User Modeling for Adaptation of Cognitive Systems

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
des Fachbereich 3
der Universität Bremen

vorgelegte

Dissertation

von

Mazen Salous

Tag des Kolloquiums:	04.06.2021
Erste Gutachterin:	Prof. Dr.-Ing. Tanja Schultz
Zweite Gutachterin:	Prof. Dr. rer. nat. Bettina von Helversen
Dritter Gutachter:	Dr.-Ing. Felix Putze

Acknowledgment

I would like to thank my lovely family: parents, wife, kids and sisters for motivating and supporting me to achieve my PhD thesis. I am very happy that my parents had gotten the visa in 2014 to travel to Germany, so they have the chance now to join me this fantastic moment. My greetings to my lovely sisters, who live abroad, but they always motivate me for achievements. A special thank to my lovely wife, who prepared for me an optimal home-office working atmosphere during this hard corona-time.

I am very grateful to my supervisors, Prof. Dr.-Ing. Tanja Schultz and Dr.-Ing. Felix Putze. They had been intensively supporting me during the thesis writing. They trusted my qualifications and guided me to learn and improve. They encouraged me for achievements and publications, and positively managed my spotlight scientific discussions. I learned a lot and enriched my expertise in cognitive modeling and machine learning fields by their professional and friendly support.

Finally, I would like to thank all my colleagues in the Cognitive Systems Lab (CSL) at the university of Bremen. Besides the scientific cooperation and discussions, we have a very comfort working atmosphere, just like a family. I would like to congratulate all my colleagues for their achievements, and wish them all the best success in their coming research.

Abstract

Computer systems have been continuously evolving since the first computer invented in the 1950's up to the revolution of smart devices, which we are witnessing nowadays. Consequently, the number of computer users has been exponentially increasing, from only few experts to hundreds of millions users: novices, intermediates, and experts.

Human Computer Interaction (HCI) has been widely discussed for improving the interaction between a computer system and its users. Interdisciplinary research evolved for integrating psychological studies with HCI studies to model cognitive skills during HCI sessions, and thus, to design an effective User Interface (UI). Due to the high dynamic nature of HCI sessions, a dynamic UI adaptation is required when user performance is impaired.

User performance impairment in an HCI session should be detected "online", i.e. during the system use, for an appropriate UI adaptation; It is valuable to detect the reason which caused the performance impairment during an HCI session, so-called HCI obstacle, because different UI adaptations are required to compensate for different HCI obstacles.

Different HCI obstacles impair several human processes, namely perception and cognition processes. Consequently, several human processes can be impaired during an HCI session. Human processes can be tracked by recording appropriate multimodal data during an HCI session: brain activity data to track the cognition process and behavioral data, depicted by encoded user actions, to track the user behaviour.

Modeling of multimodal HCI obstacles is very important because there is no "silver bullet" UI adaptation which could be activated by default. In other words, while different HCI obstacles, e.g. memory-based and visual obstacles, impair the user HCI performance, different UI adaptation mechanisms will suit each individual HCI obstacle, because an UI adaptation should appropriately compensate for the impaired human process. Moreover, a good UI adaptation mechanism for a specific HCI obstacle can be detrimental if applied for other HCI obstacles.

In this thesis, a novel user modeling based cognitive adaptive system is proposed. The adaptive system dynamically models memory-based and visual HCI obstacles during system use, and accordingly applies the suitable UI adaptation mechanism for each detected HCI obstacle. Appropriate machine learning models are used for multimodal HCI obstacles detection. The multimodal obstacle detectors outputs are passed to an overarching probabilistic model to decide for the most suitable UI adaptation mechanism.

The proposed approach is dynamic in consecutive HCI sessions, i.e. it treats not only persistent HCI obstacles which remain impairing the user performance in consecutive HCI sessions, but also volatile HCI obstacles which suddenly appear or disappear in the HCI sessions. Moreover, the model is dynamic in case of wrongly decided UI adaptation for an HCI session, where it recovers in the subsequent sessions.

The approach is systemically evaluated through data collected from many user studies, and the experimental results show that our approach: 1) models the HCI obstacles well, where it can simulate the user behaviour under different conditions, 2) detects multimodal HCI obstacles in consecutive sessions, and 3) dynamically learns from consecutive HCI sessions to accurately adapt the UI.

Deutsche Zusammenfassung

Computersysteme haben sich seit dem ersten erfundenen Computer in den 1950er Jahren bis zur heutigen Revolution der intelligenten-Geräte kontinuierlich weiterentwickelt. Infolgedessen hat die Anzahl der Computerbenutzer exponentiell erhöht, von nur wenigen Experten bis Hunderte Millionen Benutzer: Anfänger, Fortgeschrittene und Experten.

Mensch-Maschine-Interaktion (Human Computer Interaction (HCI)) wurde ausführlich diskutiert, um die Interaktion zwischen einem Computersystem und seinen Benutzern zu verbesseren. Interdisziplinäre Forschung wurde entwickelt, um psychologische Studien in HCI-Studien zu integrieren, um kognitive Fähigkeiten während HCI-Sessions zu modellieren und damit eine effektive Benutzeroberfläche (User Interface (UI)) zu entwerfen. Aufgrund der hohen Dynamik von HCI-Sessions ist eine dynamische Anpassung der UI erforderlich, wenn die Benutzerleistung beeinträchtigt wird.

Eine Beeinträchtigung der Benutzerleistung in einer HCI-Sitzung sollte "online" erkannt werden, d.h. Während der Systemnutzung, um eine geeignete Anpassung der Benutzeroberfläche zu verfügen; Es ist wertvoll, den Grund für eine Beeinträchtigung der Benutzerleistung während einer HCI-Session zu erkennen, die eine Leistungsbeeinträchtigung verursacht hat, ein sogenanntes HCI-Hindernis, weil unterschiedliche UI-Anpassungen erforderlich sind, um unterschiedliche HCI-Hindernisse zu kompensieren.

Verschiedene HCI-Hindernisse beeinträchtigen mehrere menschliche Prozesse, nämlich Wahrnehmungs- und Erkenntnisprozesse. Folglich können mehrere menschliche Prozesse während einer HCI-Session beeinträchtigt werden. Menschliche Prozesse können verfolgt werden, indem passende multimodale Daten aufgezeichnet werden: Gehirnaktivitätsdaten zur Verfolgung des Erkennungsprozesses und Verhaltensdaten, die durch codierte Benutzeraktionen, zur Verfolgung des Benutzerverhaltens dargestellt werden.

Die Modellierung multimodaler HCI-Hindernisse ist sehr wichtig, da es keine "Silver Bullet" UI-Adaption gibt, die standardmäßig aktiviert werden könnte. Mit anderen Worten, obwohl beeinträchtigen verschiedene HCI-Hindernisse

(z.B. memory-basierte und visuelle Hindernisse) die HCI-Leistung des Benutzers, unterschiedliche UI-Anpassungsmechanismen passen zu jedem einzelnen HCI-Hindernis, da die UI-Anpassung den beeinträchtigten menschlichen Prozess angemessen kompensieren sollte. Darüber hinaus kann ein guter UI-Anpassungsmechanismus für ein bestimmtes HCI-Hindernis nachteilig bei Anwendung auf andere HCI-Hindernisse.

In dieser Arbeit ist ein neuartiges kognitives adaptives System vorgeschlagen. Das System basiert auf Benutzermodellen, dynamisch modiliert memory-basierte und visuelle HCI-Hindernisse während der Systemnutzung und dementsprechend anwendet den geeigneten UI-Anpassungsmechanismus für jedes erkannte HCI-Hindernis. Geeignete Machinelearning-Modelle werden zur Erkennung multimodaler HCI-Hindernisse verwendet. Die Ausgänge der multimodalen Hindernisdetektoren werden an ein übergreifendes Wahrscheinlichkeitsmodell übergeben, um den am besten geeigneten Anpassungsmechanismus für die UI zu bestimmen.

Unser Ansatz ist in aufeinanderfolgenden HCI-Sessions dynamisch, d.h. er behandelt nicht nur anhaltende HCI-Hindernisse, die die Benutzerleistung in aufeinanderfolgenden HCI-Sessions weiterhin beeinträchtigen, sondern auch flüchtige HCI-Hindernisse, die in den HCI-Sessions plötzlich auftreten oder verschwinden.

Der Ansatz wird systematisch anhand von Daten aus vielen Benutzerstudien bewertet. Die experimentellen Ergebnisse zeigen, dass unser Ansatz: 1) die HCI-Hindernisse gut modelliert, wobei das Benutzerverhalten unter verschiedenen Bedingungen simuliert werden kann, 2) multimodale HCI-Hindernisse in aufeinanderfolgenden HCI-Sessions erkennt und 3) dynamisch aus aufeinanderfolgenden HCI-Sessions lernt, um die UI genau anzupassen.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we introduce the thesis research story and main contributions. Research challenges will be discussed and treated through all the next chapters.

Computer systems have been continuously evolving since the first invented computer in 1950s. Computer's evolution included milestones in hardware (IBM Mainframe in 1970s), software (Operating Systems in 1980s), internet (in 1990s) and social-computing (Facebook, Twitter etc in 2000s), Whitworth and Ahmad (2013). More recently, we have been witnessing in 2010s a revolution in the so-called smart devices (smartphones, smart TVs, smart watches, wearable, virtual reality VR, augmented reality AR, etc), which can be connected as an interconnected network of objects, so-called Internet of things IoT (Silverio-Fernández et al. (2018)), see Figure 1.1.

Consequently, the number of users has been exponentially increasing: from only very few professional experts, to hundreds of millions users of social-computing systems (Whitworth and Ahmad (2013)) and smart devices (Silverio-Fernández et al. (2018)). Thus, the interaction between a modern computer system and its users should suit the large variety of users: young or elderly users, experts or beginners. This motivates researchers for: 1) Modeling the Human Computer Interaction (HCI) based on human cognitive skills (Card et al. (1983)). 2) Observing the so-called *User Experience* during HCI tasks (Kuniavsky (2003)). 3) Introducing *User Modeling* concepts, e.g. Fischer (2001), for building up and modifying a conceptual understanding of the

user. 4) Eventually, inventing cognitive adaptive user interfaces (*Cognitive Adaptive UIs*) based on tracking user cognitive physiological data such as audio data(O'sullivan (2012), certified as a patent) and tactile data (Miller (2020), also certified as a patent).

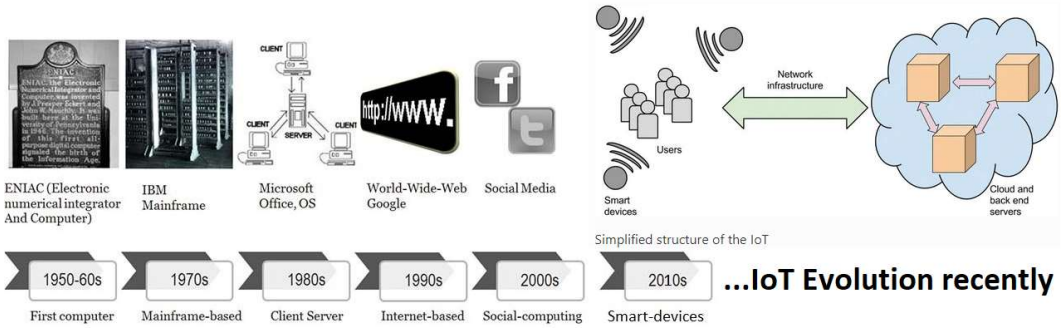


Figure 1.1 – Computer's evolution history (1950s-2000s, Whitworth and Ahmad (2013). Copyright: CC-Att-SA-3) and smart devices & IoT evolution (2010s-present, Silverio-Fernández et al. (2018)).

All theses research bodies are interleaved (HCI, *User Experience*, *User Modeling* and *Cognitive Adaptive UIs*), where researchers have in common the target of improving the *interaction* between a computer system and its users, i.e improving the HCI sessions of the said computer system. However, an HCI session can be impeded by various *interaction obstacles*, impacting a user's perception or cognition. For example, a memory-based obstacle (short: MEMOBS) caused by a secondary task memory load which impairs the human cognition process during the HCI session, or a visual obstacle (short: VISOBS) on the *User Interface* (UI) impairing the human perception process during the HCI session.

We believe that human cognitive and perception processes can be analyzed during HCI sessions to allow an *automatic UI adaptation* by detecting and compensating potential perceptive and cognitive *interaction obstacles*. That is, individual users in specific situations may suffer from different interaction obstacles while interacting with computer systems, such as memory-based or visual obstacles. Moreover, such obstacles can be either *persistent* or *volatile*; A *Persistent obstacle* in HCI exits through consecutive HCI sessions, while a *volatile obstacle* can suddenly appear or disappear in HCI sessions. Online detection of different interaction obstacles - for an individual user - enables the system to adapt the UI accordingly to tackle the expected impairment in the interaction performance.

In this thesis, we contribute to the HCI evolution in terms of *User Modeling for Adaptation of Cognitive Systems*. That is, we introduce a novel automatic user modeling approach which:

1. detects different *HCI obstacles* from observed HCI multimodal data
2. adapts the UI accordingly by selecting the most appropriate *UI adaptation* mechanism given those detected *HCI obstacles* from (1)
3. continuously checks that applied *UI adaptation* after each consecutive HCI session, and consequently consolidates the *UI adaptation*, in case of *persistent HCI obstacle* is still existing, or revert & correct it, in case of *volatile HCI obstacle* suddenly disappeared or non-optimal *UI adaptation* was decided in the previous HCI session.

In other words, we introduce an *Online Cognitive Adaptive Model* which follows potential dynamic situations of an HCI application. To sum up, our model has two main processes (sub-models): *Interaction Obstacles Detection* and *Automatic UI Support*. Both processes are connected in a closed-loop fashion to ensure a continuous support for the involved user, see Figure 1.5 for the general architecture, and follow Figure 1.6 to see those two main processes highlighted in that architecture.

After this general introduction, the structure of this chapter is organized as follows: In Section 1.1, we introduce the HCI application, together with the discussed HCI obstacles and UI adaptations, which will be used in the remaining chapters of the thesis. The general architecture is illustrated and described in Section 1.2. In Section 1.3, we highlight each sub-model in that general architecture, describe it briefly and reference to the corresponding chapter which discusses the details. After that, we list the main contributions of our thesis in Section 1.4. Finally, in Section 1.5, we summarize and introduce briefly the structure of the thesis.

1.1 HCI Application, Obstacles and UI Adaptations

In this section, we introduce the HCI exemplary application, HCI obstacles and their corresponding UI adaptations mechanisms, which will be used in the next chapters for data collection (Chapter 3), obstacles detection (Chapter 4) and UI adaptations (Chapter 5). We mention psychological (interdisciplinary) arguments for each choice.

1.1.1 HCI Exemplary Application

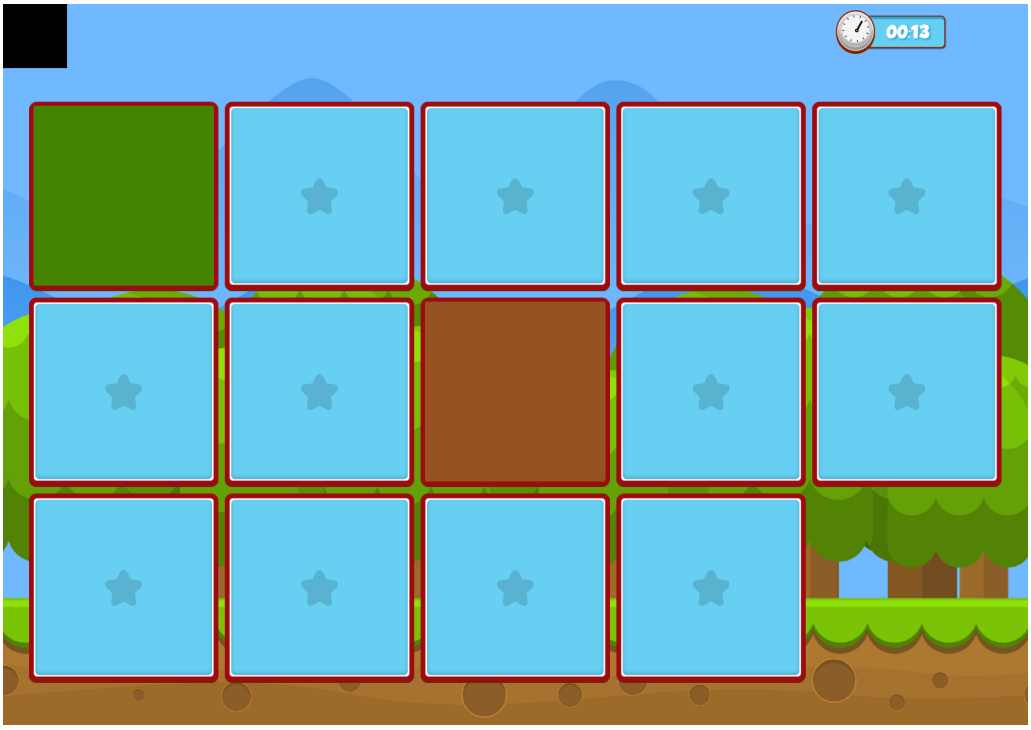


Figure 1.2 – Matching pairs game as an exemplary HCI application

We decided to use the well-established matching pairs game as an exemplary HCI application, because it involves visual perception of the UI, working memory retrieval and encoding (of both spatial and symbolic information), as well as planning and decision making, which all occur also in many other HCI tasks, such as machine or vehicle operation, decision support systems, or learning tools. We chose to implement the HCI task as a game as these generate a level of intrinsic motivation and set a natural goal for the user, Gámez et al. (2009).
Matching pairs game (also known as memory game) is a classical widely familiar family game. In this game, the player must find pairs of cards which show the same picture, always revealing two cards from a board of face-down cards in each game turn. Pairs which are revealed successfully are removed and the goal of this single-player variant is to clear the board with as few turns as possible. Figure 1.2 shows the graphical UI of the game. After a touch on a card, it is turned around and becomes visible. When the second card is revealed, both cards remain visible for 300 ms (compatible with timing parameters in the "Psychology of HCI model" Card et al. (1983)) and are

then turned back face-down. During this time, no new cards can be revealed. Card pictures are represented by uniform colors. We made this choice to avoid semantic correlates that make memorization of some cards easier or more difficult than others. The chosen colors where: red, orange, green, turquoise, yellow, blue, and pink (All participating subjects (Chapter 3) were healthy, with no color-blindness, i.e. no red-green color vision deficiency). The implementation of the task was based on an open source memory game[1] which we customized with experiment-specific cards as well as the ability to toggle different interaction obstacles and adaptation mechanisms (see below).

1.1.2 HCI Obstacles

Various human cognitive processes can be impaired during HCI tasks. We aim at automatic detection of such HCI obstacles, and later on, at automatic plausible compensation (plausible UI adaptations). In this section, we introduce and argue the HCI obstacles which we simulate in our HCI exemplary task (matching pairs game). For each HCI obstacle, we discuss *volatile* and *persistent* presence in consecutive HCI sessions. A *volatile* HCI obstacle is defined as an obstacle that suddenly appears or disappears in one of consecutive HCI sessions, while a *persistent* HCI obstacle constantly presents in those monitored consecutive HCI sessions.
According to HCI cognitive modeling literature (e.g. Card et al. (1983)), HCI modeling consists of three main processes: 1) *Perception Process* for perceiving stimuli from the computer application. 2) *Cognitive Process* for encoding perceived stimuli in the working memory (WM), interacting with long-term memory, and eventually commanding motor actions (e.g. mouse click) through the 3) *Motor Process*. HCI obstacles can typically impair any of those processes. To model HCI obstacles, we excluded in our research the treatment of motor obstacles, because it is neither easy to simulate HCI motor obstacles scenarios, nor practical to invite real handicapped subjects for such experiments. Moreover, HCI motor obstacles happen typically, and permanently, to disabled (handicapped) users, who can be supported by specific, handicapped-oriented HCI applications, without a big need to an automatic detection and compensation. However, HCI obstacles can temporarily impair the *Perception Process* or *Cognitive Process* of any computer user (young or elderly, healthy or handicapped). Thus, we aim at automatic detection of those HCI obstacles which impair the human perception and cognition processes.
In this thesis, we investigated the detection of real and simulated HCI ob-

[1] `https://github.com/sromku/memory-game`

stacles. We began first with simulated HCI obstacles, because they are controllable and easier comparable under different experimental conditions. For *Cognition Process*, WM plays a major role in successful HCIs (Card et al. (1983)). Therefore, we aim at plausibly simulating, detecting and compensating memory-based HCI obstacles. For *Perception Process*: Visual perception is the most prominent in HCI among other human perceptions (audio and tactile vibrations). Therefore, we aim at plausibly simulating, detecting and compensating visual HCI obstacles. We investigated the plausibility of such simulated HCI obstacles via subjective and objective measurements (Chapter 3), and also via machine learning predictive measurements (Chapter 4). To further investigate the validity of our approach in real-world scenarios, we tested also two variants of real HCI memory-based obstacles: low WM capacity in a complex HCI task and dementia.

1.1.2.1 Memory-based HCI Obstacle

A *volatile* memory-based HCI obstacle (*volatile* MEMOBS) is typically caused by memory load induced by a secondary task and impairs the short-term WM during HCI sessions. In any problem solving task, one might be interrupted in the middle of the task, prompting the formulation of an intention to resume the task later (Altmann and Trafton (2002)). In our HCI exemplary task (matching pairs game), we simulate such a volatile MEMOBS (HCI task's interruption) as follows: matching pairs is played with a secondary task that induces memory load, to create a volatile memory-based obstacle. Whenever the user-study participant reveals a card, a random number between 1 and 9 is spoken by a synthesized voice. The participant is asked to calculate and memorize the sum of all spoken numbers throughout the game (more details are presented in Chapter 3). It is well known that mental arithmetic interacts with working memory (WM) (Chen and Bailey (2020)) and thus such a secondary task limits available WM capacity. This effect works on two different axis: First, the required calculation interferes with the memory encoding of the revealed card; second, the memorized tentative sum is an additional item which needs to be kept in capacity-limited WM. Both mechanisms (distractions and competing items to remember) occur frequently in various HCI situations, making this task a good representative of a typical interaction obstacle. Moreover, for *persistent* MEMOBS, we investigated two real HCI memory-based obstacles: low WM capacity in a complex HCI task and dementia. That is, we collected "ground truth" data about the WM capacity of our participants using standard well-established memory-tests, then we investigated the detection of low WM capacity via our models. For dementia, we have matching pairs logs collected from games played by elderly

healthy subjects and dementia patients, where dementia is a typical persistent memory-based obstacle to be detected by our models (more details about data collections in Chapter 3, detector models in Chapter 4 and adaptive model in Chapter 5).

1.1.2.2 Visual HCI Obstacle

A *volatile* visual HCI obstacle (*volatile* visObs) is typically caused by a temporary impairment on the visual perception of UI's items. For example, a realistic use of mobile devices happens typically outdoors, where sunlight can impair the recognition of the UI's visual items (so-called glare-effects obstacle). Thus, we simulate that realistic case within our exemplary HCI task (matching pairs), see Figure 1.3.

Figure 1.3 – volatile visObs: Cards theoretically have well-distinguishable colors, but simulated sunlight causes a volatile glare-effects obstacle.

To simulate a *persistent* visObs, we emulate color-blindness visObs, also called red-green color vision deficiency, because about 8% of men and 0.5% women in the world have the common form of color blindness (Ostia et al. (2019)). In this variant, the cards of the game only show different shades of brown color to emulate red-green color vision deficiency as a visual obstacle game variant. The cards are still all different, but are very similar and not easily connected with familiar color labels to memorize card locations. This transformation is done through a brown overlay with 75% opacity to all

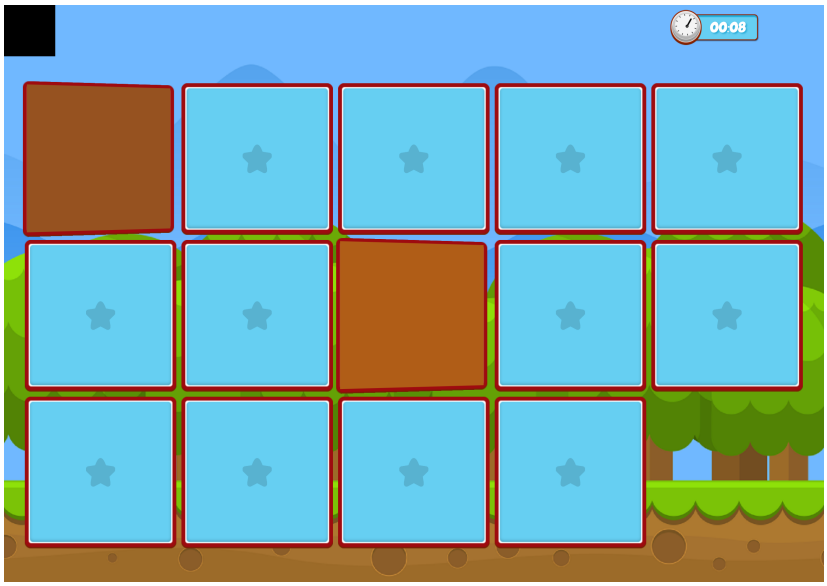


Figure 1.4 – persistent visObs: Two (actually different) cards in the VisObs variant, simulating red-green color vision deficiency

cards, moving them closer together in the color space. Figure 1.4 shows how the affected cards look like in the graphical UI. While the concrete color perception depends on the individual and the exact color-blindness condition, this mode is a good representative of how a color-blind person perceives a user interface with no further (e.g., texture-based) cues to enhance discriminability. In both Figures 1.3 and 1.4, the variance between the shown visual stimuli is extremely reduced, causing less distinguishable perceived items. According to visual perception psychological literature, such a variance between visual items reflects efficiency in encoding representations as well as efficiency in detecting their statistical properties (Siegelman et al. (2019)). Thus, low variance between visual items in an UI causes less distinguishable visual items and impairs the so-called human visual statistical learning performance (Siegelman et al. (2019)).

1.1.3 UI Adaptation Mechanisms

Our introduced cognitive adaptive model aims at automatic and continuous detection and compensation of HCI interaction obstacles. For the compensation process, shown in the upper highlighted box in Figure 1.6, we introduce hereby two different UI adaptation mechanisms regarding to the two aforementioned HCI obstacles (memObs and visObs). Namely, we introduce: 1) UI adaptation for compensating potential HCI memObs, so-called memAdapt,

and 2) UI adaptation for compensating potential HCI visObs, so-called visAdapt.
Those UI adaptation mechanisms will be used by the cognitive adaptive model. Concretely, the most probable UI adaptation mechanism will be automatically selected within the *Automatic UI Support* process, see the upper highlighted box in Figure 1.6. The detection and compensation processes are applied continuously for consecutive HCI sessions, to allow an adaptive and continuous support for computer systems' users. More details are presented in Chapter 5.

1.1.3.1 UI memAdapt

The memAdapt adaptation mechanism is designed to relieve the player's working memory. We follow Murdock Jr (1962) to stimulate last information in memory which could be not easily remembered because of primacy effects. Moreover, the rehearsal of past information in memory improves its recall probability (Pröpper et al. (2011)). Whenever a player reveals non-matching cards, the last two cards prior to these are re-revealed for a very short time. The time of the additional reveal is kept short (225 ms) and skips the usual animations for revealing to avoid extending the time between game turns, which could counter the positive effect of the memory adaptation through additional decay of other information.
We named the aforementioned UI memAdapt mechanism *Light UI memAdapt*, since it slightly adapts the UI. In addition, we tested also another variant so-called *Strong UI memAdapt* which, in contrast, re-reveals all the previous revealed cards after each non-marching turn. Although such a strong UI adaptation mechanism showed a strong improvement in user performance depicted via subjective and objective measurements, it was ranked by participants with high non-acceptance in the corresponding subjective questionnaires because of its long delay. Similarly, objective measurements show it really needs very long time, actually, instead of adapting the UI, it changes the UI, and this is not the target of cognitive adaptive systems. More details about subjective and objective measurements and discussions are presented in Chapter 3.

1.1.3.2 UI visAdapt

The visAdapt adaptation mechanism is tailored towards users with visual perception impairment. Following theories of multi-resource cognition and perception e.g. Wickens (2008), we leverage an additional modality: visual information on the playing cards is accompanied by a synthesized voice which represents the card content, utilizing an information channel which is likely not

affected by the perceptual obstacle. In our implementation, the German letters $a = /a : /$, $c = /tse : /$, $j = /yot/$, $q = /ku : /$, $x = /iks/$, $v = /fow/$ and $l = /ell/$ were chosen as voice cues, as they are short and easily distinguishable in the German pronunciation. When applied to a situation in which no visual obstacle is present, visual adaptation can have a detrimental effect. Thus, we complement the impaired visual memory by stimulating the unimpaired auditory memory, that is, psychological literature works e.g. Crowder (1972) highlighted complementary roles between auditory memory and visual memory during the perception and cognition processes, therefore, the detection and compensation of such HCI obstacles is so important for successful perception and cognition in HCI.

1.2 Cognitive Adaptive Model Architecture

In this section, we introduce the general architecture of our cognitive adaptive model. We describe hereby the overall architecture, and we only name its sub-models. Later on, we focus on each sub-model and describe its role in the two-processes scenario mentioned above: *Interaction Obstacles Detection* and *Automatic UI Support*.

Before discussing the architecture, we define the following HCI scenarios using the HCI exemplary task, obstacles and UI adaptations introduced in Section 1.1: during playing a matching pairs game as an HCI exemplary task, assume that the player's working memory has been distracted somehow, e.g. via a secondary memory load which does not belong to the memory game task, then her or his HCI performance (finding the pairs) will be impaired by such a memory-based obstacle (short: MEMOBS). Another obstacle example is for the human perception process: assume that the player cannot appropriately recognize the shown colors of revealed cards, e.g. because of sunlight glare effects or color blindness disease, then, her or his HCI performance (finding the pairs) will be also impaired by such a visual obstacle (short: VISOBS). The proposed cognitive adaptive system aims at "online" detection and compensation of such obstacles, i.e during the system use. Compensation is done by an automatic appropriate UI adaptation: UI adaptation mechanism to tackle the MEMOBS, so-called MEMADAPT (repeating information by re-revealing previously shown cards) and UI adaptation mechanism to tackle the VISOBS (auditory instructions to help the player recognizing the cards), so-called VISADAPT.

Figure 1.5 shows the general architecture of our cognitive adaptive model, with four main parts depicted from 1 to 4. In part 1, we see a real photo

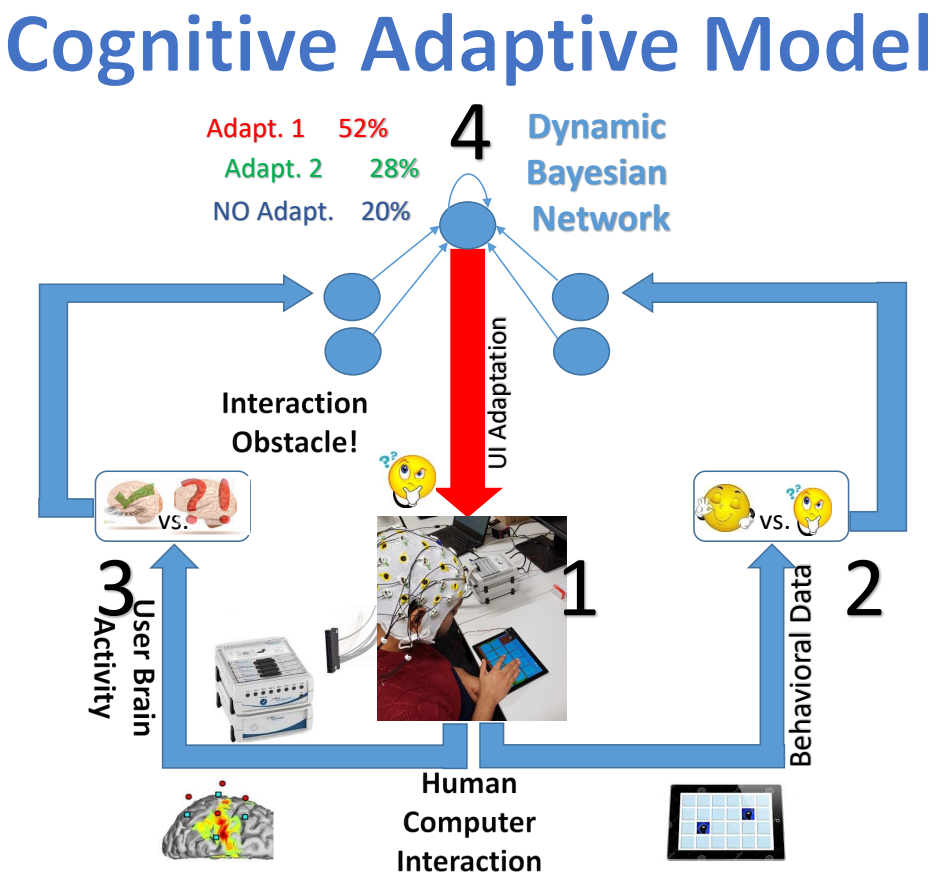


Figure 1.5 – Online Cognitive Adaptive System: general overview

from our experiments (Chapter 3) which represents an HCI task. During that HCI task, our model continuously observes multimodal data (parts 2 and 3): logged user actions as *behavioral data modality* (part 2), and brain activity data recorded via non-invasive electroencephalography (EEG) electrodes as physiological *brain activity data modality* (part 3). These recorded multimodal data will be passed on to corresponding HCI obstacle detector models:

- behavior-based HCI obstacle detector (part 2 in Figure 1.5): a binary classifier detects the presence or absence of an HCI obstacle from the recorded behavioral data.
- EEG-based HCI obstacle detector (part 3 in Figure 1.5): a binary classifier detects the presence or absence of an HCI obstacle from the recorded EEG data.

We name those multimodal HCI obstacles detectors as "*Elementary Models*" (Chapter 4), because they contribute but do not decide the final UI adaptation,

where their outputs will be passed on to the overarching *Probabilistic Model* (part 4 in Figure 1.5) which in turn decides the most appropriate UI adaptation (Chapter 5). This decided UI adaptation is applied then to the HCI application (the red arrow) for the next HCI session, which will be experienced by the user, thus, new multimodal data will be recorded accordingly, and the same process will be applied continuously to either consolidate the decided UI adaptation, in case of *persistent* HCI obstacle, or to revert and correct that decided UI adaptation in case of wrongly decided UI adaptaion in the previous session, or even in case of a *volatile* HCI obstacle which either has been detected in the previous session but suddenly disappeared in the current tested session, or suddenly appeared as a new obstacle in the current tested session.

1.3 Interaction Obstacles Detection and Automatic UI Support

In this section (and its sub-sections), we highlight the two main processes in our model: *Interaction Obstacles Detection* and *Automatic UI Support*, see Figure 1.6. The obstacle detection process (lower highlighted box) consists of two multimodal obstacle detectors (as mentioned above, called *Elementary Models*, Chapter 4): behaviour-based obstacle detector (Section 1.3.1.1) and EEG-based obstacle detector (Section 1.3.1.2). The UI adaptation process (upper highlighted box) will be briefly introduced in Section 1.3.2 (details in Chapter 5).

1.3.1 Elementary Models

In this section, we highlight and briefly introduce the multimodal HCI obstacle detectors (so-called "*Elementary models*"), see the behavioural and physiological elementary models in Figure 1.6. Each individual modality HCI obstacle detector (individual *elementary model*) will be discussed with some details and illustrations in the coming sub-sections, while detailed discussions and evaluations of all the *Elementary models* will be presented in Chapter 4.

1.3.1.1 Behavioural Elementary Model

In this section, we briefly introduce our behaviour-based HCI obstacle detector, which is highlighted in Figure 1.6 as *Behavioral Elementary Model*. We zoom-in that model to look at some details about behaviour-based HCI obstacles detection, see Figure 1.7.

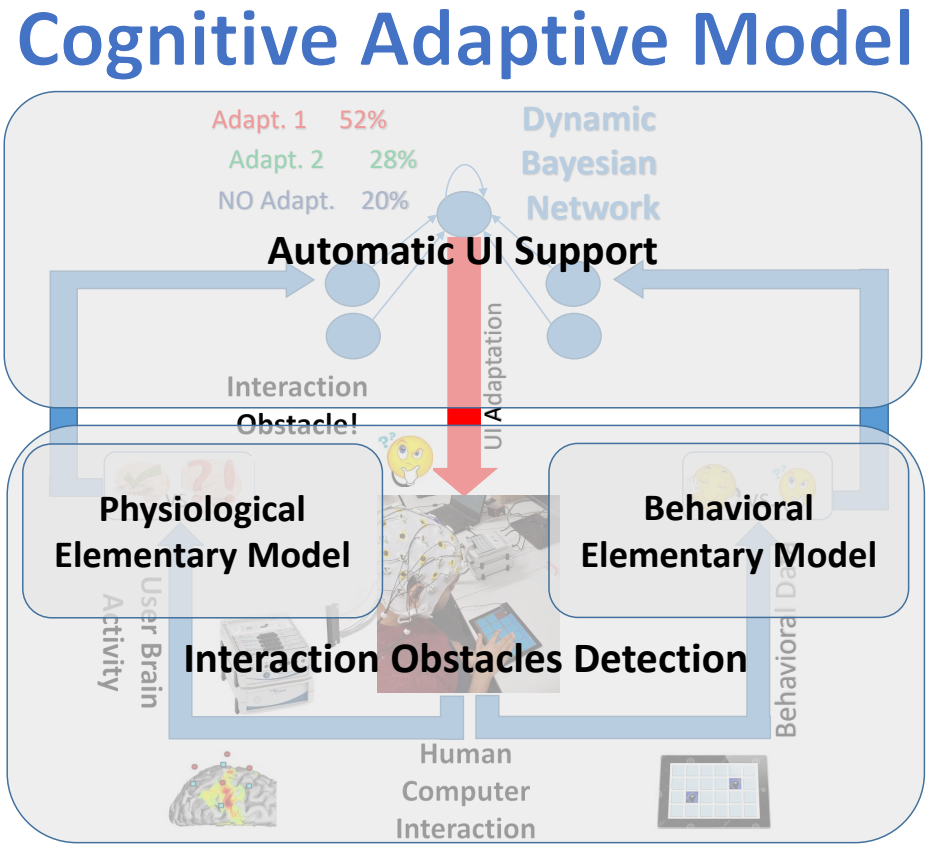


Figure 1.6 – Online Cognitive Adaptive System: HCI obstacles detection and UI adaptation

Behavioural data is a major data modality in each HCI application, because it depicts user actions during HCI, and thus it preserves the context of the interaction. User actions are not isolated from each other, i.e. there are temporal *sequential dependencies* between user actions in different time steps, which would lead to different contexts. For example, given the following classical HCI task: A user interacts with UI with different menus, each has different operational commands, e.g. File -> (new, open, close, save or exit), Edit -> (copy, paste, cut, find, find & replace) etc. HCI user will show different *HCI behaviours* with such different operational commands (different contexts), because those different commands will open different windows e.g. opening file window, finding a text window etc. Consequently, each user action (clicked command) will change the user's next actions. Similarly, in matching pairs game, a sequence of player actions determines the game state. Therefore, we need to systematically model user actions and their potential sequential dependencies to allow an automatic detection of the presence or absence

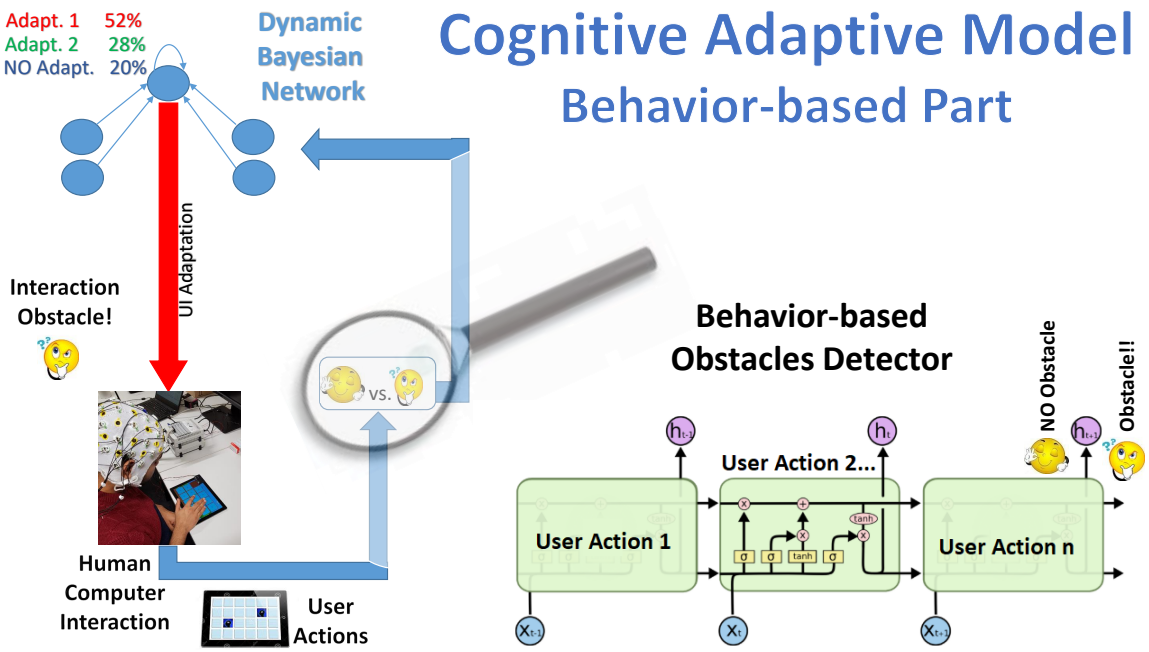


Figure 1.7 – Behavior-based HCI obstacles detection: Exploit user actions dependencies

of an HCI obstacle by exploiting such dependencies, see the illustration of user actions modeling in the zoomed-in behavior-based obstacle detector in Figure 1.7.
Details about behavioral data recording and encoding are explained in Chapter 3, and more details and evaluations about behaviour-based obstacle detectors are discussed in Chapter 4.

1.3.1.2 Physiological Elementary Model

Human Working Memory (WM) plays a major role in the HCI cognitive modeling. According to milestones in the HCI cognitive modeling literature (e.g. GOMS Card et al. (1983) and ACT-R Anderson (1990)), WM plays a major role in all the three main processes of HCI: perception, cognition and commanding motor actions. That is, WM initially receives stimuli from the perception sensory-model (visual stimuli perceived by eyes, audio stimuli by ears, or tactile vibration stimuli by skin). Then, WM encodes such stimuli and fetches related facts from Long-Term Memory (LTM). Even for motor actions, which are decided by a separated *Production Memory* (ACT-R, Anderson (1990)), WM receives those decided actions (e.g. mouse click, keystroke, screen touch etc.) from that *Production Memory* and commands them to

the motor system to do an appropriate action. To sum up, WM controls the

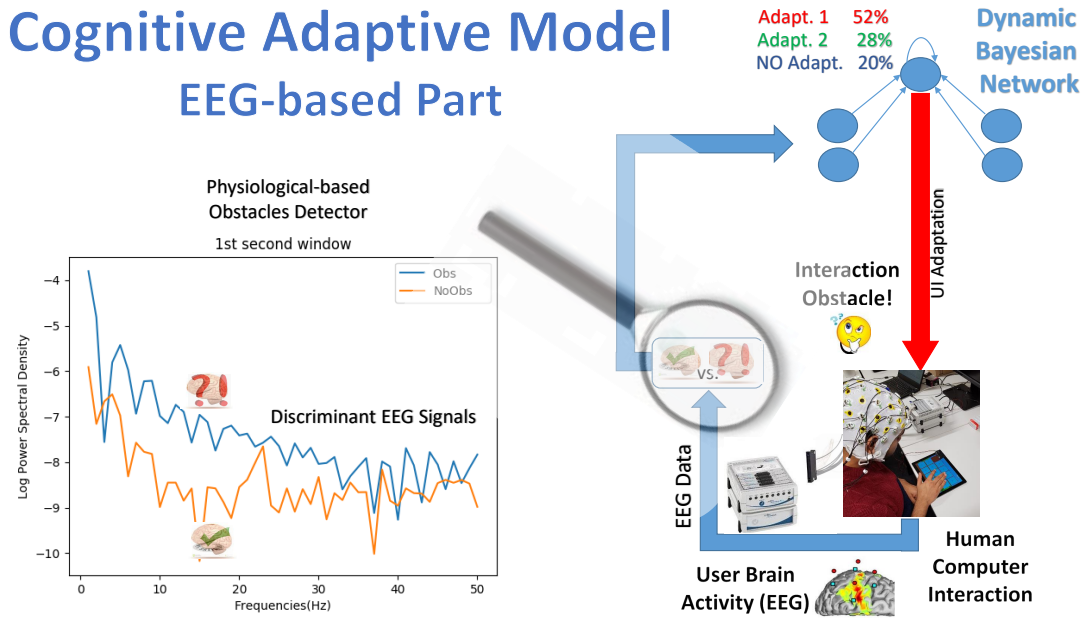


Figure 1.8 – EEG-based HCI obstacles detection: classify discriminant EEG signals

whole HCI session, because it contributes in all HCI processes. Thus, if WM is somehow impaired, human performance in HCI is expected to be weakened in terms of worse perception and cognition of stimuli, and consequently inappropriate motor actions. Continuous monitoring of WM status during HCI could detect abnormal WM's states, and thus, an automatic support, in terms of appropriate UI adaptation, can be applied to improve the user performance. WM is widely analyzed via recording brain activity data, e.g. Klingberg et al. (2002) shows a correlation between WM capacity and increased brain activity. Thus, to follow a user's WM states during an HCI session, we record and treat brain activity data (recorded as Electroencephalography: EEG) as physiological data modality.

For the recorded EEG data, we briefly introduce the corresponding physiological-based obstacle detector model, so-called *Physiological Elementary Model* in Figure 1.6. We zoom-in to look at some details about that model, see Figure 1.8. Following the EEG literature e.g. Kumar et al. (2016), we expect discriminant EEG signals with naturally different HCI sessions: No obstacle HCI session (NOOBS) vs. HCI obstacle session (OBS), see such discriminant EEG signals in Figure 1.8. In our experiments (Chapter 3), we record EEG data in HCI sessions with and without obstacles, and we train appropriate

machine learning models on the recorded EEG data to automatically discriminate it, i.e. to detect the presence or absence of an HCI obstacle during an HCI session (more details are presented in Chapter 4).

1.3.2 Probabilistic UI Adaptation

In Sections 1.3.1.1 and 1.3.1.2, we introduced the HCI multimodal obstacles detectors (Elementary models). Those elementary models pass their outputs (predictions with confidence measures) to an overarching probabilistic model, which is highlighted in the upper box of Figure 1.6.
In this section, we briefly introduce that overarching probabilistic model. Given inputs (predictions and confidences) from different multimodal HCI obstacle detectors, a Bayesian decision will be made to select the most probable UI adaptation mechanism to be applied on the subsequent HCI session (e.g. the Adapt.1 in the illustration Figure 1.3). Moreover, given that users interact with systems not only once, but often several times in succession, such a model should be able to consolidate its result across multiple subsequent interactions and to automatically recover from recognition errors, which may lead to suboptimal adaptation. In other words, the introduced probabilistic Bayesian model should be "dynamic" through time steps (consecutive sessions); It should continuously learn from every new input from the lower "Elementary models", after applying a UI adaptation mechanism in the previous HCI session, see the self loop in the upper highlighted box of Figure 1.6 (more details are presented in Chapter 5).

1.4 Main Contributions

At the end of this introduction chapter, we briefly list hereby the main contributions of this PhD thesis:

1. **Multimodal HCI obstacles detection**: behavior-based and EEG-based detection of memory-based and visual HCI obstacles (the lower highlighted box of Figure 1.6).

2. **Automatic probabilistic UI adaptation**: Given different detected HCI obstacles with confidences from multimodal obstacle detectors, the most probable UI adaptation will be automatically selected for the next HCI session (the upper highlighted box of Figure 1.6).

3. **Online Continuous Cognitive Adaptive System**: combining submodels: *Multimodal HCI obstacles detection* and *probabilistic UI Adaptation*, HCI applications can be supported with an online cognitive

adaptive system, which works in an infinite-loop fashion allowing continuous support (UI adaptation) through consecutive HCI sessions.

4. **Model Flexibility**: The model is designed in a flexible, extendable fashion; New HCI multimodal obstacle detectors can be easily added or removed under the probabilistic decision model (UI Adaptation model).

5. **Transfer Learning**: As we trained multiple HCI obstacle detector models under specific experimental design settings, we also show that such models can even perform well in other real world settings. Namely, we trained HCI memObs detectors from logs collected from healthy young subjects. For those young subjects, we simulated the memObs to be detected as secondary task memory load in the HCI task. However, we will show also that those models perform well in detecting dementia disease from elderly population, which is a valuable contribution both as a useful diagnosis (assistant) tool, and also as a prove of transfer learning ability of our models: Trained by data from young healthy subjects for simulated memObs, and performed well for detecting dementia as a real obstacle with different test settings (elderly healthy vs. dementia) compared to the train settings (young in noObs sessions vs. young in simulated memObs sessions). This contribution could be similarly extended, in a future work, by investigating the detection of real color vision deficiency disease using our visObs detectors.

1.5 Summary

In this introduction chapter, we have introduced our model architecture, only as general blocks and brief descriptions. In Chapter 2, we will discuss the HCI evolution story, including *User Experience* and *User Modeling* in the literature, and we will highlight our contributions in the presented HCI evolution story. The practical part of this thesis, so-called thesis pipeline, will be discussed in three chapters: 1) Chapter 3 describes all the user studies designed to collect data for our models. 2) In Chapter 4, we will use the data collected and discuss the different *Multimodal HCI Obstacles Detectors*, so-called *Elementary Models*. 3) In Chapter 5, we will put those *Elementary Models* together under an overarching probabilistic model which continuously decides the most probable *UI adaptation* after each *HCI session*. We will conclude in Chapter 6 with summary, visualisation for different results facets, limitation and future work discussions.

Chapter 2

User Modeling in Human Computer Interaction

In this chapter, we recap the HCI story from the literature, highlight milestones, and incorporate our user modeling based approach: multi-modal HCI obstacles detection and corresponding compensation with UI adaptations.

We have three main targets in this chapter:

1. Recap the HCI evolution since early stage milestone in cognitive modeling until recent cognitive adaptive models.
2. Highlight gaps in the recent cognitive adaptive models.
3. Incorporate our model (introduced in Chapter 1) in the HCI evolution to tackle such gaps in the state of the art adaptive models.

First, we searched for early milestones in the HCI cognitive modeling literature. The GOMS cognitive HCI models (GOMS: Goals, Operations, Methods and Selection), had become well-known in the literature: Card et al. (1980a,b, 1983). While the first two models (Card et al. (1980a,b)) mainly discussed specific HCI motor interactions (text editing and keystroke human responses), the third model Card et al. (1983) introduced a comprehensive human cognitive model so-called *Human Information-Processor* which consists of 3 interconnected sub-models as a classical input-processing-output model. Card et al. (1983) became a milestone in the HCI literature, thus, we discuss it in

details in Section 2.1. That model has been discussed in different works for extensions, highlighting powerful features and shortcomings (Olson and Olson (1995)), and for serious psychological questions and critiques (e.g., Carroll and Campbell (1986); Karat (1988); Wilson et al. (1988)), more details are presented in Section 2.2.

Later on, the design of static user-friendly User Interface (UI) has been replaced by a new concept so-called *User Experience* during HCI. A milestone work is presented by Kuniavsky (2003), details are presented in Section 2.3.1. However, users differ in gender, age, preferences, cognitive skills and other characteristics. This motivates the evolution of the so-called *User Modeling* solutions: to allow systems to better meet individual user needs and expectations. A reference milestone for this domain is presented by Fischer (2001), in which he discussed *User Modeling* in HCI and reviewed the state of the art for corresponding works along ten years, more details are presented in Section 2.3. With new technologies evolving, adaptive UI solutions have been recently proposed: O'sullivan (2012); Sguerra and Jouvelot (2019); Miller (2020), more details are presented in Section 2.4.

All those recent adaptive UI solutions have the following gaps in common: solely depend on tracking user performance, and no end-to-end adaptive system by multimodal obstacle detection and continuous UI adaptation. This will be our contribution in this thesis, where we propose a flexible and extendable model (recall Chapter 1), which can be offered as a stand-alone *User Modeling* solution, as well as *plugin solution*: it is flexible for complementing existing models with "HCI Obstacles Detection and UI Adaptation" capability, more details about the incorporation of our models in the HCI evolution are presented in Section 2.6.

After this brief introduction, the structure of this chapter is organised as follows: we recap the HCI and *User Modeling* history form the literature, and discuss background research milestones in the *Human Cognition Modeling* literature. We discuss the well-known GOMS model in Section 2.1 and its follow-up works in Section 2.2. The state-of-the-art *User Modeling* solutions are discussed in Section 2.3. In Section 2.4, we discuss the state-of-the-art of recent cognitive adaptive models and highlight the gaps which can be tackled by our models. For our introduced models, we discuss related background research works in Section 2.5, namely ACT-R theory and Cognitive Memory Model CMM. Finally, we present with illustration figures in Section 2.6, how our models could be incorporated in the HCI evolution.

2.1 GOMS Model: Human Information Processor

In this section, we discuss the GOMS model proposed by Card et al. (1983) (Goals, Operations, Methods and Selection): a psychology-based HCI model which consists of three interacting sub-models: 1) a *Perceptual Processor* for modeling the human perception of stimuli coming from the computer application, 2) a *Cognitive Processor* for modeling the cognition process by short-term and long-term memory, and 3) a *Motor Processor* for modeling the motor commands for movement related to mouse click, keystroke etc.

Card et al. (1983) modeled the human perception, cognition and responses (motor actions) as three separated, but interacting *processors*, each with its own memory buffers. Card et al. (1983) used the psychology science to estimate time parameters for perception, cognition and motor actions. Altogether, we can look at GOMS models as a classical input (perception), processing (cognition) and output (motor actions) model.

As we aim at modeling multimodal HCI obstacles within this thesis, those obstacles can typically happen in either part: perception, cognition or motor actions. However, we focus only on those HCI obstacles happening in perception and cognition processes, because in such processes the presence of HCI obstacles is dynamic and can happen to any user group: young/elderly or healthy/handicapped users. In other words, there is a need for an automatic dynamic cognitive adaptive system in perception and cognition processes, in contrast, potential HCI obstacles in the output part (motor actions) are more likely related to specific user groups (e.g. users with disabled motor capabilities for any HCI application, or specific HCI application for specific users e.g. surgery support system for doctors whom hands' movements will be limited during the surgery, etc.); Such specific user groups can be supported with specific, static handicapped-oriented systems, without a big need for such a dynamic cognitive adaptive system. Moreover, it is neither easy to simulate HCI motor obstacles scenarios, nor practical to invite real handicapped subjects for such experiments.

In the following sub-sections, we discuss in more details: 1) the input model (*GOMS Perceptual Processor*, Section 2.1.1), 2) and the processing model (*GOMS Cognition Processor*, Section 2.1.2). In both GOMS input and processing models, we incorporate our proposed adaptive system accordingly. We conclude the GOMS discussions in Section 2.2 with GOMS follow-up research.

2.1.1 GOMS Perceptual Processor

Card et al. (1983) modeled their *Perceptual Processor* system in three parts: sensors, associated buffer memories and processor. That is, eyes and ears represent sensors for receiving visual and auditory stimuli respectively, visual stimuli will be perceived in the *Visual Perception Memory*, and auditory stimuli will be perceived in the *Auditory Perception Memory*. Eventually, those stimuli will be then encoded in the short-term working memory (WM). Visual and auditory memories play then complementary roles in the cognition process, Crowder (1972).
As a simple illustration, we depict GOMS perceptual sensory system (eyes and ears), perceptual memories (visual and auditory) and perceptual processor unit in Figure 2.1. In the next sub-sections, we discuss the GOMS perceptual sensors and memories in more details.

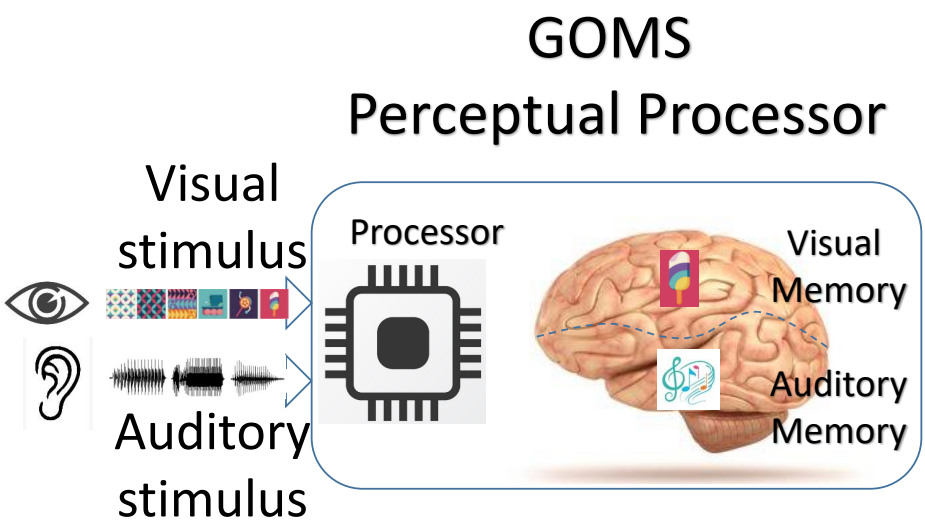


Figure 2.1 – Simple illustration of the GOMS Perceptual Processor: sensory system (eyes and ears) receives visual and auditory stimuli. Processor encodes stimuli in corresponding visual and auditory perceptual memory buffers.

2.1.1.1 Perceptual Sensors

The *Perceptual Processor* includes two separated sensory systems: *Visual System* and *Auditory System*. Card et al. (1983) modeled, based on psychological studies, time parameters for each sensory perception. For example, the authors modeled Eye-movement during HCI:

$$Eye - movement = 230[70, 700]msec$$

I.e. humans typically need 230 msec for eye movement during HCI, while these values may range from 70 msec to 700 msec depending on conditions of measurement, task variables, or subject variables. Such parameters are helpful for developers to consider restrictions in the design of UIs. Moreover, those time-based parameters can be used for automatic adaptations of UIs, by monitoring user performance according to sensory-parameters values, e.g. user's real eye movement in an HCI application. However, detecting an abnormal behaviour caused by HCI obstacle is not plausible simply from monitoring of HCI user performance, because parameters ranges (e.g. [70, 700] for eye-movement) are quite wide. To tackle this gap in HCI modeling, we introduce an automatic modeling of HCI obstacles from multimodal HCI data mentioned in Chapter 1 and further detailed in Chapter 3.

2.1.1.2 Perceptual Memories

As briefly mentioned in Section 2.1.1, according to the GOMS model, stimuli will be perceived by human "sensors" (eyes and ears), encoded in perceptual memories (visual and auditory memories), and eventually represented in WM for the cognition process. In this section, we discuss those visual and auditory perceptual memories.
Card et al. (1983) modeled those perceptual memories as buffers which hold unidentified, non-symbolic analogue to the received external stimulus. The stored information in perceptual memories are affected by physical properties of the stimulus e.g. intensity. According to psychological literature works, the authors defined time restriction parameters for determining times after which those stored information decay from perceptual memories. Namely, they modeled so-called half-life parameters, which determine times after which the probability of retrieval is less than 50%:

$$\delta_{VIS} = 200[90, 1000]msec$$

where VIS is the visual perception memory.

$$\delta_{AIS} = 1500[900, 3500]msec$$

where AIS is the auditory perception memory.

In addition to those half-life time parameters, the authors modeled also half-life capacity parameters; That is, given an HCI session with fixed time length, the capacity of visual and auditory perceptual memories are estimated in terms of numbers of letters that can be kept:

$$\delta_{VIS} = 7[5, 17]letters$$

where VIS is the visual perception memory.

$$\delta_{AIS} = 5[4.4, 6.2]letters$$

where AIS is the auditory perception memory.

As mentioned above, developers can benefit from such modeled restrictions (parameters) for designing effective UIs, and more important, cognitive adaptive systems can benefit from those parameters to monitor user performance in HCI session, and, to adapt the UI accordingly. While this modeling considers differences between users' cognitive skills simply as ranges (e.g. [90, 1000]msec for VIS), advanced *User Modeling* approaches would further follow individual differences to better meet individual needs in the designed UIs.

2.1.2 GOMS Cognitive Processor

In the GOMS model, the *Cognitive Processor* system acts as a processing system, which handles inputs from the *Perceptual Processor* system (Section 2.1.1), and sends commands to the output system: the *Motor System*. Similar to both input and output systems, this processing systems has its own processor and memory buffers. However, the processing system is more complicated than input and output systems. It has a complex processor connected to two types of memory buffers: short-term working memory (WM) for handling perceived stimuli in *Perceptual Processor* system, and long-term memory, from which the complex processor fetches corresponding knowledge facts necessary to complete the processing task, and eventually to send commands to the output system: the *Motor System*.

For simplicity, we depict the main functionalities of the *GOMS Cognitive Processor* in Figure 2.2. According to the GOMS model, the WM capacity is limited, thus, Card et al. (1983) modeled parameters for the WM capacity. Again, such parameters were defined with ranges to model potential differences between users' cognitive skills, however, this is not dynamic enough to plausibly model the WM capacity for individual users. We will see that follow-up works in Section 2.2 highlighted both mental workload and individual differences as drawbacks in the GOMS model. Later on, we will see in Section 2.4, that recent cognitive adaptive UIs track user performance during an HCI session for plausible UI adaptation, which is more dynamic. However, as mentioned in the introduction of this chapter, even recent approaches do not model *interaction obstacles* from available multimodal data for continuous and dynamic adaptation, which is our contribution for the HCI evolution, more details are presented in Section 2.6.

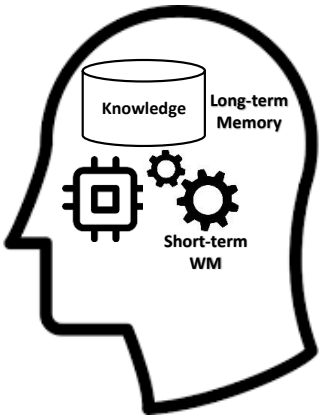


Figure 2.2 – Simple illustration of Cognition Processor according to the GOMS model: It handles perceived stimuli in the perception processor system via short-term working memory (WM). WM interacts with long-term memory to fetch/store corresponding facts for cognition.

2.2 GOMS Follow-up Research

In this section, we recap GOMS follow-up research works, in which researchers discussed GOMS extensions, highlighted powerful features and shortcomings (Olson and Olson (1995)), and psychological questions and critiques (e.g., Carroll and Campbell (1986); Karat (1988); Wilson et al. (1988)).

Olson and Olson (1995) reviewed state of the art HCI cognitive modeling models. Namely, the authors reviewed GOMS models and follow-up coordinated research, which have not only confirmed the basic GOMS parameters, but also extended the parameters that account for the time of certain component activities. The authors concluded, however, that GOMS and its follow-up models fail to capture HCI user's fatigue, individual differences, or mental workload.

Apart from discussions and parameters extensions, we focus in this section mainly on GOMS shortcomings, which constituted requirements for further follow-up research, namely *User Experience* and *User Modeling* (Section 2.3). We summarize the main GOMS shortcomings in the following list:

1. **Mental Workload:** GOMS model does not address mental workload; It does not measure WM capacity during HCI tasks, thus, it does not model plausible amounts of information which can be kept in mind while using the system. This is, however, an important challenging requirement for cognitive adaptive systems, because WM plays a major role in HCI and controls all the main cognitive processes: perception,

cognition and motor actions, where overloaded WM capacity leads to poor decision making in the HCI context. Therefore, it is an important requirement for cognitive adaptive systems to model mental workload and WM capacity during HCI tasks, Sguerra and Jouvelot (2019).

2. **Individual differences:** GOMS model does not address potential characteristics differences between individual users. That is, users vary in aspects such as gender, age, preferences, and more important, they vary in cognitive skills. According to psychology literature e.g. Cooper (1998); Gustafsson and Undheim (1996); Buss and Greiling (1999), individuals show rates of learning, speed of retrieval, and reasoning capabilities. Gomez et al. (1983, 1986) and Egan (1988) discussed GOMS follow-up works between 1983 and 1988 in terms of *individual differences*. The authors reviewed a variety of ways in which individuals differ in their use of computers. Such individual differences leaded later on to a new domain in the HCI literature, namely *User Modeling* which will be discussed in Section 2.3.

3. **Skilled users only:** The GOMS models do not apply to beginner or intermediate users. Such users spend considerable time engaging in problem solving activities, rather than simply retrieving and executing plans. They need training to move smoothly between problem solving and skilled behavior. New users for a computer system often need considerable time for learning and experiencing. Those less experienced users often make errors during an HCI task and learn what must be done next. A user is typically building the so-called "*User Experience*" during the first HCI sessions, thus, HCI researchers should model those users accordingly for enabling effective collaborative HCI, more details are given in Section 2.3.1.

   Less-experienced computer users have been of high interest to cognitive psychologists starting in the 90's. Polson and Lewis (1990) discussed a model of learning by exploration, so-called CE+. For the naming, the authors explained that the CE+ comes from the combination of their earlier work on CCT and EXPL and the inclusion of significant additional ideas. The CE+ model extends the GOMS model by incorporating users' learning from examples. CE+ model utilizes problem-solving processes for simple puzzle-like problems in order to let less-experienced users learn from examples. A milestone framework for modeling less-experienced computer users was SOAR, introduced by Laird et al. (1987). The SOAR model analyzes situations in which a user is faced with learning a new system after knowing one that shares some relevant features.

SOAR extended the GOMS model with parameters for estimating how long it takes to recognize a low experience interaction, how long it takes to set up a new goal, how quickly similar productions are retrieved and analyzed, and how many steps are executed before a solution is found.

4. **No recall of disuse:** The GOMS model does not parameterize recalling from system disuse. Even skilled computer users may make errors. HCI cognitive models need to model how computer users adapt their own behavior to recall from errors or system disuse. For GOMS model, new requirements for recalling from errors or disuse need to be incorporated in GOMS different processor and memory buffers. This requirement, among others, led researchers to new paradigms in HCI cognitive modeling: *User Experience* and *User modeling* (Section 2.3).

5. **Error-less performance:** Relevant to the previous point, GOMS parameters only model error-free performance from users. Although GOMS parameters are defined with ranges to suit a variety of performance values, such ranges (e.g. WM decay $\delta_{WM} = 7[5, 226]sec$) do not explicitly address performance drop because of HCI obstacle, or performance gain because of user competency. Thus, it is valuable to model HCI obstacles and user competencies in cognitive adaptive systems, which can then adapt their UI accordingly.

2.3 User Modeling

We have seen in the previous sections that the GOMS models are one milestone in the HCI literature, which has been widely discussed for extensions, advantages and shortcomings. Regarding shortcomings discussed by GOMS follow-up research (Section 2.2), we discuss hereby the evolution of *User Modeling* research which has been continuously addressing those issues until successfully developing adaptive UIs. Fischer (2001) presented a well-established reference work in user modeling in HCI. According to the authors, *User Modeling* research aims at defining user models for HCI applications for building up and modifying a conceptual understanding of the user. The authors distinguished also between user models and mental models which, in contrast to user models, define those systems and tasks reside in the users' mind, in interactions with others and with artifacts.

First, we discuss in Section 2.3.1 the shift in HCI research focus from User-friendly UIs towards tracking the so-called *User Experience*. Then, we discuss

in Section 2.3.2 how implicit knowledge-bases are introduced in HCI research for moving towards the so-called *collaborative HCI*.

2.3.1 User Experience

Alongside the continuous evolution of HCI research, the focus in the design of UIs has been shifting from traditional user-friendly fashion towards the so-called *User Experience* fashion. *User Experience*, for a computer application, is defined by tracking how a user experiences that application through its UIs during corresponding HCI sessions. That is, Hassenzahl (2018) systemically illustrated how a model of *User Experience* constitutes upon the so-called *product character*, i.e. computer system character for the HCI domain. According to Hassenzahl (2018), a product designer analyzes requirements to fix different product features (a product here is a computer application and its UI): content, style of presentation, functionality and style of interaction with users. These features apply in the production time (design time) when the application is implemented with a so-called *intended* product character: pragmatic and hedonic attributes. While pragmatic attributes are inextricably tied to internally generated or externally given behavioural goals, hedonic attributes are tied to individuals' self and their psychological well being. Those attributes, however, will be then experienced by each individual user. Thus, they individually constitute in real time the so-called *apparent* product character. These attribute estimate user's appeal (e.g., "It is bad"), emotional consequences (e.g., pleasure, satisfaction) and behavioural consequences (e.g., increased time spend with the product), which altogether represent *user experience attributes*.

Hassenzahl (2018) concluded that *User Experience* is subjective, where actual users' experiences with a computer application may considerably differ from these experiences intended by the designer. This highlights the importance of modeling the actual user experience online during the system use, and, adapting the UI accordingly. We find potential incorporation for our approach in the *User Experience* attributes as follows: The detection of HCI obstacles could estimate the user's appeal, and the corresponding UI adaptation would move the user status into satisfaction, or even pleasure. While we only focused in this thesis on the multimodal detection of HCI obstacles and UI adaptation, the multimodal estimation of those *User Experience* attributes before and after obstacles detection and UI adaptation is an interesting follow-up research for our work.

In Section 2.3.1.1 we follow Hassenzahl (2018) for discussing the combination of both pragmatic and hedonic attributes in product character. Then, we

present in Section 2.3.1.2 how *User Modeling* researchers treated appeal, satisfaction and pleasure attributes.

2.3.1.1 Combination of Pragmatic and Hedonic Attributes

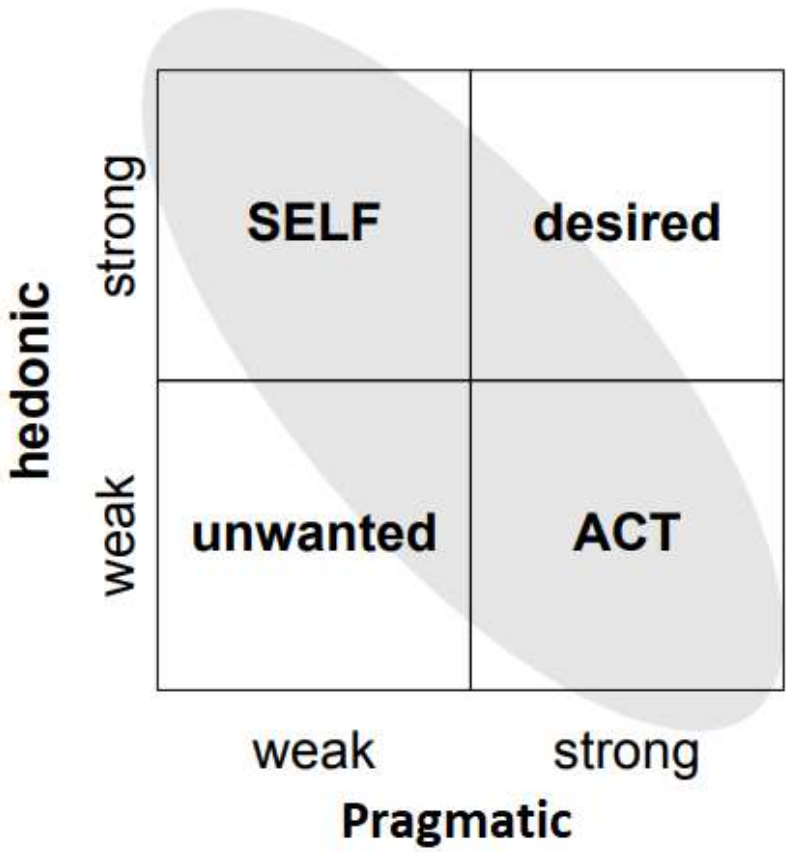


Figure 2.3 – Hassenzahl (2018): Combination between pragmatic and hedonic attributes.

Hassenzahl (2018) discussed the combination of pragmatic and hedonic attributes, which constitute together the product character. Human perception of pragmatic and hedonic attributes can be either weak or strong, thus, four types of product characters evolve, see Figure 2.3. When a product is neither able to satisfy pragmatic nor hedonic needs of potential users, its user experience is characterized definitely as "unwanted". The combination of strong pragmatic and strong hedonic attributes, in contrast, is characterized as "desired". According to the author analysis, most likely, both attribute groups will not be in balance. Hassenzahl (2018) concluded to call a primarily pragmatic product (i.e., "strong pragmatic and weak hedonic") an ACT

product and a primarily hedonic product (i.e., "weak pragmatic and strong hedonic product") a SELF product.

Hassenzahl (2018) distinguished between behavioral goals and user's self. On the one hand, behavioural goals vary and can be externally given by others or internally generated by the individual. On the other hand, users' self represent their ideals, memories, and relationships. The author emphasizes that an ACT characterized product (pragmatic) is inextricably linked to its users' behavioural goals, while in contrast, a SELF characterized product (hedonic) is inextricably linked to the users' self. Our proposed multimodal HCI obstacle detectors can be linked to these findings, where our introduced behaviour-based obstacle detection can be linked to that pragmatic ACT model, and on the other side, our introduced physiological (EEG-based) obstacle detection can be linked to the hedonic SELF model.

2.3.1.2 Appeal, Satisfaction and Pleasure

Appeal, satisfaction and pleasure are defined as *User Modeling* attributes to evaluate HCI sessions. Such attributes are momentary and take the usage-time rather than design-time into account. Researchers viewed those attributes as outcomes of experience with or through technology, e.g. Wright et al. (2003). Generally speaking, *Appeal* attribute represents user's good or bad impression about the computer application, *Satisfaction* attribute estimates user's expectation's fulfilment, and *Pleasure* (also called joy) attribute is defined to track unexpected and exciting features found by the user.

Hassenzahl (2018) concluded that while a user's good or bad impression constitutes her or his appeal attribute, a user is likely to experience combinations of satisfaction and pleasure attributes. The author discussed these satisfaction and pleasure combinations in both ACT and SELF products. For ACT product, pleasure may additionally be experienced if expectations about goal achievement (e.g., ease of achieving a goal) are excelled. Recall that SELF products are used to fulfil psychological needs rather than behavioural goals. Weak connection applies to goals and expectations in SELF products, thus, these products are less restricted and more likely lead to pleasure.

While HCI standards explicitly coincide with such attributes (e.g. ISO 9241-11 for *Satisfaction*), it is an open challenge to systematically measure those attributes. For this thesis, we can follow *User Experience* research concepts as background, and incorporate our introduced models accordingly: The *HCI Obstacles Detection* model in this thesis can be seen for estimating *Appeal* attribute, and the *UI Adaptation* model for ensuring *Satisfaction*, and even

Pleasure attributes. Our multimodal detectors' fashion (Recall Chapter 1) also coincides with those pragmatic and hedonic product characters discussed in *User Experience*, where we record behavioral data (ACT's nature: pragmatic behavioral) and physiological data (SELF's nature: hedonic user internal).

2.3.2 Implicit Knowledge-based HCI

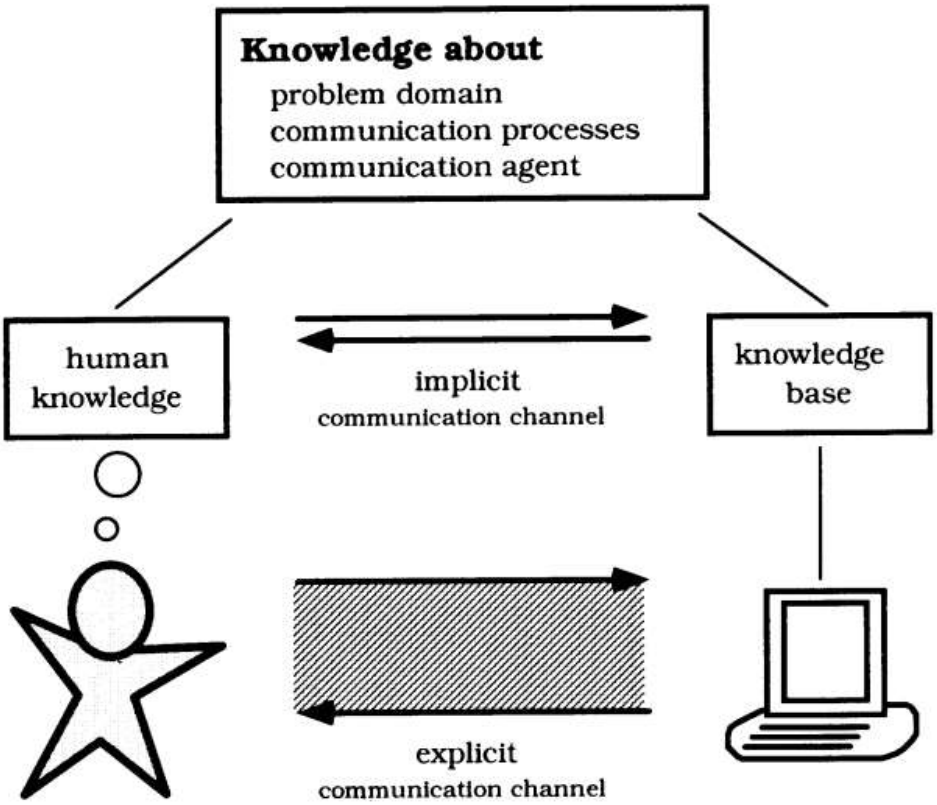


Figure 2.4 – Fischer (2001): Knowledge-based HCI.

After we discussed GOMS models in Section 2.1 as an early traditional milestone in HCI, we began with *User Modeling* discussion in Section 2.3, where we have seen how *User Models* are treated through tracking different *User Experience* attributes during the actual *Use-Time* in HCI sessions. In this section, we continue the discussion of HCI evolution by following Fischer (2001) as a milestone research work in *User Modeling*. Concretely, we present hereby how Fischer et. al. modeled the implicit interactions in HCI.

HCI includes a traditional explicit interaction channel between the two stakeholders (human user and computer application), such as mouse-click and keystroke modeled by GOMS as motor actions (Section 2.1). However, more sophisticated interface techniques, such as windows, menus, pointing devices etc. have widened this explicit communication channel. That is, in addition to the explicit communication channel, Fischer (2001) modeled an implicit communication channel for building knowledge-bases during HCI sessions necessary to ensure smooth, successful interaction between human and computer, see Figure 2.4. This implicit communication channel supports communication processes with a considerable body of knowledge about the problem domain, communication processes, and the agents involved.

- *Problem Domain*: Computer applications can be clustered in different problem domains. For each, shared knowledge can be built as problem domain knowledge-base. This knowledge describes reasonable goals and operations in the domain of specific users, thus, it constrains the number of possible user actions for user modeling purposes. Cognitive adaptive models can follow such problem domain knowledge-bases to narrow the adaptation mechanisms towards the most effective ones. Fischer (1994) and Horvitz et al. (2013) introduced user modeling approaches through modeling such problem-domain interactions and not just Human-Computer interactions.

- *Communication Process*: The communication channel between human user and computer should be modeled as collaborative HCI. That is, a corresponding knowledge-base should enable a decision about when and whether to assist the user, interrupt the user, and volunteer information to the user contextualized to the task at hand, Fischer and Stevens (1987); Horvitz (1999). We can naturally incorporate our approach of user modeling (HCI obstacles detection and UI adaptation) to such communication knowledge-base models. That is, automatic detection of HCI obstacles and corresponding UI adaptation are also introduced for similar purposes, i.e. to decide when and whether to assist the user, interrupt the user, and support the user with corresponding UI adaptation.

- *Communication Agent*: according to Fischer (2001), there is no "typical" user of a system, instead, there are many different kinds of users. The requirements of an individual user usually change with experience Mackay (1991). Moreover, stereotypes of users (Rich (1989)) such as novice, intermediate, or expert users, are inconvenient for complex systems because these attributes become dependent on a particular context rather

than applying to users globally. These factors altogether highlight a great importance of individual user modeling, so-called communication agent modeling by Fischer (2001). The author listed different techniques for individual user modeling such as:

- *Preferences*: to be set by the user Nakakoji (1993).
- *User actions*: to infer what the user really knows and does Fischer et al. (1991); Mastaglio (1990); Adachi (1998); Hill et al. (1992).
- *External events*: to communicate information about external events to the system Bolt (1984); Harper et al. (1992).

The approach of this thesis can be incorporated into this *communication agent* structure for both *user actions* and *external events*: Individual user modeling is applied by detecting HCI obstacles from *user actions* (behavior-based detection), or as *external events* (obstacles) causing changes in user states, which can be detected from physiological (EEG) data.

2.4 Adaptive User Interface

We recapped the evolution story of HCI since early stage GOMS models as a traditional milestone in HCI literature (Section 2.1), and through *User Modeling* research works (Section 2.3) including *User Experience* (Sectio 2.3.1) and *Konowledge-base HCIs* (Section 2.3.2).

Recently, we have been witnessing a revolution in inventing different smart devices in the 2010's, including high-technology adaptive UIs for computer applications. In this Section, we briefly mention three adaptive UIs, which use different data modalities for the adaptation purpose. We highlight common gaps in the recent adaptive UIs, which can be tackled by our approach: multimodal HCI obstacles detection and continuous UI adaptation.

Apart from traditional UIs using mouse and keyboards, new technologies have been continuously evolving enabling new and adaptive UI for computer systems from different data modalities, e.g O'sullivan (2012) and Miller (2020), which are certified as patents in the cognitive adaptive UI field.

O'sullivan (2012) introduced an interactive voice response system wherein users communicate with a computer over conventional telephone lines. As user modeling in this system, a dynamic Audio Output Profile (AOP) for a given user is built during the interaction. The AOP will likely be different for

each user, although users with similar demographics, age, gender, culture and socioeconomic groups may have similar AOPs. Additionally, the AOP may vary for each user during a user-device interaction (HCI session) and for the same user over time. The AOP acts as a dynamic adaptive UI by adjusting the audio speaking rate (words per minute) faster or slower, delivered with alternate inflection, prosody, nuance and speaking volume in accordance with detected responses from the user. Recently, Miller (2020) introduced a tactile glove for HCI, which acts as an adative UI with haptic feedback in case of detected abnormal HCI session using the glove.

Sguerra and Jouvelot (2019) investigated an adaptation method based on real-time tracking of human working memory. Sguerra et al. modeled the human memory based on the Moran process Moran (1958) by maintaining "quanta" numbers (weights) for each stored item. As an HCI task, they used a matching pairs game, tracked the user performance and released an adaptation signal when the performance deteriorates to a value less than a given application-based parameter (no explicit external secondary task obstacle). Thus, the approach by Sguerra and Jouvelot is an explicit performance-based tracking model, which does not consider potential temporal dependencies in behavioral data (user actions) to detect behavioural changes when applying that UI adaptation while an explicit secondary task is present.

All these recent cognitive adaptive UIs depend on the concept of user performance tracking from a specific data modality, but they do not model multimodal HCI obstacles, which can be used for continuous adaptive UI. This is the contribution of our model, which is briefly introduced in Chapter 1, and will be detailed in Chapter 5, where our model is designed for HCI multimodal obstacles detection and dynamic UI adaptation in consecutive HCI sessions.

2.5 Background: Human Cognition Modeling

In the previous sections, we recapped the HCI story since early stage milestone GOMS models in the 80's, until recent cognitive adaptive UIs starting from the 2010's. For GOMS models (80's, Section 2.1), GOMS follow-up research work (80's and 90's, Section 2.2) and user modeling and user experience (2000's, Section 2.3), we discussed how our models can be incorporated. Moreover, for the recent cognitive UIs (2010's until present, Section 2.4), we highlighted gaps of multimodal modeling of HCI obstacles for continuous adaptive UIs. Before concluding the incorporation discussion of our models in Section 2.6,

we discuss in this section background research works in human cognitive modeling.

Human cognitive skills represent a main focus of HCI modeling. Human memory, concretely short-term working memory (WM), takes the highest importance of human cognitive modeling, because WM plays the central role during perception, cognition and commanding motor actions (see Section 2.1.2). Therefore, we discuss background research works in human cognition in general (Section 2.5.1) and human cognitive memory in particular (Section 2.5.2).

2.5.1 ACT Theory

Regarding our models, we decided to use matching pairs game as an exemplary HCI task (Chapter 1). In this section, we discuss a well-established background research work, so-called ACT-R theory, which became a milestone in human cognition and human memory modeling. ACT-R stands for Adaptive Character of Thought Theory - Rational. Briefly, it models the information encoding, retaining, and retrieving processes in WM. ACT theory models stimuli, where frequent and recent stimuli strengthen the information recall. We will see in Chapter 3 how our simulation models incorporate those stimuli, frequency and recency concepts for simulating the playing of memory games. In the following section, we briefly present the evolution story of ACT-R theory as a well-established reference research in the cognitive modeling literature.

2.5.1.1 ACT Theory Evolution

The ACT-R story began with a research work called Human Associative Memory (HAM): Anderson and Bower (1973). In the HAM, the authors introduced their first theory for memory representation using buffers. According to the HAM, information processing operates through buffers, where information from outside stimuli are coded into semantically connected buffers.

Later on, Anderson extended HAM model and published the first work in ACT series (ACT-E): Adaptive Character of Thought Memory (1976). In ACT-E, Anderson detailed the HAM's memory representation theory as *Declarative Knowledge* and *Procedural Knowledge*. The *Declarative Knowledge* is modeled as connected *chunks* (so-called buffers in HAM), while *Procedural Knowledge* is intended to be modeled as rules, so-called production rules. However, how to realize those production rules was still an open challenge in ACT-E model. After seven years, Anderson (1983) further extended his ACT

model, so-called ACT* (pronounced as ACT star). In ACT*, Anderson went several steps closer towards realizing an implementation of ACT model as a computer simulation tool; Anderson psychologically argued methods in ACT* for modeling how production rules might be acquired.

Anderson continued in his cognitive architecture research, and discussed in Anderson (1993) the uncertainty of ACT*. He revised his methodology for understanding cognition through the development of rational analysis of cognition Anderson (1990). Rational analysis incorporates the complexity of human minds through a "black box" problem, in which researches exploit human behaviour for inferring information processing mechanisms. Briefly, a rational solution during cognition means to select the optimal one among many potentials, where human behavior enables reducing the problem space by providing constraints on the gamut of plausible mental mechanisms, Anderson (1990).

Rational analysis was a transition towards a new generation of ACT theory, so-called ACT-R. ACT-R has been continuously discussed, evolved, and thus, consecutive versions of ACT-R have been released from 1993 until the present version ACT-R 7 in 2016. Ritter et al. (2019) has reviewed the history of ACT theory, and discussed in more details its milestones.

2.5.2 Cognitive Memory Model (CMM)

CMM (Pröpper et al. (2011)) is a general computational cognitive model of human memory inspired by the ACT-R theory (Anderson et al. (2004)). It represents an associative memory framework, which provides a way to model dynamic association processes of the human's memory. In HCI domain, the content and context of an interaction are dynamic, i.e. they change over time. Thus, the CMM models those dynamics via associations. Associations are formed as new items become stimulated (come into focus). Thus, such new items in memory will get activated, and consequently integrated with previously active items, while old items might be gradually forgotten (if not re-activated).

The CMM models human memory as a connected graph of *concepts* and *associations* nodes. *Associations* nodes are modeled dynamically to link different *concepts* nodes, which represent objects of common sense knowledge. Pröpper et al. (2011) explained their association modeling with a simple example depicted in Figure 2.5. It shows two concepts: Karlsruhe (a city in Germany), and KIT (an institute in Karlsruhe). If someone knows Karlsruhe city and KIT, then both are modeled as concepts in her or his memory model,

Figure 2.5 – Pröpper et al. (2011): An association example in CMM. Two concepts and an associations of the type "is located in".

where both are objects of common sense knowledge, i.e. both exist in her or his long-term memory. When those concepts get activated during an HCI application, then a dynamically created association node, so-called "is located in", is also modeled in working memory (WM) to link both activated concepts, resulting in a complete context in WM: "KIT-is located in-Karlsruhe".

In CMM, concept nodes are getting stimulated, and thus fetched into WM, or faded from WM. That is, each node in WM in the CMM model has an activation value. The activation value is the likelihood for a node to be activated and fetched into the WM. It also represents an indicator for the amount of time needed for the node to be fetched from long-term memory. A node can get activated in two ways: By an external stimulus or through spreading activation, i.e. propagation through activated connected concept nodes, via those dynamically created association nodes. Briefly, based on frequency and recency of stimuli, e.g. auditory or visual stimuli, parts of WM's graph can be re-activated or faded. Various applications can benefit from the CMM model to work with a cognitive model of human memory. Putze et al. (2015) modeled different game strategies in the matching pairs game based on CMM. We follow Putze et al. (2015) to enrich our data corpus by simulating the playing of the matching pairs games under different conditions (e.g. noObs or memOb), more details follow in Chapter 3.

2.6 Summary: Incorporation of Thesis's Models

Chapter 1 and this chapter present the theoretical part of this thesis, while the remaining chapters present the practical part. While we illustrated the thesis pipeline, models and architecture in Chapter 1, we discussed the state-of-the-art evolution of related research (*HCI*, *User Modeling* and *User experience*) in this chapter. In addition, we discussed background research (ACT Theory and CMM), upon which our models are designed.

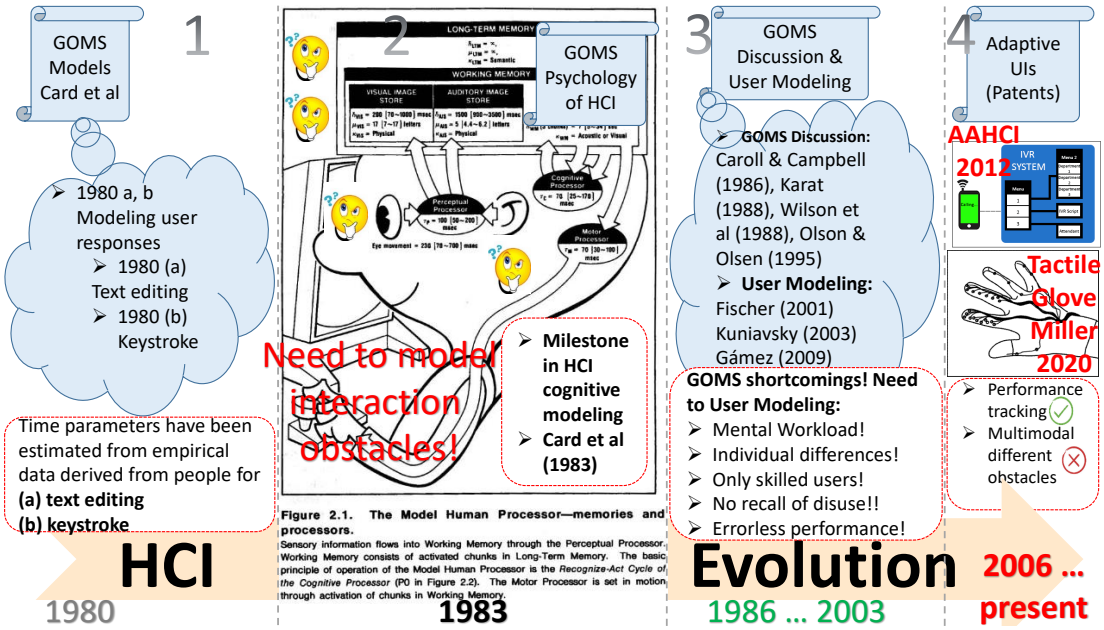


Figure 2.6 – Incorporation of our HCI Obstacle Detectors in the HCI Evolution Story.

For each research milestone discussed, we mentioned how our models can be incorporated. That is, we follow related works and background research milestones and incorporate our models accordingly. As mentioned in Chapter 1, our proposed model can be offered not only as a stand-alone cognitive adaptive system, but also as complementary model for existing adaptive systems to support them with our introduced capability: "Multimodal HCI Obstacles Detection & Corresponding UI Adaptations".

In this section, we summarize how our models can be incorporated in the evolution stories of related and background research milestones. We depict milestones in illustration figures, and briefly mention the main points of each transition and where our models can incorporate.

Figure 2.6 depicts the main discussed milestones in the evolution of HCI:

(1) GOMS models for modeling specific motor actions in HCI (text editing Card et al. (1980a) and keystroke Card et al. (1980b)).

(2) GOMS psychology of HCI was a milestone in the HCI evolution. It comprehensively modeled different processes in HCI: perception (through visual sensory systems using eyes, and auditory sensory system using ears), cognition (through WM and long-term memory) and motor actions. Our

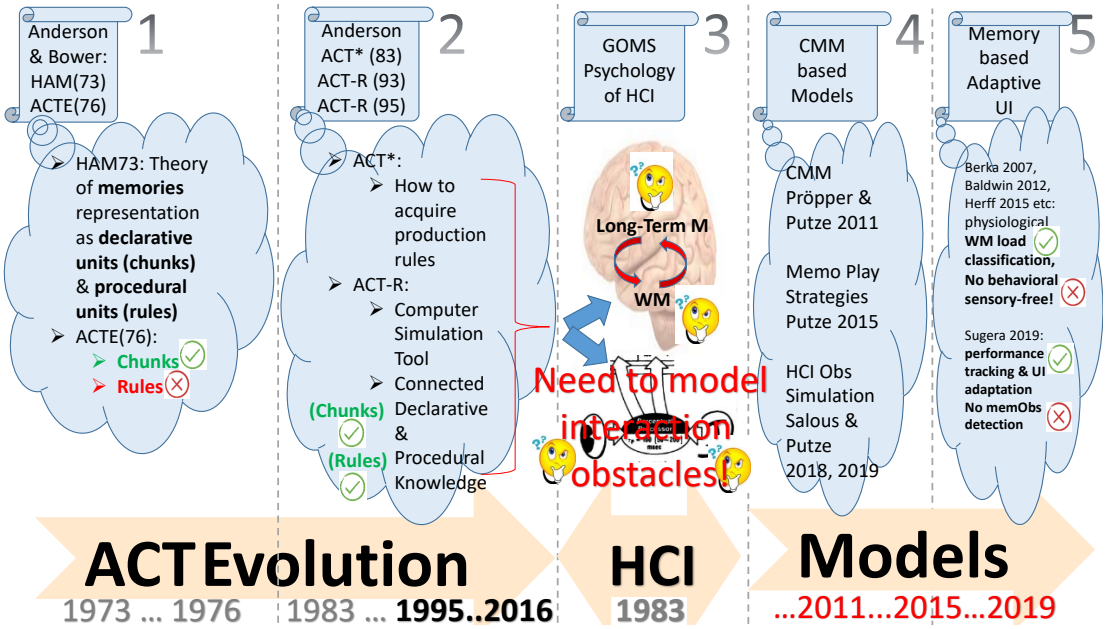


Figure 2.7 – Incorporation of our HCI Obstacle Detectors in ACT, CMM and HCI Evolution Story.

multimodal obstacle detector models can be incorporated here to detect HCI obstacles from multimodal data during perception or cognition processes.

(3) GOMS follow-up research: researchers confirmed GOMS parameters, discussed extensions, but also highlighted shortcomings, for which *User Modeling* research has been evolving (including *User Experience* models). The highlighted shortcomings represent also a requirement for our *User Modeling* approach: *User Modeling* via multimodal obstacle detection and corresponding UI adaptations.

(4) Recently, HCI and *User modeling* have been continuously evolving, and new technologies allowed inventing new adaptive UIs, certified as patents, e.g. Miller (2020); O'sullivan (2012). However, such adaptive UIs mainly track user performance for adaptation. To the best of our knowledge, there is no adaptive approach using multimodal HCI obstacles detection and corresponding UI adaptation, thus, we are contributing, within this thesis, by introducing that capability, as mentioned above, either as stand-alone cognitive adptive system, or as plug-in capability for incorporation in existing cognitive adaptive system.

Figure 2.7 depicts combination of background research milestones (1, 2, 4, and 5) and HCI milestone (3).

As discussed above, the ACT theory (including HAM, ACT-E, ACT* and ACT-R series) has been a milestone extended research in human cognitive modeling and human memory modeling: the HAM introduced first the memory representation theory using *buffers*, which have been then detailed as declarative units (*chunks*) and procedural units (*rules*).

While ACT-E realized the implementation of *chunks*, the acquisition of corresponding *production rules* for those activated chunks was still an open challenge in ACT-E. ACT* tackled that acquisition process of corresponding activated *chunks*. After that, a new generation af ACT, so-called ACT-R has been continuously evolving until the ACT-R7 released in 2016.

In ACT-R series, rational analysis is used to select the most probable method for cognition among many available methods during the human cognition process. Consequently, the concepts of activated *chunks* in WM and corresponding *rules* in long-term memory have been realized in a computer-based simulation.

This evolution of ACT-R series naturally meets the milestone of GOMS modeling (depicted briefly in (3)), where ACT-R WM *chunks* link to those perceived visual and auditory stimuli, which are encoded by GOMS also in WM. In addition, the *cognition process* discussed by GOMS model naturally meets the acquisition of *production rules* in ACT-R theory; That is, fetching corresponding information from long-term memory in GOMS is implemented in ACT-R series (1995-2016) where both activated *chunks* in WM and corresponding *production rules* in long-term memory are realized in a computer-based simulation of human cognition.

In (4), we show the background research works (CMM in 2011 and Memo Play Strategies in 2015), upon which we released our HCI obstacle detector models (2018 and 2019), and recently also UI adaptation models in 2021, as thesis contributions to the HCI evolution.

Finally, we compare our models in (5) with recent adaptive UIs, and we highlight gaps in the state-of-the-art adaptive UIs, which are tackled by our approach: multimodal modeling of obstacle, including behaviour-based sensory-free HCI memObs detectors. For more details about our models' contributions follow Chapter 4.

Chapter 3

Data Collection, Analysis and Validation

In this chapter, we discuss all the user studies, from which we collected data for our models. We discuss and evaluate the validity of the collected data from different perspectives.

In the previous chapters, we discussed HCI exemplary application, obstacles and corresponding UI adaptations. Namely matching pairs as an HCI application, memObs and visObs as HCI obstacles, and memAdapt and visAdapt as corresponding UI adaptations. We concluded to use the matching pairs game as an exemplary HCI application because it mainly relies on human memory and represents also a complex HCI task, which requires users' recognition ability of visual items, Salous et al. (2019). Moreover, we chose to implement the HCI task as a game since these generate a level of intrinsic motivation (Gámez et al. (2009)) and set a natural goal for the user. The game of matching pairs is a typical representation of many HCI tasks as it involves multi-step operations, and many different cognitive functions, including spatial and symbolic working memory, decision making, and planning. These cognitive functions appear prominently in many other HCI tasks, such as machine or vehicle operation, decision support systems, or learning tools.

In this chapter, we aim at simulating and correspondingly collecting data from these different matching pairs variants: noObs, memObs and visObs under different UI adaptations: noAdapt, memAdapt and visAdapt. Actually, alongside this thesis pipeline, we are discussing within this chapter

the *inputs* which we prepare for our ultimate target system. The ultimate target system will be discussed then, in details, within the next two chapters. However, let us briefly introduce it here once again for complete context: it is an online cognitive adaptive system which automatically adapts its UI according to the detected HCI obstacle. For such a behavioral cognitive system, we need to depict HCI user cognitive "behaviour" and "state". Thus, we collect multimodal data from our subjects, namely behavioral data (user actions) and brain activity data (electroencephalography (EEG)). While we expect EEG data to depict user's current state at a specific point of time in HCI, behavioral data enriches HCI user modeling when exploiting potential temporal dependencies between recorded behavioral data items in different time steps.

Regarding user modeling and HCI obstacles, we need to collect data from a variety of subjects and according to different obstacle types: volatile and persistent HCI obstacles, recall Chapter 1 for more details about obstacle types. Thus, we designed multiple user studies and collected HCI multimodal data (brain activity as EEG data and user actions as behavioral data) from a variety of subjects (students, healthy elderly participants, dementia patients). We collected the main data from participants at our lab, where we implemented different variants (noObs game, different obstacle and UI adaptations games) of matching pairs game on an android tablet, so-called TabData data collection (Section 3.1). Given that each participant typically plays only a limited number of games in such experiments, we enriched the collected data by simulating user behaviour during playing those different variants of matching pairs; we name such simulated data as SimData data collection (Section 3.2). Moreover, we implemented the different matching pairs game variants also as a website so we collect data worldwide, and we used this WebData data collection (Section 3.3) especially to collect data from elderly healthy subjects ($age > 60$ years).

In this chapter, we discuss the different methods we used for collecting data: Tablet Experiments (TabExp), Simulation Experiments (SimExp) and Web Experiments (WebExp). For each user study, we describe the experimental design, subjects and data. For TabExp user studies, we discuss additionally the questionnaires given to subjects for a subjective self assessment. Besides these subjective assessments, we calculate and discuss objective assessments; We calculate one-shot measurements as *Static Measurements*, and we plot continuous measurements as *Sequential Measurements* to analyze HCI behaviour. Regarding data, we evaluate the collected data and discuss its validity according to two different perspectives: 1) Data validity according to the different HCI sessions we are simulating: noObs, memObs and visObs sessions. 2) Data validity according to the different UI adaptation mechanisms we apply

as compensation for such HCI obstacles. We complement our discussions by comparing different data episodes using convenient statistical tests.

3.1 TabData Collection

We implemented different variants of the well-established matching pairs game (also called memory game) as an exemplary HCI application. For the purpose of HCI data collection, we only used the single-player mode, in which a matching pairs single-player is asked to find all the pairs with as few turns as possible in each played game variant. We mean with game variants all possible combinations between the designed HCI obstacles (noObs, memObs and visObs) and UI adaptations (noAdapt, memAdapt and visAdapt), e.g. memObs_noAdapt.

Users' actions (the selected cards) while approaching this task continuously indicate their "working memory (WM) status" and "playing behavior": 1) for analyzing a user's WM status, we recorded EEG data from our participants while they were playing the different game variants. 2) In parallel, player behaviour is encapsulated in the recorded behavioral data (sequences of user actions, i.e. the selected cards).

We discuss this section in three main parts: 1) TabExp User Studies, in which we discuss all the user studies we designed between 2017 and early 2020 alongside the thesis pipeline. For each user study, we discuss its experimental design, subjects and their participating consents, and the data collected. We also briefly mention approvals from corresponding ethics committee for such experiments. 2) TabExp Recording Procedures, in which we discuss the recording of the aforementioned multimodal data: behavioral and EEG data modalities. 3) Finally, we discuss in TabExp Assessments both subjective (questionnaires) and objective (measurements) analysis.

3.1.1 TabExp User Studies

In this section, we explain all the TabExp user studies which we designed for collecting data. All the TabExp user studies except the dementia user study (Section 3.1.1.3) took place at our Lab. We designed four TabExp user studies at our lab between 2017 and early 2020, each one was task and/or group oriented; For task oriented user studies, we designed user studies to test memObs, visObs and to test different UI adaptation mechanisms. For subjects oriented user studies, we designed user studies for young healthy subjects (students), in which we tested the memObs, visObs and UI adaptation

mechanisms. We also designed a user study for elderly healthy subjects at our Lab for the dementia detection task, for which we had also data collected from user studies for elderly dementia patients through multiple research projects: I-CARE [1], AKTIV [2], and ASARob [3].

Over these three years (2017-2020), we expanded the studies step by step until we finally reached our goal: Experiments for online (low-latency) cognitive adaptive system. For simplicity, and to gain experience with such experiments, we began in 2017 with memObs user study for only young subjects (mainly students) to detect only HCI memObs: volatile memObs simulated and detected as HCI secondary task memory load (Putze et al. (2018); Salous et al. (2019)) and real persistent memObs detected as low WM capacity for complex HCI task, Salous and Putze (2018), for more details follow Section 3.1.1.1. Then, we extended the experiment in 2018 and 2019 to detect different HCI obstacles and to test corresponding UI adaptations mechanisms, we named those user studies as *Obstacles and UI Adaptations*: we simulated HCI persistent visObs as red-green color vision deficiency (Color blindness, Salous et al. (2019)) and HCI volatile visObs simulated as glare-effects on mobile display. Additionally, we tested different mechanisms of UI adaptations for compensating the introduced HCI memObs and visObs in those user studies (more details in Section 3.1.1.2). After that, we investigated in 2019 the detection of dementia as a real persistent HCI memOb; We already had memory game logs collected from dementia patient via dementia user studies (details in Section 3.1.1.3), however, we needed to collect memory game logs from elderly healthy subjects ($age > 60$, more details in Section 3.1.1.4) to ensure a valid and balanced binary dataset: elderly healthy subjects' logs vs. dementia patients' logs.

As we will see in Section 3.1.1.1 and Section 3.1.1.2, we collected multimodal data (behavioral and EEG data) from these young-subjects user studies. However, we will see in Section 3.1.1.3 and Section 3.1.1.4 that we collected only behavioral data from those elderly-subjects user studies (dementia patients and elderly healthy subjects): on the one hand, it is not that comfortable to setup EEG-Cap and electrodes for elderly subjects, and on the other hand, we aim at detecting dementia (as a real persistent HCI memObs) from natural, sensory-free behavioral data (user actions) as a real-world compatible test.

After this general introduction about TabExp user studies, and before discussing these user studies in details in the following sections, we show

[1] I-CARE: https://www.uni-bremen.de/csl/projekte/abgeschlossene-projekte/i-care
[2] AKTIV: https://www.uni-bremen.de/csl/projekte/abgeschlossene-projekte/aktiv
[3] ASARob: https://www.uni-bremen.de/csl/projekte/laufende-projekte/asarob

hereby brief information about these user studies in the following table, see Table 3.1.

Table 3.1 – TabExp user studies summary: participating subjects in TabExp user studies.

User Study	Users	Thesis Section
memObs (2017)	31 subjects age 19-48	3.1.1.1
Obstacles And UI Adapt.1 (2018)	19 subject age 18-27	3.1.1.2
Obstacles And UI Adapt.2 (2019)	17 subjects age 17-27	3.1.1.2
Obstacles And UI Adapt.3 (2020)	16 subjects age 19-29	3.1.1.2
Dementia User Studies	15 patients mild...severe	3.1.1.3
Elderly Healthy (2019)	6 healthy $age > 60$	3.1.1.4

3.1.1.1 memObs User Study

As mentioned above, we began in 2017 with the memObs user study as a first step towards our ultimate system: cognitive adaptive system. We have seen that Working Memory (WM) strongly controls HCI because WM plays major roles in all HCI cognitive processes: perception, cognition and commanding motor actions, recall GOMS and ACT-R theory discussions in Chapter 2. Therefore, we decided to begin with collecting HCI multimodal data for investigating the detection of HCI memObs: both simulated volatile memObs, and also real persistent memObs. This user study was designed to detect secondary task memory load in HCI as a volatile memObs (Putze et al. (2018)), see a simple illustration in Figure 3.1. In addition, the data collected in this first phase was used also to classify working memory capacity (Salous and Putze (2018)), where low WM capacity during a complex HCI task is considered as real persistent HCI memObs for that complex HCI task, see a simple illustration in Figure 3.2.

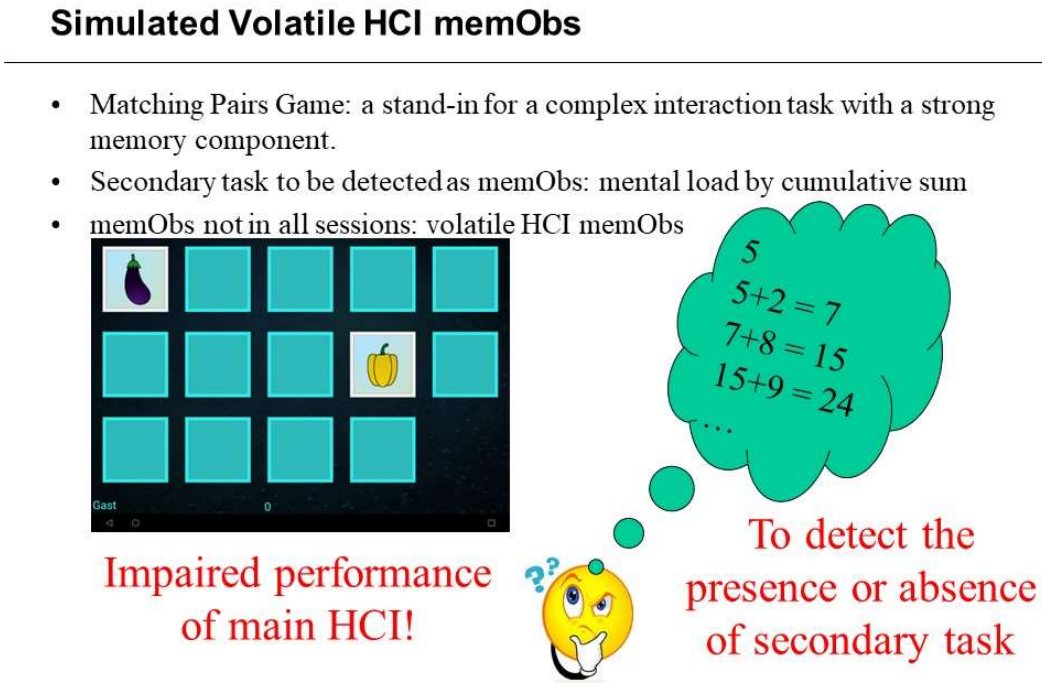


Figure 3.1 – Simulating and detecting HCI volatile MEMOBS: Cumulative sum as an HCI secondary task memory load in matching pairs game.

This first user study took place in 2017 where 31 subjects participated (22 male, 9 female, ages range from 19 to 48). All the participants gave their informed written consent. They were compensated with 10 Euros for their participation. The data collection was approved by the ethics committee of the University of Bremen. For each game variant in this experiment, brain activity is recorded as user states: *EEG data modality*, and the selected cards are logged as user actions: *Behavioral data modality*. More details about the recording procedure will be explained in Section 3.1.2. The experiment lasted about 90 minutes per subject including the EEG cap and electrodes setup, trial games and real recording games. The participants were given description about the experiment, moreover, they played additionally one trial game for each game variant to get familiar with the experiment. A specific questionnaire for each game variant has been answered by participants after finishing the corresponding game (All details about questionnaires as subjective assessments are discussed in TABEXP Assessments, Section 3.1.3). In this experiment, the participants were asked to play only two game variants in a randomized order, namely NOOBS and MEMOBS. Additionally, they were

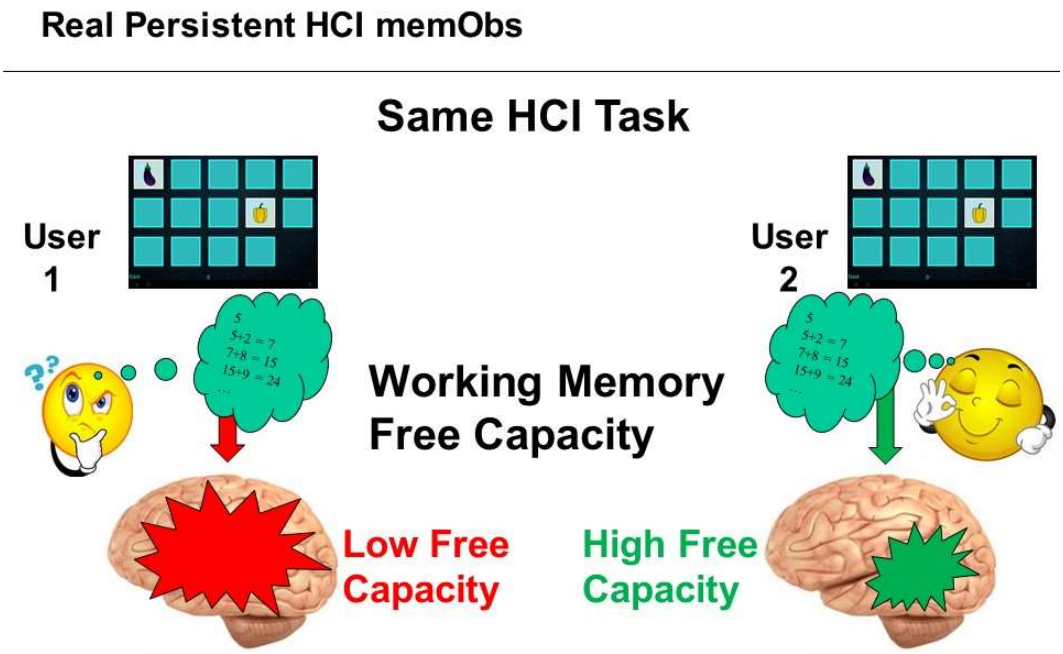


Figure 3.2 – Detection of real persistent HCI memObs: low WM capacity in a complex HCI task.

asked to do the standard MU test as a WM capacity test.

- No Obstacle (noObs): Matching Pairs game without any obstacles; The participants were asked to play the game without any parallel secondary task. Player's best performance is expected in this game compared to other games with different simulated HCI obstacles, i.e. shortest sequence of rounds is expected to finish this game.

- memory-based Obstacle (memObs): Matching Pairs with secondary task memory load as simulated volatile memory-based obstacle; Whenever the participant reveals a card, a random number between 1 and 9 will be spoken by the synthesized voice. The participant was asked to calculate and memorize the sum of all spoken numbers throughout the game. In addition to the above mentioned recorded EEG and behavioral data modalities, the cumulative sum, calculated and reported by participant, is also logged to evaluate both the main task performance (matching pairs) and the secondary task performance (cumulative sum). We expect that under this memObs condition, the performance in the main task is significantly lower as in the noObs condition.

- WM Capacity Test: To get ground truth for our low WM capacity detection task, we used standard approaches to realistically measure WM capacity of our participants. That is, WM capacity is a key to successful HCI, where WM capacity is strongly correlated to general intelligence, Conway et al. (2003). WM is relevant to preserve attention and control in all cognitive interactions. In their Cognitive Load Theory, Sweller et al. (2011) suggest that overloading a user's WM leads to deteriorated performance and requires system adaptation. Moreover, WM capacity is highly related to reasoning ability, Süß et al. (2002). For all those reasons, we implemented the Memory-Update (MU) test, Lewandowsky et al. (2010), to realistically assess the WM capacity of our subjects. That is, Lewandowsky et al. (2010) argued their MU test as a standard WM capacity estimator which has high loading on WM capacity factors. The collected MU logs will be used then as references (ground truth) in our low WM capacity detection task. Such a detection of low WM capacity is considered as a *persistent* memObs for the corresponding "complex" HCI task. As differences in WM capacity are more likely to be revealed when WM demand is high, we decided to detect low WM capacity for Matching Pairs with cumulative sum as complex, high memory load demanding HCI exemplary task, see Figure 3.2
  MU application runs multiple trials for each participant. In each trial,

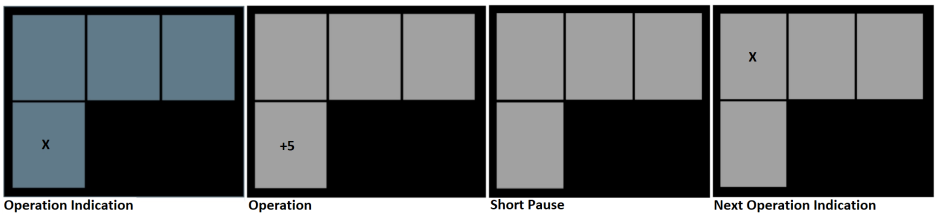


Figure 3.3 – Memory Update Test (MU): Four consecutive steps in one trial.

the application initializes a set of digits, each presented in a separate frame on the screen. Participant should memorize the presented digits. Then, the MU application indicates a random frame with an "x" symbol, and follows that frame with a simple arithmetic operation (adding or subtracting one digit). Subsequently, participant should update the initial digits in mind and memorize the final results. Figure 3.3 shows screenshots of four consecutive steps in one trial.

3.1.1.2 Obstacles and UI Adaptations

In addition to the volatile and persistent HCI memObs discussed above, we simulated another important type of HCI obstacles in the matching pairs game: visual obstacles. Similar to HCI memObs, we simulated also volatile and persistent variants of HCI visObs: red-green color vision deficiency as persistent visObs and glare-effects as volatile HCI visObs. Moreover, we implemented multiple UI adaptations as compensation for those different HCI obstacles (recall the importance and arguments of such HCI obstacles and UI adaptations in Chapter 1).
We designed different matching pairs game variants according to those two perspectives: obstacles and UI adaptation mechanisms. For obstacles, we have three different obstacle game variants: noObs, memObs and visObs. For UI adaptation mechanisms, we have also corresponding UI adaptation game variants: noAdapt, memAdapt and visAdapt. To simulate different interaction obstacles in combination with different types of (appropriate or inappropriate) adaptation, we developed several game variants which are designed to represent such situations. In total, we worked with nine different game variants, which result from the combination of interaction obstacles (including no obstacle) and adaptation (including no adaptation). These combinations include variants of successful adaptation, but also variants in which an adaptation will likely not support the user or even further impede their progress. Such situations can occur during the actual system operation when an initial decision of the system to trigger an adaptation is incorrect. It is important for the system to correct such mistakes pro-actively, to avoid confusion and the necessity of manual intervention. Therefore, we included all the 9 combinations in our user study in 2018. In addition, we investigated and assessed (subjectively and objectively, details in Section 3.1.3) different variants of memAdapt in another user study in 2019: LiMemAdapt (slight changes to UI) and StMemAdapt (strong changes to UI). Finally, we designed a volatile visObs (glare-effects) in another user study in 2020.
In each user study, young participants (mainly students) were asked to play multiple game variants in a random order; 19 participants participated in 2018 , 17 participants in 2019 and 16 participants in 2020. Similar to the memObs user study in Section 3.1.1.1, participants in these user studies were also given descriptions about all the game variants, they played one trial game for each to get familiar with the experiment while we were setting up EEG cap and electrodes to collect EEG data besides the logged selected cards as behavioral data. The experiment lasted per participant about 150 minutes. The participants were compensated with 15 Euros for their participation, they gave their consents to participate, and the experiment was approved by

the ethics committee at the university of Bremen, too.
Card pictures are represented by uniform colors. We made this choice to avoid semantic correlates that make memorization of some cards easier or more difficult than others. The chosen colors where: red, orange, green, turquoise, yellow, blue, and pink (All participating subjects were healthy, with no color-blindness, i.e. no red-green color vision deficiency). The implementation of the task was based on an open source memory game[4] which we customized with experiment-specific cards as well as the ability to toggle different interaction obstacles and adaptation mechanisms (see below). Adaptation mechanisms only switch between HCI sessions (i.e. completed games, which are usually short, with a duration of 1-2 minutes), not within one HCI session (i.e. not during the played game).
We describe the different game variants used in those user studies in the following bullet point list, where these game variants were presented and played by participants in a randomized order:

- NOOBS_NOADAPT: No Obstacle No Adaptation: The cards of this game show well-distinguishable colors, See Figure 3.4.

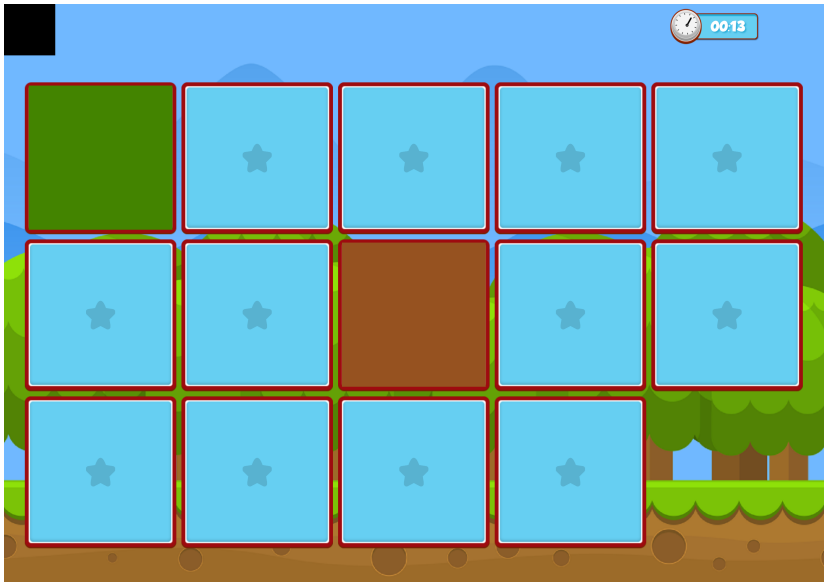


Figure 3.4 – Cards in NOOBS game show well-distinguishable colors

- MEMOBS_NOADAPT: Memory Obstacle No Adaptation: The same game as above but overlaid by a secondary task (cumulative sum) to emulate a memory-based obstacle in matching pairs task: Whenever the

[4] `https://github.com/sromku/memory-game`

participant reveals a card, a random number between 1 and 9 will be spoken by the synthesized voice. The participant is asked to calculate and memorize the sum of all spoken numbers throughout the game.

- VISOBS_NOADAPT: Visual Obstacle No Adaptation: in contrast to the NOOBS game variant, the cards of this game only show different shades of brown color to emulate red-green color vision deficiency as a persistent visual obstacle game variant. See Figure 3.5.

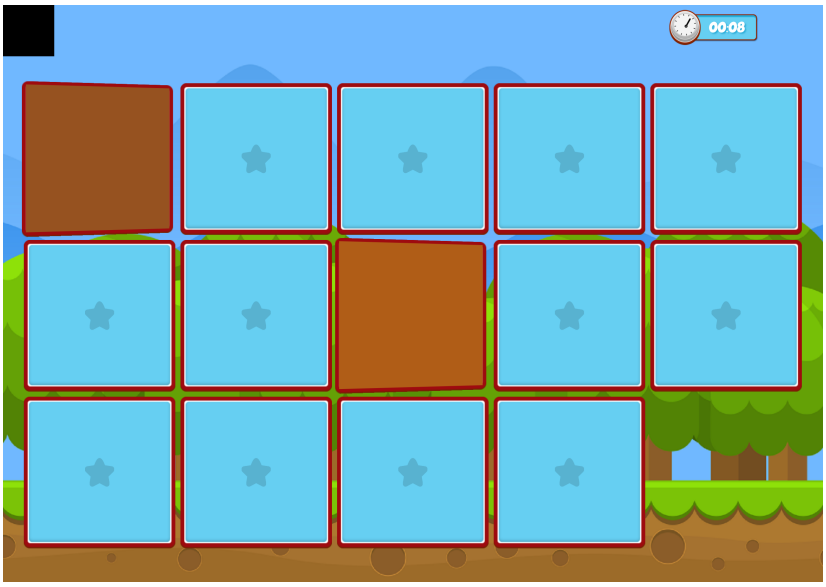


Figure 3.5 – Two (actually different) cards in the VISOBS variant, simulating red-green color vision deficiency

- GLAREOBS_NOADAPT: Glare Obstacle No Adaptation: similar to the NOOBS game variant, the cards of this game theoretically have well-distinguishable colors as motives, however, we simulate an outdoor use of mobile display where sunlight can shortly impede the recognition of UI visual items, see Figure 3.6.

- MEMOBS_STMEMADAPT: Memory Obstacle Strong Memory Adaptation: The same memory-obstacle game, but the participants are supported with a `strong UI adaptation`, where all previously revealed cards are re-revealed whenever a player selects non-matching cards.

- MEMOBS_LIMEMADAPT: Memory Obstacle Light Memory Adaptation: The participants of memory-obstacle game are supported with a `light UI adaptation`: Whenever a player reveals non-matching

Figure 3.6 – Cards theoretically have well-distinguishable colors as motives, but simulated sunlight causes glare obstacle.

cards, only the last two cards revealed in the round before are re-revealed. Showing 4 cards in a very short time frame might allow the player to pick out a pair easier.

- visObs_VisAdapt: Visual Obstacle Visual Adaptation: The same visual-obstacle game, but the participants are supported with an additional channel assistance: *voice assistance*. While simulated red-green color vision deficiency cards make it difficult for players to distinguish the cards pairs, an identifier letter is spoken by the synthesized voice for each revealed card. Thus, seven different audio-identifier letters were needed to be spoken to help the user distinguishing between the seven pairs of cards in this 14-cards game. In our implementation, the German letters $a = /a : /$, $c = /tse : /$, $j = /yot/$, $q = /ku : /$, $x = /iks/$, $v = /fow/$ and $l = /ell/$ were chosen as voice cues, as they are short and easily distinguishable in the German pronunciation. With such spoken identifiers, the participant can compensate the red-green color vision deficiency by mapping the spoken letters to their corresponding positions. Thus, we complement the impaired visual memory by stimulating the unimpaired auditory memory, that is, psychological literature works e.g. Crowder (1972) highlighted complementary roles between auditory memory and visual memory during the perception and cognition processes, therefore, the detection and compensation of such

HCI obstacles is so important for successful perception and cognition in HCI.

- memObs_VisAdapt: Memory Obstacle Visual Adaptation: In this game variant the UI with memory-based obstacle is adapted but with unsuitable adaptation (Voice identifiers). That is, the spoken voice identifiers will interfere with the spoken random numbers (emulated memory-based obstacle) and this consequently would further impair the interaction performance. Detailed assessments are discussed in Section 3.1.3.
- visObs_MemAdapt: Visual Obstacle Memory Adaptation: In this game variant the UI with visual obstacle is adapted but with unsuitable adaptation (Replay). While replay is useful in general to ease the matching pairs task, it would not be that useful with this emulated visual obstacle (very similar cards with only different shades of brown color).
- noObs_MemAdapt: No Obstacle Memory Adaptation: In this game variant the UI has no obstacle, however, it is adapted with the memory adaptation (Replay).
- noObs_VisAdapt: No Obstacle Visual Adaptation: In this game variant the UI has no obstacle, however, it is adapted with the visual adaptation (Voice identifiers).

3.1.1.3 Dementia User Studies

As mentioned above, we had collected matching pairs logs from elderly dementia patients through multiple research projects. In those projects, the researchers aimed not only to collect data from dementia patients, but also to activate their WM through different activation memories as motives, Schultz et al. (2018), see Figure 3.7 as an activation session. The game used was similar to that noObs game used in memObs user study (Section 3.1.1.1), however, only behavioral data were collected, and six pairs (12 cards) instead of seven pairs were used in the game.

For *Dementia User Studies* in this thesis, we filtered the collected matching pairs logs and used only those complete logs from dementia patients to investigate the detection of dementia as a persistent HCI memObs. We also used the medical diagnoses and reports about the participated patients for further detailed analysis. The dementia patients in our study were diagnosed in three different levels: mild, moderate or severe dementia. Additionally, the participating patients were diagnosed according to different dementia types of

Figure 3.7 – Activation session in I-CARE project for dementia patients, Schultz et al. (2018).

disease, such as Alzheimer, Korsakoff syndrome and vascular dementia. We filtered all the logs and took only logs from the patients who completed their games (15 patients).
To analyze the collected behavioral data, we calculated a sequential performance measurement by averaging the number of matching pairs found per round and plotting it as percentage per round, so-called *MatchingPairs percentage per round*. Patients played multiple games, however, we noticed that some patients played many more games than others, we name those subjects *Dominant Subjects*. Thus, if we simply average the calculated sequential measurements (MatchingPairs percentage per round) without considering such big differences in the number of played games, the results will be biased to those *Dominant Subjects*; Therefore, we normalized that sequential measurement by calculating the patients average rather than logs average. If *Dominant Subjects*, for a specific dementia group, show lower performance than other subjects' performance, the non-normalized logs average will be biased to such a (non-representative) low performance behaviour, e.g. Figure 3.8 shows that such a normalized average performance (patients_avg) of Alzheimer dementia patients is higher than the non-normalized log-based average performance, this is because the Alzheimer *Dominant Subjects* showed lower performance than other subjects' performance in the Alzheimer group. In contrast, If the dominant subjects, for a specific dementia group, show higher performance than other subjects in that group, the non-normalized logs average will be biased to such a (non-representative) high performance behaviour, e.g. Fig-

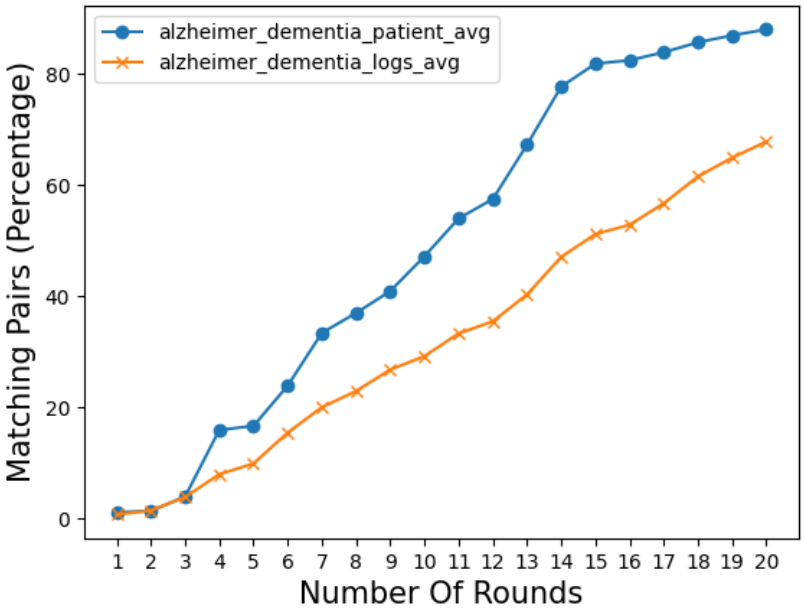


Figure 3.8 – Alzheimer Dementia: normalized performance (patients_avg) vs. non-normalized performance (logs_avg) which is biased to lower performance shown by the *Alzheimer Dominant Subjects*, who played many more games than other Alzheimer patients.

ure 3.9 shows that the normalized moderate dementia performance is lower than the non-normalized performance! In either case, we normalized the calculated average performance by considering subject-based average rather than log-based average, i.e. we calculated the final average in two steps: 1) we calculated the average performance for each subject separately, 2) then, we calculated the final average as average of subjects' averages.

3.1.1.4 Elderly Healthy User Study

For the dementia detection from matching pairs game, we also collected data from elderly healthy subjects. Both dementia user group and this elderly healthy user group act as corresponding user groups in the dementia detection task. Thus, for this user study, we used the exact setup mentioned in the dementia user study, i.e. the same cards' pictures, six pairs and only behavioral data collection. Only 6 old healthy subjects ($age > 60$) participated in this user study at our lab, therefore, we extended this experiment within the

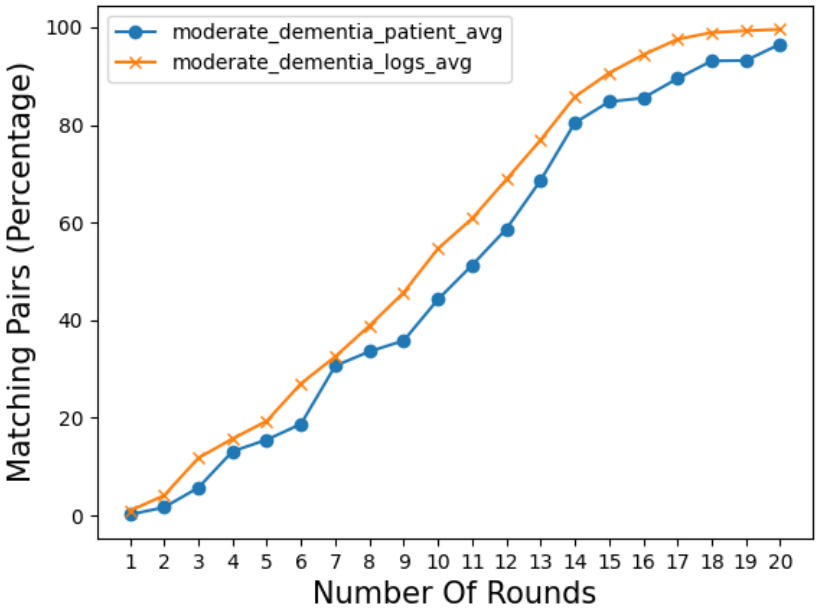


Figure 3.9 – Moderate Dementia: normalized performance (patients_avg) vs. non-normalized performance (logs_avg) which is biased to higher performance shown by the *Moderate Dementia Dominant Subjects*, who played many more games than other moderate dementia patients.

WebExp (Section 3.3) to collect memory game logs from healthy senior subjects worldwide, see Section 3.3.

3.1.2 TabExp Recording Procedures

In this section, we discuss our procedure for recording behavioral data (user actions) and brain activity EEG data (user state). All recorded data, along with experimental paradigm, are available through Open Science Framework under the InteractionObstacles-MatchingPairs project: `https://osf.io/bsudk/`.

3.1.2.1 Behavioral Data Recording

For recording behavioral data, we encoded the participant's actions in the app, i.e. the revealed cards. Each card is represented by two features: The first one encodes its position in the revealing order of motives; the second feature encodes the position of the card in the corresponding pair. For example: the

first revealed card is always encoded with the feature vector (1,1), If the second revealed card shows the same motive, it is encoded with the vector (1,2) (second card of the first motive), otherwise with the vector (2,1) (first card of the second motive). The advantage of this feature representation is that it is invariant to the actual motives but still captures the temporal relationships within a sequence. In addition to that card code feature, we calculate the following features – for each time step (revealed card) – which summarize for a game how efficient and close-to-optimal a player performed: 1) Number of cards left in the game, 2) Number of never revealed cards in the game, 3) Maximum number of times revealing the same card, and 4) Number of turns since game completion (= 0 if game not yet completed).

3.1.2.2 EEG Data Recording

Figure 3.10 – EEG recording setup

Human brain has an electrical activity which shows different characteristics in different brain cognitive processes as well as different types of human behavior; that is, sleeping shows different brain activity compared to walking, memorizing, objects recognition, mental calculations, etc Kumar et al. (2016). To record such a brain electrical activity, electroencephalography (EEG) is typically used as an electrophysiological monitoring method. It is typically a noninvasive method, i.e. specific sensors so-called electrodes will be placed along the scalp. Electrodes are conductors which transfer observed signals from brain to a machine so-called EEG-amplifier, which in turns measures such data and records it on the connected computer using specific recorder software. "EEG measures voltage fluctuations resulting from ionic current within the neurons of the brain", Niedermeyer and da Silva (2005). Concretely, in our

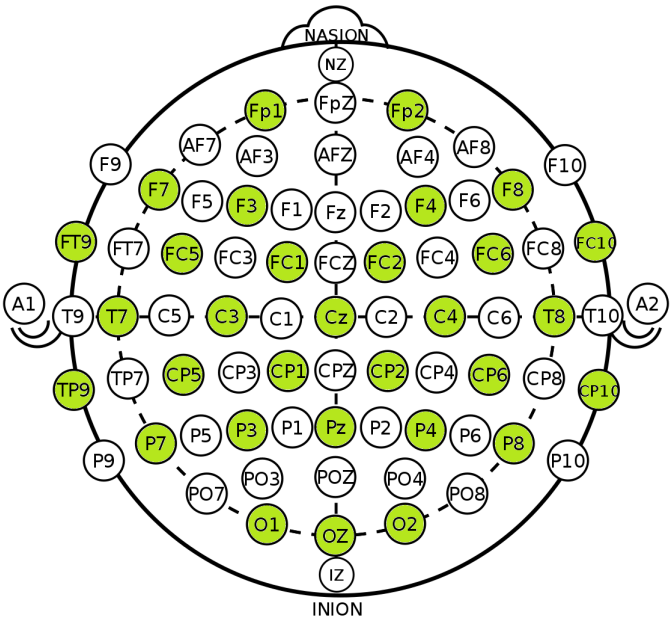


Figure 3.11 – EEG electrodes positions: green colored positions according to the 10-10 positioning system

EEG recording setup, we used a BrainProducts actiCHamp recorder with a 32 electrode actiCAP, see Figure 3.10 as a real experiment photo. The electrodes were positioned according to the international 10-10 system. The reference electrode was positioned at the location Fz, while we used these standard ActiCap positions for the other 31 electrodes: Fp1, Fp2, F3, F4, F7, F8, FC1, FC2, FC5, FC6, FT9, FC10, Cz, C3, C4, T7, T8, CP1, CP2, CP5, CP6, TP9, CP10, Pz, P3, P4, P7, P8, Oz, O1, O2, see Figure 3.11. The sampling rate used was 500Hz, and the impedance was below $10k\Omega$.
After recording EEG data, we have to analyze it, this analysis includes EEG

data pre-processing and patterns detection (presence of different *HCI obstacles* in our case). EEG data pre-processing can be categorized into two domains: time and frequency, Lotte (2014). For time domain, one should focus on event-related potentials (ERP) and investigate potential fluctuations time locked to an event, such as "stimulus onset", e.g. card revealing in our HCI tasks. For frequency domain, we analyse the shape of neural oscillations that can be observed in EEG signals; that is, EEG signals show different shapes in frequency domain under different conditions. We preferred the frequency domain rather than the ERP time domain, because of possible overlapping between multiple stimuli in time domain, e.g. card picture stimulus and spoken number stimulus in case of memObs. Thus, we discriminated EEG signals based on expected different neural oscillations under different conditions, e.g. noObs or memObs. For more details about EEG data pre-processing and discrimination, see Chapter 4.

3.1.3 TabData Assessments

In this section, we discuss subjective and objective assessments for the aforementioned TabExp user studies.
For subjective assessment, we formulated specific questionnaires for different played game variants. That is, in most TabExp user studies (all except dementia and old healthy ones), participants were asked to answer a likert-scale questionnaire after each played game. In each questionnaire, we formulated questions for two main purposes: 1) To let the participants assessing their own performance in that played game. 2) To check the validity of the different applied UI Adaptations by assessing participants' acceptance of those UI Adaptations.
For the objective assessment, we measured the participants' actual performance by calculating static measurements and analyzed their behaviour under different played game variants by plotting sequential measurements. On the one hand, Static measurements measure the participant's performance as one-shot value at the end of the game, concretely we calculate the Time Needed, Turns Needed and Errors made; An error in these games is considered when a player turns a card whose partner has been already seen before, but fails to pick up the pair. On the other hand, we continuously analyze the participant's behaviour using sequential measurements: MatchingPairs per round counts the pairs found per round, and Penalties per round counts the aforementioned errors per round.
After we collected both subjective and objective assessments, we need to statistically analyze those data. Following recommendations in the statistics literature, e.g. Corder and Foreman (2011), Wasserman (2013) and Du Prel

Table 3.2 – The main game variants (without adaptations) in comparison: average and standard deviation for questionnaire responses and game metrics.

Question	**noObs.**	**memObs.**	**visObs.**
Mental demand	3.8 (1.6)	6.4 (0.7)	4.2 (1.4)
Assessment of speed	6.0 (1.0)	3.1 (1.8)	4.7 (1.2)
Card memorability	6.1 (0.8)	3.4 (1.6)	3.9 (1.7)
Time needed [s]	41.8 (15.9)	134.4 (49.5)	60.4 (22.9)
Turns needed [#]	13.8 (2.8)	16.6 (3.1)	16.3 (2.9)
Errors made [#]	2.3 (2.3)	5 (3.1)	4.5 (3.1)

et al. (2010), we use the non-parametric Wilcoxon signed-rank test to analyze the responses of the likert-scale questionnaires, and we use the parametric paired t-test (Student t-test) to realistically analyze the aforementioned objective measurements; That is, such objective measurements constitute independent variables, which are assumed to follow a normal distribution with equal variances, calculated for convenient number of subjects (as recommended $individuals > 20$) and their theoretical frequencies are not less than 5 among the subjects.

3.1.3.1 Subjective Analysis

In this section, we discuss the questionnaire responses answered by subjects after different played games. As mentioned above, we formulated specific likert-scale questionnaires, which were given to subjects after each played game; The target is to let our subjects assessing their own performance and acceptance of different applied UI Adaptations.
A specific questionnaire for each game variant mentioned above was given to the participant when she or he finished that game variant. The important statements to be examined were "The game was mentally demanding", "I think I played through the game quickly" and "I could memorize the position of the cards well" as well as "I could sum the numbers together without problems" and "The assistance was helpful" for different game variants. For each statement, participants could assign a score between 1 and 7, with 7 being the maximum approval (Likert scale). Table 3.2 (upper half) shows that according to mental demand, assessment of speed and card memorability questions, the players found the standard game to be the easiest, the visual obstacle game to be slightly harder and the memory-based obstacle game to be significantly

[5]The experiments have been done in two separate phases: memObs_noAdapt vs. memObs_LiMemAdapt and memObs_noAdapt vs. memObs_StMemAdapt. This explains why memObs_noAdapt does not show the same results in both comparisons.

Table 3.3 – UI Adaptations in comparison: average and standard deviation for questionnaire responses and game metrics.

Question	Without assistance	With assistance
memObs_noAdapt vs. memObs_LiMemAdapt		
Mental demand	6.4 (0.7)	6.4 (0.7)
Assessment of speed	3.1 (1.8)	3.7(1.7)
Card memorability	3.4 (1.6)	3.5(1.5)
Ability to sum	3.5 (1.6)	3.5 (1.5)
assistance helpfulness	-	5.1 (2.0)
Time needed	134.4 (49.5)	119 (40.4)
Turns needed	16.6 (3.1)	14.8 (2.3)
Errors made	5 (3.1)	3.2 (2.4)
memObs_noAdapt vs. memObs_StMemAdapt[5]		
Mental demand	6.1 (0.8)	5.5 (1.1)
Assessment of speed	4 (1.3)	3.8 (1.5)
Card memorability	3.7 (1.2)	4.6 (1.6)
Ability to sum	3.3 (1.2)	3.4 (1.3)
assistance helpfulness	-	5.4 (1.5)
Time needed	110 (22.9)	159 (41)
Turns needed	14.6 (1.9)	13.4 (1.6)
Errors made	5.4 (3.1)	2.9 (2.1)
visObs_noAdapt vs. visObs_visAdapt		
Mental demand	4.2 (1.4)	3.7 (1.6)
Assessment of speed	4.7 (1.2)	5.9 (1.2)
Card memorability	3.9 (1.7)	6 (1.3)
Assistance helpfulness	-	6.7 (0.9)
Time needed	60.4 (22.9)	38.5 (14)
Turns needed	16.3 (2.9)	12.3 (3)
Errors made	4.5 (3.1)	1.4 (2.5)

harder (with $p = 0.25$, $p = 0.001$, $p < 0.001$ for the NOOBS_NOADAPT vs. VISOBS_NOADAPT three comparisons respectively, and $p < 0.001$ for all the three comparisons of NOOBS_NOADAPT vs. MEMOBS_NOADAPT).
Table 3.3 shows comparisons between each game with obstacle and its assisted game variant, e.g. MEMOBS_NOADAPT vs. MEMOBS_LIMEMADAPT. When comparing the MEMOBS_NOADAPT to MEMOBS_LIMEMADAPT game variants, we found no significant differences between the players assessments of their performances, nor of their assessment of completing the arithmetic task more easily (p-values regarding the metrics order in the table: $p = 1$, $p = 0.27$, $p = 0.07$, $p = 1$). However, the MEMOBS_STMEMADAPT game variant shows a significant difference in players' performance assessment (mental demand and card memorability questions).
With the VISOBS_NOADAPT vs. VISOBS_VISADAPT game variants, however, the players found the corresponding voice-assistance to be very helpful, giving it the highest average score in that regard. They also felt they played faster ($p = 0.005$) and could memorize the cards more easily ($p < 0.001$). Only the question concerning mental workload found no significant difference ($p = 0.23$).
UI ADAPTATION VALIDITY DISCUSSION: the subjective analysis already discussed shows high acceptance from our subjects regarding the VISADAPT and STMEMADAPT, while it shows that subjects were not sure whether the LIMEMADAPT really helps. For the VISADAPT, such high subjective acceptance also matches psychological cognitive literature, e.g. Crowder (1972), which highlights the complementary roles between visual and auditory memory; that is, the VISADAPT auditory instructions complements the visual motives to significantly improve the performance in matching pairs game, even in presence of red-green color vision deficiency.
For the MEMOBS, although subjects found that STMEMADAPT significantly improves their performance, it is indeed time consuming and it brings significant changes to the UI, making it a new UI rather than gradual online adaptation. Therefore, we have to slightly modify the LIMEMADAPT to get a valid UI ADAPTATION.
UI ADAPTATION VALIDITY IMPROVEMENT: as a followup to the validity discussion, we introduce hereby a new MEMADAPT as an improved version of the LIMEMADAPT. We avoid potential distraction happens especially during animating the re-revealed cards; that is, we still re-reveal only the last two cards when revealing a non pair, however, we avoid animation when reminding the user with those cards, instead, we only blink them shortly ($400ms$); There are three psychological visual recognition-based reasons for such an adaptation, which we induced from psychological visual recognition works such as Grill-Spector and Kanwisher (2005): 1) Object visual recognition

follows a sequence of steps, each lasts a specific time. 2) While the animation of revealed cards is a part of the original game, it impairs the intended reminding of old seen cards when it is used during the re-revealing adaptation, because it will be recognized before the actual stimuli, i.e. the color of the card. 3) Given that $200ms$ is the minimum time needed to recognize the color object (recall GOMS model discussions in Chapter 2), we only blink those last two cards showing their contents without such a confusing animation for a convenient time slice for recognition: $2 * 200 = 400ms$.
Consequently, we implemented this new UI adaptation game variant, MEMOBS_MEMADAPT, which is similar to the MEMOBS_LIMEMADAPT except blinking rather than animating last two revealed cards. We evaluated the new implemented UI adaptation by designing a new user study in January 2020 and collecting subjective assessments from participants.
UI ADAPTATIONS INTERACTION EFFECTS DISCUSSION: while we still find no significant difference in the mental load self assessment between MEMOBS_NOADAPT and MEMOBS_MEMADAPT ($wilcoxon\_stat = 30.0, p = 0.458$), we analyze the validity of UI adaptations by averaging the mental load responses from subjects and comparing the compatibility of different UI adaptations applied to different HCI obstacles, see Figures 3.12, 3.13, and 3.14.

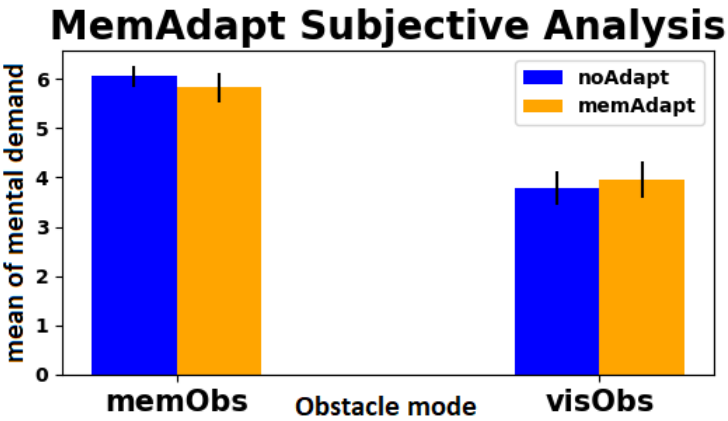


Figure 3.12 – MEMADAPT Subjective Analysis: Bars show the average of participants' self assessment under different conditions, while whiskers show the standard deviations of such averages. MEMADAPT is suitable for MEMOBS, as it improves the MEMOBS performance and deteriorates the VISOBS performance.

Figure 3.13 shows high suitability of the implemented VISADAPT for VISOBS, where it significantly improves the VISOBS performance ($wilcoxon\_stat = 8.0, p = 0.0006$), and slightly impair the MEMOBS performance ($wilcoxon\_stat = 9.0, p = 0.0832$). Figure 3.12 shows also suitability of the implemented

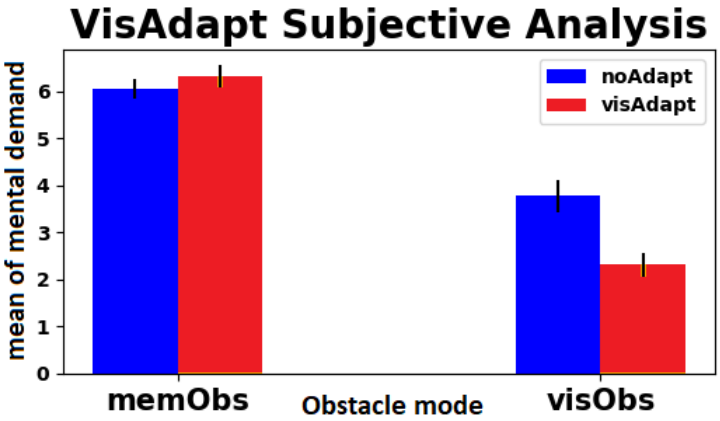
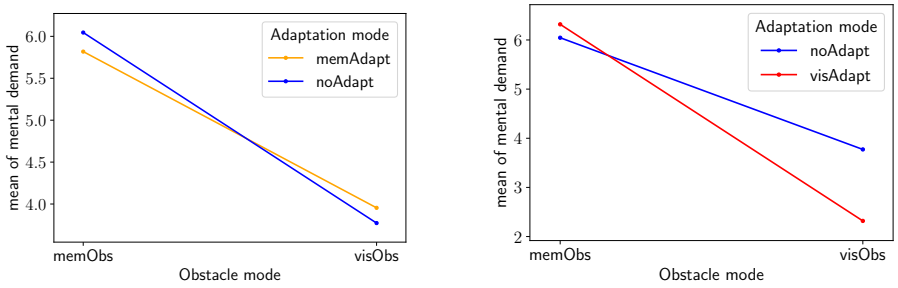


Figure 3.13 – VisAdapt Subjective Analysis: Bars show the average of participants' self assessment under different conditions, while whiskers show the standard deviations of such averages. VisAdapt is suitable for the visObs, as it improves the visObs performance and deteriorates the memObs performance.

memAdapt for memObs, but less significant than Figure 3.13; That is, Figure 3.12 shows that the implemented memAdapt improves the performance of memObs and deteriorates the performance of visObs, but neither with significant difference: $(wilcoxon\_stat = 30.0, p = 0.458)$ and $(wilcoxon\_stat = 58.5, p = 0.612)$ respectively. At least, Figure 3.12 shows that the implemented memAdapt suits the memObs more than visObs.
For analyzing interaction between different interaction obstacles and UI Adaptations, we look at the effects of UI Adaptations when applied to different interaction obstacles regarding the self-assessment of playing performance. We see significant main effects of both obstacle mode (as expected, because especially the memory obstacle has a strong impact on the mental demand) with $F = 131.49$, $p = 2.7e^{-21}$, and $\eta = 0.47$ as well as for adaptation mode (which shows that adaptation is effective) with $F = 2.67$, $p = 0.007$, and $\eta = 0.02$. Besides, we observe an interaction effect ($F = 7.67$, $p = 7.2e^{-4}$, and $\eta = 0.05$), which shows that indeed memAdapt is helpful to support the user in case of a memObs (i.e., it reduces mental load), but is detrimental otherwise. The same holds for visAdapt. Figure 3.14 illustrates the interaction effect. We observe similar effects for other usability metrics, such as playing speed, number of revealed cards, or self-assessment of playing performance.



(a) MEMADAPT (b) VISADAPT

Figure 3.14 – Interaction between adaptation mechanisms and interaction obstacles for experienced mental demand.

3.1.3.2 Objective Analysis

Besides the self assessment of performance discussed in the subjective analysis, we calculate the actual performance of participants using STATIC MEASUREMENTS, and we analyze their playing behaviour using SEQUENTIAL MEASUREMENTS.

STATIC MEASUREMENTS: The game logs recorded the total TIME NEEDED and the TURNS NEEDED. Additionally, game logs allow the calculation of the aforementioned ERRORS MADE. Table 3.2 (lower half) shows results coincide with the players assessments discussed in the upper half of the table. Concretely, Table 3.2 shows that the NOOBS_NOADAPT game has the best overall performance considering time needed, turns needed and errors made. The VISOBS_NOADAPT game shows worse results and the MEMOBS_NOADAPT game the worst. The most significant differences are the time difference ($t(17) = -8.52, p < 0.001$) and errors ($t(17) = -4.13, p < 0.001$) between the NOOBS_NOADAPT and MEMOBS_NOADAPT obstacle games, both being more than doubled in the latter.

Similarly, matching the questionnaire analysis in the previous section, Table 3.3 shows that the MEMOBS_LIMEMADAPT game came with no significant increases in performance of any of the three aforementioned metrics compared to the MEMOBS_NOADAPT game (p-values regarding the metrics order in the table: $p = 0.3$, $p = 0.06$, $p = 0.1$). Table 3.3 also shows that the VISOBS_VISADAPT game reduces time ($p < 0.001$), turns ($p < 0.001$) and errors ($p < 0.001$) made compared to VISOBS_NOADAPT game. Errors especially, were reduced by over 70% on average.

SEQUENTIAL MEASUREMENTS: beside the objective static measurements already discussed in lower parts of Tables 3.2 and 3.3, we discuss hereby, for different game variants, the *MatchingPairs per round* and *Penalties per*

round as sequential behavioral measurements, which continuously measure user actions' performance in the corresponding HCI application (revealed cards in game). First, we compare NOOBS game to different obstacles games:

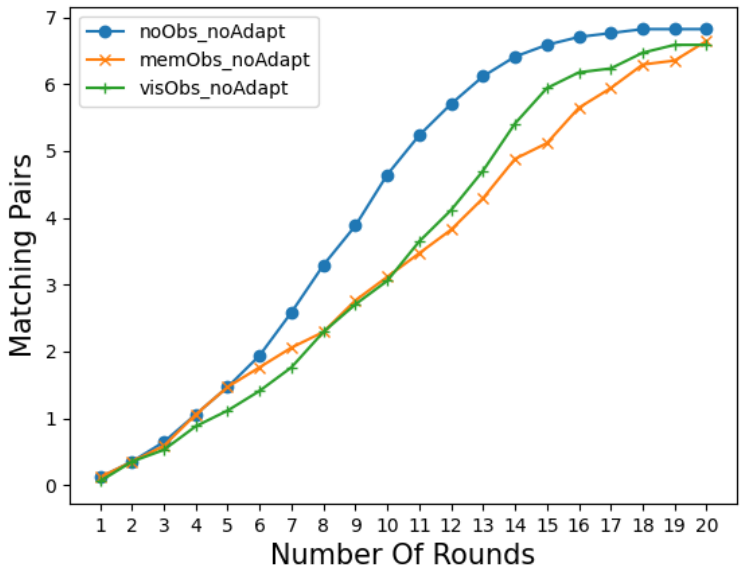


Figure 3.15 – The main game variants (without adaptations) in comparison: MatchingPairs per round as sequential measurement.

MEMOBS and VISOBS. Figure 3.15 shows the MATCHINGPAIRS PER ROUND sequential measurement for those different game variants. In that figure, we average the MATCHINGPAIRS PER ROUND lists of all subjects, for whom we have already discussed the static objective measurements in Table 3.2 (lower part). Thus, Figure 3.15 coincides with the findings we got from the static measurements discussed in the previous section; That is, it shows the best performance for NOADAPT, worse for VISOBS and the worst for MEMOBS. Moreover, we benefit from such a sequential measurement to see the performance changes sequentially, where we can see that MEMOBS looks identical with NOOBS for the first 5 rounds, while VISOBS begins from the beginning with worse performance compared to both NOOBS and MEMOBS. Afterwards, the performance of MEMOBS starts deteriorating from the round 5, meeting the VISOBS performance from round 8 to 10, and continues deteriorating afterwards to end with the worst performance. We argue such different behaviours based on the nature of the HCI obstacles we simulated in matching pairs game; That is, the MEMOBS game shows the same cards of NOOBS game (distinguishable colors), and we simulated the HCI MEMOBS as a secondary

WM load caused by cumulative sum, however, cumulative sum begins easy for the first few rounds, afterwards, it becomes harder and increasingly impedes memorizing revealed cards causing deteriorated performance. In contrast, the performance of VISOBS deteriorates from the beginning because the cards show confusing colors (red/brown shades as emulation of red-green color vision deficiency). Statistically, NOOBS performance outperforms both MEMOBS ($t(17) = 6.61534, p < 0.001$) and VISOBS ($t(17) = 7.76663, p < 0.001$). Figure 3.15 shows that both obstacles lines have more or less similar behaviour, where MEMOBS and VISOBS lines change over first and last rounds causing similar overall performance, thus, we find no significant difference between their performances ($t(17) = 0.24484, p = 0.81$). However, MEMOBS and VISOBS are

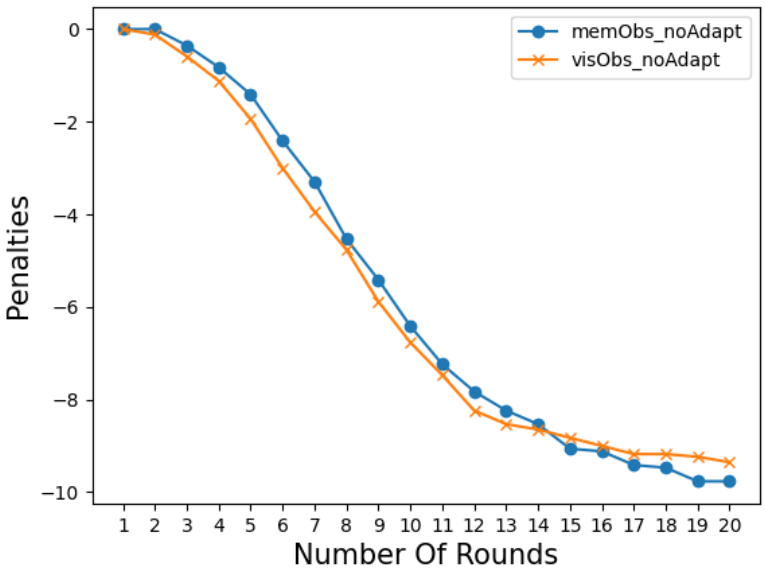


Figure 3.16 – Obstacles game variants (without adaptations) in comparison: Penalties per round as sequential measurement.

indeed different HCI obstacles, i.e. they typically cause different behaviours in terms of errors made. Therefore, we additionally analyze the PENALTIES PER ROUND sequential measurement to compare both obstacles. Figure 3.16 shows that MEMOBS and VISOBS penalties do not change over until a late round (round 14). Statistically, we do find a significant difference between those penalties ($t(17) = 4.86174, p = 0.0001$), this means we have two different behaviours. All together, our simulated HCI obstacles look valid, as they deteriorate user performance and show distinguishable behaviours. Similarly, we evaluate the UI ADAPTATIONS using such sequential measurements. For

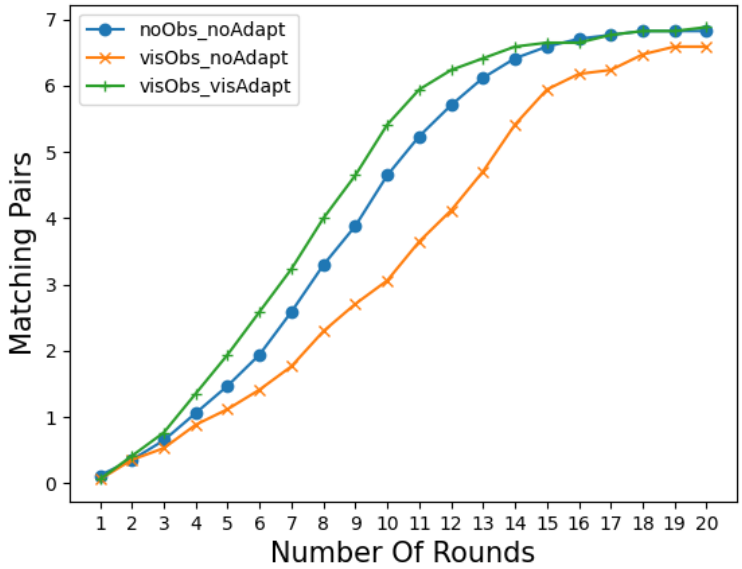


Figure 3.17 – visAdapt UI:MatchingPairs per round as sequential measurement.

UI visAdapt, we investigated compensating the impaired visual cognition by utilizing additional human cognitive process: hearing auditory instructions, recall the *UI Adaptations* introduced in Chapter 1. Figure 3.17 shows that those auditory instructions extremely improve the player performance making it even better than the standard noObs game. This also coincides with psychological literature which proves complementary roles between visual and auditory human memory, e.g. Crowder (1972). Statistically, we find a significant difference between noObs performance and visObs performance ($t(17) = 7.76663, p < 0.0001$). The performance improvement from visObs to visObs_visAdapt is also significant ($t(17) = 7.43661, p < 0.0001$). Coinciding with the aforementioned complementary roles between visual and auditory memory, we find also significant difference between visObs_visAdapt and noObs ($t(17) = 3.93427, p < 0.0001$), i.e. the auditory memory complemented the visual memory to significantly outperform noObs_noAdapt performance even in presence of visObs. The discussed MatchingPairs per Round measurement might be influenced by luck, because a player might find pairs during randomly revealing cards for exploring. Therefore, we also compare the game variants using the Penalties per Round measurement, which measures errors rather than (randomly) found pairs. Figure 3.18 shows lines coincides with Figure 3.17, where we see that visObs_visAdapt

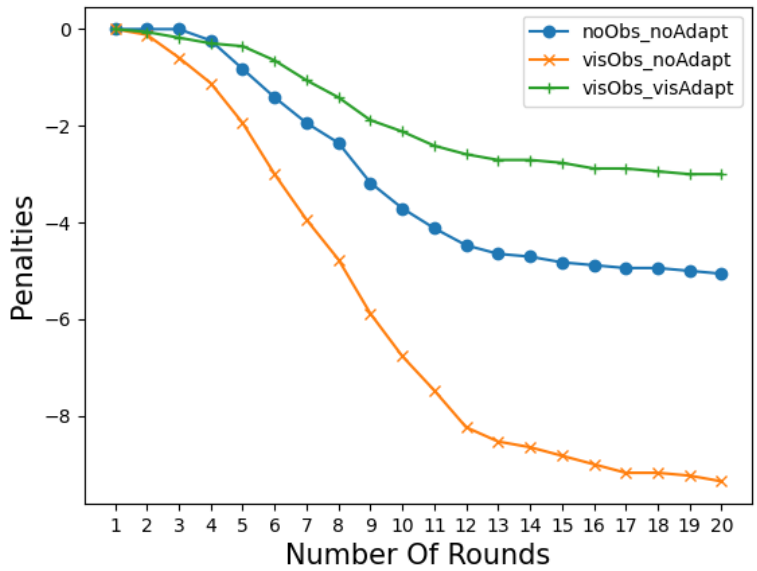


Figure 3.18 – VISADAPT UI:Penalties per round as sequential measurement.

outperforms both NOOBS and VISOBS_NOADAPT. Again, statistically we also find significant differences regarding penalties between NOOBS and VISOBS_NOADAPT ($t(17) = 7.61543, p < 0.0001$), between VISOBS_VISADAPT and VISOBS_NOADAPT ($t(17) = 7.46422, p < 0.0001$), and also between VISOBS_VISADAPT and NOOBS_NOADAPT ($t(17) = 6.84736, p < 0.0001$). For UI MEMADAPT, we have investigated a light and strong UI ADAPTATIONS; that is, UI adaptations differ regarding to the amount of changes applied in the UI: A light UI adaptation makes minor changes while a strong UI adaptation makes major changes in the UI, Salous et al. (2019). Figures 3.19 and 3.20 show the sequential performance effects of UI LIGHT MEMADAPT and UI STRONG MEMADAPT respectively. On the one hand, the UI LIGHT MEMADAPT (Figure 3.19) improves the player performance until a moderate level between NOOBS_NOADAPT and MEMOBS_NOADAPT. On the other hand, the UI STRONG MEMADAPT (Figure 3.20) further improves the player performance reaching the NOOBS_NOADAPT level. Statistically, we find significant differences between NOOBS_NOADAPT and MEMOBS_MEMADAPT ($t(17) = 6.61534, p < 0.0001$), and also between MEMOBS_LIMEMADAPT and MEMOBS_NOADAPT ($t(17) = 6.31683, p < 0.0001$). While we still find a significant difference between NOOBS_NOADAPT and MEMOBS_LIMEMADAPT ($t(17) = 5.58502, p < 0.0001$), MEMOBS_STMEMADAPT shows, as expected from Figure 3.20, no significant difference compared to NOOBS_NOADAPT.

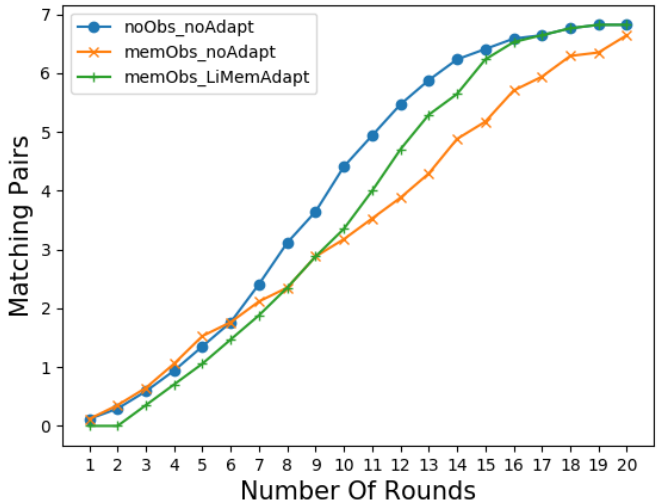


Figure 3.19 – Light MEMADAPT UI:MatchingPairs per round as sequential measurement.

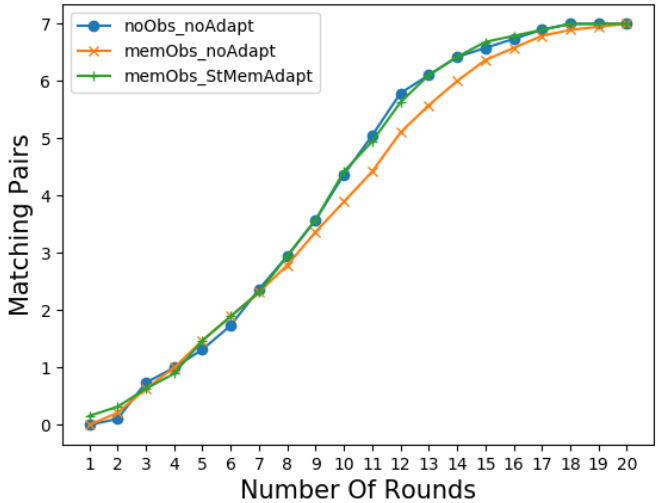


Figure 3.20 – Strong MEMADAPT UI:MatchingPairs per round as sequential measurement.

3.2 SIMDATA Collection

We aim at training appropriate machine learning models to automatically detect different HCI obstacles from different collected data modalities. In this

section, we introduce our cognitive-based simulator which we implemented to enrich the amount of behavioral data collected in TabExp user studies. That is, given that machine learning models, and especially deep learning and sequential models, require large amount of training data, LeCun et al. (2015), we modeled each subject behaviour during playing different memory game variants (Recall the game variants in 3.1.1.2) by extending the Cognitive Memory Model: CMM Pröpper et al. (2011). We were able to model player behaviour, thus, we simulated 1000 game for each subject under different conditions (game variants).

3.2.1 Cognitive-based User Simulation: Cognitive Memory Model: CMM

A challenge of generating additional training sequences for training of a sequential classification model is that the sequences have to maintain plausible temporal relationships between time slices. While traditional oversampling approaches, such as ADASYN, He et al. (2008), do not consider the temporal structure of sequences, recent approaches, such as Gong and Chen (2016), aim at capturing this temporal structure to generate plausible sequential data. We tackled that temporal structure challenge by extending the Cognitive Memory Model, Pröpper et al. (2011) (CMM).
CMM is a general computational cognitive model of human memory inspired by the ACT-R theory, Anderson et al. (2004). It has been successfully employed to model games of matching pairs, revealing different playing strategies and levels of memory performance, Putze et al. (2015). In this thesis, we extend the CMM with so-called application-based parameters (the next Section 3.2.2) to model player behaviour and thus to generate additional training episodes for each aforementioned game variant. The CMM is based on a decay-based forgetting mechanism to realistically simulate non-perfect human memory. For each item in memory (cards in our use case), the CMM maintains an activation value from which the likelihood of retrieval can be calculated. In general, the activation of an item depends on the frequency and recency of stimulations of said item. Based on the retrieval likelihood for all individual cards in a given game state, we execute a specific strategy to select which cards to reveal. This strategy balances exploration (revealing cards which are unknown to the player or not remembered) and exploitation (revealing pairs when position of both corresponding cards is remembered with high enough probability). By repeatedly executing this strategy and updating the game state, artificial sequences of game play can be generated.

The CMM has a number of free parameters, such as the degree of memory decay, which determine its predictions of memory performance. Those parameters can be optimized by a genetic optimization algorithm in CMM to best fit the empirical training data. The genetic optimizer maintains a population of CMM configurations consisting of all free parameters. The population is initialized randomly and then iteratively updated according to the standard operations of genetic optimization, mutation and selection Fogel (1994). For selection, we measure the similarity between the real references sessions and the simulated ones using two metrics: 1) MatchingPairs measure counts the number of removed card pairs for each turn. 2) Penalties measure counts the number of card pairs which have both been revealed at some point of the game but have not been removed by the player (presumably, because the location of one or both of the cards to that pair have been forgotten by the player). A CMM configuration is then selected more likely if it generates game sessions which are close to the real data according to those specified similarity metrics. As we are comparing sequences, the similarity measures are also defined per game turn and distance between two game sessions is measured as the Root Mean Squared Error (RMSE) between the two turn-wise sequences.

Playing behavior in the game of matching pairs varies between individual players, for example depending on WM capacity and differences in playing strategy. To achieve enough variance in the training data, we do not pool all available real sessions together to optimize a global parameter set; instead, we repeat the process for each session individually and only pool the resulting simulated sessions together to form a richer, more varied corpus of simulated sessions.

3.2.2 Extended CMM for Matching Pairs

The CMM has a set of parameters (e.g. memory decay) which are optimized to model the human memory performance. In total, CMM has seven parameters which are randomly initialized and then repeatedly optimized using a genetic optimization algorithm. Those parameters can be further extended with application-based parameters. In this section, we discuss the parameters with which we extended the CMM to simulate player behaviour and HCI obstacles in matching pairs game.

- **Randomizing Parameter:** this parameter is defined to realistically simulate the non-perfect human memory in matching pairs (as an HCI exemplary application). That is, according to the CMM-based WinStrategy introduced by Putze et al. (2015), we should repeatedly

calculate reveal probability for each memorized item (card), and the simulator will select the card with the highest probability to reveal, where reveal probability is updated for each card after each user action (revealing a card) based on the frequency and recency of stimulations of those memorized items. However, we should not simulate a perfect behaviour, because that is non realistic, instead, we should simulate the forgetting and the randomized card selection which differ between individuals regarding to their WM performance. Therefore, we defined the RANDOMIZING PARAMETER which will be optimized, together with CMM parameters, by the genetic algorithm in CMM.

- **Similarity Matrix:** we introduced a similarity matrix to simulate similarities between cards in matching-pairs game. That is, given all the cards in a matrix as rows and columns, the cells will contain values between 0.0 and 1.0 that define how similar to each other are the cards in the corresponding row and column. This similarity matrix is called then by the simulator for each revealed card to emulate human confusion happens especially when revealing those cards of the VISOBS game variants, including GLAREOBS game variant. Since the cards show only colors (See Figures 3.4, 3.5 and the hardest glared one 3.6), we use the CIE1976 color model[6] to calculate the similarity values between those shown colors. A new parameter, `Similarity Decay`, is also introduced, which reduces the similarity effects while game is running: user may learn and adapt to such similar cards while the game is running, moreover, the similarity effects reduce continuously with less cards in the game after detecting pairs.

3.2.3 SIMDATA Evaluation

In this section, we evaluate the validity of the simulated matching pairs games, which we generated and used to enrich the data corpus collected in our TABEXP user studies discussed in Section 3.1. For such validity evaluations, we measure the sequential similarity between those simulated games and their references (real games). We calculate the same sequential similarity measurements discussed above in the simulator: MATCHINGPAIRS and PENALTIES measurements. In the following sub-sections, we discuss those similarity measurements in two main categories: persistent and volatile HCI MEMOBS, persistent and volatile HCI VISOBS.

[6]following the implementation `https://python-colormath.readthedocs.io/en/latest/delta_e.html`

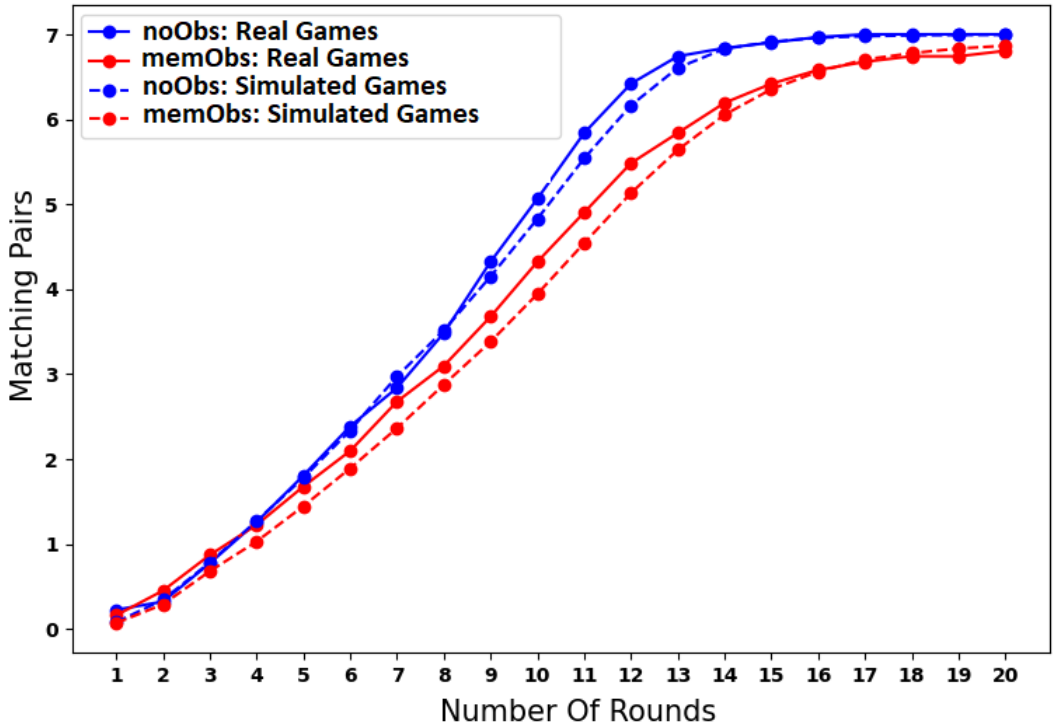


Figure 3.21 – MatchingPairs measurement to compare simulated games (dashed curves) to their references real games (solid curves) for noObs and memObs games.

3.2.3.1 Persistent and Volatile HCI MemObs

In this section, we average the *MatchingPairs per round* measurement for all the 31 subjects who participated in our first user Study, recall Section 3.1.1.1. In that user study, we defined the VOLATILE MEMOBS as secondary task memory load, and the low WM capacity as PERSISTENT MEMOBS in the corresponding HCI task. Figure 3.21 shows sequential player performance, expressed as MATCHINGPAIRS PER ROUND, for NOOBS and MEMOBS games. It shows the player performance regarding to both real games (solid curves) and simulated games (dashed curves). From the one hand, Figure 3.21 shows a remarkable drop of player performance from NOOBS game (blue solid curve) to MEMOBS game (red solid curve). From the other hand, Figure 3.21 shows similar curves between the simulated games (dashed curves) and their references real games (solid curves). The evident similarity can also be

quantified by calculating turn-wise RMSE between curves for corresponding real and simulated curves. Simulated data from class NOOBS yields an RMSE of 0.12 to the real data from class NOOBS and an RMSE of 0.41 from class MEMOBS. Distance is similarly large for simulated data from class MEMOBS (RMSE of 0.67 to real NOOBS and 0.21 to real MEMOBS).

3.2.3.2 Persistent and Volatile HCI VisObs

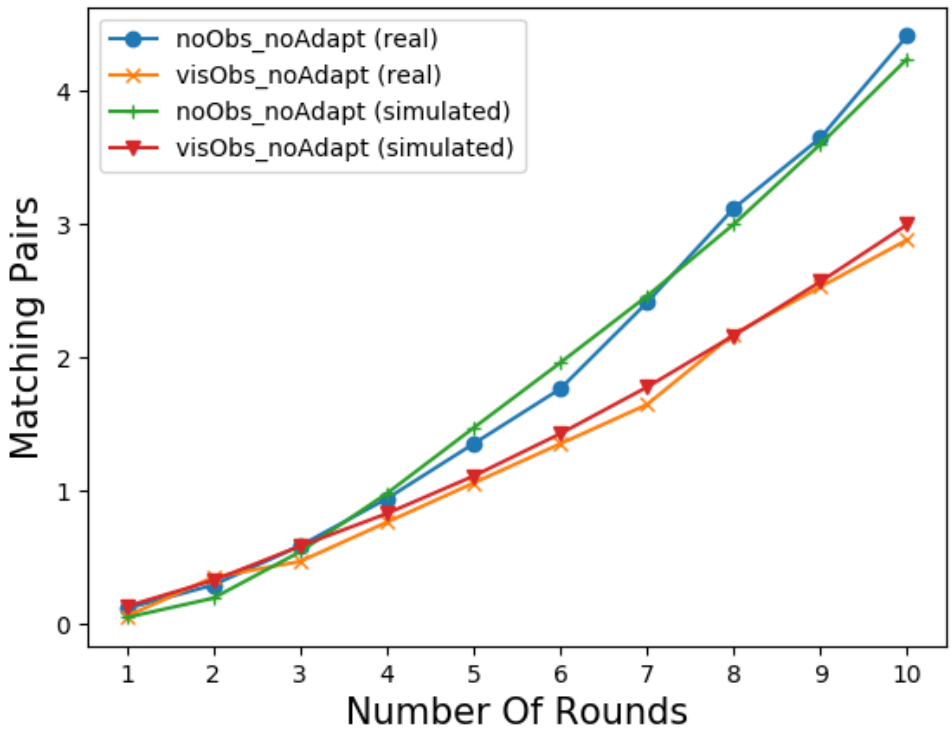


Figure 3.22 – MatchingPairs measurement to compare simulated games to their references real games for noObs and persistent visObs (Red-green color vision deficiency) games.

In this section, we evaluate the similarity between the simulated data of *Persistent and Volatile HCI VisObs* and their real data references, recall the data collections in Section 3.1.1.2. Thus, for the *Persistent VisObs* (emulated red-green color vision deficiency), we calculate the average *MatchingPairs per Round* measurement for the whole participating 17 subjects and for their 17000 simulated games (1000 per subjects).
Figure 3.22 shows a close similarity for the whole first 10 rounds (first

20 revealed cards) between the simulated and real sessions in the case of NoObs_NoAdapt and VisObs_NoAdapt games (VisObs means here the emulated red-green color vision deficiency as a persistent VisObs). Similarly,

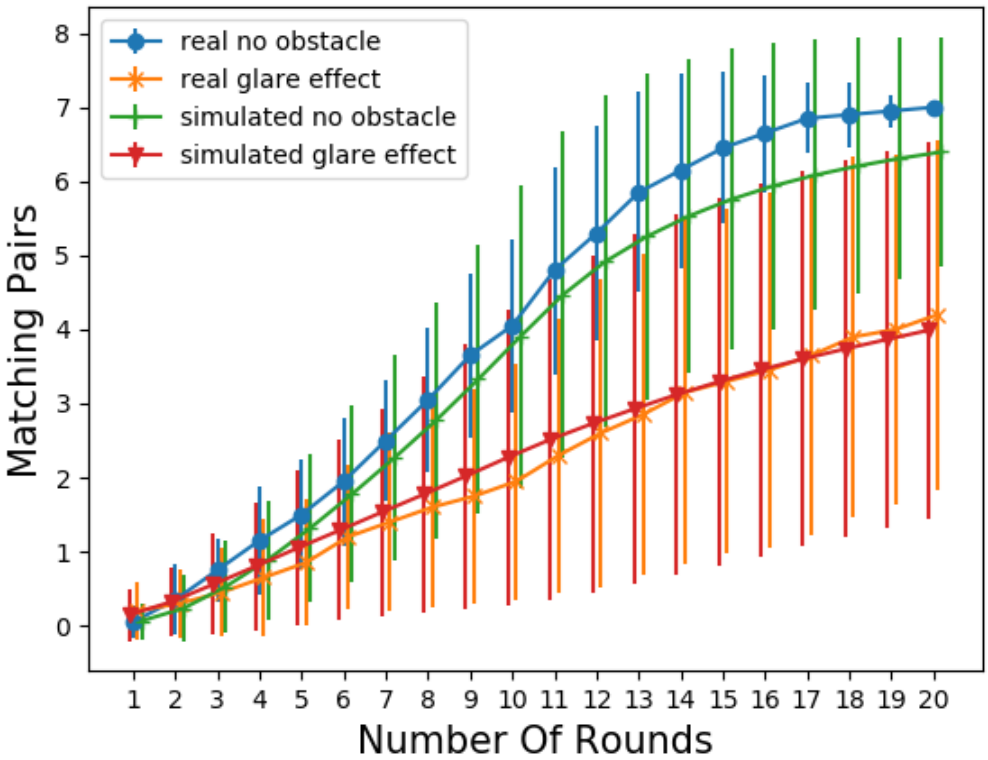


Figure 3.23 – MatchingPairs per round sequential measurement to compare simulated games to their references real games for noObs and volatile visObs (Glare effects) modes. While curves continuously show such a sequential comparison between the average performances, the vertical lines (whiskers) show also comparisons between standard deviations at each turn.

Figure 3.23 shows valid simulated volatile visObs games which follow the sequential behaviour of their real data references. Additionally, Figure 3.23 shows the whole 20 rounds recorded with whiskers as standard deviations. The simulated glare effects curve (orange curve) is very close to real data glare effects curve (red curve) for the the whole 20 turns. The standard deviations (whiskers) look also very similar for the whole 20 turns. However, we notice that the simulated noObs green curve begins similar to real noObs blue curve for the first turns, but drops for the late turns. We argue the

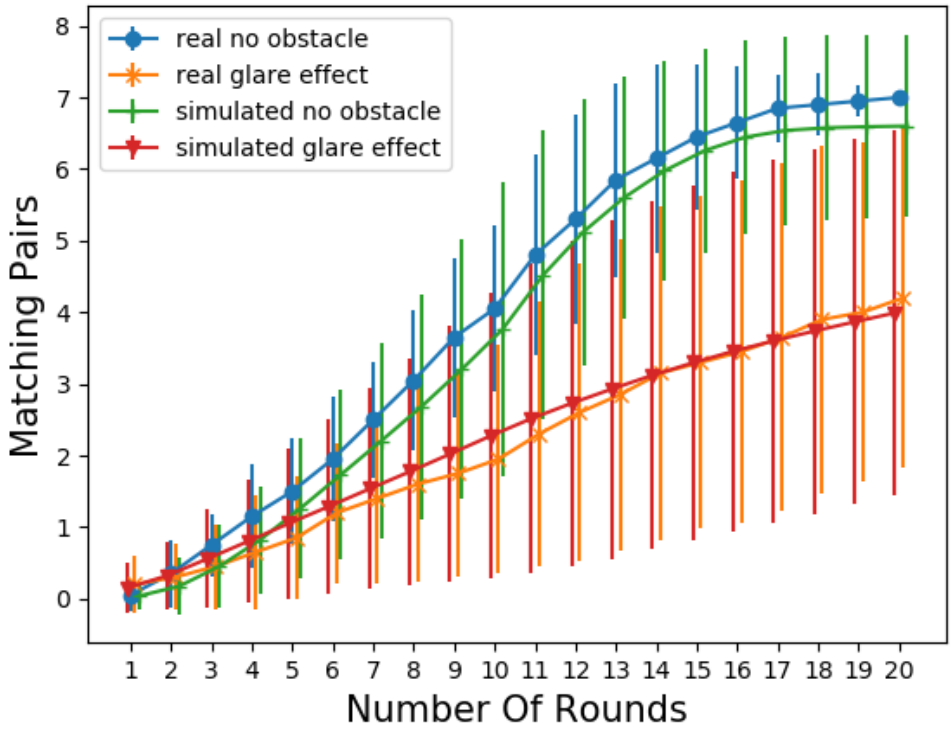


Figure 3.24 – Simulated noObs are improved compared to Figure 3.23 by defining and optimizing a randomizing decay parameter

almost perfect simulation of the VOLATILE VISOBS glare effects (and also PERSISTENT VISOBS in Figure 3.22) by the good effects simulated by the aforementioned SIMILARITY DECAY; that is, SIMILARITY DECAY realizes the decreasing of glare effects (and any visual obstacle) with more pairs found and removed, recall Section 3.2.2. Thus, it seems that we should also decay the randomizing to best fit NOOBS games. Therefore, we re-extend the CMM model with a new decay parameter so-called RANDOMIZING DECAY. This new parameter is intended to minimize the randomizing applied especially in the late turns where fewer cards exist in the field to finish the game. Figure 3.24 shows that such a parameter improves the simulation of NOOBS, where simulated data curve becomes closer to the real data curve with more similar whiskers compared to those shown in Figure 3.23.

Figure 3.25 – WebData data collection: Start page.

3.3 WebData Collection

TabExp user studies are limited to a relative small number of participants. While we introduced the SimData collection to enrich the behavioral data corpus collected in TabExp user studies, we introduce hereby the WebData collection to further enrich the behavioral data corpus with real data. That is, we implemented the same game variants mentioned in TabExp user studies as an online website under the following link: `http://csl-memory-game.informatik.uni-bremen.de/`.
Since we collected few logs from the old healthy subjects user study (Section 3.1.1.4), we began with a simplified version of this Web-based user study, which was especially, but not exclusively, oriented for old subjects, i.e. only the noObs game variant, with the same setup used in old healthy subjects (Section 3.1.1.4) and dementia patients (Section 3.1.1.3) user studies.

3.3.1 WebExp Experimental Design

The WebData collection is a website with three pages: Start, Info and Play. Figure 3.25 shows a screenshot of the Start page, where a potential subject can choose between English and German (Deutsch) languages, select the gender and age. No name is required as we always collect data anonymously. Figure 3.26 shows a screenshot of the Info page (English version). It explains, in general, the single-player mode games which will be played in the Play page.

Finally, Figure 3.27 shows a screen shot of the Play page. Player continuously sees progress info on the right-side info panel. This panel contains greeting as well as hints about the current and remaining games.

Figure 3.26 – WEB data collection: Info page.

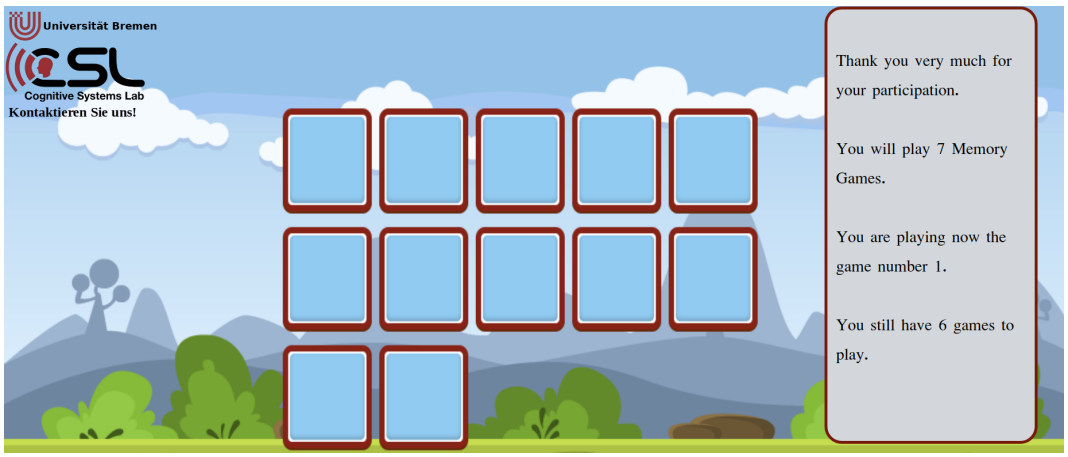


Figure 3.27 – WEB data collection: Play page.

3.3.2 WebExp Subjects

In this section, we discuss the data we collected in this Web-based user study. We assume that only healthy subjects participated in this WebExp user study (no dementia patients). We clustered the collected logs into age-based groups and further analyzed those groups by calculating, for each group, the aforementioned sequential performance measurements (MatchingPairs and Penalties). Concretely, we collected, until 21/10/2020, 10 logs from children $<= 14$, 1 log from adults $14 > age > 20$, 96 logs from young subjects $20 > age >= 30$, 31 logs from thirtieth subjects, 13 logs from fortieth subjects, no logs from fiftieth subjects and 29 logs from old subjects $age > 60$.

3.4 Data Exclusion and Limitations

Finally, we mention the invalid data which has been excluded from our analysis, and we discuss limitations of our data collection strategy.

We excluded the data collected from one participant in the first data collection 3.1.1.1, because that participant did not like the experiments and did not follow the instructions. Especially for the MEMOBS game, the participant did not calculate the cumulative summation which was designed as a secondary task memory load. As the implemented application logs the cumulative summations typed by participants, we found that all the other participants did (try to) calculate the cumulative summation, and thus, their data was valid as intended in the experiment: simulated volatile MEMOBS.

One limitation of the presented work is that the interaction obstacles were only simulated for the purpose of a controllable study with comparable conditions between all participants. However, we will show in Chapter 4 that the described approach also works with real interaction obstacles, namely naturally limited WM capacity, and dementia detection.

Nevertheless, future work needs to further tackle the challenge of transferring the developed models to users experiencing real interaction obstacles. For example, we will show in Chapter 4 how the MEMOBS detectors generalize well to detect dementia. However, we did not investigate the ability of the simulated color blindness VISOBS detector to detect a real color blindness obstacle, because we do not have data collected from color blind participants.

3.5 Conclusion

In this chapter, we discussed all the HCI data collections we prepared. The collected HCI data can be categorized in three main categories: NO OBSTACLE HCI, OBSTACLE HCI and OBSTACLE HCI WITH UI ADAPTATION. We used the well-known matching pairs game as an exemplary HCI task, because it suits the two main obstacles we aim at simulating and detecting: MEMORY-BASED OBSTACLE and VISUAL OBSTACLE; That is, Matching pairs is an intensive memory task which requires good recognition ability of the UI visual items (cards motives). Moreover, the chosen memory game seems to be a good fit to represent many typical HCI tasks: It involves visual inspection of the UI, working memory retrieval and encoding (of both spatial and symbolic information), as well as planning and decision making, which all occur also in many other HCI tasks.

Table 3.4 – Data collection summary: participating subjects in all the data collections regarding to their user studies.

Database	User Study	Users	Thesis Section
TabData	memObs (2017)	31 subjects age 19-48	3.1.1.1
	Obstacles And UI Adapt.1 2018	19 subjects age 18-27	3.1.1.2
	Obstacles And UI Adapt.2 2019	17 subjects age 17-27	3.1.1.2
	Obstacles And UI Adapt.3 2020	16 subjects age 19-29	3.1.1.2
	Dementia User Studies	15 patients mild...severe	3.1.1.3
	Elderly Healthy (2019)	6 healthy $age > 60$	3.1.1.4
SimData	-	1000 simulated log per session	3.2
WebData	-	12 elderly healthy $age > 60$	3.3

We discussed the different methods we used to collect HCI data for our models: TabExp, SimExp and WebExp user studies, see all summarized in Table 3.4.

For TabData collection, we designed consecutive user studies over three years: from 2017 to 2020; We collected multimodal data (brain activity EEG and behavioral data) from a variety of 104 subjects: healthy young subjects (EEG and behavioral data), healthy elderly subjects and dementia patients (only behavioral data). We analyzed the collected data from subjective and objective perspectives.

To enrich the collected behavioral data, we introduced a user cognitive simulation based on CMM Pröpper et al. (2011). we discussed the parameters which we defined in CMM to extend it for such a simulation of player behavior in matching pairs HCI. We compared the simulated HCI sessions to their

references real sessions using different sequential measurements to show the validity of the simulated data.

To further enrich the behavioral data corpus, we introduced the WebData collection, in which we implement the same TabExp game variants as a website to collect more behavioral data worldwide. We cluster the WebData collected based on age, so we can enrich the young and elderly healthy subjects TabExp groups. One of the clusters is for logs from children ($age < 14$), we plan to consider such a group also in our *User Modeling* solutions in our future works.

Chapter 4

Detection of HCI Obstacles

In this chapter, we discuss different HCI obstacles detectors, so-called Elementary Models in our cognitive adaptive system architecture. Elementary models use several data modalities (ElectroEncephaloGraphy (EEG) and behavioral data) to detect memory-based and visual interaction obstacles. The elementary models' outputs will be passed as inputs to the decision model: Dynamic Bayesian Network (DBN, will be discussed in the next Chapter).

The data collected throughout the thesis (discussed in Chapter 3) act as inputs for the obstacle detection. We split each data set (EEG and behavioral data modalities) for training and testing HCI obstacle detectors. We call those HCI obstacle detectors Elementary Models, because their outputs (predictions and confidences) will be passed to the probability-based decision model (DBN, the next Chapter) and thus they contribute but do not produce the final UI Adaptation decision.

As introduced in Chapter 1, we discuss memObs and visObs as HCI obstacles, and we distinguish between different obstacle types for each: volatile and persistent interaction obstacles. In this chapter, we introduce an obstacle detector model (Elementary Model) for each HCI obstacle accordingly. An *elementary model*, in our cognitive adaptive system, is a machine learning binary classifier which detects the presence or absence of a specific HCI obstacle (see parts 2 and 3 in Figure 1.5 in Chapter 1). The Elementary Models will act as "plugins" into our ultimate system: flexible cognitive

adaptive system, which we will discuss in the next chapter. Flexibility of our cognitive adaptive system is an important characteristic, that is why we train simple binary models as HCI obstacle detectors rather than one monolithic model which discriminates all obstacles at once. In other words, while we are discussing memObs and visObs detectors in matching pairs HCI, we can generalize this strategy for detecting whatever obstacles in whatever HCI application, e.g. attention distraction from eye tracking data. That is, following our pipeline from chapters 3 to 5, one should define the HCI obstacle to be detected and collect HCI data from HCI sessions with and without that obstacle (Chapter 3), train a binary obstacle detector to detect that obstacle from each data modality collected (Chapter 4), and finally, flexibly add those trained binary HCI obstacle detectors (just as plugins) into the overall probability-based UI Adaptation decision system (DBN, Chapter 5). However, we assume that the obstacles are independent from each other, i.e. these extensions will be restricted in case of dependent obstacles, thus, we plan to model dependencies between HCI obstacles in our future work.

Alongside the pipeline in this thesis, we discuss in the following sections memObs and visObs detectors, both detected based on EEG and behavioral data modalities. We follow best practices in the machine learning to choose the best model for each data modality; Lotte (2014) recommends discriminant models, especially Linear Discriminant model (LDA) and Support Vector Machine (SVM), for EEG-based pattern detection (detection of HCI obstacle in our case). For behavioral data, researchers widely used Recurrent Neural Networks (RNN), and especially Long Short-Term Memory (LSTM) to model human behavior for different prediction contexts, e.g. Gers et al. (2000); Schaul and Schmidhuber (2009); Graves et al. (2009); Eck and Schmidhuber (2002).

While multiple obstacles can occur simultaneously, we focus in this thesis on the presence of one obstacle in an HCI session (game prefix of 10 rounds, which lasts relatively short, less than one minute). If we want to consider multiple HCI obstacles at the same HCI session, then, we have also to consider potential interaction dependencies between such obstacles, and, we should introduce new UI adaptation mechanisms for such cases. Actually, modelling such interactions between obstacles is a challenge, as multiple cognitive processes are affected, e.g. if we assume that memObs and visObs present simultaneously in an HCI session, both cognition and perception processes are simultaneously affected. In Chapter 6, we will discuss this limitation and others in more details.

Generally speaking, each HCI obstacle mainly impairs a specific human cognitive process and can be best detected from a specific appropriate data modality. For memObs, WM and its recall process will be impaired, thus, EEG would be a good and suitable (noninvasive) method for detecting memObs from brain activity data modality, because noObs and memObs sessions represent two different mental loads and would be reflected as discriminant EEG electrical signals. In contrast, noObs and visObs represent the same mental task, however, the visObs session impairs the perception process and causes a confused user behavior, which could be well detected from the behavioral data modality. We do not know which obstacle occurs (memObs or visObs), i.e. during an online continuous use of the ultimate cognitive adaptive system, the tested HCI session can be with either obstacle possibility (noObs, memObs or visObs). Therefore, we need to train all possible combinations between method and obstacle detectors as Elementary Models: EEG-based memObs detector, EEG-based visObs detector, behavior-based memObs detector and behavior-based visObs detector. Moreover, we train and evaluate such detectors of persistent and volatile obstacles, and we will eventually evaluate all possible volatile and persistent cases of obstacles in the DBN model in the next chapter. According to the nature of the discussed obstacles, we expect the following combinations to be the best ones: EEG-based memObs detector because EEG is widely used for discriminating different memory demands, e.g. Berka et al. (2007); Baldwin and Penaranda (2012); Ke et al. (2014); Mühl et al. (2014), and behavior-based visObs detector because visObs will cause a confused HCI behavior which can be well depicted through potential temporal dependencies in the recorded behavioral data (e.g. Gers et al. (2000); Schaul and Schmidhuber (2009); Graves et al. (2009); Eck and Schmidhuber (2002)). Thus, we discuss in the final section in this chapter (Section 4.5) all the results of all the aforementioned combinations and investigate if our hypothesis is correct.

While we show performance differences in this chapter between different data modality based detectors, such performance differences between all the aforementioned combinations will be automatically learned as distributions by the probabilistic decision model (the next chapter) to decide the most probable UI adaptation.

4.1 Memory-based Obstacle Detector in HCI

In this section, we discuss the detection of persistent and volatile HCI memObs. As introduced in Chapter 1, we simulate and detect volatile memObs as

a secondary task memory load, and we detect persistent memObs in two different scenarios: 1) Dementia disease, which is a typical persistent HCI memObs. 2) Given a complex HCI task to healthy users, some users may suffer from low WM capacity during the HCI sessions, while other users treat the same complex HCI task easily, thus, we aim at detecting the low WM capacity as a persistent HCI memObs in a complex HCI task.
First, we begin in Section 4.1.1 with discussing state of the art models for memObs detection. Then, we discuss the different binary classifiers we designed to detect HCI memObs from different modalities: *Behavior-based* memObs detector in Section 4.1.2, and *EEG-based* memObs detector in Section 4.1.3.

4.1.1 State of The Art and Related Works

In this section, we discuss the state of the art models which aimed at detecting volatile and persistent HCI memObs. We also discuss related works, in which similar *machine learning* (ML) technologies to ours are applied, but not necessarily for the same task of detection of HCI memObs.

4.1.1.1 volatile HCI memObs

In this section, we discuss the state of the art models for the detection of volatile HCI memObs. First, we discuss the data modalities used for the detection of memObs. Second, we discuss the machine learning models used in those works. Finally, we highlight gaps in the state of the art, follow related works for solutions and show how our multimodal models contribute to tackle the said gaps.

First, according to data modalities, approaches for online detection of memory-based interaction obstacles make use of one or more modalities to classify the cognitive state of the user. As WM is such an important cognitive construct influencing many HCI tasks, there exists a large body of research which aims at measuring WM load, e.g. Putze and Schultz (2014). Typically, workload is induced by memory-loading secondary tasks, and actual measurement of WM load then takes place by classifying features from various modalities. Most prominently, researchers investigate signals from neural sources, for example recorded from electroencephalography (EEG), e.g. Berka et al. (2007); Heger et al. (2010); Baldwin and Penaranda (2012); Mühl et al. (2014); Ke et al. (2014). Moreover, researchers investigated combining EEG with other methods for recording brain activity data, e.g. Herff et al. (2015) showed how a hybrid classifier (combining EEG and fNIRS measurement) was able to reliably discriminate pairs of difficulty levels in a sequential memory task.

Brain activity is not the only modality which carries information on memory workload; Eye tracking data can also be used for this purpose: Rozado et al. (2015) demonstrated that EEG-based memory workload estimation can be combined with pupillometry data for increased accuracy. Katidioti et al. (2016) used pupillometry-based load estimation to create a task-independent interruption management system.

Second, according to the machine learning models, researchers typically used discriminant classifiers, e.g. Support Vector Machine SVM, for classifying memory workloads from EEG data. That is, discriminant models, especially shrinkage LDA and SVM, are highly recommended in corresponding literature reviews e.g. Lotte (2014). Recently, researchers have been increasingly investigating Convolutional Neural Networks (CNN) for memory classification from multimodal sensory data including EEG, e.g. Saadati et al. (2019) for classification of memory workload tasks based on fNIRS and EEG signals. However, even very recent works still follow the traditional recommendation of using discriminant models (especially SVM) for robustly classifying memory workloads from hybrid EEG and fNIRS data, Mandal et al. (2020) for example highlighted perfect performance accuracy (100%) when classifying different N-back cognitive tasks (i.e. different memory workloads) from hybrid fNIRS and EEG data using different variants of SVM (linear SVM, quadratic SVM, and cubic SVM).
Finally, regarding gaps in those related works, we find that all the presented approaches have in common that they only regard relatively short segments of sensory data (EEG, fNIRS) without resorting to a larger temporal context which can be encapsulated in behavioral data. To tackle this gap, we searched for sequential models and found that recurrent neural networks are sequential classification models which allow to take such information into account. Concretely, Long-Short Term Memory (LSTM) network is a type of recurrent neural network which uses designated memory cells to model long-range dependencies within sequential data, Hochreiter and Schmidhuber (1997). We found that LSTMs have been widely used for modeling sequential data in other contexts rather than HCI. For example, Gers et al. (2000) showed that LSTMs were able to successfully learn the rules encoded in context-free and context-dependent formal languages. Schaul and Schmidhuber (2009) investigated the use of LSTMs for modeling playing strategies in complex games and the ability of the model to generalize from small to larger game boards. Graves et al. (2009) used bidirectional LSTMs for recognition of unconstrained, connected handwriting. Eck and Schmidhuber (2002) showed that LSTMs were able to discover and reproduce the temporal structure of complex pieces of music due to their ability in detecting long-range depen-

dencies.
While LSTM has been established as a powerful modeling tool, to the best of our knowledge it has never been applied to behavioral data in the HCI context; Thus, we modeled user behavior during HCI using LSTMs to detect HCI memObs (Putze et al. (2018); Salous and Putze (2018); Salous et al. (2019)). The experimental results documented in our works have shown that for the detection of memory workload, sensory-free behavioral data may serve as a low-cost alternative to EEG, for details about our behavior-based LSTM model see Section 4.1.2.
We also followed the state of the art works mentioned above and modeled the EEG data recorded during our HCI task to detect the simulated volatile HCI memObs from brain activity modality, see Section 4.1.3.
Eventually, we will highlight complementary roles between our multimodal heterogeneous models in the detection of HCI obstacles in Chapter 5: behavioral data for modelling sequential dependencies along user actions, and EEG data for modeling user states. Briefly, we will see in Chapter 5 that a Bayesian model, upon such multimodal HCI obstacle detectors, performs well for the multimodal HCI obstacles detection and corresponding UI adaptation.

4.1.1.2 Low WM capacity as persistent HCI memObs

Due to its high importance in HCI contexts, many approaches aim at dynamically measuring WM load by sensory data (e.g. Baldwin and Penaranda (2012); Ke et al. (2014); Herff et al. (2015)). However, such approaches do not estimate general, person-specific WM capacity. Thus, there exist also standard approaches for measuring WM capacity by implementing widely used WM capacity measures (e.g., Operation: Memory-Update (MU) and Reading Span, Lewandowsky et al. (2010)). For example, in the MU task, participants memorize a series of numbers in parallel while applying a sequence of arithmetic operations to them. However, such explicit approaches for assessing WM capacity are time-consuming, stressful, and not well-integrated into HCI applications. Therefore, we introduced in Salous and Putze (2018) a sensory-free behavioral model with an LSTM for detecting low WM capacity in a complex HCI task, the LSTM details are presented in Section 4.1.2.
In addition to the behavior-based LSTM model, we followed the state of the art models mentioned in Section 4.1.1.1, and modeled the EEG data recorded during our HCI tasks to detect HCI memObs (model details are presented in Section 4.1.3).
Eventually, we integrate and benefit from both modalities in Chapter 5: behavioral modality (modeling sequential dependencies along user actions) and brain activity modality (collected via EEG to model user cognitive states). We

will see that our cognitive adaptive model will benefit from such multimodal to detect volatile and persistent HCI obstacles in consecutive HCI sessions, more details are presented in Chapter 5.

4.1.1.3 Dementia as persistent HCI memObs

Dementia is a typical aging disease that impedes HCI among many other everyday activities. Arvanitakis et al. (2019) reviewed common symptoms, diagnosis procedures and management of dementia. According to their review, dementia impairs human memory among many other cognitive processes. Therefore, we consider dementia during HCI as a persistent HCI memObs, and thus aim to detect it. Machine learning (ML) technologies have been investigated for detecting dementia from different data modalities. Weiner et al. (2017) introduced two ML-based pipelines for dementia detection from speech recordings: a manual pipeline uses manual transcriptions of features and a fully automatic pipeline uses transcriptions created by automatic speech recognition (ASR); Since speech and language functions are affected early in the course of dementia, those ML-based approaches can support clinical screening methods. Apart from speech data, classical informant-based questionnaire data were used to train ML models for detection of dementia, Zhu et al. (2020); the authors examined six different ML models and three different *Feature Selection* methods. They concluded that *Information Gain* was the most effective feature selection method, and the Naive Bayes algorithm performed the best accuracy in dementia detection (81%). Due to outstanding performance of *Deep Learning* models over traditional ML models in complex prediction tasks, and given that rapid progress in neuroimaging techniques has generated large-scale multimodal neuroimaging data, *Deep Learning* has been recently getting considerable attention in early detection and automated classification of Alzheimer disease, Jo et al. (2019); The authors prepared a systematic review for Alzheimer disease detection using *Deep Learning* and neuroimaging data reviewing works between January 2013 and July 2018. In total, they reviewed 19 works in two categories: 4 works used only *Deep Learning* and 12 works used combination of *Deep Learning* and traditional ML. They highlighted great performance from those combined models with (98.8%) as an accuracy of detection Alzheimer disease. The authors concluded in their review that *Deep Learning* solutions are still evolving for detection of Alzheimer disease, and still improving performance by incorporating additional hybrid data types (multimodal data).
While that recent review (Jo et al. (2019)) highlights a robust detection of dementia from neuroimaging data using *Deep Learning*, it is not compatible with HCI contexts to use neuroimaging data online (during an HCI session)

for the detection of dementia. Instead, we aim at using *Deep Learning* to detect dementia but from behavioral data as an HCI-compatible approach. Thus, we searched for state of the art MEMORY GAMES in the ML-based dementia detection literature, from which dementia can be detected from simple, noise-free behavioral data. Sea Hero Quest by Hyde et al. (2016) is a navigation-based memory game which has been becoming very famous with about 4.3 Millions downloads in App stores worldwide. The game aims at detecting "Early stage" dementia, as one of the early symptoms in dementia is disorders in navigational skills. While matching pairs game has been investigated to activate the memory of dementia patients (Schultz et al. (2018)), to the best of our knowledge, there is no work discussing the detection of dementia from matching pairs memory game, which is the exemplary HCI application we are using to detect different HCI obstacles, including detecting dementia as a persistent HCI obstacle.

4.1.2 Behavior-based Detector

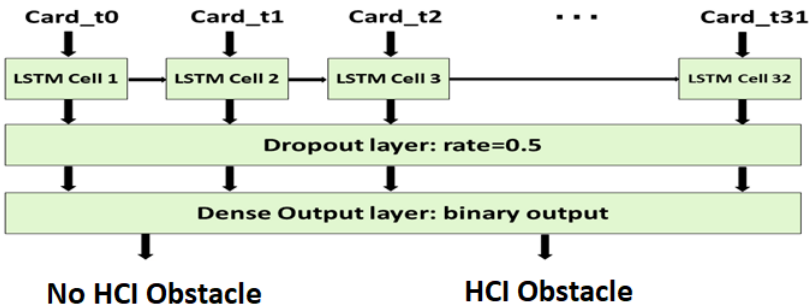


Figure 4.1 – Behavior-based sequential model for detecting HCI obstacles

In this section, we introduce our sequential model LSTM for behavior-based detection of persistent and volatile HCI memory-based obstacles (MEMOBS). Additionally, in contrast to LSTM as a sequential model, we introduce a static discriminant model (Linear Discriminant Analysis, LDA) as a baseline. Thus, we can investigate sequence-based advantages in LSTM when comparing the performance between LSTM and LDA in the evaluation section afterwards (Section 4.2).

LSTMs consist of so-called LSTM cells which explicitly store, retain, or forget information from previous time steps, see Figure 4.1. This approach is chosen to battle the challenge of vanishing gradient in traditional RNNs, Hochreiter

and Schmidhuber (1997). For deciding on the LSTM topology and hyperparameters, we followed Greff et al. (2016) who recommended random search ranges for the optimization. Moreover, we extended the randomly produced ranges with common practice values to ensure that such good common practice configurations are also tested in the optimization. For example, we extended a random search range for the LSTM num_cells parameter with multiples of 32 (e.g. 32, 64 and 128) which are commonly used in LSTMs. Thus, we ended up with an LSTM layer with 32 LSTM cells as input layer to handle the training data sequences.

Sequence items are passed consecutively to LSTM cells as consecutive time steps. The LSTM cells are interconnected where the output of one cell acts as an additional input to the next cell. Such a specific structure allows LSTM to store and retain information from previous time steps. The LSTM layer is followed by a Dropout layer ($dropout\_rate = 0.5$) acting as a regularization technique that avoids over-fitting by randomly ignoring neurons during training, Srivastava et al. (2014). Finally, the output layer is a fully connected dense layer that uses soft-max activation and outputs a binary label. To fit the model, we perform 500 epochs of stochastic gradient descent with adaptive learning rate, $lr = \frac{0.1}{\#epoch}$, using the $Adam$ optimizer, Kingma and Ba (2014).

The input to our LSTM-based model consists of the ordered sequence of consecutively chosen cards. Each card is represented by two features: The first feature encodes the card's position in the revealing order of motives; the second feature encodes the position of the card in the corresponding pair. Example: the first revealed card is always encoded with the feature vector (1,1). If the second revealed card shows the same motive, it is encoded with the vector (1,2) (second card of the first motive), otherwise with the vector (2,1) (first card of the second motive), see Figure 4.2. The advantage of this feature representation is that it is invariant to the actual motives but still captures the temporal relationships within a sequence. The encoded game logs vary in length between different sessions depending on the player's performance, i.e. the number of turns needed to finish the game. We aim at detecting the presence or absence of HCI obstacle as soon as possible, i.e. we favor short *game prefixes*. We will evaluate and discuss our detector for different game prefixes (5, 7, 10 and 15) in Section 4.2.
In addition to the raw behavioral data, we manually define statistical features, which are expected to differ between classes, for enriching the input training data. The following features summarize, for a game prefix (e.g. first 10 selected cards), how efficient and close-to-optimal a player performed: number of cards

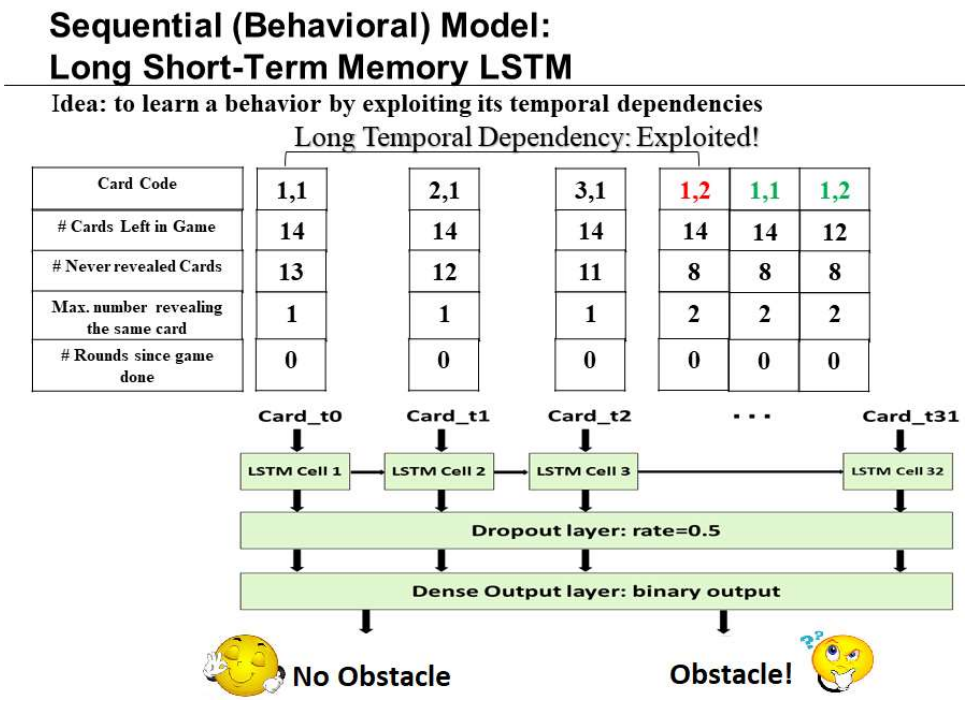


Figure 4.2 – LSTM feature vectors: LSTM learns temporal dependencies between time steps (selected cards) to detect the presence or absence of an HCI obstacle.

left in the game, number of never revealed cards in the game, maximum number of times revealing the same card and number of rounds since game completion (= 0 if game was not yet completed). For the LSTM, we calculate the features incrementally for each time step, see Figure 4.2. In contrast to the LSTM model, the LDA model considers a whole game at once. It only uses the manually defined features as one vector calculated after the end of the game (prefix). From these features, we train an LDA model (as baseline model) to classify logged games into NOOBS and MEMOBS labels. That is, the LDA model will receive only one feature vector per game as input. This vector is the last vector for the last revealed card, after which we want to evaluate both LDA and LSTM, more details for the evaluation are presented in Section 4.2. Based on these training vectors, the LDA performs a linear combination that best separate the data into the intended classes: NOOBS and *memObs*. Small number of training feature vectors, however, can lead to a poor empirical sample covariance estimator. To treat this problem, *Shrinkage LDA* is used as a regularization technique proposed by Ledoit and Wolf (2004) to improve the generalization performance of the LDA classifier.

4.1.3 EEG-based Detector

In this section, we discuss the EEG-based detection of HCI memObs. First, we briefly mention state of the art EEG-based memory classification models, and we discuss general related works in EEG-based prediction (Section 4.1.3.1). Then, we discuss our SVM model for detecting HCI memObs from EEG data (Section 4.1.3.2).

4.1.3.1 EEG Related Works

In this section, we discuss the EEG-based prediction literature for different human cognitive and emotional processes. Several user cognitive processes have been analyzed using EEG. Motor Imagery (MI) is a very prominent cognitive prediction application from EEG data. An early work by Pfurtscheller et al. (1998) discussed the online classification of right and left MI using a neural network. Hsu (2010) proposed a feature extraction method through the time-series prediction to improve the MI classification accuracy and AUC. The author used an adaptive neuro-fuzzy inference system (ANFIS) together with multiresolution fractal feature vectors (MFFVs) for feature extraction, where features are obtained from the differences of MFFVs between the predicted and actual signals, and then calculated through a window of EEG signals. For MI classification, Hsu used a linear discriminant analysis (LDA) to discriminate the extracted features vectors. Similarly, a recent work by Chaudhary et al. (2019) investigated Deep Learning to prepare an EEG features structure for MI classification. The authors transformed the input EEG signals into images by applying different time-frequency approaches, concretely the authors investigated the short-time-Fourier-transform and the continuous-wavelet-transform approaches. The resulting EEG-based images are then passed to a Deep Convolutional Neural Network (DCNN) to benefit from its very good ability in images classification. Hajinoroozi et al. (2016) discussed the prediction of driver's driving performance from EEG data. The authors retained EEG data samples from 5 seconds before each perturbation that led to either a poor or a good driving performance state. They proposed a channel-wise convolutional neural network (CCNN) to consider the unique characteristics of EEG data.
Not only cognitive processes but also emotions can be modeled by EEG: An interesting EEG-based prediction approach has been proposed by Kumar et al. (2016) to model emotions. The authors collected EEG data from participants while showing them video stimuli of four representative primary emotions based on the Navarasa theory of Ancient Indian treatise called Natya Shastra. It models nine emotions, from which the authors chose the following four

primary emotions for their experiments: Shringara (love), Raudra (anger), Veera (courage) and Bheebhatsya (disgust). The EEG analysis findings were compared with the participant's self-reports about their emotional states during the experiment. The authors concluded that human emotions can be modeled for use in HCI either as an affect assessment tool or for affect based intelligent interactions.

4.1.3.2 Discriminant EEG-based Model

In this section, we discuss our discriminant EEG-based model in three main points: first, we describe and argue the chosen classifier. Second, we briefly recall the EEG data processing which was discussed in Chapter 3, and we introduce and argue our EEG-bands based feature selection. Finally, we explain what we mean with user states, how our classifier modeled them, and how we aggregate the modeled user states for sequence-based prediction: noObs or memObs.

First, regarding the chosen classifier, we follow Lotte (2014), who found Support Vector Machines (SVM) to be effective for the classification of EEG signals. Lotte also reported that for typical small EEG data sets, neural networks rarely show performance improvements compared to other methods. Therefore, we investigate the detection of memory-based HCI obstacles from EEG data using SVM.
We use an SVM model with default parameters (Radial Basis Function (RBF) kernel, $C = 1.0$, gamma=0.1) to classify the EEG data of each tested event (revealed card) into its classes, noObs or memObs. We decided to not optimize SVM parameters on hold-out data to avoid overfitting on the relatively small sample size.

Second, regarding the feature selection, we briefly recall the EEG data processing discussed in Chapter 3, where we used an EEG sampling frequency = 500 Hz. According to the state of the art of EEG-based models, e.g. Kumar et al. (2016), EEG should be filtered and only low frequency bands in EEG are relevant for pattern detection (obstacles in our case), because high-frequency bands are expected to be contaminated with artifacts that impede classifiers from learning the data features. Kumar et al. (2016) reviewed several EEG-based prediction tasks and summarized results as which EEG band is recommended to which task. According to their review results, the power in the Delta band (0 - 4 Hz) increases during complex mental tasks. Additionally, the review results show that higher working memory loads have been reported to cause increase in low Beta bands (12 - 18 Hz) powers in frontal mid-line regions. We find those bands (Delta and low Beta) suitable

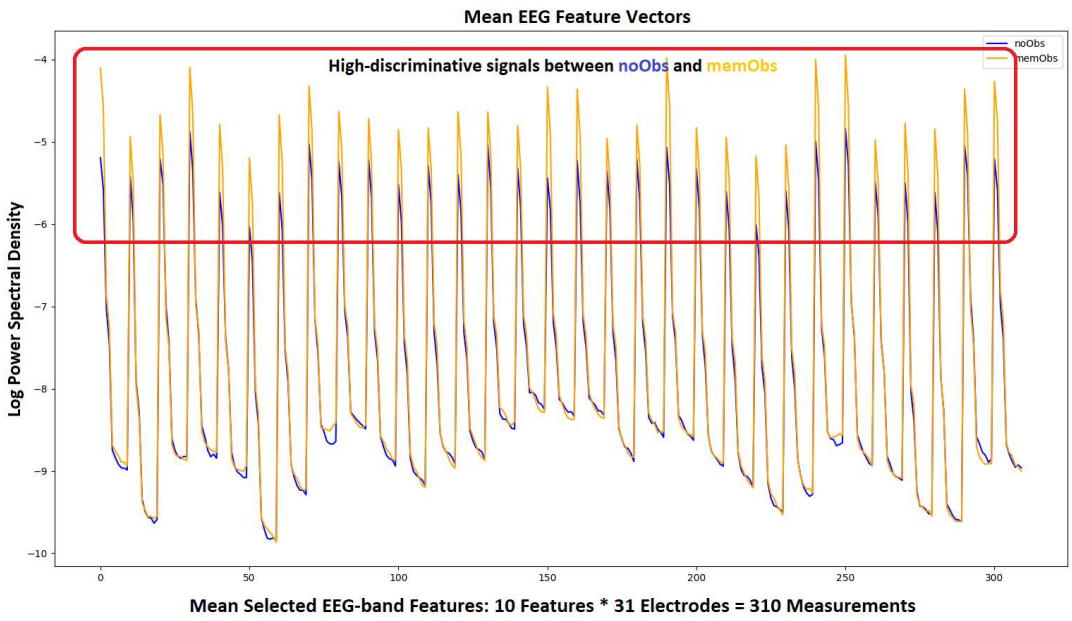


Figure 4.3 – EEG signals observed by 32 electrode actiCAP. Signals were observed in one-second window after revealing a card from a NoObs game and from MemObs game. 10 features are selected from each electrode (31 electrodes excluding the reference one). The 310 features observed from all the electrodes are averaged over all the events (revealed cards) and all the subjects (person-independent features). High-discriminative signals are shown between noObs and memObs.

for matching pairs game in general. Concretely, we expect those bands to be discriminative between noObs and memObs game variants because they are different mental tasks showing different memory loads, see Figure 4.3 which shows significantly lower spectral power in the beta band in lower WM load condition (blue line = noObs) than in high WM load conditions (orange line = memObs).

Finally, we discuss how we modeled user states, which are aggregated for sequence-based detection of HCI memObs. The user states in our HCI matching pairs game are the noObs or memObs states, which are detected after each event (revealing a card). Concretely, we modeled the reveal events by extracting EEG segments of 1000 ms (using a Hanning window) following the reveal event of each card after clicking it. We chose this 1000 ms segment as it can be connected with a relatively clear cognitive process of recall and

memorization after a new card has been revealed, reducing the variance compared to a sliding window across different segments of the data. Data from each channel (electrode) is normalized to zero mean to account for offsets and slow drifts. Thus, the SVM model, described above, detects a user state from physiological EEG data at a specific event, namely the point in time at which a card was revealed, by discriminating data in that mentioned 1000 ms window following the said reveal event.

The employed SVM predicts an individual label e.g. MEMOBS for each such an event (revealing a card). The predicted labels for the revealed cards represent the modeled user cognitive states: NOOBS or MEMOBS states. For the integration of information over time, we aggregate the SVM predictions of the first 10 turns (20 cards) and vote the predicted label and average the corresponding confidences of the individual SVM predictions for the whole sequence. More concretely, we assign the label with the highest average confidence calculated according to Platt scaling, which typically transforms the output of a classification model into a probability distribution over classes, Platt et al. (1999). We perform this aggregation for two reasons: 1) If the event-based classification performs better than the random baseline, such a confidence-based aggregation is expected to improve the prediction accuracy, and 2) it enables a comparison to the behavioral predictions which are based on sequences, and thus, it enables a UI ADAPTATION decision between the SVM and the LSTM sequence-based predictions, see the next chapter for more details.

4.2 Evaluation of Memory-based Obstacle Detection

In this section, we evaluate the introduced MEMOBS detectors under different conditions, i.e. under different UI adaptation mechanisms: NOADAPT, MEMADAPT and VISADAPT. We evaluate the detectors for the two modalities: behavioral (Section 4.2.1) and EEG (Section 4.2.2).

4.2.1 Evaluation on Behavioral Data

In this section, we evaluate our LSTM model, which has been introduced in Section 4.1.2 as a behavior-based MEMOBS detector. We discuss the evaluation regarding two perspectives: obstacle type in Section 4.2.1.1 (volatile or persistent) and UI adaptation type in Section 4.2.1.2 (NOADAPT, MEMADAPT and VISADAPT). For each discussed model, we mention its corresponding

data sets from Chapter 3, on which the said model has been trained and evaluated.

As we aim to detect HCI obstacles as soon as possible during an HCI session, we evaluate our behavior-based models after the $10th$ round, because it is plausible to model different player behaviors (e.g. whether the player recalled or forgot the revealed cards) after playing 10 rounds in our 14-cards game which is described in Chapter 3.

4.2.1.1 Obstacle Types

We discuss this section in three main points: motivation, experimental setup, and results & discussion.

Motivation: as introduced in Chapter 1, HCI obstacles can occur permanently (persistent obstacle) or temporary (volatile obstacle) in HCI sessions, therefore, it is important to treat both persistent and volatile HCI obstacles in our cognitive adaptive system. For HCI memObs, a volatile HCI memObs typically occurs because of a secondary task (temporary) memory workload, while a persistent HCI memObs typically occurs because of low WM capacity during achieving an HCI task, due to the task complexity or user lack of experience with the HCI task etc.
Moreover, it is valuable to investigate the transfer learning ability of our models, i.e. to investigate how well our models perform in other real-world experimental settings. Thus, regarding to persistent HCI memObs, we investigate how our models, which have been trained with logs collected from young subjects, generalize to detect *dementia* which is a typical HCI memObs.

Setup: According to obstacle types, we trained and evaluated the following three behavior-based memObs detectors:

1. *Volatile* memObs *Detector:* aims at detecting the presence or absence of secondary task WM load (cumulative sum) as a volatile HCI memObs, following our works in Putze et al. (2018); Salous et al. (2019). We train and evaluate this detector using the leave-one-out cross-validation method on the TapData collections collected from young subjects as described in Section 3.1.1.1 and Section 3.1.1.2

2. *Persistent* memObs *Detector (Low WM Capacity):* In the volatile memObs detector mentioned in point 1, the participants played two sessions, with and without that simulated memObs (cumulative sum). For the detection of low WM capacity, in contrast, the participants

played only one game variant (Matching pairs with cumulative sum) as a complex HCI session, and the MEMOBS detector here aims at detecting the subjects who have low WM capacity (persistent HCI MEMOBS) during achieving that complex HCI task, Salous and Putze (2018). We train and evaluate the persistent MEMOBS detector according to Monte-Carlo cross-validation method to reduce expected variability with such a relative small data set collected from 24 subjects, recall Section 3.1.1.1 for details about the experiment setup and the data collection.

3. *Persistent* MEMOBS *Detector (Dementia):* as mentioned in Chapter 1, we test the transfer learning capability of our *Volatile* MEMOBS *Detector* (point 1) by testing it on another sample and another experimental settings: elderly healthy subjects and dementia patients playing the standard matching pairs game variant. The collected logs from dementia patients (Section 3.1.1.3) should be classified as MEMOBS, while the collected logs from elderly healthy subjects should be classified as NOOBS: these logs are combined from TABEXP elderly user study (Section 3.1.1.4) and WEBDATA for subjects whose $age > 60$ in Section 3.3.2. We prepared balanced subject-based test data with 30 subjects, 15 elderly healthy test subjects ($age > 60$) whose logs should be detected as NOOBS logs vs. 15 dementia patients subjects whose logs should be detected as MEMOBS. We collected multiple logs from both elderly healthy subjects and dementia patients.
   We chose to test the transfer learning for the *Volatile* MEMOBS *Detector (Secondary Task)* rather than the *Persistent* MEMOBS *Detector (Low WM Capacity)*, because the latter detector has been trained only from matching pairs logs with the cumulative sum task to detect a low WM capacity, while the former detector has been trained with matching pairs logs (NOOBS) and matching pairs logs with the cumulative sum task (MEMOBS) showing a remarkable drop in user performance, which can be better generalized to the performance drop in the dementia testing sample: matching pairs logs played by healthy elderly subjects (NOOBS) and matching pairs logs played by dementia patients (MEMOBS).

Results & Discussion: we discuss the obstacle types results in three steps. First, we begin with *Results Description*, in which we explain and compare the results of all obstacle types reported in Table 4.1. Second, for the *Low WM Capacity Detection*, we further discuss it with a detailed analysis depicted in Figures 4.4 and 4.5. Finally, for the *Dementia Detection*, we discuss it with more details and confusion matrices.

Table 4.1 – Behavior-based MEMOBS detector for volatile and persistent HCI MEMOBS. The detector accuracy, presented for each obstacle type, is calculated at the 10th round (20 revealed cards), because after playing 10 rounds in a game with 14 cards, user behaviors of recalling or forgetting cards can be depicted from user actions.

LSTM MemObs Detector	Accuracy(%)
Chance Level	50.00
Volatile MEMOBS (HCI Secondary task WM Load)	75.90
Persistent MEMOBS (HCI Low WM Capacity)	76.66
Persistent MEMOBS (Dementia)	90.10

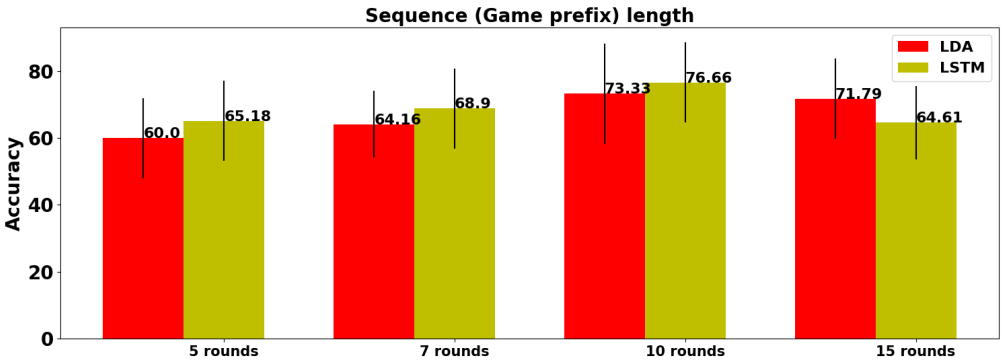


Figure 4.4 – Behavior-based sequential model LSTM for detecting low WM capacity MEMOBS after different game prefix lengths. LSTM is compared to a baseline LDA model which has been trained on the same statistical feature vectors.

Results description: Table 4.1 shows the detection performance, at the $10th$ round, of LSTM as a behavior-based MEMOBS detector for volatile and persistent obstacle types. All the volatile and persistent MEMOBS detectors outperform the baseline chance level. The best performance is shown by the dementia detector model with 90.10%, and can be explained by the modeling of the drop in playing performance which occurs in MEMOBS sessions, which is expected to be the largest in case of dementia compared to volatile MEMOBS and low WM capacity.

Low WM Capacity Detection: we further analyzed the detection performance of low WM capacity (76.66%) in Salous and Putze (2018). Namely, we compared the sequential LSTM model to a baseline model LDA which uses exactly the same feature vectors, and we compared the LDA and LSTM for

different game prefix lengths: 5, 7, 10 and 15. Figure 4.4 shows that for game prefix lengths 5, 7 and 10, both LDA and LSTM exploit the richer information of longer game prefixes as accuracy increases with the length of game prefix. In addition, the results show that LSTM outperforms LDA for short sequences. The improvement is statistically significant for short game prefixes 5 and 7 ($p = 0.001$, $p = 0.006$ and $p = 0.1$ for 5, 7 and 10 respectively, calculated using a paired t-test on the results of individual iterations). This can be interpreted by the LSTM sequential structure (recall Section 4.1.2 and Figure 4.2): LSTM can exploit, besides the incremental measurements features, temporal dependencies in input sequential data, whereas LDA uses only those static measurement features (One feature vector at the end of the tested sequence). However, most of the participants finished the Matching Pairs game before 15 rounds, i.e. raw sequential data contains no additional information for the last rounds of such games. This explains why LSTM accuracy decreases for game prefix length of 15.

Figure 4.5 shows that the difference between MU+ (high WM capacity subjects) and MU- (low WM capacity subjects) is largest between 10 and 15 rounds, which makes the discrimination easy for the LDA model. Figure 4.5 also shows a difference between MU+ and MU- behaviors for shorter prefixes, although standard deviation is high. This implies that partitioning our data into MU+ and MU- (high and low WM capacity) is feasible, even for short sequences, but predicting WM capacity cannot be immediately done from simple manual features. Therefore, the ability of the LSTM model to learn more complex temporal relationships results in improved performance.

Dementia Detection: we aimed at detecting dementia as a typical persistent HCI memObs. As mentioned above, we tested the volatile memObs detector (trained by logs collected from young subjects) with testing data collected from elderly subjects: matching pairs game logs collected from elderly healthy subjects (should be classified as noObs) and from dementia patients (should be classified as memObs). Given that the drop in play performance is expected to be largest in case of dementia, we expect the investigated young subjects based detectors (trained by logs collected from young subjects) to perform well when they are tested with elderly subjects logs (healthy elderly subjects and dementia patients). Results in Table 4.1 coincide with our expectation, where the highest performance is for the detection of dementia with 90.10%.

Figure 4.6 shows that our model perfectly detected healthy elderly subjects logs as noObs ones with 100% as an average accuracy. It also shows that 82.18% of dementia patients logs were detected correctly as persistent memObs. As mentioned above, multiple logs were collected from elderly healthy subjects

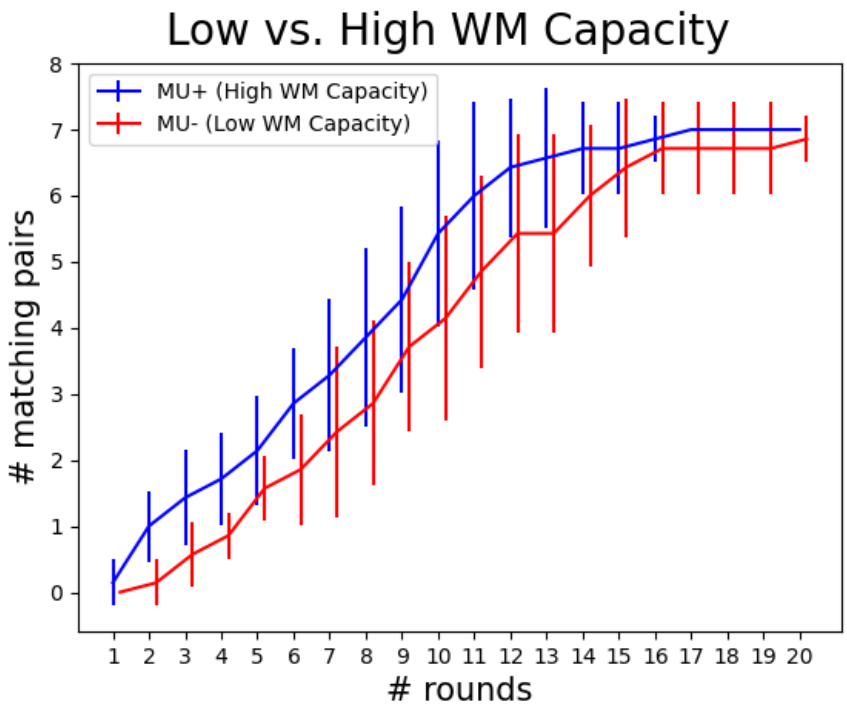


Figure 4.5 – WM Capacity Analysis: all the participants played the same HCI complex task (Matching pairs with cumulative sum). Participants with high WM capacity (MU+, calculated based on the standard MU test, Section 3.1.1.1) outperform subjects with low WM capacity (MU-). Vertical whiskers show standard deviation of calculated means.

and from dementia patients. These multiple logs guarantee more robust detection test, where only one individual matching pairs game is likely effected with luck. Moreover, we can benefit from those multiple logs collected as follows: we can vote the labeled logs of one tested subject to label that tested subject as healthy noObs or dementia patient memObs. Such Vote-based Labeling improves the results as follows: the dementia logs accuracy shown in the confusion matrix in Figure 4.6 was 82.18%, and it has been improved to 86.67% when voting the logs and labeling the subject as dementia patient, see Figure 4.7. In total the dementia detection accuracy is improved by voting mechanism from 90.10% to 92.95%, see Figures 4.6 and 4.7.

Thus, we contribute to the state of the art models for detecting dementia using ML solutions; We introduce this way a robust behavior-based, sensory-free novel model for detecting dementia from matching pairs game. As the dementia detection performance improved from 90.10% to 92.95% by voting on the

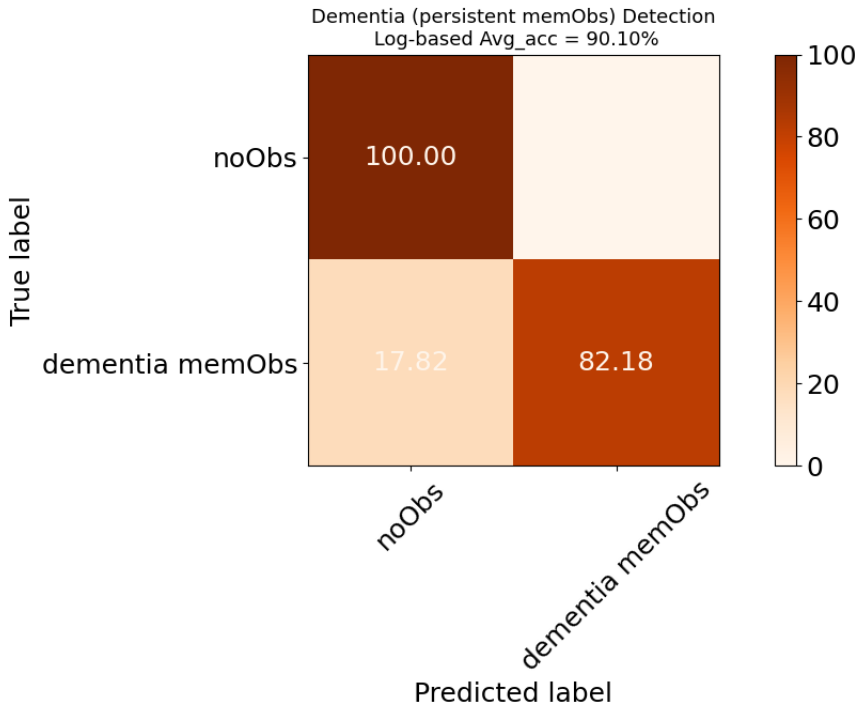


Figure 4.6 – Behavior-based detecting of dementia as an HCI persistent obstacle: Log-based evaluation.

predicted labels of multiple games played by a test subject, we conclude that the more the games played by a test subject, the better dementia detection accuracy our model performs.

4.2.1.2 HCI Conditions

We discuss this section in three main points: motivation, experimental setup, and results & discussion.

Motivation: Our ultimate cognitive adaptive system aims at an online UI adaptation during HCI (Chapter 5). Given that such an adaptation may not be optimal from the first try, the introduced system aims at continuously detecting HCI obstacles after each interaction. Thus, we need to train models for detecting HCI obstacles under different adapted UIs, i.e. under appropriate decided UI adaptation (MEMOBS_MEMADAPT) and under inappropriate decided UI adaptations (MEMOBS_NOADAPT and MEMOBS_VISADAPT).

Setup: In this section, we discuss three different MEMOBS detectors that all detect the same HCI MEMOBS, but under different conditions (UI ADAP-

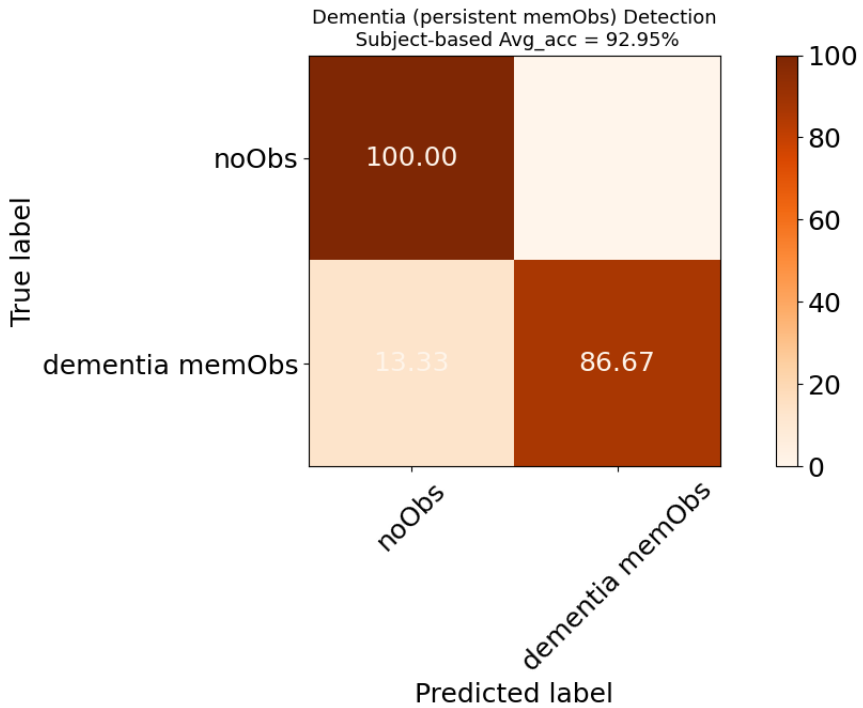


Figure 4.7 – Behavior-based detecting of dementia as an HCI persistent obstacle: Subject-based evaluation.

TATIONS). Namely, we trained models for detecting MEMOBS_NOADAPT, MEMOBS_MEMADAPT and MEMOBS_VISADAPT. Recall Chapter 3 for details about the MEMADAPT mechanism (re-revealing last revealed cards) and the VISADAPT mechanism (auditory identifiers to recognize the cards).

Results & Discussion:

Table 4.2 – Behavior-based MEMOBS LSTM detector for volatile HCI MEMOBS under different conditions: NOADAPT, MEMADAPT and VISADAPT

LSTM MemObs Detector	Accuracy(%)
Chance Level	50.0
MemObs NOAdapt	75.9
MemObs MemAdapt	64.1
MemObs VisAdapt	69.9

Table 4.2 shows that the MEMOBS detection performance deteriorates under different UI ADAPTATIONS; The most significant difference happens when trying to detect HCI MEMOBS under the corresponding UI Adaptation

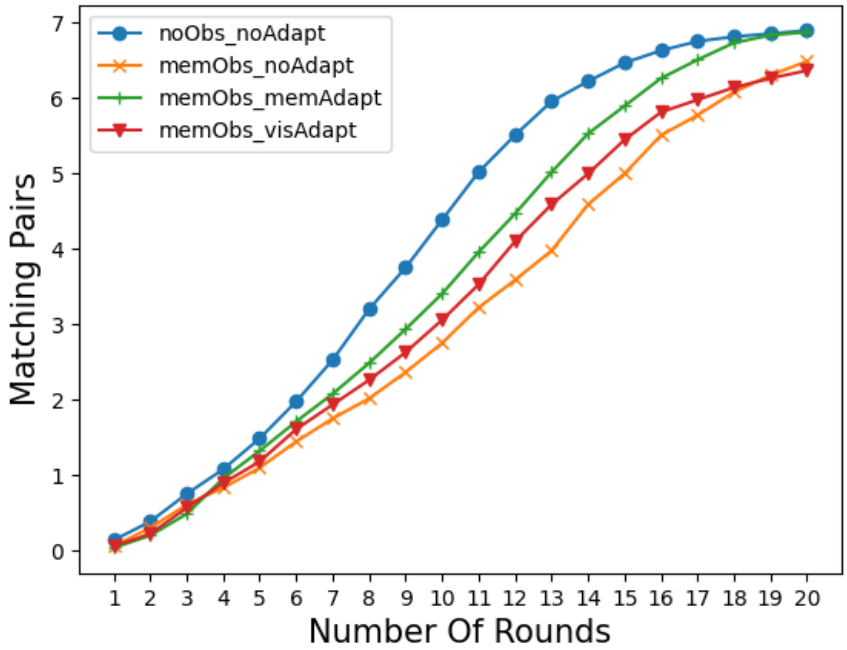


Figure 4.8 – HCI MEMOBS and UI ADAPTATION Analysis: largest difference shown between NOOBS and MEMOBS_NOADAPT. Player performance improves slightly from MEMOBS_NOADAPT to MEMOBS_VISADAPT, and more improved with the corresponding adaptation MEMOBS_MEMADAPT

(MEMADAPT), where accuracy deteriorates from 75.9% to 64.1%. Even when adapting the UI with non corresponding adaptation (VISADAPT for MEMOBS), the accuracy drops to 69.9%, but not as significant as the MEMADAPT case.

These results coincide with the data collection analysis findings discussed in Chapter 3; That is, both subjective and objective analysis showed that MEMADAPT better suits the HCI MEMOBS session than VISADAPT.

Moreover, Figure 4.8 shows that the corresponding UI adaptation mechanism of HCI MEMOBS (MEMADAPT) brings the largest improvement in performance (difference between MEMOBS_NOADAPT and MEMOBS_MEMADAPT lines), while the non-corresponding one (VISADAPT) slightly improves the performance (MatchingPairs per round). This explains the differences in Table 4.2, where it is the easiest for MEMOBS detector to detect such an HCI obstacle with NOADAPT, and a bit harder to detect it when supporting the player with a non-corresponding UI ADAPTATION, and it becomes more difficult to detect the obstacle when applying the corresponding UI

adaptation mechanism: MEMOBS_MEMADAPT; In all the three conditions, the MEMOBS detector is able to detect the presence of MEMOBS with good accuracy, significantly outperforming both random guess baseline and the static LDA baseline.

4.2.2 EEG based Evaluation

In this section, we evaluate the EEG-based SVM MEMOBS detector. We discuss this section in three main points: motivation, experimental setup, and results & discussion.

Motivation: The EEG data reflects user's cognitive states during an HCI session. In contrast to USER BEHAVIOR which lasts for relative long temporal context, USER STATE represents mainly the current, actual status. Concretely, in matching pairs HCI task, USER STATE can be observed after each user action (revealing a card). With our multimodal based approach, we want to benefit from both behavioral and EEG modalities, where EEG-based user states complement the modeled user behavior for more robust prediction of HCI obstacles.

Setup: For classification, EEG-based models typically predict one user state at a time. For our EEG-based MEMOBS detector, the predicted user states are NOOBS or MEMOBS for each revealed card. As we aim at multimodal based obstacle detection, we should synchronize the prediction point of time between our data modalities to enable a fusion between their predictions: EEG-based and behavior-based predictions. Thus, as we decided in Section 4.2.1 to predict on behavioral data along 20 user actions (first 20 revealed cards), we should accordingly aggregate the EEG-based predictions of the first 20 revealed cards and vote the predicted label as a sequence-based prediction. More details about the multimodal based fusion are presented in Chapter 5.

Results & Discussion: As mentioned above, we use the leave-one-out cross-validation method for evaluation, that means our EEG-based models are PERSON INDEPENDENT detectors.

Table 4.3 shows the person-independent performance of the EEG-based SVM for detecting MEMOBS under different UI ADAPTATIONS. If we compare Table 4.3 to Table 4.2, we notice that while the behavior-based and EEG-based detectors are very close for NOADAPT (ca.76%), the EEG-based SVM outperforms the behavior-based LSTM for detecting MEMOBS under different UI ADAPTATION. The difference is highly significant in case of applying the non-corresponding UI ADAPTATION (VISADAPT for MEMOBS), where the

Table 4.3 – EEG-based MEMOBS person-independent detector for volatile HCI MEMOBS under different conditions: NOADAPT, MEMADAPT and VISADAPT. The reported resutls are for EEG-based sequence prediction for the first 10 turns, i.e. for the aggregated and voted EEG-based predictions for the first 20 reveald cards.

SVM MemObs Detector	Accuracy(%)
Chance Level	50.0
MemObs NOAdapt	76.13
MemObs MemAdapt	71.59
MemObs VisAdapt	82.95

LSTM slightly dropped to 69.9% while the EEG-based SVM, in contrast, significantly improved to 82.95%. We argue such a difference by two points:

1. The VISADAPT auditory identifiers (Chapter 3) do not naturally suit the HCI task under the MEMOBS (cumulative Sum). EEG signals reflect the user's cognitive process, thus, EEG signals are expected highly discriminant between the cognitive process for NOOBS session and the confused cognitive process for MEMOBS_VISADAPT. Consequently, high discrimination performance is resulted from EEG (82.95%).

2. In contrast to the physiological EEG data, the behavioral data can better depict potential advantages even from that non-corresponding UI ADAPTATION (VISADAPT for MEMOBS). This makes the behavior-based discrimination (a bit) more difficult, because the user performance for MEMOBS_VISADAPT is (slightly) improved, see Figure 4.8.

4.3 Visual Obstacle Detector in HCI

In this section, we discuss the detection of persistent and volatile HCI VISOBS. As introduced in Chapters 1 and 3, we simulate and detect both obstacles in matching pairs as an exemplary HCI task: glare-effects on mobile display as a volatile VISOBS, and color blindness disease as a persistent VISOBS, as published in Salous et al. (2019).
First, we begin in Section 4.3.1 with discussing state of the art models for VISOBS detection. Then, we discuss in Section 4.3.2 the binary classifiers we designed to detect HCI VISOBS from different modalities: *Behavior-based* VISOBS detector and *EEG-based* VISOBS detector.

4.3.1 State of The Art

In this section, we discuss state of the art models which aimed at detecting HCI persistent and volatile visObs.
An early work presented by Jefferson and Harvey (2007) introduced an interactive interface for users with color vision deficiency. Users can customize the colors of any region of the screen. Thus, it is a specific interface for such a user group and it permits the user to capture, clear, copy, save the image, and view the correction control window. There is no detection of color vision deficiency obstacle in this interface. Due to the large number of color blind people (about 8% of men and 0.5% women in the world have the common form of color blindness, Ostia et al. (2019)), and due to its strong impairment expected in HCI performance, there exist works that aim at online detecting color vision deficiency visual obstacle and online compensating with corresponding UI adaptation. For example, Qaiser and Khan (2017) discussed the online detection of such an obstacle by utilizing a typical color vision test automated in mobile app. After that test, the most appropriate color scheme for that specific user is chosen. The limitation of such a UI that the specific color test cannot be smoothly integrated in HCI applications. Although Khan's UI adaptation has an advantage over Jefferson's approach as the most appropriate color scheme is chosen automatically, it can be further optimized by utilizing another communication ability e.g. by voice instructions.
Glare effects, which we detect as a volatile HCI visObs, has been discussed in early and recent HCI standards and surveys, e.g. Stewart (1992); Bevan (2001); Ozok (2009). New mobile devices utilize light sensor technologies for the detection of glare effects and an automatically adjust brightness, Kreek et al. (2010); Cheng and Bai (2012). However, such solutions depend solely on light sensors regardless behavior-based user modeling. Moreover, only adjusting light brightness may not be an optimal UI Adaptation according to environmental conditions (e.g. too sunny weather) or individual conditions (e.g. light sensitivity).
To the best of our knowledge, there is no approach that detects color blindness or glare effects obstacles from user behavioral data to adapt the UI by utilizing an additional user cognitive ability; We introduce our behavior-based detector of color vision deficiency as a persistent HCI visObs and glare-effects as a volatile HCI visObs. We aim at automatic compensation of such obstacles by utilizing additional cognitive process besides recognition: hearing of auditory instructions, as published in Salous et al. (2019).

4.3.2 Behavior-based and EEG-based Detectors

Similar to the behavior-based and EEG-based MEMOBS detectors introduced in Sections 4.1.2 and 4.1.3, we aggregate in this section the discussion of both behavior-based and EEG-based VISOBS detectors, because they have exactly the same structure and topology of MEMOBS detectors: LSTM model for behavior-based VISOBS detector and SVM model for EEG-based VISOBS detector. The only difference between those MEMOBS and VISOBS detectors

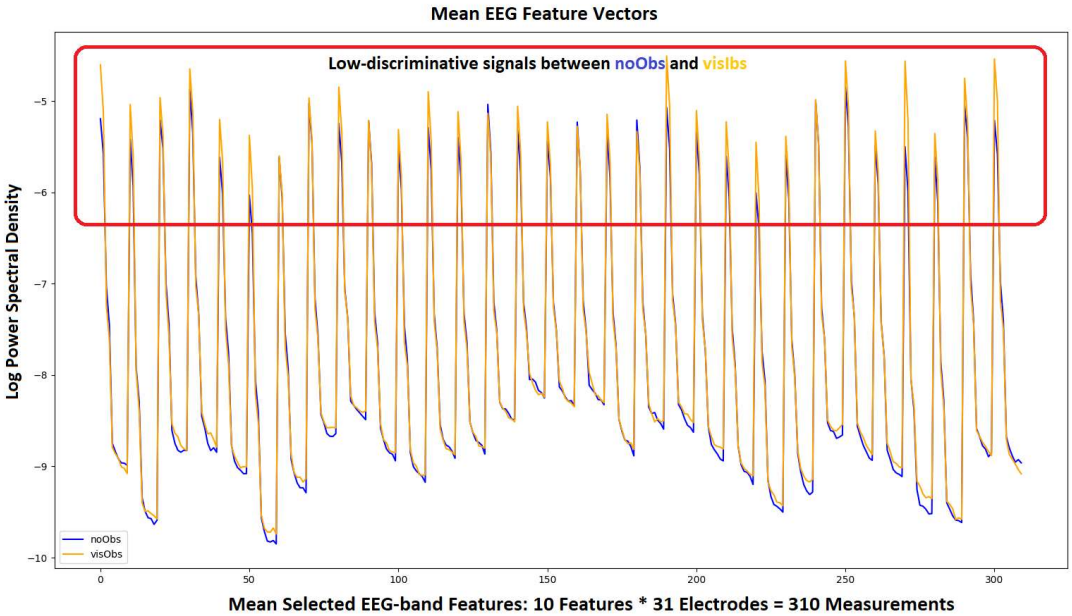


Figure 4.9 – EEG signals observed by 32 electrode actiCAP. Signals were observed in one-second window after revealing a card from a NOOBS game and from VISOBS game. 10 features are selected from each electrode (31 electrodes excluding the reference one). The 310 features observed from all the electrodes are averaged over all the events (revealed cards) and all the subjects (person-independent features). Low-discriminative signals are shown between NOOBS and VISOBS, compared to high-discriminative signals between NOOBS and MEMOBS in Figure 4.3.

is the corresponding impaired COGNITIVE PROCESS; While the MEMOBS exclusively impairs the memorizing and recall process, the VISOBS mainly, but not exclusively, impairs the visual recognition process, as a result, the memorizing and recall process is also impaired as a main process utilized during matching pairs HCI.

Since the visual recognition process is impaired, confused behavior is expected and can be modelled by LSTM, but the EEG data will not be as discriminant as in the memObs case, because the noObs and visObs sessions represent the same mental task but with different cards set (see how EEG signals are low-discriminant (interleaved) in Figure 4.9). Thus, in contrast to the memObs data, the visObs data is expected to better suit behavior-based LSTM rather than EEG-based SVM. We investigate our expectation via an evaluation in Section 4.4, and we conclude and summarize the findings in the final discussion in Section 4.6.

4.4 Evaluation of Visual Obstacle Detection

We discuss in this section the evaluation of visObs detectors. We use the leave-one-out cross validation method to train and evaluate both behavioral models and EEG-based person independent models.

4.4.1 Evaluation on Behavioral Data

In this section, we evaluate the behavior-based visObs detector regarding to two perspectives: obstacle type (volatile or persistent) and UI Adaptation type (noAdapt, memAdapt and visAdapt).

4.4.1.1 Obstacle Types

We discuss this section in three main points: motivation, experimental setup and results & discussion.

Motivation: According to Ostia et al. (2019), about 8% of men and 0.5% women in the world have the common form of color blindness. Due to its strong impairment expected in HCI performance, we simulated and investigated the detection of the red-green color vision deficiency as a persistent HCI visObs. Not only a persistent HCI visObs, but also a volatile HCI visObs, which occurs temporarily, impairs the HCI user performance. Glare-effects obstacle is a typical volatile visObs which impairs the HCI performance when the sunlight weakens the visual recognition on a mobile display (Stewart (1992); Bevan (2001); Ozok (2009)). As mentioned above, existing approaches utilize light sensors to detect a glare-effects obstacle and automatically adjust the brightness on the mobile display (e.g. Kreek et al. (2010); Cheng and Bai (2012)). Our contribution regarding the glare-effects detection is the modeling

of a user's confused behavior which is expected during a glare-effects HCI session.

Setup: We have two behavior-based visObs detectors which aim at detecting the presence of glare-effects on mobile display as a volatile visObs, and the presence of color blindness disease as a persistent visObs, Salous et al. (2019). For our models, we collected behavioral data from noObs sessions (Figure 3.4), volatile glare-effects visObs sessions (Figure 3.6) and persistent color-blindness visObs sessions (Figure 3.5). For more details, recall Chapter 3.

Results & Discussion: Table 4.4 shows the detection performance of the

Table 4.4 – Behavior-based visObs detector for volatile and persistent HCI memObs.

LSTM VisObs Detector	Accuracy(%)
Chance Level	50.0
Volatile glare-effects VisObs	77.3
Persistent color-blindness VisObs	72.1

behavior-based visObs detector for the aforementioned visObs types. While the results outperform the baseline random chance level (50%), we do compare the introduced LSTM, as a sequential model to a baseline static model LDA which uses exactly the same statistical measurements as features vector (four statistical features calculated sequentially (for each revealed card) for LSTM and statically at the game end (prefix) for LDA, recall Section 4.1.2). Similar to memObs detectors, we found that LSTM, for different obstacle types, significantly outperforms the baseline LDA model ($p < 0.05$, calculated using a paired t-test on the result of individual iterations). Again, this can be interpreted as LSTM exploits, besides the incremental measurements features, temporal dependencies in input sequential data, whereas LDA uses only those static measurement features.

4.4.1.2 HCI Conditions

We discuss this section in three main points: motivation, experimental setup, and results & discussion.

Motivation: Our ultimate cognitive adaptive system aims at an online UI adaptation during HCI (Chapter 5). Given that such an adaptation may not be optimal from the first try, the introduced system aims at continuously detecting HCI obstacles after each interaction. Thus, we need to train models

for detecting HCI obstacles under different adapted UIs, i.e. under appropriate decided UI adaptation (visObs_visAdapt) and under inappropriate decided UI adaptations (visObs_noAdapt and visObs_memAdapt).

Setup: We discuss three different visObs detectors that all detect the same HCI visObs, but under different conditions (UI Adaptations). Namely, we trained models for detecting visObs_noAdapt, visObs_memAdapt and visObs_visAdapt. Recall Chapter 3 for details about the memAdapt mechanism (re-revealing last revealed cards) and the visAdapt mechanism (auditory identifiers to recognize the cards).

Results & Discussion:

Table 4.5 – Behavior-based visObs detector for HCI visObs under different conditions: noAdapt, memAdapt and visAdapt

LSTM VisObs Detector	Accuracy(%)
Chance Level	50.0
VisObs NoAdapt	72.1
VisObs MemAdapt	66.3
VisObs VisAdapt	56.2

Table 4.5 shows that the visObs detection performance deteriorates under different UI Adaptations; The most significant difference happens for the detection of HCI visObs under the corresponding UI Adaptation (visAdapt), where accuracy deteriorates from 72.1% to 56.2%. Even when adapting the UI with non corresponding adaptation (memAdapt for visObs), the accuracy drops to 66.3%, but not as significant as the visAdapt case.

These results coincides with the data collection analysis findings discussed in Chapter 3, where both the subjective and objective analysis showed the visAdapt perfectly suits the HCI visObs, but the repetition of information in the memAdapt also a bit improves the player performance.

Moreover, Figure 4.10 shows that the corresponding UI Adaptation of HCI visObs (visAdapt) significantly improves the player performance (even exceeding the player performance in noObs_noAdapt). Convenient statistical test (paired t-test) proves that the improvement from visObs_noAdapt to visObs_visAdapt is significant ($p < 0.05$), while the performance improvement from visObs_noAdapt to visObs_memAdapt is insignificant, and the difference between noObs_noAdapt and visObs_visAdapt is insignificant, too ($p > 0.05$ in either case).

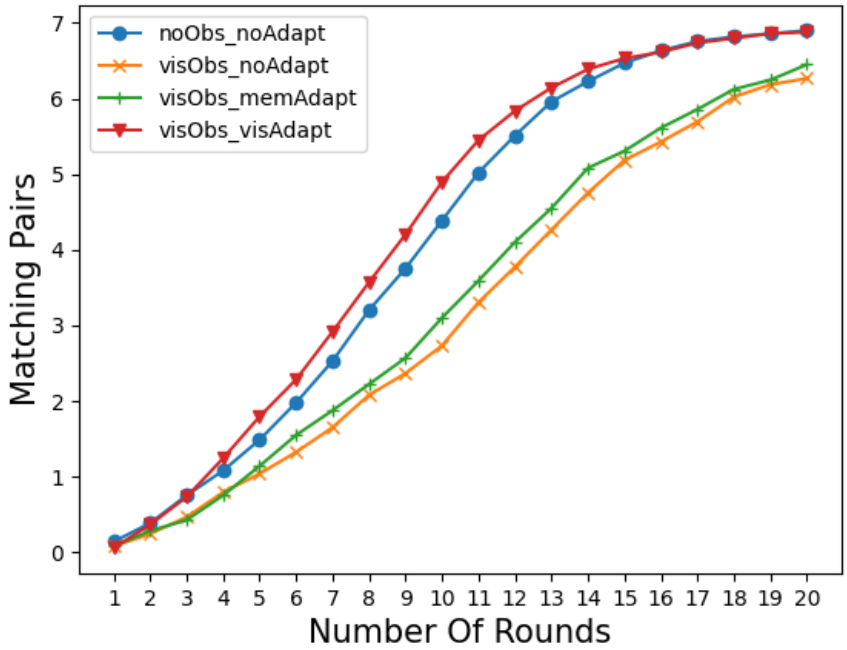


Figure 4.10 – HCI visObs and UI Adaptation Analysis: visAdapt completely tackles the visObs, where player performance improves significantly from visObs_noAdapt to visObs_visAdapt. Player performance slightly improves from visObs_noAdapt to visObs_memAdapt

As a result, both the visObs_noAdapt and visObs_memAdapt detectors are able to detect the presence of visObs with good accuracy, significantly outperforming both chance level baseline and the static discriminant baseline (LDA). However, the detection of visObs under visAdapt is challenging, because the visObs_visAdapt detector has very close behavioral data to learn from, namely noObs and visObs_visAdapt, see the red and blue lines in Figure 4.10.

This explains the significant drop in the performance of visObs_visAdapt detector to only 56.2%, which is slightly better than the chance level (50%). This is logical due to the high suitability of visAdapt to tackle the visObs. This challenge will be tackled then by the cognitive adaptive system in the next chapter, where the probabilistic system is able to learn data distributions from the Elementary Models outputs (predictions and competencies) to consolidate or revert the decided UI Adapt, see Chapter 5 for more details.

4.4.2 EEG based Evaluation

We evaluate in this section the EEG-based SVM visObs detector under different UI adaptations. The motivation for modeling EEG-based cognitive states and the EEG experimental setup are similar to what discussed in Section 4.2.2. Thus, we directly present the results & discussion.

Results & Discussion: Table 4.6 shows person-independent performance

Table 4.6 – EEG-based visObs person-independent detector for HCI visObs under different conditions: noAdapt, memAdapt and visAdapt

SVM VisObs Detector	Accuracy(%)
Chance Level	50.0
VisObs NOAdapt	64.77
VisObs MemAdapt	61.63
VisObs VisAdapt	42.04

of EEG-based SVM for detecting visObs under different UI Adaptations.

In contrast to the memObs case, if we compare Table 4.6 to Table 4.5, we notice that the behavior-based visObs detector significantly outperforms the EEG-based visObs detector for different UI Adaptations.

The difference is especially important in the visObs_visAdapt case, where subjective and objective analysis in Chapter 3 show that subjects feel and do perform very well with visAdapt, and this makes the visObs non-detectable (less than chance level) from physiological EEG data (42.04% in Table 4.6).

However, the behavior-based detector LSTM can, even not robustly, distinguish between such very close labels: noObs and visObs_visAdapt, slightly outperforming the chance level (56.2% in Table 4.5).

Such a slight difference is very important for the cognitive adaptive system (the next chapter), where a probabilistic system will be able then to learn to consolidate such a UI Adaptation, see Chapter 5 for more details.

4.5 Discussion

We have discussed a modular design of Elementary Models obstacle detectors: Each obstacle detector focuses on a single modality and the detection of a single obstacle. In contrast to the alternative of training one monolithic classifier which integrates multiple modalities on a feature level

and differentiates multiple interaction obstacles at the same time in a multi-class model, this modular approach has the advantage of maximum flexibility: when new modalities become available or have to be removed (e.g., due to a broken sensor) or when a new obstacle needs to be treated, the set of detectors can be adjusted easily, just as plugins.

We see that on average, the proposed combinations (EEG-based detector for MEMOBS and the behavior-based detector for VISOBS) outperform the other variants, validating our expectations. We highlighted high discriminant EEG signals for detecting MEMOBS. The EEG-based MEMOBS detector achieves high accuracy of 76.13% and 82.95% for both the NOADAPT and VISADAPT conditions respectively. The weaker result of 71.59% for the MEMADAPT condition is plausible as that adaptation is designed to attenuate the effect of the memory obstacle, making the differentiation more difficult. The behavior-based detector does not respond in this fashion and is weaker in every condition. We see similar patterns for VISOBS detection, where the behavior-based model outperforms the EEG-based detector on average.

4.6 Conclusion

In this chapter, we discussed different HCI obstacles detectors from different data modalities. We named such detectors as ELEMENTARY MODELS, because they contribute but do not produce the final UI ADAPTATION decision, which will be made then by the upper decision layer in our cognitive adaptive system: Probability-based DBN (Chapter 5).

Namely, we discussed in this chapter: behavior-based LSTM MEMOBS detector, EEG-based SVM MEMOBS detector, behavior-based LSTM VISOBS detector and EEG-based SVM VISOBS detector. Each detector is discussed under different potential UI adaptation mechanisms: NOADAPT, MEMADAPT and VISADAPT.

As we aim at detecting different HCI obstacles, namely MEMOBS and VISOBS, different data modalities are best suited for different HCI obstacles; While the physiological EEG data naturally best suits MEMOBS compared to behavioral data, the VISOBS, in contrast, causes a confused behavior which is well depicted from behavioral data compared to less detection performance using EEG data.

In general, both modalities could be used for different kinds of interaction obstacles. However, to reduce the number of variables in the upper probability-based decision model, we will only choose the best elementary model for each

interaction obstacle: behavior-based LSTM for detecting visObs and EEG-based SVM for detecting memObs, see Chapter 5.

Chapter 5

Online Cognitive Adaptive System

This is the final chapter in the thesis pipeline, covering the online cognitive adaptive system. A probabilistic decision model will be discussed for deciding the UI adaptation. The model architecture will be discussed, and detailed evaluations will be presented to show how such a probabilistic model decides for the most probable UI adaptation after the first HCI session, and consolidates or reverts such a decision after each subsequent HCI session.

In this Chapter, we present our final system, so-called online cognitive adaptive system. We arrive hereby to the final phase in the thesis pipeline. That is, we introduced the HCI exemplary task, interaction obstacles and corresponding UI adaptations in Chapter 1, after that, we discussed all the data collections in Chapter 3, we used then such data for training and testing different HCI obstacles detectors so-called *Elementary Models* in Chapter 4, and now we want to put all together in realistic scenarios, in which different HCI obstacles should be "flexibly" treated in parallel through the proposed cognitive adaptive system. Flexible and extendable system is guaranteed by training binary HCI obstacle detectors and integrating them into the cognitive adaptive system architecture. This is in contrast to one monolithic model discriminating all obstacles at once, which lacks of flexibility as its multimodal feature vectors need to be adjusted and the model itself needs to be dropped and trained again with each new obstacle added to the system. For example, we can easily

extend the proposed probabilistic system for detecting attention distraction obstacle from eye tracking data: following the thesis pipeline, one should simulate that HCI obstacle to be detected (Chapter 1), collect HCI data from HCI sessions with and without that obstacle (Chapter 3), train binary obstacle detectors to detect obstacles from each data modality collected (Chapter 4), and finally, flexibly integrate those trained binary HCI obstacle detectors into the overall probabilistic cognitive adaptive system discussed within this Chapter.

This chapter is structured as follows: first, we discuss related works to our proposed cognitive adaptive system in Section 5.1. In Section 5.2, we briefly recall the architecture of our system, which has been introduced in Chapter 1, and also the underlying *elementary models*, which have been discussed in Chapter 4. In Section 5.3, we introduce our Bayesian modeling approach for realizing the proposed cognitive adaptive system. Finally, we evaluate and discuss in Section 5.4 the proposed cognitive adaptive system for HCI sessions with volatile and persistent obstacles.

5.1 Related Work

In this section, we discuss related works in terms of increasing the usability of computer systems. Concretely, we discuss both customization and adaptation techniques used to tackle the visual and memory-based obstacles.

In general, the HCI community aims at improving systems usability and user acceptance. Two early works from Seffah and Metzker (2004) as well as Ferre et al. (2006) discussed difficulties in integrating Software Engineering (SE) and Human Computer Interaction (HCI) principles and best practices. While both highlighted the importance and difficulty of designing a usability-conformed UI during the SE life cycle, the authors of Ferre et al. (2006) also discussed integration proposals between SE and HCI best practices to benefit from HCI principles while developing computer systems. However, the discussed proposals belong to a static software, i.e. without UI adaptation.

Customization and adaptation techniques have been contiguously evolving for improving the HCI performance. While customization grants users control to customize the UI according to their specific needs, adaptation, in contrast, grants control to the adaptive system itself which automatically identifies users to adapt the UI according to user needs.
For customization techniques, and related to the visual obstacle discussed in this thesis, there exist works which introduced a customized UI for potential

users with color vision deficiency, see Jefferson and Harvey (2007) and Qaiser and Khan (2017). According to Jefferson and Harvey (2007), users can customize the colors of any region of the UI. However, there is no detection of the visual obstacle (color vision deficiency) in their interface. In contrast, Qaiser and Khan (2017) discussed the detection of such an obstacle by utilizing a typical color vision test automated in a mobile app. After the test, the most appropriate color scheme for this specific user is chosen. The limitation of such an UI is that the specific color test cannot be smoothly integrated in HCI applications. This limitation can be tackled by detecting such an obstacle from user behaviour. In Salous et al. (2019), we introduced a visObs detector from user behavior, and we introduced an UI adaptation visAdapt by utilizing voice instructions to compensate visual obstacles. Sarcar et al. (2018) introduced the concept of "ability-based optimization", where they use a cognitive model for evaluation of system variants in an optimization process, which accounts for customized UIs for users with sensorimotor and cognitive impairments.

For adaptation techniques, and related to the memory-based obstacle discussed in this thesis, Sguerra and Jouvelot (2019) investigated an adaptation method based on real-time tracking of human working memory. Sguerra et al. modeled the human memory based on the Moran process Moran (1958) by maintaining "quanta" numbers (weights) for each stored item. As an HCI task, they used a matching pairs game, tracked the user performance and released an adaptation signal when the performance deteriorates to a value less than a given application-based parameter (no explicit external secondary task obstacle). Thus, the approach by Sguerra and Jouvelot is an explicit performance-based tracking model, which does not consider potential temporal dependencies in behavioral data (user actions) to detect behavioural changes when applying that UI adaptation while an explicit secondary task is present. Modeling of temporal dependencies along user actions (behavioral data) is a very important gap in the HCI literature. Such temporal dependencies reflect the actual user performance, competencies and needs. Thus, appropriate modeling of behavioral data, during consecutive HCI sessions, enables an appropriate automatic UI adaptation. To tackle this gap in HCI contexts, we introduced a sequential model based on Recurrent Neural Networks in Salous et al. (2019), that shows advantages in detecting behaviour changes, especially in the presence of memory-based obstacle with UI adaptation.

As a dependable adaptive system, Gajos et al. introduced the SUPPLE Gajos and Weld (2004) as an optimization-based automatic adaptive system. It optimizes the UI adaptation task according to the device's constraints (e.g., cell phones, touch panels, etc.). It aims at minimizing the estimated effort

for the user's expected actions within an HCI task. While this became a widely used approach (e.g. Lam and Baudisch (2005); Gajos et al. (2007, 2008); Halko and Kientz (2010); Gajos et al. (2010)), it only targets user experience according to different devices but it does not consider human cognitive processes during HCI tasks.

One shortcoming of the state-of-the-art research is, that it usually stops at the initial decision and often only covers a single type of obstacle. The fact that every interaction obstacle requires a specific adaptation, together with the observation that all data-driven methods for automatic adaptation can also yield wrong results, means that the detection process needs to continue after an initial adaptation was triggered. However, it is known that the effects of different cognitive processes interact with each other and thus we should also expect an impact of an active UI adaptation on the detection of interaction obstacles.

5.2 Cognitive Adaptive Model

We briefly recall in Section 5.2.1 the architecture of our proposed cognitive adaptive system, which has been introduced in Chapter 1. In that architecture, we also briefly recall in Section 5.2.2 the underlying *elementary models*, which have been discussed in details in Chapter 4.

5.2.1 Architecture

First, we recall the general architecture introduced in Chapter 1, see Figure 5.1. During the HCI task (matching pairs game, depicted as "1" in Figure 5.1), we observe behavioural data (encoded revealed cards, depicted as "2" in Figure 5.1) and neural data (EEG, depicted as "3" in Figure 5.1). Then, such multimodal data is passed to multiple obstacles detectors, so-called *elementary models* (in our case, MEMOBS and VISOBS multimodal detectors, we briefly recall them in Section 5.2.2). Each elementary model (obstacle detector) has the task to detect the presence or absence of one specific obstacle. On top of the individual detectors, we use a Bayesian fusion component, which generates the final decision of choosing an UI adaptation as follows: the outputs of the obstacle detectors (either obstacle or NOOBS) will be passed together with the corresponding confidence scores as inputs to the Bayesian fusion component. We will discuss this Bayesian fusion component in details in this chapter, follow Section 5.3 and see Figure 5.2. The UI adaptation for the next interaction is then decided based on the underlying Bayesian model: NOADAPT, Adapt.1 (MEMADAPT) or Adapt.2 (VISADAPT). When

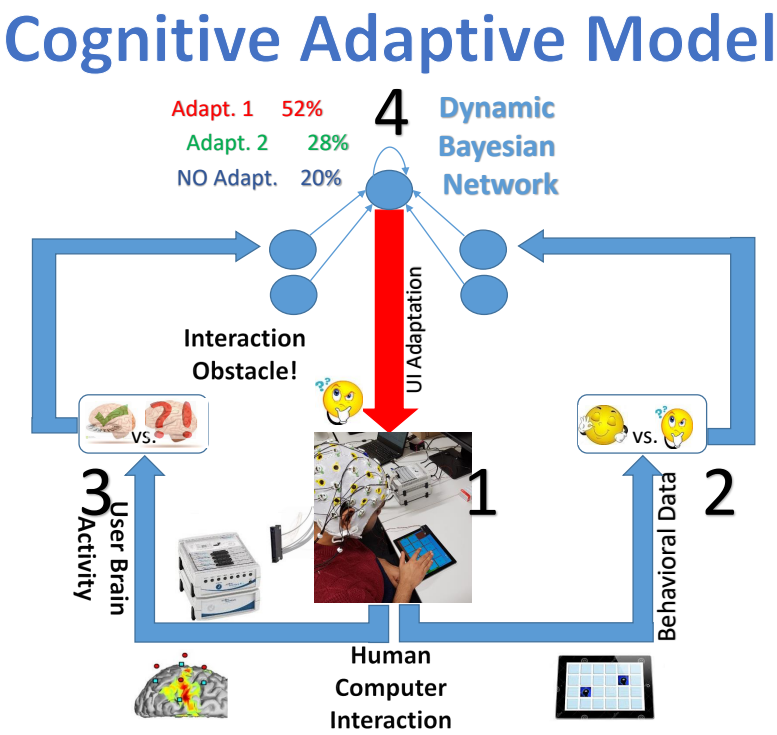


Figure 5.1 – Cognitive adaptive system general architecture. Binary obstacles detectors detect the presence of memObs and visObs from EEG (part 3) and behavioral (part 2) data modalities. On top of such models, Dynamic Bayesian Network learns the most probable UI Adaptation, e.g. Adapt.1 in this illustration Figure. Adapt.1 and Adapt.2 are illustrated as examples of UI adaptation mechanisms, e.g. MEMADAPT and VISADAPT.

the decided UI adaptation is applied in the next interaction with the same user (i.e., playing a next game), the same aforementioned steps (observe data, pass it to the detectors, and pass predictions and confidences from detectors to the Bayesian fusion model) are repeated to decide on the UI adaptation again and thus either to consolidate the result or to change erroneous UI adaptation. We decided for this architecture (in contrast to a monolithic model discriminating all obstacles at once) as it allows us to flexible add or remove different obstacles and easily consider subsequent multiple interaction.

Concretely, as a probability-based framework, we use a Dynamic Bayesian Network (DBN) which decides on the most probable obstacle and UI adaptation using inputs from the several underlying detectors over multiple time steps, follow more details in Section 5.3.

5.2.2 Elementary Models

An obstacle detector is a system component that represents a binary classifier to detect the presence or absence of a specific obstacle in a (segment of an) interaction session from a specific modality. We name such obstacle detectors as ELEMENTARY MODELS, because their outputs (predictions and confidences) will be passed to the subsequent probability-based decision model and thus they contribute to but do not produce the final UI ADAPTATION decision.

Each obstacle detector focuses on a single modality and the detection of a single obstacle under a specific UI adaptation. In Chapter 4, we discussed and evaluated obstacle detectors for all the combinations between data modalities, obstacle types and UI mechanisms. In contrast to the alternative of training one monolithic classifier which integrates multiple modalities on a feature level and differentiates multiple interaction obstacles at the same time in a multi-class model, this modular approach has the advantage of maximum flexibility: when new modalities become available or have to be removed (e.g., due to a broken sensor) or when a new obstacle needs to be treated, the set of detectors can be adjusted easily just as plugins, see the depicted parts "2" and "3" in Figure 5.1.

5.3 Recurrent Bayesian modeling of Subsequent Adaptive Sessions

In this section, we discuss our recurrent modeling of subsequent adaptive sessions in the matching pairs HCI. First, we present requirements for a recurrent decision model. Then, we introduce and argue our recurrent model choice comparing to other recurrent models.

Recurrent Bayesian Modeling Requirements: Behavioral and neural data give two complementary perspectives on the presence or absence of interaction obstacles. Naturally, we are interested in the combination of both modalities; however, one challenge is the very different nature of both models, using different machine learning algorithms and different time scales and sampling rates. Moreover, to ensure simplicity, flexibility and effectiveness (details in Section 5.3.2), each interaction obstacle detector is trained, from one data modality, to detect the presence or absence of one interaction obstacle and is not designed to differentiate between multiple obstacles.
An automatic decision for a UI adaptation may not be optimal from the first session, and the HCI consecutive sessions are not isolated from each other.

Therefore, we need to model potential dependencies between consecutive adaptive sessions to follow the dynamic nature of HCI sessions for an improved UI adaptation decision. Thus, we need a recurrent modeling to model temporal dependencies between consecutive sessions, and we need a Bayesian modeling to decide on the most probable UI adaptation mechanism.

Recurrent Bayesian Modeling Choice: We followed literature reviews which discuss Bayesian modeling (Hidden Markov Model HMM and Bayesian Network BN, Ghahramani (2001)) and recurrent dynamic modeling (HMM and Dynamic Bayesian Network DBN, Murphy et al. (2002)). According to Murphy et al. (2002), HMMs are a special case of DBNs in which the modeled state is represented by a single hidden state random variable. The authors reviewed many variants of HMM, e.g. Hierarchical HMM (HHMM) which is especially powerful for speech recognition applications. The authors concluded that HMMs are somewhat inflexible models of sequential data, which can be recurrently and efficiently modeled by DBNs. The authors recommended DBNs for recurrent Bayesian modeling of different sequential data applications, as DBNs are easy to interpret and learn: graph is directed, thus the conditional probability distribution (CPD) of each node can be estimated independently.
We employ a Bayesian late fusion scheme which takes the classification result and a confidence estimate of each individual interaction obstacle detector as input. From the output of the Bayesian model, we derive the decision for the adaptation mechanism which is applied in the next interaction session. Beyond the fusion of modalities for a single session, the task of the Bayesian model is also the integration over multiple interaction sessions. This is accomplished through the use of a DBN, which takes inputs from previous time steps, including the adaptation decision. The underlying model architecture is shown in Figure 5.2.

For implementation, we extended the Bayesian Network (BN) in PGMPY (Ankan and Panda (2015)) to create a Dynamic Bayesian Network (DBN) which learns Conditional Probabilistic Distributions (CPDs) directly from data (predictions and confidences) coming from the underlying obstacle detectors in consecutive interactions of the same user, follow more details about the DBN and the learned CPDs in Section 5.3.1.

5.3.1 Dynamic Bayesian Network

Figure 5.2 shows two loops in our recurrent Bayesian architecture: 1) the decided adaptation mechanism is fed back to its own node to benefit from such an information when re-deciding the UI adaptation (see the self loop

depicted as "1"), and 2) the decided adaptation mechanism is also passed back to the elementary models (MEMOBS detector depicted as "2a" and VISOBS detector depicted as "2b") for the next interaction. For more details, see Figure 5.3 and follow the next section (Section 5.3.2) to see how such a recurrent architecture allows an effective use of appropriate obstacle detectors in subsequent interactions.

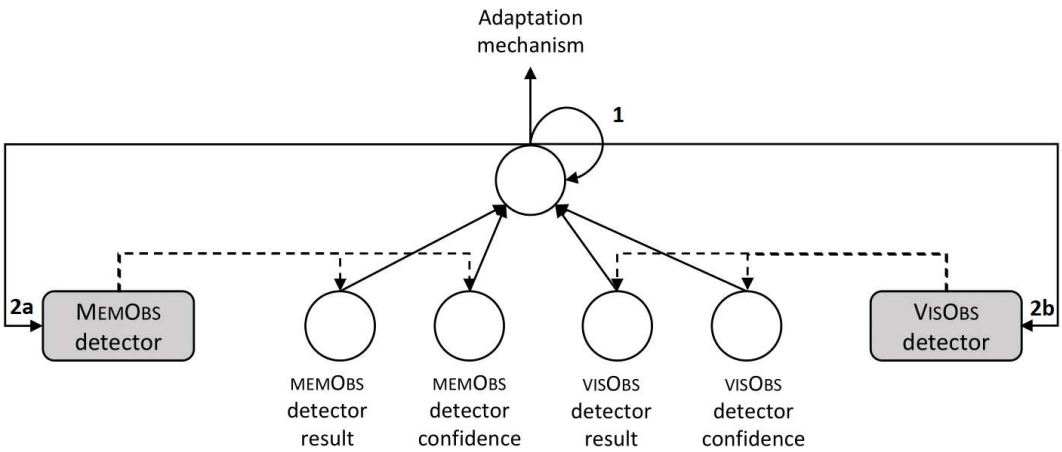


Figure 5.2 – Architecture of the Dynamic Bayesian Network for the derivation of adaptation mechanisms suited for the detected interaction obstacle.

Such a recurrent architecture allows modeling of temporal dependencies between consecutive interaction sessions, however, a traditional Bayesian network is a Directed Acyclic Graph (DAG) which prevents using such loops for modeling Conditional Probability Distributions CPDs, Stephenson (2000). Therefore, we extended the BN implementation in PGMPY (Ankan and Panda (2015)) by allowing modeling such loops, simply via unrolling them into new "dynamically" instantiated nodes for the next interaction session. Thus, while a BN only models Conditional Probabilistic Distributions (CPDs) in a Directed Acyclic Graph (DAG), our extended model, DBN, dynamically models potential temporal dependencies between consecutive interactions.

The designed DBN will learn then distributions of data coming from the underlying elementary models under different conditions, i.e. different UI adaptation mechanisms. Thus, given predictions and confidences from the underlying obstacle detectors (recall Chapter 4 for details of estimated confidences), DBN learns correlations between such data distribution and the decided UI adaptation mechanism. Follow the next section for an illustration and details of different distributions learned by the DBN under different UI

mechanisms, and how such distributions of data are fed to DBN in consecutive interaction sessions.

5.3.2 Flexibility and Effectiveness of DBN

In this section, we explain the flexibility and effectiveness of our proposed DBN: flexibility in terms of extending or removing an underlying elementary model, and effectiveness in terms of automatic selection of appropriate elementary models in each tested session, and thus, learning of several appropriate distributions of predictions and confidences from each set of elementary models regarding to the applied UI adaptation mechanism.

First, we recall the independence restriction which we mentioned in the previous chapter. We focus in this thesis on the presence of one obstacle in an HCI session (game prefix of 10 rounds, which lasts relatively short). If we want to consider multiple HCI obstacles at the same HCI session, then we have also to consider potential interaction dependencies between such obstacles and the designed UI adaptation mechanisms. We should introduce new UI adaptation mechanisms for such cases. Actually, modelling such interactions between obstacles should consider that multiple cognitive processes can be affected. For example, if the MEMOBS and VISOBS present simultaneously in an HCI session, both the cognition and perception processes are simultaneously affected. Thus, the discussion of DBN's flexibility and effectiveness is restricted to our independence assumption: single obstacle at an HCI session.

For flexibility, the introduced design in Figure 5.2 allows a flexible addition of more models for new different obstacles. That is, similar to the EEG-based MEMOBS detector and behaviour-based VISOBS detector, new binary detectors for new obstacles can be trained and added to the overall cognitive adaptive model depicted in Figure 5.2, e.g. attention distraction obstacle from eye-tracking data.

For effectiveness, note that the selected adaptation in a subsequent interaction is known as it was selected by the system in a previous iteration. Therefore, our system can always select the appropriate obstacle detectors trained on data of the corresponding adaptation mechanism in subsequent interactions, see Figure 5.3. Recall the adaptation mechanisms (NOADAPT, MEMADAPT, VISADAPT) in Chapter 1, where both MEMADAPT and VISADAPT apply different changes to the UI to tackle the drop of user performance caused by the detected obstacle. Briefly, the MEMADAPT stimulates the WM by repetition of information according to primacy and recency features, while

the visAdapt complements the impaired visual memory by facilitating the (unimpaired) auditory memory via auditory stimuli (spoken identifiers for the unrecognized cards). When the decided UI adaptation is applied in the next interaction, new behavioural data and EEG data will be recorded from the user under this new settings (UI adaptation mechanism), and thus, the appropriate obstacle detectors will be selected accordingly to detect potential obstacles in the next interaction. Consequently, DBN learns distributions (concretely, conditional probability distributions (CPDs)) of each data set coming from different settings of elementary models: noAdapt obstacle detectors, memAdapt obstacle detectors and visAdapt obstacle detectors, and accordingly decides the most probable UI adaptation mechanism for a given test data episode (predictions and confidences from elementary models for the next tested interaction session). For example: if the UI adaptation

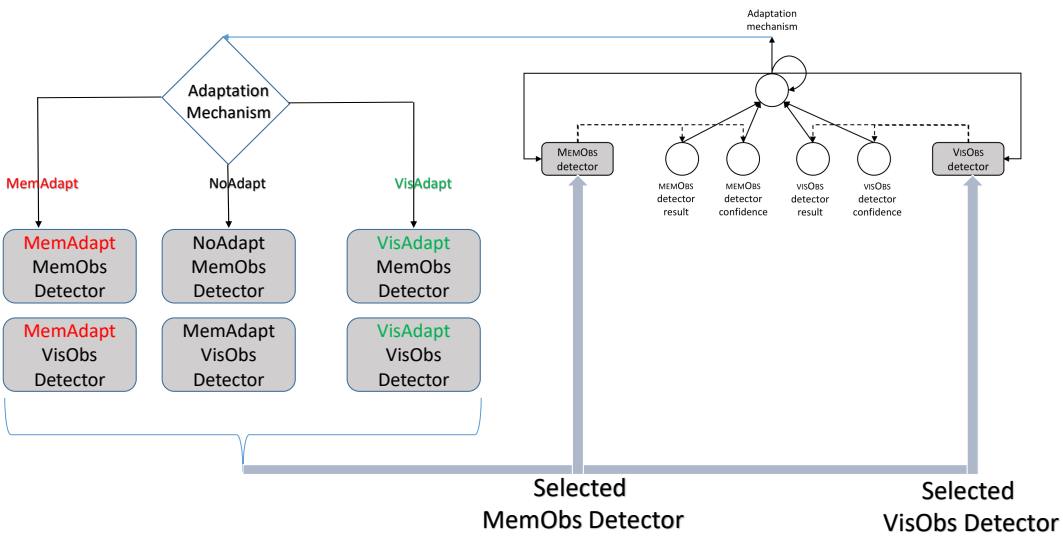


Figure 5.3 – Selection of appropriate elementary models (obstacle detectors) in a subsequent interaction session based on the selected UI adaptation mechanism in the previous interaction session.

decided in the first interaction was memAdapt, then the DBN selects the following models for the next interaction: memObs detector given memAdapt (noObs_memAdapt vs. memObs_memAdapt) and visObs detector given memAdapt (noObs_memAdapt vs. visObs_memAdapt), see Figure 5.3. We also point out that while we only maintain an EEG-based model for the detection of memObs and a behavior-based detector of visObs, the DBN always regards both modalities jointly as it decides between adaptations for both obstacles.

5.4 Evaluation

In this section, we evaluate the proposed DBN model for the detection of interaction obstacles and selection of best UI ADAPTATION. We evaluate the DBN results in consecutive HCI sessions to investigate the effect of temporal integration. That is, the proposed DBN should learn potential temporal dependencies between consecutive sessions, e.g. persistent obstacle remains through consecutive sessions or volatile obstacle suddenly appeared or disappeared. Consequently, the DBN should either keep or accordingly change the applied UI adaptation: NOADAPT, MEMADAPT or VISADAPT.

Thus, to evaluate all possible cases, we evaluate with HCI sessions in which an interaction obstacle remains in consecutive sessions (persistent), and also we evaluate with HCI sessions in which an interaction obstacle suddenly appears or disappears (volatile). Finally, we look at the impact of adaptation mechanisms on the performance in the presence of different interaction obstacles, to study the importance of correct differentiation.

5.4.1 DBN Evaluation Through Consecutive Interactions

We evaluate the DBN according to its ability of deciding the correct UI adaptation: NOADAPT, MEMADAPT or VISADAPT, thus, the baseline accuracy is 33.33%. We evaluate the DBN for consecutive interactions; that is, after the first interaction is finished (random matching pairs game variant: NOOBS, MEMOBS or VISOBS), an UI adaptation is made by the DBN (NOADAPT, MEMADAPT or VISADAPT) and can be evaluated. Similarly, after that UI adaptation decision is applied, subsequent interactions can be supported by the DBN with an UI Adaptation decision (either to keep the same UI adaptation or to change). Those decisions for the subsequent interactions can be also evaluated to show how well does the DBN learn from more information available through consecutive interactions.

For this purpose, we generate sequences of interaction sessions of the same person with persistent or volatile interaction obstacle, thus, the proposed DBN model will be changing the adaptation mechanism, and we can evaluate accordingly. This simulates the system applying the adaptation mechanism considered optimal based on the data of one session, while maintaining the ability to re-evaluate the decision once additional data becomes available.

5.4.1.1 HCI Sessions with Persistent and Volatile Obstacles

In this section, we explain the data which we prepared for training and evaluating the DBN as a cognitive adaptive model. We prepared the DBN data as multiple episodes, each consists of two consecutive interactions, to cover all the possible cases of persistent and volatile HCI obstacles, which will be explained in Tables 5.1 and 5.2.

First, we briefly recap the UI adaptation mechanisms which have been introduced in Chapter 1. Second, we discuss how our recurrent architecture ensures modeling of long sequences of multiple interactions by modeling episodes of only two consecutive interactions. Third, we briefly recap the iteration process of DBN, which is applied after each interaction. Finally, we discuss the combinations of consecutive interactions, with which the DBN is fed to learn to adapt according to persistent and volatile obstacles.

UI Adaptation Mechanisms: We introduced three UI adaptation mechnisms in Chapter 1. Briefly:

1. noAdapt applies no changes to the UI, and should be applied when no obstacle is detected.

2. memAdapt should be applied to tackle a detected memObs. It stimulates the impaired WM by repetition of information according to primacy and recency features. Concretely, it re-reveals last revealed cards after a non-matching turn.

3. visAdapt should be applied to tackle a detected visObs. It complements the impaired visual memory by facilitating the (unimpaired) auditory memory via auditory stimuli (spoken identifiers for the unrecognized cards).

Two-Interactions Based Modeling: The recurrent architecture shown in Figure 5.2 allows to model dependencies between consecutive HCI sessions. The loops "1" and "2" ("2a" and "2b") in Figure 5.2 pass the decided UI adaptation to the next HCI session. While this architecture explicitly models dependencies between two consecutive HCI sessions, it implicitly models dependencies between multiple consecutive sessions: The UI adaptation decision in the first session is passed to the second session, then, the new UI adaptation decision in the second session (consolidate or revert the first UI adaptation) is passed to the third session, and so on.

Consequently, we train the DBN with episodes of only two consecutive sessions, so-called 1st Interaction and 2nd Interaction in Tables 5.1 and 5.2. These

1st and 2nd interactions can cover potential long sequences: the first session with the second session, the second session with the third session, the third session with the fourth session, etc.

DBN Iteration Process: For each interaction, the DBN is fed with the predictions and confidences coming from the corresponding ELEMENTARY MODELS (obstacles detectors), e.g. for MEMOBS_MEMADAPT HCI interaction, the DBN is fed with predictions and confidences from multimodal MEMOBS_MEMADAPT obstacle detectors, concretely, from EEG-based SVM MEMOBS_MEMADAPT obstacle detector and behaviour-based LSTM MEMOBS_MEMADAPT obstacle detector.

Data for Persistent and Volatile Obstacles: The DBN needs to learn the dynamic nature of obstacles: persisten or volatile. In the two-interactions based modeling mentioned above, if an obstacle is detected in an HCI session, and then re-detected in the next HCI session, it is supposed as persistent obstacle (within these two consecutive sessions). Otherwise, if an obstacle detected in the first session but not detected in the second session, the obstacle is supposed to be volatile. In either obstacle case (persistent or volatile), the DBN should learn how to adapt accordingly, i.e. we have to prepare episodes of two consecutive interactions which cover all possible cases (wrong and correct decisions in the first session combined with persistent or volatile types). We present and discuss the persistent cases in Table 5.1 and the volatile cases in table 5.2.

Table 5.1 – DBN data description for persistent HCI obstacles. Consecutive sessions which cover all the cases of persistent HCI obstacles which remain constant through two tested consecutive sessions: if the decided UI adaptation mechanism in the 1st Interaction is correct (the first 3 lines:1-3), consolidate it in the 2nd Interaction because the obstacle type is persistent in these two sessions (including NOOBS). Otherwise, revert it (lines from 4 to 9).

#	1st Interaction	2nd Interaction	Description
1	NOOBS_NOADAPT	NOOBS_NOADAPT	Consolidate correct
2	MEMOBS_MEMADAPT	MEMOBS_MEMADAPT	UI ADAPT for
3	VISOBS_VISADAPT	VISOBS_VISADAPT	persistent obstacles
4	MEMOBS_NOADAPT	MEMOBS_MEMADAPT	Revert NOADAPT
5	VISOBS_NOADAPT	VISOBS_VISADAPT	for persistent obstacles
6	NOOBS_MEMADAPT	NOOBS_NOADAPT	Revert MEMADAPT
7	VISOBS_MEMADAPT	VISOBS_VISADAPT	for persistent obstacles
8	NOOBS_VISADAPT	NOOBS_NOADAPT	Revert VISADAPT
9	MEMOBS_VISADAPT	MEMOBS_MEMADAPT	for persistent obstacles

Table 5.1 shows 9 combinations which cover all the cases of persistent HCI obstacles which remain constant through consecutive sessions. That is, each episode in Table 5.1 (line) consists of two consecutive interactions (1st and 2nd Interactions columns), for which the DBN has to either consolidate or revert the UI adaptation mechanism decided in the 1st interaction for the next interaction (2nd interaction): Given that the obstacle condition is persistent (including noObs), DBN should learn to consolidate the correctly decided UI adaptation mechanism (lines 1-3) and to revert the wrongly decided ones (lines 4-9). For example, DBN learns to consolidate the noAdapt when 1st interaction with noObs was correctly decided for noAdapt (line 1), while DBN should learn to revert that noAdapt to memAdapt when the persistent obstacle is memObs while noAdapt was wrongly decided in the 1st interaction (line 4). Note that the 1st interaction and 2nd interaction data can be: the first session with the second session, and also the second session with the third session, etc. Thus, this data can cover long sequences of HCI sessions, consequently, the DBN learns temporal dependencies between multiple HCI sessions accordingly.

Table 5.2 – DBN data description for volatile obstacles. Consecutive sessions which cover all the cases in which volatile HCI obstacles suddenly appear or disappear in one of two consecutive tested sessions.

#	1st Interaction	2nd Interaction	Description
1	memObs_noAdapt memObs_memAdapt memObs_visAdapt visObs_noAdapt visObs_memAdapt visObs_visAdapt	noObs_noAdapt	Volatile obstacle disappeared in 2nd Interaction! Back to noAdapt in 2nd Interaction, regardless of correct or wrong decisions in 1st Interaction.
2	noObs_noAdapt noObs_memAdapt noObs_visAdapt	memObs_memAdapt visObs_visAdapt	Volatile obstacle appears in 2nd Interaction! Apply appropriate UI adaptation mechanism, regardless of correct or wrong decisions in 1st Interaction.

We showed all the different cases in which the obstacle is always persistent between consecutive interactions in Table 5.1.

In contrast, we show in Table 5.2 the other cases for which the interaction obstacle suddenly appears or disappears in the tested consecutive interactions (volatile obstacles). It is important to train the DBN, and investigate its ability, to handle both obstacle types (persistent and volatile), because both are realistic in real-world scenarios. Table 5.2 contains all the possible cases of

such volatile obstacles; MEMOBS or VISOBS suddenly appears or disappears in the tested consecutive interactions, thus, DBN should learn, from such data, to go back to the NOADAPT when an obstacle appeared in the 1st tested interaction and then disappeared in the subsequent interaction, regardless of correct or wrong decisions in 1st Interaction (block #1). In contrast, block #2 in Table 5.2 summarizes the cases from which the DBN learns to adapt the UI according to a volatile obstacle (MEMOBS or VISOBS) suddenly appeared in the 2nd interaction, again, regardless of correct decision (NOADAPT for NOOBS) or wrong decisions (MEMADAPT or VISADAPT for NOOBS) in the 1st interaction.
In other words, while the DBN learns typically to keep correct UI adaptation decisions (probably high confidences from ELEMENTARY MODELS) and to revert wrong ones for persistent obstacles discussed in Table 5.1, probably by learning such high and low confidences distributions which are fed to DBN, it should learn from Table 5.2, however, temporal dependencies between 1st and 2nd interactions data to detect such volatile obstacles and to decide the UI adaptation mechanism accordingly. Again, the 1st interaction and 2nd interaction data can be: the first session with the second session, and also the second session with the third session, etc. Thus, this data can cover long sequences of HCI sessions, consequently, the DBN learns temporal dependencies between multiple HCI sessions accordingly.

5.4.1.2 DBN Performance Discussion

In Section 5.4.1.1, we have explained the data structure of our proposed DBN model, concretely, Table 5.1 for persistent obstacles data and Table 5.2 for volatile obstacles data. In this section, we evaluate the DBN performance through consecutive interactions depicted as "1st Interaction" and "2nd Interaction" in Tables 5.1 and 5.2.
First, we summarize the DBN data preparation, which comes from *elementary models* outputs, for training and evaluating DBN. Then, we show and discuss confusion matrices evaluating consecutive interactions of persistent obstacles (Table 5.1) and volatile obstacles (Table 5.2).

DBN Data Preparation:
we have seen the data structure as consecutive interactions for persistent obstacles (Table 5.1) and volatile obstacles (Table 5.2). Now, we summarize how such data is prepared for DBN. As mentioned above, the DBN data comes from the underlying *elementary models* (multimodal MEMOBS and VISOBS detectors, Chapter 4). Given an HCI interaction (first 10 turns in a game), multimodal data (behavioral and EEG) is passed to their corresponding obsta-

cle detectors, which in turns produce two outputs: prediction and confidence estimation. Thus, four inputs are passed to pre-defined corresponding DBN nodes: EEG-based MEMOBS prediction, EEG-based MEMOBS confidence, behaviour-based VISOBS prediction and behaviour-based VISOBS confidence, see these four nodes in Figure 5.2 (recall Chapter 3 and Chapter 4, where we decided and argued to use EEG for MEMOBS detectors and behavioral data for VISOBS detectors).

Consecutive Interactions Evaluation:
We train and evaluate the DBN in consecutive interactions by using the leave-one-out cross validation method. Figure 5.4 shows that the DBN outperforms the baseline in the first interaction (confusion matrix A) with 72.5% as an average accuracy[1]. Furthermore, it shows that the DBN learns through the next interaction (confusion matrix B) and drastically improves, with an almost perfect average accuracy: 99.7%. For more concrete evaluation, we look at confusion matrices for first and second interactions, moreover, in the second interaction, we look also at a confusion matrix for both persistent obstacles (confusion matrix C) and volatile obstacles (confusion matrix D) discussed in Tables 5.1 and 5.2, see Figure 5.4.

The confusion matrices of first and second interaction in Figure 5.4 (A and B) show that the prediction accuracy of all labels improves when re-deciding the UI adaptation in the second interaction. That is, the accuracy of all the predicted labels in the first interaction (confusion matrix A) improves from 82.4%, 82.4%, and 52.9% to 100%, 100%, and 99.02% for NOADAPT, MEMADAPT and VISADAPT respectively. We observe the greatest improvement for the VISADAPT case (a relative improvement of 87.04%) as the model is able to recover many cases in which the original VISOBS was not detected. This is possible because the VISADAPT class does not exhibit false positives in the first interaction and evidence can accumulate.

Persistent and Volatile Obstacles Recovery:
Indeed, the confusion matrix (C) in Figure 5.4 shows that in 100% of all persistent obstacle cases, wrong decisions in the first interaction could be corrected in the second one. We can explain this improvement by the temporal dependencies which the DBN has learned between different cases of 1st and 2nd interactions in Table 5.1. In addition, given a persistent obstacle in two consecutive sessions (including NOOBS), it is expected to easier recover (i.e.

[1]note that the confusion matrices are now labeled with the UI adaptations, compared to the labeling with interaction obstacles which is done for the obstacle detectors, as the adaptations are the output of the DBN.

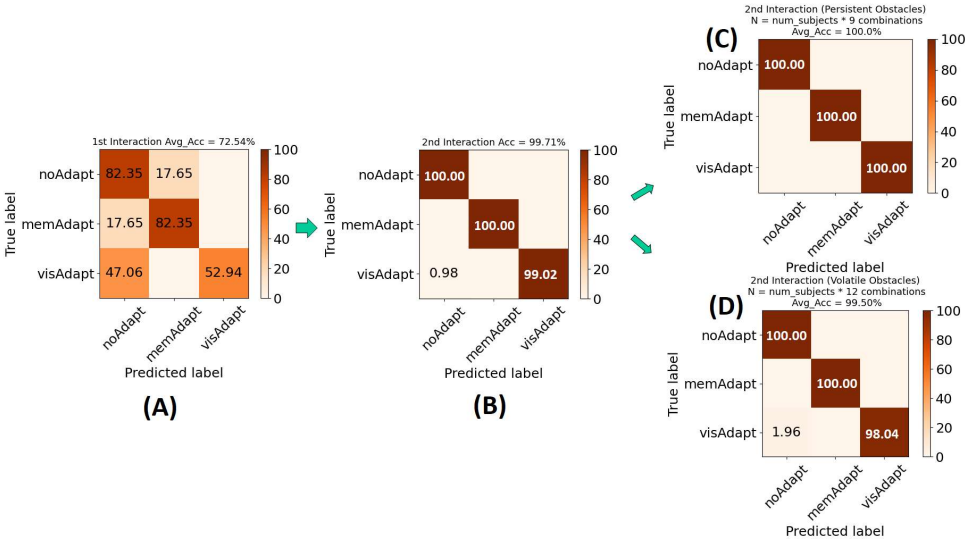


Figure 5.4 – Left (A): DBN first interaction evaluation. Middle (B): DBN second interaction evaluation overall persistent and volatile obstacles. Right (C and D): detailed evaluation for the second interaction given only persistent obstacles (C, upper matrix) and only volatile obstacles (D, lower matrix) through 1st and 2nd interactions.

to correct wrong decisions) than in a volatile obstacle case which is more dynamic.

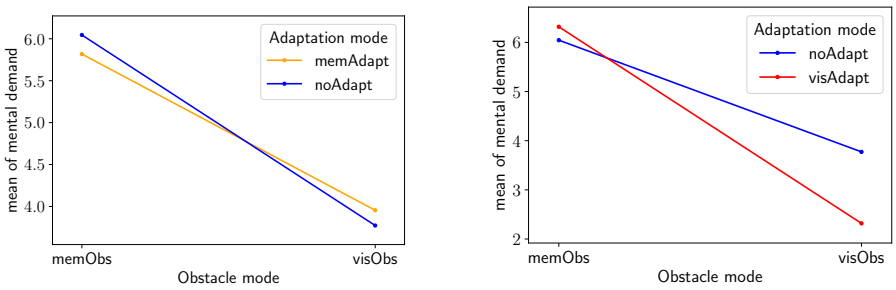
Note that not all combinations appear in the evaluation, for example there are no confusions between MEMADAPT and VISADAPT in the first interaction and thus, the corresponding combinations are not evaluated in the second iteration. Still, these cases were trained in the model and the results in Chapter 4 for different obstacle_adaptation combinations make it likely that detection performance will be on similar levels. the confusion matrix (D) in Figure 5.4 shows that even for volatile obstacles, the proposed DBN can in the 2nd interaction perfectly (avg_acc=100%) recover all the NOADAPT and MEMADAPT decisions, and most of the VISADAPT ones (avg_acc=98.04%).

Results Discussion:
If we recall the subjective and objective analysis in Chapter 3 and the predictive results in Chapter 4, we can briefly argue these DBN results as follows: subjective analysis in Chapter 3 shows that participants felt very comfortable with VISADAPT (auditory instructions) to tackle the VISOBS. Objective analysis (performance measurements) coincided with their feelings and showed that the VISOBS has been completely tackled by the VISADAPT, where different static and sequential measurements showed very close performance between

visObs_visAdapt and noObs. Elementary models evaluations in Chapter 4 went in the same direction, where the lowest accuracy has been shown for visObs_visAdapt detectors because their data labels (visObs_visAdapt and noObs) have very close behaviors. This was not the case for memAdapt and memObs, where memObs is naturally stronger than visObs, and thus, memAdapt improves the performance, but does not completely drop the obstacle effects. Thus, while DBN effectively learns data distributions (predictions and confidences) from the different Elementary Models, and even effectively learns dynamic temporal dependencies between consecutive interactions, if an error, in UI_Adapt decision, could happen, then it will be likely expected in that most confusing case: visObs_visAdapt.

5.4.2 Evaluation of Adaptive Mechanisms

Finally, we look at the impact of different adaptation mechanisms (memAdapt and visAdapt) on the usability during a session. The main purpose of the detection and discrimination of several interaction obstacles is that we expect different adaptations to be optimal for different types of obstacles, i.e., there is no "silver bullet" adaptation (which could be activated by default). To investigate the validity of this claim and the effectiveness of the adaptation, we perform a two-way ANOVA with the obstacle mode and adaptation mode as independent variables and the self-reported mental demand as dependent variable.



(a) memAdapt (b) visAdapt

Figure 5.5 – Interaction between adaptation mechanisms and interaction obstacles for experienced mental demand.

We see significant main effects of both obstacle mode (as expected, because especially the memory obstacle has a strong impact on the mental demand) with $F = 131.49$, $p = 2.7e^{-21}$, and $\eta = 0.47$ as well as for adaptation mode (which shows that adaptation is effective) with $F = 2.67$, $p = 0.007$, and

$\eta = 0.02$. Besides, we observe an interaction effect ($F = 7.67$, $p = 7.2e^{-4}$, and $\eta = 0.05$), which shows that indeed MEMADAPT is helpful to support the user in case of a MEMOBS (i.e., it reduces mental load), but is detrimental otherwise. The same holds for VISADAPT. Figure 5.5 illustrates the interaction effect.

5.5 Discussion & Conclusion

The results show that by combining behavior-based and EEG-based obstacle detectors, we are able to differentiate between multiple interaction obstacles, even in the presence of different active adaptation mechanisms. This finding enables the use of information across multiple interaction sessions to validate or correct an initial adaptation decision. This integration between consecutive sessions drastically improves classification accuracy; in a second interaction session, the system was able to predict the prevalent interaction obstacle (if any) with near-perfect accuracy, either with persistent obstacle through consecutive interactions, or also with volatile obstacle which suddenly appears or disappears in one of those consecutive interactions. This allows the system to either confirm or correct any adaptation mechanisms triggered after the initial session.

It remains to discuss the implications of these results on general HCI systems. While we investigated the detection of interaction obstacles in a specific setting, the chosen memory game seems to be a good fit to represent many typical HCI tasks: It involves visual perception of the UI, working memory retrieval and encoding (of both spatial and symbolic information), as well as planning and decision making, which all occur in many other HCI tasks. Similarly, the investigated interaction obstacles touch generic impediments to perception and cognition encountered in many different contexts. Therefore, we assume that the employed models are relatively generic and allow a transfer to different tasks or conditions: The EEG-based MEMOBS detector depends on a trigger for specific system events to segment windows, but this can be easily obtained from most applications. The behavior-based VISOBS detector relies more on specific characteristics of the task and will need to be retrained for a different target application. However, as it is based on logged user interaction, the necessary training data is easy to acquire in the background of a newly enrolled application.

The investigated obstacles were chosen to cover both persistent and volatile obstacles as well as obstacles resulting from impairment of perception and internal cognition. Nevertheless, future work needs to further tackle the

challenge of transferring the developed models to users experiencing real interaction obstacles.

Chapter 6

Summary and Outlook

In this chapter, we summarize the thesis, highlight the contributions, visualize the multifaceted results (subjective, objective, predictive), mention limitations and outlook on future work.

In this thesis we have introduced a novel approach for user modeling in HCI: Multimodal HCI obstacles detection and corresponding UI adaptation.
We began in Chapter 1 with introducing the HCI research challenge which is treated in this thesis: an increasing number of diverse computer users (e.g. young, elderly, handicapped, experts, beginners), each can be individually impeded by various HCI obstacles (e.g. memObs or visObs). HCI obstacles which in turns can be either persistent or volatile in consecutive HCI interactions, resulting with very dynamic HCI sessions which need automatic support for improving user performance. After the general motivation, we introduced our approach for enabling an automatic detection of such HCI obstacles and continuous adaptation of UI in consecutive HCI sessions. We highlighted interdisciplinary arguments for each part of our decided framework: the matching pairs game as an exemplary HCI task, volatile and persistent HCI obstacles, and corresponding UI adaptations.
In Chapter 2, we recapped the evolution story of related research bodies: HCI, *User Modeling* and *User Experience*. We presented major milestones in each research body, highlighted gaps of modeling multimodal HCI obstacles, and discussed potential incorporation of our introduced models. In the same chapter, we recapped also background research bodies: the ACT theory and CMM for human cognitive modeling. Then, we presented the combined histor-

ical evolution of both related research and background research bodies, and, we eventually discussed the incorporation of our multimodal HCI obstacle detectors into those bodies: ACT theory, HCI and *User Modeling*.
In Chapter 3, we described all the user studies that we had designed to collect data for our experiments. For each user study, we presented the age and gender distribution of participants, their questionnaire responses as subjective analysis, and their calculated performances as objective analysis. In addition to data from real participants, we simulated the matching pairs playing behaviour under different conditions: with or without different HCI obstacles and/or UI adaptation. We discussed our simulation system, and evaluated those simulated logs by comparing their behaviour to their real logs references.
In Chapter 4, we explained our multimodal HCI obstacles detectors, namely EEG-based SVM memObs detectors and behaviour-based LSTM visObs detectors. We named them *Elementary Models*, because they contribute to but do not decide the final UI adaptation mechanism, which will be finally decided by the Dynamic Bayesian Network (DBN).
Chapter 5 represents the final chapter in the thesis pipeline, in which we integrate the *Elementary Models* together, show how they pass their outputs and confidences to the DBN as an overarching probabilistic model. We showed in evaluations how the DBN learns sequential dependencies between consecutive HCI sessions: to accordingly adapt the UI, consolidate or revert that UI adaptation in the next HCI sessions.

In the following sections, we summarize and visualize the thesis results, and highlight plausibility and contributions. While we visualize multifaceted results (subjective, objective, predictive) in Section 6.1, we highlight in Section 6.2 how valuable such correlated multifaceted results are for dynamically supporting users with (continuously improving) UI adaptations through consecutive HCI sessions. Then, we highlight in Section 6.3 a useful real-world *Transfer Learning* application of our HCI memObs detectors models: Dementia detection. Finally, we mention limitations of our models in Section 6.4, and we discuss potential follow-up research future works in Section 6.5.

6.1 Multifaceted Results

In this thesis, we have presented and discussed results from various facets: 1)*Subjective Facet*: subjective analysis results in Chapter 3. 2)*Objective Facet*: objective analysis results in Chapter 3. 3)*Predictive Facet*: *elementary models* prediction results in Chapter 4 and DBN results in Chapter 5.
We name the combination of all those 3 facets *Multifaceted Results* or *3-*

Faceted Results. In this section, we summarize and visualize all the *3-Faceted Results* in the thesis: (subjective, objective, predictive). We discuss the *3-Faceted Results* together, and highlight the plausibility of results across their facets. We integrate the results and discuss them from different perspectives: Multimodal data perspective (Section 6.1.1) and UI adaptation perspective (Section 6.1.2).

6.1.1 Multimodal Data Perspective

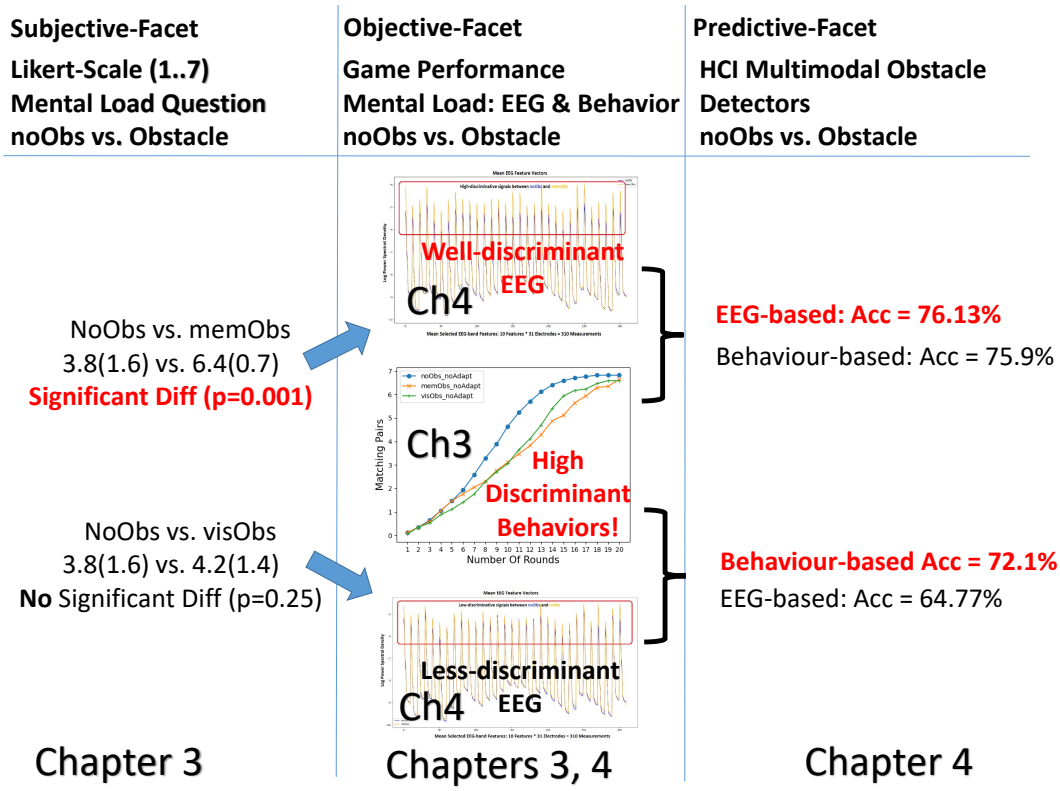


Figure 6.1 – Illustration of Multifaceted results from multimodal data perspective.

HCI session can be impeded by a variety of interaction obstacles, and multiple cognitive processes can be affected. Thus, we record multimodal data during HCI sessions to detect the presence of HCI obstacles. Generally speaking, each HCI obstacle can be naturally best detected from one specific data modality, e.g. MEMOBS causes an extra load on working memory in addition to the normal expected load during the HCI task. Thus, EEG data (as a brain activity data modality) is expected to be well-discriminant between NOOBS and MEMOBS. In contrast, VISOBS causes no extra memory load, thus, it is

expected to result in less-discriminant (interleaving) EEG signals between NOOBS and VISOBS. Instead, VISOBS causes confusion, which can be best detected from confused user behaviour depicted as recorded user actions. To evaluate this analysis, we integrate all the three multifaceted results from the thesis pipeline: subjective analysis (Likert-scale mental load question) and objective analysis (plotted EEG signals and behavioral performance) from Chapter 3, and predictive results (HCI multimodal obstacle detectors) from Chapter 4. We illustrate all the 3Faceted-Results in Figure 6.1, where we find plausible results and correlations across all the 3 facets:

1. *Subjective Facet*: participants feel that memObs game needs significantly higher mental load than noObs, while they feel no significant additional mental load when comparing visObs to noObs. See the first column in Figure 6.1, and recall Chapter 3 for more details.

2. *Objective Facet*: Recorded EEG signals from subjects coincide with their subjective findings, showing well-discriminant EEG signals between noObs and memObs, and less-discriminant EEG signals between noObs and visObs. Behavioural Data, depicted as a sequential performance measurement (matching pairs per round) shows discriminant behaviours when comparing NOOBS to both MEMOBS and VISOBS. See the second column in Figure 6.1.
   This is an average based comparison between average subjective results and average EEG-based and behavioral objective results. While we could also investigate a subject-wise comparison, we did only the average-based comparison, because we train person-independent models (predictive results).

3. *Predictive Facet*: results also coincide with subjective and objective analysis, showing the best accuracy for MEMOBS detection using EEG data modality, and for VISOBS detection using behavioral data modality, which is a good discriminant modality also for both MEMOBS and VISOBS. See the third column in Figure 6.1.

Flexible and Extendable Design

After we have seen plausible correlations in our multifaceted results, we summarize and highlight hereby our modular design for HCI obstacles detection. The modular fashion of our multimodal HCI obstacle detection approach is one of the powerful contributions of this thesis, because it ensures flexibility, plausibility and extend-ability for our introduced model. That is, we neither combine heterogeneous data modalities nor train monolithic obstacle detectors for discriminating all obstacles at once, because such approaches clearly lack of flexibility and re-usability. Instead, we train simple binary obstacle

detectors from each data modality alone, so called *Elementary models*. This is very plausible for real world scenarios in which one can face a whatever obstacle which can be best detected from one data modality; Multimodal *Elementary models* will pass their outputs and confidences, where all those modular models will predict the presence of their HCI obstacle. In case of a "novel obstacle" (e.g. MEMOBS session tested by VISOBS *Elementary model*), low confidences are expected to result with the predictions outputs, while appropriate tested sessions are expected to result, in contrast, with high confidences. Multimodal *Elementary Models* pass their outputs together with confidences to the overarching probabilistic model (DBN), which in turns predicts the most suitable UI adaptation, because it learnt, in the training phase, the distributions of predictions and confidences from those *Elementary models*, recall Chapter 5 for DBN details.

6.1.2 UI Adaptation Perspective

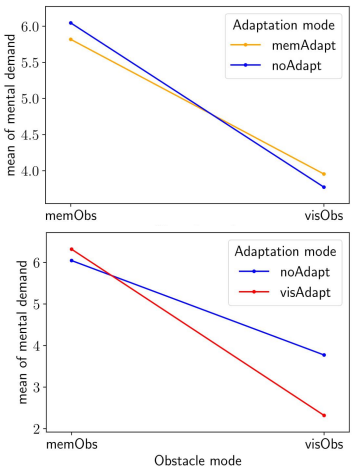
The ultimate target of this thesis is to enable an automatic and continuous support for computer users, in terms of suitable UI adaptations. For this purpose, we designed different UI adaptations (Chapter 3), from which the proposed probabilistic model DBN (Chapter 5) will continuously select the most suitable UI adaptation (during each tested HCI session, different obstacles predictions and confidences are received from the above mentioned *Elementary Models*, proposed in Chapter 4).

In this section, we integrate the multifaceted results (subjective, objective and predictive) from the UI adaptation perspective. Thus, we look at participants' subjective impressions of the different designed UI adaptation mechanisms, and we compare their subjective feelings, depicted via Likert-scale questions' responses, to the actual impact of such UI adaptation mechanisms on the user performance (objective impact). The objective impact has been depicted from real performance measurements of participants during HCI sessions under those tested UI adaptation mechanisms. We also compare the subjective and objective results facets to the corresponding predictive results facet: obstacle detectors (Chapter 4) and DBN results under different UI adaptation mechanisms (Chapter 5).

In Figure 6.2, we visualize the multifaceted results from *UI Adaptation* perspective. In the subjective facet, we show a two-way ANOVA with the obstacle mode and adaptation mode as independent variables and the self-reported mental demand as dependent variable. We observe a significant interaction effect ($F = 7.67$, $p = 7.2e^{-4}$, and $\eta = 0.05$), which shows that indeed the designed *UI Adaptations* are helpful to support the user in case of

Subjective-Facet

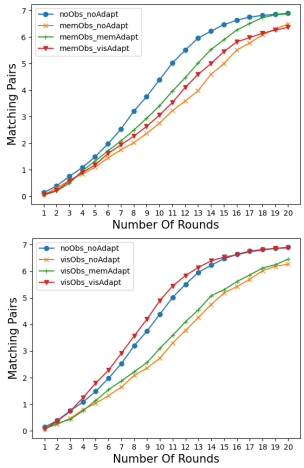
Mental Load Likert-scale Interaction effect between Obs. Mode & Adapt. Mode



Chapter 3

Objective-Facet

Objective Analysis: User Performance noObs vs. Obs_UIAdapt



Chapter 3

Predictive-Facet

HCI Obstacle Detector noObs vs. Obs_UIAdapt

LSTM MemObs Detector	Accuracy(%)
MemObs NOAdapt	75.9
MemObs MemAdapt	64.1
MemObs VisAdapt	69.9

SVM MemObs Detector	Accuracy(%)
MemObs NOAdapt	76.13
MemObs MemAdapt	71.59
MemObs VisAdapt	82.95

LSTM VisObs Detector	Accuracy(%)
VisObs NoAdapt	72.1
VisObs MemAdapt	66.3
VisObs VisAdapt	56.2

SVM VisObs Detector	Accuracy(%)
VisObs NOAdapt	64.77
VisObs MemAdapt	61.63
VisObs VisAdapt	42.04

Chapter 4

Figure 6.2 – Illustration of multifaceted results from UI Adaptation perspective.

a corresponding HCI obstacle (e.g. *memAdapt* for *memObs*), and detrimental otherwise (e.g. *memAdapt* for *visObs*).

In the objective facet (2nd column in Figure 6.2), we show the user performance (matching pairs per round) under different conditions: NOOBS and *obstacle* with different UI adaptation mechanisms. These objective result (both figures in the 2nd column) coincide with the subjective mental load responses, where corresponding designed *UI adaptations* (e.g. *memAdapt* for *memObs*) have statistically significant improvements in user performance, while non-corresponding *UI adaptatios* (e.g. *visAdapt* for *memObs*) show no significant improvement in user performance.

The third facet (predictive facet) goes also in the same direction, where those subjective and objective measurements are reflected on the different obstacle detectors performances (accuracy): The best accuracy is gained when detecting an obstacle in an HCI session without a UI adaptation (NOADAPT), while the worst accuracy is gained when detecting an obstacle under its corresponding *UI Adaptation*.

The presented correlations between different results facets (subjective, objective and predictive) show, on the one hand, validity for the designed experiments, and on the other hand, they highlight *plausible behaviours* shown by the different obstacle detectors (*Elementary Models*), e.g. the worst accuracy is gained when detecting an obstacle under its corresponding *UI Adaptation* (3rd column in Figure 6.2). Such plausible behaviours of *Elementary Models* will generate realistic distributions of predictions and corresponding confidences, which will feed the probabilistic overreaching model (DBN) to dynamically select the most probable UI adaptation, see Section 6.2.

6.2 Dynamic Support with UI Adaptations

In this section, we summarize the DBN results discussed in Chapter 5, and argue them according to those integrated multifaceted results in Section 6.1. Figure 6.3 shows that the DBN outperforms the baseline in the first interaction (confusion matrix A) with 72.5% as an average accuracy (chance level baseline is 33.33% as we have three adaptation mechnisms: noAdapt, memAdapt and visAdapt). Furthermore, it shows that the DBN further learns through the next interaction and drastically improves, with an almost perfect average accuracy: 99.7% (confusion matrix B). For more concrete evaluation, we look at confusion matrices for first and second interactions (A and B), moreover, in the second interaction, we look also at a confusion matrix for both persistent obstacles (confusion matrix C) and volatile obstacles (confusion matrix D).

In the first interaction (A), confusion matrix results show the highest accuracy for both noAdapt and memAdapt (82.35%), and the lowest for visAdapt (52.94%). While even the latter performance outperforms the chance level baseline (33.33%), considerable number of sessions can be wrongly not adapted in that first interaction (47.06%). However, the 2nd interaction results (confusion matrix B) show that DBN learns sequential dependencies through consecutive sessions well, and drastically improves, with an almost perfect average accuracy: 99.7%. The prediction accuracy of all labels improves when re-deciding the UI adaptation in the second interaction. That is, the accuracy of all the predicted labels in the first interaction improves from 82.4%, 82.4%, and 52.9% to 100%, 100%, and 99.02% for noAdapt, memAdapt and visAdapt respectively.
We observe the greatest improvement for the visAdapt case (a relative improvement of 87.04%) as the model is able to recover many cases in which the original visObs was not detected. This is possible because the visAdapt class does not exhibit false positives in the first interaction and evidence can

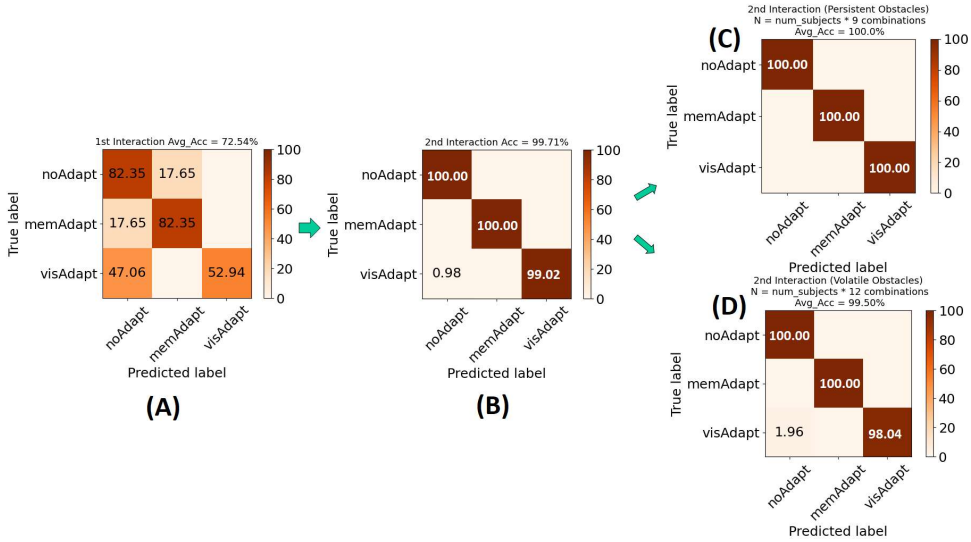


Figure 6.3 – Left (A): DBN first interaction evaluation. Middle (B): DBN second interaction evaluation overall persistent and volatile obstacles. Right (C and D): detailed evaluation for the second interaction given only persistent obstacles (upper matrix, C) and only volatile obstacles (lower matrix, D) through 1st and 2nd interactions.

accumulate. This also coincides with the multifaceted results discussed above. For obstacle mode, we conclude from the multifaceted results in Figure 6.1 that MEMOBS is stronger than VISOBS according to all the multifaceted results: subjective responses, objective performance and predictive accuracy. For adaptation mechanism mode, however, we conclude from Figure 6.2 that VISADAPT is subjectively and objectively much stronger than MEMADAPT.

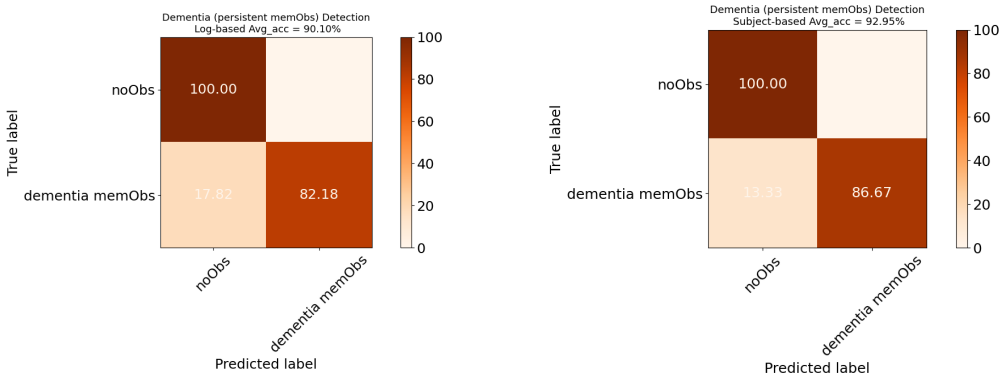
If we combine both obstacle mode and adaptation mode arguments, we can argue the weaker performance of VISOBS detectors compared to MEMOBS detectors: for example, in case of corresponding *UI Adaptation*, VISOBS is difficult to be detected from a VISOBS_VISADAPT session, and more difficult than detecting a MEMOBS from a MEMOBS_MEMADAPT session, because in the former session the weaker obstacle (VISOBS) is combined with the stronger adaptation mechanism VISADAPT. Thus, such difficulties in detecting VISOBS, compared to MEMOBS, argue that lowest VISADAPT accuracy in the first interaction (confusion matrix A). However, DBN can learn sequential dependencies between consecutive sessions, and drastically improves in the next HCI sessions, even with volatile or persistent cases of obstacles, see the 2nd interaction confusion matrices in Figure 6.3 (B, C and D).

6.3 Transfer Learning: Dementia Detection

In general, machine learning models, trained for specific experiment settings, are much more valuable when they show transfer learning capabilities, e.g. when those trained models can perform well also for other experiment settings. That is, researchers typically design lab-based settings for mimicking real-world scenarios, and train and test models. Such models are much more valuable when they also perform well in real-world scenarios, from which typically only small amounts of data in less-controlled settings can be collected.

In this section, we recall the powerful and very useful transfer learning capability of our behaviour-based volatile MEMOBS detector models, which have been shown in Chapter 4 to perform well in detecting real MEMOBS: dementia. That is, such models have been trained from game logs collected from young subjects, for whom we simulated a MEMOBS as a secondary-task memory load. We trained those models from behavioral data (user actions, i.e. game logs) collected from young healthy subjects, mainly university students. As we explained in details in Chapter 3 and Chapter 4, the experiments and the detector model were designed as follows: an LSTM has been trained from logs collected from young subjects for discriminating between their NOOBS logs and MEMOBS logs (secondary task memory load).

Having those models trained with easily adjusted lab-based settings with such young subjects, we tested the same models on logs collected from standard matching pairs games played by elderly subjects: healthy elderly subjects' logs should be labeled by our models as NOOBS, and dementia patients' logs should be labeled as MEMOBS. The models performed well (average accuracy: 90.10% for elderly subjects: healthy and dementia patients), even outperforming the average test accuracy regarding the same lab-based settings (average accuracy:75.9% for young subjects: NOOBS and MEMOBS). We explained this improvement when we tested our models with that real-world dementia settings by the largest drop of dementia patient performance. Our models were able to discriminate dementia playing behaviour from elderly healthy playing behaviour. This is especially a powerful generalization, because one can collect a lot of training data episodes from young subjects, extend or adjust the model and/or setup etc. and re-collect training data from young volunteer subjects, which is easier and more practical than collecting training data from the population of elderly. As we collected multiple logs from elderly subjects (both healthy and dementia patients), we investigated also whether that accuracy (90.10%) further improves when we vote predicted labels and evaluate subject-label (so-called subject-acc) rather than evaluating each single log (log-acc), see Figure 6.4.



(a) LOG_BASED ACC (b) SUBJECT_BASED ACC

Figure 6.4 – Dementia detection: multiple testing logs enables improved subject_based acc.

These multiple logs guarantee more robust detection test, where only one individual matching pairs game is likely effected by luck. We see in Figure 6.4 that dementia logs accuracy (82.18%) is improved to (86.67%) for labeling dementia patients, and in total the dementia detection accuracy is improved by voting mechanism from 90.10% to 92.95%. Thus, we contribute to the state of the art models for detecting dementia using a machine learning solution; We propose a robust behaviour-based, sensory-free novel model for detecting dementia from matching pairs game as a memory-intensive practical test. The more the games played by a test subject, the better and more confident (by reducing the luck factor) accuracy of dementia detection is achieved. The most important point is, such models were easily trained, and will be also easily adjusted, by logs collected from **young** subjects, who are easily available volunteers population for potential follow-up research.

6.4 Limitations

In this section, we list limitations which we have in our current approach. The following limitations (challenges) can be treated in follow-up future works:

- One limitation of the presented work is that the interaction obstacles were only simulated for the purpose of a controllable study with comparable conditions between all participants. However, we have shown in Chapter 4 that the described approach also works with real interaction obstacles, namely naturally limited WM capacity. The investigated obstacles were chosen to cover both persistent and volatile obstacles

as well as obstacles resulting from impairment of perception as well as internal cognition.
Nevertheless, future work needs to further tackle the challenge of transferring the developed models to users experiencing real interaction obstacles. For example, we showed how the MEMOBS detectors generalize well to detect dementia. However, we did not investigate the ability of the simulated color blindness VISOBS detector to detect a real color blindness obstacle, because we do not have data collected from color blindness participants.

- As mentioned in Chapter 4, we considered the presence of a single obstacle in a an HCI session. However, multiple HCI obstacles could simultaneously occur at the same HCI session. In this case, we have to model potential interaction dependencies between such parallel obstacles. Moreover, we should introduce new UI adaptation mechanisms for tackling multiple parallel obstacles. Actually, modelling of such interactions between obstacles is a challenge, as multiple cognitive processes will be affected, e.g. if we assume that the MEMOBS and VISOBS present simultaneously in an HCI session, both the cognition and perception processes will be simultaneously affected, and thus, the well discriminant EEG signals shown in this thesis for discriminating MEMOBS would be affected by confused behaviour resulted by VISOBS.
  We find challenges in different approaches for detecting the presence of multiple obstacles in one session. For Example, how well can each individual obstacle detector detect the presence of its obstacle? Would a new model which is trained from new data, collected under the presence of multiple obstacles, be realistic to detect the presence of parallel obstacles together? Moreover, how to design an effective UI adaptation to tackle parallel obstacles together, where compensating an obstacle can strengthen the effects of other parallel obstacles, e.g. auditory identifiers which are very effective to tackle a VISOBS will not suit an impaired WM in case of parallel MEMOBS. For all these challenges, we assumed that in a relative short HCI session (prefix of matching pairs game with 10 rounds), no multiple parallel obstacles occur.

- We tackled the problem of a one-fits-all UI by detecting HCI obstacles and compensating accordingly. However, we investigated, as proof-of-concept, one UI Adaptation for each discussed HCI obstacle (except memAdapt for which we did investigated strong and light UI Adaptations variants), i.e. we are offering one-fits-all UI Adaptation which in turns may not fit all users for tackling the detected HCI obstacle.
  Thus, an important follow-up future work would be to design multiple

UI Adaptation for each discussed HCI obstacle, and to evaluate the presence of HCI obstacle with each one. The Elementary obstacles detectors should be extended, and the DBN should learn then to select the most suitable UI adaptation for that user. To do so, however, we need a large number of diverse subjects, who evaluate those multiple UI Adaptations (e.g. with likert-scale value). We would need to build user models, encapsulating e.g. user identity, impression (likert-scale responses), gender, age, etc. Thus, ML models should learn correlations between the collected multimodal data and such UI Adapt preferences, which would probably further improve the user modeling achieved. However, this is an open challenge as it needs very overloading experiments.

6.5 Outlook: Adaptive Driver Assistance System

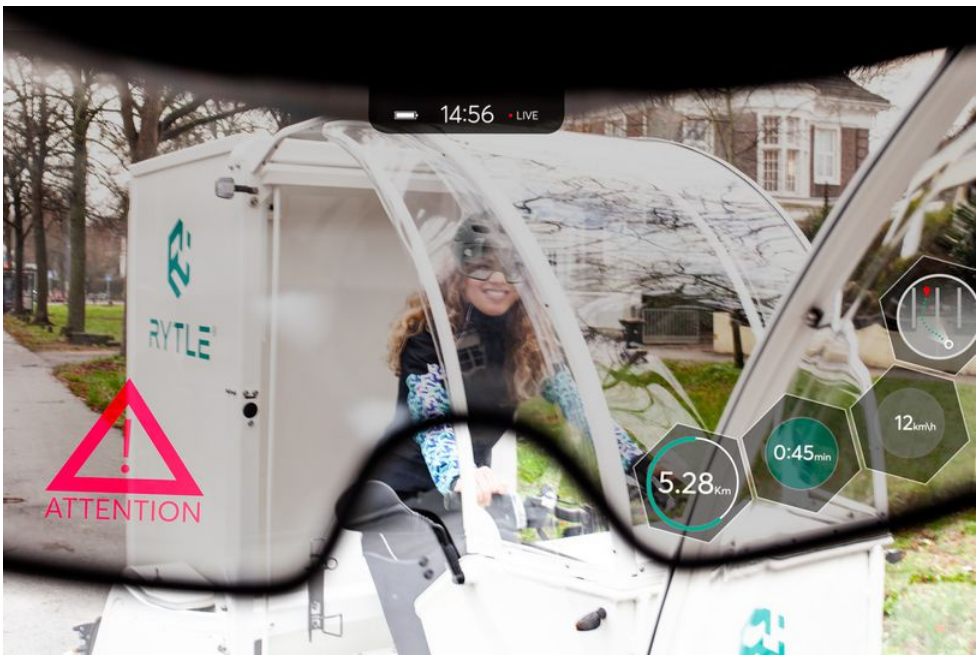


Figure 6.5 – SmartHelm project: Modeling attention distraction and warning support. Copyright: this figure is created by Gordon Linnemann, 12.12.2019, Bremen, in the context of the SmartHelm project

In this section, we introduce a potential application, in which machine learning approaches for cognitive modeling and user support would be beneficial. As we investigated the cognitive modeling of MEMOBS and VISOBS in HCI, we could

investigate modeling of another important obstacle: *Attention Distraction*, which is a very important obstacle in HCIs tasks, especially in attention-based activities such as driving modern vehicles equipped with computer-based assistant systems.

When someone drives or bikes, her or his attention plays a crucial role for safety. Distracted attention, even for a relative short time, may cause accidents. Cognitive modeling of attention states is a hot topic in current research driving assistant systems. For example, in the SmartHelm project Küster et al. (2020), the researchers are aiming at modeling attention states of a Cargo cyclist from recorded multimodal data: EEG and eye-tracking. The cyclist, who is intended to be supported via a smart helmet, will be supported with augmented reality real-time alerts in case of detected *attention distraction*, see Figure 6.5.

Although the SmartHelm is a specific HCI application (driving assistance), our models in this thesis can be incorporated, extended and re-used for the purpose of detection of HCI obstacles. That is, the detection of *attention distraction* can be achieved via multimodal obstacle detectors from EEG and eye-tracking data. Then, those models can be integrated under the overarching DBN probabilistic model, which should be extended with new UI adaptation mechanisms, namely *Alerts*, see Figure 6.5. It would also be valuable to investigate the detection of memObs and visObs with the SmartHelm during riding; While we could investigate the detection of low WM capacity as a memObs, we have to simulate realistic variants of visObs to be detected during riding. Moreover, we could also extend our proposed modular architecture with more obstacles to be detected, e.g. rider fatigue from EEG, eye-tracking and behavioral data (rider actions).

Glossary

Elementary Models An elementary model is a part of our introduced cognitive adaptive model. That is, each obstacle detector (memory-based obstacle or visual obstacle) is considered as an elementary model because it contributes in but does not produce the final decision of UI adaptation, which will be decided by the probabilistic model.. 11

Online Cognitive Adaptive Model Online Cognitive Adaptive Model uses inputs recorded from multimodal human cognitive processes (e.g. EEG and behavioral data) to automatically detect an interaction obstacle and correspondingly adapt the UI. It consists of a probabilistic decision model which decides the best UI adaptation based on inputs from different obstacles detectors so-called elementary models.. 3

Probabilistic Model Probability-based Model is the overarching decision model upon the underlying elementary models in our introduced cognitive adaptive model architecture. It uses inputs from underlying elementary models, and decides correspondingly the most probable UI adaptation.. 12

User Experience *User Experience*, for a computer application, is defined by tracking how a user experiences that application through its UIs during corresponding HCI sessions. That is, Hassenzahl (2018) systemically illustrated how a model of *User Experience* constitutes upon the so-called *product character* (i.e. application character).. 1, 2

User Modeling *User Modeling* research aims at defining user models for HCI applications for building up and modifying a conceptual understanding of the user, Fischer (2001). 1, 2

HCI Human Computer Interaction (HCI) is a multidisciplinary field of study focusing on the design of computer technology and, in particular, the interaction between humans (the users) and computers (computer can be a desktop, laptop, tablet or smart phone). 1

memAdapt Memory Adaptation is a change to the User Interface (UI) to tackle the expected deteriorated user performance caused by memory obstacle, for example repetition of information in the UI. 10

memObs Memory-based Interaction Obstacle is a condition which impedes human memory during Human Computer Interaction, for example a memory-loading secondary task, deficiency in short-term memory or dementia.. 2, 10

visAdapt Visual Adaptation is a change to the User Interface (UI) to tackle the expected deteriorated user performance caused by visual obstacle, for example auditory instructions.. 10

visObs Visual Interaction Obstacle is a condition which impedes human visual recognition process during Human Computer Interaction, for example glare effects on mobile display or red-green colour vision deficiency.. 2, 10

UI User Interface is the means by which the user and a computer system interact, in particular the use of input devices and software. This includes the Graphical User Interface (GUI) as well as other interaction channels such as auditory instructions. 2

Bibliography

Tarō Adachi. *Utilization of usage data to improve organizational memory*. PhD thesis, University of Colorado, 1998.

Erik M Altmann and J Gregory Trafton. Memory for goals: An activation-based model. *Cognitive science*, 26(1):39–83, 2002.

John R Anderson. Cognitive science series. the architecture of cognition, 1983.

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. ISSN 1939-1471(Electronic);0033-295X(Print).

John Robert Anderson. *The adaptive character of thought*. Psychology Press, 1990.

JR Anderson. Rules of the mind hillsdale. *New Jersey, Laurence Erlbaum Associates,, 319p*, 1993.

JR Anderson and GH Bower. Human associative memory: revised edition. *Hillsdale, Winston & Sons, Washington DC*, 1973.

Ankur Ankan and Abinash Panda. pgmpy: Probabilistic graphical models using python. In *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*, 2015.

Zoe Arvanitakis, Raj C Shah, and David A Bennett. Diagnosis and management of dementia. *Jama*, 322(16):1589–1599, 2019.

Carryl L Baldwin and BN Penaranda. Adaptive training using an artificial neural network and eeg metrics for within-and cross-task workload classification. *NeuroImage*, 59(1):48–56, 2012.

Chris Berka, Daniel J. Levendowski, Michelle N. Lumicao, Alan Yau, Gene Davis, Vladimir T. Zivkovic, Richard E. Olmstead, Patrice D. Tremoulet, and Patrick L. Craven. EEG Correlates of Task Engagement and Mental

Workload in Vigilance, Learning, and Memory Tasks. *Aviation, Space, and Environmental Medicine*, 78:B231–B244, 2007.

Nigel Bevan. International standards for hci and usability. *International journal of human-computer studies*, 55(4):533–552, 2001.

RA Bolt. The human interface where people and computers. *Meet Lifetime Learning Publications*, 1984.

Davie M Buss and Heidi Greiling. Adaptive individual differences. *Journal of Personality*, 67(2):209–243, 1999.

Stuart K Card, Thomas P Moran, and Allen Newell. Computer text-editing: An information-processing analysis of a routine cognitive skill. *Cognitive psychology*, 12(1):32–74, 1980a.

Stuart K Card, Thomas P Moran, and Allen Newell. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7):396–410, 1980b.

Stuart K Card, Thomas P Moran, and Allen Newell. The psychology of human-computer interaction. hillsdale, nj.: Lawrence rrlbaum associates. *Inc.*, 1983.

John M Carroll and Robert L Campbell. Softening up hard science: reply to newell and card. *Human-Computer Interaction*, 2(3):227–249, 1986.

Shalu Chaudhary, Sachin Taran, Varun Bajaj, and Abdulkadir Sengur. Convolutional neural network based approach towards motor imagery tasks eeg signals classification. *IEEE Sensors Journal*, 2019.

Edward H Chen and Drew H Bailey. Dual-task studies of working memory and arithmetic performance: A meta-analysis. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 2020.

Cuing-Hung Cheng and Ying-Wen Bai. Using fuzzy logic and light-sensor for automatic adjustment of backlight brightness in a mobile computer. In *2012 IEEE 16th International Symposium on Consumer Electronics*, pages 1–5. IEEE, 2012.

Andrew RA 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.

Colin Cooper. *Individual differences*. Arnold London, 1998.

Gregory W Corder and Dale I Foreman. Nonparametric statistics for non-statisticians, 2011.

Robert G Crowder. Visual and auditory memory. 1972.

Jean-Baptist Du Prel, Bernd Röhrig, Gerhard Hommel, and Maria Blettner. Choosing statistical tests: part 12 of a series on evaluation of scientific publications. *Deutsches Ärzteblatt International*, 107(19):343, 2010.

D. Eck and J. Schmidhuber. Finding temporal structure in music: blues improvisation with LSTM recurrent networks. In *Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing, 2002*, pages 747–756, 2002.

Dennis E Egan. Individual differences in human-computer interaction. In *Handbook of human-computer interaction*, pages 543–568. Elsevier, 1988.

Xavier Ferre, Natalia Juristo, and Ana M Moreno. Obstacles for the integration of hci practices into software engineering development processes. In *Encyclopedia of Human Computer Interaction*, pages 422–428. IGI Global, 2006.

Gerhard Fischer. Domain-oriented design environments. *Automated software engineering*, 1(2):177–203, 1994.

Gerhard Fischer. User modeling in human–computer interaction. *User modeling and user-adapted interaction*, 11(1-2):65–86, 2001.

Gerhard Fischer and Curt Stevens. Volunteering information—enhancing the communication capabilities of knowledge-based systems. In *Human–Computer Interaction–INTERACT'87*, pages 965–971. Elsevier, 1987.

Gerhard Fischer, Andreas C Lemke, Thomas Mastaglio, and Andres I Morch. The role of critiquing in cooperative problem solving. *ACM Transactions on Information Systems (TOIS)*, 9(2):123–151, 1991.

David B Fogel. An introduction to simulated evolutionary optimization. *IEEE transactions on neural networks*, 5(1):3–14, 1994.

Krzysztof Gajos and Daniel S Weld. Supple: automatically generating user interfaces. In *Proceedings of the 9th international conference on Intelligent user interfaces*, pages 93–100. ACM, 2004.

Krzysztof Z Gajos, Jacob O Wobbrock, and Daniel S Weld. Automatically generating user interfaces adapted to users' motor and vision capabilities. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 231–240, 2007.

Krzysztof Z Gajos, Jacob O Wobbrock, and Daniel S Weld. Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 1257–1266, 2008.

Krzysztof Z Gajos, Daniel S Weld, and Jacob O Wobbrock. Automatically generating personalized user interfaces with supple. *Artificial Intelligence*, 174(12-13):910–950, 2010.

Eduardo H Calvillo Gámez, Paul Cairns, and Anna L Cox. From the gaming experience to the wider user experience. *People and Computers XXIII Celebrating People and Technology*, pages 520–523, 2009.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10):2451–2471, 2000.

Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. In *Hidden Markov models: applications in computer vision*, pages 9–41. World Scientific, 2001.

Louis M Gomez, Dennis E Egan, Evangeline A Wheeler, Dhiraj K Sharma, and Aleta M Gruchacz. How interface design determines who has difficulty learning to use a text editor. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 176–181, 1983.

Louis M Gomez, Dennis E Egan, and Cheryl Bowers. Learning to use a text editor: Some learner characteristics that predict success. *Human-Computer Interaction*, 2(1):1–23, 1986.

Zhichen Gong and Huanhuan Chen. Model-based oversampling for imbalanced sequence classification. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1009–1018. ACM, 2016.

Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):855–868, 2009.

Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.

Kalanit Grill-Spector and Nancy Kanwisher. Visual recognition: As soon as you know it is there, you know what it is. *Psychological Science*, 16(2): 152–160, 2005.

Jan-Eric Gustafsson and Johan Olav Undheim. Individual differences in cognitive functions. 1996.

Mehdi Hajinoroozi, Zijing Mao, Tzyy-Ping Jung, Chin-Teng Lin, and Yufei Huang. Eeg-based prediction of driver's cognitive performance by deep convolutional neural network. *Signal Processing: Image Communication*, 47:549–555, 2016.

Sajanee Halko and Julie A Kientz. Personality and persuasive technology: an exploratory study on health-promoting mobile applications. In *International conference on persuasive technology*, pages 150–161. Springer, 2010.

Richard H. R. Harper, Michael G. Lamming, and William M. Newman. Locating systems at work: implications for the development of active badge applications. *Interacting with Computers*, 4(3):343–363, 1992.

Marc Hassenzahl. The thing and i: understanding the relationship between user and product. In *Funology 2*, pages 301–313. Springer, 2018.

Haibo He, Yang Bai, E. A. Garcia, and Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, June 2008.

Dominic Heger, Felix Putze, and Tanja Schultz. An adaptive information system for an empathic robot using EEG data. In *Social Robotics*, pages 151–160. Springer, 2010.

C. Herff, O. Fortmann, Chun-Yu Tse, Xiaoqin Cheng, F. Putze, D. Heger, and T. Schultz. Hybrid fNIRS-EEG based discrimination of 5 levels of memory load. In *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 5–8, 2015.

William C Hill, James D Hollan, Dave Wroblewski, and Tim McCandless. Edit wear and read wear. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 3–9, 1992.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166, 1999.

Eric J Horvitz, John S Breese, David Heckerman, David Hovel, and Koos Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. *arXiv preprint arXiv:1301.7385*, 2013.

Wei-Yen Hsu. Eeg-based motor imagery classification using neuro-fuzzy prediction and wavelet fractal features. *Journal of Neuroscience Methods*, 189(2):295–302, 2010.

Matthew Hyde, Maxwell Scott-Slade, Hugo Scott-Slade, Michael Hornberger, Hugo Spiers, Ruth Dalton, Christoph Hoelscher, Jan Wiener, and Veronique Bohbot. Sea hero quest: The world's first mobile game where anyone can help scientists fight dementia. 2016.

Luke Jefferson and Richard Harvey. An interface to support color blind computer users. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1535–1538, 2007.

Taeho Jo, Kwangsik Nho, and Andrew J. Saykin. Deep learning in alzheimer's disease: Diagnostic classification and prognostic prediction using neuroimaging data. *Frontiers in Aging Neuroscience*, 11:220, 2019. ISSN 1663-4365. doi: 10.3389/fnagi.2019.00220. URL `https://www.frontiersin.org/article/10.3389/fnagi.2019.00220`.

John Karat. Software evaluation methodologies. In *Handbook of human-computer interaction*, pages 891–903. Elsevier, 1988.

Ioanna Katidioti, Jelmer P Borst, Douwe J Bierens de Haan, Tamara Pepping, Marieke K van Vugt, and Niels A Taatgen. Interrupted by your pupil: An interruption management system based on pupil dilation. *International Journal of Human–Computer Interaction*, 32(10):791–801, 2016.

Yufeng Ke, Hongzhi Qi, Feng He, Shuang Liu, Xin Zhao, Peng Zhou, Lixin Zhang, and Dong Ming. An eeg-based mental workload estimator trained on working memory task can work well under simulated multi-attribute task. *Frontiers in human neuroscience*, 8, 2014.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Torkel Klingberg, Hans Forssberg, and Helena Westerberg. Increased brain activity in frontal and parietal cortex underlies the development of visu-

ospatial working memory capacity during childhood. *Journal of cognitive neuroscience*, 14(1):1–10, 2002.

Conrad Aleksander Kreek, Jerry Mailloux, Ahmed Hashi, James Robinson, Ken Wu, Robert J Lowles, Andrew Bocking, Dale Brubacher-Cressman, and Bruce Winter. Automatic screen and keypad brightness adjustment on a mobile handheld electronic device, April 20 2010. US Patent 7,701,434.

Jyotish Kumar et al. Affective modelling of users in hci using eeg. *Procedia Computer Science*, 84:107–114, 2016.

Mike Kuniavsky. *Observing the user experience: a practitioner's guide to user research*. Elsevier, 2003.

Dennis Küster, Johannes Schering, Christian Janßen, Felix Putze, Jorge Marx Gómez, and Tanja Schultz. Intelligente und aufmerksamkeitssensitive systeme in der fahrradmobilität. In *Mobilität, Erreichbarkeit, Raum*, pages 143–158. Springer, 2020.

John E Laird, Allen Newell, and Paul S Rosenbloom. Soar: An architecture for general intelligence. *Artificial intelligence*, 33(1):1–64, 1987.

Heidi Lam and Patrick Baudisch. Summary thumbnails: readable overviews for small screen web browsers. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 681–690, 2005.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Olivier Ledoit and Michael Wolf. Honey, i shrunk the sample covariance matrix. *The Journal of Portfolio Management*, 30(4):110–119, 2004.

Stephan Lewandowsky, Klaus Oberauer, Lee-Xieng Yang, and Ullrich KH Ecker. A working memory test battery for matlab. *Behavior Research Methods*, 42(2):571–585, 2010.

Fabien Lotte. A tutorial on eeg signal-processing techniques for mental-state recognition in brain–computer interfaces. In *Guide to Brain-Computer Music Interfacing*, pages 133–161. Springer, 2014.

Wendy E Mackay. Triggers and barriers to customizing software. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 153–160, 1991.

S Mandal, BK Singh, and K Thakur. Classification of working memory loads using hybrid eeg and fnirs in machine learning paradigm. *Electronics Letters*, 56(25):1386–1389, 2020.

Thomas W Mastaglio. A user modelling approach for computer-based critiquing. Technical report, COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 1990.

Representational Memory. Paivio's levels of meaning as experiential model and conceptual framework anderson, 1. r. language. memory and thought, 1976.

Samuel A Miller. Tactile glove for human-computer interaction, October 6 2020. US Patent 10,795,448.

Patrick Alfred Pierce Moran. Random processes in genetics. In *Mathematical proceedings of the cambridge philosophical society*, volume 54, pages 60–71. Cambridge University Press, 1958.

Christian Mühl, Camille Jeunet, and Fabien Lotte. Eeg-based workload estimation across affective contexts. *Frontiers in neuroscience*, 8, 2014.

Bennet B Murdock Jr. The serial position effect of free recall. *Journal of experimental psychology*, 64(5):482, 1962.

Kevin P Murphy et al. Dynamic bayesian networks. *Probabilistic Graphical Models, M. Jordan*, 7:431, 2002.

Kumiyo Nakakoji. *Increasing shared understanding of a design task between designers and design environments: the role of a specification component*. PhD thesis, Citeseer, 1993.

Ernst Niedermeyer and FH Lopes da Silva. *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.

Judith Reitman Olson and Gary M Olson. The growth of cognitive modeling in human-computer interaction since goms. In *Readings in Human–Computer Interaction*, pages 603–625. Elsevier, 1995.

Conrado F Ostia, Dionis A Padilla, Febus Reidj, G Cruz, Roy Jayson D Galang, Anne Shirley M Josafat, and Edrick F Victoria. Electronic vision system with personalized calibration of color compensation for people with partial color vision deficiency using raspberry pi digital image processing. In *2019 5th International Conference on Control, Automation and Robotics (ICCAR)*, pages 315–319. IEEE, 2019.

Daniel O'sullivan. Adaptive human computer interface (aahci), December 6 2012. US Patent App. 12/475,681.

A Ant Ozok. Survey design and implementation in hci. *Human-Computer Interaction: Development Process*, 253, 2009.

Gert Pfurtscheller, Christa Neuper, Alois Schlogl, and Klaus Lugger. Separability of eeg signals recorded during right and left motor imagery using adaptive autoregressive parameters. *IEEE transactions on Rehabilitation Engineering*, 6(3):316–325, 1998.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Peter G Polson and Clayton H Lewis. Theory-based design for easily learned interfaces. *Human–Computer Interaction*, 5(2-3):191–220, 1990.

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. Springer, 2011.

Felix Putze and Tanja Schultz. Adaptive cognitive technical systems. *Journal of neuroscience methods*, 234:108–115, 2014.

Felix Putze, Tanja Schultz, Sonja Ehret, Heike Miller-Teynor, and Andreas Kruse. Model-based evaluation of playing strategies in a memo game for elderly users. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 929–934. IEEE, 2015.

Felix Putze, Mazen Salous, and Tanja Schultz. Detecting memory-based interaction obstacles with a recurrent neural model of user behavior. In *Proceedings of the 2018 International Conference on Intelligent User Interfaces*, IUI '18, pages 205–209, Tokyo, Japan, 2018. ACM. ISBN 978-1-4503-4945-1/18/03. doi: 10.1145/3172944.3173006. URL `https://www.csl.uni-bremen.de/cms/images/documents/publications/putze_salous_2018iui.pdf`.

Abu Zohran Qaiser and Muhammad Taha Khan. Adaptive interface for accommodating colour-blind users by using ishihara test. *arXiv preprint arXiv:1712.03329*, 2017.

Elaine Rich. Stereotypes and user modeling. In *User models in dialog systems*, pages 35–51. Springer, 1989.

Frank E Ritter, Farnaz Tehranchi, and Jacob D Oury. Act-r: A cognitive architecture for modeling cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, 10(3):e1488, 2019.

David Rozado, Andreas Duenser, and Ben Howell. Improving the performance of an eeg-based motor imagery brain computer interface using task evoked changes in pupil diameter. *PloS one*, 10(3):e0121262, 2015.

Marjan Saadati, Jill Nelson, and Hasan Ayaz. Convolutional neural network for hybrid fnirs-eeg mental workload classification. In *International Conference on Applied Human Factors and Ergonomics*, pages 221–232. Springer, 2019.

Mazen Salous and Felix Putze. Behaviour-based working memory capacity classification using recurrent neural networks. In *ESANN 2018 – 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 159–164, Brugge, Belgium, 2018. ISBN 978-2-87587-047-6. URL `https://www.csl.uni-bremen.de/cms/images/documents/publications/salous_putze_2018esann.pdf`.

Mazen Salous, Felix Putze, Markus Ihrig, and Tanja Schultz. Visual and memory-based hci obstacles: Behaviour-based detection and user interface adaptations analysis. In *IEEE International Conference on Systems, Man, and Cybernetics*, Bari, Italy, 2019. URL `https://www.csl.uni-bremen.de/cms/images/documents/publications/salous_putze_SMC19.pdf`.

Sayan Sarcar, Jussi PP Jokinen, Antti Oulasvirta, Zhenxin Wang, Chaklam Silpasuwanchai, and Xiangshi Ren. Ability-based optimization of touchscreen interactions. *IEEE Pervasive Computing*, 17(1):15–26, 2018.

Tom Schaul and Jürgen Schmidhuber. Scalable Neural Networks for Board Games. In *Artificial Neural Networks – ICANN 2009*, pages 1005–1014. 2009. ISBN 978-3-642-04273-7 978-3-642-04274-4.

Tanja Schultz, Felix Putze, Timo Schulze, and Lars Steinert. I-care. *Zukunft der Pflege Tagungsband der 1. Clusterkonferenz 2018*, page 94, 2018.

Ahmed Seffah and Eduard Metzker. The obstacles and myths of usability and software engineering. *Communications of the ACM*, 47(12):71–76, 2004.

Bruno Sguerra and Pierre Jouvelot. An unscented hound for working memory and the cognitive adaptation of user interfaces. 2019.

Noam Siegelman, Louisa Bogaerts, and Ram Frost. What determines visual statistical learning performance? insights from information theory. *Cognitive Science*, 43(12):e12803, 2019.

Manuel Silverio-Fernández, Suresh Renukappa, and Subashini Suresh. What is a smart device?-a conceptualisation within the paradigm of the internet of things. *Visualization in Engineering*, 6(1):1–10, 2018.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Todd Andrew Stephenson. An introduction to bayesian network theory and usage. Technical report, IDIAP, 2000.

TFM Stewart. The role of hci standards in relation to the directive. *Displays*, 13(3):125–133, 1992.

Heinz-Martin Süß, Klaus Oberauer, Werner W Wittmann, Oliver Wilhelm, and Ralf Schulze. Working-memory capacity explains reasoning ability—and a little bit more. *Intelligence*, 30(3):261–288, 2002.

John Sweller, Paul Ayres, and Slava Kalyuga. Cognitive load theory, volume 1 of explorations in the learning sciences, instructional systems and performance technologies, 2011.

Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

Jochen Weiner, Mathis Engelbart, and Tanja Schultz. Manual and automatic transcriptions in dementia detection from speech. In *INTERSPEECH*, pages 3117–3121, 2017.

B. Whitworth and A. Ahmad. The social design of technical systems: Building technologies for communities. 2013.

Christopher D Wickens. Multiple resources and mental workload. *Human factors*, 50(3):449–455, 2008.

MD Wilson, PJ Barnard, TRG Green, and A MacLean. Knowledge-based task analysis for human-computer systems. In *Working with computers: Theory versus outcome*, pages 47–87. Academic Press, 1988.

Peter Wright, John McCarthy, and Lisa Meekison. Making sense of experience. In *Funology*, pages 43–53. Springer, 2003.

Fubao Zhu, Xiaonan Li, Haipeng Tang, Zhuo He, Chaoyang Zhang, Guang-Uei Hung, Pai-Yi Chiu, and Weihua Zhou. Machine learning for the preliminary diagnosis of dementia. *Scientific Programming*, 2020, 2020.

Publications

List of Publications

The following bibliography lists all the publications, in which we published the models and the experimental results of this thesis:

Felix Putze and Mazen Salous. Multimodal differentiation of obstacles in repeated adaptive human-computer interactions. In *26th International Conference on Intelligent User Interfaces*, IUI '21, Texas, USA, 2021. ACM.

Felix Putze, Mazen Salous, and Tanja Schultz. Detecting memory-based interaction obstacles with a recurrent neural model of user behavior. In *Proceedings of the 2018 International Conference on Intelligent User Interfaces*, IUI '18, pages 205–209, Tokyo, Japan, 2018. ACM. ISBN 978-1-4503-4945-1/18/03. doi: 10.1145/3172944.3173006. URL `https://www.csl.uni-bremen.de/cms/images/documents/publications/putze_salous_2018iui.pdf`.

Mazen Salous and Felix Putze. Behaviour-based working memory capacity classification using recurrent neural networks. In *ESANN 2018 – 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 159–164, Brugge, Belgium, 2018. ISBN 978-2-87587-047-6. URL `https://www.csl.uni-bremen.de/cms/images/documents/publications/salous_putze_2018esann.pdf`.

Mazen Salous, Felix Putze, Tanja Schultz, Jutta Hild, and Jürgen Beyerer. Investigating static and sequential models for intervention-free selection using multimodal data of eeg and eye tracking. In *Proceedings of the Workshop on Modeling Cognitive Processes from Multimodal Data*, pages 1–6, 2018.

Mazen Salous, Felix Putze, Markus Ihrig, and Tanja Schultz. Visual and memory-based hci obstacles: Behaviour-based detection and user interface adaptations analysis. In *IEEE International Conference on Systems, Man,*

and Cybernetics, Bari, Italy, 2019. URL https://www.csl.uni-bremen.de/cms/images/documents/publications/salous_putze_SMC19.pdf.

Supervision

List of Supervisions

The following list shows the bachelor thesis works which we supervised during the PhD period:

1. **Title:** Erkennung und Unterscheidung von gedächtnis-basierten und visuellen Interaktions-Hindernissen mittels EEG.
   Prepared by: Annita Klassen.
   Year: 2018.

2. **Title:** Recognising and dynamically adapting to visual and memory-based HCI obstacles based on behavioural data.
   Prepared by: Rafael Miranda.
   Year: 2018.

3. **Title:** Analyse und Vergleich von Benutzerverhalten und Gehirnaktivität bei verschiedenen Versionen eines Memory-Spiels zum Test adaptiver Hilfestellungen.
   Prepared by: Markus Ihrig.
   Year: 2019.

4. **Title:** Behaviour-based detection of visual interaction obstacles with 1D and 2D Convolutional Neural Networks.
   Prepared by: Anthony Mendil.
   Year: 2020.