuli in the LCT task was much larger than for the Sternberg task (2 s for Sternberg vs. ca. 8 s for the LCT), opening up many opportunities for

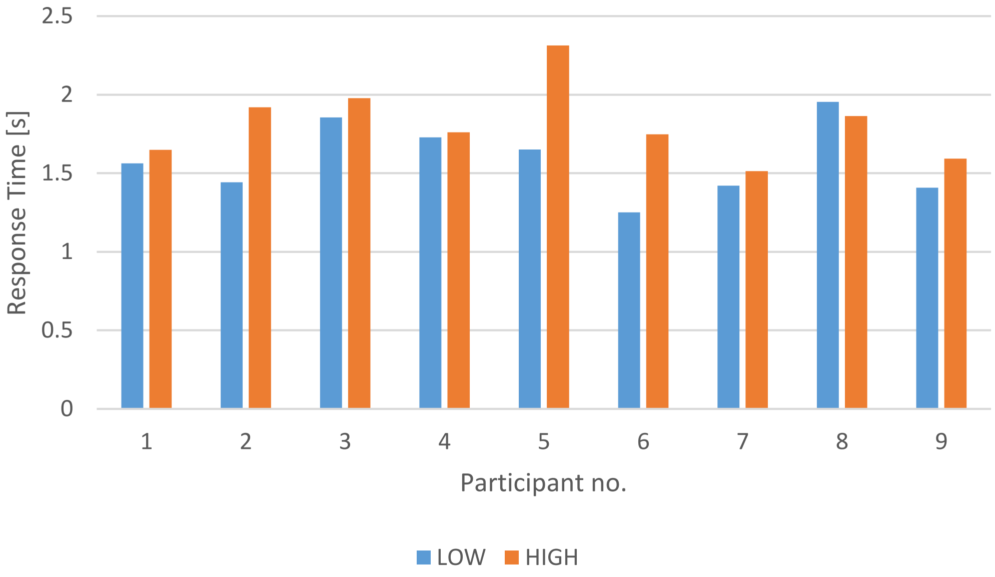


Figure 3.17 – Breakdown of individual average response time for the `LCT` data set.

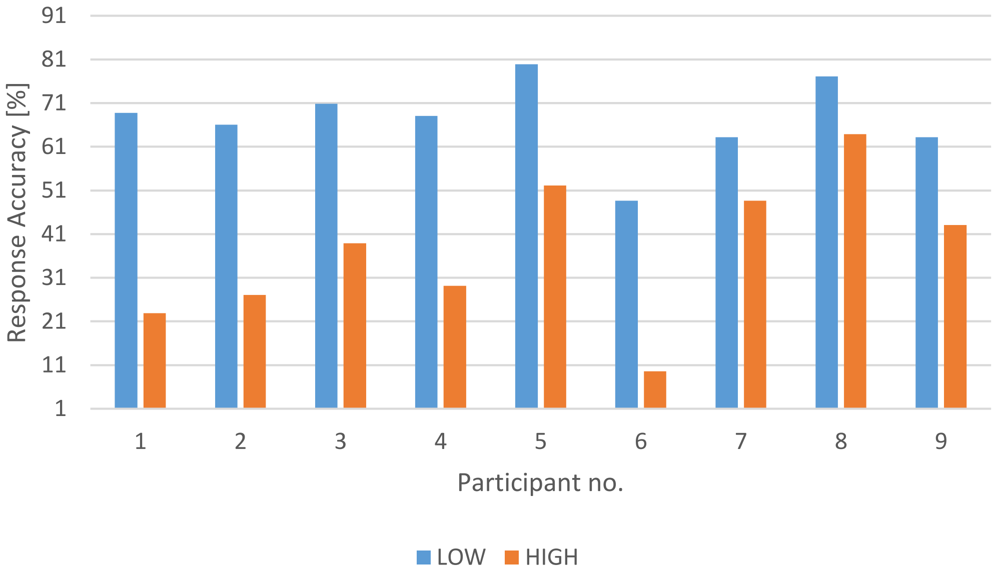


Figure 3.18 – Breakdown of individual average response accuracy for the `Sternberg` data set.

individual response strategies. This situation makes modeling of workload effects of course much more challenging for the `LCT` data set.

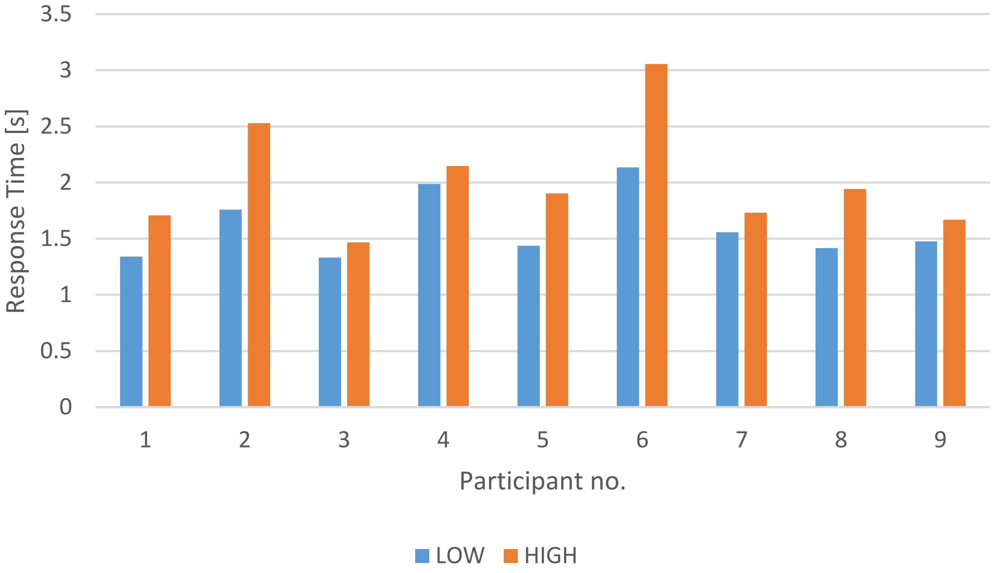


Figure 3.19 – Breakdown of individual average response time for the `Sternberg` data set.

Condition	Accuracy [%]	Response Time [s]
`LOW` (`Sternberg` data set)	32.3 (3.34)	1.60 (0.29)
`HIGH` (`Sternberg` data set)	15.75 (6.29)	2.02 (0.49)
difference `HIGH` vs. `LOW`	-16.55*	0.42*
`LOW` (`LCT` data set)	40.89 (5.78)	1.51 (0.32)
`HIGH` (`LCT` data set)	37.95 (7.46)	1.75 (0.38)
difference `HIGH` vs. `LOW`	-2.94*	0.24*

Table 3.6 – Average PA performance of all test participants for different workload levels in two data sets. Standard deviation given in parentheses. An asterisk denotes a significant difference between `HIGH` and `LOW` at $\alpha = 0.01$. Note that the `LOW` condition in both cases contains the identical PA task. The data sets are presented separately because of performance differences.

Evaluation of Individualized Non-Adapted Model

In this section, we evaluate the performance of the individualized, but non-adapted model (i.e. without dummy model) to the human performance in the two data sets `Sternberg` and `LCT`. For evaluation of model prediction, we use the prediction error (PE) metric, which is the average absolute difference between empirically measured and predicted value. Often, we look at the relative PE, i.e. the ratio between PE and the human performance.

Table 3.7 shows the PE of the individualized non-adapted PA model when predicting the performance parameters of the individual participants for LOW and HIGH workload level. For this evaluation, we look at the metrics response time and response accuracy of the Paired Associate task. We report the absolute PE as well as PE relative to the respective metric measured for the human participants. We compare model predictions to the human performance in the final run in a session for each workload level, as this run contains the weakest learning effects). Unsurprisingly, the individualized PA model can very reliably predict performance for the LOW condition (this is not trivially true as parameters were estimated on a different run of the PA task): PE averaged across all participants is below 2% for both data sets. However, PE increases substantially when transferring this model to one of the high workload conditions. For the Sternberg data set, predicted response accuracy is nearly twice as large as measured empirically (92% PE for response accuracy in the HIGH condition of the Sternberg data set). For both data sets, the average response time of actual test participants was more than 20% higher than predicted by the non-adapted model (21.4% for the Sternberg data set, 25.8% for the LCT data set). This large prediction error for the HIGH workload level mandates the use of a workload-adaptive model.

Condition	PE Accuracy	PE Response Time
LOW (Sternberg)	0.39 (=1.12%)	0.02 (=1.25%)
HIGH (Sternberg)	14.49 (=92.0%)	0.43 (=21.4%)
LOW (LCT)	0.21 (=0.51%)	0.02 (=1.32%)
HIGH (LCT)	6.3 (=15.4%)	0.39 (=25.8%)

Table 3.7 – Average absolute and relative prediction error (PE) for applying the individualized, **non-adapted** PA model to data from different conditions.

Evaluation of the Individualized Oracle-Adapted Model

To evaluate the benefit of workload-adaptive models, we start by analyzing the dummy-model approach using a workload oracle. The workload oracle directly derived the correct workload level (high or low) from the task condition and propagated this value to the ACT-R model to switch the dummy model on or off. Table 3.8 presents average absolute and relative prediction error when applying the workload-adaptive model using workload oracle. As we assume perfect workload recognition, performance prediction in both LOW conditions was identical to the performance of the non-adapted model in Table 3.7 (which always operated without dummy model). When looking at the results for HIGH conditions, we see that the prediction error compared

Condition	PE Accuracy	Δ Acc	PE Resp. Time	Δ RT
LOW (Sternberg data set)	0.39 (=1.12%)	-	0.02 (=1.25%)	-
HIGH (Sternberg data set)	5.52 (=35.0%)	-57%	0.27 (=13.4%)	-8.0%
LOW (LCT data set)	0.21 (=0.51%)	-	0.02 (=1.32%)	-
HIGH (LCT data set)	6.3 (=15.4%)	-0.0%	0.30 (=16.1%)	-9.7%

Table 3.8 – Average absolute and relative prediction error for applying the individualized, **oracle-adapted** PA model to data from different conditions. Also given is the reduction of PE from the prediction of the non-adapted model: "Δ Acc" is the absolute reduction in relative PE compared to the prediction of the non-adapted model. "Δ RT" is the same for response time.

to the non-adapted model was reduced for all situations and metrics. For easier comparison, we report the difference between non-adapted and oracle-adapted prediction relative to the PE of the non-adapted model. For the Sternberg data set, the result is most convincing, with a reduction of 57.0% absolute (61.96% relative) in prediction error for response accuracy. PE for response time was reduced by 8.0% (37.38% relative). Both reductions in prediction error for the Sternberg data set were statistically significant ($p \leq 0.001$ for both response accuracy and reaction time).

For the LCT data set, we also observe a reduction in prediction error, at least for response time (absolute reduction of PE by 9.7%, which is 37.6% relative). This reduction barely not significant at $p = 0.07$. Regarding response accuracy, we did not expect to observe an effect of using the oracle-adapted model, as the LCT dummy model mostly affected stimulus perception and response generation, but not memory retrieval. The fact that the reduction in response time was not significant is caused by the high inter-person variability for the LCT data set; for one participant in this data set, response time even decreases in the HIGH condition. The current parameter-free model (i.e. a model which has no individual parameters which allow an individual adjustment the dummy model) is not able to predict the occurrence of such "paradox" performance beforehand. The big advantage of a parameter-free model is that for all other participants, it is able to provide a significant improvement in prediction quality without the necessity of additional data for fitting model parameters. This means that the workload-adaptive model is not more complex (i.e. does not have more parameters) than the non-adaptive one. If we remove the participant with "paradox" performance from the analysis, the reduction in prediction accuracy (which then increases to 12.0%, which is 46.51% relative) becomes significant ($p = 0.01$).

The quantitative difference between both data sets also shows that the implementation of a generic, task independent dummy model is not realistic: For example, a model which would accurately predict the effect of the LCT secondary task would inevitably underestimate the impact of the Sternberg secondary task on response accuracy. A consequence of this observation is that at least a rough estimate of the cognitive resource requirements of the secondary task is necessary to select an appropriate dummy model, as well as an estimate on the task intensity (i.e. the probability with which the dummy model is active during `HIGH` workload).

Evaluation of Empirical Workload Model

In the last part of the evaluation, we want to replace the workload oracle from the previous section with the workload predictions of an empirical workload model. Before we can do that, we first need to evaluate the ability of the empirical workload model to discriminate `LOW` and `HIGH` workload level. For this purpose, we evaluate the classification accuracy after temporal smoothing. Averaged over all participants, we achieved a person-dependent classification accuracy of 84.1% ($\sigma = 14.1$) for discriminating between `LOW` and `HIGH` for the `LCT` data set and a mean person-dependent accuracy of 64.9% ($\sigma = 15.3$) for for discriminating between `LOW` and `HIGH` for the `Sternberg` data set. The substantially lower accuracy for the latter evaluation can be explained by the non-continuous workload induced by the Sternberg task (learning and query phase) in comparison to the LCT: On average across all participants, the temporally smoothed workload level for the query phases of the Sternberg task is 0.15 higher than the recognized workload level for learning phases.

Evaluation of the Individualized EEG-Adapted Model

Up to this point, we have assumed a perfect workload oracle to differentiate between low and high workload condition for triggering the dummy model. This is of course a best-case assumption as workload prediction based on empirical models will always be prone to classification errors. Such errors may reduce the benefit of modeling high workload situations. Additionally, they may lead to the activation of the dummy model in low workload conditions. In the following, we quantify the effect of replacing the workload oracle with a realistic empirical model. For this purpose, we modify the fraction of time during which the dummy model is activated for each condition, based on the workload estimates of the individual participants. When using

Condition	PE Accuracy	Δ Acc	PE Resp. Time	Δ RT
LOW (Sternberg data set)	3.44 (=9.92%)	+8.8%	0.11 (=6.9%)	+5.65%
HIGH (Sternberg data set)	7.34 (=46.6%)	-45.4%	0.26 (=12.9%)	-8.5%
LOW (LCT data set)	0.34 (=0.83%)	+0.32%	0.08 (=5.31%)	+3.99%
HIGH (LCT data set)	6.1 (=14.8%)	-0.6%	0.32 (=17.2%)	-8.6%

Table 3.9 – Average prediction error for applying the individualized, **EEG-adapted** PA model to data from different conditions. Also given is the reduction in PE from the prediction of the non-adapted model: "Δ Acc" is the absolute reduction in relative PE compared to the prediction of the non-adapted model. "Δ RT" is the same for response time.

a workload oracle, the dummy model is activated exactly at the desired ratio (50% for Sternberg-Dummy and 25% for LCT-Dummy, see Subsection 3.5.2) during HIGH conditions and turned of completely during LOW conditions. For an EEG-adapted model, those ratios changes depending in the classification accuracy a of the empirical workload model: The dummy model is then activated $a \cdot 50\%$ of the time during HIGH conditions and $1 - a$ of the time during LOW conditions. Table 3.9 summarizes the prediction error of the adapted models using those individual workload performance values. We see that while prediction for the HIGH conditions is less accurate than when using a workload oracle, we still outperform the non-adapted model. At worst, we see an increase in prediction error of 8.8% for response accuracy in the Sternberg condition. This value is close to the measured error probability of the empirical workload model. For all other performance metrics, the increase of prediction error is much smaller. A caveat is of course that when using an EEG-based workload prediction, this will also be active during times of LOW workload. Workload prediction errors during LOW phases would degrade model prediction compared to the baseline (non-adaptive) model. To quantify whether workload prediction is still beneficial for the model when regarding both LOW and HIGH sections, we need to compare the magnitude of improvement in prediction error for the HIGH results with the magnitude of deterioration for the LOW condition. For example for the Sternberg data set, we have an improvement of 45.4% for the HIGH condition (EEG-adapted model vs. non-adapted model) compared to a degradation of 8.8% for the LOW condition. Assuming equal distribution of both conditions, this results means that on average, we still achieve a substantial net benefit using the EEG-adapted models.

Those results indicate that it is possible to combine empirical modeling of human workload with computational modeling of task execution in a cognitive architecture to provide adaptivity to varying conditions. All presented components (ACT-R model, EEG data acquisition and workload modeling) are available as real-time capable components. This allows the system to predict performance of a participant during the actual operation of the task, even in the case of dynamically changing workload conditions.

3.5.5 Discussion

In the evaluation, we showed the importance of model parameter individualization and workload adaptation for the purpose of model tracing in situations of variable workload. For both the `Sternberg` and the `LCT` data set, we could show a significant reduction in prediction error for the workload-adaptive model compared to the non-adaptive. This is an important finding which is new to the research community.

The `LCT` data set proved to be more challenging, as individual differences were not only present for baseline performance but also for the impact of adding a secondary task. This can be seen in Figures 3.16 and 3.17, where we see some participants which were not influenced negatively at all by the `HIGH` workload level (e.g. participants 1, 7, and 8), while performance suffered strongly for other participants (e.g. participants 6 and 9). When applying the same adaptation procedure (i.e. behavior of the dummy model, frequency of the dummy model) to all participants, this lead to overestimation of the impact of workload in some cases. This opens up more opportunities for model individualization, but would also require more data for calibrating those models. Still, we could show that on average, the dummy model approach significantly outperforms a non-adaptive model in predicting task accuracy and reaction time. This is also true if workload prediction is not perfectly accurate.

Additionally, we saw that the impact of the two secondary tasks on the performance in the main task was different. This observation implies that switching dummy models between tasks would lead to a substantial degradation of prediction accuracy. A consequence is that while we do not need to precisely model the secondary task with a full-blown ACT-R model, we still need context information on the resources it consumes. This shows the necessity of maintaining multiple dummy models for different (types of) secondary tasks.

For the development of interaction systems, we can now use an ACT-R model of the interaction task (which may already exist, for example for the purpose of user simulation during interface prototyping) and a selection of dummy models to predict performance. As a full ACT-R model of the main task needs to be available, this approach is especially appropriate for strictly goal-oriented tasks for which a small set of clearly defined strategies exists. It should be noted that an ACT-R model cannot reliably predict the performance of a single trial of the primary task. This is not a limitation of the dummy model approach but due to the variability of human performance itself. The model can however be applied well to situations in which workload changes are rare compared to the frequency of new trials.

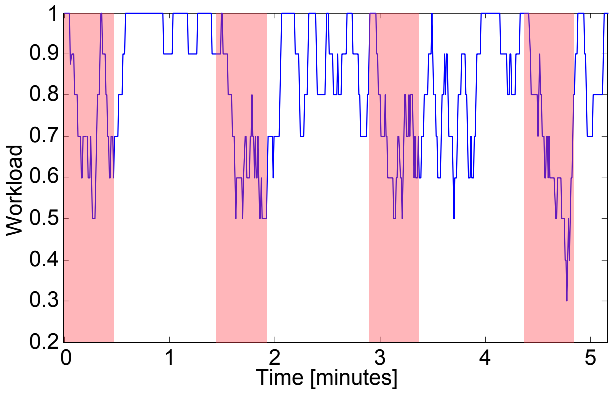


Figure 3.20 – Differences in recognized workload levels for learning phases (red) and query phases of the Sternberg task.

For future work, we propose to develop more sophisticated strategies to activate the dummy model during HIGH workload. In the current implementation, the dummy model is activated randomly during phases of high workload. However, for most cognitive tasks workload is not distributed uniformly across time. For example, in the Sternberg task there is a distinct resource demand for the learning and the query phases, which implies different workload characteristics. This can also be observed in the recognized workload patterns. Figure 3.20 shows the (temporally smoothed) EEG-based workload prediction during one run of the HIGH condition of the Sternberg data set. We have marked the four learning phases of the Sternberg task in red. The other segments of the block belong to the recognition phases of the task. We note differences of recognized workload levels within the output for the dual task condition. It is clearly visible that the recognized workload in these phases is lower than during the query phases.

3.5.6 Comparison of Explicit and Implicit Modeling of Workload

In the two previous sections, we proposed and evaluated two approaches to model the effect of workload on behavior and performance in cognitive tasks. The explicit overlay approach used different workload-dependent parameter sets (optimized to data from human participants operating at different workload levels) for the DMM memory model. The implicit dummy model approach instead used the Threaded Cognition mechanism of ACT-R to implement a dummy model which occupied cognitive resources. In multiple experiments, we documented that both approaches for workload adaptation of computational cognitive models significantly improved the prediction performance in dynamic workload conditions compared to non-adapted models. In this subsection, we will compare both approaches, as they show distinct advantages and disadvantages. For this purpose, we look at four different criteria for comparison: validity within the architecture, generalizability, predictability and modeling complexity.

Regarding validity within the architecture, the dummy model approach provides a more natural fit to the ACT-R architecture as it does not introduce new mechanisms but works with any completely unmodified ACT-R 6 installation. In contrast, the overlay approach requires the introduction of new mechanisms and parameters to core modules of an architecture (the memory model in our case). As the numerous studies which validated the mechanisms of the DMM were performed using the original memory module without workload-dependent parameter sets, the overlay approach loses some validity compared to the unmodified model and the dummy model approach.

Generalizability is concerned with the question whether all aspects of cognition are effected by the adaptation. Workload adaptation using the dummy model approach is implicit as it does not directly interfere with the modeling components. Consequently, it can potentially influence any ACT-R module, as long as those resources are covered by the dummy model. The overlay approach explicitly influences only the memory aspect of cognition. Generalization of the approach to other modules would require the modeler to identify and optimize new parameter sets for those modules from scratch.

While its generalizability makes the implicit approach very versatile, a side effect of this approach is the limited predictability of the impact of the dummy model on the main task model. While this is a result of the implicit and generalizing nature of the approach and may lead to the identification of effects that would otherwise go unregistered, it also requires more engineering effort to avoid unexpected behavior (i.e. which modules are influenced in which regard): As the interference between main task model and dummy model depends on the cognitive resources required by both tasks, as well as the timing and order of resource requests, it is not easy to predict the exact effect of a model prior to its execution. In comparison, the explicit overlay approach modifies clearly defined elements of the parameter space, which have a known impact on predicted behavior and performance. This ensures that only the intended effects of workload level occur. In the analysis in Subsection 3.4.3, we saw that the optimized parameter sets reflected very plausible effects on cognition.

Next, we compare the two approaches with respect to modeling complexity. In Section 3.5.4, we saw that different dummy models have very different impact on the prediction accuracy. Using the dummy model approach therefore requires additional information on the type of secondary task to infer information about the required resources to execute it. This information could either be provided a-priori by the designer (which would lead to a static system design), or be derived from context information. Another possibility would be to employ empirical cognitive models to derive information about the workload characteristics of the secondary task (for example, in Section 3.4, we showed that it is possible to reliably discriminate two perceptual modalities, which are two types of cognitive resource in regard of the dummy model). The overlay approach is of low modeling complexity, as it only defines five existing variables of the declarative module which need to be set in the workload-dependent parameter sets. Adjusting those parameters for different workload levels requires only a straight-forward optimization (for example with a genetic algorithm).

Another aspect of model complexity is the fact that the dummy model approach requires a complex cognitive architecture, while the direct overlay approach can be easily implemented in an isolated model. This reduces the entry barrier for including workload adaptive computational cognitive models into interaction systems.

Table 3.10 summarizes the outcome of the comparison for all four criteria. We see that there is no approach which is superior to the other for all relevant criteria. This implies that the designer has to weight the different aspects to make a decision for a specific use case.

Criterion	Explicit Approach	Implicit Approach
Validity within Architecture	-	+
Generalizability	-	+
Predictability	+	-
Model Complexity	+	-

Table 3.10 – Comparison of explicit and implicit approach to represent the effects of workload level in a computational cognitive model.

3.6 Prediction of Learning Opportunities & Learning Situations

In the previous section, we demonstrated how empirical cognitive models can help to improve the predictive power of computational models in the context of dynamically changing user states. In this section, we will show that this collaboration of models is not necessarily an unidirectional relation, i.e. that the information flow is not restricted to the direction from empirical to computational cognitive model. We develop a joint model which relies on information from both a empirical and computational modeling component. In the evaluation of this joint model, we will show that it yields predictions on a user's state which cannot be extracted from the individual models. We will investigate this joint model for the use case of analysis of strategy learning during interaction. This is a frequent cognitive HCI task, for example when using the computer as a tutor to improve problem-solving skills and when learning about the operation of the computer itself, for example when exploring a new software. There are many opportunities for a system to support the user in such a situation. For example, a system could estimate to what extend the user already learned the information associated with the current learning item. This would help to predict in which situations the system can expect the user to act skillfully on their own and in which situations they still need support. The strategy of providing help or generating additional learning opportunities can then adapt to the predicted situation.

3.6.1 Computational and Empirical Models of Learning

Reinforcement Learning (RL) is a fundamental mechanism of adaptive behavior in humans. It is often implicitly involved in Human-Computer Interaction (e.g. when users learn to operate a new software) but can also be explicitly employed as part of a predictive user model for adaptive systems. The underlying computational models of the learning progress are usually individually calibrated through behavioral data (e.g. response probabilities). In recent years, neural activity (as measured by EEG or fMRI methods) has become a relevant source of information for empirical real-time user modeling. The feasibility of a joint computational model was illustrated by [ABFF10], who showed how the prediction of user states in an intelligent tutoring system can be substantially improved by providing a joint model which combines predictions of a computational cognitive model with an empirical cognitive model using fMRI data. However, in order to successfully apply this approach, neural markers need to be identified that can be integrated into user models and which are easily accessible (e.g. derived from EEG) in real-time during task operation. In this section, we employ a joint cognitive model consisting of a computational RL model combined with an empirical cognitive model based on EEG markers. The goal of this joint model is to identify learning situations in an associative learning task with delayed feedback.

In RL, organisms learn to select sequences of actions that maximize their received reward over time based on the reward signals (feedback) associated with task outcomes. This can be achieved through temporal difference learning (TD), which assigns credit based on the temporal proximity of actions to outcomes. The authors of [FA06] demonstrated how a TD-based RL model can predict learning performance by TD-based reward propagation in a complex associative learning task with delayed feedback. One neurophysiological approach for studying RL is to analyze the Feedback Related Negativity (FRN). The FRN is a frontocentral neural response appearing 200-300ms after the presentation of feedback indicating prediction errors (i.e., a mismatch between mental model and observation). [WA11] documents that prediction error can be used in a task with delayed feedback to predict the occurrence of FRN for task states immediately followed by feedback as well as intermediate states. The authors present this effect as evidence for credit assignment to intermediate states from future rewards. [CFKA10] moves from time domain analysis to frequency analysis and links prefrontal theta synchronization to adaption effects in a probabilistic reinforcement learning task. A Q-Learning model was used to estimate prediction errors, which indicated whether a situation reflects a learning opportunity. While the work mentioned above ([FA06, WA11, CFKA10]) explicitly addresses the processing of prediction errors, there are other cognitive processes and corresponding neurological markers related to learning events, for example working memory activity [CF12]. Early work on the relation of EEG synchronization/de-synchronization and memory processes has identified theta synchronization and alpha desynchronization during supposed memory processes [Kli96, JT02, WMR00, OTO^{+}06]. Regarding alpha oscillations, following research has also identified "paradoxial" alpha synchronization during cognitive activity, which in subsequent work [KDS^{+}99, SKD^{+}05, KSH07, THS^{+}07] was reinterpreted as a possible inhibition of task irrelevant cortical processes or conscious inhibition of cognitive processes impeding the task.

In this section, we establish a joint empirical and computational cognitive model of learning situations in a complex associative learning task, particularly considering memory encoding and feedback processing. We selected a complex learning task where a sequence of interdependent decisions is required to achieve a desired outcome. Learning such action sequences is both, common and important in HCI, for example when trying to achieve a particular result with an unfamiliar software. The frequency-based EEG-analysis of such a task in combination with a computational RL model – both in comparison of mean values and on a single-trial basis – is novel to the research community.

3.6.2 Task Design

The behavioral task employed is a modified version of the task used in [FA06]. Formally, it is an abstract tree-search which requires three binary decisions to move from the root node to a leaf node. Feedback about the success of a decision sequence is provided when reaching a leaf node. When reaching a non-target leaf node (failure), participants are moved back to the last node where they were still on path to the target. When reaching the target leaf node (success), one learning trial is complete and the participant is returned to the root node for the next trial. Semantically, the task is framed as a "strange machine", which has four buttons (red, yellow, green, blue) and a display showing its current state in a "unknown language" (a pronounceable but meaningless German expression such as "Tarfe"). In each state two of the buttons are active to move the machine into the next state. After three button presses, the machine either reaches the target node or a failure node and is reset as described above. The task goal is to learn to reach the target node without failures. To increase the learning load, each node has a set of three state labels, which are associated with different response options. At each visit of a node one state label from this set is randomly selected and displayed. We use the term *node* to reference an element of the internal representation and the term *state* to represent the external, visible representation of a node. A person who is operating the task is never directly informed about the current node he or she is in, only about the resulting state. We further define a *correct state* as a state for which a sequence of inputs exists which leads to the target leaf node without resetting. Any other state is called *incorrect state*. Finally, a *learned state* is a correct state for which the correct decision has been learned. See Figure 3.21 for a summary of the internal structure and the display of a node. Table 3.11 summarizes all important terms which are used to describe and analyze the "strange machine" task.

Term	**Explanation**
Node	Internal element of a machine.
Leaf Node	Node for which feedback (success or failure) is given.
Inner Node	Any node which is not a leaf node.
Target Node	Leaf node which successfully concludes a trial (success).
State	External visible representation of a node.
Correct State	State on a path to the target node.
Learned State	State for which the correct decision has been learned.
Step	Transition between an outgoing state and an incoming state.

Table 3.11 – Important terms for the analysis of the "strange machine" task.

Changes to the orignal task design of [FA06] were the reduction of state label set sizes from four to three (to reduce time to learn one instance), the removal of spatial terms (e.g. "left") for the button names (replaced by colors to avoid spatial cognition aspects), and the optimization for clean EEG recordings (no mouse input, visually compact display).

3.6.3 Data Collection

Using the "strange machine" task, data were collected from 34 participants. All participants were university students (23 female, mean age 23.1 years). Participants gave written consent and were paid for their participation. 18 participants completed two instances of the task, 16 completed only one.

The data collection procedure consisted of brief instructions followed by 15 practice trials and a main learning phase with 100 trials[12]. If participants completed the main learning phase in less than 45 minutes, a second learning phase with a differently labeled version of the machine was conducted. Stimuli were presented on a 20" screen, at approximately 1 m distance to the

[12] For the first 8 participants the main learning phase lasted 120 or 160 trials, which due to ceiling effects was subsequently reduced to 100.

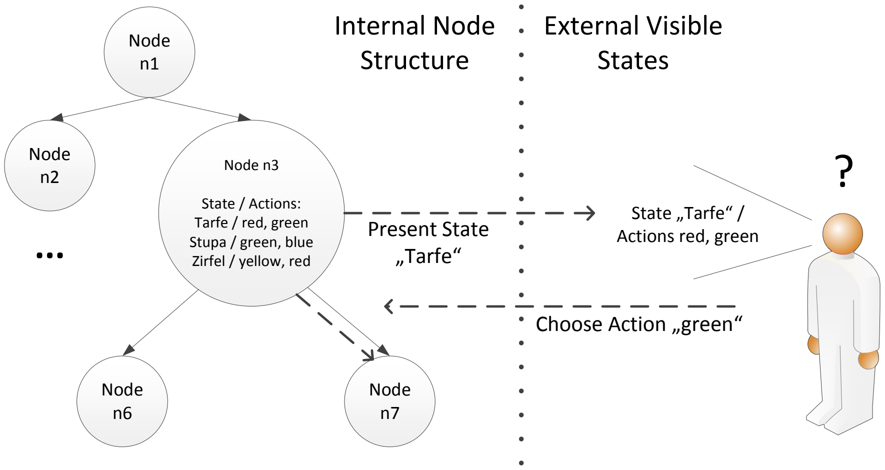


Figure 3.21 – Internal structure and external view of the “strange machine” task.

participant. All items were presented within a square of 2 cm to reduce the amount of eye movements necessary to perceive all relevant information. All relevant information on the execution of a task instance was logged to a file. This log file contained the sequence of presented states, together with the participant's choice for each state. Each event (state presentation and action selection) was annotated with a time stamp.

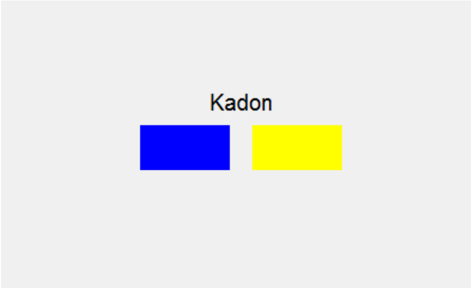


Figure 3.22 – GUI of the “strange machine” task.

EEG was recorded from 29 scalp electrodes placed according to the international 10-20 system using actiCAP active electrodes and actiCHamp amplifiers (Brain Products, Germany) at a sampling rate of 500 Hz with Cz as recording reference. The EEG data were re-referenced to a common average reference and segmented into windows of 400 ms length starting 100 ms after a new state is displayed. Windows containing ocular artifacts were identified and removed by testing for correlation of Fp1 and Fp2 above a threshold of 0.97 within the regarded time frame. This procedure rejects approximately 4.5% of all trials. This means each window contains data from processing the feedback (either a new state or direct feedback at a leaf node) following a decision step. Each window was normalized by subtracting the mean from 250–150 ms before stimulus. For band power analysis, we used the Thomson's multitaper power spectral density (PSD) estimate [Tho82]. The relevant (sub-)bands for analysis were estimated on an individual basis following the method of [Kli99]. The averaged PSD was then z-normalized for each participant.

3.6.4 Reinforcement Learning Model

Similar to [FA06], we used a Reinforcement Learning approach to model human learning behavior. We employed the Naive Q-Learning (NQL) algorithm (see listing 1), a variant of Watkin's $Q(\lambda)$ [SB98] to model the participants' learning progress. NQL is a Temporal Difference (TD) method with eligibility traces. The work of [WA11] demonstrates that TD methods are capable of reproducing human learning behavior and predict the generation of propagated FRNs. This work also demonstrated the benefit of eligibility traces for the purpose of closely fitting human behavioral data. Reward was selected to be $+7$ for the target node, -1 for the non-target leaf nodes and 0 for any inner nodes. Temperature and λ were fixed at 1.0 and 0.1, respectively. Learning rate α was optimized between 0.02 and 0.3 for each participant individually to account for the large inter-participant variance in performance. Each state label (not the node itself) is a state of the RL model, with two possible actions corresponding to the buttons of that label. For each session, a new model was initialized and trained using the action sequence as denoted in the corresponding maze log file. This allowed us to trace the learning from observation in each individual session. To quantify learning opportunities, we define *uncertainty* as the entropy of the Softmax probability distribution [SB98] resulting from the action Q-scores for a specific state. Until any feedback > 0 has been propagated to a state, this will result in a maximum uncertainty value of $\log 2$. When a state accumulates propagated rewards, uncertainty converges towards zero. As we can use this definition only for correct states, we define certain incorrect nodes to have a negative Q-score $< -\epsilon$ for both actions. The benefit of the notion of uncertainty compared to the classic notion of prediction error - which is defined as the update delta of the Q-score of the outgoing state for a certain step (see for example [WA11]) - is that it is defined in terms of states and not in terms of steps. Therefore, it can help a tutoring system to identify states which are not yet sufficiently well learned.

Algorithm 1: The Naive $Q(\lambda)$ algorithm

```
$\forall(s,a)$ : Initialize $Q(s,a) = 0$ and $e(s,a) = 0$, $trial = 0$
Set $\pi$ = Gibbs Softmax Method with static temperature $\tau$
while $trial$ ¡ max allowed trials do
  $\forall(s,a)$: $e(s,a) = 0$
  Initialize $s$, choose $a$
  for all steps in an episode do
    $s' \leftarrow DestinationState(s,a)$
    Choose $a'$ of $s'$
    $a* \leftarrow argmax_b Q^\pi(s',b)$
    $\delta \leftarrow r + Q^\pi(s',a*) - Q^\pi(s,a)$
    $e(s,a) \leftarrow e(s,a) + 1$
    for all $s,a$ do
      $Q^\pi(s,a) \leftarrow Q^\pi(s,a) + \alpha\delta e(s,a)$
      $e(s,a) \leftarrow \lambda e(s,a)$
    end for
    $s \leftarrow s'$
    $a \leftarrow a'$
    if s is end-state then
      end episode
    end if
  end for
  if $r > 0$ then
    $trial++$
  end if
end while
```

3.6.5 Analysis

We investigate the relation between the prediction of computational RL model and empirical EEG data to identify situations in which learning occurs. We do this in two main steps: First, we use the RL model to predict learning opportunities and look at neurological correlates in the EEG data. Second, we detect from those learning opportunities the learning situations which actually lead to correct decisions in the future. This second step shows how empirical model and computational model interact to identify learning situations better than each of them could do individually.

For the analysis of EEG synchronization and desynchronization (i.e. increase and decrease of power in a certain frequency band), we concentrate on two effects that are related to feedback processing and memory encoding: Theta synchronization in the prefrontal cortex and alpha synchronization in the occipital cortex. We average PSD between electrodes O1 and O2 to represent occipital activity and average PSD between electrode positions Fz, Fc1, Fc2 to represent prefrontal activity.

Identification of Learning Opportunities

We assume that memory encoding occurs systematically when new information on the task is learned from the feedback at the end a step. We therefore have to identify those situations which allow learning. For this purpose, we look for steps between an uncertain outgoing state and a certain incoming state. Such a step allows the transfer of knowledge about the certain step to the uncertain one. To separate the steps into classes, we use the RL model and apply two thresholds to dichotomize uncertainty: A strict threshold t_s (selected to characterize 80% of all values as 'high uncertainty') and a tolerant t_t threshold (selected to characterize 30% of all values as 'high uncertainty'). Those thresholds were chosen such that each of the resulting classes still contained enough (i.e. more than 10 on average for one participant) steps. We use t_s to label outgoing states as (un)certain and t_t to label incoming states as (un)certain. This choice minimizes the number of missed learning opportunities, as it maximizes the number of uncertain incoming and certain outgoing states. The left side of Figure 3.23 summarizes the class definition: Class `LEARN` denotes a learning opportunity, class `NO-INFO` denotes absence of a learning opportunity due to missing information and class `SATURATED` denotes absence of a learning opportunity due to an already saturated knowledge. We expect to see pronounced differences between `LEARN` on the one hand and `NO-INFO` and `SATURATED` on the other hand. We expect the latter two classes to be similar. To avoid class imbalance, we only include the first five occurrences of each state in each class in our analysis. Statistics are calculated on the normalized averaged PSD distributions for the respective classes as a two-sided paired t-test. To rule out that low-frequency ocular artifacts confound the results, we verified that there was no significant difference in eyeblink frequency between the different classes during preprocessing.

Figures 3.24 and 3.25 show average prefrontal theta power and average occipital alpha power calculated for the three classes separately. In Figure 3.25, we see an increase in alpha power from the `NO-INFO` class to the `LEARN` class in

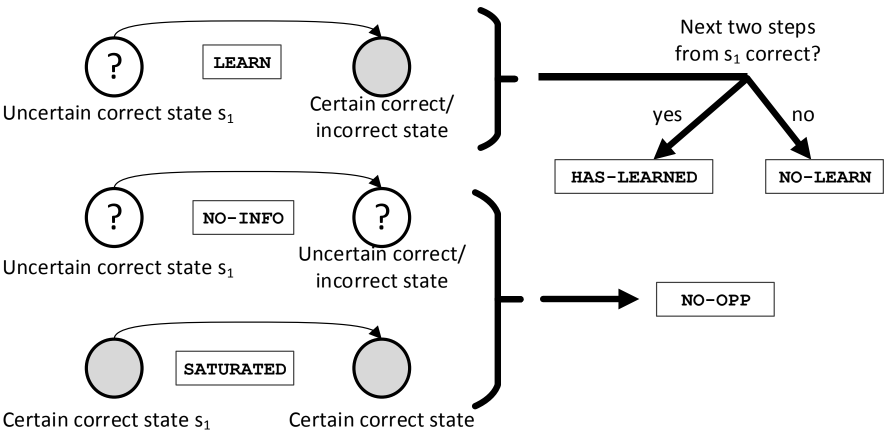

LEARN
Uncertain correct state s1
Certain correct/ incorrect state
NO-INFO
Uncertain correct state s1
Uncertain correct/ incorrect state
SATURATED
Certain correct state s1
Certain correct state
Next two steps from s1 correct?
yes
no
HAS-LEARNED
NO-LEARN
NO-OPP

Figure 3.23 – Definition of learning opportunities (left side) and learning situations (right side) as derived from the RL model to form the classes for evaluation of classes.

the occipital cortex, while there is no significant difference between `NO-INFO` and `SATURATED`. Analogously, we see a difference between `NO-INFO` class to the `LEARN` and `SATURATED` classes in the theta band for the prefrontal cortex in Figure 3.24. However, those differences in the regarded bands marginally miss statistical significance: $t(36) = 1.48$, $p = 0.07$ for occipital alpha and $t(36) = 1.62$, $p = 0.057$ for prefrontal theta. One reason for this lack of significance is that learning opportunities denote the potential for learning, but do not always lead to memory encoding as the participant overlooks the opportunity or is not able to correctly memorize the new information.

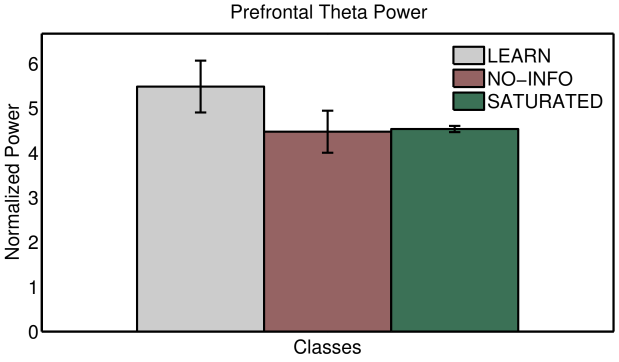


Figure 3.24 – Theta power at the prefrontal cortex for the classes `LEARN`, `SATURATED` and `NO-INFO` for learning opportunities. Whiskers indicate standard error.

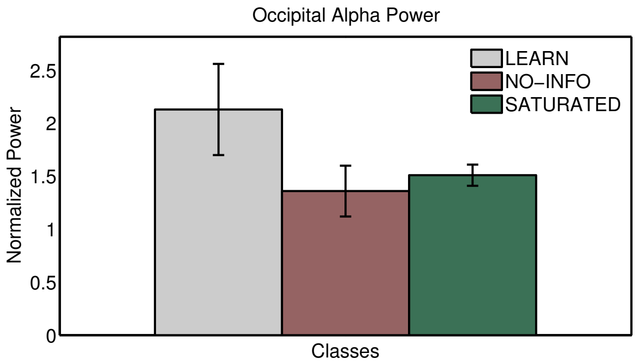


Figure 3.25 – Alpha power at the occipital cortex for the classes `LEARN`, `SATURATED` and `NO-INFO` for learning opportunities. Whiskers indicate standard error.

Identification of Learning Situations

The criteria we defined in the RL model yield a reasonable prediction whether a learning situation occurs during a specific step. In the previous analysis, we assumed the definition of a learning situation as a given ground truth to investigate neurological markers for learning. However, we concluded that the computational model can only yield a noisy prediction of a successfully learning event. To quantify this predictive power, we introduce the term of a *learned state*. A learned state is a correct state s for which holds that the next two steps in the log file which have s as outgoing state stay on the correct path. 38% of all steps labeled as learning situations do not result in a learned state[13]. In the following, we combine this prediction by the computational RL model with the information of EEG to detect those missed learning opportunities. We propose that the observed alpha and theta synchronization effects are caused by cognitive processes of learning situations. This implies that when separating learning opportunities in learned and not-learned outgoing states, we should observe a similar difference in PSD: Learned outgoing states show a level of alpha and theta synchronization which is not present for missed learning opportunities. To investigate this hypothesis, we sort the steps from the `LEARN` class of the positive and negative learning opportunities by this criterion, forming the `HAS-LEARNED` and the `NOT-LEARNED` classes. Steps which are not categorized as learning opportunities form the `NO-OPP` class, see the right side of Figure 3.23. Fig-

[13] This number depends of course on the threshold applied to the uncertainty level of the outgoing step. A lower threshold leads to fewer false alarms but also increases the number of missed learning opportunities.

ures 3.26 and 3.27 show the band power for the three different classes, now resulting in a significant ($t(35) = 2.74$, $p < 0.005$) increase in individual alpha power from the non-learned to the learned steps, as well as a significant difference in theta power ($t(35) = 1.76$, $p < 0.05$) in the prefrontal cortex. The steps in the `NOT-LEARNED` class are not significantly different from steps in `NO-OPP` for both occipital cortex and prefrontal cortex.

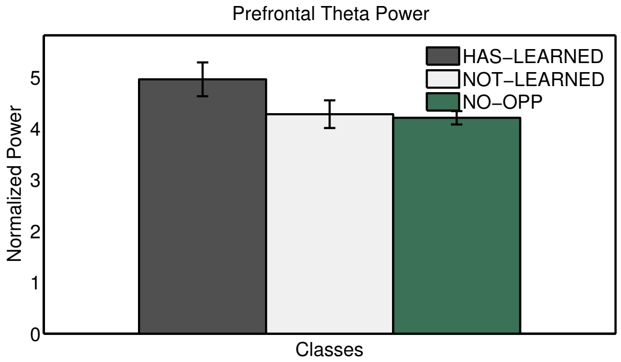


Figure 3.26 – Theta power at the prefrontal cortex for the classes `HAS-LEARNED`, `NOT-LEARNED` and `NO-OPP` for learning situations. Whiskers indicate standard error.

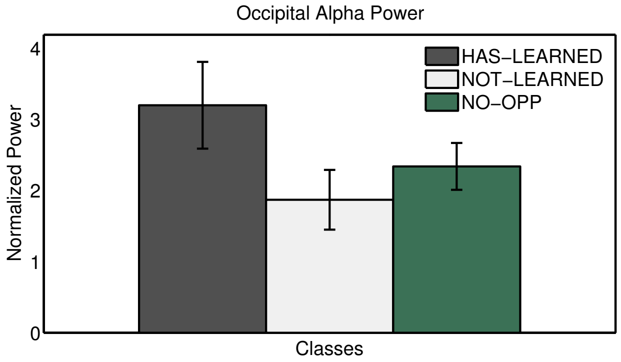


Figure 3.27 – Alpha power at the occipital cortex for the classes `HAS-LEARNED`, `NOT-LEARNED` and `NO-OPP` for learning situations. Whiskers indicate standard error.

We should note that the different classes are not distributed equally in time over the course of one session: For example, steps that constitute learning opportunities are rare at the beginning and the end of a session. To exclude that the observed effects are of temporal origin, we compare steps between inner nodes from the beginning and the end of each session. We do not see a significant difference between steps at the beginning and at the end. This indicates that the previous results measured a real learning effect and not a temporal effect.

Note that while the empirical model is able to differentiate actual learning situations from missed learning opportunities, it still requires the RL model to identify learning situations: When we remove the prediction of learning opportunities and directly compare PSD distribution in prefrontal and occipital cortex for learned and not-learned states, the previously observed difference between both classes becomes non-significant in this condition ($p > 0.1$ for both regions and the corresponding bands). This indicates that we are able to identify learning situations but that this is only possible using both, the computational and the empirical model.

Single Trial Classification of Learning Situations

To make this significant difference accessible for a tutoring system, we need to provide prediction of learning situations on a single trial basis. For this purpose, we train a Naive Bayes classifier to separate the `HAS-LEARNED` and the `NOT-LEARNED` class. As features, we use individual occipital alpha power and prefrontal theta power. We evaluate this classifier in a participant-dependent leave-one-out crossvalidation. To exclude cases where one class receives too few training samples, we remove the most imbalanced sessions where the majority class contains more than 70% of all samples from the analysis. The resulting classifier yields an average recognition accuracy of 71.0% which is significantly better ($t(25) = 2.49$, $p = 0.01$) than the baseline accuracy of 59.6% (relative frequency of majority class `NOT-LEARNED`), as determined by a one-sided paired t-test of classification accuracy vs. size of majority class for each subject. The average improvement over the baseline is 19.7% relative.

To conclude, our results show that we can use the computational RL model to identify learning opportunities in an associative learning task, despite delayed feedback. We further showed that we can combine the model with an EEG based empirical model to predict learning success. This is also feasible on a single trial basis. The empirical model and the computational model each on their own were not able to perform this prediction reliably.

3.7 Discussion

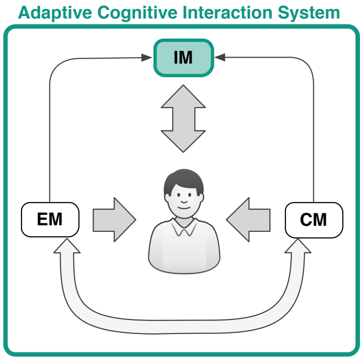
In this chapter, we described the development of computational cognitive models for use in adaptive cognitive interaction systems. We used those models to predict two complex cognitive user states, memory configuration (in Section 3.3) and learning situations (in Section 3.6). Empirical cognitive models would not have been able to predict those user states on their own. This does not mean that computational cognitive models exist independently of empirical cognitive models. On the contrary, we presented and compared two different approaches to adapt the predicted behavior and performance of a computational cognitive model to different workload levels, as estimated by an empirical cognitive model. These extensions to computational cognitive models improved the prediction of individual performance in situations of dynamically changing workload, which is an important contribution to the application of models in an HCI context. In the final Section 3.6, we showed that both, computational and empirical cognitive models can contribute to the joint prediction of a user state. Taken together, the results in this chapter show that a combination of different types of models is can predict user states which could not have been predicted accurately with only one type of model. These results are new to the research community and will advance the research on cognitive modeling for interaction systems.

Chapter 4

Interaction Manager & Usability Evaluation

In this chapter, we describe the development and evaluation of an adaptive interaction manager AIM. Following the presentation of the general interaction framework, the focus of this chapter is on multiple user studies to investigate several objective and subjective usability aspects of adaptive cognitive interaction systems. Finally, we outline the development of a cognitive user simulation.

4.1 Introduction



In the last chapters, we described our contributions to empirical and computational cognitive modeling. We demonstrated that cognitive models are able to model user states and cognitive processes. In this chapter, we leverage those results to develop end-to-end adaptive cognitive interaction systems. For this purpose, this chapter concentrates on two aspects: On the one hand, we turn our attention to the final component of such interaction systems, the interaction manager which engages the user and manages input, output and discourse. On the other hand, we investigate objective and subjective usability aspects of adaptive cognitive interaction systems to determine whether adaptive interaction is beneficial to the user in a measurable way.

We start by reviewing the relevant related work, regarding interaction management in general and adaptive interaction management and user modeling in particular. Afterwards, we introduce our own AIM adaptive interaction manager for a rapid development of adaptive multimodal interaction systems. Then, we present the design and evaluation of three different interaction systems which were implemented using AIM. We investigate several usability aspects of adaptive systems. The first system is an end-to-end workload-adaptive information presentation system which adapts its speaking style and mode of information presentation to the user's workload level. We show that such adaptation yields significant benefits in effectiveness, efficiency and user satisfaction. For the second system, we investigate the effect of different workload-adaptation strategies in terms of intrusiveness. Our results indicate that there is a discrepancy between the objective benefit of a strategy and the subjective assessment by the users. Third, we describe a system which detects and reacts to the user's confusion resulting from recognition errors by a gesture recognizer. We show that a pro-active reaction to confusion by the system increases the robustness of the recognizer with higher efficiency than manual correction methods. Finally, we look at the development of a cognitive user simulation, which incorporates computational models of memory and workload. We show that such simulation is able to generate plausible interactions.

4.2 Related Work

In this section, we review related work, i.e. the fundamentals of interaction management and the State-of-the-Art in terms of adaptive interaction systems and user modeling and simulation.

4.2.1 Fundamentals of Interaction Management

In this section, we present the fundamental components of an interaction system. We furthermore give some examples of interaction systems to show how those components can be implemented. The term interaction system refers to any system which receives input from the user and outputs information to the user via one or multiple modalities. In this thesis, we adopt

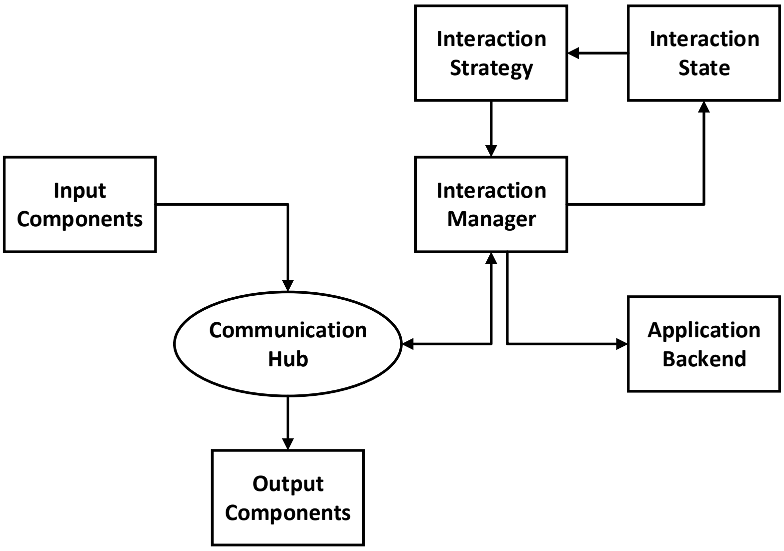


Figure 4.1 – Major components of an interaction system. Adapted from [BR09].

the terminology from spoken dialog systems as those are interaction systems which handle the most complex input and discourse structures. Since spoken dialog systems have evolved to multimodal interaction systems during the last decade, their design forms a general basis for all, also non-speech based interaction systems. The general structure of such an interaction system is depicted in Figure 4.1. A *communication hub* connects the input and output components with the central interaction manager. Examples of input components are Automatic Speech Recognition (ASR) combined with a Natural Language Understanding (NLU) unit, a gesture recognizer or simply a keyboard. Examples of output components are Text-to-Speech (TTS) systems, a Graphical User Interface (GUI) or a virtual 3D avatar. The *interaction manager* receives input events via the communication hub. Using this information, it updates the multi-dimensional *interaction state* which stores all information to represent the progress of the interaction, the assumed state of the user, and other context information. The interaction state is then used by the *interaction strategy* to decide on the action of the system, which is then executed by the interaction manager. Actions can either trigger activity of the application backend or be propagated to the output components.

For implementing the interaction strategy, there exists a number of approaches ranging from very simple, rigid architectures to flexible, generic approaches. [McT02] categorizes those approaches into three groups. The first one, based on Finite State Machines (FSMs), is one of the simplest ways to implement an interaction strategy. Here, the interaction state is identical to the current state of the FSM. Outgoing transitions between states represent the allowed user actions or the encoded system actions taken in that state. One of the main challenges of FSM based strategies is that they do not scale reasonably well with the number of parameters which define the interaction discourse and the number of possible user inputs for each state [McT02]. The second category is called “frame-based systems”, which are representatives of the more general category of systems using the Information State Update (ISU) approach [LT00]. Here, the strategy is implemented as a set of rules which operate on the interaction state. This state comprises variables which represent “the mental state of the agent” or a “structural view of the dialogue” [LT00]. Compared to strategies based on FSMs, the frame-based approach is more flexible as it allows implicit definition of the state-action space. Strategies based on this approach therefore scale much better and offer high flexibility to include all relevant attributes within the interaction state. The third category in [McT02] is “agent-based systems”. Those systems model the user and the system as agents which try to collaboratively solve a joint task. Agent-based systems use techniques from general Artificial Intelligence to represent interaction state and select system actions. Implementations of this approach use logical interference or planning algorithms. While the author of [McT02] himself promotes agent-based systems as the most advanced category of interaction strategies, this evaluation is actually not clear-cut. The behavior of agent-based systems is very difficult to understand and predict for the interaction strategy designer. We will see later in this section and also in our own contributions (Section 4.5) that small differences in system behavior can make a large difference in the usability evaluation. Controllability and predictability of system behavior for the designer is therefore a critical property of an adaptive interaction strategy.

Another approach to interaction strategies which has recently (i.e. after the publication of [McT02]) caught the attention of many researchers is based on Reinforcement Learning (RL). This approach aims at learning optimal system behavior from data or simulation. Strategies based on RL represent the dialog state as a (Partially Observable) Markov Decision Process ((PO)MDP). On this (PO)MDP, an optimal policy is learned. A challenge of this approach is that it requires a lot of training data or a realistic user simulation to generate training episodes. This is feasible as long as the interaction state is of low dimensionality but becomes challenging [WY07] once more variables are added to it. For adaptive interaction systems which incorporate information on the user state, we expect the interaction state to be of high dimensionality, for example representing detected user states. Therefore, application of RL is not always feasible. Additionally, training of adaptive strategies requires the availability of data or of sophisticated user simulations to reflect the user behavior in different user states. Such resources are rarely available, as the number of possible combinations of domains and user states is very large.

The general framework of interaction management is brought to life by several existing interaction system frameworks. The TAPAS interaction manager [Hol05] is based on a rule-based architecture. It maintains a set of slot-value pairs to keep track of the dialog progress and an abstract dialog state. TAPAS evaluates the slots and the dialog state when selecting rules eligible for execution. TAPAS has been applied to build interaction systems which autonomously acquire new information about people [PH08] or objects [HNW08]. Olympus is an interaction system [BRH$^+$07] which includes the Ravenclaw interaction manager [BR09] as central component (together with modules for ASR and TTS). Ravenclaw consists of two main components, the task tree and the agenda. The task tree represents the structure of interaction plans for achieving certain goals in the target domain. The agenda is a stack of agents which represent the current focus of attention in the interaction. Agents consume matching input from the user to be activated and to update their state. The OpenDial interaction framework [Lis14] is a toolkit which is based on probabilistic rules for interaction management, defined in an easily accessible XML format. Probabilistic rules allow the implicit definition of graphical probabilistic decision models. While the value of designated parameters of those rules can be automatically learned (e.g. with RL), the structure of the rules is defined manually by an expert to specify domain knowledge.

While the presented approaches are generally able to represent a user state as part of their interaction state, special consideration is required to implement the adaptation mechanisms. [FDH12] develops a taxonomy of different types of adaptation. The authors define four main mechanisms: “Modification of function allocation”, “Modification of task scheduling”, “Modification of interaction” and “Modification of content”. The authors give numerous examples of how those generic mechanisms could turn into concrete mechanisms for certain applications. For a workload-adaptive system, those mechanisms could be directly applied: Function allocation and task scheduling allocate the right tasks at the right time to the user, while assigning others (e.g. tasks which are too difficult in the given situation or which are well-suited for automation) to automatic processing. Modification of interaction refers to change in presentation style to accommodate a certain level of workload, for example by modifying speaking rate of synthesized speech. Modification of content could for example be realized by limiting choices for selection from a menu.

While those mechanisms characterize the desired effects of adaptation, [Jam09] systematically analyzes a number of usability challenges to be addressed for the design of adaptive interfaces, which may not be visible immediately. The main points are: “the need to switch applications or devices” (e.g. because non-standard sensors are required), The “need to teach the system” (e.g. because a statistical user state model needs calibration data for each user), “narrowing of experience” (e.g. the user does not learn to handle high workload situations themselves), “unsatisfactory aesthetics or timing” (e.g. because of automatic system adjustments instead of handcrafted behavior patterns), “need for learning by the user” (e.g. because the user has to accept that they must not compensate for high workload but should let the system intervene), “inadequate control over interaction style” (e.g. because the adaptation removes certain menu items), “threats to privacy” (e.g. as the user is constantly monitored), “inadequate predictability and comprehensibility” (e.g. as workload recognition and adaptation triggers are not transparent), and “imperfect system performance” (e.g. as workload recognition still has double digit error rates for most applications). As indicated by the given examples, all of these aspects are directly relevant for workload-adaptive interfaces and need to be considered. While there exists literature which indicates the general feasibility of workload adaptive systems (see next subsection), those aspects have not been considered to the full extend.

The claim of building adaptive cognitive interaction systems is that those systems will improve human-computer interaction. However, there is no universal metric to quantify the quality of an HCI application. [MEK$^+$09] define a large taxonomy of usability aspects from hedonistic to pragmatic criteria, which all influence the final acceptability of a system. Therefore, we need to clearly define the criteria by which we evaluate adaptive cognitive interaction systems. We claim that adaptive systems are more robust, more efficient and more satisfying for the user than static, non-adaptive interaction systems for the same task. All three criteria are measurable in an evaluation: Robustness, we can measure by evaluating recognition accuracy of an input recognizer or by evaluating task success metrics, i.e. number of solved sub-tasks. Efficiency, we can measure as information throughput or using a cost metric of interaction. User satisfaction, which is a subjective quality criterion, we can measure with questionnaires.

4.2.2 Adaptive Interaction Systems

In this section, we look into the related work on adaptive interaction systems which model and timely react to the users traits and states, for example by adapting parameters of language generation components [MW10]. One of the earliest targets for adaptive interaction systems was the emotional user state. For example, [FRL11] presented a wizarded tutoring system which was adaptive to the uncertainty level of the student. In follow-up research, [FRL12] investigated the automation of the system behavior for adaption. [GR08] described a gaming interface based on emotional states using discourse features like history of interaction and actual user command. [NL07] described a user modeling approach for an intelligent driving assistant, which derives the best system action (in terms of driving safety) given estimated driver states. States included emotion and personality, partially derived from physiological measurements like heart rate. [Con02] presented an educational dialog system that decides for different user assistance options based on the emotional state. The work applied the cognitive OCC appraisal theory (by Ortony, Clore and Collins), which relates the users' emotions with their goals and expectations. Pattern-based adaptation approaches have been proposed which base an adaptation decision on the observation of user behavior and internal state changes [BM10]. The adaptation patterns described recurring problems and proposed solutions. Affective computing has been most successful in the area of virtual agents [Con13] and tutoring systems [BANAG10]. This seems plausible as both are domains in which a broad variety of emotions is intrinsically present. For general purpose applications, often the only emotion which is consistently present is anger [BvBE^{+}09] as a post-hoc appraisal of interaction problems. While this may be useful as an indicator of erroneous system behavior (see also below), it limits the impact of affective interfaces for general purpose applications for scenarios in which full-blown (and therefore easily detectable) emotions are rare.

There are also a number of systems which adapt to the user's workload level. We can find most work in this regard in the domain of adaptive automation. The authors of [WLR00] evaluated workload recognition for the Multi-Attribute Task Battery, using six EEG channels and other physiological sensors, combined in a neural network for classification. In high workload condition, two of the subtasks were turned off. In a study with seven subjects, the authors showed that this adaptation improved the performance for the remaining tasks drastically. [KDB^{+}07] showed that a real-time detection of workload can be successfully used to manage distractions while driving in real driving situations. The authors showed that a real-time recognition of workload (induced by a cognitively complex distraction task) based on band power features of the EEG can improve reaction time significantly in a simple, response task which mimics the user's reaction to a traffic warning system. The system suppressed this response task when high workload is detected and thus only requires a response in low workload situations. [CV04] employed a similar adaptation mechanism to reduce distraction from cell phone notifications in high workload situations. User state was assessed by a fusion of multiple physiological modalities. Corresponding to the detected workload level, one of four notification styles (ring, vibrate, silence) was selected, according to the indicated user preferences.

In [WR07], subjects controlled a remotely piloted aircraft in scenarios with variable difficulty levels. The authors evaluated an adaptation mechanism which reduced task speed and memory load when triggered by an artificial neural network classifying workload in real-time. The results showed that physiologically triggered adaptation was significantly better than random interventions. However, the effects were very different for low and high performers of which the latter benefit more from a targeted adaptation. The authors of [CE13] used a similar scenario. The presented system provided support in form of target highlighting. Their main finding was that the effects of co-adaptation only occurred at the third day of the experiment. Before that, a manually triggered adaptation proved more effective and less subjectively demanding. The authors speculated that this effect is caused by changes in subjects' strategies or in an active or indirect modification of the generated EEG signals, similar to a biofeedback system.

[BSF$^+$06] compared the effect of adaptive automation versus adaptable automation for two different tasks. While an adaptive system adjusts itself based on observation of their users, an adaptable system may be modified by the users themselves to fit their current needs. In the adaptive case, automation was triggered by a threshold on the engagement index calculated from relative EEG band power. The authors saw that performance increased and subjectively perceived workload decreased for the adaptive system which did not require user-initiated intervention. They also observed that for the adaptable case, users hesitate to manually activate automation even in situations in which they would benefit from doing so. There is also evidence that the user's workload level is relevant for spoken dialog systems. In [VL11], the authors investigated the effect of different interaction strategies of users at different workload levels (assessed from questionnaires and biosignal analysis) in a driving situation. They revealed that a guided interaction strategy of the system was less cognitively demanding in high-workload situations compared to a open, user-initiative strategy. [GTH$^+$12] also reported that user behavior differs between different workload levels. Those behavior differences comprised of user obedience to system requests, response behavior to confirmations, the number of barge-ins and internal consistency of user answers. The authors concluded that “the system should alter its behavior to match user behavior”.

One example of a study on effects of adaptation on subjective usability aspects is given in [GCTW06]. Here, the authors explored different adaptation strategies to promote functionality which was frequently or recently used in the past. They investigated both the perceived benefits as well as the perceived costs (e.g. caused by suboptimal choices of the system or its unpredictability) and show that even slightly different strategies are located at vastly different locations in the benefit-cost space. They also show that there is no general relationship between perceived benefit and user satisfaction. In [GET$^+$08], the authors further explored the effect of predictability and accuracy of adaptive graphical user interfaces on usability.

Overall, we conclude that there is research on the development of workload-adaptive interaction systems and that the general effectiveness of such systems was demonstrated. However, there are limitations on the current State-of-the-Art: Most user studies concentrate on purely objective aspects of task performance, and disregard other aspects of usability, including subjective measures of user satisfaction. The systems under investigation are also limited to unimodal input based on mouse and keyboard and output via graphical user interface. In our contribution to this field, we will investigate multimodal adaptive interface and look at both objective and subjective success criteria.

4.2.3 Reinforcement Learning and User Simulation

In this section, we describe systems which use Reinforcement Learning (RL) to automatically determine an optimal interaction strategy. We focus on examples which use RL to train strategies which adapt to certain user states or traits. To train interaction strategies requires to iterate a large number of training episodes. Usually, those episodes are provided by a user simulation, which generates plausible user’s actions depending on the interaction context and the last system utterances. When looking at adaptive systems, the user simulation needs to model the user states which are used for adaptation and the influence of those states on the user’s behavior.

The application of RL for the optimization of dialog strategies has already been established in the dialog community. Its applications range from simple early systems [LKSW00] to more systematic investigations in more complex domains [RL10]. While the application of RL for industry applications is still a matter of discussion (because of high computational complexity and difficulty of maintenance) [PP08], it is established in the research community for its ability of automatically optimizing complex strategies in non-trivial state and action spaces. We can also identify some works which extend RL approaches to include information about a user's states or traits. For example, in [BPNZ06] RL was used with partially observable Markov models, which include information on the user's affective state, such as stress. In simulation, the authors show that the learned strategy handled stress-induced user errors better than handcrafted strategies. [TŢM08] presented a speech-enabled therapist robot which adapted its personality to certain personality traits of its rehabilitation patients. The system used online RL to train its behavior. [SH10] used RL to integrate adaptive turn-taking behavior depending on the urgency and certainty expressed by the simulated user. However, this approach lacked an integration of models which take cognitive processes and inner states explicitly into account to enhance the statistical models with prior knowledge about human behavior.

Due to data spareness and the complexity of the optimization problem faced, nearly all works which apply RL for strategy optimization rely on some kind of user simulation. The most prominent types of user simulations are statistically motivated and range from simple bigram models, which models the most likely user action as reaction to the last system action (first introduced by [ELP97]), to more advanced Bayesian Networks [PRI09] and inverse RL for imitation learning from experts [CGLP11]. A survey on those statistical user simulation techniques can be found in [SWSY06]. [RL11] is an example of a statistical user model for Reinforcement Learning. The user simulation was trained on data of multimodal human-computer interactions gathered in a Wizard-of-Oz setting. The authors showed that the learned strategy outperformed those executed by the individual human wizards, as it could learn optimal behavior from data of different humans. Purely statistical user simulation techniques create behavior which is satisfyingly realistic when regarding each utterance separately. However, they lack coherence over longer periods of time and adaptivity to changing conditions which are not represented in the training data in more complex interaction scenarios. Approaches to counter this problem usually introduce some kind of (at least partially) non-statistical, rule-based model. One example is the agenda based simulation approach [STW^{+}07, KGJ^{+}10]. Here, a regularly updated agenda of the user determines the user's goals and derived action plans and can handle multiple, changing and stacked goals. Another example is the hybrid MEMO [MEE$^+$06] system which focused on the rule-based modeling of errors made by the user due to misunderstandings caused by the interface. Injection of expert knowledge to statistical user simulations in the form of rules or in the form of data selection was also used to represent different user traits and states. For example,[LCGE$^+$11] modified the appropriateness of the generated user utterances regarding the previous system request to represent different levels of cooperativeness. [JL09] simulated users of different levels of expertise by including a corresponding variable to a statistical model for expression generation in a helpdesk domain. [GWM10] showed that by selecting training data from different age groups, they were able to train a user simulation that models the behavior of young and old users differently. [Eng14] used the concepts of needs and plans derived from needs as described in the PSI architecture [DSS99] to build a cognitive user simulation. Needs were used to enable and weight different subgoals to determine the generated speech acts of the simulated user. While most models for user simulation are static during the interaction, [EH05] is an exception, in that it used multi-agent learning in the context of dialog systems to train the stategies of two agents: a dialog system and a simulated user were represented as learning agents in a RL framework. The approach is used to generate dialogs which require agreement of both agents on certain items.

4.2.4 Error-aware Interfaces

In this subsection, we focus on interfaces which are able to automatically correct recognition errors which occur during the recognition of user input. Such error-aware interfaces are able to deal with the user state confusion, which results from erroneous system feedback following such recognition errors. There exists a number of systems which make use of confidence scores to estimate the presence of recognition errors [GHW08, SSYH12]. However, when statistical models are unreliable and generate incorrect results, it is unreasonable to expect a very reliable confidence estimate. For example, [VKS10] shows that confidence scores in ASR correlate well with recognition performance for well-trained models but confidence reliability starts to deteriorate for models which are trained on little data or data which does not match the testing data.

Therefore, researchers investigated options to more directly predict erroneous system behavior. In Chapter 2.5, we showed how Error Potentials (ErrPs) can be extracted from EEG signals as markers of unexpected system feedback caused by misinterpretation of the user's input. HCI research investigates how ErrP detection can be leveraged to construct error-aware interfaces. Unsurprisingly, the idea to use ErrP detection for improving interaction has been first introduced in the context of Brain-Computer-Interfaces (BCIs), for which the necessary equipment is already in place. BCIs as input or control device suffer from far-from-perfect recognition rates. A standard technique to remedy this is to always repeat each input several times. This increases robustness but leads to a low transfer rate [KSM$^+$08]. The detection of ErrPs allows to increase accuracy and therefore increase the potential transfer rate. [CCM$^+$12] showed how to detect ErrPs during operation of a P300 speller BCI. They suggested to use the second best recognition result of the BCI in case of a detected ErrP and showed in simulations that it improved performance. [SBK$^+$12] pursued a different approach and deleted the previously given input in case an ErrP was detected and prompted the user to repeat. They showed that the use of an online ErrP classifier to significantly increase transfer rate.

[LvGG$^+$11] used detected ErrPs to adapt the weight parameters of a logistic regression model for BCI operation to better represent the (assumingly) misclassified trial. They use simulation and offline analysis of data from eight subjects to show that this process improves classification accuracy. Similarly, [FBC$^+$10] used classification of ErrPs during operation of a gesture recognition system to improve its performance by adaptation of the gesture recognizer to different users. In contrast to our proposed approach in Section 4.6, their system did not immediately react to the detected ErrPs by error correction. Trials which were classified correctly (i.e. did not result in an ErrP) were added to the training data to train a personalized gesture recognizer. This selection of adaptation data addressed the challenge of unsupervised adaptation that the addition of misclassified trials can result in performance degradation instead of improvement.

[VS12] proposed an ErrP recognition system based on a consumer-level EEG device. They performed subject-dependent two-class ErrP classification, using a test set of 80 trials of Flanker Task execution and achieved a recognition accuracy of about 0.7. The authors did not report how the classes are distributed in the test data but showed a similar accuracy for both classes. Using a simulation of ErrP classification with different error rates, they showed that already an ErrP detection rate between 0.65 and 0.8 can be beneficial for the enhancement of interactive systems in order to detect user errors spatial selection with the Flick technique on a touch surface. The authors analyzed accuracy improvements of allowing manual corrections when an ErrP is detected, but do not analyze costs or other usability aspects.

While there exists a large corpus of usability investigations on gesture-based interaction systems, we are not aware of studies on the impact and handling of recognition errors. However, recovery from error states is a required feature for all types of interaction in which the interpretation of user input is error-prone. Most advanced are probably spoken dialog systems. [HG04] described strategies for an interactive humanoid robot to recover from dead-end situations during dialog. They showed that by allowing repeated inputs in cases of an inconsistent discourse, the number of completed task goals could be substantially increased compared to a system without recovery strategy. [BR08] described ten distinct strategies to recover from non-understanding situations (i.e. the speech recognizer detects speech but does not yield a result) and empirically evaluated the performance impact of the different strategies. Most of the presented strategies involve one of several alternatives of “reprompting” the user after a non-understanding. [ZSHM10] performed a similar study comparing different error recovery strategies and concluded that. Research in speech processing uses confidence measures to identify erroneous sections of a speech recognition result and propose n-best hypotheses [SSYH12].

4.3 AIM: Adaptive Interaction Manager

To realize adaptive cognitive interactive systems, we developed our own light-weight interaction manager: The Adaptive Interaction Manager (AIM). The goal of this interaction manager was to provide a flexible architecture for multimodal interaction systems with a focus on adaptation to user states. The basic architecture of the framework is depicted in Figure 4.2. It is based on an event handling mechanism to which an arbitrary number of input modules can be connected. Input modules act send messages to AIM which convey information about the user's input, information about the user or changes in the environment to the system in form of events. Messages are converted to events, which are queued and regularly evaluated in the event loop. Each event belongs to a certain event type. For each event type, a handler is registered. Those handlers define the integration of an event into the interaction state. The variables which form the interaction state are an abstraction of the relevant interaction parameters of the current session, for example the type of the past system actions and user actions or the detected workload level. The interaction state also encapsulates any computational modeling components which are used during interaction. The interaction state is used by the interaction strategy to select the actions of the system. The interaction strategy can also decide to take no actions. All actions which are scheduled by the interaction strategy are queued and than executed within the execution loop. Scheduling, execution and termination of actions trigger events which can in turn be used to update the interaction state. As long as an action is scheduled but not executed, the execution schedule can be modified by (re)moving queued actions. Actions which are executed but not terminated can be aborted. This allows the interaction manager to react in an adaptive fashion to changing user states, which may render actions which are currently scheduled or in the process of execution as inappropriate to the new state. The interaction manager can segment longer actions into sub-units to allow switching interaction behavior during the execution of an action (e.g. because of a rising workload level).

AIM allows the application of different interaction strategies. Besides an implementation for strategies trained by Reinforcement Learning, the most mature implementation uses a rule-based information state update approach. In each iteration of the execution loop, a set of rules is evaluated for execution. Rules determine the actions of the interaction manager. They consist of two components: The preconditions formulated using the variables of the interaction state and execution bindings accessing the connected external components. If the preconditions, which test for certain configuration of the interaction state, of a rule are fulfilled, it is activated and the connected system actions are scheduled for execution. Rules are assigned a priority which determines the executed action in case that multiple rules are eligible. This architecture follows the Blackboard design pattern where input from various sources is gathered in a central data structure which is used to decide on the selected system actions. It should be noted that this architecture is similar to the symbolic rule selection mechanism of ACT-R, see Section 3.2.1.

The current implementation of the AIM supports a large variety of input and output components. Examples of such input modules are a speech recognizer, a user state recognizer, a head movement tracker or a keyboard. For an adaptive cognitive interaction system, empirical cognitive models form one of the most important types of input sources. An empirical cognitive model runs as a stand-alone component, sending events to the AIM regularly or when changes in user state are detected. Examples for supported output components range from text console output over several Text-to-Speech (TTS) engines (currently supported: OpenMary[1] and any TTS using the Microsoft

[1] http://http://mary.dfki.de/

Speech API[2]) to graphical output on a computer screen. This variety of output components allows the system to chose appropriate output modalities, e.g. when it observes that perceptual workload of a certain modality is higher than of others, see Section 2.4. The interaction manager can also connect to the Thinking Head [LLP11], a 3D avatar which supports lip-synchronous speech synthesis. The *Interaction Manager* can send both, TTS commands as well as visual commands to the avatar. This allows to trigger synthesized mimics accompanying the speech output. For TTS, the interaction manager is able to set the parameters of the selected voice via Speech Synthesis Markup Language[3], enabling adaptive speech synthesis. For example, the system can speak more slowly or add emphasis when the system detects a high workload level of its user.

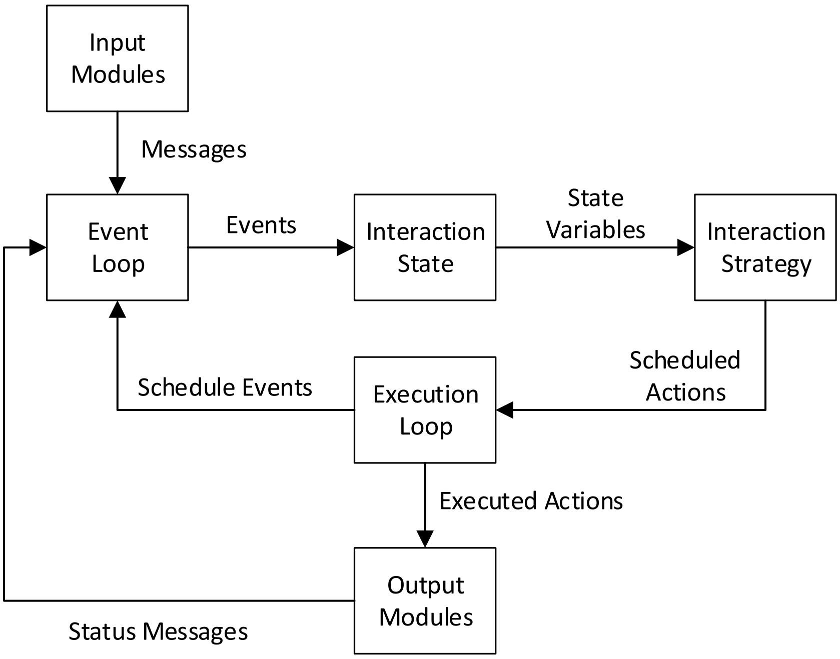


Figure 4.2 – Main components of the AIM.

For communication with external components, AIM supports different middleware implementations which allow the connected components to communicate across operating systems, programming languages and physical machines. The current implementation supports the `one4all` middleware used traditionally for the ARMAR robotic system and the `lcm` (lightweight communication and marshalling) middleware[4]. For speech recognition, AIM em-

[2] http://msdn.microsoft.com/en-us/library/ee125077.aspx

[3] http://www.w3.org/TR/speech-synthesis/

[4] https://code.google.com/p/lcm/

ploys a grammar-based recognizer build with the BioKIT decoder [TWG+14]. Tagged grammars are also used for natural language processing, similar to the approach described in [FHW04]. Customized extensions can be easily attached to AIM by a generic connector module. For example, the system was connected to the ARMAR robotic head [AWA+08] to control its head movement imitating natural communication gestures like nodding, head bopping, etc. in order to generate the impression of an emphatic communication partner. If some input components are not available or should not be used to reduce complexity of an experiment, a Wizard-of-Oz GUI is available, for example to send virtual speech recognition results or simulated workload recognition to the interaction system.

AIM has been employed successfully to implement a number of interaction applications for various scenarios. It was demonstrated successfully in multiple public presentations. In the next sections, we present a number of adaptive cognitive interaction systems which were implemented using AIM. These sections serve several purposes: First, we demonstrate the versatility of the presented AIM. Second, we develop a number of adaptation strategies for simple and complex interaction systems, making use of empirical and computational cognitive models. Third, we perform several user studies to show that those adaptive cognitive interaction systems yield a significant improvement in usability compared to static, non-adaptive interaction systems. As introduced in Section 4.2.1, we expect adaptive cognitive interaction systems to provide usability improvements in three different ways: First, we expect the systems to be more robust towards mistakes by the user and by the system. Second, we expect the systems to be more efficient, i.e. allowing a higher throughput. Third, we expect the system to be more satisfying for the user. To measure robustness, we compare task error rates and recognition accuracy between adaptive and non-adaptive interaction systems. To measure efficiency, we compare task execution time or other task execution costs. To measure satisfaction, we evaluate questionnaires to capture subjective assessment of adaptive and non-adaptive systems. In the following user studies, the presented systems will be evaluated using those metrics.

4.4 Workload-Adaptive Information Presentation System

To investigate whether an end-to-end adaptive cognitive interaction system could actually be beneficial to a user, we conducted a user study evaluating different usability aspects comparing the fully automatic adaptive system to two static baseline systems and to a omniscient oracle system as a gold standard. The main research questions of this section are: 1) Can a workload-adaptive interaction system provide a measurable benefit to the user compared to a non-adaptive interaction system in terms of task success, efficiency and user satisfaction? 2) Is the empirical workload model accurate enough to provide this benefit in an end-to-end system? 3) What is the relationship between accuracy of the empirical workload model accuracy and the achieved benefit?

In this experiment, a user worked with an information presentation system called Robert, represented by a robotic head as shown in Figure 4.3. The user's task was to systematically write down information the system presented him or her via synthesized speech output. During this task, we dynamically manipulated the participant's workload level by enabling and disabling a secondary task. An empirical workload model based on EEG is employed to discriminate low and high workload situations. This information is used by the interaction strategy to switch between different information presentation behaviors of the system. We compare this system with two different non-adaptive baseline systems as well as with a an oracle-based system which acts as a gold standard.

4.4.1 Adaptation Strategy

The strategy of presenting this information to the users is adapted to their brain patterns recognized from the EEG data. Robert has two different *behavior styles* which can be switched seamlessly between two utterances: The Low behavior style is designed for brain patterns which correspond to low mental workload, and the High behavior style is designed for brain patterns corresponding to high workload conditions. Although the style of presentation differs between Low and High, the content of information stays the same:

The Low behavior style focuses on high information throughput, i.e. only short pauses between utterances and between different database entries are made. Whenever possible, multiple information chunks are merged into one utterance and phone numbers are presented in a block-wise fashion. However, as Robert is designed to be a social robot, maximizing efficiency is not the only criterion but will be complemented by politeness. Thus, Robert takes the time to convey information in complete sentences to mimic a polite communication partner.

The High behavior style on the other hand is tuned towards situations in which the user has to divide his cognitive resources between two tasks which he or she executes in parallel. As this multi-tasking may cause memory capacity reduction, split attention, and limited processing capabilities, the High behavior style accommodates the situation by presenting information in an isolated fashion, giving only one attribute at a time and reporting phone numbers as single digits. Furthermore, pauses are extended between utterances and database entries such that the user has more time to deal with the secondary task. Reporting time is conserved by limiting the information to the attribute name and value, thus minimizing utterance duration and omitting politeness.

The motivation for the selected behavior styles bears its origin in theories of information processing. It is known that complexity of processing an utterance and therefore the mental effort required to decode it can be measured in number of tokens, presence of certain grammatically constructs or sentence orderings [CK92]. This is also supported by neuroscientific evidence showing patterns of increased neural activity during the processing of more complex utterances [JCK+96]. As a consequence, simplification of language structure reduces complexity of information processing and therefore reduces workload. Additionally, we use the theory of threaded cognition [ST08] to see that introducing longer pauses in the interaction behavior of the system frees exclusive cognitive resources (e.g. working memory, visual attention, motor control) to allow for the secondary task to be executed. This avoids congestion of tasks at those exclusive resources and again reduces workload.

Robert's Behavior style	Low	High
Pause duration	short (500ms)	long (2000ms)
Number presentation	blockwise	isolated
Items per utterance	multiple	single
Formulations	polite	concise
Example utterances	The name of the next person is Heidi Kundel. Her telephone number is 52-11-66-3.	Heidi Kundel Telephone: 5-2-1-1-6-6-3

Table 4.1 – Low and High behavior styles for information presentation.

The interaction strategy of the information system defines in which fashion switches take place between the two behavior styles over the course of a section. For the experiments described below, we implemented four strategies: AlwaysHigh, AlwaysLow, EegAdaptive, and Oracle.

Presentation strategy	Behavior style
AlwaysLow	Fixed to Low
AlwaysHigh	Fixed to High
EegAdaptive	Derived from EEG
Oracle	Derived from ground truth on secondary task

Table 4.2 – Presentation strategies and corresponding behavior styles of the information system Robert.

The AlwaysHigh and the AlwaysLow strategies define baseline systems which ignore the current state of the user but rather stick to one behavior style. The EegAdaptive strategy uses the recognized brain patterns to select an appropriate behavior (i.e. High when brain patterns corresponding to high mental workload are detected, and Low otherwise). As a gold standard, we also define the Oracle strategy which switches between behavior styles according to the reference information on the secondary task, i.e. instead of relying on potentially noisy information from EEG data, it selects the optimal behavior for each utterance according to the contextual information of whether the secondary task is currently running or not. Behavior switches may occur at any point during information presentation, even within one utterance. Tables 4.1 and 4.2 summarize Robert's presentation strategies and corresponding behavior styles. All strategies were implemented using the AIM interaction manager.

4.4.2 Experimental Setup

We designed a multi-level evaluation study in which participants had to perform two tasks, partly in dual-tasking fashion to induce different levels of mental workload. In the primary task participants were asked to manually fill in a paper form according to spoken instructions given by Robert. Performance criteria are correctness and completeness of the information filed on paper. In the secondary task participants processed a variant of the cognitive Eriksen flanker task [ES79], in which horizontal arrays of five arrows are displayed (e.g. <<><<). Participants were expected to report the orientation of the middle arrow by pressing the corresponding left or right key on the keyboard. Performance criteria are correctness and reporting speed. Apart from the objective performance measures correctness, completeness, and reporting speed, we collected subjective user judgments by a questionnaire. Based on the questions we evaluated how users perceived the interaction quality and efficiency, to what degree users noticed the adaptation of the EegAdaptive and Oracle strategy, and how changes in strategy and behavior style impact the subjective user experience.

Figure 4.3 shows the experimental setup. Robert was present in form of a humanoid robot head [AWA+08] which talked to the participants using text-to-speech synthesis. The participants faced paper forms to be filled in as well as a desktop computer to execute the flanker task.

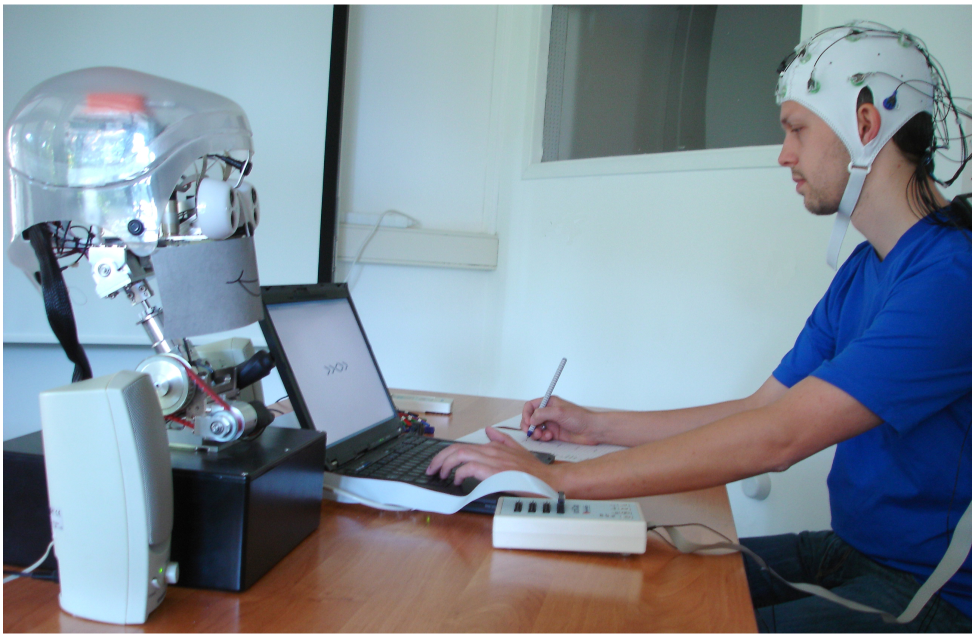


Figure 4.3 – Recording setup with Robert (left side), the Computer for the secondary task (center) and participant wearing an EEG cap (right side).

In total 20 participants entered the experiment and completed five sections which were recorded consecutively in one session. In the first section A, EEG data were recorded to train a person-dependent empirical workload model for each participant. In four subsequent sections B_1 to B_4 we varied the presentation style in which Robert gives instructions to the participant. In each section B_i one of the strategies AlwaysHigh, AlwaysLow, EegAdaptive, and Oracle was applied consistently throughout the section. To eliminate the impact of bias effects such as fatigue, the order of strategies was randomly chosen. Each section consists of a fixed sequence of two alternate segments with and without secondary task. Transitions between segments were marked by an acoustic signal. Each segment lasted approximately one minute. Table 4.3 summarizes the experimental design. Each participant performed five sections of four minutes duration each, resulting in about 20 minutes data per participant, summing up to about 400 minutes data for all 20 participants.

For each Participant	Segment: Single 1 Minute	Segment: Dual 1 Minute	Segment: Single 1 Minute	Segment: Dual 1 Minute
Section A	EEG brain pattern training			
Section B_1	AlwaysHigh	AlwaysHigh	AlwaysHigh	AlwaysHigh
Section B_2	AlwaysLow	AlwaysLow	AlwaysLow	AlwaysLow
Section B_3	EegAdaptive	EegAdaptive	EegAdaptive	EegAdaptive
Section B_4	Oracle	Oracle	Oracle	Oracle

Table 4.3 – Structure of a session and amount of evaluation data. Order of B_i was randomized.

Prior to the main experiment we performed a pilot study on five participants to calibrate task difficulty and duration. The main purpose was to ensure that all test conditions (i.e. all behavior styles with and without secondary task) significantly differ from each other and do not result in overloaded or underchallenged users. The final study was performed on 20 new participants between 21 and 29 years old, who participated voluntarily in the study. All participants are students or employees of the Institute for Anthropomatics at Karlsruhe Institute of Technology (KIT). Each participant signed a consent form prior to the experiments. None of the participants had any prior experience with the EEG-based workload recognition system.

The employed system included an online empirical workload model which was calibrated for the user in a training phase of 4 minutes. To collect training data, each condition of the experiment (i.e. each combination of system behavior and workload condition, see below) is executed by each participant prior to the actual testing phase. This data collection simultaneously served as task training for the participants. The employed workload recognition system is implemented as described in Section 2.3, using only spectral EEG features for classification. Another difference is that the system was trained in a person-dependent fashion, i.e. using training data collected before the execution of the actual experiment.

Table 4.4 lists all questions of the questionnaire the participants answered immediately after each section B_i, so for each participant we collected in total four questionnaires. Each question was assigned to a 6-point scale. The items deal with the adaptation capabilities of the robot (Q1), the appropriateness of its behavior (Q2, Q3), its social competence (Q5, Q6) and an overall

Id	Question Text
Q1	How strongly did the robot adapt to the switch between the conditions with and without secondary task?
Q2	How appropriate was the behavior of the robot in conditions without secondary task?
Q3	How appropriate was the behavior of the robot in conditions with secondary task?
Q4	Would you like to work together with a robot with this behavior?
Q5	How do you judge the behavior of the robot concerning “friendliness”?
Q6	How do you judge the behavior of the robot concerning “empathy”?
Q7	How do you judge the behavior of the robot in general?
Q8	Experienced time pressure*
Q9	Experienced accomplishment*
Q10	Experienced effort*
Q11	Experienced frustration*

Table 4.4 – Questionnaire for subjective evaluation of presentation strategies. Items marked with * are extracted from the Nasa TLX workload scale.

judgment (Q4, Q7). Items Q8 to Q11 were adopted from a subset of the Nasa TLX scale [HS88] to evaluate the experienced workload.

Strategy	Correctness (robot)	Completion (robot)	Correctness (flanker)
AlwaysLow	86%	98%	69%
AlwaysHigh	96%	58%	87%
EegAdaptive	96%	85%	82%
Oracle	94%	85%	86%

Table 4.5 – Completion and correctness rates for the robot instruction and the Eriksen flanker task, averaged over 20 participants.

4.4.3 Evaluation

To analyze the outcome of this experiment, we evaluate (1) the performance of the real-time EEG-based brain pattern recognizer for discrimination between high and low mental workload, (2) the impact of the presentation strategies on the users' task performance, and (3) the users' overall subjective appeal to the end-to-end system.

Regarding workload recognition, we achieve an average accuracy of 83.5% ($\sigma = 6.5$) with accuracy values ranging between 70.8% and 94.0%. The main reason for non-perfect recognition is a delay between changes in task demand and changes in recognized workload. We explain this by effects of the temporal smoothing of workload estimates over time as well as the fact that switches in task demand do not immediately lead to changes in the workload level. We will later see whether this performance is high enough to achieve measurable benefits during interaction. Note that the classification accuracy averaged over all time slices is not a-priori a measure of the end-to-end system's quality in terms of strategy adaptation. This is due to the fact that only a fraction of the recognized workload decisions have an impact on the system's strategy changes. Only the recognition results at the utterance boundaries influence the strategy since within utterances the behavior remains unchanged. Therefore, we performed an experiment in which we limited the classification accuracy calculation to the relevant decision points. This resulted in an average accuracy of 81%, which is reasonably close to the overall recognition accuracy of 83.5%. We therefore conclude that the overall performance of the recognizer is indeed a robust estimator of the system performance in the adaptation task.

Table 4.5 gives the correctness and completion rates of robot and flanker task performance averaged over all 20 participants for the four presentation strategies. The numbers show that the presentation strategy AlwaysLow outperforms all other strategies in terms of completion rate due to the high throughput. In contrast, for the AlwaysHigh strategy participants only manage to complete about half of the items. However, AlwaysLow trades this high completion rate with a dramatically lower correctness rate. Since AlwaysLow leaves only few resources for the participants to properly carry out the secondary flanker task, AlwaysLow is outperformed by the other strategies in terms of flanker correctness rate. In comparison, both adaptive strategies EegAdaptive and Oracle are able to maintain a reasonable completion rate while keeping the correctness rate at the same level as the conservative AlwaysHigh strategy. Furthermore, it can be observed that the fully automatic strategy adaptation applying EegAdaptive compares favorably with the Oracle strategy, indicating that the EEG-based recognition of brain patterns results in a fairly reliable switching behavior. Overall, we conclude that adaptive strategies improve the information presentation by switching behavior styles without hurting task performance.

Item	AlwaysLow	AlwaysHigh	EegAdaptive	Oracle
Q1	2.0	2.5	4.5	5.4
Q2	4.6	4.1	4.9	5.1
Q3	2.3	4.3	3.9	5.1
Q4	2.2	3.3	3.6	4.8
Q5	3.1	3.8	3.7	4.3
Q6	2.2	2.6	3.4	4.4
Q7	2.8	4.0	3.9	4.8
Q8	5.3	3.2	4.0	3.5
Q9	3.0	3.8	3.7	4.0
Q10	5.1	3.5	4.4	4.0
Q11	4.0	2.5	3.0	2.5

Table 4.6 – Agreement score (1 = no agreement, 5 = strong agreement) to the items of the user satisfaction questionnaire for the different strategies.

Table 4.6 summarizes the results of the user satisfaction questionnaire. The result for question Q1 shows that both adaptive strategies (EegAdaptive and Oracle) are indeed perceived as being adaptive. This observation is in accordance with the objective effectiveness of adaptivity measured by the EEG-based brain pattern recognition rate. For appropriateness of behavior, we differentiate between behavior in absence of a secondary task, i.e. single-tasking (Q2) and in presence of a secondary task, i.e. dual-tasking (Q3). For single-tasking, the relative drop from the best to the worst strategy is as small as 24.4% (4.1 for AlwaysHigh to 5.1 for Oracle). For dual-tasking, the participants clearly prefer the High behavior: The gap between the worst and the best ranked strategy increases to 54.9% (2.3 for AlwaysLow to 5.1 for Oracle). We explain this observation by the fact that the benefit of both behavior styles is perceived asymmetrically: While High improves throughput and convenience of the information presentation, Low can make the difference between successful task completion and mental overload. Still, the order of strategies for single-tasking is as expected: AlwaysLow, EegAdaptive and Oracle have very similar scores with non-significant differences while the slow AlwaysHigh strategy is perceived worst. For dual-tasking, the EegAdaptive strategy scores slightly worse than Oracle and AlwaysHigh which perform both optimally in dual-tasking segments (AlwaysLow is indisputably the worst strategy). EegAdaptive usually switches to the correct strategy but with a small delay. As described above, this delay is determined by the window size of temporal integration in the classifier and the fact that a switch of behavior style takes place only between utterances. We assume that a more immediate classification mechanism, a more flexible adaptation scheme and scenarios with longer segments of constant mental workload will mitigate this effect.

The two questions Q4 and Q7 define a metric for overall perceived quality of the system. Both items are strongly correlated ($r = 0.86$). The results reveal a clear quality gap between AlwaysLow and the other strategies. While Oracle outperforms the others by far, the average difference between AlwaysHigh and EegAdaptive is much smaller. This observation is somewhat surprising given the significant differences in objective performance criteria. However, it can be explained by the fact that the EegAdaptive strategy depends solely on the recognition performance of the brain pattern classification. This dependency is expressed in higher standard deviations of most items for EegAdaptive compared to Oracle (which works in a deterministic way). Table 4.7 further investigates this issue. Most of the items are significantly correlated with recognition accuracy. When splitting the data into two groups according to the section's recognition rate (below average vs. above average, denoted *acc.* $\leq \varnothing$ and *acc.* $> \varnothing$), the distance to the scores of Oracle is reduced for the better sections and thus the gap between EegAdaptive and AlwaysHigh increases. In summary, we observe a distinct user preference for EegAdaptive over the non-adaptive strategies given a sufficiently high recognition accuracy. This observation supports our assumption that workload classification performance is a key factor which determines subjective evaluation of system behavior. This means that further improvement of brain pattern classification will directly translate to improvements of user satisfaction.

To further analyze the perception of the four presentation strategies, Q5 and Q6 asked for how friendly and emphatic the behavior was perceived over the section. Q6 reveals that the adaptive strategies (EegAdaptive and Oracle) were indeed perceived as most empathic. Adaptivity and perceived empathy are highly correlated ($r = 0.73$ between Q1 and Q6). This indicates that developing adaptive strategies for interactive systems is an important step towards the implementation of systems which obey the Media Equation (see Section 1.1); the Media Equation implied that users expect empathy from advanced interaction systems with natural input or output modalities. For friendliness, no significant differences between strategies were observed. We ascribe this to the fact that both behavior styles could lead to a perception of friendliness: While High speaks in complete and thus more polite sentences, Low produces minimal phrases which might be perceived as more considerate given the stressful tasks.

Item	d_{low} := O-E and E-O	d_{high} := O-E (*acc.* $\leq \varnothing$)	d_{low}-d_{high} (*acc.* $> \varnothing$)	ρ between acc.
Q1	1.1	0.6	0.5	-0.40
Q2	0.4	0.1	0.3	-0.18
Q3	1.5	0.8	0.7	-0.51*
Q4	1.3	1.1	0.2	-0.35
Q7	1.3	0.6	0.7*	-0.51*
Q5	0.8	0.5	0.3	-0.74*
Q6	1.5	0.4	1.1*	-0.54*
Q8	-0.8	-0.1	-0.7	0.29
Q9	0.3	0.3	0.0	-0.24
Q10	-0.9	0.1	-1.0*	0.46*
Q11	-0.6	-0.3	-0.3	0.24

Table 4.7 – Comparison of user satisfaction items between participants with below-average (*acc.* $\leq \varnothing$) and above-average (*acc.* $> \varnothing$) workload recognition results. The first column shows the average difference between EegAdaptive and Oracle (E-O) for the respective questionnaire item for participants in the *acc.* $\leq \varnothing$ group. The second column contains the same value for the *acc.* $\leq \varnothing$ group. The third column shows the difference between the first two columns. The last column presents the correlation coefficient ρ between workload classification accuracy and E-O calculated for all participants. Scores are mapped such that high values are favorable for the system.

Questions Q8 to Q11 investigate the experienced workload in single- and dual-tasking segments. The dimensions time pressure (Q8), accomplishment (Q9), effort (Q10), and frustration (Q11) show similar patterns: AlwaysHigh expectedly performs best, receiving scores which indicate relatively low workload. Oracle gets very close to those bounds. This shows that an adaptive strategy is able to reach near-optimal workload levels while it flexibly makes the most of cognitive resources whenever available in single-task situations. AlwaysLow is indisputably much worse in all regards compared to adaptive strategies. EegAdaptive approaches the lower workload bound and performs (with exception of Q10) more similar to AlwaysHigh than to AlwaysLow. This indicates that the fully automatic adaptive strategy EegAdaptive is a very reasonable approximation to the Oracle strategy.

4.4.4 Discussion

To conclude, we documented that an end-to-end adaptive cognitive system yields significant, measurable benefits for the user compared to a non-adaptive system. This is the case even for a workload recognition system with an average recognition error rate of 16.5%. Comparing with results from other empirical workload models (see Section 2.2.3) and the results from our own large-scale evaluations (see Section 2.3), we see that this error rate is representative for empirical workload models. Therefore, we conclude that recognition performance which can typically be achieved is good enough to provide benefits to the user. As we showed a connection between workload recognition performance and improvement in subjective system assessment (compare results for *acc.* $\leq \varnothing$ and *acc.* $> \varnothing$ in Table 4.7), we conclude that further improvement of workload recognition will translate to further improvements of interaction quality.

One limitation of the presented study is the fact that the interaction in the presented study was uni-directional, i.e. the user did not give information to the system. Furthermore, while the workload level was switched dynamically, the main task itself was homogeneous and did not require reaction to different stimuli or planning. In the following section, we will address these two limitations.

4.5 Intrusiveness Levels of a Workload-Adaptive Support System

In the previous section, we successfully documented that a workload adaptive interaction system can provide measurable benefits to the user. In that case, the definition of optimal adaptive behavior could be derived directly from psychological fundamentals (see Section 4.4.1); we validated the strategy choice in an informal post-hoc Wizard-of-Oz study, which revealed that human wizards in the role of the information presentation system followed an adaptation strategy which is very similar to the strategy performed by the EegAdaptive strategy. One reason for the straight-forward design of the adaptive strategy was the fact that communication was uni-directional and the task did not involve complex cognition. Consequently, adaptation only transformed the presentation style of the system but did not interfere with the task directly. For more complex HCI applications, the designer of an adaptive strategy has to chose between several possible system behaviors.

We believe that one of the major factors which determines the acceptance of adaptive system behavior is its level of intrusiveness. In Section 4.2.1, we discussed that a comprehensive adaptation does not only involve the "Modification of presentation" (as did Robert in the previous section) but also defines mechanisms to influence task allocation and task scheduling. Adaptive behavior which makes use of such mechanisms provides a system with more options to support its user but is also more intrusive as it interferes with the user's plans and actions. We know from [Jam09] that such interference may lead to a perceived loss of control and loss of predictability. Consequently, a highly intrusive support might be able to provide more substantial help, but risks to be rejected by its users. Understanding the effects of different levels of intrusiveness is therefore crucial to the development of adaptive systems.

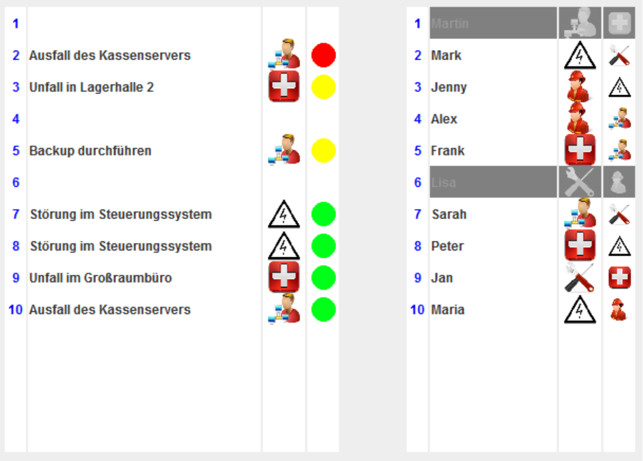
In this section, we describe a user study which we carried out to investigate how a workload-adaptive system can optimally support the participant. We compare adaptation strategies which differ in level of intrusiveness by objective and subjective performance metrics. To our knowledge, this is the first contribution on this topic to the research community and one of the few works on usability evaluation of user state adaptation.

4.5.1 Task Design

In this section, we describe the primary and secondary task which we used in our study. The goal of the presented study was to systematically investigate support strategies with different levels of intrusiveness. We did so for the a scenario which mimics the demands of a complex HCI task: A continuous stream of inputs which require the participant's attention, planning and decision making under time pressure as well as multitasking. Those are typical requirements for professionals like emergency call agents, air traffic controllers or dispatcher.

The participants of the study were told to work as a dispatcher in a factory, allocating workers with different skills to different incoming requests on demand. Requests appeared randomly in a list. Each request had a title (e.g. 'mechanical defect in factory 2') and a skill requirement. Each request had a duration assigned, after which it disappeared ("timeout") and was marked as failed. The duration to timeout was not known to the participants. Each worker had a primary skill and a secondary skill. The participant was instructed to assign workers to requests which matched the required skill with the worker's primary or at least secondary skill. Only if neither was possible, an unskilled worker was to be assigned to not let the request disappear unaddressed. The quality of an assignment depended on the time elapsed from the arrival of the request to the assignment (shorter was better). When a worker was assigned, the corresponding request was removed from the list and the worker became unavailable for a period of time. As the number of workers was limited, sometimes the participant had to decide whether to wait for the return of a worker or to perform a sub-optimal assignment immediately.

A total of 30 requests was presented in one run of this task with lasted 3.5 minutes. Requests were distributed randomly in time. We call this primary task the *Dispatcher Task* (DT). The DT was operated using a graphical user interface (GUI) which showed requests and the roster of available workers. Workers were assigned to requests by keyboard commands. Figure 4.4 shows a screen shot of the DT-GUI. The DT required constant visual attention, quick decision making and planning, as assigning one worker may limit options to handle other tasks satisfactorily. The frequency of new requests maintained a constant workload level over the course of one run of the DT. Still, the random generation of requests lead to a heterogeneous distribution of cognitive demand over time.

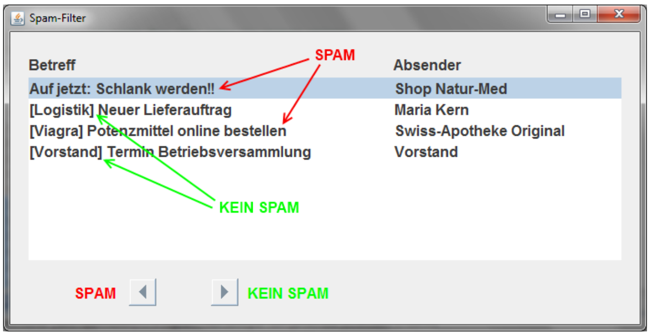


1	
2	Ausfall des Kassenservers
3	Unfall in Lagerhalle 2
4	
5	Backup durchführen
6	
7	Störung im Steuerungssystem
8	Störung im Steuerungssystem
9	Unfall im Großraumbüro
10	Ausfall des Kassenservers

1	Martin
2	Mark
3	Jenny
4	Alex
5	Frank
6	Lisa
7	Sarah
8	Peter
9	Jan
10	Maria

Figure 4.4 – User Interface of the Dispatcher Task with list of requests (with skill requirement and timer) on the left hand side and list of workers (with primary and secondary skill) on the right hand side.

In some configurations of the experiment, the participant had to handle an additional secondary task, in which the participant had to sort out important messages from a continuous stream of e-mails. E-mails which were not dealt with in a certain time window are discarded and scored as failures. This secondary task ran in an additional window on the same screen as the DT and required the participant to constantly divide his or her attention between the two tasks, making well-planned distribution of workers in the DT much more difficult. We call this secondary task the *Mail Task* (MT). We also refer to the configurations without MT as *low workload condition* and the configurations with MT as *high workload condition*. We will justify this naming scheme later in Subsection 4.5.5 by comparing subjective workload ratings.



Betreff	Absender
Auf jetzt: Schlank werden!!	Shop Natur-Med
[Logistik] Neuer Lieferauftrag	Maria Kern
[Viagra] Potenzmittel online bestellen	Swiss-Apotheke Original
[Vorstand] Termin Betriebsversammlung	Vorstand

Figure 4.5 – User Interface of the Mail Task.

4.5.2 Assistant with Different Support Strategies

In this section, we describe the multimodal assistant which we implemented to support the participant in coping with the DT, particularly in the high workload condition. The assistant was able to send notifications to the user using synthesized speech and graphical highlighting in the DT GUI. It was also able to automatically perform assignments in the DT. The assistant implemented several different support strategies. Those support strategies were designed to automatically take over some of the dispatcher decisions, thus reducing the number of required mental operations in a given time. The assistant was implemented using the AIM interaction manager. AIM was extended to receive messages from the DT and MT about events in those tasks (e.g. newly arrived requests) and to send commands for worker assignment.

Note that the assistant was not omniscient concerning request durations and request requirements, i.e. the assistant performed optimal assignments only with a certain probability and assigned worker with secondary skill match or with no skill match otherwise. This simulated the fact that in a typical use case in a professional environment, the user is an expert whose knowledge is difficult to reproduce automatically. The expert knowledge is instead approximated with an error prone heuristic. For the DT, we implemented the following "heuristic": For a given request, this heuristic selected a (random) worker with fitting primary skill with a probability of 70%. If no such worker was selected (or none was available), it selected a (random) worker with fitting secondary skill with a probability of 50%. If no such worker was selected (or none was available), it selected a (random) worker with no fitting skill.

The support strategies mainly differed in their level of intrusiveness. More intrusive strategies are potentially more effective as the participant has to make fewer decisions, but they also might reduce the (subjective and objective) level of control. This is because the participant cannot predict the decisions of the support strategy, and therefore cannot consider these decisions during planning. When the support strategy interferes with the participant's plan (e.g. assigns workers to other requests as designated by the participant) the participant cannot control the task as desired and has to adjust.

The three support strategies with decreasing level of intrusiveness were as follows: The `ACT` strategy intervened as soon as the number of active requests exceeded a threshold. Then, it selected the oldest request and assigned a worker to it (using the heuristic mentioned above). The strategy was constrained such that it could not act while the participant had a pending partial selection (i.e. a worker or a request were selected but the assignment was not submitted yet) to avoid interference with the participant's decisions. Each automatic assignment was reported to the participant via speech synthesis with a statement that names the quality of skill match and the name of the processed request. This feedback was useful as it informed the participant about the intervention by the assistant and the resulting change in the DT state. It also helped the participant to judge if he or she needed to pay more attention to the DT to improve assignment quality. However, ignoring the system statements did not influence the course of action. The `OPT-OUT` strategy allowed the participant to exert more influence on the intervention by the assistant. Instead of directly executing an assignment, the assistant proposed it to the participant by verbalizing it and by simultaneously highlighting it in the graphical interface. Those proposals are guaranteed to not interfere with a partial selection of the participant. While the proposal was pending, the participant was able to operate the interface in the usual way. After five seconds, the proposed assignment was executed (if still valid). The participant had the ability to suppress this execution by pressing a button. In that case, the proposal was discarded and potentially replaced by a new one. The `OPT-IN` strategy reduced the intrusiveness of the system even further. It generated and presented assignment proposals to the participant in the same fashion as `OPT-OUT` did. The main difference was that for `OPT-IN`, the system required a key press to accept and execute the proposed assignment. If not accepted within a certain period of time, the proposal was discarded and potentially replaced by a new one. The 'support strategy' which did nothing, i.e. did not give any support, is referred to as `NONE`.

We used synthesized speech as output modality for all three strategies, as the visual load of both DT and MT was already high and because the participant was able to process verbalized information regardless of his or her focus of visual attention. While the strategies were mainly designed to support the participant in a high workload situation, the system was not directly aware of the presence of a secondary task. If adaptive behavior is desired, we therefore need to provide an empirical workload model.

4.5.3 Empirical Workload Model

To provide adaptive behavior, the system had to recognize the user's workload level. The setup of the empirical workload model mainly followed Section 2.3: Classification was performed on windows of EEG data of 2 s duration with an overlap of 1.5 s. For preprocessing, the influence of ocular artifacts was reduced by performing Independent Component Analysis and automatically removing components containing artifact patterns. Frequency-based features (power of 28 frequency bins from 4 Hz to 45 Hz) were calculated for each channel and then concatenated into a feature vector of 896 dimensions. A binary classifier based on Linear Discriminant Analysis was trained person-dependently to separate low vs. high workload conditions. Temporal smoothing of the results of ten subsequent windows was performed to improve recognition stability.

In contrast to the experiment in Secion 4.4, workload recognition was performed offline. For the usability evaluation of different adaptation strategies, we relied on a workload oracle, as introduced in Section 4.4. The benefit of using a workload oracle is a reduced setup time, which is required to allow the evaluation of different strategies within one session. Furthermore, the inter-participant variation in workload classification performance would confound the results of an adaptation strategy comparison. This effect was documented in the previous Section 4.4, where we showed a relation between workload classification accuracy and benefit of adaptation[5]. The use of a workload oracle allowed us to focus on the comparison of strategies, independently of individual workload classification accuracy.

[5] In Section 4.4, this relation did not negatively impact the results as we only investigated one adaptation strategy. However, when comparing multiple adaptation strategies as in this section, different workload classification accuracies would confound the effect of different adaptation strategies.

A workload oracle does not limit the transfer of our results to real-world applications: In Section 4.4, we showed that a workload classification error rate of 16.5% was low enough to achieve a measurable benefit for the interaction similar to the benefit of a workload oracle. If we can achieve a comparable offline workload classification performance in the present dispatcher scenario, we can conclude that the results for a workload oracle will transfer to system with an online empirical workload model.

4.5.4 Experimental Setup

The experimental setup was as follows: First, the participants were introduced to the DT with written and oral instructions. After that, the participants performed several training runs of the DT to familiarize themselves with the keyboard layout and the task flow. Then, the four strategies were executed in four randomly ordered runs of the DT. Immediately before a run, the corresponding strategy was explained and demonstrated. After each run, the participant filled out a questionnaire on the user experience in that session. Table 4.8 presents the items of the questionnaire. The questionnaire covered several aspects of user satisfaction which we deemed relevant for assessing the quality of a strategy. It included statements on subjective task performance, quality aspects of the system behavior, attribution of success and intrusiveness. The questionnaire used a five point Likert scale. Subjective workload was estimated using the NASA TLX questionnaire [HS88]. After the four runs with DT only, the Mail Task was introduced and demonstrated. The participants performed a training run of the high workload condition (i.e. DT+MT). Again, the order of the four support strategies was determined randomly and the participants performed four runs with subsequent questionnaire. With this setup, we recorded a total of 16 sessions. Participants were all university students or staff members. Participants were paid for their participation in the study. 12 of those sessions were performed with EEG recordings for the analysis of a person-dependent empirical workload model. EEG was recorded using a 32 channel BrainVision actiCap with active electrodes, sampled at 500 Hz and referenced at Pz.

4.5.5 Evaluation

For the evaluation, we look at five research questions: 1) Do the support strategies lead to an improved task performance? 2) Is there a difference in the extend how they do so? 3) How are different support strategies assessed

Item	Text
Q1	I had no problems with handling the task.
Q2	I am content with my performance.
Q3	I pressed keys randomly.
Q4	I was in control of the task.
Q5	The assistant supported me.
Q6	I could work relaxedly.
Q7	I listened carefully to the assistant.
Q8	The assistant helped in a timely fashion.
Q9	I felt patronized by the assistant.
Q10	The assistant distracted me from the task.
Q11	I felt I was not up to the task.
Q12	The assistant allowed accurate task execution.
Q13	The assistant behaved obtrusive.
Q14	The assistant allowed fast task execution.
Q15	I wanted to succeed without support.
Q16	Task success was on me.
Q17	It was pleasant to work with the assistant.
Q18	I had to work against the assistant.
Q19	I would chose to work with the assistant.

Table 4.8 – Items of the user satisfaction questionnaire to evaluate the support strategies of the DT.

subjectively by the participants? 4) Does the benefit of support strategies change with the workload condition? 5) Does the subjective rating of support strategies change with the workload condition? If the answer to 4) and 5) turns out to be "yes", we could make a strong argument for the application of adaptive system behavior, i.e. switching between support strategies, depending on the detected workload level.

Prior to the analysis, we filtered the data by removing *high performer*, i.e. participants who were able to handle the Dispatcher Task with a success rate of 100% even in the presence of the Mail Task while their Mail Task success rate is > 95%. For those participants, no assistance of any type can improve their performance in the DT and this will also influence their subjective assessment of the support strategies. Of the 16 participants, five fit the definition of a high performer. For the analysis of task performance and user satisfaction, we excluded those participants to avoid ceiling effects. From exploratory sessions, we estimated that high performer exhibited similar behavior as the other participants when task difficulty was increased.

Therefore, the results in this section are likely to also apply to high performers in cases in which they actually require support.

Performance Metrics & Workload Assessment

In this section, we define and analyze the metrics which we used to evaluate and compare the different strategies. We employed three different objective performance metrics for the DT and one performance metric for the MT: For both tasks, we measured success rate (SR) as the relative number of items (i.e. requests for the DT, e-mails for the MT) that were handled before they expired. For the DT, we additionally evaluated response time (RT) as the time it took to deal with a request (only regarding requests which were eventually dealt with at all) and assignment quality (AQ) as the average match between assigned worker skill and request requirements (assignment quality of 2 means a primary skill match, assignment quality of 1 is a secondary skill match and assignment quality of 0 is no match).

		DT SR	**DT RT**	**DT AQ**	**MT SR**	**MD**
		[%]	[s]	[Quality]	[%]	[TLX Score]
Low Workload	`NONE`	91	7.18	1.64	-	13.5
	`OPT-IN`	86	7.65	1.52	-	13.9
	`OPT-OUT`	93	7.04	1.63	-	13.9
	`ACT`	99	5.15	1.67	-	12.9
High Workload	`NONE`	79	8.95	1.39	93	15.8
	`OPT-IN`	88	8.35	1.47	92	15.4
	`OPT-OUT`	87	8.30	1.48	95	14.2
	`ACT`	99	5.61	1.67	96	13.8

Table 4.9 – Average performance measures for Dispatcher Task (DT) and Mail Task (DT): Success Rate (SR), Reaction Time (RT), Answer Quality (AQ). Also given is the 'mental demand' dimension (MD) of the TLX questionnaire on a scale from 0 (low subjective workload) to 20 (high subjective workload).

Table 4.9 summarizes averaged performance metrics for all eight runs (4 support strategies in low and high workload condition). We first note that, unsurprisingly, the Mail Task had a strong impact on the performance in the Dispatcher Task: The average success rate of the DT dropped significantly[6]

[6] For all differences reported as 'significant' in this section, this refers to a paired, one-sided t-test with $\alpha = 0.05$. Tests were family-wise error-corrected for multiple testing

by 13.2% relative ($t = 3.23$, $p = 0.004$) from 91% to 79% between the low workload condition and the high workload condition for the NONE strategy; likewise, the average answer quality (AQ) dropped significantly by 15.2% relative ($t = 3.13$, $p = 0.005$) from 1.64 to 1.39 and the average reaction time (RT) rose significantly by 24.7% relative ($t = 2.38$, $p = 0.02$) from 7.18 to 8.95. Furthermore, the 'mental demand' dimension of the TLX questionnaire rose significantly by more than 17% relative ($t = 3.97$, $p = 0.001$) from 13.5 to 15.8. We focus on the 'mental demand' dimension as of all TLX dimensions, it varies the most with workload level and strategy and is the best indicator of mental workload (as opposed to dimensions like 'frustration' or 'physical demand').

We now look at the impact of the different strategies on performance metrics in the low workload condition. There was a maximum improvement in SR of 8.2% relative from NONE to ACT (from 91% to 99%), but also a small decrease of 5.5% relative from NONE to OPT-IN (from 91% to 86%). We see the same pattern for AQ, with ACT being better than NONE and OPT-IN being worse. The OPT-OUT strategy performed similar to NONE for both SR and AQ. Regarding RT, only ACT provided a substantial reduction compared toNONE, by 28.3% relative from 7.18 s to 5.15 s. This is reasonable as ACT is the only strategy where multiple assignments can be processed truly in parallel. Regarding subjective workload assessment, there is a non-significant reduction of 4.4% relative in the 'mental demand' dimension of the TLX from NONE to ACT (13.5 to 12.9). In contrast, the strategies OPT-IN and OPT-OUT increased the mental demand compared to NONE in the low workload condition, as they impose additional decisions on the participant.

In the high workload condition, the gain of employing a support strategy was more substantial than for the low workload condition: SR now improved by up to 24.9% relative for ACT compared to NONE (from 79% to 99%) and overall, all strategies yielded significantly higher SR than NONE ($t = 3.13$, $p = 0.005$ for ACT, $t = 3.24$, $p = 0.004$ for OPTIN and $t = 2.11$, $p = 0.03$ for OPTOUT). While in the low workload condition, there was no notable difference in AQ, in the high workload condition, there was an improvement in AQ of 20.1% relative for ACT compared to NONE ($t = 3.06$, $p = 0.006$), from 1.39 to 1.67. This means than under high workload, it was not only possible for the participants to handle more requests when the ACT, but the performed assignments were also better. This is an interesting result as the assistant is programmed to perform sub-optimally compared to an

using the Bonferroni-Holm method. We report the t-value and the resulting p-value for each test.

expert user. Also for OPT-IN and OPT-OUT, we see small improvements in AQ between 5% and 6% relative compared to NONE. RT is again only influenced positively by the ACT strategy (reduced from 8.95 s to 5.61 s). Overall, we see substantial improvements for the ACT strategy and positive effects for all three strategies. ACT increased all performance metrics to or above the levels of the low workload condition with the NONE strategy. Differences in task performance are also reflected in subjective workload assessment, measured by the TLX questionnaire. The 'mental demand' dimension drops significantly from 15.8 to 13.8 between NONE and ACT in the high workload condition ($t = 2.11$, $p = 0.03$). On the other hand, the OPT-IN strategy shows no significant difference in this dimension compared to NONE (15.4 vs. 15.8). The difference in mental demand between OPT-OUT and NONE barely misses significance ($t = 1.71$, $p = 0.058$ for the difference between 15.8 and 14.2).

In summary, we see that the ACT strategy yields the highest improvement compared to NONE. However, all three strategies were able to improve the DT performance in the high workload condition. The ranking of strategies which we can derive from those results on performance metrics (in high workload) is: ACT > OPT-OUT > OPT-IN > NONE. This ranking corresponds to an ordering of strategies from highest intrusiveness to lowest intrusiveness.

Subjective Evaluation by Factor Analysis

Next, we analyze the answers for the satisfaction questionnaires to evaluate how participants judged the different supporting strategies. Table 4.10 summarizes the results (Refer to Table 4.8 for the content of each questionnaire item). A general summary of the participants' judgment is given by the overall acceptance (Q19) of the strategies. Compared between runs with and without MT, acceptance for all three strategies improved for the high workload condition compared to the low workload condition (agreement to Q19 increases from 3.9 to 4.0 for OPT-IN, from 1.5 to 3.7 for OPT-OUT and from 2.0 to 3.5 for ACT). This can be explained by the fact that participants reported that they were less ambitious to handle the DT completely on their own in the high workload condition compared to the low workload condition (agreement to Q15 decreased by 30.7% relative averaged across all strategies). This result is highly relevant for the application of adaptive user interfaces as it shows that supportive behavior should not be activated all the time to be helpful, but must adapt to the user's workload level. The ranking of the strategies derived from acceptance (Q19): OPT-IN is significantly preferred to

OPT-OUT (4.0 vs. 3.7 for the high workload level; $t = -2.72$, $p = 0.006$) which is preferred to ACT (although the difference is less pronounced and therefore not significant: 3.7 vs. 3.5 for the high workload level; $t = -1.02$, $p = 0.16$). This means the order of preference is reversed compared to the order derived from task performance improvement (see Table 4.9). Such discrepancy between objective performance metrics and subjective user satisfaction is long known in usability research [FHH00].

	Low Workload				**High Workload**			
Item	NONE	OPT-IN	OPT-OUT	ACT	NONE	OPT-IN	OPT-OUT	ACT
Q1	4.2	3.5	3.9	3.5	2.8	3.1	2.9	3.5
Q2	4.1	3.4	3.6	3.6	2.9	3.5	3.7	3.5
Q3	1.4	2.0	1.9	1.5	2.0	2.1	2.4	2.2
Q4	3.8	3.3	3.2	3.2	2.1	2.8	3.0	3.1
Q5	-	3.9	3.2	2.3	-	1.7	4.0	3.4
Q6	-	2.5	2.5	2.2	-	3.0	3.0	2.8
Q7	-	3.4	3.5	1.5	-	3.0	2.8	1.9
Q8	-	3.1	2.8	2.0	-	3.2	3.2	3.1
Q9	-	2.2	3.5	3.9	-	2.1	2.4	3.9
Q10	-	2.9	3.7	3.4	-	2.1	2.5	2.6
Q11	-	2.2	2.5	2.5	-	2.2	2.2	2.5
Q12	-	3.2	2.4	2.0	-	3.4	2.9	3.0
Q13	-	2.5	3.3	3.9	-	1.8	2.5	3.5
Q14	-	3.6	3.3	2.4	-	3.5	3.0	3.1
Q15	-	2.2	3.2	3.8	-	1.9	2.2	2.4
Q16	-	3.8	3.8	3.7	-	3.6	3.6	3.2
Q17	-	3.1	2.8	2.6	-	3.5	3.4	3.1
Q18	-	1.7	3.1	3.6	-	1.5	2.3	2.7
Q19	-	3.9	2.5	2.0	-	4.0	3.7	3.5

Table 4.10 – Results of the satisfaction questionnaire for the different strategies and workload levels. Q5–Q19 do not apply to the NONE strategy. 1 = strong disagreement, 5 = strong agreement.

When taking a more detailed look at the items of the questionnaire in Table 4.8, we see that those covered different aspects of the interaction. To group the items, we performed an explorative factor analysis with Varimax rotation on all items but Q19. We extracted five factors (see Table 4.11), which in total explained more than 70% of the variance in the data. A χ^2 test indicated that five factors are sufficient ($p = 0.04$) to explain the data. We see how the items of the questionnaire are grouped together in semantically meaningful factors. The resulting factors could be interpreted as representing objective benefit of the assistant (F1), its obtrusiveness (F2), the amount of control exerted by the participant (F3), the desired level of independence (F4) and the level of experienced overload (F5). Figure 4.6 presents the questionnaire item scores for the different strategies, averaged across the items loading on the corresponding factors. We see significant differences between strategies for F1, F2 and F3. In contrast, differences between strategies for F4 and F5 were not significant. This result means that participants perceive the benefits (F1) of `ACT` compared to `OPT-OUT` in reversed order compared to the objective criteria (2.39 vs. 2.99; $t = -1.76$, $p = 0.047$). The same phenomenon can be observed when comparing `OPT-OUT` to `OPT-IN`: `OPT-OUT` is perceived as less helpful as `OPT-IN` (2.99 vs. 3.2; $p = 0.09$), which is in contrast to the results of the objective metrics. One reason for this result may be that participants evaluated `ACT` as much more obtrusive (F2) compared to `OPT-IN` (3.18 vs. 2.47; $t = 4.47$, $p = 0.0007$) which was perceived as slightly more obtrusive than `OPT-IN` (however not significantly: 3.18 vs. 3.06; $t = 0.68$, $p = 0.25$). This indicates that perceived obtrusiveness was dominated by whether the assistant performed assignments autonomously (`OPT-OUT`, `ACT`) or not (`OPT-IN`); the ability to suppress automatic assignments is less important. Furthermore, participants also felt that they lost control over the task (F3) from `OPT-IN` to `OPT-OUT` (2.08 vs. 2.83; $t = 3.66$, $p = 0.001$) and from `OPT-OUT` to `ACT` (2.83 vs. 3.59; $t = 3.31$, $p = 0.002$). F4, i.e. the desire for independence from the assistant, on the other hand did not vary with strategy. We explain this by the fact that desire for independence is a stable personality trait and therefore not dependent on the strategy. Experienced overload (F5) also does not change with the strategy. This is in line with the observation that none of the six workload dimensions of the NASA TLX correlated significantly with acceptance ($r \leq 0.19$ for all dimensions).

As the overall acceptance item Q19 was excluded from factor analysis, we can predict this item from the resulting factors. For this purpose, we estimated a linear regression model with Q19 as dependent variable and F1 to F5 as independent variables. Table 4.12 shows the resulting model. The overall model achieved an r^2 of 0.51. Looking at individual factors, F2 and F4 were significant predictors of Q19 ($p = 0.0005$ and $p = 0.04$, respectively) and most strongly influenced the overall acceptance of the system. In contrast, the influence of the objective benefits of the assistant (i.e. F1) is not

Id	Loading Items	$\sum \sigma^2$	Interpretation
F1	Q8 (0.87), Q12 (0.78), Q14 (0.76)	21%	objective benefit of assistant
F2	Q9 (0.88), Q13 (0.93), Q18 (0.61)	36%	obtrusiveness of assistant
F3	Q2 (0.96), Q4 (0.65), Q6 (0.50)	47%	task control
F4	Q10 (0.73), Q15 (0.69), Q16 (0.51), Q18 (0.52)	59%	desired independence
F5	Q1 (0.92), Q3 (-0.60), Q7 (0.56)	70%	overload

Table 4.11 – Result of the factor analysis for user satisfaction questionnaire items. Given are the items which load on each factor, the cumulative explained variance ($\sum \sigma^2$) and our interpretation of each factor.

significant. Of the most influential factors, F2 was much more positive for OPT-IN than for the other two strategies, and was slightly more positive for OPT-OUT compared to ACT, but not significantly. This explains the observed preference pattern reflected by Q19, which behaves analogously. This result means that perceived intrusiveness is indeed a key predictor of agreement, as hypothesized in the introduction. The fact that F4 is also a predictor of acceptance indicates that not only situational workload plays a role for strategy acceptance, but also the user's personality.

Indep. Variable	Estimate	p-value
Intercept	3.83	0.0001*
F1	0.18	0.35
F2	-0.79	0.0005*
F3	-0.08	0.61
F4	0.39	0.04*
F5	0.04	0.76

Table 4.12 – Linear regression model $Q19 = \sum_{i=0}^{5} \beta_i \cdot F_i$ with factors from Table 4.11 as independent variables ($F_0 = 1$ is the intercept) and Q19 (acceptance) as dependent variable. An asterisk indicates a β_i which is significantly different from 0.

Workload Classification

We finally evaluated the classification accuracy of the empirical workload model. We assigned to all data of low workload conditions the class label LOW and assign to all data of high workload condition the class label HIGH. An EEG-based empirical workload model (see Section 2.3) was trained to separate the two classes. For this binary classification problem, we performed

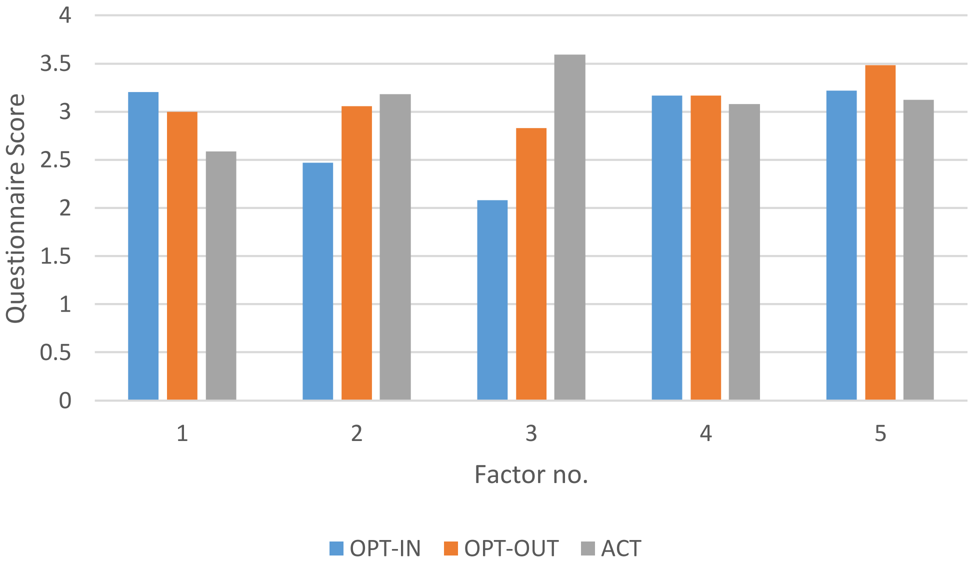


Figure 4.6 – Questionnaire scores for the factors resulting from factor analysis, given for the different support strategies. Questionnaire score measures agreement to the factor: 1 = strong disagreement, 5 = strong agreement.

an offline analysis in a person-dependent 16-fold cross-validation. The average classification accuracy which resulted from cross-validation was 75.8%, with a standard deviation of 16.8%. If we excluded one participant for which technical problems had compromised data quality of some electrodes, classification accuracy improved to 79.3% with a standard deviation of 12.1%. This indicates that the system was able to reliably differentiate between the different workload levels, with similar accuracy as reported in Section 2.3. Given that the training material is very heterogeneous (i.e. data containing different assistant strategies), this is a satisfying result. In Section 4.4, we showed that a classification accuracy in a similar range already allowed adaptive automation with substantial improvements in task performance and user satisfaction compared to non-adaptive systems.

One limitation of the present workload evaluation is the fact that the order of the workload conditions was fixed (in contrast to the order of the support strategies). This could have potentially biased the workload classifier towards learning temporal effects instead of workload differences. However, in Section 2.3, we showed that the employed workload classifier was robust against such ordering effects. As the classification setup in that evaluation was similar to the present one, we are optimistic that this robustness also transfers to the workload classification in this section. Note that this ordering limitation will not substantially influence the behavioral results of the study. In general, participants improve their skills on both DT and MT, i.e. switching the order of workload conditions would only emphasize the benefit of the support strategies which we showed in the evaluation.

4.5.6 Discussion

There are three main points we conclude from this user study: First, supporting behavior is helpful to the user and generally accepted, but only in high workload conditions. This shows the importance of adaptive user interfaces which only assist when required. Second, the level of intrusiveness is a major determinant of how well a specific support strategy is perceived. The objectively most successful support strategy was ranked low compared to a more acceptable, less intrusive alternative. Third, we reconfirmed the robustness of the EEG-based empirical workload model in the employed scenario.

The results give no easy indication on what strategy is optimal in the given scenario. A designer will have to decide whether an additional performance gain is worth the cost of discontented users. This decision will for example be driven by the costs of mistakes during the main task. A reliable workload classifier can help to only activate intrusive support when necessary to reduce the negative effect on user satisfaction of to a minimum. Although the employed main task was presented with a cover story of a dispatching scenario, it was abstract in nature and we expect results to transfer to any task which has the same main properties, i.e. 1) heterogeneous but frequent inflow of requests, 2) distraction by a secondary task and 3) the property that work can be freely and independently distributed between human operator and machine. Examples of such tasks are air traffic control or crisis management, assuming that there are is an algorithm to automatize the processing of requests (see for example [PHM$^+$12] for automation of air traffic control). While automation always comes with the risk of introducing errors compared to the performance of a human expert, a workload-adaptive assistant can limit the activation of automation to situations of high workload, in which the human is not able to handle the task on his or her own.

A limitation of the present study is the fact that workload recognition was performed offline. We argue that previous research indicates that online workload recognition is feasible at levels which are sufficient to provide significant usability improvements. Still, an analysis with an online workload recognizer for selection of supporting behavior would introduce realistic errors. Another limitation is that the strategies we investigated in this work are limited to local decisions, i.e. each request is treated independently. This is helpful as it maximizes the flexibility of the system and allows immediate switching between different behaviors. As the inflow of future tasks cannot be predicted by the system or the user, this flexibility is needed. Still, in slightly different scenarios, long-term variants of the presented strategies (which would store the user's decision on desired support over a certain period of time) could provide a more efficient and less intrusive behavior.

4.6 Self-Correcting Error Aware Interface

Natural input modalities like speech or gesture recognizers have become broadly available. However, while those input techniques are an important step towards intuitive and efficient HCI, there are some important aspects which are still lacking. One major challenge when using machine learning and pattern recognition techniques to interpret the user's input is the substantial risk of errors compared to traditional input devices like keyboards. Reasons are on the one hand limitations of generalizing from a finite set of training samples to data of high intrinsic variability and on the other hand inherent ambiguities of complex, natural input patterns. Recognition errors often lead to an inefficient and unsatisfying interaction. Such system behavior leads to a user state of confusion and is detrimental to the interaction flow.

In this section, we propose the design of an error-aware interface which is able to pro-actively detect erroneous system behavior in reaction to the input of a user. To detect errors, we exploit the fact that a discrepancy between the expected and the observed system behavior results in characteristic brain activity of the user. This brain activity is called Error Potential (ErrP) and appears almost immediately after erroneous feedback is presented following a user action. It can be measured by EEG. In Section 2.5, we already described the development of a person-adapted empirical cognitive model for the user state confusion. In this section, we will use such a mechanism to improve the quality of a gesture-based interface. Different strategies for recovery from error are discussed and evaluated.

In the following, we will introduce a gesture-based selection task and define different recovery strategies which are employed to respond to detected gesture recognition errors. For evaluation, we collect and evaluate a data corpus of 20 participants – called *Gesture-Sim* – who perform the gesture task. We evaluate the performance and associated costs of a large number of recovery strategy variants in a simulation. Afterwards, we perform a user study of 10 participants – called *Recovery-Study* – using three selected recovery strategies to compare user satisfaction between automatic and manual strategies.

To our best knowledge, the proposal of an end-to-end self-correcting gesture interface is completely new to the research community. The same holds for the presented systematic evaluation, which includes a comparison of different strategies and takes into account both objective and subjective performance metrics.

4.6.1 Experimental Setup

To evaluate the potential of adaptive cognitive interaction systems which model the user state confusion, we designed an experiment using a recognizer for pointing gestures. The experimental setup consisted of a gesture-based selection task, a gesture recognizer and the self-correcting interface.

Gesture Task

In this section, we describe the pointing gesture task which we used as scenario to evaluate different error recovery strategies. In the employed task, participants selected and dragged images presented on a large projection screen to a certain spot of a 2x3 matrix, depending on the content of the image (color and vehicle type). Figure 4.7 shows the interface of the task. To classify the six possible pointing gestures, we used data from a wireless sensor wristband equipped with an inertial-measurement-unit (IMU) fixed to the participant's right hand.

Before execution of the actual experiment, each participant trained a predefined movement sequence: From a resting position, the participant moved the arm to point at the bottom left corner (where the image was shown), paused for about a second, moved the arm to the target cell of the matrix in a smooth motion, paused for about a second and returned to the resting position. This schema ensured consistent execution quality across all participants. The task supported the participant in the correct execution by giving acoustic feedback when a pausing position could be ended. After the gesture was completed, the recognition result was displayed as a large overlay showing an abstract pictogram representing color and vehicle type of the recognized class (see Figure 4.8). At this point, an ErrP classifier evaluated

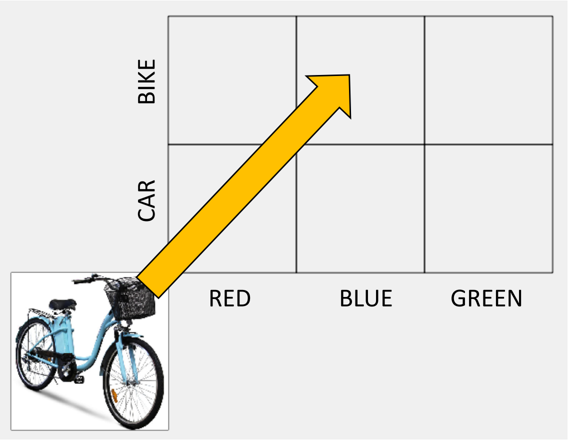


Figure 4.7 – Interface of the gesture task. Users were asked to drag the image appearing in the lower left corner to the corresponding matrix cell, according to its color and vehicle type. The arrow in the figure is for illustrative purposes but was not visible to the participants.

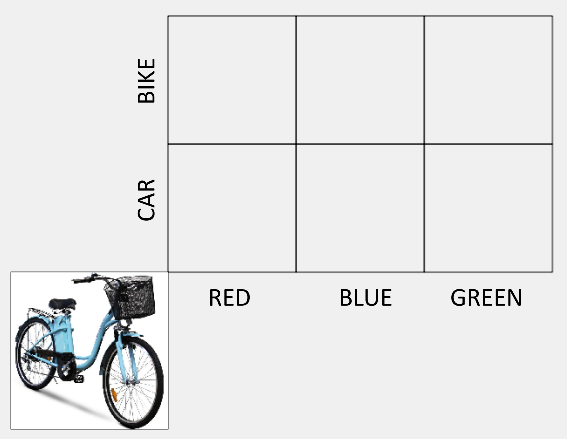


Figure 4.8 – Feedback presentation after a gesture input, indicating type and color corresponding to the recognized class.

the subsequently recorded EEG to recognize whether an error has occurred. Such a stimulus-locking is important as the recognition of ErrPs in EEG depends on temporal patterns in the range of milliseconds.

Gesture Recognition

In this section, we describe the person-independent gesture recognition system which we used to recognize the selected matrix cell from the performed gesture. Arm motion was sensed with a sensor equipped wristband. We used a jNode sensor [SvLG$^+$12], an open research sensor platform which contains a 9 degrees-of-freedom inertial measurement unit (IMU). Sensor data was sampled at 50 Hz and sent wirelessly to a computer. We applied a three-stage processing chain consisting of a segmentation, a feature extraction and a classification stage. It should be noted that the gesture recognizer was deliberately not optimized towards high accuracy. An almost perfect recognizer would not be of use in our scenario, since we investigate recovery strategies from errors. We headed for an accuracy of about 75%.

We employed a two-step segmentation which first identified segments of motion and then separated the actual pointing gesture from other hand movements. In the first step, the motion data was segmented into parts containing motion and parts containing no motion (idle). A segment was detected as motion whenever the angular rate exceeded a given empirically determined threshold. The motion segment ended if the angular rate was below the threshold and remained below it for at least 200 ms. In the second step of segmentation, we modeled the motion sequence with a finite state automaton. Since the movement sequence followed a strict schema, this was a feasible approach for this study. The finite state automaton had four states called UNDEFINED, POINTSTART, GESTURE, and POINTEND. Whenever the segmentation step detected a motion/idle change, we checked for a state transition. The start state was UNDEFINED, which captured all motions that occurred between two gestures. The POINTSTART state corresponded to the initial pointing on the picture at the bottom left corner of the display. The transition into the state POINTSTART was performed if the acceleration in the axis perpendicular to the back of the hand was within a range of $0.98\,\mathrm{m/s^2}$ of an experimentally determined reference value. This means, the orientation of the hand in 3D space was compared to a reference orientation based on the measured earth acceleration. The reference orientation depended on the height and distance of the projected image relative to the user and was therefore dependent on the local environment. The next motion segment triggered the transition into the state GESTURE, which indicated the execution of the actual gesture. The next idle segment triggered transition into the state POINTEND, indicating the pause at the target position. The next motion segment lead to the transition back into UNDEFINED. The motion segment corresponding to the state GESTURE was used for the actual classification of the pointing gesture. The employed six classes corresponded to the six matrix cells. Figure 4.9 summarizes the state transitions of the second step of segmentation.

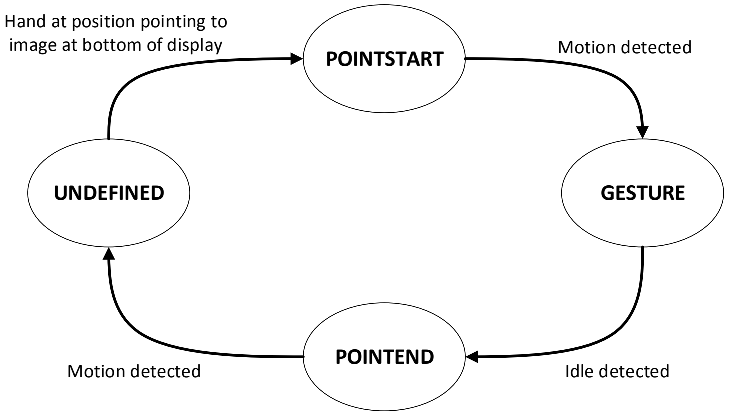


Figure 4.9 – Finite state automaton for the second step of segmentation of the gesture recognizer.

For the classification stage, we used a Gaussian mixture model (GMM) with five Gaussians per class for maximum likelihood classification. Features were computed on the complete segment associated with the GESTURE state in the finite state automaton. Preprocessing consisted of a mean subtraction to compensate constant offsets introduced by gravity (mean calculated on previous trials) and signal smoothing with a running average filter of order five. Due to the drift and noise present in inertial sensor readings, the technique to reconstruct the actual trajectory performed in 3D space is error prone. As a result, we could not simply compute the direction and length of the performed gesture reliably. Instead, we computed the angle of motion in each of the three axes, the L_2-norm of these three angles and the duration of the gesture segment. The angles were computed by integrating the angular rate measurements from the gyroscope over the whole gesture segment. The resulting feature space therefore had five dimensions. The classifier was evaluated in cross-validation, yielding a person-independent recognition accuracy of 77% (on the *Gesture-Sim* corpus). Therefore the gesture recognizer met the criteria for our experiments. We also computed a confidence estimate for the gesture classification: Scores of the GMMs for each feature vector were normalized to a probability distribution across classes. The distance between the normalized score of the highest and second-highest scoring class was used as a confidence estimate of the classification result.

4.6.2 Self-Correcting Interface

For the described scenario, the self-correcting interface was designed as follows: The gesture classifier received input from the IMU and outputs a probability distribution for six classes corresponding to the six possible matrix cells. The most likely class was used to present feedback on the recognized class to the user. The EEG data following this feedback was analyzed; if an ErrP was detected in this data, a recovery behavior was started to correct the error. The exact nature of this recovery behavior depended on the implementation of the recovery strategy. Different strategies may have different characteristics in terms of recovery accuracy, recovery costs and other factors.

In Section 2.5, we already showed than an EEG-based empirical confusion model can reliably detect ErrPs caused by erroneous feedback during the gesture task. In the user experiments comparing different recovery strategies however, the ErrP component was replaced with a simulated ErrP classifier which received the ground truth gesture class and the output of the gesture classifier to detect ErrPs with a recall and precision of 0.8, respectively. Simulating the ErrP classification allowed us a better control over the distribution of errors and therefore a better comparability between sessions. Furthermore, it reduced the setup time, and thereby allowed us to record data from a larger number of participants. The results from the isolated analysis of the ErrP classifier justify this simplification of the experimental setup as they showed that we are able to achieve this performance from real EEG data. It should be noted that the described ErrP classifier (see Section 2.5) can also be operated in online mode, therefore the results from the experiment can be generalized to an end-to-end system with EEG-based classifier.

Next, we define the different analyzed recovery strategies. All recovery strategies were implemented using the AIM framework, which received n-best lists as input from the gesture recognizer and sent commands to the graphical user interface to determine the displayed feedback and the task flow. The most basic strategy was the REPROMPT strategy. REPROMPT reacted to a detected error by prompting the user to repeat the input. The initial gesture data was discarded and the second gesture was used as final classification result. We did not repeat the correction procedure after the first repetition as frequent handling of the same error might lead to unexpected EEG signal patterns. The 2ND-BEST strategy was a modification of REPROMPT that does not always reprompt if an error was detected. Instead, it inspected

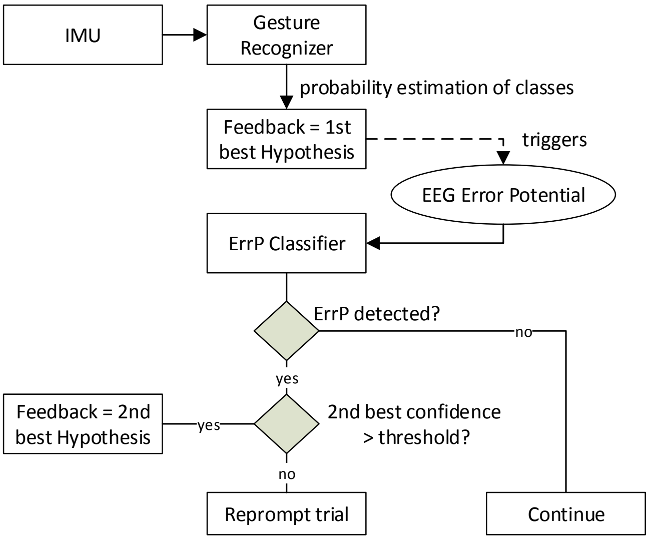

IMU
Gesture Recognizer
probability estimation of classes
Feedback = 1st best Hypothesis
triggers
EEG Error Potential
ErrP Classifier
ErrP detected?
no
yes
Feedback = 2nd best Hypothesis
yes
2nd best confidence > threshold?
no
Reprompt trial
Continue

Figure 4.10 – Flowchart of the 2ND-BEST strategy.

the probability distribution of the all classes and picked the second best class. However, this estimate might be unreliable as it was based on a probability distribution which had just been indicated as erroneous by the detection of an ErrP. Therefore, we only used the second best class if its re-normalized confidence (i.e. probability mass of the first best class distributed equally across all remaining classes) was above a certain threshold t. Otherwise, the user was asked to repeat the input once. Figure 4.10 shows the control flow of both correction strategies (REPROMPT is a special case of 2ND-BEST with threshold $t = \infty$).

As we also want to compare the automatic correction strategies with user-triggered correction, we defined the MANUAL strategy which required the user to actively report a recognition error. This could for example be executed with a “manual override” button on the wristband, which we simulated in our experiments by a two-handed keyboard command issued by the user. When triggered, the system performed a reprompt of the last trial. This strategy had the advantage of near-perfect recognition of error events but did impose additional overhead on the user. For comparison, we defined the strategy NONE, which did not provide any mechanism to recover from detected error situations.

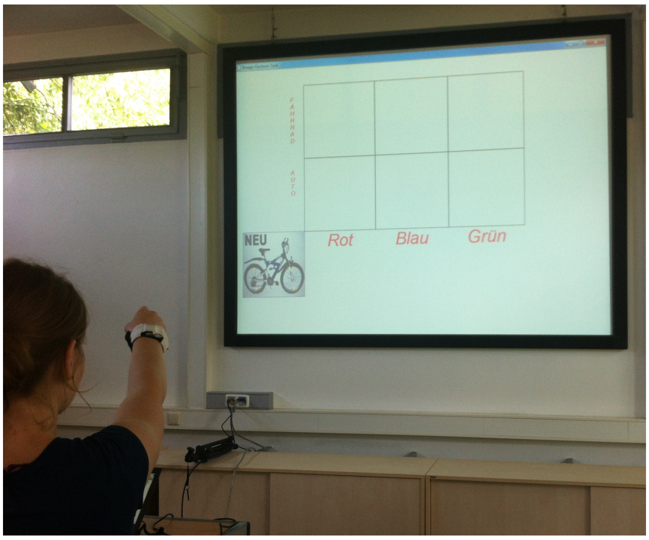


Figure 4.11 – Execution of the gesture task.

4.6.3 Results

Simulation-based Evaluation of Recovery Strategies

Using the setup described in Section 4.6.1, we recorded a total of 20 sessions from 20 participants. All participants were university students or employees. During the experiments, participants first performed a number of training trials and then three blocks with 35 trials each. Between two blocks was a pause of several minutes for the participant to rest. This is the *Gesture-Sim* corpus which we used to evaluate the baseline gesture classifier and to simulate the effects of different recovery strategies.

First, we have a closer look at the gesture classification performance. The average confidence values yielded by the gesture classifier show a difference of 0.84 vs. 0.64 for the correct and the incorrect results, respectively. This difference is barely significant with an average p of 0.079 (one-sided t-test on each fold). This result indicates that the confidence value alone was not reliable enough to detect errors. An additional knowledge source, like the proposed EEG-based ErrP detection, is necessary to reliably identify errors. For all misclassified trials, Figure 4.12 shows a histogram of the rank of the correct gesture class within the n-best list. We see that most of the trials were concentrated at the lower ranks. For 52.2% of all misclassified trials, the second best estimate was the correct one. This indicates that if the 2ND-

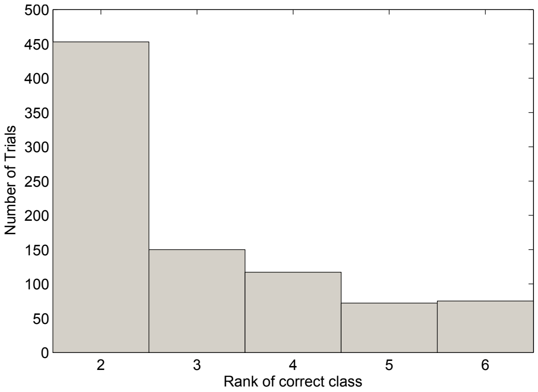


Figure 4.12 – Histogramm of ranks of the correct gesture class within the n-best list (only for misclassified trials).

BEST strategy is pursued, one can expect to reduce the error rate by about 50%.

In the following, we evaluate the different recovery strategies for their impact on recognition accuracy of the gesture recognizer and the associated costs in form of additional gestures. We did this in simulation, where we used the actual results from the gesture recognizer for each trial but simulated ErrP classification and recovery strategy, including all additional gesture classifications during recovery. This method gave us the opportunity to evaluate a large number of recovery strategies with different parameters to study and compare their effects.

To quantify the effect of the different recovery strategies, we defined the *corrected recognition accuracy* which is the fraction of correctly identified gestures after applying the effects of the recovery behavior of the system. Corrected recognition accuracy takes into account the error of the gesture recognizer as well as both types of errors of the ErrP classifier (false positives and false negatives). Corrected recognition accuracy $\hat{a}$ is calculated as follows:

$$\begin{aligned}\hat{a} =& a \cdot (1 - p_{cor}) + a \cdot p_{cor} \cdot a_{TP} \\ &+ (1 - a) \cdot p_{incor} \cdot a_{FP}\end{aligned} \tag{4.1}$$

where p_{cor} is the probability of detecting an ErrP if the gesture input was classified correctly (i.e. the false alarm rate of the ErrP classifier) and p_{incorr} is the probability of detecting an ErrP if the gesture input was classified incorrectly (i.e. the precision of the ErrP classifier). a is the raw accuracy of the gesture recognizer and can be estimated during crossvalidation. a_{TP} and a_{FP} are the probabilities of a successful recovery when an ErrP was identified correctly (i.e. true positive) or incorrectly (i.e. false positive) for the initial gesture input. For REPROMPT, we simply have $a = a_{TP} = a_{FP}$, as every detected ErrP leads to an additional unmodified gesture trial. For the pure 2ND-BEST strategy (i.e. $t = 0$), we have $a_{TP} = p_{2ND}$ and $a_{FP} = 0$, where p_{2ND} is the probability that the second best estimate is correct, given the first one was already excluded. p_{2ND} can be estimated from the histogram in Figure 4.12 as the fraction of second best results of all wrongly classified gesture trials (52.2% in our case). For 2ND-BEST with $\boldsymbol{t} > 0$, we have:

$$\begin{aligned} a_{TP} &= P(c_{2ND} \leq \boldsymbol{t}) \cdot a + P(c_{2ND} > \boldsymbol{t}) \cdot p_{2ND} \\ a_{FP} &= P(c_{2ND} \leq \boldsymbol{t}) \cdot a \end{aligned} \tag{4.2}$$

In Equation 4.2, c_{2ND} is the renormalized confidence of the second best recognition result. Again, the necessary probabilities can be estimated from the histogram in Figure 4.12 by counting only the trials with high confidence.

We also assess the costs of each correction strategy to measure system efficiency. As metric, we calculate the costs of a correction as the average number of additional user inputs (gestures or key presses) necessary to perform the correction. For REPROMPT, this amounts to one gesture for each detected error, including false alarms. Those costs are reduced for 2ND-BEST which in favorable cases corrects errors without any additional costs for the user. For MANUAL, we have no false alarms but additional costs for triggering the correction. We favorably assume that the signal to trigger manual correction is always issued correctly by the user. Correction costs can be estimated in cross-validation by counting the number of (automatically or manually triggered) reprompts.

To assess the performance of the different strategies using those metrics, we perform a block-wise leave-one-out cross-validation of the gesture recognition system and calculate a number of statistics on errors and confidence values. The cross-validation simulates online recognition, i.e. normalization parameters for each trial are only calculated from the data previous to this trial in chronological order. For each fold, we evaluate the gesture classifier on the testing block and use equations 4.1 and 4.2 to estimate corrected recognition accuracy for the different correction strategies.

Table 4.13 summarizes the results, averaged across all folds. We see that for all correction strategies, the corrected accuracy exceeds the baseline accuracy (i.e., NONE) and therefore improves the overall performance of the gesture classifier. As expected, MANUAL yields the highest improvement, followed by REPROMPT. Still, 2ND-BEST ranks only 3.1-9.9% worse than REPROMPT (depending on the selected threshold, see below) and provides a statistically significant improvement over NONE (block-wise one-sided paired t-test, $t = 4.38$, $p = 0.002$). When we rank the strategies by their correction costs, we get the reverse result compared to the ranking based on corrected accuracy: The MANUAL strategy imposes correction costs of 0.46 on the participant, i.e. the participant has to execute 1.5 commands on average per desired input. This is reduced to 0.34 (a reduction by more than 26% relative) when using the REPROMPT strategy, which can detect automatically most of the situations which require a correction. The costs are further reduced to 0.19 (a reduction by more than 58% compared to MANUAL) for the 2ND-BEST strategy (with $t = 0.7$), which in many cases requires no additional user input for correction and never incurs higher costs than REPROMPT.

A caveat for the 2ND-BEST strategy is that while it fixes a number of erroneously classified gestures, it also reacts to a number of false alarms with no chance of recovery (in contrast to REPROMPT which can still generate a valid final result in case of a false alarm). To some degree, this is mitigated by the confidence threshold t applied to the second best result: With $t = 0$, the corrected accuracy of 2ND-BEST is 71%, i.e. below the raw accuracy of NONE. Using a confidence threshold of 0.7, 53% of all error trials are selected for suggesting the second best result. This yields a tuning parameter with which the designer can select between different trade-offs between accuracy and correction costs: Table 4.13 lists results for three different parameter settings to demonstrate this.

Strategy	Corr. Accuracy	Corr. Costs
NONE	77.0%	0
MANUAL	93.0%	0.46
REPROMPT	86.8%	0.34
ROW-COL-PEPROMPT	76.4%	0.34
SELECTIVE-PEPROMPT	88.1%	0.34
2ND-BEST (t=0.5)	78.2%	0.14
2ND-BEST (t=0.7)	81.4%	0.19
2ND-BEST (t=0.9)	84.1%	0.27

Table 4.13 – Performance measures of the different error correction strategies.

It is surprisingly difficult to beat the simple 2ND-BEST strategy (and its special case REPROMPT) in terms of corrected accuracy. We explored a number of other strategies which use allegedly clever mechanisms to improve the recovery process. For example, the ROW-COL-REPROMPT strategy tries to estimate the correct row or column of the image matrix (by identifying the row or column with the highest cumulative confidence after removing the first best result) from the initial gesture and only reprompts this reduced set. Indeed, accuracy of the gesture recognizer rises to 88% when only the one missing dimension (i.e. row or column) has to be estimated from the reprompt. However, errors in the automatic selection of the correct row or column, which inevitably prevent a successful correction, lead to a non-competitive corrected accuracy of 76.4%, which is worse than NONE. An alternative which performs better is SELECTIVE-REPROMPT. It also limits the number of reprompted items but selects the three best classes from the initial gesture input, including the one marked as incorrect by the ErrP detector. This leads to an accuracy of 88.1%, reducing the number of errors of 9.8% relative compared to REPROMPT. However, we pay for this benefit by the fact that we are required to re-arrange the matrix for the reprompt (e.g. moving the three candidates to a joint row) to actually benefit from a simplified pointing and classification task. Informal user tests showed that this is highly confusing to users.

As false alarms and missed errors are not symmetrical in their influence on the performance of a recovery strategy, we also need to look at the impact of different precision/recall values of the ErrP classifier. For example, it may be favorable to tune the classifier towards a higher precision at the cost of reduced recall, to avoid false alarms which might invalidate correct gesture inputs. In Section 2.5.6, we showed that it is possible to achieve a precision of 0.96 by modifying the class balance of training data. Simultaneously, this step reduces recall to 0.76. While this results in a slightly lower F-Score of 0.84 compared to the optimal system, its the false positives which cause the most trouble for automatic recovery; this is especially true for 2ND-BEST, which cannot recover successfully in such situations. As a consequence, corrected recognition accuracy improves by 3.5% relative for REPROMPT and 6.3% relative for 2ND-BEST when adjusting precision and recall of the ErrP detector by +0.1 and −0.1, respectively. The overall ranking of recovery strategies stays the same, although the gap to the MANUAL strategy is reduced, as it does not benefit from the adjustment.

User Satisfaction of Recovery Strategies

The results up to this point indicate that is is possible to develop a self-correcting interface that significantly improves the accuracy of the employed gesture recognizer. However, usability of such a system does not only depend on efficiency and effectiveness and we still have to investigate whether users will accept the different correction strategies. From a general perspective, [Jam09] systematically analyzed a number of undesired side-effects of adaptive interfaces. The author discussed the cause of such side-effects and ways to remedy them. For our application, the most relevant side-effects are "Inadequate Control over Interaction Style" and "Inadequate Predictability and Comprehensibility", as the user has no control over when a correction is triggered and has no way to predict when an error is detected, as well as "Imperfect System Performance" (of the ErrP classifier), which may lead to negative user experience when a correct interpretation is discarded, are the main challenges specific to the presented system. To investigate whether those challenges affect user satisfaction, we recorded the *Recovery-Study* corpus, a user study in which we compared REPROMPT and 2ND-BEST (with $t = 0.7$) to the MANUAL correction strategy. Of all evaluated strategies, those three are the most basic and therefore allow a principal comparison of recovery strategies with different levels of autonomy. After a training phase to accommodate with the gesture recognizer, each participant performed 35 trials for each of the three strategies in random order and filled a usability questionnaire. Averaged across all participants, raw recognition accuracy was 76.2%. Table 4.14 describes the actual performance of the different correction strategies. Compared to the simulation results, we can conclude that in simulation we made satisfactory predictions on accuracy and correction costs. Table 4.15 summarizes the items of the questionnaire and the results. Items were presented with a 5-point Likert-scale, with 1 indicating no agreement and 5 indicating the highest possible agreement. In the following, we analyze the corresponding questionnaire responses. Given the limited sample size, not all results are significant, but the tendencies give a good impression on the perception of the different strategies.

We see that users were not daunted by the self-correcting interfaces. By tendency, the self-correcting systems were evaluated more positively compared to the system with manual correction regarding all presented questionnaire items. However, there was a difference between REPROMPT and 2ND-BEST in which items they more distinctively differed from MANUAL. Due to the reduced number of manual interventions necessary, automatic correction by the 2ND-BEST strategy was perceived as less strenuous, less te-

Strategy	**Corr. Accuracy**	**Corr. Costs**
MANUAL	91.7% (93.0%)	0.52 (0.46)
REPROMPT	84.0% (86.8%)	0.37 (0.34)
2ND-BEST (t=0.7)	79.7% (81.4%)	0.18 (0.19)

Table 4.14 – Performance measures of the different error correction strategies for the *Recovery-Study* corpus. For reference, we repeat the simulation results in parantheses.

Questionnaire Item	REPROMPT	2ND-BEST	MANUAL
felt supported	3.6	3	3.1
system reacts proactively	4.4*	3.5*	2.1
errors corrected reliably	3.1	3.2	2.6
system predictable	3.4	3.1	2.5
system intuitive	4.6*	4.3	4.1
user has control	4.0*	3.4	3.3
felt observed	1.3	1.8	1.8
pleasant experience	3.3	3.7	3.0
system strenuous	2.9	2.4	2.8
correction tedious	3.0	2.4*	3.5
system impedes user	2.5	2.4	2.9
system confusing	2.2*	2.9	3.0

Table 4.15 – Subjective evaluation of correction strategies (1 = no agreement, 5 = high agreement). An asterisk denotes a significant difference (one-sided, paired t-test, $\alpha = 0.05$) between MANUAL and the respective automatic strategy.

dious and more pleasant than manual correction. Comparing both automatic strategies regarding ease-of-use, users preferred the REPROMPT strategy. REPROMPT was evaluated as the least confusing, most predictable, most intuitive strategy. For REPROMPT, there also was a stronger perception of pro-active behavior compared to both other strategies. The reason we see for this difference between both self-correction strategies is that 2ND-BEST provided the more complex user interface behavior as it added new elements to the interaction flow, while REPROMPT only repeated elements which were already familiar to the user. Both behaviors also differed in their handling of false alarms. While reprompting a user after presented feedback can be interpreted as a confirmation of an unreliable classification result, the 2ND-BEST behavior explicitly discarded the initial classification result and gave the user no opportunity to counter this behavior.

4.6.4 Conclusion

In this section, we showed that it is possible to improve accuracy of a gesture recognizer using ErrP classification to enable pro-active recovery from recognition errors. We discussed all components necessary for an end-to-end error-aware interface and evaluated different recovery strategies, looking at both objective and subjective evaluation metrics.

To our best knowledge, the presented study is the first in the research community which analyzes how a detection of the user state confusion can be used to recover from recognition errors. We provide an extensive evaluation in simulation to analyze the effect of recovery strategies on both recognition accuracy and recovery costs. Furthermore, we conduct a user study on the subjective assessment of recovery strategies. This study also confirms the simulation results in live human-computer interaction.

The presented recovery strategies are very general in nature. Reprompting for additional user input or re-interpretation of the original input are both strategies which apply to any noisy input modality. Also, the simulation-based evaluation of recovery strategies can be easily extended to other strategies and be applied to other applications. When transferring the results to another domain, one may have to adjust the employed cost model as requesting additional user input may be more or less costly (or measured on a completely different scale, e.g. time) than in this section. This may also change the user preference for different recovery strategies. Another influencing factor is the baseline performance of the input classifier. A very high baseline accuracy also demands a high ErrP classification precision. Otherwise, nearly all detected errors will be false alarms.

One limitation of the described approach is that it is relying on time-locked evaluation of EEG data relative to feedback presentation. This works well for situations were the system can give instantaneous feedback in an unambiguous way, for example global feedback on a graphical user interface. This becomes more challenging when feedback is given more locally, is not immediately obviously erroneous or is spreading across longer periods of time. In such cases, the ErrP classification has to rely on additional information sources (e.g. eye tracking to track when a presented feedback was perceived) or become less relying on temporal alignment (e.g. by using methods from [MRM13]).

4.7 Cognitive Interaction Simulation

In Chapter 3.3, we introduced a memory model for application in interaction systems. In this section, we show how employing this memory model helps us to simulate plausible user behavior and to predict relevant system utterances in a complex interaction scenario.

Cognitive user simulation is a paradigm of testing system prototypes for their usability by having cognitive models interact with the software to predict task performance and efficiency. However, existing approaches (see Section 3.2.2) are restricted (1) to implement the whole cognitive model within a given cognitive architecture (which imposes severe restrictions on the software architecture of the interaction system and the user model), (2) to traditional graphical user interfaces, and (3) to a small number of fixed behavioral strategies of the simulated user. In contrast to computational cognitive models, there exist statistical user simulation approaches (see Section 4.2.3) for training and evaluation of spoken dialog systems. Those allow a greater variability in behavior, but have no representation of cognitive processes. To our best knowledge, the approach of combining statistical and computational modeling for user simulation is new to the research community.

In this section, we propose to combine both approaches by integrating computational cognitive modeling aspects with a statistically trained user simulation. This combination brings together the benefits of both approaches: It uses State-of-the-Art techniques of flexible statistical user simulation which is employed in a complex application domain. The user model of this simulation is extended by a computational cognitive model to provide a plausible prediction of human behavior and performance, for example with regard to memory or workload level. With this approach, we can exploit validated computational models without being restricted to a specific cognitive architecture. This approach is in the same spirit as the development of a stand-alone memory model for interaction systems in Section 3.3.

The application context in this section is the development of an interactive multimodal tourguide system in the car. The tourguide acts as a navigation system that also provides information to the driver on Points of Interest (POI) along a route through the fictional city of Los Santos. During one episode of the tourguide scenario, traffic junctions and POIs occur at different points in time, while system and user can ask questions or exchange information. The utterances of systems and users as well as external events (e.g. a POI occurring) result in a dynamically changing context for the interaction, influencing the appropriateness of the information which the system can provide. Other factors also determine the interaction context: POIs belong to different categories (e.g. restaurants, museums, etc.) and are of different relevance for the driver. Furthermore, the driver experiences different workload levels caused by variable road conditions and external events. The workload level influences the user's behavior and performance. The tourguide system tries to provide appropriate information to the driver in appropriate complexity using appropriate modalities.

The tourguide scenario has two main challenges: First, as mentioned above, the driving task and the occurring distractions result in a variable workload level of the user. The system has to predict the impact of the workload level on the driver's cognitive abilities, for example when processing information given by the system. Second, the system has to deal with an ever-changing context in the dynamic environment. Therefore, we need to integrate components in our interaction system that are able to explicitly model, predict, and cope with the imperfect user as well as the varying attentional focus and memory content to ensure a seamless and successful interaction experience.

Especially in interaction scenarios which are not directly task-driven, system utterance selection is not trivial: While we follow a clearly defined goal of providing as much interesting information as possible, the system has no clear order or priority of information chunks to present. The same is true if we want to simulate a user for evaluation or automatic strategy learning. To create coherent user behavior, we need to provide a dynamic, workload-adaptive memory model, as seen in Sections 3.3 and 3.4. This section describes how we employ a memory model for utterance selection for both the system and the simulated user. The primary goal of the utterance selection is to find an utterance that is most relevant in the current context and of most interest to the user.

4.7.1 Cognitive User Model

The basis for the cognitive user simulation is a cognitive user model. This model provides predictions on the cognitive state of the users, for example their memory or their workload level. This user model can be applied in two different ways: First, it can be used in a generative fashion within the user simulation component to create plausible user behavior which is consistently driven by the simulated cognitive state. This state is updated by system utterances and external events. Second, the cognitive user model can also be used by the interaction system in a predictive fashion. Using this perspective, the system tries to trace the cognitive state using the model to generate the most usable system utterances. Figure 4.13 (left) illustrates both applications of the user model and their interplay in the simulation of interactions between user and system.

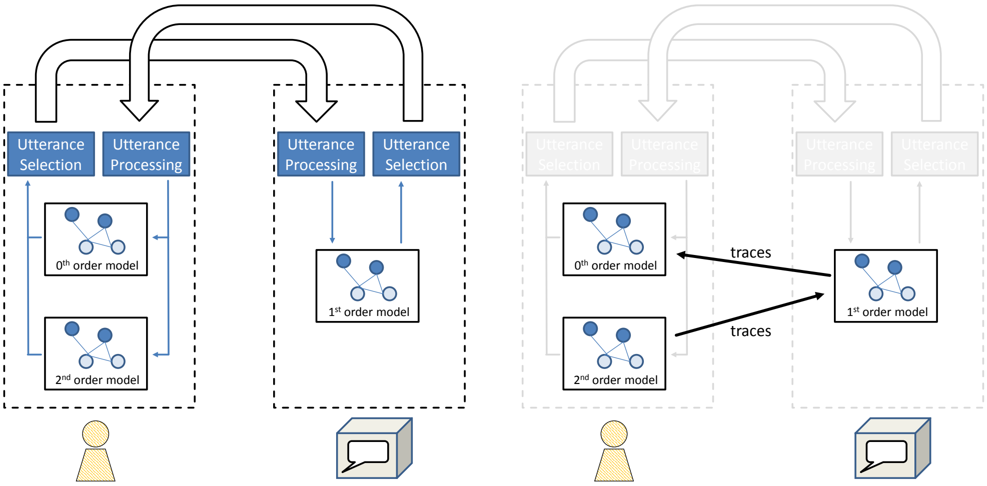


Figure 4.13 – The implemented simulation framework with different user models for system and simulated user (left). Hierarchy of user models which trace the memory state of lower order models (right).

For simulating realistic interactions between user and system, it is not sufficient to provide one global user model which is accessed by both simulated user and system. One important characteristic of spoken interaction is the process of grounding [Bre98], i.e. the process of establishing a joint understanding of the topic and discourse. In the context of cognitive user models, this involves the process of aligning the user's memory with the memory model of the system. Transferring this process to the application of cognitive user simulation – where the simulated user is also represented by a user model – mandates that the system does not have access to the memory model of the simulated user. Instead, it is required to indirectly trace the state of the model using information from the discourse. The system issues grounding utterances to reduce the information mismatch between system and user, by providing information about POIs. The user also has the aim of grounding the interaction by providing information on his or her memory state which they believe the system is not yet aware of (i.e. the user indirectly traces the memory model of the system, which again indirectly traces the user's memory).

Therefore, a complete interaction simulation setup with an autonomous system and a simulated user contains multiple user models. Those user models represent the user's cognitive state at different levels of indirection. We call the level of indirection of a user model its *order*. In our simulation setup, there are three user models, as summarized by the right side of Figure 4.13: The 0^{th}-order model represents the actual cognitive state and is used for user simulation. It receives input from the environment and from system utterances as perceived by the user (i.e. they pass through an error model). The 1^{st}-order model is maintained by the system during interaction and represents the system's view on the user. It receives input from the environment and from perceived user utterances. Finally, there is the 2^{nd}-order model which is also under control of the user and which represents the user's understanding of the 1^{st}-order model and receives input from the environment and from the actual user utterances.

In the following, we describe the main components of the cognitive user model: (1) the memory model to predict which memory items are activated in a dynamically changing context, (2) the workload model to predict the influence of the workload level on the memory model, (3) a strategy optimization module to learn system and user behavior with Reinforcement Learning, and (4) the utterance generation module to select utterances which for simulated user and system which optimally fit to the memory activation.

4.7.2 Memory Modeling

A major building block of the cognitive user model is its memory model. It is implemented following the Dynamic Memory Model (DMM) described in Section 3.3. We use the memory model to represent the current activation of different information, from which we derive plausible user actions and optimal system actions.

Importance of an Item

From the definition of activation in Section 3.3, we see that the activation values of memory items summarize multiple influences and time scales of memory effects: On the one hand, base level activation can be interpreted as the result of learning from frequent presentations of the corresponding memory item. On the other hand, the volatile spreading activation does not result from past presentations of the item but from associations to other activated items. To create questions of the user or information statements of the system, we need to identify the most relevant memory items for a given context. For this purpose, we have two different criteria on activation: First, we do not want to ask or present information which is highly activated due to frequent and recent presentation causing a high base level activation: Such information is already known to the user. Second, items which are not activated at all will be uninteresting or unexpected for the user. We therefore define the concept of *importance* of an item as the ratio between spreading activation and base level activation. This marks a chunk as important if it was activated through spreading from associated memory items, but not directly presented.

Interest of an Item

To identify the most relevant memory items (e.g. to select the most appropriate system utterance), it is not sufficient to use the activation value as an indication of relevance. Pure activation caused by stimulation from the context is not a very precise parameter to determine relevance for the user. For example, activation of a POI rises when it is mentioned by the system, but this POI may be irrelevant to the driver, because he or she is not interested in this category of POI. We therefore define an additional value called *interest*. Interest is the static component of relevance and accounts for the fact that certain information is intrinsically more interesting for the simulated user than other, independently of the current activation values. For example, a user might always be more interested in churches than in sport stadiums, even if items from the latter category are currently more activated.

Interest is modeled as a probability distribution over a discrete scalar variable. Probabilities are modeled using a Bayesian network based on the same topology as the semantic memory network, but using only the ontological associations. Using this Bayesian representation, interest in general categories is related to the interest value of associated specific chunks. It is possible to define a-priori distributions to represent expert knowledge on item interest. This model can be used both in a generative manner in the 0^{th}-order model (to model individual differences in interest by setting evidence to the nodes of the network) and in a predictive manner in the 1^{st}-order model.

For interest generation, at the beginning of each episode, the 0^{th}-order model sets evidence to all category nodes in the interest network. The evidence is generated by iterating over the model beginning at the root nodes and randomly generating evidence from the respective distribution given the already set nodes. For interest prediction, information from questions or statements that express a degree of interest (e.g. “I am very interested in churches”) are integrated into the 1^{st}-order model by setting evidence in the nodes or the interest network which correspond to the mentioned concepts.

Model Updating & Utterance Processing

At the beginning of an episode, we begin with a memory network with no activation besides noise for all model orders. While the user drives on a route, stimuli for an increase of activation come from the *environment model* which represents events outside the driver cabin that influence the user. Examples are events which trigger the activation of certain memory items. Such events are based on tracking the user’s position on his route, fetching nearby points-of-interest (POI) from our database and activating them in the memory model. This activation spreads to connected nodes for which the spreading activation (but not the base activation) rises and therefore their importance. In case of the 1^{st}-order model, memory items can also be stimulated by direct user intervention, asking for information on the POI. The same is true for the 0^{th}-order model and system utterances.

To connect the user and system utterances with the items from the memory model, we tag the utterances with the associated memory items. For user utterances, the associated memory items are usually objects and concepts that the utterance requests information about. For system utterances, the tagged memory items are to relations that describe the knowledge that is encoded within the utterance. When an utterance of one agent (system or user) is perceived by the other agent, the chunks associated to that utterance are retrieved and stimulated in the respective memory model. In the case of system utterances, this will stimulate relations which often connect to other chunks previously not activated for the user and which are then brought into focus. In the case of user utterances, stimulation targets objects and spreading is received by adjacent relations, which gives the system cues on new useful information.

4.7.3 Workload Modeling

A big advantage of the presented utterance selection scheme is the ability to naturally integrate various cognitive modulators that influence the selection process. Mental workload is a very crucial modulator for memory and utterances selection. Therefore, it is modeled in our cognitive user simulation. The implemented workload model follows the multi-resource model of Wickens [Wic08]. In this model, workload is not represented as a uniform scalar variable but consists of different dimensions of workload. Those dimensions correspond to processing stage, input modality, response modality or processing code and type of visual processing. The purpose of the the multiple-resource-model (MRM) is to determine workload in a dual-tasking situation. The general principle of the model is to compare the resource demands for two cognitive tasks which are executed in parallel. Examples for cognitive tasks in the tourguide domain are “processing a system utterance”, “generating an utterance”, or “processing an external event” (different external events result in different cognitive tasks). The higher the overlap between the two tasks, the higher the workload which results from dual-tasking.

Resource demand of a task is formalized as a resource vector which denotes the cognitive resources required for the execution of this task on the four MRM dimensions on an integer scale. To calculate workload, we check the interference between the tasks on each dimension, i.e. the amount of overlap across all dimensions. For each overlapping dimension, a constant value is added to the workload level. Besides interference, another component which contributes to the overall workload level is the combined task demand of the individual tasks.

To integrate this model into the user simulation, we must define tasks and corresponding resource vectors. Tasks can be associated to actions of the user, to actions of the system (which are perceived and processed by the user) or to environmental events. It is also possible to define tasks associated with ongoing cognitive activity, for example driving or a generic dummy task similar to the one used in Section 3.5 to reflect increased workload level caused by tasks outside of the scope of the simulation. To adapt the MRM for our system, we need to include a number of extensions to the original approach by Wickens: First, we extend it by improving the handling of more than two tasks. We do this by repeating the interference calculation for each pair of tasks, adding up the resulting workload values. This leads to quadratic growth of comparisons with the number of tasks. Second, as the original model by Wickens has no notion of time, we extend it by adding a duration to each scheduled task. Then, each time slice of the simulation, all tasks which do not have their duration exceeded are evaluated to calculate the overall workload level.

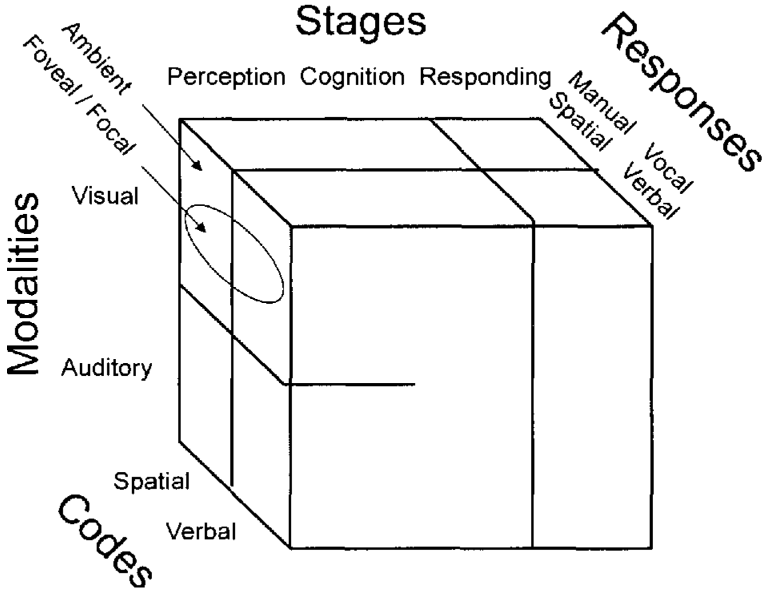


Figure 4.14 – The four dimensions of Wickens multiple-resource workload model [Wic08].

The last extension of the MRM focuses on the handling of situations of mental overload. Once the general workload level exceeds a certain threshold t, not all scheduled tasks can be executed with full quality [GWK+08]. To handle such situations, we follow the analogy of understanding cognitive tasks executed in parallel as threads in an operating system [ST08]: This analogy leads to the introduction of priority values which are assigned to each cognitive task. When t is exceeded, all scheduled tasks are sorted by priority. Beginning with the task of highest priority, tasks are executed with full quality until the cumulative workload induced by them reaches t. The remaining tasks are removed from the schedule and not executed. The task which is responsible for the workload exceeding t is still executed but with reduced quality.

The overall workload model influences how the simulated user processes information. First, it controls the user's chance of disregarding a system utterance. Second, it influences how well the information of a regarded system utterance is memorized by the user. Disregarding of a system utterance happens with a probability depending on the overall workload level. More precisely, a sigmoid function is used to map overall workload level to a failure probability[7]. The difficulty for successfully processing an utterance depends on its complexity. This is modeled by assigning a task with lower intrinsic task demand and lesser cognitive resource demand to utterances of low complexity (i.e. utterances which contain few semantic concepts). When a system utterance passes this check (i.e. is not disregarded by the user), a high workload still influences the activation and learning of the presented items in the memory model, following the approach presented in Section 3.4. This results in a lower activation and faster forgetting for items when workload is high.

We demonstrate the functioning of the workload model with an example. We first consider a user who is driving a car on a curved rural road with moderate traffic. We model driving as a task of unlimited duration. The resource vector associated to this task is shown in the first entry of Table 4.16. The table lists the resource demands for the driving task in terms of the dimensions of the MRM: visual perception (both focal and ambient), spatial processing codes (processed information are mainly distances, velocities, etc.) and manual responses. The resource demands are derived from the driving model described in [Sal06]. When the user listens to the system which gives direction information to the user, we add another task "Information (a)" for the duration of the system utterance. The resource vector of this task loads on auditory perception and the processing of verbal and spatial information, see again Table 4.16. We calculate the interference of the two tasks which yields an overall workload of 4.2 (see Table 4.17). This value results from the sum of 1) number of individual resource demands of the two tasks and 2) the workload resulting from resource demand overlap (in the example: Perception/Cognition cognition stage, Spatial processing code). The exact

[7] A degraded execution quality leads to an additional multiplicative penalty on failure probability.

resulting value depends on the weights between summands 1) and 2) and the score which is attributed to each resource overlap.

It should be noted that Wickens himself saw the merit of his model rather in the relative comparison of different situations than in the absolute prediction of performance [Wic08]. Consequently, the interpretation of absolute workload values should be considered as highly context-dependent. Reference values (e.g. to derive parameters for error probability calculation) can be established as suggested by [GWK$^+$08] by choosing prototypical task combinations for which task performance or subjective workload values are known. For example, [vE11] investigated the resulting workload level of different combinations of driving tasks and secondary tasks in an expert workshop.

Returning to the example, what happens if another distraction takes place during the processing of the system information? This depends on the resource vector which is added. In our example, we add an auditory distraction ("Distraction(a)") representing the user listening to a radio message. The resource vector of this task has high interference with the already scheduled tasks, which leads to a steep rise in overall workload when added to the tasks "Driving" and "Information(a)" (from 4.2 to 12.3). The dimensions which contribute to this rise are the "verbal" processing code and the "auditory" perceptual modality.

One way to remedy this problematic situation is to enable the system to switch information presentation modalities. The system could replace the auditory presentation with a visual one of semantically equivalent information. For the example of a directional information, a spatial hint (e.g. an arrow) could be presented on an in-car display or a heads-up display on the windshield. Compared to the task "Information(a)", the visual information task "Information(v)" replaces the auditory component with a visual one and removes the processing of verbal information. As a consequence, overall workload level drops from 12.3 to 10.7.

Note that the visual information is not superior in general to the verbal one. For example, the combination of the visual information task with a distraction task which uses visual and spatial resources (e.g. avoiding an overtaking ambulance, "Distraction(v)") leads to even higher overall workload compared to the combination of auditory information with an auditory distraction (see Table 4.16). This interaction between different tasks underlines the necessity of modeling and recognition of different dimensions of workload to select the optimal system action.

Task	Resource Vector
Driving	Perception.Visual.Ambient.Spatial Perception.Visual.Focal.Spatial Cognition.Spatial Responding.Manual
Information(a)	Perception.Auditory.Verbal Cognition.Verbal Cognition.Spatial
Distraction(a)	Perception.Auditory.Verbal Cognition.Verbal
Information(v)	Perception.Visual.Focal.Spatial Cognition.Spatial
Distraction(v)	Perception.Visual.Focal.Spatial Perception.Visual.Ambient.Spatial Cognition.Spatial Responding.Manual

Table 4.16 – Examples of state and action attributes for system and user. Perceptual modalities: a = auditory, v = visual

Task 1	Task 2	Task 3	Overall Workload
Driving	-	-	0.4
Driving	Information(a)	-	4.2
Driving	Information(a)	Distraction(a)	12.3
Driving	Information(v)	Distraction(a)	10.7
Driving	Information(v)	Distraction(v)	16.8

Table 4.17 – Overall workload level resulting from different combinations of tasks. Perceptual modalities: a = auditory, v = visual

4.7.4 Joint Strategy Optimization

The previous subsections described the computational modeling components of the user model. In this subsection, we describe the statistical, RL-based approach to determine the behavior of system and simulated user.

In many cases of interaction strategy optimization, an RL framework is already in place for the training of interaction strategies of the system (see Section 4.2.3). This framework can be naturally extended to form a framework in which both the system and a simulated user are represented as agents that use RL to jointly optimize their strategies.

In terms of RL theory, we transform the single-agent learning problem as described in Section 3.2.1 to a multi-agent problem. The single-agent problem is usually defined to consist of an agent representing the interaction system and an environment which primarily consists of a static user simulation that describes the behavior of a generic user. In the multi-agent problem, there are two learning agents, one representing the user and the other one presenting the system. Both agents can actively influence the joint environment. The task which both agents are trying to solve – i.e. creating a efficient, effective and pleasant interaction – is cooperative in nature, although the formal definitions of rewards, states and actions can differ between both agents.

State Update & Action Selection

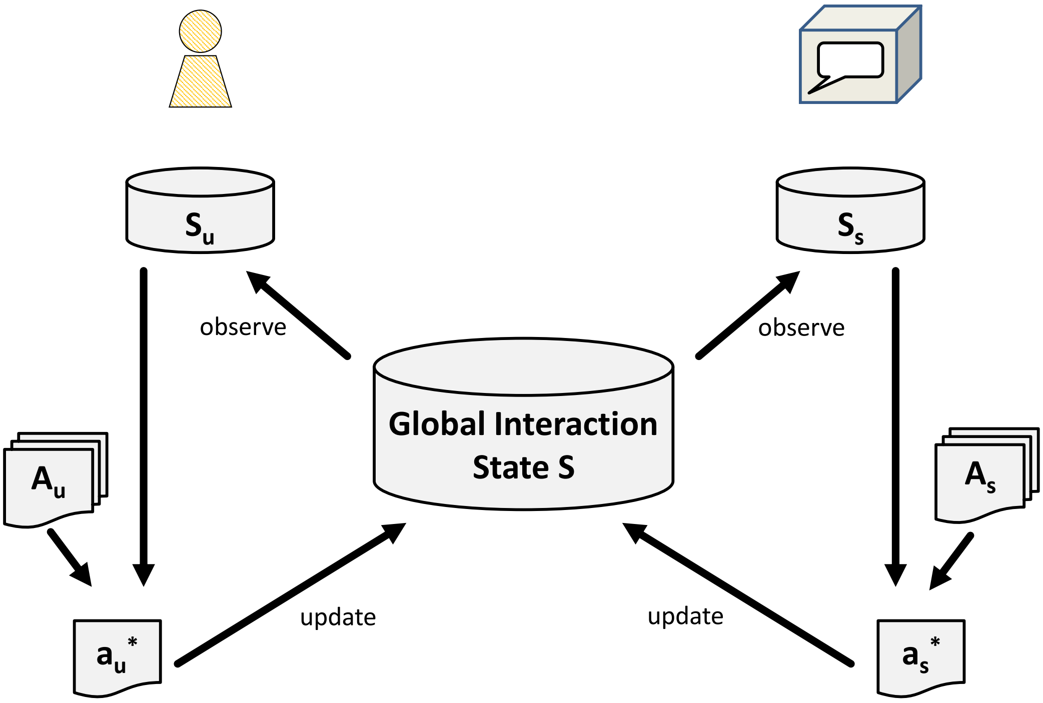


Figure 4.15 – The implemented RL-based simulation framework with two simultaneously acting agents representing the user and the system.

The setup for the learning framework is designed as follows (see Figure 4.15): For every time slice, both agents simultaneously observe the current state from their perspective (S_u and S_s) and chose an action (a_u^* and s_s^*) from their available action set (A_u and A_s) according to their exploration strategy and their learned Q-values. Our current implementation explores according to the

Softmax paradigm during the training phase and uses greedy exploitation during simulation. The action sets always contain an *empty action* that produces no output to handle situations in which an agent decides to remain silent. The actions of both agents are simultaneously used to update the global interaction state S. The agents are then informed about the state update and perform a learning step (both agents implement the Sarsa(0) learning rules, which is a variant of Q-Learning as described in Section 3.2.1).

States and actions consist of several discrete attributes. The attribute set comprises attributes which describe the discourse and internal states of the user (derived from the computational modeling components of the user model). Examples for state and action attributes are given in Table 4.18. With the introduction of many attributes to the state and action space, generating a sufficient number of training episodes can quickly become impossible. A modularization approach which we describe later proposes a solution to this dimensionality problem.

Dimension	**Explanation**
State Dimensions	
FLOOR	who is currently speaking (user, system, both or none)?
WORKLOAD	degree of mental workload of the user.
WORKLOADSENSOR	workload as estimated by a noisy sensor of the system.
Action Dimensions	
SPEECHACT	which speech act is this action part of?
UTTERANCESTATE	at which position (begin, middle, end) of the utterance is this action?
COMPLEXITY	how complex is the associated utterance? (estimated from number of concepts in the utterance)

Table 4.18 – Examples of state and action attributes for system and user.

The two agents do not have the same perspective on the global dialog state. For example, while the simulated user can directly access its true internal state which drives its behavior, this state is not directly visible to the system agent. Instead, the system agent has to rely on the output of simulated sensors which reflect the noisy output of the empirical cognitive models that observe the user. This separation corresponds to the different orders of user models defined in Section 4.7.1.

Actions & Speech Acts

For time representation, a dialog is partitioned into short time slices of 0.5 s. The actions of simulated user and system therefore work on a sub-speech act level. Modeling the actions of the agents on a speech act level (as it is common in statistical dialog simulation) would be too restrictive to allow a prompt reaction to changes in the behavior of the other agent or of the dynamic environment. While actions of both agents are defined on a very fine time scale, they can be grouped to describe the interaction on the speech act level. Each action belongs to exactly one speech act. Because the action selection takes place on the fine time scale, speech acts can be aborted or restarted at every point of the interaction. This is especially relevant in dynamic scenarios. However, condensed status information on the speech acts (e.g. the fraction to which a speech act is completed) is set in form of attributes of the corresponding action to influence the action selection process (see Table 4.18 for examples). More generally: To each action, background information is attached which is not directly visible to the action selection process but stores information from which action attributes can be derived. Most importantly, this comprises the concrete utterance which realizes the corresponding speech act. The utterance also determines the duration of the speech act, i.e. how many subsequent actions have to be executed for the speech act to be completed. When a speech act is completed (i.e. all corresponding actions were executed), the attached utterance is processed by both agents. For the agent which perceives the utterance, it passes through an error model. This error model can lead to disregarding the utterance, with a probability depending on certain model parameters (e.g. average mental workload during the utterance in the case of the simulated users). This simulates both malfunction of the speech processing components of the system (i.e. rejection of a user utterance as no ASR result could be produced which matched the current context) and missed utterances due to mental overload of the user.

Training of Strategies

The training of the two agents consists of two stages: Learning from action scripts and learning by exploration. An *action script* defines one complete episode of interaction, consisting of system and user actions as well as external events in a fixed temporal order. Those episodes either describe one critical situation the designer wants to confront the learning agents with or are created by annotating existing interactions. The action scripts help the agents to learn the fundamentals of meaningful interaction. The scripts are processed in a flexible way that randomly introduces slight modifications to the script in each iteration, resulting in a broader range of trained episodes. The second stage of learning lets both agents explore the state-action space autonomously, using Softmax exploration as described above.

The motivation for this two-stage approach is that immediately using Softmax exploration with randomly initialized learning tables would lead to very chaotic behavior in the initial steps of multi-agent training. Therefore, we start with a training on action scripts and switch to Softmax exploration when a baseline behavior has been learned. When creating learning episodes using the Softmax exploration strategy, we have to consider that mostly optimal behavior with occasional random explorations can still create chaotic behavior due to frequent abortions of speech acts. To counter that, we create a certain small amount of sessions in which the exploration probability is globally set to zero.

Cognitive Urges

To generate plausible user behavior, it is crucial to define an appropriate reward function for the user agent. What the user desires is dependent on the state of the interaction, but also on the inner state of the user and his or her individual preferences. For example, a user under high mental workload may be less interested in non-crucial information than a user who is less busy and may welcome an entertaining interaction. It is clear that a reward which reflects a large set of desires cannot be defined as a monolithic function.

Instead, we calculate the weighted sum of the output of multiple reward functions. Each of those represents a single requirement the user has for a satisfying interaction. Different reward functions may be orthogonal or even adversarial to each other. For example, the desire to maintain a low mental workload favors other actions than the desire to acquire additional information by maximizing the interaction efficiency. To find a cognitively valid structure of reward functions, we resort to the concept of *urges* as defined by Dörner [DSS99] in his cognitive Ψ-architecture. We adapt this concept to the domain of conversational interaction systems. An urge models one unique fundamental need of an agent and should be as universal as possible for the application in information-providing interaction systems. For the tourguide application, we define the following urges: The most general desire in this context is the *information urge*, i.e. the desire to acquire more information. It is determined by calculating the sum of importance scores for all chunks, weighted by the corresponding interest value. Additionally, the *activation grounding urge* is the user's desire to reduce the perceived discrepancy between his or her own mental state and the mental model he or she believes the system has of him or her. This would enable the system to provide more suited information. Formally, we model this urge as the discrepancy between 0^{th}-order model and 2^{nd}-order model in activation by calculating a sum of weighted differences between importance scores. The *interest grounding urge* does the same for interest scores. All three urges can be seen as a specialization of Dörner's *cognitive urges* which describe the desire for competence and a reduction of uncertainty.

In addition, there is the urge that controls the need to obey social conventions in spoken interactions (in analogy to Dörner's *social urges*), i.e. to follow the common rules of interaction: turntaking, finishing utterances, etc. As the agents initially do not know anything about appropriate behavior in dialog, they have to learn it with the help of adequate reward functions, which are combined in the social urge. Each of those functions rewards or punishes a specific behavior in the interaction. Examples for those kinds of behavior are a penalty for aborting unfinished utterances, a penalty for barge-in, reward for respecting the turn order, etc. This urge allows both participants to show compliant dialog behavior. Due to the modular urge structure, agents are still able to break the learned social rules when necessary, e.g. when an emergency forces the immediate abortion of an utterance. Finally, there is a *workload reduction urge* (corresponding to Dörner's urge of energy preservation) which targets a low workload of the user.

Modularization

A learning algorithm for interaction strategies in dynamic environments faces the problem of a large state and action space. A larger number of dimensions comes with exponentially growing need for training episodes to still guarantee convergence to a local maximum. In addition, larger state and action spaces become difficult to store in computer memory. We deal with this situation by decomposing the learning task into smaller *modules*. Each module is implemented as its own RL-agent. All modules which belong to the simulated are combined to one meta-agent (and likewise for the system). The urge concept introduced in the previous paragraph induces a natural structure for this modularization. Each module is associated with its own reward function and a separate view on the state and action space. This way, the module only sees those parts of the state and action space which are relevant for its own task defined by the reward function. To determine the action of an agent, its observed state and its available action set are mapped into the reduced state and action space of each module. The modules then consult their own scoring tables and the scores for each action are accumulated over all modules. This approach is similar to the "greatest mass" algorithm [Kar97]. Each module is associated with a weight that determines the influence of the corresponding module on the decision process. Each module can also maintain a separate set of learning parameters (e.g. learning rate or discount factor) to account for the fact that the learning problems of different modules may be of different complexity and set on different time scales (e.g. some work on action level and some on speech act level). Our current implementation contains several mapping mechanisms to offer full flexibility in reducing the state and action space. This includes complete removal of irrelevant attributes (e.g. the workload attribute of the dialog state for the module handling turn order), transforming an attribute to a coarser granularity by identifying certain attributes with each other (e.g. by collapsing a 5-point scale to a 3-point scale) and conditional mappings to preserve more details for certain critical states or actions and remove them for the rest. A configuration file allows the simple definition of which attributes and mapping rules belong to a certain module, which reward rules apply and how the default learning parameters are defined. Note that the modularization concept does not imply that individual cognitive concepts are treated in isolation. For example, the module which corresponds to the "information urge" retrieves user state information on memory activation and on the workload level.

4.7.5 Utterance Selection

The action selection process for both system and simulated user consists of two intertwined steps: An action is selected using the RL-based mechanism (as described in the previous subsection) and in addition, a concrete realization of the speech act to which the action belongs is selected in form of an utterance (for example, "This restaurant serves affordable French cuisine" is an utterance which realizes the speech act `give_information`). For a given speech act, we define an optimality criterion that defines the best matching utterance in the current context. For example, the optimality criterion for utterances belonging to the `give_information` speech act is maximum importance of the memory items associated with the utterance. Evaluating these criteria for all eligible utterances (see below for a definition of eligibility) of a speech act, the optimal utterance is selected. The selection of the speech act itself is then performed by the RL-based speech act selection, which uses condensed information on the selected utterance for each speech act (e.g. a score representing the goodness of the best utterance). This information is propagated by the utterance selection to the RL-based speech act selection in form of state variables.

Before the speech act specific utterance selection process starts, we perform a pre-selection to identify the eligible utterances for each agent (system or user). We exclude those utterances that do not contain any item with context activation[8] as those do not refer to any item that was part of the (subjective) discourse of the agent. We further exclude utterances that do not have context activation for memory items that are marked as mandatory context referents for the given utterance (e.g. “tower” in the utterance “How high is this tower?”). After this pre-selection, only utterances that are valid within the current context are considered, which simplifies the later steps of the selection process.

We now present the agent-specific methods of utterance selection for speech acts in the tourguide domain for both system and simulated user.

System Utterance Selection

The primary goal of the utterance selection of the system is to find an utterance that is both interesting to the user and relevant given the current estimated configuration of the user’s memory model. This is exactly the definition of interest and importance as given in Section 4.7.2. The system therefore iterates over all utterances remaining after the pre-selection and calculates the expected reduction of the information urge (see Section 4.7.4) which would take place when the corresponding utterance is processed by the user. This leads to selection of utterances that target relations that have high spreading activation (because they are relevant in the current context), low base activation (because they are new to the user) and a high interest score. The utterance with highest reduction is selected. When multiple utterances share the highest score, one of those utterances is selected randomly. The information on information urge reduction is then stored in a state vari-

[8] i.e. base activation or spreading activation

able which is used by the RL module which corresponds to the “information urge”.

User Utterance Selection

For a simulated user, the goal of utterance selection is different from that of the system: The user has to enable the system to derive the most relevant and interesting information by sending cues on their own memory configuration. Therefore, it is the user’s goal to reduce the mismatch between the own memory and their belief on the model configuration the system has of his memory. This corresponds to the “activation grounding urge” and the “interest grounding urge”. In our implementation, the user has two speech acts available to achieve this goal: `ask_most_important` selects an utterance that asks a question on one or more concepts in the domain and mainly conveys information on the importance of certain items. The utterances can be generic (e.g. “What do you know about that tower?”) or specific in nature (e.g. “Who is the architect of this tower?”) and are tagged with the chunks in the knowledge base that are associated with this chunk. The second speech act is `give_metainfo` which yields information on the level of interest the user has in a certain concept.

When the `ask_most_important` speech act is selected, the simulation has to determine the optimal utterance given the current memory configuration of the user. The optimality criterion for the utterance selection of this speech act is as follows: An utterance is optimal if it is associated to memory items that are highly activated in the 0^{th}-order model but not in the 2^{nd}-order model. These items are witnesses for mismatch between user and system memory model. Processing of an utterance which is associated to these items will reduce this mismatch. This is because stimulating the items in the 0^{th}-order model will increase activation from a low baseline, but stimulating them in the 2^{nd}-order model will increase activation only slightly from an already high baseline. The reduction of mismatch is the essence of a grounding process. Since the `ask_most_important` utterances of the user do not only convey information on importance but also on interest, the mismatch reduction of each chunk is weighted by its interest score.

A similar principle applies to the `give_metainfo` speech act. We select an utterance that best fits the level of interest in the 0^{th}-order model and for which the difference between the actual interest and the expected interest in the 2^{th}-order model is large. To break ties, the simulated user prefers positive results, i.e. utterances that indicate strong interest.

4.7.6 Evaluation

To demonstrate the feasibility of our approach, we resort to evaluation of complete interactions by human judges. This idea has been introduced by [AL08] as an alternative to the prevalent family evaluation methods which use similarity-based metrics to compare the generated interactions with existing corpora. The drawback of similarity-based approaches is that they are very conservative in scenarios where many different interaction flows are possible. Additionally, they require the existence of a data corpus in the scenario which is identical to the simulated one. Using human judges for evaluation circumvent both problems and target the goal of generating subjectively plausible interactions.

Within the tourguide scenario, we generate interactions using the described simulation framework, using a list of predefined timed context events (e.g. POIs along the route or workload triggers). We use different setups of the memory model and utterance generation to vary the interaction strategy for comparison (see Table 4.19). The **SM** interaction was generated using a fully trained strategy with activated memory model, corresponding to the setup which would be applied for a cognitive interaction system. This interaction is compared to two different baseline interactions. The first one, **nSnM** was generated by using a completely random strategy and without memory model. This resulted in an interaction with frequent barge-ins and chaotic behavior. The second baseline, **SnM**, was generated by using a fully trained strategy but also without memory model. This removed the ability for both user and system to estimate whether additional information was useful and which information was most relevant. The resulting interactions still have a satisfying "surface structure", but lack semantic coherence. The reason for choosing two different baseline approaches was to differentiate between effects by the interaction strategy and the memory model. The additional baseline **nSM** was omitted because the use of a memory model would not be beneficial without an optimized strategy: The memory model supports optimal utterance selection but without optimal speech act selection, the optimal utterances will not be propagated. As a gold standard, we use a handcrafted "oracle" interaction **HC** that was designed specifically for the scripted interaction context. Note that for better comparison, **HC** is limited to the same set of utterances as the generated interactions but with manually defined selection and timing.

Interaction Strategy	Strategy	Memory
no strategy, no memory (nSnM)	-	-
strategy, no memory (SnM)	+	-
strategy, memory (SM)	+	+
handcrafted (HC)	++	++

Table 4.19 – Characteristics of the different employed interaction strategies.

Example Interaction

Table 4.20 shows an example of a completely generated interaction using the **SM** strategies for the system and the user trained in the described RL framework. The example is set in the tourguide domain, where the simulated driver rides through the fictional city Los Santos, passing several POIs. The example lists the utterances generated by both agents as well as external events that were triggered during the interaction and which influenced the dialog flow. The example is translated from German so given timings are only approximations.

The example demonstrates how a meaningful conversation emerges from agents which start with no hardcoded knowledge on “adequate” interaction behavior. With the help of the RL training and the implemented cognitive models, they learned to take and finish their turns, respect a changing workload level (e.g. when the user remains quiet during the difficult traffic situation at time 28.0 to 58.0) and to produce utterances suited to the context of the interaction.

Simulation Quality Assessment

The following list presents the (translated) questionnaire items for the evaluation of the generated interactions. Many of the employed items relate to the Gricean Maxims [Pau75]. Those maxims describe fundamental principles of human communication based on the cooperative principle, regarding quality, quantity, relevance and manner. Items marked with a star were added in a second version of the questionnaire. They were included to cover for phenomena which arise in the studied scenario but which are not handled by the existing questions based on the Gricean Maxims. Each item was presented with a 6-point scale ranging from 1 (=very poor) to 6 (=very good).

1. **SYSTEM:** Rating of the system’s interaction behavior:

Time (in s)	Event/Utterance
6.0	**Workload (caused by traffic)**
8.5	Left!
16.0	**POI (UFO restaurant)**
16.0	*Tell me something about this UFO!*
19.5	This UFO actually contains an exclusive restaurant.
22.5	*Tell me something about this UFO!*
25.5	The architect was inspired by an UFO, which he saw crash in the nearby desert.
28.0	**Steering (Left)**
30.5	**Workload (caused by traffic)**
32.5	Please turn right.
40.0	**Steering (Right)**
40.0	Right!
42.0	**POI (UFO restaurant)**
55.0	The UFO restaurant was designed by the famous architect Gonzales Sanchez.
58.0	**Steering (Right)**
60.0	*How is this building called?*
62.5	It is called the “Revolving UFO restaurant”.
64.0	*Can I meet celebrities in the restaurant?*
67.5	The singer Joe Brown is a regular guest here.
...	...

Table 4.20 – Excerpt of a generated interaction in the tourguide scenario between simulated user and system. It contains external events (bold), user utterances (italic) and system utterances (normal font).

(a) Appropriateness of the amount of presented information (e.g. as much as necessary, as little as possible)

(b) Quality of the presented information

(c) Relevance in context of the presented information

(d) Manner of utterances and interaction behavior

(e) Respect to the changing workload of the user*

(f) Intelligibility of the synthesized voice*

2. **USER:** Rating of the user’s interaction behavior:

   (a) Appropriateness of the amount of presented information (e.g. as much as necessary, as little as possible)

   (b) Quality of the presented information

   (c) Relevance in context of the presented information

   (d) Manner of utterances and interaction behavior

   (e) Intelligibility of the synthesized voice*

3. **OVERALL:** Rating of the interaction overall:

   (a) Coordination between system and user*

   (b) Overall naturalness

   (c) Overall quality

29 people (22 men and 7 women, all of them technical students of the Karlsruhe Institute of Technology) participated in the study as judges. Interactions were generated in the simulation, then synthesized using Text-to-Speech component with different voices for system and user. In the driving simulator software[9], we recorded a driving scene which corresponded to the scenario of the interaction and which was synchronized with the triggered events. This video was played to the judges in combination with the synthesized interaction to create a realistic and multimodal experience for them.

Table 4.21 shows the scores for each of the interactions averaged for the questions concerning the simulated driver, the system, the overall interaction and for all questions. Multiple pairwise t-tests (corrected for multiple testing) were performed to show that all four interactions differ in average score significantly ($p < 0.01$). The order of the scores confirmed our expectation that **nSnM** < **SnM** < **SM** < **HC**. It also demonstrates that while **SnM** already performs significantly better than **nSnM**, **SM** is much closer to **HC** in average score than to the baseline interactions. This shows that our approach is able to generate meaningful and to some degree natural interaction behavior. We also see that this result is accountable to both the trained RL-based behavior used by **SnM** and **SM** and to the memory model and the memory-based utterance selection which is only used by **SM**. A correlation analysis between the scores for different items and the average score shows that the mean correlation coefficient for the items concerning the system ($r = 0.72$) is higher than the coefficient for items dealing with

[9] a modified version of Grand Theft Auto: San Andreas with the Multi Theft Auto extension.

the simulated user ($r = 0.52$). However, the strongest correlation ($r = 0.84$) is for items that evaluate the interaction as a whole and rate naturalness and coordination between both participants. This stresses the importance of a joint development and evaluation of user simulation and system, as we performed it in this section.

	nSnM	**SnM**	**SM**	**HC**
user	2.20	3.53	3.59	4.37
system	1.91	2.16	4.01	4.29
overall	1.14	1.86	3.45	3.79
average	1.83	2.63	3.67	4.18

Table 4.21 – Averaged scores for 29 human judges, evaluating the different simulation modes.

4.7.7 Conclusion

In this section, we saw how the systematic integration of computational cognitive models can be employed for cognitive plausible dialog simulation. The user simulation was evaluated in a tourguide scenario, which is characterized by a user's dynamically changing cognitive state, for example caused by external events which influence workload level and interaction context. The user simulation includes 1) a model of memory activation and interest 2) a multi-dimensional model of workload based on the Multiple-Resource-Model. A hierarchy of model instances of different order was used to represent the cognitive state of the simulated user as well as the indirect tracing of this cognitive state by the interaction system. We showed how those cognitive models can be used to generate system and user utterances which are plausible in the context of interaction history and external events. The computational models where integrated in a joint strategy optimization based on Reinforcement Learning for the system and the simulated user. The strategy optimization uses the concept of cognitive urges to structure and reduce the state-action space for optimization.

We demonstrated the plausibility of the simulation by having human judges evaluate generated interactions by criteria of plausibility and well-formed communication. To our best knowledge, this section presents the first result to the research community on the usage of computational cognitive models in statistical user simulation. It therefore provides to the research community a basis for the investigation of optimal adaptive system behavior in dynamic interaction scenarios.

One limitation of the presented approach is that both the system and the simulated user make use of the same memory model structure. While we designed the simulation to maintain a hierarchy of separated models for user and system, this design still introduces a bias for behavior which is tailored towards the this specific model design. The focus of the presented evaluation was on the plausibility of the generated interactions. This goal was not jeopardized by the chosen design as the evaluation by human judges was completely oblique to the employed models. However, once the goal changes to training strategies for optimal information throughput, the current approach may be too optimistic regarding the grounding process between system and user. To remedy this limitation, we suggest to perform an evaluation of the trained systems with different user simulation approaches (following the idea of [SSWY05]) or to ultimately evaluate the trained strategies in a study with real users.

The evaluation concentrated on the plausibility of the generated utterances and while it addressed the appropriateness in reaction to changing workload levels, it did not focus on the different aspects of the employed workload model. This model bears much potential for future investigations. In sections 2.3 and 2.4, we showed empirical cognitive models for general workload level and input modality discrimination. Both aspects of user state are reflected as dimensions in the workload model. Regarding both, the model generates plausible and useful predictions, for example that it is beneficial to resort to auditory system output when primarily visual secondary tasks are executed (i.e. the resulting overall workload and the resulting failure probabilities are lower). Still, the model guides us to more factors which are relevant for the predicted behavior. For example, also the type of processing code (i.e. spatial vs. verbal) influences the overall workload. This means, for a complete model, not only the modality but also the type of processed information should be available, to discriminate the visual processing of text (verbal) from the visual processing of motion and position (spatial). We expect this differentiation to be challenging using EEG-based empirical models only, as it requires information which is not clearly localized and easily accessible. However, context information and other sensors might help to provide this information.

4.8 Discussion

In this chapter, we described the development of AIM, the adaptive interaction manager. The AIM integrates information from cognitive models to adapt its interaction behavior to predictions on user states. To show the versatility of this approach, we implemented and evaluated three different applications regarding different aspects of adaptive interaction management: adaptation of information presentation style, variable levels of intrusiveness, and strategies for proactive recovery from recognition errors. As shown in the related work section, adaptive interaction systems have been already investigated in the literature. However, most existing research concentrated on traditional graphical user interfaces and investigated only task success as quality metric. Our main focus and also our main contribution to the research community is the extensive evaluation of multimodal adaptive systems by investigating both objective and subjective quality metrics. The results from the conducted experiments show that with the methods developed in this thesis it is possible to create cognitive adaptive interaction systems which provide a significant measurable benefit for the user compared to non-adaptive interaction systems. However, we also saw that the impact of an adaptation strategy strongly depends on its exact behavior. In Sections 4.5 and 4.6.2, we compared different adaptation strategies and showed that even small differences in system behavior strongly influenced objective and subjective performance measures. This implies that the development of the adaptation strategy is a central research topic and not just an afterthought to cognitive modeling. This thesis provides fundamental findings to this topic and lays an important foundation for future research.

Chapter 5

Conclusion

5.1 Discussion

In this thesis, we developed the general framework of adaptive cognitive interaction systems. We showed the realization of such a system for several user states, including mental workload, confusion, and memory. Every component of an end-to-end implementation of an adaptive cognitive interaction system was developed and thoroughly evaluated, in isolation and in combination with other components.

While some of the findings in this thesis are specific to the investigated scenarios and applications, we made a number of important and general findings which will be relevant for future developments of adaptive cognitive interaction systems. Many of the results of this thesis are to our best knowledge completely new to the research community. In the following, we review the key results:

Regarding empirical cognitive models, we explored several approaches to provide person-independent or person-adapted recognition of user states from physiological sensors: We developed an empirical cognitive model for the user state workload. We validated this on an exceptionally large data corpus with multiple cognitive tasks which allowed us to test the model for different definitions of workload, including task engagement, task count and task difficulty. To classify the user state workload type, we developed and validated the first hybrid EEG-fNIRS passive BCI. For the user state confusion, we developed an EEG-based empirical model which reduced setup time by training data selection and investigated transfer of classification models between tasks. All these contributions, which focus on short setup time and evaluation in realistic scenarios, enable the application of empirical cognitive models in the HCI context.

Regarding computational cognitive modeling, we provided and evaluated a stand-alone dynamic memory model component for the prediction of associative memory processes in human-computer interaction. For the first time, we demonstrated the successful modulation of computational models to different workload levels. For this purpose, we compared two fundamentally different approaches: The explicit overlay approach and the implicit dummy-model approach. Each approach has its own advantages and drawbacks. Together, both approaches form a toolbox to adapt the prediction of computational cognitive models to different user states. To adapt the computational model, real-time information on the user's workload level is required. This information can be provided by an empirical workload model. As another example for the combination of a computational cognitive model with an empirical cognitive model, we show how to predict a user state (learning situations) which no single type of model was able to predict. The described contributions enable the application of computational cognitive models in the HCI context which rich application domains and dynamically changing workload levels.

Regarding interaction management, we implemented the light-weight adaptive interaction manager AIM which we used for the successful realization of multiple adaptive cognitive interaction systems. Most of the developed components – models and interaction systems – have been employed in several different scenarios and provide important building blocks for rapid development of end-to-end adaptive cognitive interaction systems. Using those components, we were able to demonstrate significant, measurable benefit of adaptive interaction behavior, considering system accuracy, efficiency and user satisfaction. For the first time, we provided a systematic evaluation of adaptive interfaces which does not only use objective quality criteria but also of subjective quality criteria. When comparing strategies of different levels of intrusiveness, we demonstrated that objective and subjective measures of usability may be negatively correlated with regard to adaptive systems. We also implemented the first error-aware gesture interface which compared different error recovery strategies. We showed that automatic error recovery was able to reduce gesture recognition error rate and was preferred by the users compared to manual correction.

All those contributions show the importance of an integrated end-to-end perspective during the development of adaptive cognitive interaction systems: For the development of empirical cognitive models, it was important to regard the constraints which were imposed by the envisioned interactive application: selection of relevant user states, required set-up time and computational complexity, required classification accuracy to achieve a measurable benefit. For the development of computational cognitive models, we had to regard the implications by the HCI context: topic drifts in large application domains, dynamically changing workload levels, existence of user states which could not be captured by computational cognitive modeling alone. For the development of adaptive interaction management, we had to investigate the development of strategies which optimally map various detected user states to system behavior. For all those reasons, we aimed for scientific contributions to all three aspects of adaptive cognitive interaction systems and our results show that this endeavor was successful.

One major challenge of this work was the combination of methods and terminology from machine learning, neuroscience, cognitive psychology and usability research. We had to coin common terms for the components of an adaptive cognitive interaction systems, define how the different user states were represented within the various models, define the interactions between those models and finally define the mappings of user states to behavior of an interaction system. Another challenge of this work was the desired thorough evaluation of the proposed methods in several user studies. For each building block of an adaptive cognitive interaction system, we provided empirical data and statistical analysis to investigate the reliability as well as the limitations of the developed methods. Overall, recordings of more than 250 participants have been conducted to collect data and to validate the created models. This large number of participants indicates that we performed thorough testing of the developed models and interaction systems with real users on large corpora. We believe that this is an important feature of this thesis as it documents the high validity of the results and and shows that their dependency on effects specific to certain individual recordings is small.

5.2 Future Work

While this work laid a solid foundation for the development of adaptive cognitive interaction systems, there are still open research questions which occurred during the creation of this thesis. The goal of this final section is to outline a number of directions for future research which naturally emerge from the results of this thesis.

Regarding empirical cognitive models, one limitation of this work is that user states and models were mostly investigated in isolation, i.e. there is for example no interaction between the workload model and the confusion model. However, we expect the combination of the already investigated user states workload and confusion to to have a huge potential for improving the assessment of the overall user state: On the one hand, high workload increases the probability of errors (e.g. because input gestures are not executed as precisely compared to low workload situations). On the other hand, high workload also increases the probability of a user to miss the erroneous feedback caused by misinterpreted input, which would result in the absence of an error potential as marker for this error. A combined empirical model of those user states could represent those dependencies and learn it from data.

One important future step regarding computational cognitive models would be the development of additional models for certain executive functions for the application in an HCI context. One crucial component would be a model of visual perception. Information on user's gaze is relevant for any task which presents visual information to the user, for example for the myriad of graphical user interfaces. There exist approaches for computational modeling of gaze on a fine-grain level [DU07] as well as on a more strategically-abstract level [BB11]. In [PHK$^+$13], we showed how a combination of eyetracking technology and EEG can be used for a completely automatic temporal-spatial localization of attention of a person in a gaze-based attention task. As we showed that the combination of empirical and computational cognitive modeling can yield benefits which cannot achieved by either approach in isolation, it seems both feasible and fruitful to also pursue this approach for models of gaze and visual attention.

Regarding the evaluation of adaptive cognitive interaction systems, future research should concentrate on extending the presented investigation in two dimensions: Long-term evaluation of adaptive systems and investigation of co-adaptation effects. First, the presented experiments were limited in time during which the users were exposed to the system and the adaptive technology. Although each experiment contained training stages to help the user familiarize with the system, learning effects are still likely to occur. More importantly, user preferences regarding adaptive behavior could change over time, because users get familiar with the effects of adaptation and learn to better predict system behavior, which could lead to increased trust in the decisions of the system. The second research direction we propose for investigation is co-adaption [Mac00]. Co-adaptation occurs as a user's reaction to the changes in behavior of an adaptive system. Additionally, users do not only react to the adaptive system, but also to the initial state which triggered the adaptation. For example, in highly interactive HCI, a high workload state will not only affect how the user processes the output of the system, but also how the user generates input to the system. This has implications for the optimal behavior of the system, which could for example involve switching to a system initiative approach to reduce the user's workload. The joint strategy optimization in Section 4.7 is a first step in this direction and shows how multi-agent learning can be used to simulate co-adaptation.

Bibliography

[ABB$^+$04] John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, 2004.

[ABFF10] John R. Anderson, Shawn Betts, Jennifer L. Ferris, and Jon M. Fincham. Neural imaging to track mental states while using an intelligent tutoring system. *Proceedings of the National Academy of Sciences*, 107(15):7018–7023, 2010.

[ABLM98] John R. Anderson, Dan Bothell, Christian Lebiere, and Michael Matessa. An integrated theory of list memory. *Journal of Memory and Language*, 38(4):341–380, 1998.

[ADL99] Wendy S. Ark, David C. Dryer, and Davia J. Lu. The emotion mouse. In *Proceedings of the 8th International Conference on Human-Computer Interaction*, page 818–823, 1999.

[AHR07] Robert St. Amant, Thomas E. Horton, and Frank E. Ritter. Model-based evaluation of expert cell phone menu interaction. *Transactions on Computer-Human Interaction*, 14(1), 2007.

[AL08] Hua Ai and Diane J. Litman. Assessing dialog system user simulation evaluation measures using human judges. In *The 46th Annual Meeting of the Association for Computational Linguistics*, page 622–629, 2008.

[And07] John R. Anderson. Using brain imaging to guide the development of a cognitive architecture. *Integrated models of cognitive systems*, page 49–62, 2007.

[AP08] Brendan Z. Allison and John Polich. Workload assessment of computer gaming using a single-stimulus event-related potential paradigm. *Biological psychology*, 77(3):277–283, 2008.

[AR99] John R. Anderson and Lynne M. Reder. The fan effect: New results and new theories. *Journal of Experimental Psychology: General*, 128(2):186–197, 1999.

[ARL96] John R. Anderson, Lynne M. Reder, and Christian Lebiere. Working memory: Activation limitations on retrieval. *Cognitive Psychology*, 30(3):221–256, 1996.

[AS68] Richard C. Atkinson and Richard F. Shiffrin. Human memory: A proposed system and its control processes. In *The psychology of learning and motivation*, volume 2, pages 89–195. Academic Press, 1968.

[AWA$^+$08] Tamin Asfour, Kai Welke, Pedram Azad, Ales Ude, and Rüdiger Dillmann. The karlsruhe humanoid head. In *8th International Conference on Humanoid Robots*, pages 447–453, 2008.

[AYG12] Kai Keng Ang, Juanhong Yu, and Cuntai Guan. Extracting effective features from high density nirs-based BCI for assessing numerical cognition. In *International Conference on Acoustics, Speech and Signal Processing*, page 2233–2236, 2012.

[AZD03] Hojjat Adeli, Ziqin Zhou, and Nahid Dadmehr. Analysis of EEG records in an epileptic patient using wavelet transform. *Journal of Neuroscience Methods*, 123(1):69–87, 2003.

[Bac09] Joscha Bach. *Principles of Synthetic Intelligence PSI: An Architecture of Motivated Cognition*. Oxford University Press, 2009.

[Bad92] Alan D. Baddeley. Working memory. *Science*, 255(5044):556–559, 1992.

[BANAG10] Mohamed Ben Ammar, Mahmoud Neji, Adel. M. Alimi, and Guy Gouardères. The affective tutoring system. *Expert Systems with Applications*, 37(4):3013–3023, 2010.

[Bar09] Thomas Barkowsky. CASIMIR–a computational architecture for modeling human spatial information processing. *Complex Cognition*, page 93, 2009.

[BB11] Thomas Bader and Jürgen Beyerer. Influence of user's mental model on natural gaze behavior during human-computer interaction. In *2nd Workshop on Eye Gaze in Intelligent Human Machine Interaction*, 2011.

[BBH89] James C. Byers, Alvah C. Bittner, and Susan G. Hill. Traditional and raw task load index (TLX) correlations: Are paired comparisons necessary. *Advances in industrial ergonomics and safety I*, page 481–485, 1989.

[Bec12] Lucas Bechberger. *Modeling human memory performance under influence of cognitive workload.* Bachelor thesis, Karlsruhe Institute of Technology, 2012.

[BHvE+12] Anne-Marie Brouwer, Maarten A. Hogervorst, Jan B.F. van Erp, Tobias Heffelaar, Patrick H. Zimmerman, and Robert Oostenveld. Estimating workload using EEG spectral power and ERPs in the n-back task. *Journal of neural engineering*, 9(4):045008, 2012.

[BIA+11] Scott C. Bunce, Kurtulus Izzetoglu, Hasan Ayaz, Patricia Shewokis, Meltem Izzetoglu, Kambiz Pourrezaei, and Banu Onaral. Implementation of fNIRS for monitoring levels of expertise and mental workload. In *Foundations of Augmented Cognition. Directing the Future of Adaptive Systems*, number 6780 in Lecture Notes in Computer Science, pages 13–22. Springer Berlin Heidelberg, 2011.

[BLK10] Cyril Brom, Jiří Lukavský, and Rudolf Kadlec. Episodic memory for human-like agents and human-like agents for episodic memory. *International Journal of Machine Consciousness*, 02(02):227–244, 2010.

[BLMP12] Ligia Batrinca, Bruno Lepri, Nadia Mana, and Fabio Pianesi. Multimodal recognition of personality traits in human-computer collaborative tasks. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, page 39–46, New York, USA, 2012.

[BLR+05] Chris Berka, Daniel J. Levendowski, Caitlin K. Ramsey, Gene Davis, Michelle N. Lumicao, Kay Stanney, Leah Reeves, Susan H. Regli, Patrice D. Tremoulet, and Kathleen Stibler. Evaluation of an EEG workload model in an aegis simulation environment. In *Defense and Security*, volume 5797, pages 90–99, 2005.

[BLT$^+$11] Benjamin Blankertz, Steven Lemm, Matthias Treder, Stefan Haufe, and Klaus-Robert Müller. Single-trial analysis and classification of ERP components — a tutorial. *NeuroImage*, 56(2):814–825, 2011.

[BM69] William F. Battig and William E. Montague. Category norms of verbal items in 56 categories a replication and extension of the connecticut category norms. *Journal of Experimental Psychology*, 80(3, Pt.2):1–46, 1969.

[BM10] Matthias Bezold and Wolfgang Minker. A framework for adapting interactive systems to user behavior. *Journal of Ambient Intelligence and Smart Environments*, 2(4):369–387, 2010.

[BPM$^+$11] Felix Biessmann, Sergey Plis, Frank C. Meinecke, Tom Eichele, and Klaus-Robert Müller. Analysis of multimodal neuroimaging data. *Reviews in Biomedical Engineering*, 4:26–58, 2011.

[BPNZ06] Trung H. Bui, Mannes Poel, Anton Nijholt, and Job Zwiers. A tractable DDN-POMDP approach to affective dialogue modeling for general probabilistic frame-based dialogue systems, September 2006.

[BR08] Dan Bohus and Alexander I. Rudnicky. Sorry, i didn't catch that! In *Recent Trends in Discourse and Dialogue*, number 39 in Text, Speech and Language Technology, pages 123–154. Springer Netherlands, 2008.

[BR09] Dan Bohus and Alexander I. Rudnicky. The RavenClaw dialog management framework: Architecture and systems. *Computer Speech & Language*, 23(3):332–361, 2009.

[Bra91] Jason Brandt. The hopkins verbal learning test: Development of a new memory test with six equivalent forms. *Clinical Neuropsychologist*, 5(2):125–142, 1991.

[Bre98] Susan E. Brennan. The grounding problem in conversations with and through computers. In S. R. Fussell and R. J.

Kreuz, editors, *Social and Cognitive Approaches to Interpersonal Communication*, pages 201–225. Lawrence Erlbaum, Hillsdale, USA, 1998.

[BRH$^+$07] Dan Bohus, Antoine Raux, Thomas K. Harris, Maxine Eskenazi, and Alexander I. Rudnicky. Olympus: An open-source framework for conversational spoken language interface research. In *Proceedings of the NAACL-HLT Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, page 32–39, Stroudsburg, USA, 2007.

[BSF$^+$06] Nathan R. Bailey, Mark W. Scerbo, Frederick G. Freeman, Peter J. Mikulka, and Lorissa A. Scott. Comparison of a brain-based adaptive system and a manual adaptable system for invoking automation. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(4):693–709, 2006.

[Büt10] Phillippe Büttner. "hello java": Linking ACT-r 6 with a java simulation. In *Proceedings of the 10th International Conference on Cognitive Modeling*, pages 289–290, Philadelphia, USA, 2010.

[BvBE$^+$09] Felix Burkhardt, Markus van Ballegooy, Klaus-Peter Engelbrecht, Tim Polzehl, and Joachim Stegmann. Emotion detection in dialog systems: Applications, strategies and challenges. In *3rd International Conference on Affective Computing and Intelligent Interaction*, pages 1–6, 2009.

[CB07] James A. Coan and J. B, editors. *Handbook of emotion elicitation and assessment*, volume viii of *Series in affective science.* Oxford University Press, New York, NY, US, 2007.

[CBE12] Emily B. J. Coffey, Anne-Marie Brouwer, and Jan B. F. van Erp. Measuring workload using a combination of electroencephalography and near infrared spectroscopy. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 56(1):1822–1826, 2012.

[CCM$^+$12] Adrien Combaz, Nikolay Chumerin, Nikolay V. Manyakov, Arne Robben, Johan A. K. Suykens, and Marie M. Van Hulle. Towards the detection of error-related potentials and its integration in the context of a p300 speller brain–computer interface. *Neurocomputing*, 80:73–82, 2012.

[CD10] Rafael Calvo and Sidney D'Mello. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1(1):18–37, 2010.

[CDWG08] Scotty D. Craig, Sidney D'Mello, Amy Witherspoon, and Art Graesser. Emote aloud during learning with AutoTutor: Applying the facial action coding system to cognitive–affective states during learning. *Cognition & Emotion*, 22(5):777–788, 2008.

[CE13] James C. Christensen and Justin R. Estepp. Coadaptive aiding and automation enhance operator performance. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, page 0018720813476883, 2013.

[CF12] Anne G. E. Collins and Michael J. Frank. How much of reinforcement learning is working memory, not reinforcement learning? a behavioral, computational, and neurogenetic analysis. *European Journal of Neuroscience*, 35(7):1024–1035, 2012.

[CFKA10] James F. Cavanagh, Michael J. Frank, Theresa J. Klein, and John J. B. Allen. Frontal theta links prediction errors to behavioral adaptation in reinforcement learning. *NeuroImage*, 49(4):3198–3209, 2010.

[CGLP11] Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefevre, and Olivier Pietquin. User simulation in dialogue systems using inverse reinforcement learning. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association*, Florence, Italy, 2011.

[Cha09] Samuel G. Charlton. Driving while conversing: Cell phones that distract and passengers who react. *Accident Analysis & Prevention*, 41(1):160–173, 2009.

[Cho59] Noam Chomsky. A review of BF skinner's verbal behavior. *Language*, 35(1):26–58, 1959.

[Chr92] Sven-Åke Christianson. *The Handbook of Emotion and Memory: Research and Theory.* Psychology Press, 1992.

[CK92] Hintat Cheung and Susan Kemper. Competing complexity metrics and adults' production of complex sentences. *Applied Psycholinguistics*, 13(01):53–76, 1992.

[CKE03] Andrew R. A. Conway, Michael J. Kane, and Randall W. Engle. Working memory capacity and its relation to general intelligence. *Trends in Cognitive Sciences*, 7(12):547–552, 2003.

[CL07] Hua Cai and Yingzi Lin. An experiment to non-intrusively collect physiological parameters towards driver state detection. SAE Technical Paper 2007-01-0403, Warrendale, USA, 2007.

[CL11] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. In *Transactions on Intelligent Systems and Technology*, volume 2, 2011.

[CLC06] Robert E. Cochran, Frank J. Lee, and Eric Chown. Modeling emotion: Arousal's impact on memory. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, 2006.

[Con02] Cristina Conati. Probabilistic assessment of user's emotions in educational games. *Applied Artificial Intelligence*, 16(7-1):555–575, 2002.

[Con13] Cristina Conati. Virtual butler: What can we learn from adaptive user interfaces? In *Your Virtual Butler*, number 7407 in Lecture Notes in Computer Science, pages 29–41. Springer Berlin Heidelberg, January 2013.

[Cow93] Nelson Cowan. Activation, attention, and short-term memory. *Memory & Cognition*, 21(2):162–167, March 1993.

[Cow09] Roddy Cowie. Perceiving emotion: towards a realistic understanding of the task. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3515–3525, 2009.

[CP10] Belkacem Chikhaoui and Hélène Pigot. Towards analytical evaluation of human machine interfaces developed in the context of smart homes. *Interacting with Computers*, 22(6):449–464, 2010.

[CTN09] Yujia Cao, Mariët Theune, and Anton Nijholt. Modality effects on cognitive load and performance in high-load information presentation. In *Proceedings of the 14th International*

Conference on Intelligent User Interfaces, New York, USA, 2009.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[CV04] Daniel Chen and Roel Vertegaal. Using mental load for managing interruptions in physiologically attentive user interfaces. In *Extended Abstracts on Human Factors in Computing Systems*, page 1513–1516, USA, 2004.

[Dan13] Christopher L. Dancy. ACT-RΦ: A cognitive architecture with physiology and affect. *Biologically Inspired Cognitive Architectures*, 6:40–45, 2013.

[DCDM+05] Ellen Douglas-Cowie, Laurence Devillers, Jean-Claude Martin, Roddy Cowie, Suzie Savvidou, Sarkis Abrilian, and Cate Cox. Multimodal databases of everyday emotion: facing up to complexity. In *Proceedings of Interspeech*, Lissabon, 2005.

[DHS12] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley & Sons, November 2012.

[DK12] Sidney D'Mello and Jacqueline Kory. Consistent but modest: a meta-analysis on unimodal and multimodal affect detection accuracies from 30 studies. In *Proceedings of the 14th international conference on Multimodal interaction*, New York, USA, 2012.

[DLS10] Nate Derbinsky, John E Laird, and Bryan Smith. Towards efficiently supporting large symbolic declarative memories. In *Proceedings of the 10th International Conference on Cognitive Modeling*, Philadelphia, PA, USA, 2010.

[DM04] Arnaud Delorme and Scott Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of neuroscience methods*, 134(1):9–21, 2004.

[DM10] Scott A. Douglass and Christopher W. Myers. Concurrent knowledge activation calculation in large declarative memories. In *Proceedings of the 10th International Conference on Cognitive Modeling*, 2010.

[DMBHT+08] Renato De Mori, Frederic Bechet, Dilek Hakkani-Tur, Michael McTear, Giuseppe Riccardi, and Gokhan Tur. Spoken language understanding. *IEEE Signal Processing Magazine*, 25(3):50–58, May 2008.

[Dör01] Dietrich Dörner. *Bauplan für eine Seele.* Hogrefe & Huber, 2001.

[DSS99] Dietrich Dörner, Harald Schaub, and Stefan Strohschneider. Komplexes problemlösen - königsweg der theoretischen psychologie? *Psychologische Rundschau*, 50(4):198–205, 1999.

[DU07] Jeronimo Dzaack and Leon Urbas. Cognitive model data analysis for the evaluation of human computer interaction. In Don Harris, editor, *Engineering Psychology and Cognitive Ergonomics*, number 4562 in Lecture Notes in Computer Science, pages 477–486. Springer Berlin Heidelberg, 2007.

[DVP$^+$13] Sabine Deprez, Mathieu Vandenbulcke, Ron Peeters, Louise Emsell, Frederic Amant, and Stefan Sunaert. The functional neuroanatomy of multitasking: Combining dual tasking with a short term memory task. *Neuropsychologia*, 51(11):2251–2260, 2013.

[DWM13] Chris Dijksterhuis, Dick de Waard, and Ben L. J. M. Mulder. Classifying visuomotor workload in a driving simulator using subject specific spatial brain patterns. *Frontiers in Neuroprosthetics*, 7, 2013.

[EH05] Michael S. English and Peter A. Heeman. Learning mixed initiative dialog strategies by using reinforcement learning on both conversants. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Stroudsburg, USA, 2005.

[ELP97] Wieland Eckert, Esther Levin, and Robert Pieraccini. User modeling for spoken dialogue system evaluation. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, USA, 1997.

[EM81] Thomas D. Erickson and Mark E. Mattson. From words to meaning: A semantic illusion. *Journal of Verbal Learning and Verbal Behavior*, 20(5):540–551, 1981.

[Emo06] Bruno Emond. WN-LEXICAL: An ACT-r module built from the WordNet lexical database. In *Proceedings of the Sev-*

enth International Conference on Cognitive Modeling, Trieste, Italy, 2006.

[Eng14] Klaus-Peter Engelbrecht. A user model for dialog system evaluation based on activation of subgoals. In *Natural Interaction with Robots, Knowbots and Smartphones*, pages 363–374. Springer New York, January 2014.

[ERB$^+$08] Rick Evertsz, Frank E. Ritter, Paolo Busetta, Matteo Pedrotti, and Jennifer L Bittner. CoJACK—achieving principled behaviour variation in a moderated cognitive architecture. In *Proceedings of the 17th conference on behavior representation in modeling and simulation*, page 80–89, 2008.

[ES79] Charles W. Eriksen and Derek W. Schultz. Information processing in visual search: A continuous flow conception and experimental results. *Perception & Psychophysics*, 25(4):249–263, 1979.

[FA06] Wai-Tat Fu and John R. Anderson. From recurrent choice to skill learning: A reinforcement-learning model. *Journal of Experimental Psychology: General*, 135(2):184–206, 2006.

[FA08] Wai-Tat Fu and John R. Anderson. Extending the computational abilities of the procedural learning mechanism in ACT-r. Technical report, Carnegie Mellon University, 2008.

[Fai09] Stephen H. Fairclough. Fundamentals of physiological computing. *Interacting with Computers*, 21(1–2):133–145, 2009.

[FBC$^+$10] Kilian Förster, Andrea Biasiucci, Ricardo Chavarriaga, Jose del R Millan, Daniel Roggen, and Gerhard Tröster. On the use of brain decoded signals for online user adaptive gesture recognition systems. In *Pervasive Computing*, number 6030 in Lecture Notes in Computer Science, pages 427–444. Springer Berlin Heidelberg, 2010.

[FDH12] Karen M. Feigh, Michael C. Dorneich, and Caroline C. Hayes. Toward a characterization of adaptive systems a framework for researchers and system designers. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 54(6):1008–1024, 2012.

[FHH00] Erik Frøkjær, Morten Hertzum, and Kasper Hornbæk. Measuring usability: are effectiveness, efficiency, and satisfaction really correlated? In *Proceedings of the Conference on Human Factors in Computing Systems*, New York, USA, 2000.

[FHW04] Christian Fügen, Hartwig Holzapfel, and Alex Waibel. Tight coupling of speech recognition and dialog management-dialog-context dependent grammar weighting for speech recognition. In *Proceedings of Interspeech*, 2004.

[Fit54] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391, 1954.

[FJ08] Pierre W. Ferrez and José del R. Millan. Error-related EEG potentials generated during simulated brain computer interaction. *IEEE Transactions on Biomedical Engineering*, 55(3):923–929, 2008.

[FM00] Myra A. Fernandes and Morris Moscovitch. Divided attention and memory: Evidence of substantial interference effects at retrieval and encoding. *Journal of Experimental Psychology: General*, 129(2):155–176, 2000.

[FMS$^+$12] Siamac Fazli, Jan Mehnert, Jens Steinbrink, Gabriel Curio, Arno Villringer, Klaus-Robert Müller, and Benjamin Blankertz. Enhanced performance by a hybrid NIRS–EEG brain computer interface. *NeuroImage*, 59(1):519–529, 2012.

[FRL11] Kate Forbes-Riley and Diane Litman. Designing and evaluating a wizarded uncertainty-adaptive spoken dialogue tutoring system. *Computer Speech & Language*, 25(1):105–126, 2011.

[FRL12] Kate Forbes-Riley and Diane Litman. Adapting to multiple affective states in spoken dialogue. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '12, Stroudsburg, USA, 2012.

[GC95] Gabriele Gratton and Paul M. Corballis. Removing the heart from the brain: compensation for the pulse artifact in the photon migration signal. *Psychophysiology*, 32(3):292–299, 1995.

[GCTW06] Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. Exploring the design space for adaptive graphical user interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, New York, USA, 2006.

[GE11] Tobias Gehrig and Hazim K. Ekenel. A common framework for real-time emotion recognition and facial action unit detection. In *Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2011.

[GET+08] Krzysztof Z. Gajos, Katherine Everitt, Desney S. Tan, Mary Czerwinski, and Daniel S. Weld. Predictability and accuracy in adaptive user interfaces. In *Proceedings of the Conference on Human Factors in Computing Systems*, page 1271–1274, New York, USA, 2008.

[GFW+06] Harold P. Greeley, Eric Friets, John P. Wilson, Sridhar Raghavan, Joseph Picone, and Joel Berg. Detecting fatigue from voice using speech recognition. In *International Symposium on Signal Processing and Information Technology*, 2006.

[GHW08] Philipp W.L. Große, Hartwig Holzapfel, and Alex Waibel. Confidence based multimodal fusion for person identification. In *Proceedings of the 16th International Conference on Multimedia*, New York, USA, 2008.

[GR08] Milan Gnjatović and Dietmar Rösner. Adaptive dialogue management in the NIMITEK prototype system. In *Perception in Multimodal Dialogue Systems*, number 5078 in Lecture Notes in Computer Science, pages 14–25. Springer Berlin Heidelberg, 2008.

[GRG+11] Glenn Gunzelmann, L. Richard, Kevin A. Gluck, P. A, and David F. Dinges. Fatigue in sustained attention: Generalizing mechanisms for time awake to time on task. In *Cognitive fatigue: Multidisciplinary perspectives on current research and future applications*, Decade of Behavior/Science Conference., pages 83–101. American Psychological Association, Washington, DC, USA, 2011.

[GTH+12] Milica Gašić, Pirros Tsiakoulis, Matthew Henderson, Blaise Thomson, Kai Yu, Eli Tzirkel, and Steve Young. The effect of cognitive load on a statistical dialogue system. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 74–78, Stroudsburg, PA, USA, 2012.

[GWK+08] Rebecca Grier, Christopher Wickens, David Kaber, David Strayer, Deborah Boehm-Davis, J. Gregory Trafton, and

Mark St John. The red-line of workload: Theory, research, and design. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 2008.

[GWM10] Kallirroi Georgila, Maria K. Wolters, and Johanna D. Moore. Learning dialogue strategies from older and younger simulated users. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Stroudsburg, USA, 2010.

[HCG+09] Leanne M. Hirshfield, Krysta Chauncey, Rebecca Gulotta, Audrey Girouard, Erin T. Solovey, Robert J. K. Jacob, Angelo Sassaroli, and Sergio Fantini. Combining electroencephalograph and functional near infrared spectroscopy to explore users' mental workload. In *Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience*, number 5638 in Lecture Notes in Computer Science, pages 239–247. Springer Berlin Heidelberg, 2009.

[HD01] Peter A. Hancock and Paula A. Desmond, editors. *Stress, workload, and fatigue.* Human factors in transportation. Lawrence Erlbaum Associates Publishers, Mahwah, USA, 2001.

[HDFB09] Theodore J. Huppert, Solomon G. Diamond, Maria A. Franceschini, and David A. Boas. HomER: a review of time-series analysis methods for near-infrared spectroscopy of the brain. *Applied optics*, 48(10):D280–D298, 2009.

[HG04] Hartwig Holzapfel and Petra Gieselmann. A way out of dead end situations in dialogue systems for human-robot interaction. In *Proceedings of the 4th International Conference on Humanoid Robots*, 2004.

[HHF+14] Christian Herff, Dominic Heger, Ole Fortmann, Johannes Hennrich, Felix Putze, and Tanja Schultz. Mental workload during n-back task—quantified in the prefrontal cortex using fNIRS. *Frontiers in Human Neuroscience*, 7(00935), 2014.

[HNW08] Hartwig Holzapfel, Daniel Neubig, and Alex Waibel. A dialogue approach to learning object descriptions and semantic categories. *Robotics and Autonomous Systems*, 56(11):1004–1013, 2008.

[HO00] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4–5):411–430, 2000.

[Hol05] Hartwig Holzapfel. Building multilingual spoken dialogue systems. *Archives of Control Sciences*, Vol. 15, no. 4:555–566, 2005.

[HP05] Jennifer A. Healey and Rosalind W. Picard. Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):156–166, 2005.

[HPA$^+$10] Dominic Heger, Felix Putze, Christoph Amma, Michael Wand, Igor Plotkin, Thomas Wielatt, and Tanja Schultz. BiosignalsStudio: a flexible framework for biosignal capturing and processing. In *Proceedings of the 33rd Annual German Conference on Artificial Intelligence*, Karlsruhe, Germany, 2010.

[HPS11] Dominic Heger, Felix Putze, and Tanja Schultz. Online recognition of facial actions for natural EEG-based BCI applications. In *Affective Computing and Intelligent Interaction*, Lecture Notes in Computer Science, pages 436–446. Springer Berlin Heidelberg, January 2011.

[HS88] Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In *Advances in Psychology*, volume 52 of *Human Mental Workload*, pages 139–183. North-Holland, 1988.

[Ick93] William Ickes. Empathic accuracy. *Journal of Personality*, 61(4):587–610, 1993.

[ICM$^+$12] Inaki Iturrate, Ricardo Chavarriaga, Luis Montesano, Javier Minguez, and Jose del R Millan. Latency correction of error potentials between different experiments reduces calibration time for single-trial classification. *Proceedings of Annual International Conference of the Engineering in Medicine and Biology Society.*, 2012:3288–3291, 2012.

[Ior10] Borislav Iordanov. HyperGraphDB: A generalized graph database. In *Web-Age Information Management*, number 6185 in Lecture Notes in Computer Science, pages 25–36. Springer Berlin Heidelberg, 2010.

[Jam09] Anthony David Jameson. Understanding and dealing with usability side effects of intelligent processing. *AI Magazine*, 30(4):23–40, 2009.

[Jas58] Herbert Henri Jasper. The ten twenty electrode system of the international federation. *Electroencephalography and clinical neurophysiology*, 10:371–375, 1958.

[JCK$^+$96] Marcel Adam Just, Patricia A. Carpenter, Timothy A. Keller, William F. Eddy, and Keith R. Thulborn. Brain activation modulated by sentence comprehension. *Science*, 274(5284):114–116, 1996.

[JH91] Christian Jutten and Jeanny Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

[JHFB06] Danny K. Joseph, Theodore J. Huppert, Maria A. Franceschini, and David A. Boas. Diffuse optical tomography system to image brain activation with improved spatial resolution and validation with functional magnetic resonance imaging. *Applied optics*, 45(31):8142–8151, 2006.

[JL09] Srinivasan Janarthanam and Oliver Lemon. A two-tier user simulation model for reinforcement learning of adaptive referring expression generation policies. In *Proceedings of the 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Stroudsburg, USA, 2009. Association for Computational Linguistics.

[JMW$^+$00] Tzyy-Ping Jung, Scott Makeig, Marissa Westerfield, Jeanne Townsend, Eric Courchesne, and Terrence J. Sejnowski. Removal of eye activity artifacts from visual event-related potentials in normal and clinical subjects. *Clinical Neurophysiology*, 111(10):1745–1758, 2000.

[JSW$^+$99] Anthony Jameson, Ralph Schäfer, Thomas Weis, André Berthold, and Thomas Weyrath. Making systems sensitive to the user's time and working memory constraints. In *Proceedings of the 4th International Conference on Intelligent User Interfaces*, New York, USA, 1999.

[JT02] Ole Jensen and Claudia D Tesche. Frontal theta activity in humans increases with memory load in a working memory task. *The European journal of neuroscience*, 15(8):1395–1399, 2002.

[JY02] Qiang Ji and Xiaojie Yang. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging*, 8(5):357–377, 2002.

[KA07] Kenneth R. Koedinger and Vincent Aleven. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, 19(3):239–264, 2007.

[Kai67] Thomas Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, 1967.

[Kar97] Jonas Karlsson. *Learning to Solve Multiple Goals*. PhD thesis, University of Rochester, 1997.

[KBB+12] Janina Krell, Sabrina Benzinger, Klaus Boes, Jeremias Engelmann, Dominic Heger, Felix Putze, Tanja Schultz, and Alexander Stahn. Physical activity, brain function and cognitive performance in young adults-a cross-sectional study. In *Medicine and Science in Sports and Exercise*, volume 44, Philadelphia, USA, 2012.

[KCM10] M. Asif Khawaja, Fang Chen, and Nadine Marcus. Using language complexity to measure cognitive load for adaptive interaction design. In *Proceedings of the 15th international conference on Intelligent user interfaces*, New York, USA, 2010.

[KCVP07] Julien Kronegg, Guillaume Chanel, Svyatoslav Voloshynovskiy, and Thierry Pun. EEG-based synchronized brain-computer interfaces: A model for optimizing the number of mental tasks. *Transactions on Neural Systems and Rehabilitation Engineering*, 15(1):50–58, 2007.

[KDB+07] Jens Kohlmorgen, Guido Dornhege, Mikio Braun, Benjamin Blankertz, Klaus-Robert Müller, Gabriel Curio, Konrad Hagemann, Andreas Bruns, Michael Schrauf, and Wilhelm Kincses. Improving human performance in a real operating environment through real-time mental workload detection. In *Toward Brain-Computer Interfacing*, pages 409–422. 2007.

[KDS+99] Wolfgang Klimesch, Michael Doppelmayr, Josef Schwaiger, Petra Auinger, and Thomas Winkler. ‘paradoxical’ alpha synchronization in a memory task. *Cognitive Brain Research*, 7(4):493–501, 1999.

[KGJ+10] Simon Keizer, Milica Gašić, Filip Jurčíček, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. Parameter estimation for agenda-based user simulation. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '10, page 116–123, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[Kli96] Wolfgang Klimesch. Memory processes, brain oscillations and EEG synchronization. *International Journal of Psychophysiology*, 24(1–2):61–100, 1996.

[Kli99] Wolfgang Klimesch. EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain Research Reviews*, 29(2–3):169–195, April 1999.

[KM97] David E. Kieras and David E. Meyer. An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*, 12(4):391–438, 1997.

[KM11] Christian A. Kothe and Steve Makeig. Estimation of task workload from EEG data: New and current tools and perspectives. In *Proceedings of the Engineering in Medicine and Biology Society*, pages 6547–6551, 2011.

[KMSM13] Christian Keitel, Burkhard Maess, Erich Schröger, and Matthias M. Müller. Early visual and auditory processing rely on modality-specific attentional resources. *NeuroImage*, 70:240–249, 2013.

[KP12] Jens Kober and Jan Peters. Reinforcement learning in robotics: A survey. In *Reinforcement Learning*, number 12 in Adaptation, Learning, and Optimization, pages 579–610. Springer Berlin Heidelberg, January 2012.

[KRDP98] Wolfgang Klimesch, Harald Russegger, Michael Doppelmayr, and Thomas Pachinger. A method for the calculation of induced band power: Implications for the significance of brain oscillations. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 108(2):123–130, 1998.

[KSH07] Wolfgang Klimesch, Paul Sauseng, and Simon Hanslmayr. EEG alpha oscillations: The inhibition–timing hypothesis. *Brain Research Reviews*, 53(1):63–88, 2007.

[KSM$^+$08] Dean J. Krusienski, Eric W. Sellers, Dennis J. McFarland, Theresa M. Vaughan, and Jonathan R. Wolpaw. Toward enhanced p300 speller performance. *Journal of Neuroscience Methods*, 167(1):15–21, 2008.

[Kuh05] Friedemann Kuhn. Methode zur bewertung der fahrerablenkung durch fahrerinformations-systeme. *World Usability Day*, 2005.

[LAH$^+$09] Mei Yii Lim, Ruth Aylett, Wan Ching Ho, Sibylle Enz, and Patricia Vargas. A socially-aware memory for companion agents. In *Intelligent Virtual Agents*, number 5773 in Lecture Notes in Computer Science, pages 20–26. Springer Berlin Heidelberg, 2009.

[LBCK04] Terry C. Lansdown, Nicola Brook-Carter, and Tanita Kersloot. Distraction from multiple in-vehicle secondary tasks: vehicle performance and mental workload implications. *Ergonomics*, 47(1):91–104, 2004.

[LCGE$^+$11] Ramón López-Cózar, David Griol, Gonzalo Espejo, Zoraida Callejas, and Nieves Ábalos. Towards fine-grain user-simulation for spoken dialogue systems. In *Spoken Dialogue Systems Technology and Design*, pages 53–81. Springer New York, January 2011.

[LCW11] Darren J. Leamy, Rónán Collins, and Tomas E. Ward. Combining fNIRS and EEG to improve motor cortex activity classification during an imagined movement-based task. In *Foundations of Augmented Cognition. Directing the Future of Adaptive Systems*, page 177–185. Springer, 2011.

[LDG12] Tomás Lejarraga, Varun Dutt, and Cleotilde Gonzalez. Instance-based learning: A general model of repeated binary choice. *Journal of Behavioral Decision Making*, 25(2):143–153, 2012.

[LDR00] Marsha C. Lovett, Larry Z. Daily, and Lynne M. Reder. A source activation theory of working memory: cross-task prediction of performance in ACT-r. *Cognitive Systems Research*, 1(2):99–118, June 2000.

[LFDE05] Henry Lieberman, Alexander Faaborg, Waseem Daher, and José Espinosa. How to wreck a nice beach you sing calm incense. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, page 278–280, New York, USA, 2005.

[Lis14] Pierre Lison. *Structured Probabilistic Modelling for Dialogue Management*. PhD thesis, University of Oslo, 2014.

[LJB13] Nanxiang Li, Jinesh J. Jain, and Carlos Busso. Modeling of driver behavior in real world scenarios using multiple noninvasive sensors. *Transactions on Multimedia*, 15(5):1213–1225, 2013.

[LKSW00] Diane J. Litman, Michael S. Kearns, Satinder Singh, and Marilyn A. Walker. Automatic optimization of dialogue management. In *Proceedings of the 18th Conference on Computational Linguistics*, page 502–508, Stroudsburg, USA, 2000.

[LLP11] Martin Luerssen, Trent Lewis, and David Powers. Head x: Customizable audiovisual synthesis for a multi-purpose virtual head. In *AI 2010: Advances in Artificial Intelligence*, page 486–495. Springer, 2011.

[LNR87] John E. Laird, Allen Newell, and Paul S. Rosenbloom. SOAR: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64, 1987.

[LR11] Shengguang Lei and Matthias Rötting. Influence of task combination on EEG spectrum modulation for driver workload estimation. *Human Factors*, 53(2):168–179, 2011.

[LS04] Hugo Liu and Push Singh. ConceptNet — a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, 2004.

[LT00] Staffan Larsson and David R. Traum. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, 6(3&4):323–340, 2000.

[LvGG$^+$11] Andreas Llera, Marcel A. J. van Gerven, Victor M. Gómez, Ole K. Jensen, and Hilbert J. Kappen. On the use of interaction error potentials for adaptive brain computer interfaces. *Neural Networks*, 24(10):1120–1127, 2011.

[Mac00] Wendy E. Mackay. Responding to cognitive overload: Co-adaptation between users and technology. *Intellectica*, 30(1):177–193, 2000.

[Mat03] Stefan Mattes. The lane-change-task as a tool for driver distraction evaluation. *Quality of Work and Products in Enterprises of the Future*, page 57–60, 2003.

[McT02] Michael F. McTear. Spoken dialogue technology: enabling the conversational user interface. *Computing Surveys*, 34(1):90–169, 2002.

[MCWD06] Cynthia Matuszek, John Cabral, Michael Witbrock, and John DeOliveira. An introduction to the syntax and content of cyc. In *AAAI Spring Symposium*, 2006.

[MD12] Behnam Molavi and Guy A Dumont. Wavelet-based motion artifact removal for functional near-infrared spectroscopy. *Physiological measurement*, 33(2):259–270, 2012.

[MEB$^+$09] Florian Metze, Roman Englert, Udo Bub, Felix Burkhardt, and Joachim Stegmann. Getting closer: tailored human–computer speech dialog. *Universal Access in the Information Society*, 8(2):97–108, 2009.

[MEE$^+$06] Sebastian Möller, Roman Englert, Klaus Engelbrecht, Verena Hafner, Anthony Jameson, Antti Oulasvirta, Alexander Raake, and Norbert Reithinger. MeMo: Towards automatic usability evaluation of spoken dialogue services by user error simulations. In *Ninth International Conference on Spoken Language Processing*, 2006.

[MEK$^+$09] Sebastian Möller, Klaus Engelbrecht, Christine Kühnel, Ina Wechsung, and Benjamin Weiss. A taxonomy of quality of service and quality of experience of multimodal human-machine interaction. In *Proceedings of the International Workshop on Quality of Multimedia Experience*, pages 7–12, 2009.

[Mil95] George A. Miller. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[Mil03] George A Miller. The cognitive revolution: a historical perspective. *Trends in Cognitive Sciences*, 7(3):141–144, 2003.

[Min61] Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, January 1961.

[Min74] Marvin Minsky. A framework for representing knowledge. *The Psychology of Computer Vision*, June 1974.

[MJ05] Natasha Merat and A. Hamish Jamson. Shut up i'm driving!: Is talking to an inconsiderate passenger the same as talking on a mobile telephone? In *Proceedings of the 3rd International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*, pages 426–432, Rockport, USA, 2005.

[MJBB11] Andrea Mognon, Jorge Jovicich, Lorenzo Bruzzone, and Marco Buiatti. ADJUST: An automatic EEG artifact detector based on the joint use of spatial and temporal features. *Psychophysiology*, 48(2):229–240, 2011.

[MML$^+$09] Emily Mower, Angeleki Metallinou, Chi-Chun Lee, Abe Kazemzadeh, Carlos Busso, Sungbok Lee, and Shrikanth Narayanan. Interpreting ambiguous emotional expressions. In *Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction*, pages 1–8, 2009.

[Mon03] Stephen Monsell. Task switching. *Trends in Cognitive Sciences*, 7(3):134–140, 2003.

[MRM13] Amar R. Marathe, Anthony J. Ries, and Kaleb McDowell. A novel method for single-trial classification in the face of temporal variability. In *Foundations of Augmented Cognition*, number 8027 in Lecture Notes in Computer Science, pages 345–352. Springer Berlin Heidelberg, 2013.

[MS99] Akira Miyake and Priti Shah, editors. *Models of working memory: Mechanisms of active maintenance and executive control.* Cambridge University Press, New York, USA, 1999.

[Mur05] Atsuo Murata. An attempt to evaluate mental workload using wavelet transform of EEG. *Human Factors*, 47(3):498–508, 2005.

[MW10] François Mairesse and Marilyn A. Walker. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction*, 20(3):227–278, 2010.

[NB05] Clifford Ivar Nass and Scott Brave. *Wired for speech: How voice activates and advances the human-computer relationship.* MIT press Cambridge, 2005.

[New94] Allen Newell. *Unified Theories of Cognition.* Harvard University Press, 1994.

[NJH$^+$05] Clifford Nass, Ing-Marie Jonsson, Helen Harris, Ben Reaves, Jack Endo, Scott Brave, and Leila Takayama. Improving automotive safety by pairing driver emotion and car voice emotion. In *Proceedings of the Human Factors in Computing Systems*, New York, USA, 2005.

[NL01] Clifford Nass and Kwan Min Lee. Does computer-synthesized speech manifest personality? experimental tests of recognition, similarity-attraction, and consistency-attraction. *Journal of Experimental Psychology: Applied*, 7(3):171–181, 2001.

[NL07] Fatma Nasoz and Christine L. Lisetti. Affective user modeling for adaptive intelligent user interfaces. In *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*, number 4552 in Lecture Notes in Computer Science, pages 421–430. Springer Berlin Heidelberg, 2007.

[NSS59] Allen Newell, John C. Shaw, and Herbert A. Simon. Report on a general problem-solving program for a computer. In *Proceedings of the International Conference on Information Processing*, page 256–264. UNESCO House, Paris, 1959.

[NWR10] Hugh Nolan, Robert Whelan, and Richard B. Reilly. FASTER: Fully automated statistical thresholding for EEG artifact rejection. *Journal of Neuroscience Methods*, 192(1):152–162, September 2010.

[OK06] Klaus Oberauer and Reinhold Kliegl. A formal model of capacity limits in working memory. *Journal of Memory and Language*, 55(4):601–626, 2006.

[OTO$^+$06] Daria Osipova, Atsuko Takashima, Robert Oostenveld, Guillén Fernández, Eric Maris, and Ole Jensen. Theta and gamma oscillations predict encoding and retrieval of declarative memory. *The Journal of Neuroscience*, 26(28):7523–7531, 2006.

[PAB$^+$10] Gert Pfurtscheller, Brendan Z Allison, Günther Bauernfeind, Clemens Brunner, Teodoro Solis Escalante, Reinhold Scherer, Thorsten O Zander, Gernot Mueller-Putz, Christa Neuper, and Niels Birbaumer. The hybrid BCI. *Frontiers in neuroscience*, 4:3, 2010.

[Pau75] Grice H Paul. Logic and conversation. *Syntax and semantics*, 3:41–58, 1975.

[Pau08] Annie Pauzie. A method to assess the driver mental workload: The driving activity load index (DALI). *IET Intelligent Transport Systems*, 2(4):315, 2008.

[PBZ11] Laetitia Perre, Daisy Bertrand, and Johannes Ziegler. Literacy affects spoken language in a non-linguistic task: an ERP study. *Language Sciences*, 2:274, 2011.

[Pei07] Jonathan W. Peirce. PsychoPy—psychophysics software in python. *Journal of Neuroscience Methods*, 162(1–2):8–13, 2007.

[PH00] Tim Paek and Eric Horvitz. Grounding criterion: Toward a formal theory of grounding. Technical report, MSR Technical Report, 2000.

[PH08] Felix Putze and Hartwig Holzapfel. IslEnquirer: Social user model acquisition through network analysis and interactive learning. In *Proceedings of Spoken Language Technology Workshop*, pages 117–120, Goa, India, 2008.

[PHK$^+$13] Felix Putze, Jutta Hild, Rainer Kärgel, Christian Herff, Alexander Redmann, Jürgen Beyerer, and Tanja Schultz. Locating user attention using eye tracking and EEG for spatio-temporal event selection. In *Proceedings of the International Conference on Intelligent User Interfaces*, Santa Monica, USA, 2013.

[PHM$^+$12] Thomas Prevot, Jeffrey R. Homola, Lynne H. Martin, Joey S. Mercer, and Christopher D. Cabrall. Toward automated air traffic control—investigating a fundamental paradigm shift in human/systems interaction. *International Journal of Human-Computer Interaction*, 28(2):77–98, 2012.

[Pic00] Rosalind W. Picard. *Affective Computing.* MIT Press, 2000.

[PMDR08] Jason A. Palmer, Scott Makeig, Kenneth K. Delgado, and Bhaskar D. Rao. Newton method for the ICA mixture model. In *International Conference on Acoustics, Speech and Signal Processing*, 2008.

[PMP10] Ming-Zher Poh, Daniel J. McDuff, and Rosalind W. Picard. Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. *Optics Express*, 18(10):10762, 2010.

[PP08] Tim Paek and Roberto Pieraccini. Automating spoken dialogue management design using machine learning: An industry perspective. *Speech Communication*, 50(8–9):716–729, 2008.

[Pro11] Robert Pröpper. *JAM: Java-Based Associative Memory - Implementation, Analysis and Comparison of Memory Models for Human Computer Interaction.* Project thesis, Karlsruhe Institute of Technology, 2011.

[Pra98] Prahl, Scott. Tabulated molar extinction coefficient for hemoglobin in water, 1998.

[PRI09] Olivier Pietquin, Stéphane Rossignol, and Michel Ianotto. Training bayesian networks for realistic man-machine spoken dialogue simulation. In *Proceedings of the 1rst International Workshop on Spoken Dialogue Systems Technology*, Irsee, Germany, 2009.

[PWCS09] Anne Porbadnigk, Marek Wester, Jan-Peter Calliess, and Tanja Schultz. EEG-based speech recognition impact of temporal effects. In *Proceedings of 2nd Biosignals Conference*, page 376–381, Porto, Portugal, 2009.

[PY10] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[RDMP04] Susana Rubio, Eva Díaz, Jesús Martín, and José M. Puente. Evaluation of subjective mental workload: A comparison of SWAT, NASA-TLX, and workload profile methods. *Applied Psychology*, 53(1):61–86, 2004.

[RH97] G. Robert and J. Hockey. Compensatory control in the regulation of human performance under stress and high workload: A cognitive-energetical framework. *Biological Psychology*, 45(1–3):73–93, 1997.

[RL10] Verena Rieser and Oliver Lemon. Learning human multimodal dialogue strategies. *Natural Language Engineering*, 16(01):3–23, 2010.

[RL11] Verena Rieser and Oliver Lemon. Learning and evaluation of dialogue strategies for new applications: Empirical methods for optimization from small data sets. *Computational Linguistics*, 37(1):153–196, 2011.

[RN96] Byron Reeves and Clifford Ivar Nass. *The media equation: How people treat computers, television, and new media like real people and places*, volume xiv. Cambridge University Press, New York, USA, 1996.

[RRS06] Frank E. Ritter, Andrew L. Reifers, and Michael J. Schoelles. Lessons from defining theories of stress. In *Integrated Models of Cognitive Systems*. 2006.

[RVRAS06] Frank E. Ritter, Dirk Van Rooy, Robert S. Amant, and Kate Simpson. Providing user models direct access to interfaces: an exploratory study of a simple interface with implications for HRI and HCI. *Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 36(3):592–601, 2006.

[RY01] Frank E. Ritter and Richard Young. Embodied models as simulated users: introduction to this special issue on using cognitive models to improve interface design. *International Journal of Human-Computer Studies*, 55(1):1–14, 2001.

[S$^+$66] Saul Sternberg et al. High-speed scanning in human memory. *Science*, 153(3736):652–654, 1966.

[Sal06] Dario D. Salvucci. Modeling driver behavior in a cognitive architecture. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(2):362–380, 2006.

[SB98] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[SBB06] Holger Schultheis, Thomas Barkowsky, and Sven Bertel. LTM-c: an improved long-term memory for cognitive architectures. In *Proceedings of the Seventh International Conference on Cognitive Modeling*, page 274–279, 2006.

[SBK$^+$12] Martin Spüler, Michael Bensch, Sonja Kleih, Wolfgang Rosenstiel, Martin Bogdan, and Andrea Kübler. Online use of error-related potentials in healthy users and people with severe motor impairment increases performance of a p300-BCI. *Clinical neurophysiology: official journal of the International Federation of Clinical Neurophysiology*, 123(7):1328–1337, 2012.

[SCL$^+$11] Jyotirmay Sanghvi, Ginevra Castellano, Iolanda Leite, André Pereira, Peter W. McOwan, and Ana Paiva. Automatic analysis of affective postures and body motion to detect engagement with a game companion. In *Proceedings of the International Conference on Human-Robot Interaction*, pages 305–311, 2011.

[SF04] Angelo Sassaroli and Sergio Fantini. Comment on the modified beer–lambert law for scattering media. *Physics in Medicine and Biology*, 49(14):N255, July 2004.

[SG12] Marvin R. G. Schiller and Fernand R. Gobet. A comparison between cognitive and AI models of blackjack strategy learning. In Birte Glimm and Antonio Krüger, editors, *KI 2012: Advances in Artificial Intelligence*, number 7526 in Lecture Notes in Computer Science, pages 143–155. Springer Berlin Heidelberg, January 2012.

[SGC$^+$09] Erin Treacy Solovey, Audrey Girouard, Krysta Chauncey, Leanne M. Hirshfield, Angelo Sassaroli, Feng Zheng, Sergio Fantini, and Robert J.K. Jacob. Using fNIRS brain sensing in realistic HCI settings: Experiments and guidelines. In *Proceedings of the 22nd Symposium on User Interface Software and Technology*, page 157–166, New York, USA, 2009.

[SH10] Ethan O. Selfridge and Peter A. Heeman. Importance-driven turn-bidding for spoken dialogue systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, USA, 2010. Association for Computational Linguistics.

[SH13] Robert Speer and Catherine Havasi. ConceptNet 5: A large semantic network for relational knowledge. In Iryna Gurevych and Jungi Kim, editors, *The People's Web Meets NLP*, Theory and Applications of Natural Language Processing, pages 161–176. Springer Berlin Heidelberg, 2013.

[SKD⁺05] Paul Sauseng, Wolfgang Klimesch, Michael Doppelmayr, Thomas Pecherstorfer, Roman Freunberger, and Simon Hanslmayr. EEG alpha synchronization and functional coupling during top-down processing in a working memory task. *Human Brain Mapping*, 26(2):148–155, 2005.

[SKZ⁺07] Alois Schlögl, Claudia Keinrath, Doris Zimmermann, Reinhold Scherer, Robert Leeb, and Gert Pfurtscheller. A fully automated correction method of EOG artifacts in EEG recordings. *Clinical Neurophysiology*, 118(1):98–104, 2007.

[SL07] Holger Schultheis and Shane Lile. Extending ACT-r's memory capabilities. In *Proceedings of the Second European Cognitive Science Conference*, 2007.

[Slo96] Steven A. Sloman. The empirical case for two systems of reasoning. *Psychological Bulletin*, 119(1):3–22, 1996.

[SLY11] Frank Seide, Gang Li, and Dong Yu. Conversational speech transcription using context-dependent deep neural networks. In *Proceedings of Interspeech*, 2011.

[SNG⁺02] Jongho Shin, Shrikanth Narayanan, Laurie Gerber, Abe Kazemzadeh, and Dani Byrd. Analysis of user behavior under error conditions in spoken dialogs. In *Proceedings of Interspeech*, 2002.

[Spe27] Charles Spearman. *The abilities of man*, volume xxiii. Macmillan, Oxford, England, 1927.

[SSB09] Björn Schuller, Stefan Steidl, and Anton Batliner. The INTERSPEECH 2009 emotion challenge. In *Proceedings of 10th Annual Conference of the International Speech Communication Association*, 2009.

[SSB⁺10] Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian Müller, and Shrikanth Narayanan. The INTERSPEECH 2010 paralinguistic challenge. In *Proceedings of 11th Annual Conference of the International Speech Communication Association*, 2010.

[SSB⁺11] Björn Schuller, Stefan Steidl, Anton Batliner, Florian Schiel, and Jarek Krajewski. The INTERSPEECH 2011 speaker state challenge. In *Proceedings of 12th Annual Conference of the International Speech Communication Association*, page 3201–3204, 2011.

[SSB+12] Björn Schuller, Stefan Steidl, Anton Batliner, Elmar Nöth, Alessandro Vinciarelli, Felix Burkhardt, Rob van Son, Felix Weninger, Florian Eyben, and Tobias Bocklet. The INTERSPEECH 2012 speaker trait challenge. In *Proceedings of 13th Annual Conference of the International Speech Communication Association*, 2012.

[SSB+13] Björn Schuller, Stefan Steidl, Anton Batliner, Alessandro Vinciarelli, Klaus Scherer, Fabien Ringeval, Mohamed Chetouani, Felix Weninger, Florian Eyben, and Erik Marchi. The INTERSPEECH 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism. In *Proceedings of 14th Annual Conference of the International Speech Communication Association*, 2013.

[SSG+01] Eric H. Schumacher, Travis L. Seymour, Jennifer M. Glass, David E. Fencsik, Erick J. Lauber, David E. Kieras, and David E. Meyer. Virtually perfect time sharing in dual-task performance: Uncorking the central cognitive bottleneck. *Psychological Science*, 12(2):101–108, March 2001.

[SSMvC02] André J. Szameitat, Torsten Schubert, Karsten Müller, and D. Yves von Cramon. Localization of executive functions in dual-task performance with fMRI. *Journal of Cognitive Neuroscience*, 14(8):1184–1199, 2002.

[SSO11] Emilio Sardini, Mauro Serpelloni, and Marco Ometto. Multi-parameters wireless shirt for physiological monitoring. In *Proceedings of the International Workshop on Medical Measurements and Applications Proceedings*, 2011.

[SSWY05] J. Schatzmann, M.N. Stuttle, K. Weilhammer, and S. Young. Effects of the user model on simulation-based learning of dialogue strategies. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding*, pages 220–225, 2005.

[SSYH12] Stoyanchev Stoyanchev, Philipp Salletmayr, Jingbo Yang, and Julia Hirschberg. Localized detection of speech recognition errors. In *Proceedings of the Spoken Language Technology Workshop*, pages 25–30, 2012.

[ST08] Dario D. Salvucci and Niels A. Taatgen. Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review*, 115(1):101–130, 2008.

[STW+07] Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Proceedings of The Conference of the North American Chapter of the Association for Computational Linguistics*, Stroudsburg, USA, 2007.

[SvLG+12] Philipp M. Scholl, Kristof van Laerhoven, Dawud Gordon, Markus Scholz, and Matthias Berning. jNode: A sensor network platform that supports distributed inertial kinematic monitoring. In *Proceedings of the Ninth International Conference on Networked Sensing Systems*, 2012.

[SWF05] Miriam Spering, Daniel Wagener, and Joachim Funke. The role of emotions in complex problem-solving. *Cognition and Emotion*, 19(8):1252–1261, 2005.

[SWMP00] Gerwin Schalk, Jonathan R Wolpaw, Dennis J McFarland, and Gert Pfurtscheller. EEG-based communication: presence of an error potential. *Clinical Neurophysiology*, 111(12):2138–2144, 2000.

[SWSY06] Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, 21(02):97–126, 2006.

[SZBM05] Dario D. Salvucci, Mark Zuber, Ekaterina Beregovaia, and Daniel Markley. Distract-r: rapid prototyping and evaluation of in-vehicle interfaces. In *Proceedings of the Conference on Human Factors in Computing Systems*, page 581–589, New York, USA, 2005.

[SZH+08] Angelo Sassaroli, Feng Zheng, Leanne M. Hirshfield, Audrey Girouard, Erin Treacy Solovey, Robert J. K. Jacob, and Sergio Fantini. Discrimination of mental workload levels in human subjects with functional near-infrared spectroscopy. *Journal of Innovative Optical Health Sciences*, 01(02):227–237, 2008.

[Tho82] David J. Thomson. Spectrum estimation and harmonic analysis. *Proceedings of the IEEE*, 70(9):1055–1096, 1982.

[THS$^+$07] Anil M. Tuladhar, Niels ter Huurne, Jan-Mathijs Schoffelen, Eric Maris, Robert Oostenveld, and Ole Jensen. Parieto-occipital sources account for the increase in alpha activity with working memory load. *Human Brain Mapping*, 28(8):785–792, 2007.

[TŢM08] Adriana Tapus, Cristian Ţăpuş, and Maja J. Matarić. User—robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy. *Intelligent Service Robotics*, 1(2):169–183, 2008.

[Tur50] Alan M Turing. Computing machinery and intelligence. *Mind*, page 433–460, 1950.

[TWG$^+$14] Dominic Telaar, Michael Wand, Dirk Gehrig, Felix Putze, Christoph Amma, Dominic Heger, Ngoc Thang Vu, Mark Erhardt, Tim Schlippe, Matthias Janke, Christian Herff, and Tanja Schultz. BioKIT - real-time decoder for biosignal processing. In *Proceedings of 15th Annual Conference of the International Speech Communication Association*, Singapore, 2014.

[vE11] Dominik van Engelen. *Attention Drivers!* PhD thesis, RWTH Aachen, 2011.

[VKA$^+$11] Sudha Velusamy, Hariprasad Kannan, Balasubramanian Anand, Anshul Sharma, and Bilva Navathe. A method to infer emotions from facial action units. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 2028–2031, 2011.

[VKS10] Ngoc Thang Vu, Franziska Kraus, and Tanja Schultz. Multilingual a-stabil: A new confidence score for multilingual unsupervised training. In *Proceedings of the Spoken Language Technology Workshop*, pages 183–188, 2010.

[VL11] Akos Vetek and Saija Lemmelä. Could a dialog save your life?: analyzing the effects of speech interaction strategies while driving. In *Proceedings of the 13th international conference on multimodal interfaces*, New York, USA, 2011.

[VS12] Chi Vi and Sriram Subramanian. Detecting error-related negativity for interaction design. In *Proceedings of the Conference on Human Factors in Computing Systems*, New York, USA, 2012.

[VSL88] Scott R. Vrana, Ellen L. Spence, and Peter J. Lang. The startle probe response: A new measure of emotion? *Journal of abnormal psychology*, 97(4):487, 1988.

[vWSG84] Wim van Winsun, Joseph Sergeant, and Reint Geuze. The functional significance of event-related desynchronization of alpha rhythm in attentional and activating tasks. *Electroencephalography and Clinical Neurophysiology*, 58(6):519–524, December 1984.

[WA11] Matthew M. Walsh and John R. Anderson. Learning from delayed feedback: neural responses in temporal credit assignment. *Cognitive, Affective, & Behavioral Neuroscience*, 11(2):131–143, 2011.

[Wat89] Christopher John Cornish Hellaby Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, 1989.

[WBG05] Torsten Wilhelm, Hans-Joachim Böhme, and Horst-Michael Gross. Classification of face images for gender, age, facial expression, and identity. In *Proceedings of the International Conference on Artificial Neural Networks*, page 569–574, Warsaw, Poland, 2005. Springer.

[WBM+02] Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, and Theresa M. Vaughan. Brain–computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791, 2002.

[Wel67] Peter D. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967.

[WGM+09] Maria Wolters, Kallirroi Georgila, Johanna D. Moore, Robert H. Logie, Sarah E. MacPherson, and Matthew Watson. Reducing working memory load in spoken dialogue systems. *Interacting with Computers*, 21(4):276–287, 2009.

[WHW+12] Ziheng Wang, Ryan M. Hope, Zuoguan Wang, Qiang Ji, and Wayne D. Gray. Cross-subject workload classification with a hierarchical bayes model. *NeuroImage*, 59(1):64–69, 2012.

[Wic08] Christopher D. Wickens. Multiple resources and mental workload. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(3):449–455, 2008.

[WLR00] Glenn F. Wilson, Jared D. Lambert, and Chris A. Russell. Performance enhancement with real-time physiologically controlled adaptive aiding. *Proc. of the Human Factors and Ergonomics Society Annual Meeting*, 44(13):61–64, 2000.

[WMFG08] Christopher Whalen, Edward L Maclin, Monica Fabiani, and Gabriele Gratton. Validation of a method for coregistering scalp recording locations with 3d structural MR images. *Human brain mapping*, 29(11):1288–1301, 2008.

[WMR00] Sabine Weiss, Horst M. Müller, and Peter Rappelsberger. Theta synchronization predicts efficient memory encoding of concrete and abstract nouns. *Neuroreport*, 11(11):2357–2361, August 2000.

[WR07] Glenn F. Wilson and Christopher A. Russell. Performance enhancement in an uninhabited air vehicle task using psychophysiologically determined adaptive aiding. *Human Factors*, 49(6):1005–1018, 2007.

[WWC$^+$03] Martin Wolf, Ursula Wolf, Jee H. Choi, Vladislav Toronov, L. Adelina Paunescu, Antonios Michalos, and Enrico Gratton. Fast cerebral functional signal in the 100-ms range detected in the visual cortex by frequency-domain near-infrared spectrophotometry. *Psychophysiology*, 40(4):521–528, 2003.

[WWo99] Jason Weston, Chris Watkins, and others. Support vector machines for multi-class pattern recognition. In *ESANN*, volume 99, page 219–224, 1999.

[WY07] Jason D. Williams and Steve Young. Scaling POMDPs for spoken dialog management. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2116–2129, 2007.

[YRM$^+$12] Yan Yang, Bryan Reimer, Bruce Mehler, Alan Wong, and Mike McDonald. Exploring differences in the impact of auditory and visual demands on driver behavior. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, page 173–177, New York, USA, 2012.

[ZK11] Thorsten O. Zander and Christian Kothe. Towards passive brain–computer interfaces: applying brain–computer interface technology to human–machine systems in general. *Journal of Neural Engineering*, 8(2):025005, April 2011.

[ZPRH09] Zhihong Zeng, Maja Pantic, Glenn I. Roisman, and Thomas S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009.

[ZSHM10] Alexander Zgorzelski, Alexander Schmitt, Tobias Heinroth, and Wolfgang Minker. Repair strategies on trial: Which error recovery do users like best? In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[ZTL$^+$07] Zhihong Zeng, Jilin Tu, Ming Liu, Thomas S. Huang, Brian Pianfetti, Dan Roth, and Stephen Levinson. Audio-visual affect recognition. *IEEE Transactions on Multimedia*, 9(2):424–428, 2007.

Chapter 6

Addendum

6.1 List of Supervised Student Theses

The following four tables present a list of all student theses (Studienarbeit, Diplomarbeit, Bachelorarbeit, or Masterarbeit) that were (co-)supervised by the author in the context of this work.

Student	**Year**	**Thesis Title**
Markus Müller	2012	Implementierung und Evaluation session-invarianter EEG-basierter Workload Erkennung
Christian Waldkirch	2013	Entwicklung von Strategien zur Fehlererkennung und Fehlerbehandlung fr ein gestenbasiertes Eingabesystem
Steven Colling	2013	Recognition of Player Deficits using Game-Triggered Intervention and Bayesian Networks
Rainer Kärgel	2014	Transfer Learning for EEG-based workload classification
David Kaufman	2014	Temporale Ereignislokalisation mit EEG in einer blickbasierten Aufmerksamkeitsaufgabe

Table 6.1 – List of all supervised “Masterarbeiten”.

Student	Year	Thesis Title
Dominic Heger	2009	Towards Automatic Recognition of Personality for Human-Machine Interaction
Jan-Philip Jarvis	2011	Multimodal Person Independent Recognition of Driver Mental Workload
Daniel Reich	2011	A Real-Time Speech Command Detector for a Smart Control Room
Robert Pröpper	2011	Adaption of cognitive models to dynamically changing mental workload
Christian Herff	2011	Speech related activations in the brain: Differentiating between speaking modes with fNIRS
Jeremias Engelmann	2012	Untersuchung des Zusammenhangs von EEG basierten ereigniskorrelierten Potentialen und physischer und kognitiver Fitness bei jungen Studierenden
Ying Wei	2012	Experimentelle Induktion und biosignalbasierte Erkennung Positiver und Negativer Stimmungen beim Autofahren
Sebastian Hesslinger	2013	Hybrid NIRS-EEG based classification of auditory and visual perception processes
Christian Harnisch	2014	User Simulation in Cognitive Dialog Systems: An Application of the Multiple Resource Model

Table 6.2 – List of all supervised “Diplomarbeiten”.

Student	Year	Thesis Title
Ivana Kajic	2011	Dynamic Modeling of EGG Patterns for Different Perceptual and Cognitive Tasks using Hidden Markov Models
Steven Colling	2011	Appraisal-basierte Modellierung von Emotionen in der Interaktion mit einem virtuellen Beifahrer
Joscha Borne	2012	Kognitive Modellierung komplexer strategischer Entscheidungsprozesse mittels EEG und Reinforcement Learning
Johannes Meyer	2012	Kognitive Modellierung komplexer strategischer Entscheidungsprozesse mittels EEG und Reinforcement Learning
Lucas Bechberger	2012	Modeling human memory performance under influence of cognitive workload
Sebastian Mendez	2012	Transfer entropy for extracranial EEG analysis with TRENTOOL in a visual cued motor task
Matthias Sazinger	2013	Entwicklung und Evaluation von Benutzerschnittstellen zur Unterstützung bei einer komplexen Aufgabe unter verschiedenen Workload-Bedingungen
Patrick Klinowski	2013	Designing a workload adaptive dialog system with flexible initiative
Michael Axtmann	2013	Online detection of error related potentials using electroencephalography and spatial filtering
Kalin Katev	2013	Comparison of individualized Reinforcement Learning models with real-life subjects for a complex learning task
Simone di Stefano	2013	Classification of perceptual modality and processing code in multimodal cognition processes using EEG
Vincent Beckert	2013	Adaption of a Cognitive Decision-Making Model through the Application of Workload Recognition

Table 6.3 – List of all supervised “Bachelorarbeiten”.

Student	Year	Thesis Title
Rikard Öxler	2009	Planung und Aufbau eines Fahrsimulators zur Untersuchung kognitiver Dialogstrategien (Schwerpunkt Software)
Frieder Reinhold	2009	Planung und Aufbau eines Fahrsimulators zur Untersuchung kognitiver Dialogstrategien (Schwerpunkt Hardware)
Jan-Philip Jarvis	2010	Multimodale biosignalbasierte Workloaderkennung im Fahrzeug
Florian Krupicka	2010	Selection and Implementation of a Cognitive Memory Model
Daniel Reich	2010	Integration of a Recognizer in a Multimodal Smart Control Room
Robert Pröpper	2011	JAM: Java-Based Associative Memory – Implementation, Analysis and Comparison of Memory Models for Human Computer Interaction
Jochen Bieler	2011	Analyse und Wirkung von Musik anhand von EEG-Daten
Sebastian Hesslinger	2011	Analyzing n-back EEG Data with Auditory and Visual Stimuli using ICA-based Spatial Filtering
Dorothea Kintz	2013	Datenanalyse zur Vorbereitung einer Workload-Komponente für ACT-R
Dimitri Majarle	2013	Klassifizierung EEG basierter ereigniskorrelierter Potentiale aus visuellen und auditiven Aufgaben
Lena Zwar	2013	Confidence-based Methods to Improve Reliability and Fusion Performance of EEG-based Workload Recognition System

Table 6.4 – List of all supervised "Studienarbeiten".

6.2 List of Publications

The following bibliography lists all publications of which the author is author or co-author.

Publications of Felix Putze

[1] Johannes Singler, Peter Sanders, and Felix Putze. MCSTL: The multi-core standard template library. In *Euro-Par 2007 Parallel Processing*, number 4641 in Lecture Notes in Computer Science, pages 682–694. Springer Berlin Heidelberg, January 2007.

[2] Felix Putze, Peter Sanders, and Johannes Singler. Cache-, hash- and space-efficient bloom filters. In *Experimental Algorithms*, number 4525 in Lecture Notes in Computer Science, pages 108–121. Springer Berlin Heidelberg, January 2007.

[3] Felix Putze and Hartwig Holzapfel. IslEnquirer: Social user model acquisition through network analysis and interactive learning. In *Proceedings of Spoken Language Technology Workshop*, pages 117–120, Goa, India, 2008.

[4] Felix Putze and Tanja Schultz. Cognitive memory modeling for interactive systems in dynamic environments. In *International Workshop Series on Spoken Dialogue Systems Technology*, Irsee, Germany, 2009.

[5] Felix Putze and Tanja Schultz. Towards cognitive dialog systems. In *1. Fachtagung Biophysiologische Interfaces*, Berlin, Germany, 2009.

[6] Dominic Heger, Felix Putze, and Tanja Schultz. Online workload recognition from EEG data during cognitive tests and human-computer interaction. In *Proceedings of 33rd Annual German Conference on Artificial Intelligence 2010*, Karlsruhe, Germany, 2010.

[7] Felix Putze and Tanja Schultz. Utterance selection for speech acts in a cognitive tourguide scenario. In *Proceedings of 11th Annual Conference of the International Speech Communication Association*, page 3014–3017, Makuhari, Japan, 2010.

[8] Dominic Heger, Felix Putze, and Tanja Schultz. An adaptive information system for an empathic robot using EEG data. In *Social Robotics*, page 151–160. Springer, 2010.

[9] Felix Putze, Jan-Philipp Jarvis, and Tanja Schultz. Multimodal recognition of cognitive workload for multitasking in the car. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*, pages 3748–3751, Istanbul, Turkey, 2010.

[10] Daniel Reich, Felix Putze, Dominic Heger, Joris Ijsselmuiden, Rainer Stiefelhagen, and Tanja Schultz. A real-time speech command detector for a smart control room. In *Proceedings of 12th Annual Conference of the International Speech Communication Association*, page 2641–2644, 2011.

[11] Jan Jarvis, Felix Putze, Dominic Heger, and Tanja Schultz. Multimodal person independent recognition of workload related biosignal patterns. In *Proceedings of the 13th International Conference on Multimodal Interfaces*, ICMI '11, page 205–208, Alicante, Spain, 2011.

[12] Dominic Heger, Felix Putze, and Tanja Schultz. Online recognition of facial actions for natural EEG-based BCI applications. In *Affective Computing and Intelligent Interaction*, Lecture Notes in Computer Science, pages 436–446. Springer Berlin Heidelberg, January 2011.

[13] Robert Pröpper, Felix Putze, and Tanja Schultz. JAM: Java-based associative memory. In *Proceedings of the Paralinguistic Information and its Integration in Spoken Dialogue Systems Workshop*, pages 143–155, January 2011.

[14] Janina Krell, Sabrina Benzinger, Klaus Boes, Jeremias Engelmann, Dominic Heger, Felix Putze, Tanja Schultz, and Alexander Stahn. Physical activity, brain function and cognitive performance in young adults-a cross-sectional study. In *Medicine and Science in Sports and Exercise*, volume 44, Philadelphia, USA, 2012.

[15] Christian Herff, Felix Putze, Dominic Heger, Cuntai Guan, and Tanja Schultz. Speaking mode recognition from functional near infrared spectroscopy. In *Proceedings of Annual International Conference of the Engineering in Medicine and Biology Society (EMBC)*, page 1715–1718, San Diego, USA, 2012.

[16] Christian Herff, Dominic Heger, Felix Putze, Cuntai Guan, and Tanja Schultz. Cross-subject classification of speaking modes using fNIRS. In *Proceedings of the 19th International Conference on Neural Information Processing*, page 417–424, Doha, Quatar, 2012.

[17] Felix Putze and Tanja Schultz. Cognitive dialog systems for dynamic environments: Progress and challenges. In *Digital Signal Processing for In-Vehicle Systems and Safety*, page 133–143. Springer, 2012.

[18] Dominic Heger, Reinhard Mutter, Christian Herff, Felix Putze, and Tanja Schultz. Continuous recognition of affective states by functional near infrared spectroscopy signals. *Brain-Computer Interfaces*, 1(2):832–837, 2013.

[19] Dominic Heger, Felix Putze, Christoph Amma, Michael Wand, Igor Plotkin, Thomas Wielatt, and Tanja Schultz. BiosignalsStudio: a flexible framework for biosignal capturing and processing. In *Proceedings of the 33rd Annual German Conference on Artificial Intelligence*, Karlsruhe, Germany, 2010.

[20] Felix Putze, Daniel V. Holt, Joachim Funke, and Tanja Schultz. Combining cognitive modeling and EEG to predict user behavior in a search task. In *Proceedings of the International Conference on Cognitive Modeling*, page 303, Berlin, Germany, 2012.

[21] Felix Putze, Markus Müller, Dominic Heger, and Tanja Schultz. Session-independent EEG-based workload recognition. In *Proceedings of 6th Biosignals Conference*, pages 360–363, Barcelona, Spain, 2013.

[22] Felix Putze, Dominic Heger, Markus Müller, Christian Waldkirch, Yves Chassein, Ivana Kajic, and Tanja Schultz. Profiling arousal in response to complex stimuli using biosignals. In *Proceedings of 6th Biosignals Conference*, pages 347–350, Barcelona, Spain, 2013.

[23] Felix Putze, Jutta Hild, Rainer Kärgel, Christian Herff, Alexander Redmann, Jürgen Beyerer, and Tanja Schultz. Locating user attention using eye tracking and EEG for spatio-temporal event selection. In *Proceedings of the International Conference on Intelligent User Interfaces*, Santa Monica, USA, 2013.

[24] Felix Putze, Dominic Heger, and Tanja Schultz. Reliable subject-adapted recognition of EEG error potentials using limited calibration data. In *6th International Conference on Neural Engineering*, San Diego, USA, 2013.

[25] Tanja Schultz, Christoph Amma, Michael Wand, Dominic Heger, and Felix Putze. Biosignale-basierte mensch-maschine schnittstellen. *at–Automatisierungstechnik*, 61(11):760–769, 2013.

[26] Christian Herff, Dominic Heger, Felix Putze, Cuntai Guan, and Tanja Schultz. Self-paced BCI with NIRS based on speech activity. In *Proceedings of 5th International BCI Meeting*, Asilomar, USA, 2013.

[27] Dominic Heger, Christian Herff, Felix Putze, and Tanja Schultz. Towards biometric person identification using fNIRS. In *Proceedings of 5th International BCI Meeting*, Asilomar, USA, 2013.

[28] Dominic Telaar, Michael Wand, Dirk Gehrig, Felix Putze, Christoph Amma, Dominic Heger, Ngoc Thang Vu, Mark Erhardt, Tim Schlippe, Matthias Janke, Christian Herff, and Tanja Schultz. BioKIT - real-time decoder for biosignal processing. In *Proceedings of 15th Annual Conference of the International Speech Communication Association*, Singapore, 2014.

[29] Felix Putze and Tanja Schultz. Investigating intrusiveness of workload adaptation. In *Proceedings of International Conference on Multimodal Interfaces*, Istanbul, Turkey, 2014.

[30] Christian Herff, Dominic Heger, Ole Fortmann, Johannes Hennrich, Felix Putze, and Tanja Schultz. Mental workload during n-back task—quantified in the prefrontal cortex using fNIRS. *Frontiers in Human Neuroscience*, 7(00935), 2014.

[31] Felix Putze and Tanja Schultz. Adaptive cognitive technical systems. *Journal of Neuroscience Methods*, 234, July 2014.

[32] Felix Putze, Daniel V. Holt, Tanja Schultz, and Joachim Funke. Model-based identification of EEG markers for learning opportunities in an associative learning task with delayed feedback. In *Proceedings of 24th International Conference on Artificial Neural Networks*, Hamburg, Germany, 2014.