PRESTK: Situation-Aware Presentation of Messages and Infotainment Content for Drivers

Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultät I der Universität des Saarlandes

vorgelegt von
Christoph Endres

Saarbrücken,
20. November 2012

Datum des Kolloquiums:
19. Dezember 2012

Dekan:
Prof. Dr. Mark Groves

Vorsitzender der Prüfungskommision:
Prof. Dr. Antonio Krüger

Gutachter:
Prof. Dr. Dr. h.c. mult. Wolfgang Wahlster
Prof. Dr. Stefan Panzer
Prof. Dr. Albrecht Schmidt

Akademischer Beisitzer:
Dr.-Ing. Jörg Baus

Eidesstattliche Versicherung[1]

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.
Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, den 20. November 2012

Christoph Endres

[1]Gemäß *Dienstblatt der Hochschulen des Saarlandes Nr. 21*, ausgegeben zu Saarbrücken 19.Juni 2007, Promotionsordnung der Naturwissenschaftlich-Technischen Fakultäten der Universität des Saarlandes, vom 11. Januar 2007, S. 254ff., Anlage 1.

Danksagung

Das Verfassen einer Dissertation funktioniert nicht ohne die Hilfe und Unterstützung anderer.

Gemäß der alten Volksweise, die besagt, dass der Müller bei Tag und bei Nacht stets wach ist, waren auch bei *Christian Müller* die Gelegenheiten für Besprechungen, Ratschläge und Brainstroming selten an Bürozeiten oder -räumlichkeiten gebunden. Ein solcher Einsatz ist nicht selbstverständlich und verdient besonderen Dank. Er war es auch, der im Januar 2010 daran geglaubt hat, dass ich es schaffen kann, in der vorgesehenen Zeit von drei Jahren eine Dissertation zu schreiben. Ich freue mich sehr, dass er damit recht behalten hat.

Prof. Wolfgang Wahlster hat sich insbesondere in der Endphase der Arbeit viel Zeit genommen, um wertvolle Anmerkungen zu geben. Auch hier ist es nicht selbstverständlich, dass ein so vielbeschäftigter Mann mehrstündige Betreuungstermine am Wochenende ermöglicht. Auch dafür Danke!

Die beiden weiteren Gutachter, *Prof. Stefan Panzer* und *Prof. Albrecht Schmidt*, unterstützen meinen knappen Zeitplan mit besonderem Einsatz. Danke!

Beim Erstellen der Demonstratoren zu meiner Arbeit haben meine Kollegen *Mehdi Moniri*, *Sandro Castronovo*, *Rafael Math*, *Michael Feld* und *Daniel Braun* viel Arbeit und auch Freizeit investiert. Generell hat mir die Arbeit mit den Kollegen in der Automotive-Gruppe viel Spaß gemacht. Danke für die schöne Zeit!

Martin Buchheit war trotz seines schillernden Lebenswandels im Dunstkreis zwischen saarländischer Rocklegende und angehendem Ordensmann immer ein kompetenter und interessierter Ansprechpartner im Bereich Mathematik. Danke!

Mangels Englischunterricht am altsprachlichen Gymnasium sind meine Englischkenntnisse mehr von Bauchgefühl als von Grammatikkenntnissen geprägt. Ich danke daher besonders auch *Katrin Reis* und *Alexis Palmer* für’s Korrekturlesen.

Susanne van Muhlken hat mich mit wertvollen Informationen über ihre Dissertation zu einem ähnlichen Thema versorgt. Auch dafür vielen Dank!

Insgesamt habe ich seit 1994 am DFKI und am Lehrstuhl von Prof. Wahlster viele nette Kollegen gehabt. Zu viele, um sie namentlich zu nennen, aber sie sind nicht vergessen.

Insbesondere danken möchte ich auch meiner Familie, die etwas länger auf die Beendigung meines Studiums warten mussten: Meinen Eltern *Paul* und *Maria*, meiner Frau *Catrin* und unserer Tochter *Juliana Marleen*. Ich verspreche, dass der Papa in Zukunft weniger Wochenenden im Büro verbringen wird.

Die Faszination für Wissenschaft und Technik begleitet mich seit meiner Kindheit. Ich verdanke sie einer vielleicht etwas seltsam anmutenden Ansammlung realer und fiktiver Vorbilder, die sie auch heute noch aufrechterhalten:

Tobias Findteisen, *Albert Einstein*, *Karl Benz*, *Nikola Tesla*, *Konrad Zuse*, *Alan Turing*, *Emmet Brown*, *Wau Holland*, *George Boole*, *Évariste Galois*, *Zefram Cochran*, *Kurt Gödel*, *Arthur C. Clarke*, *Carl Friedrich Gauß*, *Shawn Fanning*, *Susan Calvin*, *Linus Torvalds*, *Bill Gosling*, *Isaac Newton*, *Léon Theremin*, *Sally Ride*, *Sheldon Cooper* und insbesondere *Isaac Asimov*.

Danke!

Short Abstract

The amount of in-car information systems has dramatically increased over the last few years. These potentially mutually independent information systems presenting information to the driver increase the risk of driver distraction. In a first step, orchestrating these information systems using techniques from scheduling and presentation planning avoid conflicts when competing for scarce resources such as screen space. In a second step, the cognitive capacity of the driver as another scarce resource has to be considered.

For the first step, an algorithm fulfilling the requirements of this situation is presented and evaluated. For the second step, I define the concept of *System Situation Awareness (SSA)* as an extension of Endsley's *Situation Awareness (SA)* model. I claim that not only the driver needs to know what is happening in his environment, but also the system, e.g., the car. In order to achieve SSA, two paths of research have to be followed:
(1) Assessment of cognitive load of the driver in an unobtrusive way. I propose to estimate this value using a model based on environmental data.
(2) Developing model of cognitive complexity induced by messages presented by the system.

Three experiments support the claims I make in my conceptual contribution to this field. A prototypical implementation of the situation-aware presentation management toolkit PRESTK is presented and shown in two demonstrators.

Kurzzusammenfassung

In den letzten Jahren hat die Menge der informationsanzeigenden Systeme im Auto drastisch zugenommen. Da sie potenziell unabhängig voneinander ablaufen, erhöhen sie die Gefahr, die Aufmerksamkeit des Fahrers abzulenken. Konflikte entstehen, wenn zwei oder mehr Systeme zeitgleich auf limitierte Ressourcen wie z. B. den Bildschirmplatz zugreifen. Ein erster Schritt, diese Konflikte zu vermeiden, ist die Orchestrierung dieser Systeme mittels Techniken aus dem Bereich Scheduling und Präsentationsplanung. In einem zweiten Schritt sollte die kognitive Kapazität des Fahrers als ebenfalls limitierte Ressource berücksichtigt werden.

Der Algorithmus, den ich zu Schritt 1 vorstelle und evaluiere, erfüllt alle diese Anforderungen.
Zu Schritt 2 definiere ich das Konzept *System Situation Awareness (SSA)*, basierend auf Endsley's Konzept der *Situation Awareness (SA)*. Dadurch wird erreicht, dass nicht nur der Fahrer sich seiner Umgebung bewusst ist, sondern auch das System (d.h. das Auto). Zu diesem Zweck müssen zwei Bereiche untersucht werden:
(1) Die kognitive Belastbarkeit des Fahrers unaufdringlich ermitteln. Dazu schlage ich ein Modell vor, das auf Umgebungsinformationen basiert.
(2) Ein weiteres Modell soll die Komplexität der präsentierten Informationen bestimmen.

Drei Experimente stützen die Behauptungen in meinem konzeptuellen Beitrag. Ein Prototyp des situationsbewussten Präsentationsmanagement-Toolkits PresTK wird vorgestellt und in zwei Demonstratoren gezeigt.

Contents

List of Figures

List of Tables

“...und ich, selbst ich komme aus dem Staunen nicht raus, dass es fast keine Fortbewegung mehr ohne Computer gibt. Fußgänger, Radfahrer nehmen wir mal aus, bis auf weiteres, aber sonst kommt doch kein Mensch mehr von Ort zu Ort ohne Computer, ob auf der Straße, der Schiene oder in der Luft ...
Für Sie ist das banal, aber verstehen Sie, was da in mir vorgeht?” [Del11]

— (Konrad Zuse, 1910-1995)

Part I

Introduction

Chapter 1

Introduction and Motivation

1.1 A Historical Perspective

I would like to start this thesis with two anecdotes from the history of motorized vehicles.
On August 17th, 1896, the 44 year old Bridget Driscoll (cf. figure 1.1), accompanied by her teenage daughter May, crossed the grounds of the Crystal Palace in London and was struck by a Roger-Benz car belonging to a motoring exhibition. The car was capable of driving 8 mp/h (13 km/h) but was deliberatly limited to 4 mp/h (6.4 km/h). Eye-witness Florence Ashmore desribed the car as going at "a tremendous pace". The driver of the car, Arthur Edsell, rung the bell of his car and shouted, but Driscoll just stood there "bewildered" on the road uncertain what to do. The driver, who back then of course did not have a driver's license yet, had no training how to react in such a situation or any education on which side of the road to drive. As a result, Bridget Driscoll became the first fatal pedestrian victim of a car accident. The coroner Percy Morrison examind the case. There was no prosecution, but Morrison said he hoped that "such a thing would never happen again"[1].
But not only the hope for reducing car accidents has been around since the beginning of automotive driving. In 1930s, Nicholas Trott wrote in the New York Times "A grave problem that developed in New Hampshire, spread to Massachusetts, and crept over to Albany, now has all the motor-vehicle commissioners of the eastern states in a wax. It's whether radios should be allowed on cars. Some states don't want to permit them at all-say they distract the driver and disturb the peace. The manufacturers claim that the sound of Rudy Vallee's voice is less disturbing than backseat conversation.
Massachusetts leans toward the middle of the road. The commissioner there thinks the things should be shut off while you are driving, but that you should be allowed to take culture with you into the wilderness. The whole problem is getting very complex, but the upshot is that you'll probably be allowed to take your radio anywhere, with

[1] www.bbc.co.uk/news/magazine-10987606

Figure 1.1: Bridget Driscoll (†08/17/1896), first pedestrian victim of a car accident.

possibly some restriction on the times when you can play it."[2]
By now, the use of radio and other entertainment systems in the car is legally possible, despite the increasing complexity of their user interfaces which changed dramatically since the 1930ies. Also, driver distraction and prevention of car accidents (especially lethal car accidents) is still a growing concern.
In this thesis, I describe my concept of a situation-aware car as a small contribution towards the overall goal of decreasing driver distraction and thus reduce casualties.

1.2 Short Problem Description

Complex information systems require carefully designed Human Machine Interfaces (HMIs) for intelligently providing their operator with the right information at the right moment and for a sufficiently long period of time for him to fully process the content of the message. At the same time, the information system should be unintrusive and not overwhelm the user with too much information. In a nutshell: timing, modality, and awareness of situation complexity each play an important role. I propose a combination of research in the respective established research areas of scheduling, presentation planning, and human factors as a foundation for designing intuitive complex systems.
The results of this research effort are presented in the context of the automotive domain, which is a suitable test area due to its highly dynamic user environment and

[2] www.drivers.com/article/351/

increasing complexity of information systems. This should however not be seen as a limitation of the generality of the question answered in this thesis, which can be applied to any domain with similar demands.

1.3 Intelligent Mediation

Ovans and Havens coined the term “Intelligent Mediation” to solve the problem of presenting information in real-time on limited resources with consideration to the limited attention of the operator: “Operator interfaces to supervisory-control systems are often highly complex, cumbersome to extract information from, and overwhelmingly verbose in the face of abnormal operating conditions” [OH93]. The authors propose replacing the conventional operator interface with an intelligent interface. The aim is to provide mediation of the control system output and present the operator with “intelligently formatted information”. They raise the question of how to design such a system, as it is rather complex, requiring the real-time allocation of limited interface resources. As a solution, the authors propose to use an expert system architecture and methodology.

1.4 From Human Machine Interface (HMI) to Driver Vehicle Interface (DVI)

The term *Human Machine Interface (HMI)* is traditionally used to describe the space where interaction between a user/operator and a machine takes place. In computer science, this encompasses typically classic command line interfaces (CLI), graphical user interfaces (GUI), and tangible user interfaces (TUI).
For the communication between a driver and in-vehicle systems, the term HMI is not quite appropriate anymore due to several reasons. The term machine has a connotation of a worker operating large, stationary machinery in a factory or plant. The complexity of a modern, distributed user interface in which the user is both requesting services from the system as well as providing information to the system explicitly as well as by unobtrusive detection is by far higher and more sophisticated.
A solution to the increasing dissatisfaction with this term in the automotive community was offered at the AutomotiveUI’2012 conference where James Foley of Toyota coined and fostered the term *Driver Vehicle Interface (DVI)*, which immedeately caught on. Although the word *interface* falsely has an undertone of a single point of connection instead of a complex interaction environment, the term DVI is by far more appropriate than HMI.
In this thesis, I will use the terms HMI and DVI synonymously, since the former has

been widely used in literature for a long time already and cannot simply be relaced everywhere, but I am referring to the more advanced concept described here.

1.5 Challenges in Automotive Information Presentation

The traditional automotive user interface was relatively simple, lever-, knob-, and dial-based, and stayed mostly unchanged for many decades after its invention. Due to the massive increase in the availability of computerized systems over the past two decades, more and more information and communication systems have been included, providing additional information services related to both driving and navigation, as well as to in-car entertainment. We can expect the upcoming availability of Car-2-Car and Car-2-X communication to further boost this trend.
At the same time, the car as an intelligent environment imposes two resource restrictions on information systems. On one hand, space and availability of information channels are limited. For instance, screen space in the view field of the driver is limited, the auditory channel is not suitable for providing more than one piece of information at a time, etc. The second resource limitation is the driver himself: As a human being, his cognitive capacities are limited (to a varying degree among different subjects), and this limitation needs to be considered as well. Figure 1.2 shows the dually restricted data processing flow in complex information systems.
[WT97] define three categories of resource-sensitive processes: (1) *resource-adapted* processes are processes optimized for limited use of resources; (2) *resource-adaptive* processes offer a fixed processing strategy, given the amount of available resources as a parameter; and (3) *resource-adapting* processes dynamically adapt at runtime to the available and potentially changing amount of resources (cf. [CS10]). For the problem discussed here, we aim at a solution between resource-adaptive and resource-adapting.
Due to the usual mutual independence of information systems, two or more applications might simultaneously try to access the same limited communication channel, or the overall combined complexity of presented information might overwhelm and distract the driver. Not displaying important information would be a critical decision as well. A carefully balanced approach for this problem is the challenge we are facing here. With increasing instrumentation, the problem will not only affect the driver, but also the front- and back-seat passengers. The proposed solution can be used with minor modifications for all three parties.
However, the focus of this thesis is on the driver as the person with the highest impact on passenger safety.

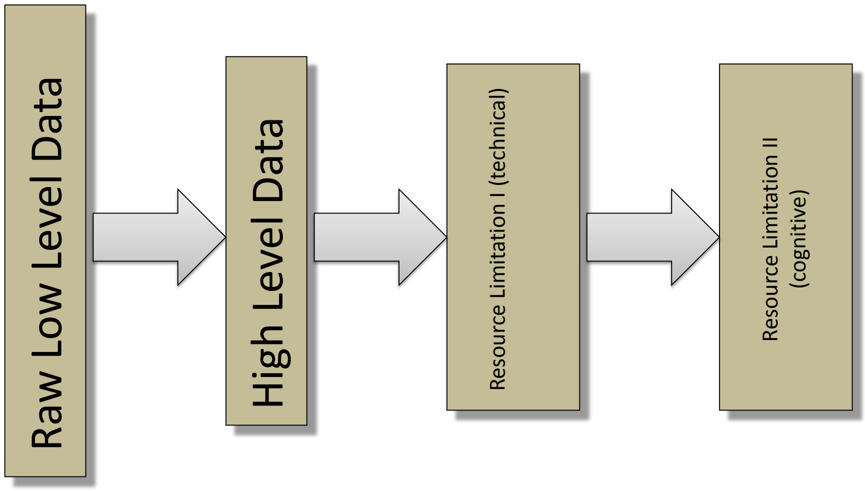


Figure 1.2: Dual resource restriction in complex information systems.

1.6 sim^{TD}

The project sim^{TD} provides the background for this thesis. It was intended as a proof-of-concept for communicating car technologies. In a joint effort between the automotive industry, the state of Hessen and several academic institutes, results from automotive research in the past decades have been put to practical application and tested extensively.
A large-scale field test in the area of Frankfurt (Germany) in the second half of 2012 provides experience about the practical use of communicating cars technology: 120 cars collect data from 40.000 hours of driving "in real life".
At DFKI, we worked on the Driver Vehicle Interface, and especially on orchestrating incoming information. In this thesis, I cover my work in this project as well as extensions based on lessons learned along the way. The project sim^{TD} is described in more detail in chapter 4.

1.7 Objective

In the sim^{TD} project, our focus was on providing a solution which considers both the temporal aspects of information presentation (scheduling) as well as the way in which information is presented (presentation planning). The requirements of the project could successfully be fulfilled, but experience during the project and trends in recent literature implied that such a system should as well consider the current cognitive state of the driver and adapt to it. This lead to the definition of the *System Situation Awareness* concept. The aim of this thesis is to explore in-car presentation management in the intersection of the research areas of scheduling, presentation planning and situation awareness. It extends previous work and incorporates lessons learned from work in the sim^{TD} project (see Figure 1.3).

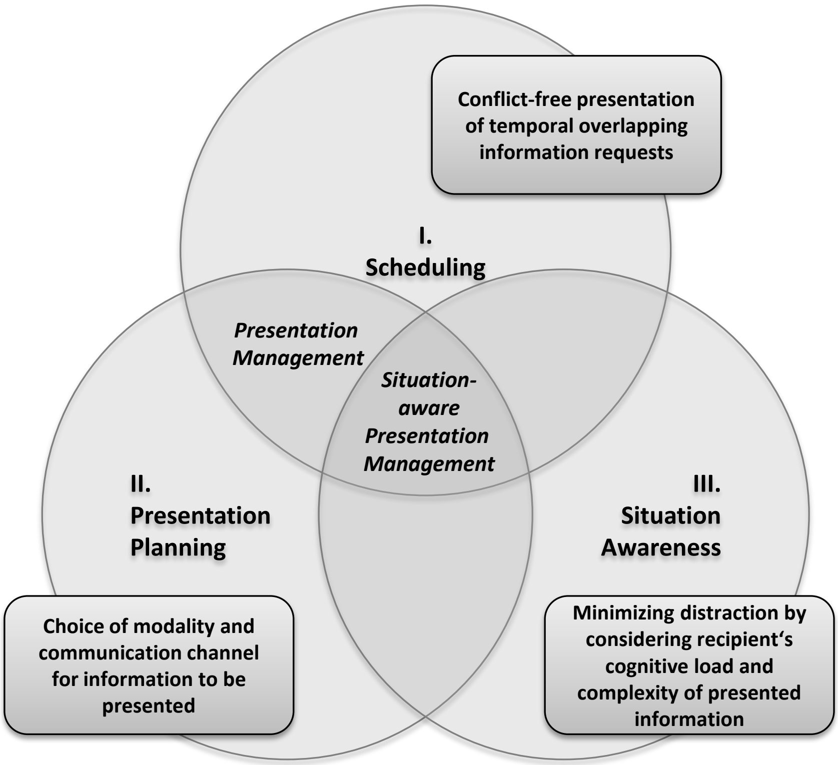


Figure 1.3: Topics of this thesis.

1.8 Key Research Questions

My research is guided by the following questions:
(1) What is the difference between cars and conventional ubiquitous intelligent environments and how can the specific challenges with respect to presentation planning arising from those differences be taken into account?
(2) To what extent are traditional presentation planning mechanisms applicable to the domain and which changes have to be considered?
(3) How can the cognitive complexity of a presentation be estimated and used for presentation planning with context assessment?
(4) How can an explicit priority management be developed that takes into account both the driving situation and the cognitive limitations of the driver?
(5) How can we define situation awareness on a system level?
(6) Which are the core components of a flexibly applicable system for situation-aware information presentation and how can they be combined into a comprehensive toolkit?

Part II

Theoretical Foundation

Chapter 2

Underlying Concepts in Computer Science, Artificial Intelligence, and the chosen Domain

The underlying concepts of this thesis are manifold, especially due to its interdisciplinary nature. In this chapter, I will briefly introduce some of the concepts stemming from computer science, artificial intelligence, and the chosen domain of automotive research. Presentation planning and situation awareness will be discussed in separate chapters.

2.1 Intelligent Environments

In 1991, Mark Weiser wrote a landmark paper [Wei91] about his vision of "Ubiquitous Computing" (ubicomp), which fostered the development of a whole research area. Names and definitions for this paradigm, which is sometimes also called *pervasive computing* or *ambient intelligence* vary, but the underlying idea remains: A post-desktop model of human-computer interaction, in which everyday objects take care of information processing in an intuitive way. In the ideal implementation of ubicomp, the user will not even be aware anymore of using technology. Weiser describes this concept in his own words:

> "Ubiquitous computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives." (Mark Weiser, 1952–1999)

weiser proposes three different granularities of ubicomp devices: *tabs* (small devices), *pads* (hand-held devices) and *boards* (interactive display devices). In the early nineties, this may have sounded like a science fiction fantasy to most people, but over the last two decades, a lot of Weiser's vision has been realized. Smartphones, tablet computers, and public information kiosks have become a part of our everyday life.

A closely related concept is *intelligent environments*, interactive spaces with embedded systems that bring computation into the physical world. As with ubicomp, terminology for this concept differs. In [EBM05], I distinguished between intelligent environments, augmented reality, and distributed mobile systems, and argued that the border between those concepts is not always clear. However, the principle can be found in an increasing variety of installations nowadays, and in a variety of contexts. One popular research trend is, for instance, *ambient assisted living*, support for elderly people to stay independent for a longer time using assisting technology [Ale08].

At the same time, the description of an intelligent environment matches pretty closely the interior of modern premium cars: A lot of assistance systems are integrated ambiently in the driver's or passenger's environment, and the computing power hidden "under the hood" increases constantly. The DFKI project SmartWeb [Wah04] provided groundbreaking work in including an intelligent dialog system in a car.

In consequence, I claim that cars should be considered intelligent environments as well [EC10, EMM11], and research efforts should be combined in all involved research areas to support this common goal (cf. figure 2.1).

Meanwhile, the automotive industry has taken an ad-hoc approach toward building in-car interfaces. Internationally-recognized standards are few; "best practices" dominate instead. At the same time, the introduction of Car-2-Car and Car-2-X communication as means for wireless communication between cars, or between a car and the environment, respectively, widens the scope for in-car applications and assistance systems. Furthermore, Car-2-X communication is, unlike Car-2-Car communication, not limited to WLAN-based ad-hoc communication over short distances. The increase in connectivity with high-speed communication technologies such as UMTS and LTE connects the intelligent environment car to other intelligent environemnts in the life of the driver, such as the instrumented home, the instrumented office, or the social environment in form of social networks:

On the way to work in the morning, the driver can be presented with information about his schedule, tasks, and meetings for the day. Some of his tasks might require additional information or preparation, such as reading additional documents for instance. Upon request by the driver, the car can connect to the office and trigger the print-out of necessary documents, so that they are readily available at his arrival at the office.

But the added connectivity can not only be used for increasing productivity; streamlining the entertainment program for the evening is possible as well. The GNAB[1] system for instance already offered download of music and videos from a mobile device to the

[1] http://www.golem.de/0503/37084.html

entertainment system at home. The driver could select and request a movie he wishes to watch at home, and the download starts immedeatly and is availabe upon arrival at his home.
Also, social networks play an increasing role in the life of a growing number of people. The urge to stay connected unfortunately sometimes leads to dangerous behaviour, such as texting with a mobile device while driving. In previous research [EBM11], we described and prototyped means to assist the driver here as well and provide more safety. This approach was extended by determining the context of the user in order to offer better assistance for the tasks at hand [BEM11].
Combining the research in driver assistance systems, multimodal interfaces, Car-2-X communication and intelligent environments provides a powerful foundation.
With this thesis, I intend to realize a part of this interdisciplinary work.

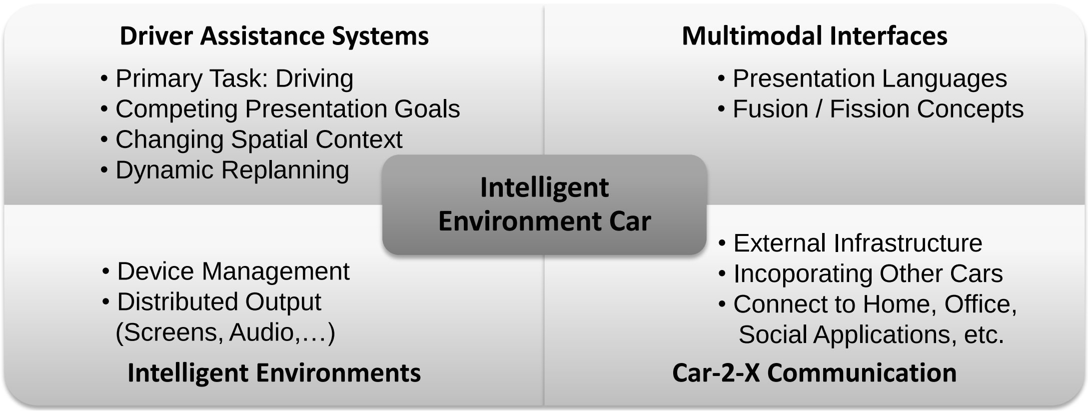


Figure 2.1: The car as an intelligent environment.

2.2 Automotive Research

For over a century now, the invention and development of the car has been inspiring many companies, researchers, and most of all a large percentage of its immense and increasing base of users. It is not surprising that a lot of research is centered around this domain, ranging from safety aspects over performance and design considerations up to details such as driver seat ergonomics. In my research, I work on the driver-vehicle interface and the situational awareness of the machine. Figure 2.2 summarizes the requirements for an advanced driver assistance system (ADAS).
Especially in the sim^{TD} project, we build on previous research results in this area and center our work around a practical evaluation of proposed innovations. This does not restrict the theoretical contribution in any way, as important questions arise in the combination of previous results, fostering additional research.
As an introduction to this area, I will give a short summary of important work done

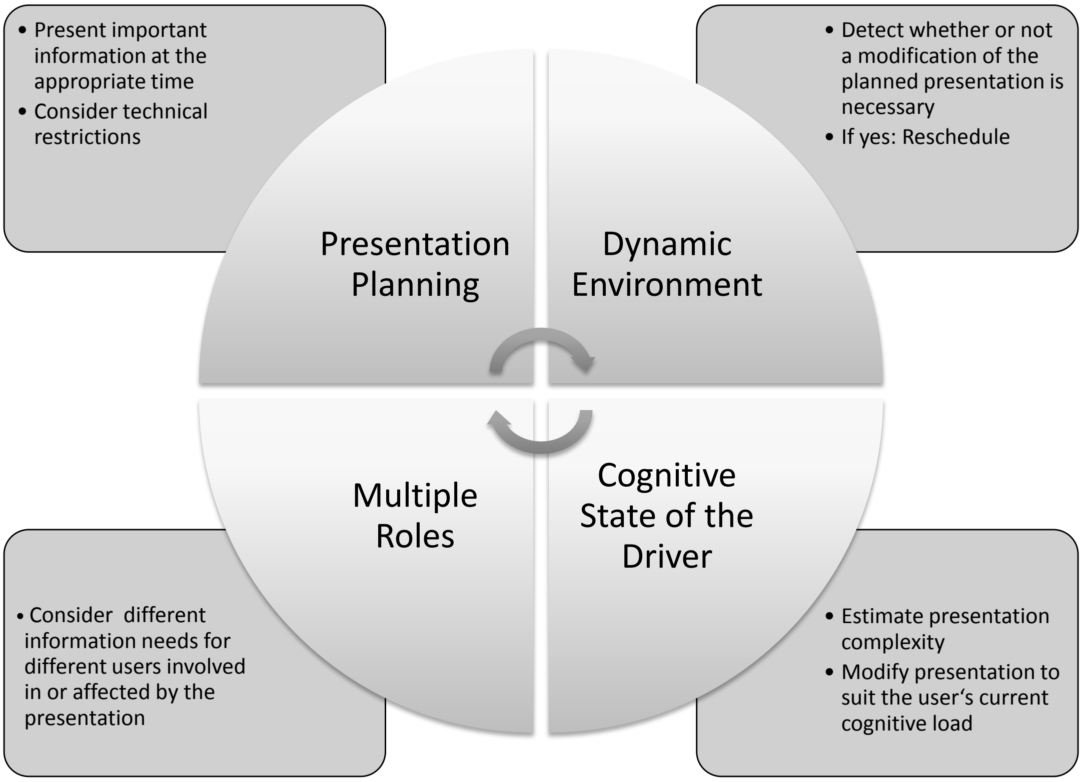


Figure 2.2: Requirements for an improved Advanced Driver Assistance System (ADAS).

in automotive research which was relevant for my work in the sim^{TD} project. Further and more recent literature in the automotive domain is discussed in the related work chapter 6.

[Nåb07] perform a user study to test the suitability of an in-car warning system, which is implemented in several variations. The user study examines the reliability of the warning message when no management system is active, with their rule-based DiWiTSA system active, and with the extended Super-DiWiTSA system active.
The rules for DiWiTSA are simple: When several active systems fire at the same time, only the one with the higher priority is displayed. No lower priority message interrupts a higher priority message, whereas a higher priority will interrupt a lower priority. The same warning cannot be delivered twice in a given time frame. The Super-DiWiTSA system is basically identical, with the extension of one more rule: If a higher priority information will soon reach a critical value, a lower priority warning will be suppressed and the one with higher priority issued instead. The overall results of the study were favorable for the system, and the participants displayed a positive attitude towards the use of such a system: Warnings did produce the appropriate and desired driver responses, i.e., braking for Forward Collision Alert (FCA), steering for Lane Departure Warning (LDW), and looking back on the road for Distraction Warn-

ing (AttenD). The authors found no evidence that warning management influences the reaction and response in comparison to no management in their setup: "A possible explanation of the driver behaviour could be that drivers react on the warning by looking back to the road, but respond based on their own assessment of the scenario (and not on the interpretation of the warning)." The authors recommend to adapt the warning to threat level, so that more crucial warnings are experienced as more relevant by the driver.

[MMFM07] report on the WATCHOVER project which aims at the enhancement of road safety. Consideration of vulnerable road users (VRU) such as pedestrians and bicyclists is of special interest in that project. In this paper, a user requirements survey is described and a resulting system design is presented. The system is based on communication modules enabling the information and warning exchange between cars and VRUs. We extended this approach in our research, e.g., in [CEF+11].

[BEAM06] describes the Volvo concept of Intelligent Driver Information Systems (IDIS). The Interaction Manager (IM) component is introduced. The purpose of the IM is to resolve conflicts between different systems, or between systems and the current driving situation. The envisioned key functions are:

- Allocation management of I/O resources
- Application adaptation
- Conflict resolution
  - between applications with respect to their interaction with the driver
  - between system initiated events and secondary tasks performed by the driver
  - between applications and the driving situation

The main component of the IM is a Driver Vehicle Environment Monitor (DVEM) which is connected to an Application Coordinator (AC). A prototype of the system has been implemented as proof-of-concept. No user study is presented; testing is mentioned as "further possibility" in the conclusion.

[AAD+05] introduce the "intelligent core" principle of next generation driver vehicle interaction systems, which is part of the COMMUNICAR project. The aim is to facilitate the connection between the increasing number of on-board information systems and the user's personal information systems. At the same time, driver distraction should be minimized. The proposed information manager works like an expert system with rule-based mechanisms implemented by human experts. Three classes of rules are distinguished: "total level of risk (TLR)"-rules, priority rules, and output modality rules. The development of a continuous interaction flow between driver and vehicle is envisioned. To a certain extent, the envisioned system considers situation awareness, e.g., by not only evaluating how the driver is driving at the moment, but for how long he has been driving already. Some interesting issues are discussed:

- Multiple use of a single device by reconfiguration, resulting in functions only being available when needed.
- Configurability to the context and the driver's preferences and driving style.
- Safe use of nomadic devices.
- Seamless connectivity to the driver's personal information.

The overall goals of the COMMUNICAR project are similar to the ones followed in the sim^{TD} project.

[APAB06] discuss the communication and interaction strategies in the AIDE [GSB$^+$04] and COMMUNICAR projects. Their research is centered around the following five issues: (1) simultaneous use of one or more systems, (2) harmonizing a huge volume of messages, (3) guaranteeing driver and vehicle safety, (4) optimizing interaction between driver and vehicle, and (5) reducing the number and severity of accidents while promoting mobility. The underlying system is the rule-based system already discussed in the previous section. The extensions of the AIDE system over the previous COMMUNICAR system include: addition of multimodal interaction technologies, more intuitive use, simplification of the cockpit (extending the "reusing device" approach), consideration of context, reconfiguration for different driver preferences and characteristics.

[Kos04] describes a local danger warning system based on vehicle ad-hoc networks. The communication between the cars is based on single-hop dedicated short-range communication (DSR). If the system is activated, the car listens for hazard messages, conditionally forwards them and checks whether they are relevant to the current driving situation. If yes, the danger is shown as an icon on the screen. The system described here and tested in a simulation was later refined and put to use in the sim^{TD} project.

2.3 Anytime Algorithms

An *anytime algorithm* (sometimes also referred to as *"interruptable algorithm"*) is commonly described as an algorithm that can return a valid solution to a problem even if it is interrupted before the end of its runtime [Zil96]. This is the main characteristic distinguishing it from normal algorithms which run until the completion of

their runtime. The quality of the solution returned increases with more time given to the algorithm. Reasons for using an anytime algorithm usually stem from scarce resources such as computing power and/or time available until a solution is needed. A performance profile is used to estimate the quality of the results depending on the input and the amount of time available to the algorithm. The quality here can be defined in different ways, usually in terms of certainty (probability of correctness), accuracy (bounds of error), and specifity (amount of particulars).
The term anytime algorithm was coined by [DB88], who applied it to path finding problems in time-dependent planning.

2.4 Scheduling

In this thesis, I will introduce a problem which is a variation of the Resource-constrained Project Scheduling Problem (RCPSP). The problem is defined as **Resource-constrained Scheduling Problem (RCSP)**. Particularly, we apply the approach to the problem of scheduling a large number of driver warnings based on Car-2-Car communication (also known as *cooperative vehicles*).
The resource-constrained project scheduling problem (RCPSP) is a combinatorial optimization problem [BLRK83]. Due to its high practical importance, it has been analyzed for several decades now.
[ADN08] describes RCPSP as considering "resources of limited availability and activities of known durations and resource requests, linked by precedence relations". An example is given in figure 2.3: Different jobs with temporal dependencies need to be scheduled on a limited number of machines and processed in the right order in a timespan as minimal as possible [ADN08]
RCPSP is an extension of the job shop scheduling problem (JSP). [GJ79] provides the following definition, which is used as a basis here.

Definition 1 (JSP). *The job shop scheduling problem (JSP) consists of a set $\mathcal{A}$ of activities each with a positive integer-valued duration, d_i. $\mathcal{A}$ is partitioned into projects (jobs), and with each project is associated a total ordering on that set of activities. Each activity specifies a resource on which it must execute without interruption. No activities that require the same resource can overlap in their execution.*

Finding a solution to this problem with minimal makespan (difference between minimum start time and maximum end time) is NP-hard. RCPSP is more complicated than JSP, since each task requires not only a processor but also additional scarce resources. Since it can be reduced to the simpler JSP, it is NP-hard as well [GJ79, BLRK83].
I introduce the resource-constrained scheduling problem (RCSP), which is similar but adds additional constraints to start- and endtime of activities. Unlike the original RCPSP, it has a dynamic component from disruption management. [CLLH01] defines

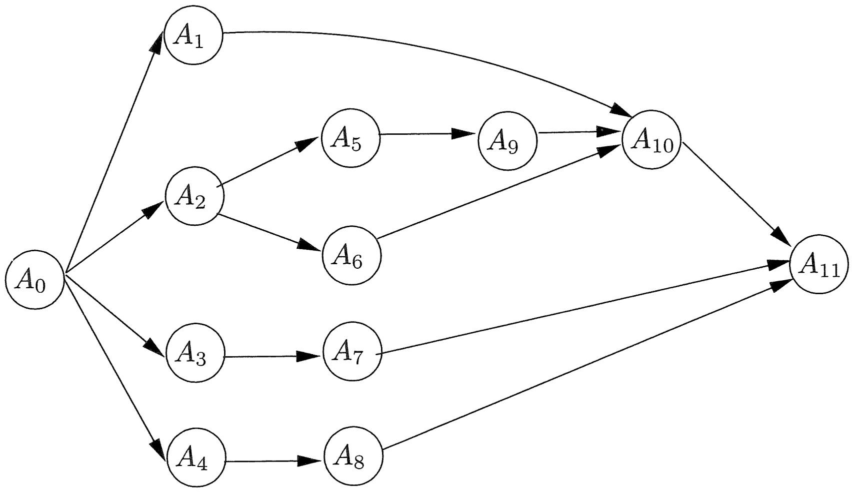


Figure 2.3: An example for the resource-constrained project scheduling problem (RCPSP).

a disruption as "a state during the execution of the current operation, where the deviation from plan is sufficiently large that the plan has to be changed substantially." In a similar manner, [KJF07] defines disruption management (DM) as "the process of responding to an unforeseen disturbance occurring during the execution of planned and scheduled operations.". According to the author, DM aims at the selection of appropriate repair actions to minimize the negative impact typically associated with disruption.

In the automotive domain, particularly in the field of Advanced Driver Assistance Systems (ADAS), we face the challenge of scheduling a growing number of assistance systems trying to communicate simultaneously with the driver over a limited amount of communication channels. This differs from RCPSP in that scheduling priority lies not on the order of activities but on preserving the requested presentation times as closely as possible. Moreover, following [BBM07], we do not need only to be able to distinguish between plans that satisfy goals and those that do not without providing further means of discrimination between successful plans. For the application at hand, we also need to have information about how "good" a plan is, thus "enabling the planner to distinguish between successful plans of differing quality" (ibid.). Furthermore, the nature of the application demands for an anytime behavior. The desired algorithm to solve the problem should be able to output a solution at any time, which should be as close as possible to the optimal solution.

For RCPSP, several algorithms have been proposed, reaching back to the 1960's with a Branch-and-Bound approach by [Joh67], adopted later by [SDK78, CAVT87, BKST98]. [PWW69, PR76] proposed Zero-One programming solutions, while more recent approaches use Linear [BK00], Constraint [DAM05] and Genetic Programming [KJF07].

The Branch-and-Bound approach is closest to the solution I present. It is based on graph search and effective bounding (e.g., pruning). Linear, Integer and Zero-One-Programming are variations of a standard optimization problem. Genetic programming is a search heuristic which mimics natural selection behavior.

It is important to keep in mind that the RCPSP and my RCSP are similar, but not interchangeable, e.g., solutions for the original problem are not applicable to the newly defined problem due to the different nature of the problems.

For solving RCSP in the given highly dynamic domain, we need an anytime algorithm which finds a good (not necessarily perfect) solution very quickly. Both tree search and genetic algorithms fulfill these requirements. In this thesis, I present a tree search approach with effective pruning.

Chapter 3

Presentation Planning

Presentation planning, i.e., the *automation* of Intelligent Multimedia Presentation Planning (IMMP), is a traditional research field in the area of artificial intelligence.
[BFF$^+$97] emphasizes that "multimedia presentation design is not just merging output fragments, but requires a fine grained coordination of communication media and modalities." Furthermore, the authors state that "a presentation system should be able to flexibly generate various presentations for one and the same information content in order to meet individual requirements of users and situations, resource limitations of the computing system, and so forth."
[vM99] summarizes the requirements of an IMMP in five issues:

1. **Content selection** *Which pieces of information* are to be presented? Are there dependencies between them, and, if so, which piece of information has to be presented before which other piece of information, i.e., how can we linearize the information to be presented?

2. **Media selection** *How* should the selected information be presented? Several constraints have to be taken into consideration, both from the system's perspective and from the recipients perspective.

3. **Medium specific generation** *How* should the selected information be rendered for the selected output medium?

4. **Cross references** For a coherent presentation, presented information in different media or modalities should relate to each other, either implicitly or explicitly (cf. [ES10]). References to previously presented information also support the recipient in understanding the presentation.

5. **Layout** *Where* should the information be presented? Who is the recipient, and where is his current focus of attention?

To establish a common ground for discussion, I start with describing and delimiting terms used in connection with presentation planning in accordance with and loosely following the definitions in [BFF+97]:

A *medium* is a physical space in which to realize perceptible entities. Considering the human sensory apparatus as recipient or "target system", the medium is a physical space in which perceptible entities are realized. Distinctions can be made between different perception channels (visible, audible, haptic, olfactory, gustatory) and further refined, e.g., monochrome or colored, two dimensional or three dimensional, etc.
The term *medium* can also refer to a type of information and/or a representation format, which is basically just another perspective for the same concept. Instead of describing the space of realization, the (nowadays usually digital) representation entity is used. The image bitmap represents the image on the physical screen, the audio file represents the audio output on the speaker, etc.
Multimedia refers to the use of multiple media in either sense of the previous definitions of media. The multimedia concept video-with-sound for instance can be composed of the media audio and video.

The *modality* of an piece of information refers to the particular way it is encoded, e.g., written or spoken language, 2D or 3D visualization, etc.
Multimodality refers to the use of multiple modalities to encode information, e.g., presenting numerical data both with a table as well as with a graph. It is noteworthy that one single medium can be used for different modalities, even simultaneously, for instance spoken text and background music on the same audio channel.

Both medium and modality are used to communicate information to a recipient. A *presentation* is a composition of media objects with the goal of conveying information to the user of the system.

3.1 Multimodal Presentation Description Languages

In this section, I present description and classification of several multimodal presentation description languages in a broader sense. The detailed classification can be found in appendix A. The scope encompasses document, data and process markup languages, graphic description languages and interaction definition languages that could be used for presenting information in different scopes and granularities. Figure 3.1 provides an overview and classification of the presented languages.
The aim of this survey is to take a look at existing approaches which will later in this document be used to specify requirements for an automotive presentation markup language compliant with the objectives of this thesis.

Classification Sheet

Each language is introduced with a short classification sheet covering the basic aspects of the language. The *year* denotes the beginning of the development, the first specification, and in some cases also the termination of development. The *domain* refers to the intended usage of the language, such as web page or slide presentation, for example. The *target platform* is the system which is generating, exchanging or interpreting the code. The term *origin* refers to the company, organization, or person developing or specifying the language. Since not all languages described here are on the same level, the *stage of development* is listed separately, ranging from concept over specification up to product or standard. Another important aspect is the *license* under which the language is published. *Compatibility* is another important factor for evaluating a language, as well as the available *tools*. The *language structure* is usually a tag-based markup language; most of the languages in this section are XML-based. It is beneficial if a language has a large *community* behind it and a certain *impact*. The latter is rather difficult to assess objectively; we used Google Trends and expert opinions as estimates. For different usages, different *media support* is necessary and thus a criterion for classification. The *web page* of the language provides, where available, further information.

Due to the wide range of languages discussed here, classification information beyond the basic information is not provided in a table but in text form. Example code is also provided, where available. Unless noted otherwise, example code is either taken from official documentation or self-written.

Due to the high impact of standardization, languages with W3C support are especially important, and marked in Figure 3.1.

Terms and Acronyms

Lua (Portuguese for “moon”, no acronym) is a cross-platform, lightweight multi-paradigm scripting language with “extensible semantics” as its primary goal. It was created 1993 by members of the Tecgraf group at the Pontifical Catholic University of Rio de Janeiro, Brazil and made available under MIT License. Lua and its dialect Metalua have become popular for lightweight scripting of interactions or animations in combination with markup languages such as 3DMLW (see page 198).

OpenGL (Open Graphics Library) is a cross-language, multi-platform API for generating computer graphics (2D and 3D), mainly used for CAD and computer games. It was originally developed by Silicon Graphics Inc. (SGI) in 1992 and is now managed by the non-profit technology consortium Khronos Group.

WebGL is a Javascript API based on OpenGL and developed by the aforementioned Khronos Group.

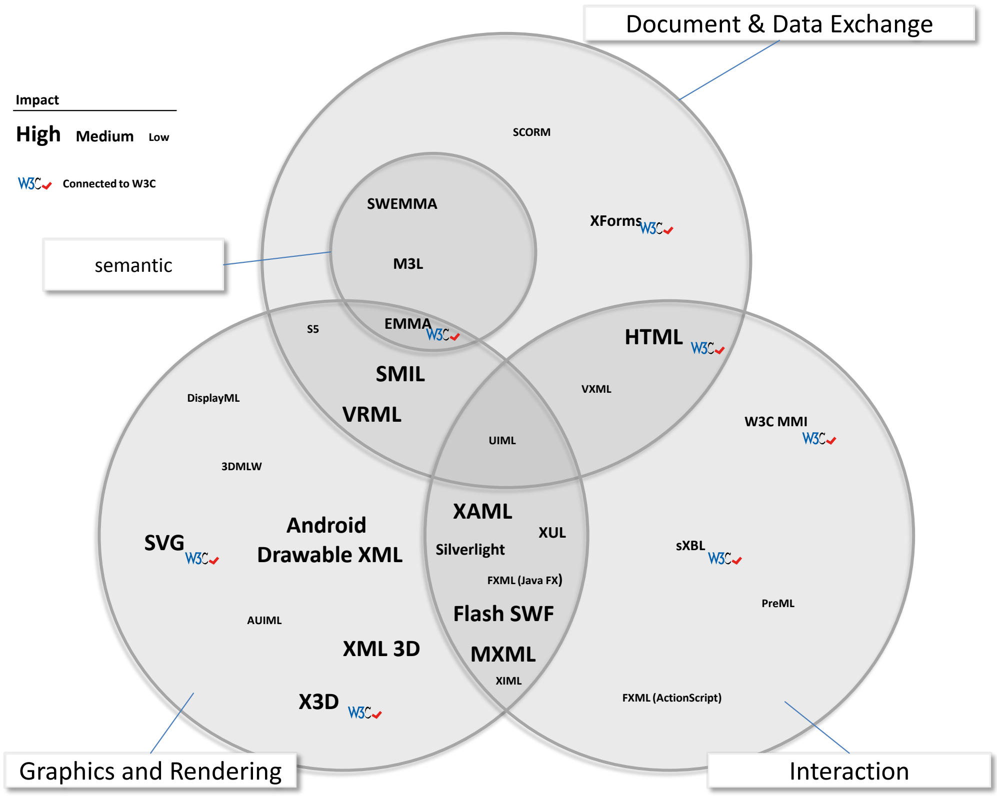


Figure 3.1: An overview of presentation languages according to their main objective and the extend of their impact.

The term *Rich Internet Application (RIA)* is not strictly defined. It usually refers to an application with connection to a specific web page, either running as a browser plugin or a stand-alone application. It shares many of the characteristics of a traditional desktop application.

An *Integrated Development Environment (IDE)* is a software application for developers which aims at convenience in editing source code and facilitating the development process. An IDE usually consists of editor, compiler, and debugger.

An *Application Programming Interface (API)* is a specification of an interface for communication between software components.

Software Development Kit (SDK) is a set of software development tools.

A *mobile device* or *hand-held device* is a small, portable computer that fits in the palm of a hand, and that usually uses a small touchscreen for interaction. *Personal*

Digital Assistants (PDA) were popular mobile devices in the late 1990s/early 2000s, but are nowadays replaced by cell phones and smart phones.

A *tablet computer* is a portable flat computer with a touchscreen, and differs from a mobile device mainly by its bigger size. Tablet computers became popular in 2010 when Apple Computers introduced the iPad. The variety of hand-held and tablet devices running similar applications ("apps") imposes several restrictions on the development and layout of user interfaces.

Classification of Languages

As mentioned previously, the extensive survey of almost thirty different presentation languages and related markup languages can be found in appendix A; in this section I would like to discuss some of these which are of special interest as being at the core of specialized presentation languages (cf. figure 3.2).
In 1998, one of the earliest popular attempts to formalize presentation content in a formal language was the *Synchronized Multimedia Integration Language SMIL* (cf. page 213). I consider this the starting point of presentation languages. SMIL is recommended by the W3C and very popular for online lectures and interactive presentations, since it defines all necessary elements such as markup for timing, layout, animations, visual transitions, media embedding, etc. SMIL is also highly integrated with other W3C languages; for example SMIL can be used as a means for animating vector graphics in SVG or be integrated in VoiceXML, MusicXML or RSS.
The *Extensible MultiModal Annotation markup language EMMA* (cf. page 202) is primarily used as a data interchange format between components of a multimodal system. The content is usually machine-generated, not authored. The purpose of intercomponent data exchange distinguishes this language from the others discussed in this section. EMMA is focused on semantic annotation of user input with automatically extracted information.
SWEMMA (cf. page 215) is an extension of the EMMA standard developed at DFKI in the context of the Smartweb project. In addition to EMMA, it provides tags for wrapping the result of processed information, the status of a process monitoring annotation, out of vocabulary words, and a turn-ID associated with each element.
The *MultiModal Markup Language M3L* (cf. page 207) was developed in the SmartKom project. The underlying idea is to cover all data interfaces belonging to a complex dialog system in one single, coherent language. The language definition has been decomposed in 40 different schema specifications in order to provide thematic organization and make the specification process manageable. The data flow between user input and system output continuously adds information to the M3L expression and refines it. Important semantic aspects are encoded in specific element structures. Complementing the language definition, an API has been developed as a lightweight

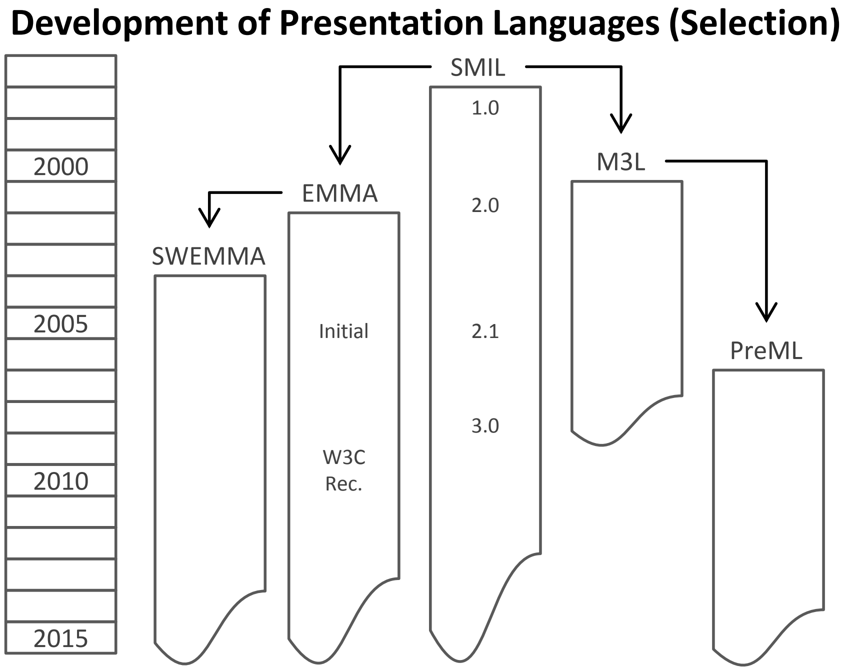


Figure 3.2: Time line and connections of selected presentation languages.

programming interface.

The *Presentation Markup Language PreML* was developed at DFKI and is, like M3L, used both for description of multimedia content and for small-footprint inner-system communication protocol. It was developed in the context of the Theseus project [HW11, Son10] but not limited to it. The expressiveness of the language ranges from encoding presentation tasks from the system to be presented to the user up to transmitting events like pointing gestures from the user back through the multimodal dialog system. It is derived from an ontological representation of the presentation task.

Summary and Discussion

The previous section provides a survey of presentation representation and markup languages, ranging from document description over data transport means to sophisticated interaction annotation. We can distinguish between interpreted and compiled descriptions.

While most languages have been designed for a specific task, we can see some patterns emerge. Figure 3.3 shows for instance the connection between user interface markup, user interface logic, compiled user interface, and interaction at runtime. This pattern can be found in Adobe Flash and in Microsoft Silverlight.

Other recurring themes can be found:

- The attempt to simplify development of the same GUI for different target plat-

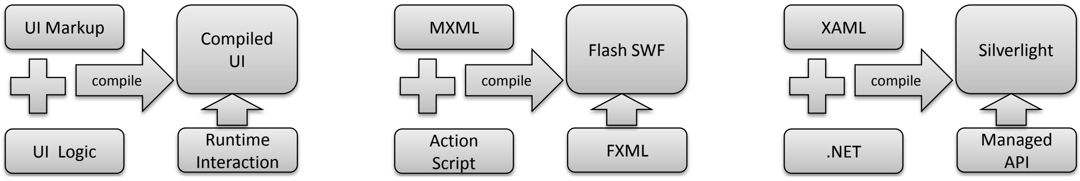


Figure 3.3: Patterns of combining markup languages.

forms or designs using renderers, e.g., AUIML, Android Drawable XML, or also in HTML using style sheets.

- Using a language not only for annotation of a user interface but also for transport of any interaction-related data through the system, e.g., PreML or M3L.

Furthermore, there is a strong trend towards XML-based languages. New languages are usually defined based on XML; in the case of VRML, which is based on the proprietary Open Inventor format, the successor X3D also changed syntax to XML.
Later in this thesis, I will describe how to use the lessons learned from existing languages to specify the requirements for an automotive presentation markup language suitable for situation-aware presentation management, which have already been communicated in the collaboration of our automotive research group with the W3C.

3.2 Development of Presentation Planning

In this section, I describe the development of research in intelligent multimedia presentation planning. The goal is to provide a historical background to the work presented in this thesis. The research projects presented here are closely related to DFKI for several reasons: 1. The research at DFKI shows prototypically the development of the field, and 2. the contribution of DFKI to the body of research is significant. Figure 3.4 shows a timeline of selected projects and research topics.

WIP (1989–1993)

WIP[1] is one of the earliest projects dealing with presentation planning. The underlying idea is the generation of personalized presentations based on information in a knowledge base, taking into consideration various factors such as available time, previous knowledge of the user, user preferences, etc. The most popular example of the WIP system was the user manual of a coffee maker.
According to [WAB$^+$92], the WIP project started in May 1989 for a 4-year period and was divided into three subgroups:

[1]WIP is an acronym for the german term *Wissensbasierte Informationspräsentation* (knowledge-based information presentation)

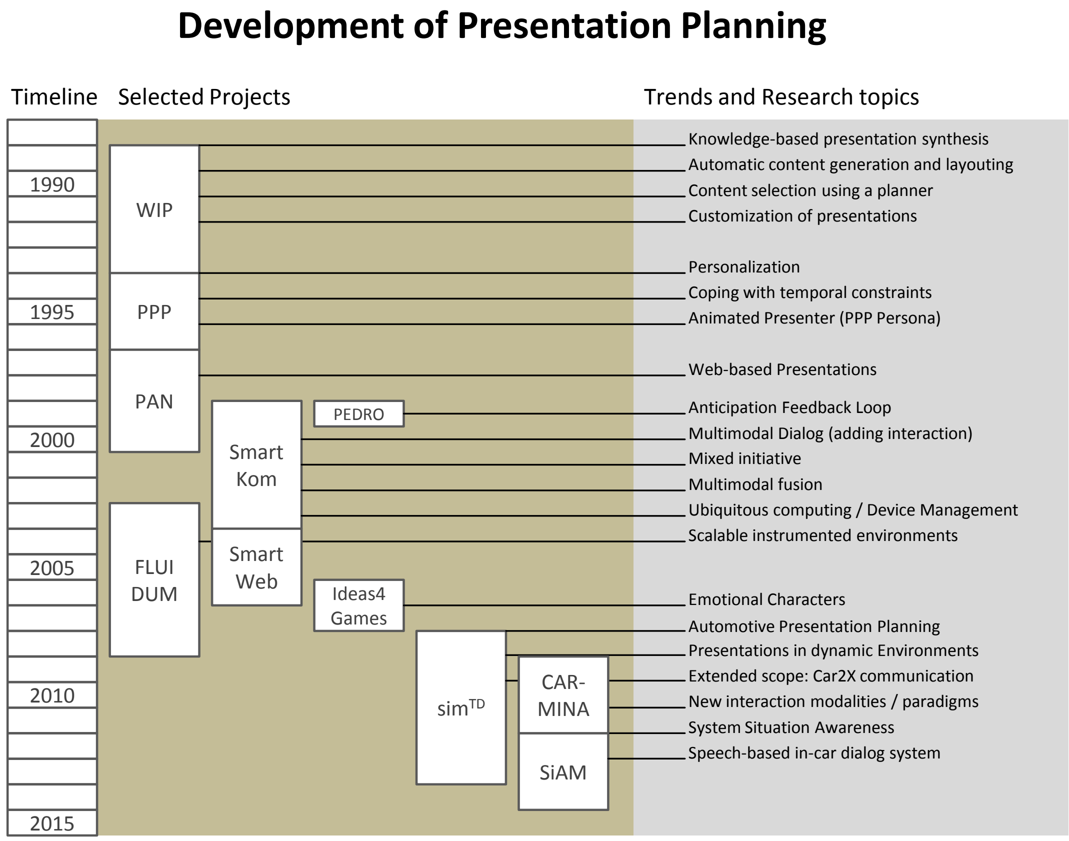


Figure 3.4: The development of Presentation Planning visualized using selected projects.

1. Presentation Planning: Focuses on problems of context-directed selection of content, automated graphics design, coordination of text and graphics, and constraint-based layout.
2. Language Generation: Incremental and parallel generation of text using lexicalized tree-adjoining grammars with feature unification.
3. Knowledge Representation: Extending the expressiveness of the terminological logic used in WIP with regard to the representation of temporal relations, action structures, default values, and exceptions.
[AFG$^+$93] state that the project most closely related to WIP is the COMET project [FM91], as both projects focus on the coordination of text and graphics. However, major differences exist (cf. table 3.1): Layout for instance is one of the latest stages in the COMET system, in which previously generated text and graphics are combined, while in WIP layout considerations are addressed at early stages of the planning process, based on the assumption that layout is an important carrier of meaning. Accordingly, the WIP system follows a modular approach for its architecture, with two parallel processing cascades for the incremental generation of text and graphics, with various forms of interaction between them. A presentation planner is responsible

	WIP	**COMET**
Layout	early layouting	late layouting
Planner	operator-based	schema-based

Table 3.1: Differences between the projects WIP and COMET.

for determining the contents and selecting an appropriate combination of presentation modes. The result of this planning process is a hierarchically structured plan of the document to be generated, with a more or less abstract presentation goal at the top and specific tasks at the bottom. The planner is described in detail in [AR93].
More information on the WIP project can be found in [Wah92].

PPP (1994–1996)

PPP, short for Personalized Plan-based Presenter, was the follow-up project to WIP. Instead of the generation of a static document explaining the use of a coffee machine, the example here was an animated explanation of the functionality of a modem. By adding animation, i.e., a temporal component, the orchestration of different presentation parts according to temporal constraints became a new issue in the research [AR96].
As a new feature, an animated character, the so called PPP Persona, was introduced [Mül00]. From an architectural perspective, the system can be divided into two main parts: multimedia generation (generation of text, graphics, and gestures) and multimedia display (layout manager and persona server), which are mediated by the PREPLAN presentation planner [RAM97].
Previous related work to the PPP Persona exists; the most well known example is the animated assistant used in Microsoft Office. But while previous work had focused on an integrated one-purpose agent, the Persona approach aimed at developing a skilled presentation agent independent of any particular application. Several requirements had to be met by the Persona, e.g., it had to be conversant with a broad variety of gestures and body postures, while at the same time adopting a lively behavior without distracting the user. The actions performed by the Persona can be classified into five groups: high level presentation acts, idle-time acts, reactive behavior, low-level navigation acts, and basic postures/acts.
An evaluation of the Persona in comparison with other animated agents can be found in [Lie10]. More information on PPP is available at [AMR97] and [And00].

PAN (1997–2000)

The project PAN, an acronym for Planning Assistant for the Net, was aimed at the development of plan-based information assistants for Internet users. The main tasks hereby involve the generation and execution of plans in dynamic environments, man-

aging the underlying domain model, and the implementation of the cooperative and adaptive behavior of the assistant.
Several important technologies were developed in PAN:
TrIAs - Trainable Information Assistants. The knowledge representation of TrIAs is based on domain-specific ontologies. By defining concepts and their relationships, a common representation for all system components as well as a means to integrate previously unrelated information is provided. These ontologies are used in (a) the formalization of the domain planning operators, (b) the description of web sites, and (c) the formulation of information requests [BD98b, BD99].
HyQL - The Hypertext Query Language. The main purpose of HyQL is the operationalization of basic information gathering processes. HyQl considers the web as a computable dynamic graph structure where the nodes are static or dynamic documents and the edges are links between them. Special focus in the design of the language was on robustness, which was added by the use of context information. The language structure resembles the database query language SQL [BDP01].
The Programming By Demonstration (PBD) paradigm. The PBD paradigm describes an approach of generating HyQL code snippets by interactive selection of the desired information by the user in a training dialog. This procedure is performed iteratively until the desired result is achieved [BDP00].
The Infobeans concept. The result of the PBD training dialog are so-called info beans, mainly consisting of HyQL code snippets [BD98a].
A popular demonstrator of the PAN project was the *Personal Picture Finder* [EMW99, End99a], which was one of the first web-based image search engines. By massive parallel web requests [End99b] and fast (heuristic) image analysis, this web-based application was able to provide photos of a person by using the person's first and last name as sole input. Another example was an interactive travel agent which was able to plan a complete trip for a user, including selection of hotel, train and flight schedules, information on events and sightseeing, etc.

PEDRO (1999)

PEDRO is an extension of the previously discussed PPP system presented in the PhD thesis of [vM99]. Due to the importance of the approach, it is presented separately here.
The focus of the work is on resolving ambiguity in graphical user interfaces and on the understanding of referring expressions. In order to achieve this goal, modeling of decoding problems is discussed in detail. The user interface is evaluated by decomposing it into its components and analyzing these components separately, followed by an overall evaluation and feedback to the system. The evaluation of the display part covers:
Ambiguity of the meaning of symbols. The problem consists in ambiguous or unknown

words, symbols or code. Important factors here are relative frequency of these items, absolute frequency, and context.
Understanding referring expressions. The author distinguishes between anaphoric referring expressions, cross-media referring expressions, and multimedia referring expressions. The focus of her research is on anaphoric expressions. Important factors are: relative perceptual salience, relative perceptual similarity, and the user's domain knowledge.
Locating information. The user searches the interface both top-down as well as bottom up. Relevant factors include background knowledge ("where the information can be expected", e.g., the title of a slide is at the top) and the conspicuousness of the information.
Identifying world referents. This problem is similar to identifying referring expressions. Important factors are the salience of both the pictorial reference as well as of the real world object. In the PEDRO system, a-priori values are used.
Recency. The more recently a depiction was used, the more likely the user will understand it.
Speed of user's references. "Slow inferences in general make more use of working memory, which can be irritating or distracting". In order to find a referenced object quickly, the desired object must be highlighted in some form on the display, and at the same time there should not be too many similar objects on the screen.
PEDRO uses a-priori information on the user, i.e., at system start the initial state is not an empty user model, but a stereotype to be refined.
In the overall evaluation of the display, the results of the partial evaluation are combined according to their relevance for the overall understanding of the system. This information is in turn provided back to the system in a so called *anticipation feedback loop*.
The advantage of this procedure is that a presentation to be shown to the user can be analyzed beforehand in terms of how easy it would be for the user to decode it, and, if problems are found, the presentation can be changed again.
The system was verified in two user studies on finding objects and decoding meaning.

SmartKom (1999–2003)

The SmartKom system adds symmetric multimodality, i.e., multimodal dialog, to the previously discussed approaches by using an embodied conversational agent. It was one of the most advanced dialog systems worldwide and is a landmark in the history of intelligent user interfaces [Wah06].
The architecture is based on the MULTIPLATFORM testbed, previously used in the VERBMOBIL project [Wah00], which consists of a sophisticated blackboard architecture to enable interprocess communication [HKM+03, HNM+04]. Communication across the system platforms was based on the M3L presentation language.

As a multimodal dialog system, SmartKom combines speech, gesture, and facial expressions for both user input and system output [WRB01]. Its aim is to provide an anthropomorphic and affective user interface by personification of the animated character representing the interface. The underlying idea is to enable the user to delegate tasks to the character, considering him as an assistant. The character attempts to recognize the goals of the user, perform the desired tasks, and present the results in an adequate manner [Wah03].
The SmartKom project provided three prototypes at different scales [RAB+03]:
SmartKom Public. As a public installation, SmartKom Public is an advanced multimodal information and communication kiosk for airports, train stations, or other public places. It supports information seeking and assists in personal communication.
SmartKom Home. For in-home use, SmartKom offers this multimedia portal to information services. It extends the kiosk by providing assistance in using and controlling consumer electronics, such as TVs or VCRs. It is based on a portable pad computer and has two different operation modes, lean-forward and lean-backward; the latter one being restricted to verbal communication.
SmartKom Mobile. This version is based on a PDA front end and is similar in functionality to modern smartphones.

Fluidum (2003–2008)

The FLUIDUM project (www.fluidum.org) was a junior research group ("Nachwuchsforschergruppe") at Saarland University managed by Andreas Butz, who later continued the project at LMU in Munich. Its aim was to build instrumented environments at three different scales (desk-, room-, and building-level) in order to investigate interaction techniques for ubiquitous computing. Based on the Ubiquitous Computing paradigm by Weiser [Wei91], the hardware was integrated as subtly as possible into the environment. As one of the major challenges, the middleware between a large heterogeneous set of different devices and the applications controlling them was developed [End03] as a foundation.
Research in FLUIDUM included the introduction of new interaction metaphors, unintrusive control of attention using ambient soundscapes [JB05], and the use of object surfaces and walls as interaction space (e.g., [BSS04]).
Further information on FLUIDUM can be found by [HWTB07] and [BJ05].

SmartWeb (2004–2006)

SmartWeb was the follow-up project to SmartKom and headed in a similar direction, building on the experience of the previous project [Wah07].
It was designed as a context-aware dialog system supporting its user everywhere and at any time with information by providing access to the semantic web [Wah04]. The

system is based on the W3C standards *Resource Description Framework* (RDF/S) and the *Web Ontology Language* (OWL) [RBE+05].
Originally, it was realized as a mobile application on a PDA [SEH+07], but later on extended as an in-car system.

Ideas4Games (2006–2007)

The aim of Ideas4Games was the "creation of interactive expressive characters with a consistent behavior" which "comes with a whole range of challenges, such as interaction design, emotion modeling, figure animation, and speech synthesis" [GSC+08]. It is based on the emotion model ALMA introduced by [Geb07].
One of the goals was to demonstrate the advance of current research in artificial intelligence over state-of-the-art industrial implementations of non-player characters (NPC). The resulting prototype "AI Poker" consisted of an interactive poker table, where the user could play with real, RFID-tagged cards against two animated talking virtual characters [SGC+08]. Events occurring during the game, e.g., the human player dealing cards too slowly, or the NPC having especially good or bad cards, were evaluated by the system and resulted in emotional behaviors from the characters. The emotions were most plainly perceivable by gestures and voice pitch of the characters, but also more subtly included in things like the breathing frequency.
The interaction was scripted using the Scenemaker tool, which allows the developer to model interaction by visual editing of a hierarchical finite state machine and by editing scenes in a text editor while at the same time annotating gestures to be performed [GKKR03, GMK11].

sim^{TD} (2008–2013)

The sim^{TD} project, described in more detail in chapter 4, was intended as a proof-of-concept for Car-2-Car and Car-2-X technologies, by building a prototype based on the research in that area in the past decades and evaluating it in a large field test.
In terms of presentation planning, this project extended the traditional approaches in a variety of ways:
Time-critical presentations: Unlike most of the data presented in former multimodal systems, in-car warnings are not only time-critical, but also include potentially safety-critical information.
Dynamic changing environment: Constant replanning becomes necessary in the driving context. For instance, if the system calculated that the driver has to be warned about a traffic sign at a certain moment, then this plan may have to be changed in case the driver accelerates or decelerates before reaching the position of the traffic sign.
Extension of presentation scope: Car-2-X communication extends the spatial scope for presentation beyond the user's environment, i.e., the interior of the car, by making

it possible to trigger messages to be displayed in another car.

Carmina (2009–2011)

Project CARMINA is centered around multimodal interfaces for car passengers that support the interaction (a) of the passengers with the car and mobile Internet services, (b) between the passengers inside the car, and (c) between the passengers and the road environment.
Support for combining different interaction modalities, such as speech, physical manipulation of haptic interfaces, and touch-free gestures [ESM11] was one of the main goals. Additionally, the external context was taken into account [CEF+11] by building ad-hoc multi-hop P2P networks based on top of Car-2-Car infrastructure [Kun11].
Research in CARMINA was guided by the following research questions:[2]

- How can the systems dialog behavior be synchronized with and adapted to a rapidly changing physical and spatial context induced by the cars motion?
- How can the current multimodal fusion algorithms be generalized to support multi-party interaction and tangible interfaces where new approaches to mutual disambiguation of modalities are needed?
- How can the multimodal fission and media allocation algorithms be extended to cope with a situation that involves many LED indicators, multiple screens, and at least two audio output areas in premium cars of the future?
- However static the role allocation (driver, co-driver, passengers) during a single ride might be: How can the dialog be adapted to multiple role modalities or usage modes (e.g., chauffeur mode vs. family mode)?

SiAM (2012–2014)

The recently started project SiAM extends the research of both sim^{TD} as well as Carmina. Its main goal is to provide the scientific and technological foundation for situation-aware multimodal communication in modern cars. The perspective here is more holistic than previous approaches: not only the individual car, but the entirety of vehicles and their interconnection with each other and their environment is being considered. It is assumed that in the future, the car will not only be considered as a means of transportation, but rather a technology and communication hub: as a mobile Internet hub, mobile sensor hub, and mobile computer server at the same time.
While of course the main purpose of the car will still be personal mobility, other goals such as safety, increase of mental comfort, enabling mobile working, and improvement of infotainment and entertainment become more and more important.

[2] cf. automotive.dfki.de

Relation to this thesis

The work described in this thesis is a continuation of traditional presentation planning research described here and can be considered a link between sim^{TD} and CARMINA on one side by building on the experience gained in these projects, and SiAM on the other side to which it contributes by adding situation awareness to the included dialog system.

At the same time, limitations apply to the automotive environment in the form of real-time requirements. From a practical point of view, this limits the applicability of traditional plan-based presentation planning approches that have been used in the WIP and PPP projects for example. The focus shifts depending of the urgency of the information to be presented from real-time generation of presentation to real-time selection and configuration of pre-defined presentation alternatives. For long-term presentation goals, such as introducing the functionality of a new car to the driver, traditional planning methods can still be applied as an overlay to the more reactive system used in critical situations. Section 11.1 describes the change between the planning mode and the reactive mode of the system in more detail. On the spectrum ranging from fully planned presentations to merely reactive planning, I position my system somewhere in the middle between those two approaches.

Chapter 4

The sim^{TD} Project: In-car Scheduling and Presentation Planning

The work in the project sim^{TD} was the foundation of the research provided in this thesis. The presentation management mechanism I developed for sim^{TD} is the predecessor of the PRESTK system presented here. In numerous workshops at the testing area in Friedberg, and also by evaluation in a large scale field test, valuable lessons were learned, helping to refine theoretical concepts with practical experience on the road. The basic information about sim^{TD} is presented here; further concepts and information about the role of DFKI and especially my development in sim^{TD} are discussed in chapter 8.

The name sim^{TD} is an acronym for *Sichere Intelligente Mobilität–Testfeld Deutschland.*[1]

The presentation manager developed for this project (cf. chapter 8) was aimed at a single-screen HMI where the screen is partitioned into several areas. The original design was based completely on look-ahead planning, but exceptions had to be included later on due to time-critical "real life" requirements on the road. A part of the content of this chapter I have documented previously in the sim^{TD} working document W22.2 "HMI Handbuch" (HMI manual) in German [CEM$^+$12].

4.1 General Description

Objectives and Timeline

According to the official webpage simtd.de, the project is pursues the following principle objectives:

[1] Safe Intelligent Mobility–Testfield Germany

- Increased road safety and improved efficiency of the existing traffic system through the use of Car-2-X communication.
- Definition and validation of a roll-out scenario for the identified functions and applications for scientific questions through practice-oriented experiments and field-operational tests.
- Consolidation of Car-2-X functions from the categories of traffic efficiency, driving and safety as well as value-added services.
- Definition, analysis, specification and documentation of those functions that are to be developed and tested, as well as of the resulting requirements for the overall system for selected functions and tests.
- Development of test and validation metrics and methods in each phase of the overall system development in order to allow measurement and evaluation of the results.
- Consolidation and harmonization of requirements from the perspective of feasibility and performance as well as their compatibility of requirements within the sub-projects.
- Verification of functions and requirements within the context of individual milestones.

Consortium

The consortium of the sim^{TD} project consits of major german automotive OEMs (Original Equipment Manufacturers), suppliers, network operators, research institutes, universities, and public institutions (cf. figure 4.1).

Relation to this Thesis

Involvement in the sim^{TD} project and development of a presentation management component to be used in a car on the road, both on the testing area in the Ray Barracks in Friedberg as well as in a large scale field test, was a necessary and important step towards the development of the PRESTK system.
The algorithm developed for sim^{TD} covers both scheduling and selection of different presentation strategies. It also uses a dynamic priority management approach.
During development and testing, important requirements for in-car presentation management could be identified. The PRESTK approach takes advantage of this valuable information and can be considered the logical “next step” after sim^{TD}.

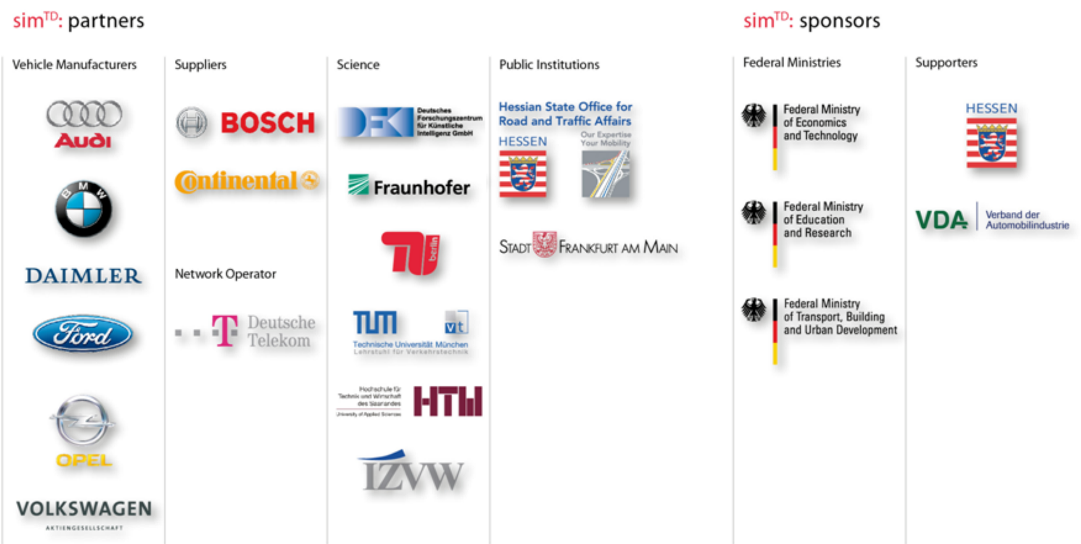


Figure 4.1: The consortium of the sim^{TD} project (source: sim^{TD}).

Hardware setup

With multiple OEMs being involved, one focus of the sim^{TD} project was to ensure interoperability and compatibility of the installed systems. After all, Car-2-Car communication on a large scale only makes sense if it is manufacturer-independent.
There are two computers installed in each car, the **Application Unit (AU)** and the **Car Communication Unit (CCU)**. The AU is running the main part of the system and the application logic, while the CCU is in charge of communication, GPS signal and sensor information. An eight-inch **display** with a resolution of 800x480 pixels running a custom-developed Flash GUI is installed in the middle console of the car. This screen also provides touch functionality for user interaction.

Usability Rules

Usability rules ensure that presentation tasks are always presented in an ideal manner for the driver to be notified with minimal distraction. Several objectives are pursued:

- A simple and intuitive usage, ensuring and supporting that the user has a simple mental model of the system. There should never be two different pieces of information simultaneously in the focus of the presentation, not even in two different modalities.

- A consistent logic of usage on all screens.

- Minimized interruption of user-triggered dialogs.

- Warnings are more important than hints, which are in turn more important than information.
- Minimized annoyance through minimized use of audio channel.
- Sufficient time to process information; no time pressure unless necessary.
- Some information cannot be presented while driving.
- Possibility for the user to choose a function with low priority to be displayed on the screen, even if there is a lot of information already displayed.

Software Infrastructure

To facilitate the development process between multiple parties involved, the OSGi framework has been used. OSGi stands for Open Services Gateway initiative and is a module system and service platform for the Java programming language that implements a complete and dynamic component model. By doing so, it extends the JavaVM beyond its included features.
OSGi supports the remote control of so-called bundles, which can be installed, started, stopped, updated and uninstalled without requiring a system reboot.
At DFKI, we developed the Human Machine Interface (HMI) bundle and the navigation bundle for the sim^{TD} project.

4.2 Presentation Tasks and Scheduling Requirements

The requirements for presentation task scheduling reflect the aims and the structure of the sim^{TD} project. This encompasses the selection of information to be presented, the information available in the system, and the selection and layout of the HMI communication channels.
The design and layout of the sim^{TD} HMI is guided by project constraints, e.g., a uniform HMI for all OEMs, ease of installation, and economical considerations. As a result, the HMI communication channels are rather minimalistic, consisting of one touch-screen and an audio channel for text-to-speech messages and short audio signals, such as warning tones.
From an HMI point of view, the information flow in the sim^{TD} system is a process with four stages (see Figure 4.2). First, context information such as position, vehicle status (current speed, etc.), and sensor data is acquired. Based on that information, several mutually-independent functions decide whether or not a presentation needs to be triggered. The mutual independence of these functions is an important factor in the

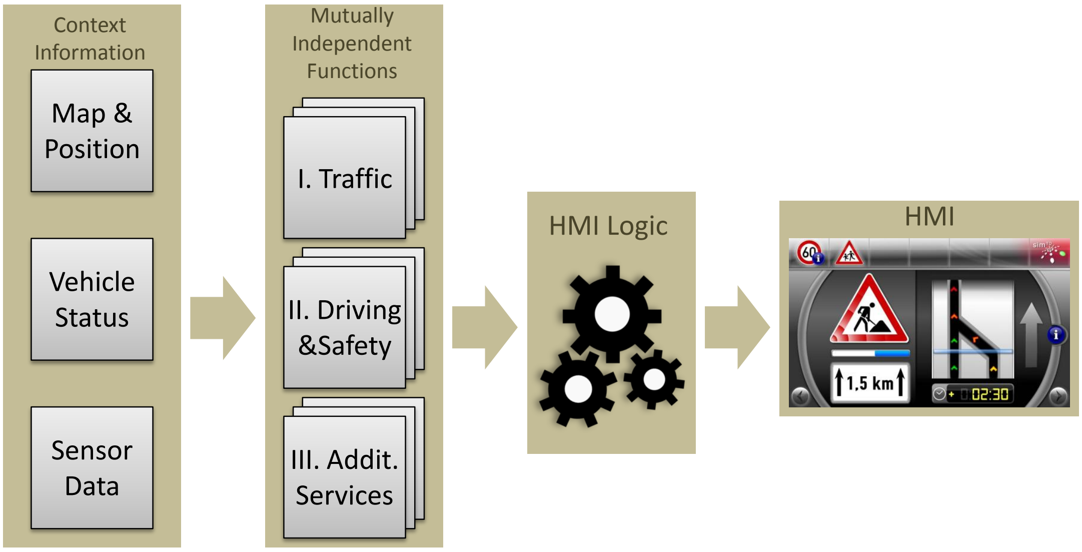


Figure 4.2: Information flow in the sim^{TD} HMI.

occurrence of scheduling conflicts. Functions in sim^{TD} are grouped in three different categories: traffic, driving and safety, and additional services. In the HMI bundle, which is the component containing the logic of the HMI, these potentially conflicting presentation tasks are stored and processed, while still being accessible for updates by their respective functions. Once the scheduled time for the presentation is reached, the presentation is triggered and sent to the HMI, which is in charge of rendering and displaying the presentation.

Presentation Task Types

As mentioned previously, presentations in the sim^{TD} system fall into three different categories, described as "main functions". Category I, *traffic*, covers information on traffic flow and traffic control. Category II, *driving and safety*, includes detailed information on local danger warnings and information to assist the driver. In category III, *additional services*, we find localized information and Internet-based services. Figure 4.3 shows a detailed overview of the relevant sim^{TD} functions.

Global and Local Priorities

The prioritization process in sim^{TD} takes place on two levels.
The *global priority range* of a function is agreed on between all parties involved. At DFKI, a workshop was conducted to set a range for each function priority as a subset of the overall range $[0, 100]$. Three values were set: minimal priority, typical priority, and maximal priority. The typical priority of safety-critical functions was set at a value of 85.

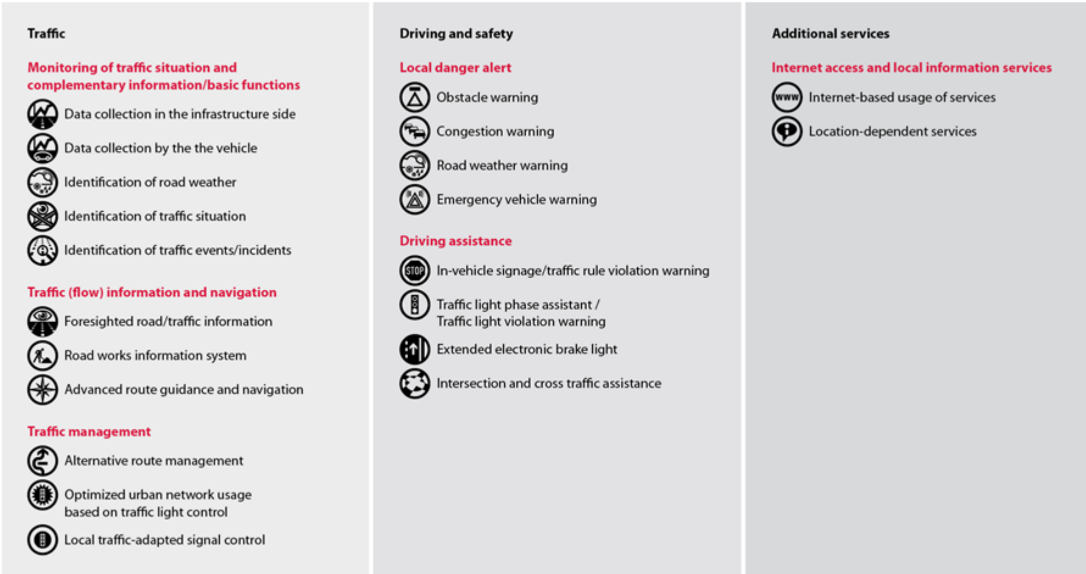


Figure 4.3: Relevant functions (source: sim^{TD}).

The *local priority* of the instance of a presentation is a value in the range $[-100, 100]$. It is intended as a means for the function creating the presentation task to position a specific instance in the global priority range. A value of 0 indicates that the typical value of this range is appropriate, while a value of -100 or 100 respectively indicate the lowest or highest value in the global priority range. Other values are linearly matched to this range. The resulting priority is called *global priority*.

Presentation Task Lifecycle

There are several basic functions for the generation and control of presentation tasks.
Create presentation task: As soon as a function is aware that it wants to show a presentation on the HMI, a presentation task can be created. This is done by instantiating a presentation model with at least a minimal set of parameters, which can be changed again later on. Usually, the minimal set of parameters consists of start time, duration, and relative priority.
Update presentation task: There are two different phases in which a presentation task can be updated. Before its actual presentation, basically every parameter of the presentation task can be updated, including its start time. Once it is on the screen, the options for updating are slightly more limited. In some cases, updates while being displayed are part of the presentation concept, e.g., progress bars while traversing a construction site.
Cancel presentation task: Functions may cancel their presentation tasks at any time during their lifecycle, either before they are presented or while they are already being presented.

Virtual Screen Concept

The sim^{TD} HMI uses the concept of virtual screens. The driver can at any time switch between three different sceen-views on one physical touch-screen, by using the navigation buttons to the left and right. Figure 4.4 visualizes this concept and the logical order of the three screens:

Main Screen.

The main screen is the default view of the HMI if navigation is not used. Its logical position is in the middle of the three virtual screens. In its main area, complex graphics, text and symbols (or a combination thereof) is displayed. The symbol area (upper area) shows information that is relevant in the current context but not important enough to be displayed on the main area of the screen. This symbol area is visible on all virtual screens. The symbols can be used as buttons to obtain more information.

Navigation Screen.

This screen shows navigation information and an interactive area for selecting points of interest (POIs) and additional information. The navigation screen is to the left of the main screen.

Options Screen.

This screen is used for configuring system settings. Furthermore, the driver can report obstacles on the road interactively here. The options screen is to the right of the main screen.

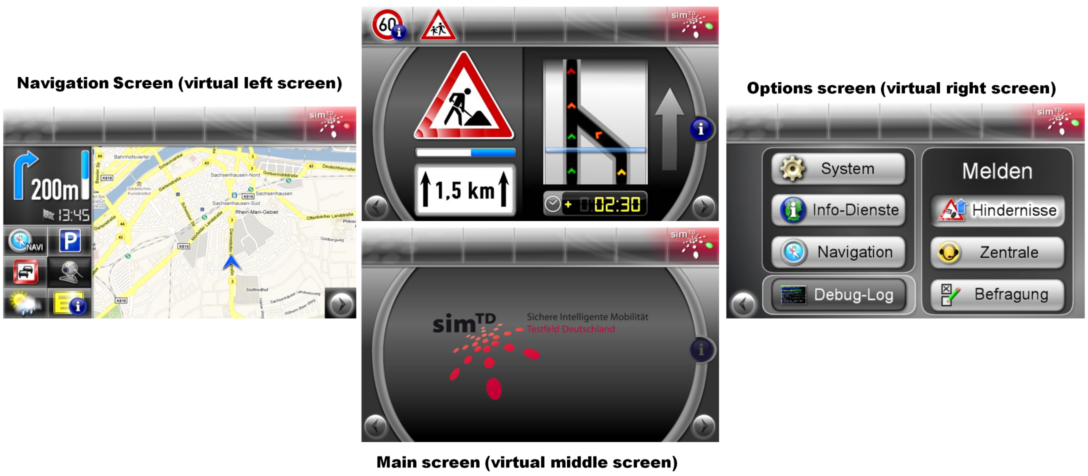


Figure 4.4: The sim^{TD} concept of virtual screens.

4.3 Field Test

The sim^{TD} system was tested and refined in multiple workshops over a span of two years on the test area located in the Ray Barracks (abandoned U.S.Army casern, famous for stationing Elvis Presley) in Friedberg/Hessen, Germany. Predefined courses and events in Car-2-Car and Car-2-X communication were tested using a fleet spanning about three dozens of cars from all major german car manufacturers. Special infrastructure has been installed such as road side units (RSU) and a traffic light installation capable of Car-2-X communication.
The tests were aimed at ensuring the overall functionality of the sim^{TD} system. Building upon that, a large-scale field test on open roads started in the area of Frankfurt in August 2012. It is the biggest Car-2-X communication field test worldwide. 120 cars of different OEMs with identical HMIs built in (cf. figure 4.5) are testing functionality, suitability for daily use, and efficiency of the sim^{TD} technology. The overall aim is to have a working Car-2-X communication system as a proof of concept. 450 drivers of ages 23 til 65 are involved. Special care is taken to consider non-technical experts as drivers. Until the end of 2012, a total of 40.000 driving hours with the system is planned.

Figure 4.5: Top: Cars of different OEMS with built-in sim^{TD} system (images: M. Leissl / sim^{TD} press photos). Bottom: The sim^{TD} fleet at the field test in Frankfurt (image courtesy of S. Castronovo).

Chapter 5

Situation Awareness

5.1 Definitions of Situation Awareness

The term *Situation Awareness* is used by many people, and with a variety of different meanings. Its origin lies in the area of military aircrafts and pilot training, but it has been extended since the 1990ies to a more general research of human factors in a variety of domains. [End95] mentions aircraft, air traffic control, large-system operations, tactical and strategic operations, and others.

Another term frequently used is *Situation Assessment*. This term describes the process of acquiring Situation Awareness, as opposed to *Situation Awareness*, which is the result of that process (see Figure 5.1).
Many different definitions of Situation Awareness can be found in literature; they mainly are more complex descriptions of the concept that Mica Endsley described with "Most simply put, Situation Awareness is knowing what is going on around you." Here are two more examples:

> Situation awareness is "adaptive, externally-directed consciousness that has as its products knowledge about a dynamic task environment and directed action within that environment" [SH95]

Vidulich emphasizes the fact that Situation Awareness is not static but rather continuously changing:

> Situation assessment (resulting in situation awareness) is "continuous extraction of environmental information, integration of this information with previous knowledge to form a coherent mental picture in directing further perception and anticipating future events." [VDVM94]

[Fla95] adds a few critical thoughts to the discussion of definitions:

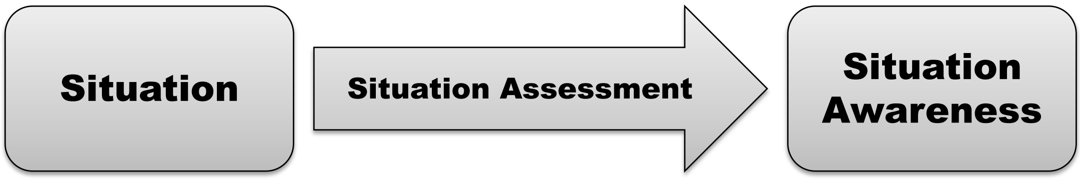


Figure 5.1: Connection between Situation, Situation Assessment, and Situation Awareness.

> "Clearly, SA is an appropriately descriptive label for a real and important behavioral phenomenon (Level 2 concept). The danger comes when researchers slip into thinking of SA as an objective cause of anything (Level 3 concept). A statement that SA or loss of SA is the leading cause of human error in military aviation mishaps (e.g., [HSP91]; cited by [SPBS95], this issue) might be criticized as circular reasoning: How does one know that SA was lost? Because the human responded inappropriately. Why did the human respond inappropriately? Because SA was lost. Is this keen insight or muddled thinking?"

In other words: We have to be careful when using the concept of Situation Awareness, and avoid getting carried away in cyclic reasoning or definitions.

Endsley's definition [End87] of situation awareness is popular and widely used:

> "Situation awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" (Mica R. Endsley)

One interesting aspect of this definition is that it does not specify the agent or recipient of the perception. While it is commonly assumed to be the operators (pilot, air traffic controller, etc.) perception, we might as well take the point of view of the system's perception of the elements in the environment. In this thesis, I use the term *Situation Awareness* according to Endsley's definition, but widen it to the system's point of view.

5.2 Endsley's Model of Situation Awareness

Endsley's model (cf. figure 5.2) shows situation awareness as part of a feedback loop, where the state of the environment influences SA, which in turn influences decision making, which in turn influences performance, which in turn closes the loop and influences the state of the environment.
Endsley distinguishes three levels of SA:

SA Level 1: Perception of the Elements in the Environment

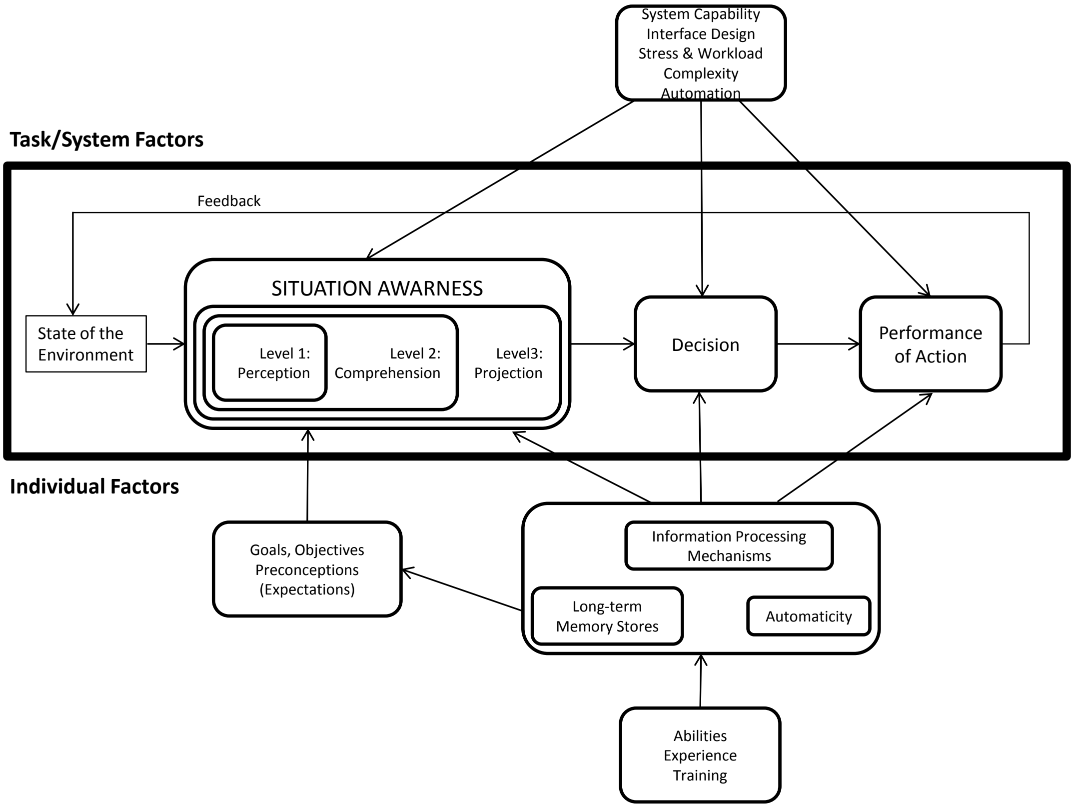


Figure 5.2: Endsley's Model of Situation Awareness based on [End95]

As a first step in achieving SA, the relevant elements in the environment need to be perceived in terms of their status, attributes, and dynamics. [End95]: "An automobile driver needs to know where other vehicles and obstacles are, their dynamics, and the status and dynamics of one's own vehicle."

SA Level 2: Comprehension of the Current Situation
The disjointed elements perceived in level 1 are synthesized to a holistic picture of the environment, attributing their significance from the perspective of the operator's goals.

SA Level 3: Projection of Future Status
As the third and highest level of SA, the future status of the environment and the objects therein are projected. It is achieved by combining knowledge of the current status of objects with the knowledge about their dynamic, as obtained in levels one and two. [End95]: "An automobile driver also needs to detect possible future collisions in order to act efficiently."

Time plays an important role in that model. SA is not necessarily acquired instantaneously, but accumulated over a period of time, e.g., depending on the history of perceptions. *Space* as well is crucial here; among other reasons because it can have an impact on timing. Knowing the speed and distance of an object enables estimates of its future position.

Five years later, [End00] extended the concept of SA with the concept of the *Information Gap*, stating that more data does not necessarily mean more information, but rather that the gap between the data produced and the information needed is getting bigger, which results in a higher need for sorting and processing the data.

5.3 Basic Theories on Cognitive Load

5.3.1 Yerkes-Dodson Law

In 1908, Yerkes and Dodson [YD08] examined the effect of electrical shocks on the learning behavior of laboratory mice. The mice were guided in a pathway with two options, a black door and a white door, and using the black door was followed by electrical shocks of different intensity.
Their results have become a basic law of performance, the Yerkes-Dodson Law (sometimes also referred to as Yerkes-Dodson Principle):

> "Performance improves as arousal increases until a point at which time it decreases." (Yerkes-Dodson Law)

Accordingly, the connection between performance and arousal is curvilinear and looks like an inverted U (cf. figure 5.3).
There are several problems with the experiments of Yerkes and Dodson, discussed for instance by [Sta04]:
1. The methods of calibrating electrical shocks were rather crude, i.e., the measurement was not exact.
2. There is an underlying assumption of a linear dependency between strength of the stimulus (electrical shock) and level of arousal, which is neither explicitly stated nor validated.
3. Although widely accepted, the transfer between behavior of laboratory mice and human beings has not been validated.
On the other hand, Yerkes and Dodson were rather modest about their findings and did not claim to have found a law of broad applicability, so it would be unfair to blame them for any hasty conclusions other scientists made based on their results. The "law-status" was raised almost half a century later by Broadhurst [Bro57], and the curvilinear connection between arousal and performance, despite being sometimes disputed, stays untouched in the scientific literature since [Sta04].

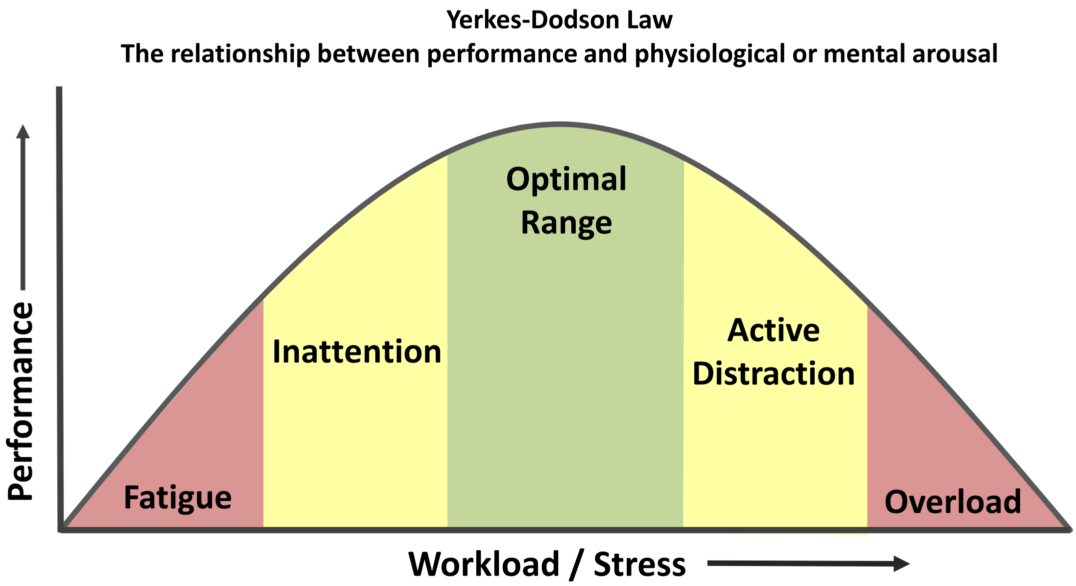


Figure 5.3: The Yerkes-Dodson-Law as depicted in [CRM11] (courtesy of Bryan Reimer, MIT.)

Other, competing theories evolved, such as the Hull-Spence drive theory, which postulates a linear connection between drive (synonymous for arousal) and a well-learned response, and a decrease in performance if a task is not well learned. This theory did not last long due to a lack of empirical support and even anecdotal evidence to the contrary.

5.3.2 ALMA: A Layered Model of Affect

ALMA, a layered model of affect, was introduced by [Geb07]. It is based on the emotion concept of cognitive psychology, which states that emotions are a reaction to the evaluation of the environment. Emotions per se are not a part of this thesis, but the concept of temporal decay of short-term emotions is relevant: It can serve as a reference for implementing the decay of cognitive demand of a presentation on a driver.
Figure 5.4 shows the emotion model of ALMA. For the generation of emotions, an abstract evaluation language is used. This language serves as an interface between the model and software systems accessing it. The calculation of emotions is based on recent research in cognitive psychology. It is based on statistic knowledge of the personality and the results of dynamic evaluation. The underlying theories are the works of Ortony, Clore, and Collins ("OCC", [OCC90]) and the big five personality model [BM91], also known as OCEAN-model after its five factors: openness, conscientiousness, extroversion, agreeableness, and neuroticism.

Emotions are, unlike personality, dynamic and decay in intensity over time. Their

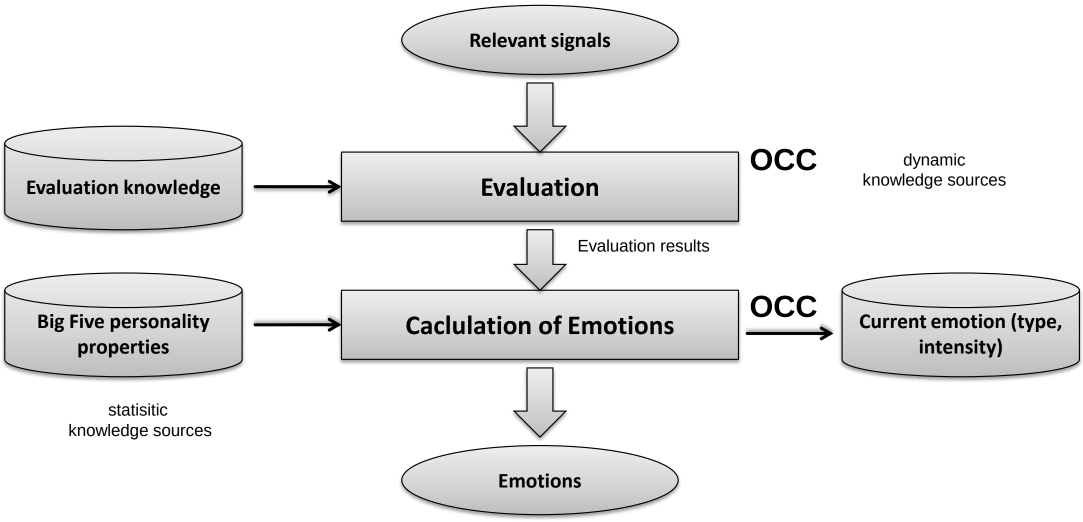


Figure 5.4: The layered ALMA model for calculation of emotions as described in [Geb07].

decay is usually modeled linearly over time, with the assumption that when an emotion reaches the minimal intensity in the model, it is not there anymore. Of course, other decay functions are important too. [Geb07] proposes three different implementations inspired by decay processes in nature as possible candidates for duplicating the emotional decay process in an algorithm:

1. **Linear decay.** This model idealizes, for instance, the flow of a fluid from a jar through a hole of fixed size, or the emptying of a battery. The loss per time unit stays constant.

2. **Exponential decay.** The most common example here is radioactive decay.

3. **Tangent hyperbolic decay.** This function models for instance the flow of heat from a body, or several growth or decay processes in nature.

All of these functions can be individually adapted and parametrized in the ALMA implementation.

5.4 Cognitive Load Assessment

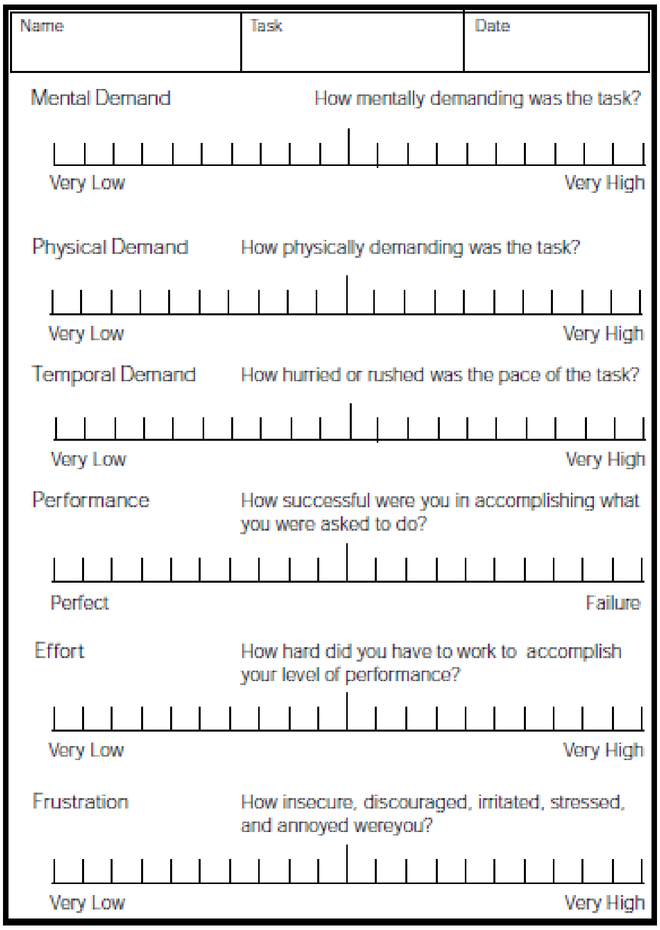
Although the definitions of *Cognitive Load* (CL, sometimes also called *mental workload* or *cognitive workload*) slightly differ from each other, they are typically similar to Wicken's definition as "the relationship between the cognitive demands of a task and the cognitive resources of the user" [Wic02]. A more detailed definition is given by [BJ99]: "the demands placed on a person's working memory by (a) the main task that she is currently performing, (b) any other task(s) she may be performing concurrently, and (c) distracting aspects of the situation in which she finds herself". [Cai07]

provides a survey of alternate definitions.
Traditionally, workload assessment techniques are divided into three groups: subjective measures (questionnaire based, self-reported), performance-based measures, and physiological measures. By widening the scope of assessment beyond actual measuring, we might add a fourth category of deducting cognitive workload from the environment.
In this section, we discuss several methods for CL assessment with respect to their usefulness in in-car applications and their feasibility for non-intrusive measurement while driving.

5.4.1 Subjective Measures

A simple and reliable way to assess a subject's workload is self-reporting, assuming that the person is cooperative and capable of introspection and reporting their perceived workload, either directly or by answering questions resulting in a measure. Commonly, questionnaires for self-reporting workload refer to a task already performed. One of the most widely known methods here is the NASA Task Load Index (NASA-TLX). Self-reporting of workload usually covers a single task and cannot be used without extension or modification to report on a complex situation involving several, potentially overlapping, tasks.
Applying questionnaires is an intrusive procedure (adding another task to the subject's working memory) and can only be done after the task has been performed.
Some of the tests are intended to be administered "online" right after performing the task, but then the test may interfere with performance in subsequent tasks.
None of the online questionnaires are designed for real-time assessment.
[AID06] provides an extensive survey over the different methods. [RDMP04] compares and evaluates three of these methods.
It is important to keep in mind that most of these questionnaires are not designed for automotive applications, and not all of them measure the same dimensions–if they are multidimensional at all.
In this section, a selection of subjective measures is presented and compared in terms of their applicability to the automotive domain.

NASA Task Load Index (NASA-TLX)
The NASA Task Load Index (NASA-TLX), which was developed in a "multi-year research effort aimed at empirically isolating the factors that are relevant to subjective experiences of workload" [HS88], was originally intended for crew complement in the aviation domain. Since its introduction in the mid-eighties, it has spread significantly beyond the original application, focus and language [Har06]. It is designed as a short questionnaire with 6 questions to be answered on a 21 point scale (cf. figure 5.5). The result of the test after a complex evaluation, including weighting of the imposed



Name	Task	Date

Mental Demand — How mentally demanding was the task?

Very Low — Very High

Physical Demand — How physically demanding was the task?

Very Low — Very High

Temporal Demand — How hurried or rushed was the pace of the task?

Very Low — Very High

Performance — How successful were you in accomplishing what you were asked to do?

Perfect — Failure

Effort — How hard did you have to work to accomplish your level of performance?

Very Low — Very High

Frustration — How insecure, discouraged, irritated, stressed, and annoyed wereyou?

Very Low — Very High

Figure 5.5: The NASA Task Load Index (NASA TLX) as an example of questionnaire-based self assessment of cognitive load.

workload and rating the magnitude of each factor for the test, is a multidimensional numerical value on six subscales, only one of them being mental demand. The other subscales are physical demand, temporal demand, own performance, effort, and frustration.

Bedford Scale

The Bedford Scale [CSGB$^+$89] uses a completely different approach. It is a unidimensional rating scale designed to "identify operator's spare mental capacity while completing a task". It uses a hierarchical decision tree guiding the user to a rating scale value between one and ten (cf. figure 5.6). It is an obvious advantage of the process that in each step of the decision tree, the symptoms of having exactly that level of workload are verbally described. This prevents the user from a natural tendency to avoid the extreme values of the scale, even if appropriate.

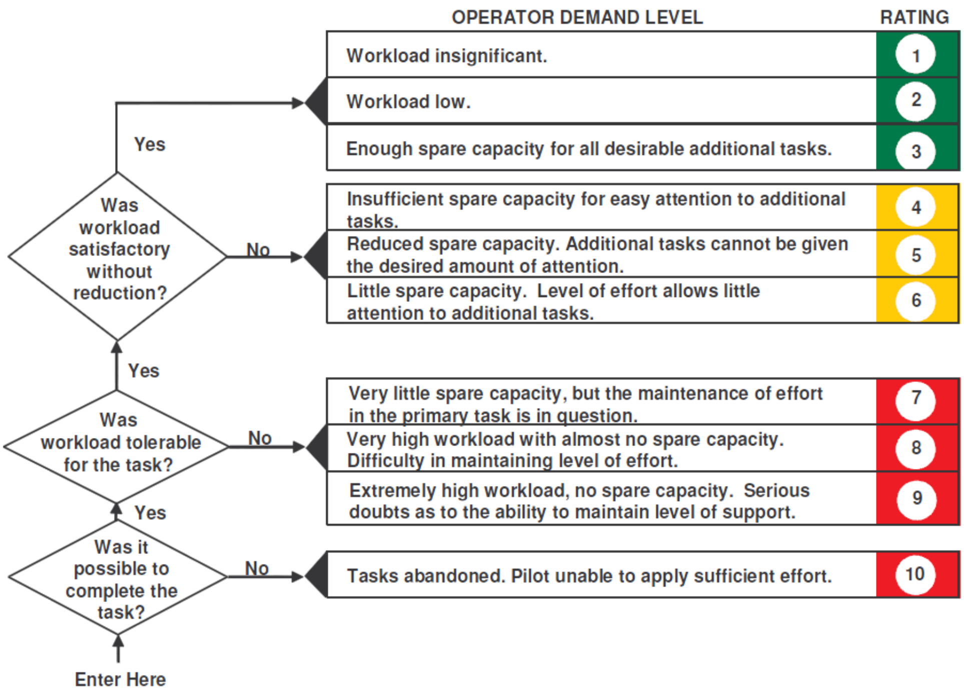


Figure 5.6: The Bedford Scale as an example of decision-tree-based self assessment.

Subjective Workload Dominance

The *S*ubjective *WOR*kload *D*ominance (SWORD) technique, as another example, is based on mutual comparison between tasks [VWS91]. As described in [AID06], the user gets a list of all possible pairs of given tasks and has to rate the mutual comparison between them on a 17-level scale. A judgment matrix is then calculated based on this data. If this matrix is consistent, relative ratings of each task can be determined.

Method	Type	Result dimensions	Reference
NASA-TLX	linear questionnaire	multi-dimensional	[HS88]
Bedford Scale	decision tree	uni-dimensional	[CSGB$^+$89]
SWORD	linear questionnaire	uni-dimensional (task ranking)	[VWS91]

Table 5.1: Subjective measures in comparison

5.4.2 Performance-Based Measures

Assuming that an increased CL diminishes human performance, we can use performance measures as an indicator of actual workload. This assumption is backed by the Yerkes-Dodson-Law [YD08] (see Section 5.3.1). The basic statement is–rephrased for our domain–that the driver's performance is best at a medium level of arousal/workload, i.e., he should neither be bored nor overwhelmed. [HNE02] also examined the impact of cognitive distraction and showed that it has a negative influence on driving performance and safety, especially on the driver's visual behavior.

Two approaches of performance measures are feasible in an automotive environment: measuring the driving performance and measuring the reaction time to events such as displayed information or events outside the car.

Driving performance

Recent literature on measuring the driver's CL strongly emphasizes the role of speed and steering wheel angle and their respective change over time. This is very convenient, since this information is easily acquired using the car's CAN-bus [Bos91].
[IKL11] built a prototype to estimate driver distraction in a simulator based on a Fast Fourier Transformation (FFT) of the steering wheel angle.
[TWV11] use an artificial neural network (NN) to determine the driver's current level of distraction. Using a three-layered Multi-Layer-Perceptron, a single numerical value as the level of distraction (ranging from one to five) is deducted from four input variables: speed, speed variation, steering wheel angle and steering wheel angle variation. An adaptive system taking driver distraction into consideration was evaluated as being superior in terms of perceived safety and usability to the non-adaptive version. Models based on neural networks have proven successful previously, e.g., [AA10].
[SP11] estimates CL complexity using both performance and physiological data in a simulator. As performance measures, the lateral position variation and steering wheel activity is observed. That data is then fed into a radial-basis probabilistic neural network (RBPNN).

Reaction time and time perception

Reaction time is a convenient way of measuring performance. [ME09] clearly shows a direct impact of driver and situational factors on brake reaction time (BRT) and acceleration / deceleration reaction time (ADRT). [CWSS08] measured the impact of distraction by mobile phones on the driver's reaction time. Many other examples can be found in literature.
As another interesting aspect, CL seems to directly influence the perception of time. In a user study, [BBW09] measured the difference between time intervals produced by a driver in different situations and compared the mean deviation from the actual time with the CL of the driver measured by other means. Results show a direct connection, i.e., perceived time correlates with actual cognitive workload.

5.4.3 Physiological Measures

Although usually used in medical research and medical examination for obtaining information of state and performance of major organs, we can also use physiological sensors for obtaining information about the state of the subject. Most suitable for our purpose are–for obvious reasons–parameters which can not consciously be modified by the subject. Some of the measures discussed in the following sections are described in more detail in [DK10]. For our purpose, it is important to find a completely non-intrusive method of measuring. Even small intrusions, like placing a sensor on a finger, which are easily accepted in a user study, are unlikely to find acceptance by the driver in every-day driving.

Heart Rate (HR) and Heart Rate Variability (HRV)
The heart rate (HR) is defined as the number of heart beats in a fixed time interval, usually a minute. Heart rate variability (HRV) is concerned with the degree and pattern of variation in a series of heart beats. The average heart frequency of a person is individual and age-dependent. It is also dependent on the subject's state of arousal. A deviation from the usual HR or HRV of a subject can be used as a measurement for stress or, in consequence, CL. There are no widely accepted standard values for HRV. [RFA09] investigate the application of HRV analysis on electrocardiography (ECG) data for assessing the arousal state of the driver. As a result, the presented measure proved suitable for detecting arousal (not for emotion recognition).

Respiration
Respiration consists of inhaling and exhaling air. Respiration frequency is like, the previously discussed heart frequency, age-, subject- and situation-dependent. While babies usually breathe between 30 and 40 times per minute, the frequency lowers with age to an average 8 to 20 times per minute for an adult. Table 5.2 provides a more detailed survey. Various sources vary in the exact number.
The respiration rate can decrease in a relaxed state and increase in (usually physically) demanding situations. A respiration belt can be used to measure the respiration rate.

Skin conductance (SC)
One of the fastest ways of physiologically measuring a human's state is through Galvanic Skin Response (GSR). It is also one of the oldest methods, dating back to the 19th century. The underlying principle is measuring the electrical resistance produced by the skin, i.e., the epidermis (upper skin layers).
[SMR$^+$02] details that when "an outgoing sympathetic nervous burst occurs to the skin, the palmar and plantar sweat glands are filled up, and the SC increases before the sweat is removed and the SC decreases. This creates a SC fluctuation." In the same article, the authors establish a correlation between both the number and the amplitude of skin conductance fluctuation and the level of arousal. The context of their research

Age	min value	max value
Newborns	30	60
Less Than 1 Year	20	40
1–3 Years	20	30
3–6 Years	20	30
6–12 Years	18	25
12–17 Years	15	20
Adults Over 18	12	20

Table 5.2: Average respiratory range by age in breaths per minute (sources: [LPTD09, DeB04])

is monitoring the state of a patient during surgery.

Temperature
Body temperature, or rather change of body temperature, indicates (among individual differences) the stress level of a person. Measuring body temperature is a relatively easy procedure, and flexible in terms of location for measurement.
[MCT+10] showed the connection between warning messages or obstacles shown in a driving simulator environment with the sudden change of body temperature.

Eye movement
Saccades are quick simultaneous movements of both eyes in the same direction. [MMG02] conducted two experiments on the connection between reflexive saccades and the subject's working memory. Evidence for a connection was found. This can be used for assessing CL. In a mobile environment–such as a car–the practical feasibility on the other hand is somewhat limited. Neither attaching an eye-tracking device to the driver's head nor the fixation of the head are beneficial in terms of safety or non-intrusiveness. But if the eye tracker is fixed to the dashboard, and the driver's head can move in the full range of normal driving activity, measuring eye movements becomes difficult and error-prone.

Pupil diameter
Mean Pupil Diameter Change (MPDC) and Mean Pupil Diameter Change Rate (MPDCR) are commonly used measures in pupil diameter analysis. [PKSH10] show in a driving simulator experiment that a correspondence exists between MPDC and driving performance under experimental conditions. Also, their results "indicate that the MPDCR shows promise as a pupillometric measure of cognitive load". It is especially useful for measuring rapid changes in cognitive load, and their experiments serve as proof-of-concept that reliable remote eye tracking in a car is possible.
On the other hand, [KPR12] showed empirically that size and luminance of objects in the viewfield can influence pupillary light reflection (PLR) and may obscure

cognitive-load related pupil diameter changes.

Voice analysis
[BJ99] discuss how CL is reflected in the user's speech. Symptoms such as sentence fragments and articulation rate are modeled into a Bayesian network for interpretation. The authors showed that assessing CL via speech input is indeed feasible. [MGJ$^+$01] extends this approach and examines the effect of time pressure and CL on speech empirically.
These concepts are very suitable for speech-based dialog systems; an IVIS on the other hand does not necessarily require speech input from the driver, and forcing the driver to speak might, especially in critical situations, add further distraction. Due to potential and likely background noise, the automotive environment is also not well suited for speech input.

Applicability of CL measures in an automotive context

As we discussed in this section, applying traditional CL measuring techniques is not always desirable in our domain. Important features are real-time conduction, immediate availability of results (e.g., results do not have to be entered manually in the system), and unintrusiveness. Table 5.3 compares the advantages and disadvantages of different approaches.

Measure	Real-Time	Immediate	Intrusive
Subjective	- -	-	- -
Performance	++	+	++
Physiological	++	++	- -

Table 5.3: Suitability of cognitive load assessment for real time automotive applications is limited.

5.4.4 Cognitive Load Estimation by Context

As shown in Figure 5.7, current CL might also be estimated using another path, i.e., by assessing the impact of the environment on the driver. Although the context might not be sufficient for an exact estimate of the driver's state, we can safely assume some factors to be influential to his cognitive demands. Driving on the highway or in dense city traffic is probably more demanding than driving on a quiet rural road. Driving at a moderate speed is less stressful than driving at very high speed or being stuck in a traffic jam. Also, environmental conditions such as noise level inside and outside the car can be measured and considered. The car's built-in information systems might keep a history of information presented to the driver, from which we can conduct the cognitive demand. A lot of information flooding the driver in a very short period of

Figure 5.7: Connection between situation, cognitive load and driving performance.

time is likely to raise his CL.
[Mül05] used Dynamic Bayesian Networks (DBNs) and data obtained from the car directly to generate a continuous estimate of the driver's load. In a second step, the DBNs were transformed into arithmetic circuits for efficiency reasons, especially considering the usually limited computing power of a vehicle. This concept can be adapted and extended to other information sources in order to increase the quality of the estimate.
This topic will be discussed later on in part III of this thesis.

5.5 Information Complexity Estimation

Given this background, we will now take a closer look at methods found in literature to analyze parts of a presentation and the effects of different parameters on cognitive load, reaction time, and performance.

Imbeau et al. performed an extensive user study on that subject [IWWC89]. The results of this study were analyzed again in more detail in [IWB93]. In a simulated vehicle, forty subjects were asked to read aloud words presented in eight second intervals on two displays which emulate written legends on an instrument panel while driving in night time conditions.
The characteristics of the words presented were varied in four different dimensions and combinations thereof. The variations include eight different chromaticities, two brightness levels, four character sizes, and two levels of word complexity.[1] The results of this study are presented in table 5.4.
As an example, figure 5.8 shows the effect of age combined with inverse character size on the glance time predicted by the model described in [IWB93]. The main goal of their work was to "provide designers with integrated quantitative performance data that will help them answer design questions and evaluate design alternatives." Using their model, user responses to various changes in parameters can be predicted "off-line", i.e., without the need of another user study.

[WK99] performed a study to determine whether different kinds of visual attention

[1]Imbeau et al. define word complexity by a combination of frequency of word occurrence in the English language and the number of syllables.

Effect	Dependent measures		
	Glance	Vocal	Lane deviation
Age	✓	✓	✓
Character size	✓	✓	✓
Color	✓	✓	
Brightness	✓	✓	✓
Word complexity	✓	✓	
Age x character size		✓	
Age x brightness	✓	✓	
Character size x color	✓	✓	
Character size x brightness	✓	✓	✓
Color x brightness		✓	
Character size x word complexity	✓	✓	
Age x character size x brightness	✓	✓	✓
Character size x brightness x color		✓	

Table 5.4: The effect of age, character size, color, brightness, word complexity and combination thereof on glance time, vocal response time, and lane deviation [IWWC89].

rely on a common substrate, i.e., whether or not there is a common underlying mechanism for visual attention. This question is important in the context of this thesis, since a common mechanism would serve as a bottleneck in visual attention and explain effects appearing when attempting to divide visual attention. Three different experiments were performed, each comparing an attentionally demanding task with an easier task using identical stimuli: (1) peripheral shifting, (2) object matching, and (3) a non-spatial conjunction task. Functional magnetic resonance imaging (fMRI) was used to determine activated brain areas. Two areas could be identified,[2] which were activated during these tasks, but not in a fourth experiment of comparing a difficult language task with an easier control task. Hence, it can be assumed that the two involved brain areas are required for visual attention and form a scarce cognitive resource.

[Wes99] performed two experiments of taxing selective attention processes on the efficiency of working memory processes in relation to normal aging. The results show that the presence of task-irrelevant information disrupted the working memory process, which could be measured to a greater extend in older than in younger adults. The effect of distracting information in terms of a higher frequency of intrusion errors could be observed in both younger and older adults. Memory-based errors were only significant with older adults. In conclusion, it is suggested that distraction disrupts the ability to maintain a coherent stream of goal-directed thought and action in gen-

[2]One of the areas is at the junction of intraparetial and traverse occipital sulci (IPTO), the other in the anterior intraparietal sulcus (AIPS).

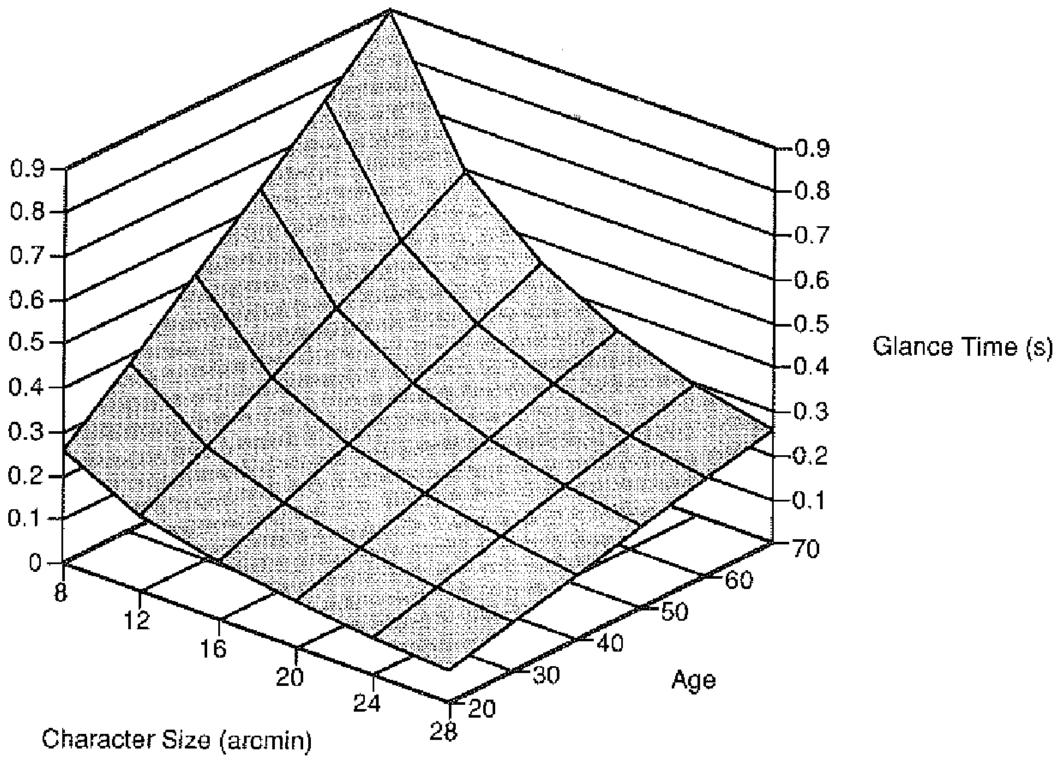


Figure 5.8: The effect of age and inverse character size on glance time [IWB93].

eral. Additionally, the encoding and retention of relevant information in older adults is affected. As a consequence, disruptions of information and presence of distracting information should be avoided when designing a safety-critical user interface such as an in-car HMI.

[LKLS99] performed a study aimed at investigating the driver's ability to detect the deceleration of the car ahead while executing a mobile-phone-related task. Participants were subjected to three situations: (1) looking at the car ahead (control), (2) continuously dialing a series of three random integers (divided visual attention), and (3) performing a memory and addition task (non-visual attention). Meanwhile, the car in front was decelerating at $0.47 m/s^2$. The impairment of the divided attention task in comparison to the control task was about 0.5s in brake reaction time and 1s in terms of time-to-collision. The performance while executing the non-visual task was slightly but not significantly better. As a conclusion, the authors claim that neither hands-free nor voice-controlled phone interfaces could significantly remove safety problems associated with the use of mobile phones in the car.

[BS01] investigate the effect of infrequent task-irrelevant deviations in the frequency of a tone on the perception of tone duration and a corresponding visual distraction paradigm. Subjects were asked to distinguish between short- and long-duration stimuli and instructed to press a button to long stimuli. Short and long stimuli were presented with equal probability, and in addition, both stimuli could be in standard version or two deviant versions. The deviant versions differed slightly in a task-irrelevant dimension. The study revealed significantly longer reaction time in cases when the distractor stimuli were present. Consequence for this thesis: It is not only important to present information properly and sufficiently long for the recipient to process them; it is equally important to avoid distraction in task-irrelevant dimensions as far as possible.[3]

[KMG+06] performed a study on the difference between vocal commands and virtual sound cues while navigating. The effects were observed both with and without cognitive load on the subject. Their hypothesis was that sound would cause less cognitive load than spoken spatial commands. No significant difference was found in the low-load condition, but significant difference in the high-load condition, where the navigation task could be completed in less time when guided by audio cues instead of spatial language. As a potential consequence to the field of automotive research, navigation systems should switch from spoken commands to well-known sound cues when the driver encounters high cognitive load.

Similar to the experiment of [LKLS99] on divided attention, [VAB11] investigated the effects of divided hearing attention. Subjects were asked to interact with an audio menu, while being exposed to another audio source. Options for distinguishing the audio menu from the additional audio source were (a) spatial audio techniques or (b) interrupting the additional audio stream. The additional audio stream was a podcast for high cognitive load or classical music for low cognitive load. Results of the study showed that, under low cognitive load, the spatial audio technique was preferred and the interruption technique significantly less considered. Conversely on high cognitive load, these preferences were reversed.

[DS11] describe an approach to speech-driven UIs for drivers. The authors discuss the impact of linguistic complexity on cognitive load. This measure can be used for estimating complexity of both written instructions and spoken information. This paper will be presented in more detail in the following related work chapter.

The work of [FD12] does not immediately concern the complexity of presented information, but rather distractions caused by interior lighting. Participants in a study were asked to perform two different tasks while sitting in a stationary car on a test road at night: (1) detect pedestrians on the road ahead, and (2) rate the subjective brightness of a reflected veiling on the windscreen. The results have implications on

[3] Of course not all possible distractions are under the control of the system.

how car interiors should be lighted, as the reflection of the windshield has a significant impact on the task of detecting pedestrians on the road ahead.

[RMW$^+$12] reports first results on an exploratory study. The objective is to assess the impact of typeface design on glance time on a multi-line menu display, which is one of the aspects not yet considered by [IWWC89] and [IWB93]. Data from 42 participants was collected. The results were consistent in terms of total glance time and number of glances were consistent. The authors recommend optimization of type face characteristcs as a low-cost method for reducing interface demand.

This survey shows that a substantial amount of research has been done on formal parameters which influence the complexity of a presentation or the environmental conditions affecting the driver's perception. This provides us with a solid foundation to build a situation-aware presentation toolkit. My own research in this area focuses on the extension of this foundational work by adding a formal analysis of the display layout and its complexity (cf. section 10.3).

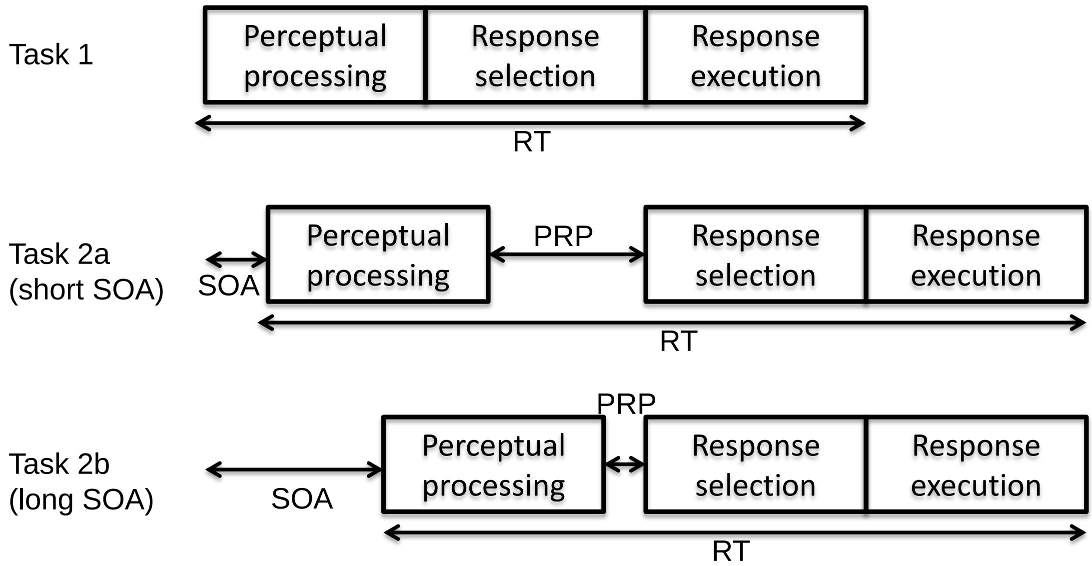
Chapter 6

Related Work

In the previous chapters, literature in the respective fields of scheduling, presentation planning, and situation awareness has been covered. Now, I will take a look at relevant publications over the last few years covering these fields in combination.

Managing in-vehicle distractions–evidence from the Psychological Refractory Period paradigm [HJC10]

In analogy to Endsley's model (cf. sec. 5.2), human task performance can be described as a linear sequence of three processing stages: (a) perception, (b) response selection, and (c) response execution. The central bottleneck model of task performance postulates that multitasking is possible in perception and response execution, but not in processing: A central response selection stage is in charge of processing and generates a bottleneck. If two tasks simultaneously access processing, one of them has to wait until the other one completes this stage. The resulting delay in processing the queuing task is called Psychological Refractory Period (PRP) effect (cf. fig. 6.1). [HJC10] examine this effect for in-vehicle distractions in order to determine the "task-free" interval required before a braking event to ensure safe braking: If another task is currently being in the processing stage, braking would be delayed, which results in severe security consequences. A study was performed with 48 subjects, i.e., drivers, who were split in six groups and were randomly assigned an in-vehicle task defined by stimulus (three levels) and response modality. On close proximity, a lead vehicle brakeing event occurred. The Stimulus Onset Asynchrony (SOA), i.e., the time between secondary task and the braking of the lead vehicle, was varied in order to determine the length of the PRP effect and the result in reaction time. As a result of this study, the existence of dual-task interference could be demonstrated and the necessary SOA could be determined as a 350ms interval.
In short: No secondary task should be executed 350ms before a brakeing event.
Although this paper offers some interesting results, practical implications are limited: One of the goals in designing driver assistance systems is to warn as early as possible



RT: Reaction Time; SOA: Stimulus Onset Asynchrony; PRP: Psychological Refractory Period
Figure 6.1: The Psychological Refractory Period effect under dual-task conditions.

in case of imminent danger or an upcoming braking event. In such a situation, the driver is informed immediately. No 350ms interval for planning ahead is available.

Enabling Micro-Entertainment in Vehicles Based on Context Information [AKS$^+$10]

There is a lot of idle time while drivers are stuck in a traffic jam or waiting in front of red lights. These times could be used to entertain the driver, under the conditions that (a) specially tailored content for short entertainment periods is available, and (b) waiting times can be predicted reliably.
[AKS$^+$10] performed an online survey assessing which forms of entertainment and which types of content are accepted by the driver as useful for in-car entertainment. 127 people participated in this survey. The result for the forms of entertainment showed a clear preference for audio content, followed by emails and audio and images. Videos, text and games were considered less suitable. Concerning the types of content, a clear preference for news was given, followed by weather, cartoons, information on nearby sites, and sports. Little interest was shown towards location-based advertising, stock information, or advertisement in general.
In order to enable this so-called micro-entertainment according to the results of the user study, the authors developed an algorithm for identifying traffic light zones (TLZ) as shown in figure 6.2 and proposed a method for estimating waiting times.
Regarding the recent development in Car-2-X communication, the latter will become obsolete over the next few years, as both cars and their environment, e.g., traffic lights,

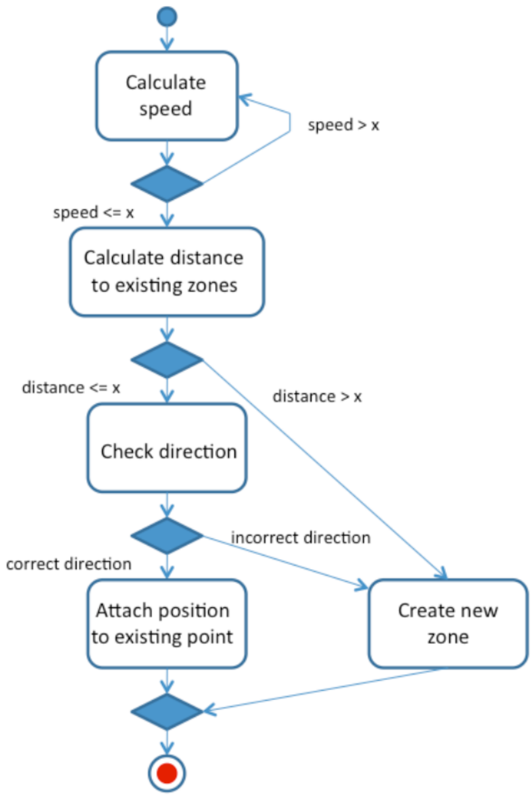


Figure 6.2: Algorithm proposed by Alt et.al. [AKS+10] for identifying traffic light zones (TLZ).

will become increasingly equipped with Car-2-Car or Car-2-X communication infrastructure respectively, so the car will know exactly the remainder of the waiting time.
This paper covers the less noted "left half" of the Yerkes-Dodson law (cf. section 5.3.1). Entertaining the driver prevents him from getting bored, which is also a factor decreasing driving performance. For the PRESTK system, I will introduce a similar approach.

On Timing and Modality Choice with Local Danger Warnings for Drivers [CCMM09]

Local danger warnings have to be delivered quiickly and efficiently to the driver. The choice of modality can play an important role in doing so. [CCMM09] performed an experimental study on the effectiveness of five different modality variants: speech, text-only, icon-only, and two combinations of text and icons.
Ten subjects attended the study. They were given a primary, visually demanding task as a substitute for the driving task. The task consisted in a "spot the differences" picture puzzle. Performance of this task was not evaluated, however, the subjects became

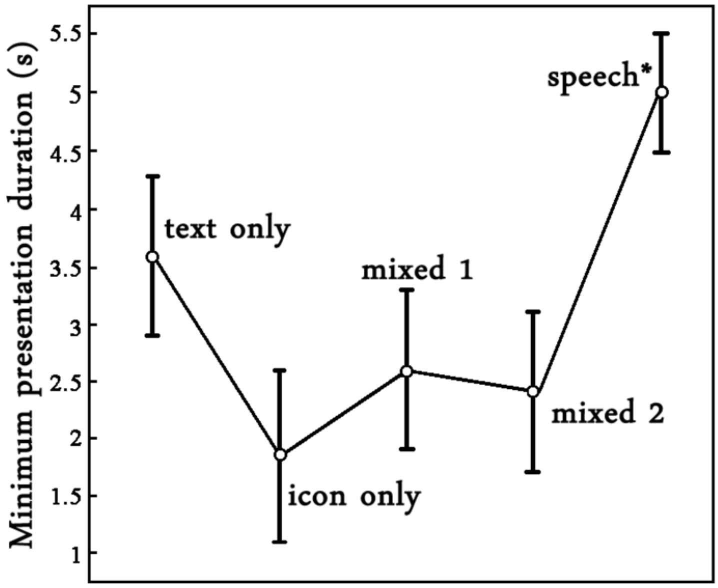


Figure 6.3: Timing results in seconds for visual modality variants. (*) For speech, the minimum presentation length is determined by utterance duration (source: [CCMM09]).

very engaged in this task.
While this primary task was performed, subjects got interrupted with warning messages in different modalities, consisting of three components: 1. type of obstacle, 2. location, and 3. distance. Two seconds later, the warning was repeated in the same modality, but with a 50:50 chance of being altered. Subjects were asked to select a button indicating whether or not the message had been changed. Three buttons were available for this task, labeled "same", "different" and "not sure". The short time period between first and second presentation of the message did not require any long-term memory; it was however sufficient to make sure that the subject knew what the obstacle is, where it is and how far it is.
By varying the presentation duration, the authors determined the time necessary to fully grasp all necessary details of the warning message. The results of this study are shown in figure 6.3. Text-only required the most time, while icon-only was perceived fastest. The two mixed versions are–not surprisingly–in between. For the speech condition, the minimum presentation duration was determined by the length of the utterance, but could be decoded reliably in nearly all cases.
This study presents us with essential information when building a presentation management system, as it provides numerical values for the minimal presentation duration in different modalities.

A Visual Programming Language for Designing Interactions Embedded in Web-based Geographic Applications [LEMN12]

A Visual Programming Language (VPL) is, in accordance of the authors with previous definitions, "any graphical language that lets programmers create programs by handling program elements graphically rather than textually." [LEMN12] present a VPL for designers to graphically describe human computer interactions with the aim of generating the corresponding web-based application. The VPL described is not intended for computer scientists or developers, but for a broader audience.
The focus of the VPL is on interaction, which is particularly interesting, since it is the most complex part of a program to describe. The domain chosen by the authors lies in the field of web-based geographic applications. The specified language is based on UML.
As a starting point, the underlying model of the geographic application modeling is described. It consists of three parts: 1. the *content* part, in which the geocontent is described by structured information consisting of type, value and possible representations; 2. the application *interface* as the visualization layer allowing the content to be visualized in various forms; 3. the *interaction* as communication between the user and the system.
After that, the visual specification of interactions is specified, based on UML-like diagrams. Interactions that can be specified are user action, external system reaction, and internal system reaction (such as projection, selection, or calculation).
The presented approach is put in context with the geocontent library WINDMash as application.
Finally, the authors evaluate their VPL in a user study, which obtained overall positive feedback as result.
Unlike the PRESTK system, the described VPL is not limited to presentation representation but also covers interaction. In our group, we approach this in the recently started SiAM project, and we include PRESTK as part of the dialog manager.

The Automotive Ontology: Managing Knowledge Inside the Vehicle and Sharing it Between Cars [FM11]

In this paper, the authors detail the ideas previously presented in [FE10]. The underlying assumption is that cars are increasingly equipped with different systems which meet the driver's and passenger's demand for safety, connectivity, and comfort. Personalization plays an important role in all of these systems, but acquiring information on people should be unintrusive and done with as little redundancy as possible. Once obtained, information should be available to different services, and it should persist even if this person is located in another seat or even switches to a completely different car. A central knowledge component can store all obtained information and provide the necessary facts about the user to all on-board systems. In order to build such a

component, the authors propose an Automotive Ontology and present a detailed user model and a context model in this paper.
Design considerations are discussed, and the underlying concept and relation types presented.
Four types of extensions are presented which play an important role in the automotive context and need to be considered on such a fundamental level that they affect the overall design of the ontology:
1. *Temporal Aspects.* For a qualified decision, it is important to know for each fact in the database when it was recorded and for how long this information is valid. Especially physical context, as well as the state of the user, are subject to fast changes. However, outdated information is not deleted, but instead labeled as “historical data” and kept for future reference. Three reasons for doing so are given: (1) keeping track of past events (e.g., draw a map of the car’s route), (2) extrapolate from previous data (e.g., weather forecast), (3) learn typical behavior and state of the user.
2. *Uncertain Knowledge and Reasoning.* Much of the collected information in a car can be uncertain for a variety of reasons, e.g., uncertain sensor data or inferences drawn from uncertain data. Handling uncertainty is a traditional field in the area of Artificial Intelligence and is no insurmountable problem. However, the ontology must provide some means to annotate facts with uncertainty values, and reasoning based on uncertain information must be handled carefully.
3. *Dynamic Locations.* Location modeling is obviously an important aspect in a dynamic environment such as a car. The ontology has to deal with that.
4. *Privacy.* In a shared knowledge base, privacy becomes a major concern. The authors propose approaches to handle this problem.
The ontology design is still in research and needs to be tested further. In this thesis, I propose an extension for modeling the driver’s cognitive load in several dimensions (cf. section 9.4).

Multi-modal Presentation of Medical Histories [Hal08]

This paper deals with a problem similar to in-car information presentation, but in a different field: The multimodal presentation of a potentially very extensive and very complex medical history in such a way that the important information can be easily perceived is a huge challenge. [Hal08] describes a visualization architecture which uses graphical devices as well as natural language and is integrated into a cooperative system for navigating through complex images. One of the goals is to show how generated natural language can be used to improve the usability of the graphical user interface, and, conversely, how the graphical user interface can be used to specify the content of the medical reports.
The author identifies a list of seven general requirements of the users and six specific requirements for the user interface which should be satisfied. A time-line based visual navigator using so-called *chronicles* is presented. Furthermore, the proposed proce-

dure of content selection, document planning, and aggregation of text documents is described. Finally, an approach to integrating text and graphics is presented.
As an outlook, the author describes upcoming work on handling incomplete data without implying misleading information. The uncertainties in the underlying data have to be marked as uncertain in the chronicle and displayed using different visualization modalities.
Overall, this paper deals with an amount of information comparable to that in a car, or even more, while at the same time the correct interpretation of this data has to be fast and efficient, because it results in potentially safety-critical human decision making. Some of the problems addressed here are also tackled by the Automotive Ontology introduced by [FM11].

A Presentation Model for Multimedia Summaries of Behavior [MF08]

[MF08] introduce the Multimedia Summarizer of Behavior (MSB). Its goal is the automatic generation of multimedia content which summarizes the behavior of a dynamic system at appropriate levels of abstraction. Additional information is included in order to facilitate understanding of the presented summary.
The aim of this paper is to provide a presentation model for the MSB. It includes a general strategy of discourse designed especially for the purpose of informing about the behavior of the dynamic system. In order to do so, a discourse strategy that connects informative components with rhetorical relations is developed.
The presentation model is the knowledge base of a hierarchical task network (HTN) planner [EHN94]. It includes planning tasks, planning methods, preconditions and operators. The planning tasks in the MSB are called communication goal, and the planning methods are the discourse patterns. A pattern consists of sub-goals which correspond to rhetorical relations. Hence, the whole discourse strategy becomes a hierarchy of communication goals and discourse patterns. The planner selects the most appropriate candidate from the discourse pattern and gathers additional information according to the rhetorical relations. Eventually, the goals are refined with sub-goals. A 3D viewer is used to visualize the information.
The authors apply this mechanism to the field of hydrology, for which it was developed, but explicitly state its applicability to road traffic networks.
For an in-car information system in the context of this thesis, the approach can be used to summarize traffic and accident information in a range nearby or on the route taken by a driver.

Generating Route Instructions with Varying Levels of Detail [ZHM$^+$11]

Navigation systems provide driving instructions by notifying the driver of impending lane or direction changes, which is called “turn-by-turn directions”. However, the majority of routes either start or end in familiar areas, and it has been shown previously

that the need for navigation instructions decreases in a familiar area. To make things worse, uncalled for navigation instructions disrupt and annoy the driver, taking his attention away from the driving task.
[ZHM$^+$11] propose a layered model of navigation instruction details, where the highest layer contains the original set of instructions as would be given by the navigation system, and the lower layers successively contain less detailed information. An application using these layers can at each point switch between layers and provide the driver with the appropriate amount of information, according to his knowledge of the area and information needs.
To evaluate the concept presented, the authors performed two user studies, one pre-study on the road, and a main study online.
In the pre-study on the road, the layer of detail (LOD) required by drivers for partially known routes was determined. 19 participants were asked to drive a distance between 20 and 50 kilometers, either starting or ending in a familiar area. Instead of using a navigation system, directions were given by a fellow passenger acting as an instructor. Instructions started at the most abstract level of pre-calculated directions, and, if not sufficient, the instructor switched to a higher LOD. Results showed that in the three sections of the drive (start, transition, destination), different LODs were requested by the driver.
In the main study online, participants were shown instructions for randomly generated routes, based on their area of familiarity. Again, the routes were divided into start, transition, and destination areas, and the LOD varied according to the familiarity of the subject with the area. The participants had to (1) rate instructions for each section, (2) compare the personalized instructions with Google Maps directions, and (3) arrange the personalized instructions according to their needs.
Results of the study confirmed the assumption that personalized instructions are preferred over general instructions.
Although limited to driving instructions, the underlying mechanism here can be used in a variety of other in-car systems and can be useful in identifying and reducing unnecessary information. Being aware of the driver's knowledge, for instance, can make a warning about a traffic jam which he encounters every day on the commute to work necessary, or might even lead to interesting new information not considered previously, e.g., to tell the driver there is no traffic jam today where he expects one to be. This concept has also been elaborated by [Hor11] as the "surprise me" principle.

A Study on User Acceptance of Proactive In-Vehicle Recommender Systems [BSW11]

There is a large amount of data which can be provided by modern in-vehicle systems, but if every piece of information which might be of interest is presented to the driver, the problem of information overload can become very serious. Especially, having to choose from a large amount of points of interest (POI) can be difficult.

Proactive recommender systems are a promising approach to reduce the amount of information.
The authors of this paper present a scenario for a gas station recommender and use the Technology Acceptance Model (TAM) [Dav89] to evaluate their approach in a user study.
A user interface is designed for the use on the central information display (CID) of the car. The design follows three requirements:
(1) *Unobtrusiveness.* By choosing a two-stage approach, the driver is first presented with the information in a subtle way (i.e., small icon) and only later on with a bigger pop-up. This enables him to notice the important information by glancing at it, actively requesting more information at his convenience.
(2) *Accessibility.* All relevant information should be easy to access.
(3) *Transparency.* Since recommendations are given without being requested by the user, the reason for the recommendation should be clear at all time.
The authors performed a user study with the gas station recommender system developed by these guidelines and evaluated it in terms of perceived usefulness, trust, perceived ease of use, and attitude to use. The results are overall positive.
The design guidelines presented in this paper are similar to the ones we used in developing the sim^{TD} HMI. However, [BSW11] use only one type of information to be presented in their study, while in realistic applications the combination of different kinds of information poses a much more difficult challenge to the system design.

Support for Modeling Interaction with Automotive User Interfaces [SPKS11]

The aim of [SPKS11] is to provide both a model and tools for rapid prototyping and evaluation of user interfaces in a car. Unlike the usual approach, to evaluate a user interface with a user study, the authors propose an adapted version of the Keystroke-Level Model (KLM), based on empirically collected data of typical in-car operations. The KLM is a model aiming at predicting how long an expert would need to perform a certain task on a specific desktop system. It is based on decomposing the task into low-level operations such as keystrokes or mouse-pointing, and adding up the assumed completion times of the sub-tasks to get the overall task performance time. The necessary operators (covering new, adapted, unchanged, and not applicable operators) are shown in table 6.1.
The exact timing value for each operation was determined in a user study. Model and values were combined in the MI-AUI software, which predicts the time necessary to complete a specific task on a specific in-car user interface.
An interesting aspect of this paper is the consideration of mental preparation time as part of task completion, as an additional perspective to the cognitive load topic.

Operator	Type	Description
Turn (T)	new	Turn movement on rotary controls
Homing (H)	adapted	Change between steering wheel and interface
Finger Movement (F)	adapted	Movement between controls
Keystroke (K)	adapted	Button press
Attention Shift (AS)	adapted	Shift between different parts of the interface
Mental Preparation (M)	adapted	Mental preparation of the driver
Response Time (R)	unchanged	System latency
Pointing (P)	N/A	Mouse pointing
Drawing (D)	N/A	Interaction with pointing device
Button Press (B)	N/A	Mouse button press

Table 6.1: Adapted operators from the Keystroke-Level Model (KLM) for evaluation of in-car systems.

The Impact of an Adaptive User Interface on Reducing Driver Distraction [TWV11]

[TWV11] discuss the adaptive In-Car Communication System (ICCS) prototype MIMI (Multimodal Interface for Mobile Info-communication) and its impact on driver distraction. The authors claim that current ICCSs attempt to minimize manual and visual distraction, but more research needs to be done to reduce cognitive distraction, where cognitive distraction is defined as including any thoughts that absorb the attention of the driver. Furthermore, few ICCSs today consider the driver's context. Cognitive workload, user performance, and situation awareness can be used as triggers in an adaptive user interface. In the MIMI prototype, the driver distraction level is determined using an artificial neural network (NN) using speed, speed delta, steering wheel angle and steering wheel angle delta as input parameters (cf. figure 6.4).
The parameters are obtained via the car's controlled area network (CAN). The NASA TLX questionnaire was used to measure the participants' cognitive load.
A user study based on the Lane Change Test (LCT) was set up, with 30 participants, comparing an adaptive version of MIMI with a non-adaptive version in terms of safety, driver distraction and usability. Participants had to perform ten tasks while driving in a simulator running on a PC. The display was projected at the wall and the participants used a simple steering wheel and pedals to control the driving simulator. All ten tasks were related to telephone or texting activities, such as dialing a number, confirming an incoming message, etc.
Overall, the adaptive version of the MIMI prototype had better results in perceived safety and usability.
Although the system in this paper is limited to communication functions, the general results are applicable to other in-car systems as well.

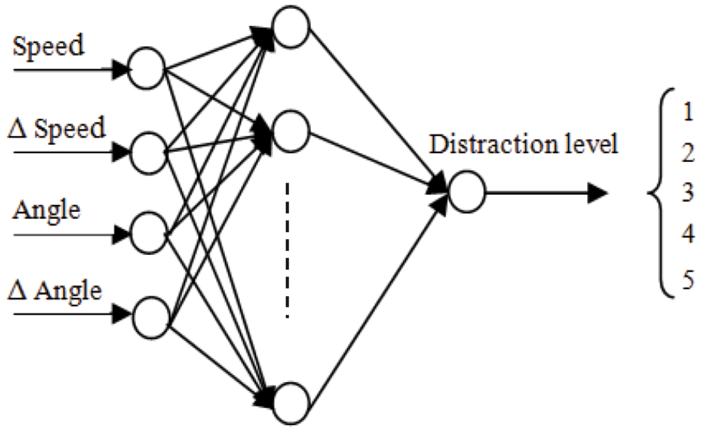


Figure 6.4: Determining distraction level using speed, angle, and variation thereof in a neural network (source: [TWV11]).

A Dynamic Content Summarization System for Opportunistic Driver Infotainment [RLH11]

[RLH11] present an approach to providing information to the driver at low-demand times using an embedded natural language processing (NLP) system. Prediction of low-demand times is based on previous work by other researchers, so the main focus of this paper is to propose and implement an algorithm to summarize a given text to a given length corresponding to the time length of the utterance by a speech synthesizer (cf. figure 6.5). Additional flexibility is added by providing the possibility of adjusting the speaking rate of the synthesizer.
As a result, the implemented algorithm was able to summarize given text accurately for a given time slice.

Driver Distraction Analysis based on FFT of Steering Wheel Angle [IKL11]

In a work-in-progress paper, [IKL11] propose an approach to deducing driver distraction using Fast Fourier Transformation (FFT) on the steering wheel angle spectrum over time. The steering wheel angle is a one-dimensional signal, with the advantage of being represented in every car and easily accessible using the car's CAN bus. Using three different distractions (telephoning, navigating, and inserting a CD in the CD player), the assumed distraction level was compared to the result of the steering wheel angle calculation, and a correlation was found. The correctness of these results will be analyzed and verified in more detail in future research.

Linguistic Cognitive Load: Implications for Automotive UIs [DS11]

[DS11] propose in this position paper an approach to speech-driven user interfaces for automotive drivers. The goal is the development of a speech interface that modulates

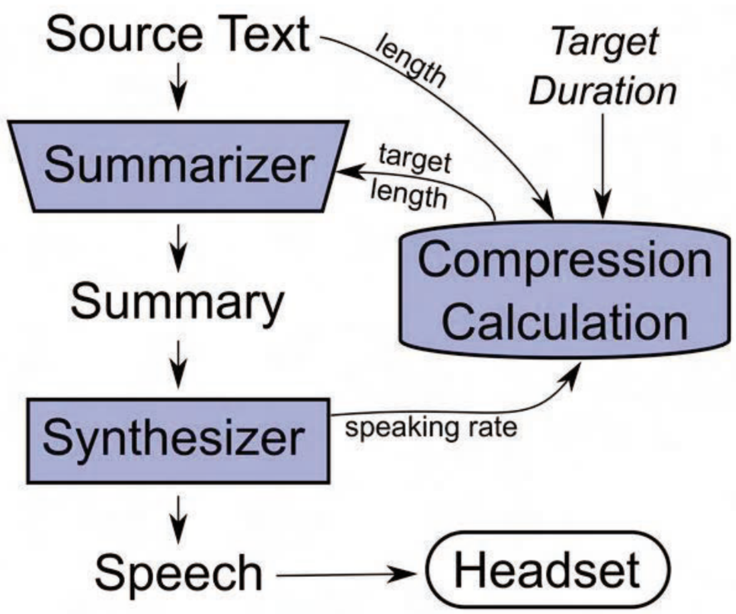


Figure 6.5: Dynamic text summarization to a specific target utterance length as proposed by [RLH11].

its complexity according to ongoing driving conditions by using psycholinguistic measures of language complexity. Their proposed research program includes three steps: "(a) study of the relationship between divided attention and different levels of linguistic representation (particularly semantic, syntactic, discourse); (b) design of a system to mediate in-car synthetic speech prompt production; and (c) measurement of the effects of in-vehicle dialog system interaction."
In the context of my thesis, two aspects of this paper are of special interest:
1. The literature overview of existing measures for cognitive load and the attempt to integrate them. Linguistic complexity can be measured using surprisal [Lev08], i.e., the negative log-probability of a word in a given context [Hal01]. Another defining measure is the accessibility of previous material when integrating it with new material (dependency locality theory, DLT [Gib00]). In a previous paper, [DK09] integrated these two approaches into a single model of sentence processing called "Prediction Theory".
2. The architecture proposed for a dialog complexity manager (cf. figure 6.6). An interesting aspect here is that not only is the system's linguistic complexity measured, but that of the driver's feedback as well. This approach integrates to a certain extent the model of Endsley (cf. chapter 5) of situation awareness with my proposed extension of system situation awareness (cf. section 9.1).

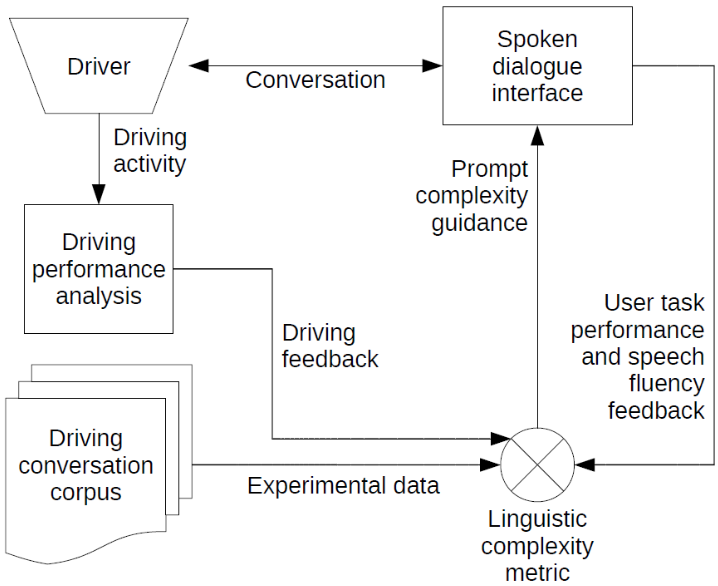


Figure 6.6: The dialog complexity manager architecture proposed by [DS11].

Utilizing Pupil Diameter to Estimate Cognitive Load Changes During Human Dialog: A Preliminary Study [KMPH11]

[KMPH11] explore the use of pupil diameter for evaluation of the effects of different dialog behaviors on the driver. Their claim is that less-structured verbal tasks are more representative for HMIs of the future than highly structured tasks such as question-answer tasks. In this paper, the authors present a study in which they test whether or not pupil diameter can be used to detect major changes in the cognitive load of the driver. For this test, participants are paired up, one being the driver and the other one the other conversant. The driver is driving in a driving simulator, with the primary task being to follow a lead car at 55 mph in daytime conditions on a road with some curves. The additional task is to remotely play a game of Taboo with the other conversant. The expectation is that, while performing the additional task, the cognitive load of the driver is higher and will drop right after the end of the game. Results of eye-tracking the driver during the study confirm this hypothesis.

Estimating Cognitive Load Complexity Using Performance and Physiological Data in a Driving Simulator [SP11]

[SP11] suggest estimating the driver's cognitive load using estimation models based on radial basis probabilistic neural networks (RBPNN). The design of the model is based on a user study where participants drove in a simulator and were asked to com-

plete auditory recall tasks in order to raise their cognitive load. The input of the RBPNN consisted of different combinations of four sources, two of them based on driving performance and two on physiological measures: 1. standard deviation of lane position (SDLP), 2. steering wheel reversal rate (SRR), 3. heart rate (HR), and 4. skin conductance level (SCL).
The result of the study shows that the best results were obtained using SCL only, followed by a combination of SCL and SDLP. The combination of SRR and HR scored worst.
These results are relevant in the context of my thesis, as they both put recent work by [IKL11] and [TWV11] in perspective and back my claim that a decent, unintrusive assessment of drivers cognitive load is difficult, whereas cognitive load estimation by context can and will be used as an alternative.

Enhancing Assessment of In-Vehicle Technology Attention Demands with Cardiac Measures [LB10]

[LB10] discuss the difference between attention assessment based on driving performance measures and attention assessment based on physiological measures, specifically heart rate measures in this case. While the authors agree that decrease in driving performance is a result of divided attention, they argue that the absence of diminished driving performance does not necessarily indicate that there are no attention costs involved in the current task performance: "While it may mean that attention resources are not shared between the two tasks, it may also mean that attention resources were shared but not overtaxed."
In conclusion, the authors support the use of additional cardiac measures in testing in-vehicle systems in order to obtain a more detailed model of their induced distraction level. However, they do not propose to use these measures in everyday driving.

Tell Me More, not just "More of the Same" [IBH10]

[IBH10] introduce the "Tell Me More" system, which, based on an online news story, searches for additional information on the same news item. Unlike previous systems, the classification of additional information is not just a binary classification ("new" and "not new"), but instead analyzed further and classified as supplying additional quotes, additional actors, additional figures, and additional information. The system uses five core modules, i.e., processing steps:
1. *Content Gathering.* Additional content is gathered using search engines.
2. *Content Filtering.* Documents which are either exactly the same as the original document or completely different are removed from the gathered content. Latent Semantic Analysis (LSA) [LD97] is used.
3. *Text Analytics.* Using statistical and heuristic models, for each new article entities, figures and quotes are detected. Also, a vector representation of each paragraph is

computed.
4. *Difference Metrics.* New documents are compared with the original document.
5. *Presentation.* Based on the difference metrics, new information is ranked and presented.
Although this paper does not have a direct connection to the automotive domain, the procedure presented here is relevant in the context of this thesis. By applying the same approach to two different variations of a message to be presented to a driver, we can compute the difference between the simpler variant and the more detailed variant. This difference in turn can be used as a foundation for a measure of textual complexity. This offers an extension or alternative to previous linguistic complexity measures such as [DK09]. However, the amount of text presented to the driver is at the moment rather small, but might not necessarily stay that way, if reliable methods for unintrusive delivery of longer messages to the driver can be found.

Local Danger Warnings for Drivers: The Effect of Modality and Level of Assistance on Driver Reaction [CMC$^+$10]

[CMC$^+$10] present a user study aimed at selecting the most appropriate presentation factors when presenting in-car warnings of emergent road obstacles. Two factors were investigated in this study: *modality* and *level of assistance*. Four variants for the modality were used: speech, visual and speech, visual warning with blinking cue, and visual warning with sound cue. For the level of assistance, two variants were provided: with or without action suggestion. The study was performed in a driving simulator and evaluated in terms of effectiveness, efficiency and satisfaction:
Effectiveness is the accuracy and completeness with which users achieve specified goals.
Efficiency is the amount of temporal and cognitive resources expended to achieve the goals.
Satisfaction is the user's positive attitude towards the use of the interface.
32 persons participated in the study, which was conducted in a real car with a driving simulation projected on the windshield and took about two hours. While driving, visual warning messages were presented on a display placed in the middle console of the car. The driving speed could be controlled at two levels, 120 km/h and 60 km/h, using the gas pedal and the brake pedal to switch between these two speed levels. The basic requirement however was to drive at 120 km/h wherever possible. Participants were asked to react to obstacles shown in a distance of between 800 and 1300 meters by changing to the offside lane if the driver was on the nearside lane, and change back after the obstacle, or to brake if the obstacle was on the offside lane or on the right roadside, and accelerate again after passing the obstacle. Warnings were presented before the obstacle became visible.
The study obtained detailed results and recommendations for the design of in-car warning systems, e.g., purely auditory or purely visual modalities are both insufficient

for presenting high-priority warnings, but the combination of both can be beneficial, especially speech and visual messages. Speech messages should be kept short and the critical information presented first.

Heart on the Road: HRV Analysis for Monitoring a Driver's Affective State [RFA09]

[RFA09] investigated the application of heart rate variability (HRV) analysis based on electrocardiography data for identifying possible threads in a driving situation. They recorded a "personal affective profile" of a driver by recording the same route at a specific time of day several times, using both ECG data and GPS data. Two possible fields of usage are proposed:
1. If, when driving the same route again at the same time of the day (varied slightly by traffic jams or other delays), at any point the HRV value deviates significantly from the previously recorded data, a warning could be issued to the driver, as this deviation might be an indicator of imminent danger.
2. Road segments could be classified by a service provider according to the arousal state of the drivers using these road segments.
The work presented in this paper adds another dimension to the discussion on driver stress: As the subjects were not feeling stressed during the experiment, the measured values can be used as an indicator for unconscious stress. Furthermore, higher arousal states could be measured at higher traffic volume, which supports my hypothesis that the state of the driver can–to a certain extent–be deduced by environmental factors (cf. section 9.3).

An On-Road Assessment of the Impact of Cognitive Workload on Physiological Arousal in Young Adult Drivers [RMC$^+$09]

[RMC$^+$09] present an on-road user study aiming at assessing the change in the driver's heart rate and skin conductance while the cognitive load is artificially increased using an auditory delayed digital recall (n-back) task. 26 subjects participated in a 30-minute drive and their data was recorded. As a result, the validity of using heart rate and skin conductance for measuring cognitive load could be confirmed. While heart rate increased almost linearly with the complexity of the secondary task, skin conductance raised significantly at the lowest level and did not change much with subsequent increase of difficulty. Results are shown in figure 6.7.

A Cognitive Schema Approach to Diagnose Intuitiveness: An Application to Onboard Computers [FII09]

Intuitive interaction is the result of unconscious application of prior knowledge to a new task or a new environment. [FII09] investigated in a user study "whether match-

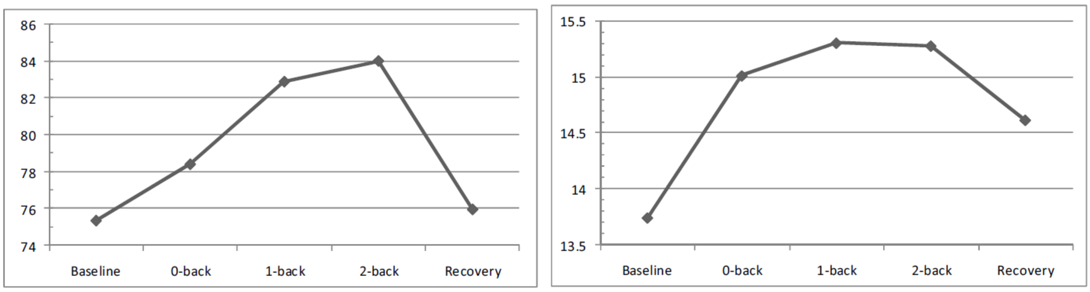


Figure 6.7: Mean heart rate (bpm) change (left) and mean skin conductance (micromhos) change (right) in different stages of the experiment as describes in [RMC+09].

ing all the states of an interface with a function (Inductive Group) in comparison to a word (Perceptive Group) fosters schema induction and interacts with familiarity." Subjects were 43 students who were novices in the use of onboard computers. The participants were instructed to read the target on an interface, press a "GO" button and then read the achieved state and click "YES" or "NO" respectively, depending on whether it matched the target or not. The results of the study suggested some interesting design recommendations: Few errors for both the Perceptive and the Inductive Groups indicate a transfer of prior schemas. More errors for the Perceptive compared to the Inductive Group reflect a positive induction effect.

Predicting User Action from Skin Conductance [LN08]

[LN08] performed a user study using a simple arcade game, while recording physiological data of the participants. They used a physiological measuring device called ISAX, which is capable of recording heart rate (HR), heart period variability (HPV), and skin conductance level (SCL). These data, along with information about the user's mouse clicks, were fed into a neural network. By using this neural network, predictions could be made on user actions during the game. Especially SCL turned out to be very useful, resulting in the network being able to predict the user action 2 seconds before it happened. This could be–if further refined–very useful in designing in-car warning systems, as predicting the driver's actions can give us a temporal advantage in the ability to warn or inform the driver of possible consequences. Or, to the contrary, we could suppress a warning if an upcoming reaction has been determined already.

Exploring Differences in the Impact of Auditory and Visual Demands on Driver Behavior [YRM+12]

The authors performed a study where the performance metrics of drivers carrying out a visual-manipulative task were compared with the performance metrics of drivers carrying out an auditory task in-vehicle while driving. These two types of secondary tasks resulted in the same level of self-reported workload, however perfomance-based and physiological measures showed different results. Their findings are consistent with Wickens' Multiple Resource Theory [WSV83] and thus provide some support for my classification of the driving task (cf. section 9.2.2).

Heart Rate Measures Reflect the Interaction of Low Mental Workload and Fatigue During Driving Simulation [THS$^+$12]

In this papers, the authors emphasize on the fact that heart rate measures can be used both to assess mental workload as well as fatigue, i.e., special precautions are necessary to distinguish these two concepts and obtain valuable results. In a study, monotonic mental workload was assessed under changing conditions of driver fatigue in a simulator setting. Using different cardiovascular measures, a distinction between tracking task and driving task was attempted and succeeded. The authors provide valuable information on the distinction of the several measures and advise which to use when.

Situation-Aware Personalization of Automotive User Interfaces [Gar12]

A simulator-based prototype of an in-car personalization system is presented. In a first step, user actions are recorded and associated with contextual parameters. Learned user preferences can automatically be executed afterwards. These features resemble the *Programming By Demonstration (PBD* paradigm in the PAN project (cf. section 3.2). In a second step, well-known personalized features such as navigational shortcuts or other automated routines can be proposed in the form of a list for the driver to choose from. The system is a prototype implementation which has not been evaluated yet.

Data Synchronization for Cognitive Load Estimation in Driving Simulator-based Experiments [MK12]

The growing trend to measure cognitive load during a user study and the heterogenous sources and measuring devices result in an increasing need to carefully synchronize data measured in an experiment. The authors of this paper propose an architecture for orchestrating a driving simulator, an eye-tracker and one or more physiological monitors. Both the hardware side (cf. figure 6.8) and the software side of the solution is presented. The system was tested and overall allows for a sampling rate of 133Hz of synchronized data measures during a study.

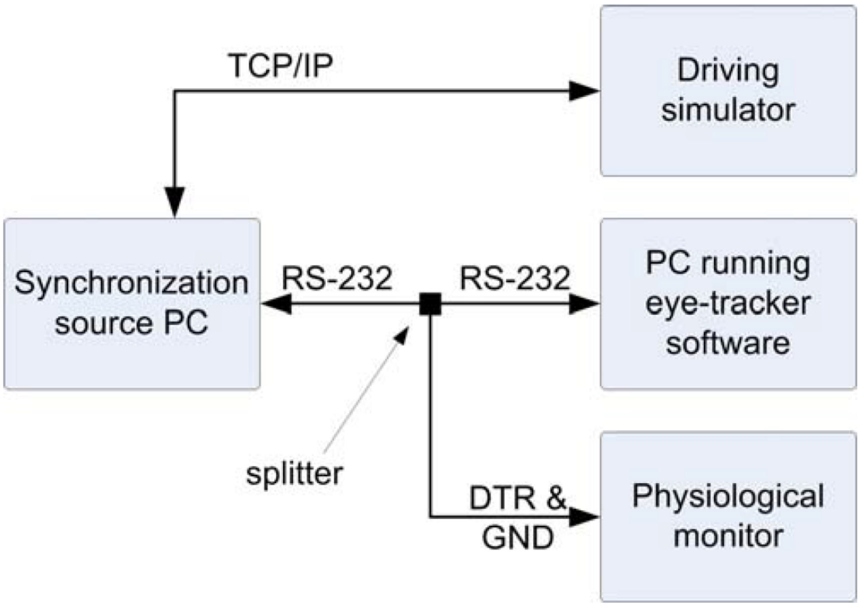


Figure 6.8: Data Synchronization in a driving simulator proposed by [MK12]

Emotional Adaptive Vehicle User Interfaces: moderating negative effects of failed technology interactions while driving [ALDG12]

This paper is located in the field of Natural User Interfaces (NUI) and explores the possibilities of emotion recognition in spoken user input. The aim is to use this information in their Voice User Help (VUH) in error recovery scenarios. By detecting negative emotions of the driver caused by system failures or suboptimal system behavior, the response of the system can be adapted in an attempt to increase the user's mood. In order to achieve that, the authors present an emotional taxonomy for the VUH in the two dimensions valence and arousal. Furthermore, the emotion recognizer tool based on the Praat software [Boe02] and the weka data mining tool [HFH+09] is presented. The recognition rate of the presented tool depends on the number of test data and ranges from 71% up to 90%.
It is by now still unclear which reaction of the system to an upset driver would be most beneficial. While part of the literature indicates that adapting the systems emotional state to the emotional state of the driver, others are in favour of counterbalancing the drivers mood to a neutral or positive value.

Summary

The table in figure 6.2 shows a classification of the previously discussed related work according to and in perspective of the goals of this thesis. As most of the papers discussed are from relevant conferences in the automotive field and are also very recent, we can observe current tendencies in automotive research here.
The most obvious trend is towards assessment of cognitive load, whereas the usage of this acquired information in terms of cognitive load assessment is not always very clear or has not been considered yet. Formalization mechanisms are few and far between. Concerns of standardizing an manufacturer-independent, in-car presentation language are barely noticeable [EFM12b].
Scheduling plays a minor role in research, but is sometimes used in connection with finding time windows for presenting tertiary task information and messages. Presentation planning is a topic of research, but still by far not as prominent as cognitive load. In cognitive load assessment, physiological measures are used to a certain extend, but mainly in an experimental setting, and a trend towards more unintrusive methods is obvious. Questionnaire-based methods are mainly used as additional measures in user studies. Perfomance based measures such as the unobtrusive measure and evaluation of steering wheel angle and speed are discussed prominently in three of the papers presented in this chapter.
The term "situation awareness" becomes increasingly important over the last few years.

There is no system yet to my knowledge containing all the features necessary to completely fulfill my requirements for a flexibly applicable system for situation-aware information presentation. The increasing and strong efforts in this direction are mainly aimed at cognitive load assessment and consideration while other parts for a comprehensive toolkit are usually neglected.
I will present my requirements in the following part III and describe my research towards a presentation system fulfilling them.

	Scheduling		Pres. Planning		Cognitive Load			
	dynamic	situation aware	Formalization	Complexity Est.	Assessment	in real-time	Formalization	Consideration
[HJC10]	✗	✓	✗	✗	(perf. based)	✓	✗	✓
[AKS+10]	✓	✓	✗	(✓)	by context	(✓)	✗	✓
[CCMM09]	(✓)	(✓)	✗	✓	✗	✗	✗	✗
[LEMN12]	✗	✗	✓	(✓)	✗	✗	✗	(✓)
[FM11]	✗	✗	✓	✗	✗	✗	✗	✗
[Hal08]	✗	✗	✓	(✓)	✗	✗	✗	(✓)
[MF08]	✗	✗	✓	✗	✗	✗	✗	✗
[ZHM+11]	✗	✗	✓	✓	by context	✓	✓	✓
[BSW11]	✗	✗	✓	(✓)	✗	✗	✗	✗
[SPKS11]	✗	✗	✓	✓	✗	✗	(✓)	(✓)
[TWV11]	✗	✗	✗	✓	perf. based	✓	✗	✓
[RLH11]	✓	✓	✗	✗	by context	✓	✗	✓
[IKL11]	✗	✗	✗	✗	perf. based	✓	✗	✗
[DS11]	✗	✗	✗	✓	✗	✗	✗	✗
[KMPH11]	✗	✗	✗	✓	physiological	✓	✗	✗
[SP11]	✗	✗	✗	✗	perf., physiol.	✓	✗	✗
[LB10]	✗	✗	✗	✗	physiol.	✓	✗	✗
[IBH10]	✗	✗	✗	(✓)	✗	✗	✗	✗
[CMC+10]	✗	✗	✗	✓	✗	✗	✗	✗
[RFA09]	✗	✗	✗	✗	physiol.	✓	✗	✗
[RMC+09]	✗	✗	✗	✓	physiol.	✓	✗	(✓)
[FII09]	✗	✗	✗	✓	✗	✗	✗	✗
[LN08]	(✓)	✗	✗	✗	physiol.	✓	✗	✗
[YRM+12]	✗	✗	✗	✗	all combined	✓	✗	✗
[THS+12]	✗	✗	✗	✗	physiol.	✓	✗	(✓)
[Gar12]	(✓)	(✓)	✗	✗	(perf. based)	✓	✗	(✓)
[MK12]	✗	✗	✗	✗	✗	✓	✗	(✓)
[ALDG12]	(✓)	(✓)	✗	✗	perf. & context	✓	✓	✓
PresTK	✓	✓	✓	✓	by context	✓	✓	✓

legend: ✓ present in paper; (✓) implicitly present; ✗ not present in paper

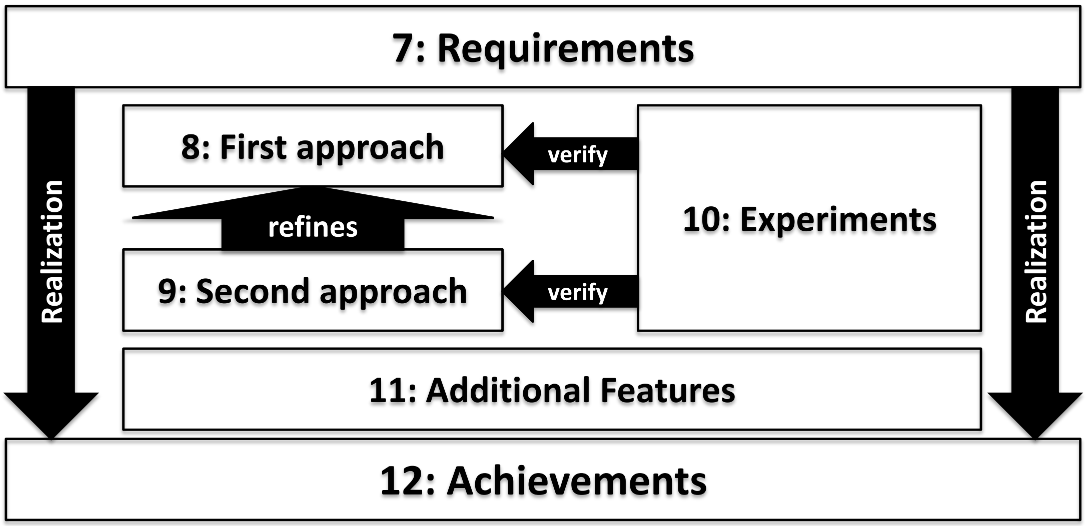
Table 6.2: Related Work

Part III

Situation-Aware Presentation of Messages and Infotainment Content for Drivers

Overview

This section of the thesis details my contributions to automotive research.
After discussing the theoretical background and related work in the previous part, chapter 7 starts by identifying the requirements for a system situation awareness in a car. Chapter 8 documents the first approach in the sim^{TD} system and the concepts developed for it. In extension, chapter 9 builds on the experience and lessons learned and introduces a second approach, which additionally covers system situation awareness. Both approaches are empirically evaluated in chapter 10. Finally, chapter 11 discusses some potential additions to the system presented and their prospects for successful implementation.
To wrap up this conceptual part of the thesis, the requirements are revisited in chapter 12, and the progress achieved is evaluated.



Chapter 7

Conceptual Requirements for System Situation Awareness

In the previous part of the thesis, various aspects of my work were described in terms of theoretical background and related work. In this part, I discuss my contribution to the research area, leading to the implementation of the presentation toolkit PRESTK, which is described in part IV.
First, the requirements imposed on the system need to be clearly identified. The requirements are partially given by the requirements of the sim^{TD} project, partially "lessons learned" from practical work in the project and complemented with the wish to go beyond sim^{TD} and add system situation awareness.
The basic problem is intelligent mediation as described in [OH93], adapted and extended to an in-car environment. Several, sometimes mutually independent information systems attempt to communicate important information over a limited set of communication channels to a recipient (i.e., driver) with a limited amount of cognitive abilities. As the information to be mediated is changing in a highly dynamic driving environment, and at the same time is potentially safety-critical, communication to the driver has to be as efficient as possible.
The problem of presenting mutually independent information systems can be considered as a search problem in three dimensions (cf. figure 7.1):

1. **Time:** To avoid simultaneous presentation of information which cannot in fact be processed simultaneously by the recipient, scheduling presented information on a time axis is necessary.

2. **Modality:** As the availability of information channels is limited, both spatially and technically, conflicts of two or more presentations occupying the same communication channel need to be avoided.

3. **Complexity:** Since the cognitive resources of the recipient are limited also, each piece of information has to be presented as simply as necessary and with

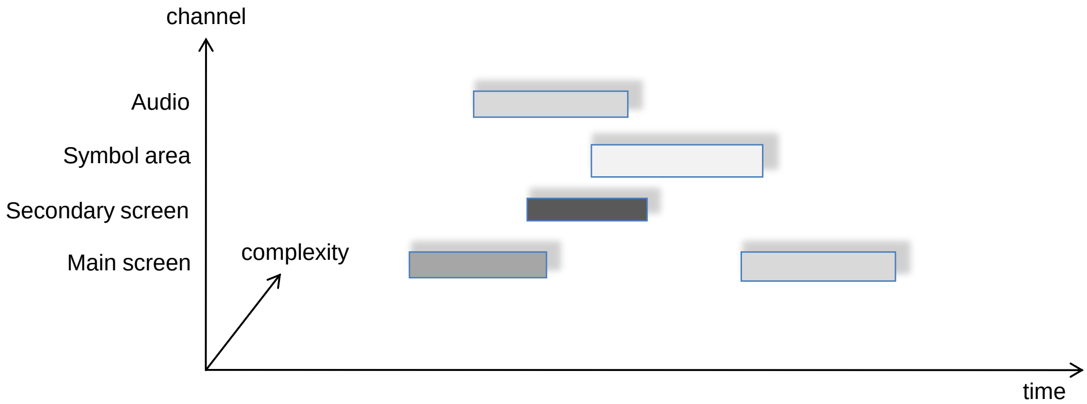


Figure 7.1: Three dimensions of presenting information.

as much detail as possible to maximize the benefit provided by the system to the driver.

Other factors also play an important role, e.g., the priority of the presented information, which makes the problem more complex in reality.
The requirements of the envisioned presentation toolkit include:

- **Extensibility.** It should be possible to include requirements not yet formalized. Not knowing what these requirements are, we can of course not be completely certain of satisfying these requirements, but an open, component based architecture is a suitable foundation.

- **Dynamic.** The toolkit needs to react to a dynamically changing environment. Information to be presented might change at any time due to the dynamic nature of the context, and no parameter can be taken for granted.

- **Anytime functionality.** Also caused by the dynamic context, the time available for decision making may vary and especially might be shorter than initially assumed. The system needs to react at any time with the best possible solution to a given problem.

- **System Situation Awareness.** In analogy to Endsley's model, the system needs to be aware of the current situation and react accordingly. This implies the following three requirements:

  - **Cognitive Load Assessment.** A suitable and sufficiently accurate model of the current state of the user is required at any time.
  - **Presentation Complexity Estimation.** The complexity of presented information for different presentation strategies needs to be estimated in order to assess its impact on the driver and choose alternatives.

  - **Cognitive Load Consideration.** The system needs to adapt to the driver.

- **Formalization of Acquired Information.** Pursuant to the requirement of extensibility, information such as driver's cognitive load needs to be formalized in an open format and made available in a way that is easily accessible to other systems.

In the next chapters, the conceptual background required for such a system is presented.

Chapter 8

Solution Provided for the sim$^{\text{TD}}$ HMI

The Human Machine Interface (HMI) as described in [CEM$^+$12] offers a central, common user interface for all functions and their respective use-cases in the car. This encompasses, on one hand, the visual and/or acoustic delivery of information and warnings to the driver. On the other hand, the interface receives input from the driver concerning the operation and configuration of the functions provided.

Input and Output Modalities

The sim$^{\text{TD}}$ HMI incorporates the following modalities:

1. Complex graphics, such as the schematic depiction of a multi-lane street with color annotations visualizing the traffic flow. The term "complex" refers here to the composition of the graphic itself, not to the driver's perception. The graphic is composed in a way that it is quickly, easily, and unambiguously for the driver to comprehended.

2. Symbols such as traffic signs and other pictograms. The symbols can be shown in big (filling most of the display area) or in small versions in the symbol area of the screen, where they will also be shown when the main display area is not (or no longer) available ("priority management").

3. Written text such as "Attention! Broken down vehicle in 400 meters!" (as a text string on the screen).

4. Speech output via TTS, such as "Attention! Broken down vehicle in 400 meters!" over the car speakers.

5. Audio signals, e.g., signal tones, to focus attention on the display of important information or to deliver a warning message.

6. A two-dimensional map with Points Of Interest (POI) and markers. Each POI on the map may contain additional information that can be displayed as an overlay on the screen.

7. Buttons for interaction.

8. Virtual keyboard for typing in information (while the vehicle is not being operated).

Presentation Models

Presentation models describe the manner in which given use-cases should be displayed on the HMI. The presentation models contain both information on the use-case to be displayed as well as at least one display strategy, covering the following aspects: How should a given presentation task be displayed if all HMI resources are available (preferred or optimal strategy)? How should the use-case be presented in case the main screen is already occupied by an active use-case or a use-case of higher priority (parallel presentation of information)? How and to what extent can important parameters of the use-case be adapted at runtime to suit the given situation?
By this we ensure that our scope of action is not limited to the boolean option "task can either be displayed or not", but can consider several fall-back options which ideally make use of the available resources. The presentation models (including graphics) were developed at DFKI in cooperation with the respective function development teams, based on their requirements.

Virtual Screens

The single display available to the driver in the sim^{TD} environment is split into three virtual screens which can be switched using on-screen navigation buttons.
The **main screen** is the default view on the HMI. It is mainly used for displaying warnings and travel information. It consists of a main area and a so-called symbol area. The main area offers the possibility for displaying complex graphics, text, large symbols, progress bars or a combination of these elements. The symbol area is suited for up to six iconic depictions of information which is of immediate relevance in the current context, but not as important in terms of priority as the content on the main area. The symbol area is visible on all three virtual screens.
A navigation map is the main component of the **navigation screen**. This screen is not the default view, but has to be selected manually by the driver. Additionally, warnings and points of interest (POI), such as events, parking lots, traffic cameras, etc. can be displayed. The navigation screen also shows the symbol area at the top.

The **option screen** offers functionality to change relevant system settings and provide feedback to the system.

Presentation Manager

The presentation manager is the core part of the system. Presentation requests are received and managed here. The scheduler is located here as its main building block. Via an interface to the system's logging function, we ensure that all events occurring in presentation management and scheduling are logged in a way that later evaluation is possible.

HMI Allocation Tree

The HMI Allocation Tree is a tree structure encoding the structure of available output channels as well as their current allocation (see Figure 8.1). The root of the tree is labeled "HMI" and encompasses the complete set of all available output devices or channels, which are its child nodes. In the case of the sim^{TD} HMI, these are the audio channel (here called text-to-speech, TTS, according to its main usage), the main area of the screen (MA) and the symbol area of the screen. The symbol area in turn has a selection of symbol places as children, which are leaves and not further differentiated. The main area is divided in several so-called logical screens, of which the main screen again is divided in main screen left and main screen right.
In this way, the allocation tree not only provides well-defined information on its structure, but also allows for annotation of the leaves with current or intended usage. When given a tree encoding the current state of the HMI allocation and a tree encoding a presentation strategy for a presentation task, the problem of detecting a potential conflict in attempting to apply the display strategy to the current allocation can be reduced to checking whether or not the set of annotated leaves is disjoint.

Resource-Constrained Scheduling Problem (RCSP)

RCSP consists of a set of tasks (activities) with fixed start times and durations, which should be executed without interruption on a given (limited) resource. Thus, conflicts occur when two or more tasks attempt to use the same resource simultaneously. The objective is to resolve all conflicts while modifying the set of tasks as little as possible. It is important to keep in mind that our problem, as presented here, differs significantly from the original RCPSP and thus the approaches mentioned previously (cf. section 2.4) cannot simply be adopted.

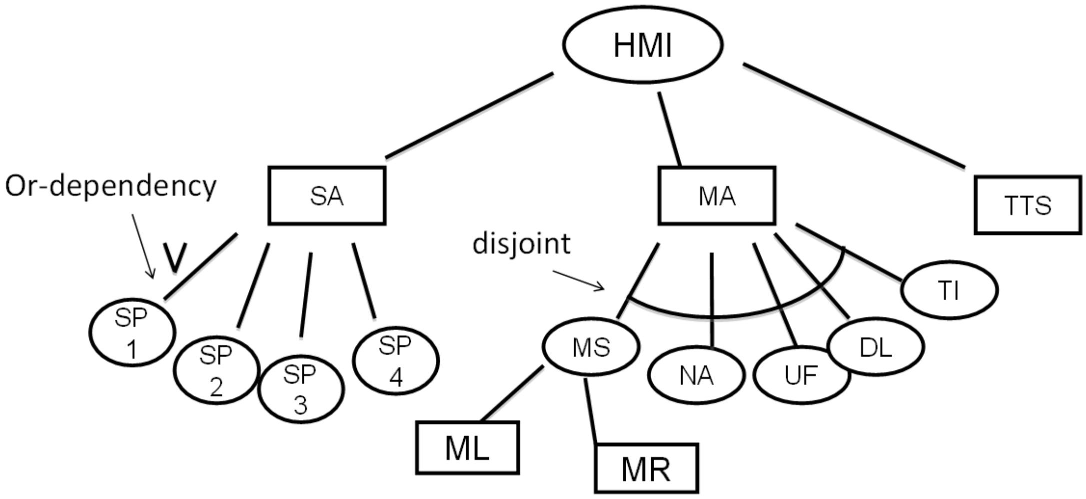


Figure 8.1: HMI Allocation Tree.

Presentation Task Scheduling Requirements in the Application Domain

My approach is based on the requirements of the on-the-road sim^{TD} project, hence tests and evaluation are performed in three different scenarios as close to reality as possible: the rural road scenario (basic complexity), the motorway scenario (intermediate complexity), and the urban road scenario (high complexity). I argue that, due to its run-time behavior, the graph-based approach is suitable for the particular application domain at hand. Results are presented in terms of quality of the solution (conflict resolution), runtime behavior and pruning effects to the size of the search tree. In addition to the scenarios derived from the actual field test, a hyper-real stress test is presented to demonstrate the performance of our solution.
I apply the RCSP approach to the problem of scheduling a large number of driver warnings based on Car-2-Car communication (also known as cooperative vehicles).
As a reminder: The sim^{TD} test field is located in the Frankfurt-Rhine-Main area of Hessen, Germany. With up to 120 vehicles and more than 100 roadside stations installed by the Hessian traffic centre (VZH) and the Integrated Traffic Management Center Frankfurt (IGLZ), Car-2-X communication is tested under real conditions. The Frankfurt-Rhine-Main area is an important German traffic hub with major traffic generators such as Frankfurt Airport, Frankfurt Trade Fair and the stadium. Large parts of the area are characterized by high traffic density and therefore allow experiments on all road safety and traffic efficiency functions under normal as well as high-load conditions. Figure 8.2 shows a map of the test field. It is comprised of three scenarios: I Rural road, II / III Motorway, and IV Urban road. The order of the scenarios represents their increasing use of the limited HMI resources. These assumptions are based on the number of applications active in the respective area, the traffic density

and the density of road site stations (stationary wireless communication units) used in the field test.
The *Rural Road Scenario* involves the federal rural roads B3 and B455, including main through-roads and junctions to the A5 (E451) motorway. A large number of traffic lights and road sections are equipped with roadside stations. The focus in this scenario is on testing the systems local danger alerts and information about construction sites (see Figure 8.3).
The *Motorway Scenario* involves the A5 (E451) motorway between junctions Bad Nauheim and Westkreuz Frankfurt as well as several motorways surrounding the city of Frankfurt. The region is divided into two parts with different densities of roadside stations (marked with II and III, respectively, in figure 8.2). The focus for the motorway scenario – apart from monitoring the traffic situation and identifying traffic events – is on traffic forecasting. Applications comprise an information and micro-navigation system for construction sites, end-of-traffic-jam alerts, traffic sign assistance, traffic information, and route deviation management.
The *Urban Road Scenario* comprises an important part of the main roads through the city of Frankfurt. This includes all relevant traffic generators, such as Frankfurt Trade Fair, the main station, Commerzbank arena, Niederrad, and Frankfurt Airport. One roadside station is placed at each major signalized intersection, mostly realizing distances of less than 500m. The focus of this scenario is – in addition to the general traffic situation survey – on local traffic-actuated traffic light control, location information services and testing of light phases and intersection / cross-traffic assistant systems.
Figure 8.3 gives an overview on the complexity of the scenarios and specifies the role of RCSP in the given application context. A large number of presentations try to access (potentially) simultaneously a limited resource (the Human-Machine Interface, HMI). Not all requests are equally important. Output channels are distributed over different modalities: multiple visual channels (symbols and main screen), auditory outputs, and text-to-speech (TTS). Hence, unlike traditional presentation planning, we look at a highly dynamic scenario in which presentation requests come in at any time during runtime. As indicated by the dashed triangle, a second resource limitation has to be taken into account: the limited cognitive resources of the driver. Here, questions should be addressed like, for example, how long the minimal display time for a piece of information should be, or whether or not overlapping presentations on different channels (TTS for one message and simultaneous display of another message) is beneficial. However, this is beyond the scope of this thesis. I refer the interested reader to [CCMM09, CMC$^+$10, MCT$^+$10].
Presentation requests are issued by the "functions" listed on the right side of Figure 8.3. Functions act as individual subsystems that have no knowledge of the available resources or the activities of other functions. Presentation requests are issued ("announced") as soon as the need for a warning becomes apparent. Hence, the sched-

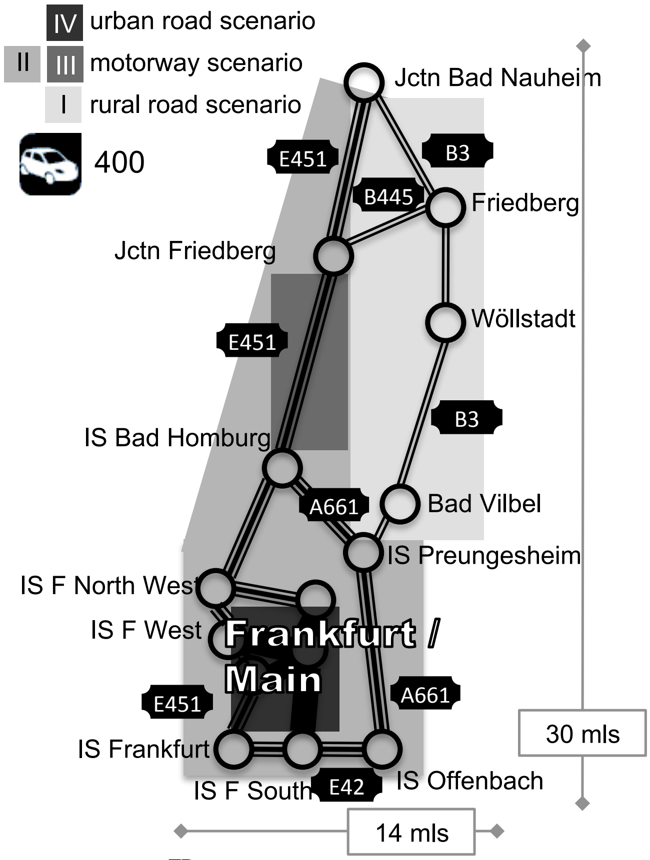


Figure 8.2: The sim^{TD} test field around Frankfurt/Main, Germany.

uler can allocate the HMI resources beforehand (planning time does not delay crucial warning messages). Nevertheless, the scenario involves a multiplicity of presentations and is highly dynamic.

Proposed Graph-Based Solution

In short, our problem is a dynamic set of tasks (presentation requests) simultaneously trying to access a limited resource. Tasks have a start time, duration, a hard minimal duration, a priority, and a desired resource (presentation strategy). We look for a solution which resolves resource conflicts while modifying the tasks as little as possible. A more formal definition is given below.

I propose a two-step graph-based solution to the problem [EM12]. Step one is transforming dynamic planning to static planning at runtime. Dynamic planning thus is by construction equivalent to having a series of static planning problems. The main problem here is to identify subsets of the set of presentation tasks and the appropriate order of resolving multiple different conflict sets. Identifying overlapping presentations is

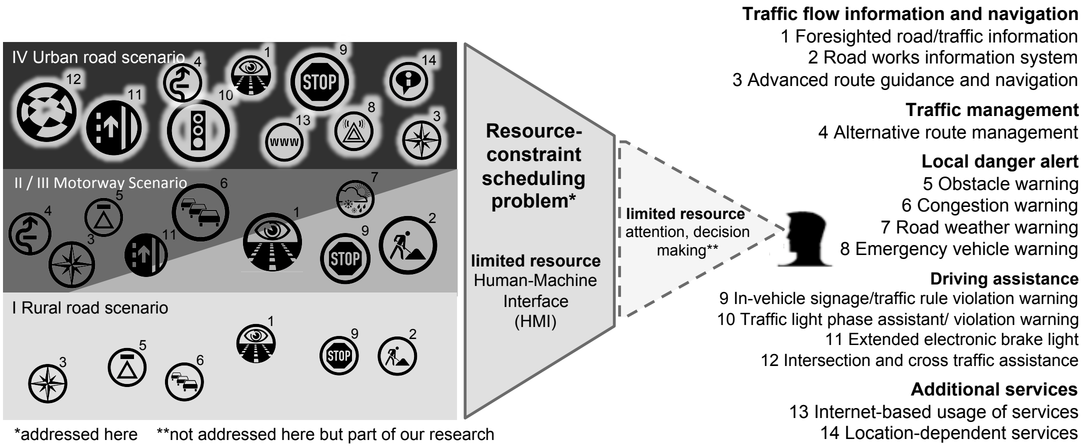


Figure 8.3: Complexity of the scenarios I – IV and the role of RCSP in the given application context.

a straightforward task at first sight. However, since the solution to that conflict could again cause other conflicts, the appropriate set of tasks to be included in the conflict set is not always obvious. Here we have to solve the trade-off between scope of the solution and runtime, which depends heavily on the number of tasks in the conflict set.

Step two of our approach is transforming our problem into a tree search problem for which well known algorithms can be applied.

Let $\mathcal{T}$ be a set of presentation tasks $\mathcal{T} = \{t_1, t_2, ..., t_M\}$. Let each presentation task t_i consist of a start time, an end time, a priority, and additional information about the presentation, such as rendering information, distance to event, etc. which are not of immediate relevance for the scheduling.

Definition 2 (Scheduling problem RCSP)**.** *The RCSP $\mathcal{P}$ is defined as:*
$\mathcal{P} = <\mathcal{T}^C, \mathcal{M}>$ *where*

- $\mathcal{T}^C$ *is a set of conflicting presentation tasks,* $\mathcal{T}^C = \{t_1^C, t_2^C, ..., t_M^C\}$.
- $\mathcal{M}$ *is a set of modifying actions (short: modifications) as conflict resolving strategies,* $\mathcal{M} = \{m_1, ..., m_N\}$

In our implementation, modifications are postponing, preponing, shortening (beginning or end of task), switching resource (not considered here) and canceling a task.

Definition 3 (Search tree)**.** *We define a search tree $\mathcal{S}(\mathcal{P})$ as follows:*

- *The root R of the tree contains the set of conflicting tasks* $\mathcal{T}^C$
- *Each edge e is a pair of a presentation task and a modification applied to it,* $e = <t_i^C, m_j>$

Algorithm 1 Scheduling using BFS tree search

```
solution ← nil
minPenalty ← ∞
i ← 0
N_0 ← {R}
while |N_i| > 0 do
   for all n ∈ N_i do
      for all m ∈ M do
         if applicable(m,n) then
            n' ← apply(m,n)
            N_{i+1} ← N_{i+1} ∪ {n'}
            if isSolution(n') and p(n')<minPenalty then
               minPenalty ← p(n')
               solution ← n'
            end if
         end if
      end for
   end for
   i ← i+1
end while
```

- *A cost function p(E), which defines a positive integer penalty for each edge, based on the severity of the modification (see Definition 7).*
- *Each node is the result of the modification of the previous edge to the parent node*
- *The penalty p(n) of a node n is the sum of the penalties of all edges on the branch leading to it*
- *A node n containing a presentation task set $\mathcal{T}$ without conflicts is a leaf.*

By design, the leaf with the lowest cost is the best solution to our problem. Using the tree $\mathcal{S}(\mathcal{P})$ described in Definition 3, we can now search the solution with the Breadth First Search (BFS) Algorithm 1.

Pruning In order to reduce complexity and runtime, several pruning mechanisms are applied in the algorithm: A branch with a total penalty bigger than the best solution found so far minus the minimum action penalty will be pruned; a branch that encodes a solution already encoded in another branch (maybe with permutated actions) will be pruned. Additional implicit pruning is achieved using the function *applicable(a,n)* (see line 8 of Algorithm 1):

Definition 4 (applicable(m,n))**.** *A modification m is applicable to a node n if and only if:*

- *n is not a leaf*
- *applying m to n will not create another conflict*
- *m will not shorten any task below a given minimal display time*
- *p(n') < minPenalty*
- *n'* $\notin \mathcal{N}_j$, $0 \leq j \leq i+1$
- *m will not move any task completely out of its original scope*

Anytime Behavior Our algorithm shows an "anytime" behavior, i.e., it incrementally finds better plans [BBM07]. Once a solution is found, its metric value can be used as a bound for future solutions.

Lemma 5. *Algorithm 1 produces a result and terminates.*

Proof. Any conflict set can be solved at high penalty by canceling all but one task. By design, this solution will be found at depth M-1 when no better solution is found before. Since no path is followed with a higher penalty than the current best solution, the depth of the tree is limited and the search space finite. A tree search on a finite tree will terminate by design. □

Theorem 6. *If Algorithm 1 terminates and a sound (or no) pruning has been used, then the last solution found is optimal.*

Proof. We have shown in Lemma 5 that the algorithm produces a result and terminates. Since the algorithm does not prune any path with lower penalty than the current solution (e.g., follows all paths possible bearing a better solution), the best solution can never be pruned. Since the current best solution is always stored as result, the result must be optimal. □

The metric we use here is a penalty of the actions needed in order to modify the conflict set to a conflict-free set. It is defined in more detail below.

Conclusion

The Resource-constrained Scheduling Problem (RCSP) applied here is, just like the well-known Resource-Constrained Project Scheduling Problem (RCPSP), a combinatorial optimization problem. The automotive domain, in which we encountered this problem, is highly dynamic and requires fast responses, e.g., an anytime behavior of the algorithm tackling this problem. I presented a tree-search based solution for the RCSP. I will analyze its runtime behavior, solution quality increase and space consumption in section 10.1. Although the algorithm is space-consuming when running until termination, very good results are produced fast and the calculation can be stopped then. The evaluation shows that the solution presented is suitable for the scenario at hand.

Chapter 9

Introducing System Situation Awareness

In chapter 8, I described our work on sim^{TD} with emphasis on my own contribution: the RSCP scheduling problem and a tree-search based solution to it, which fulfills the requirements of presentation management in the in-car environment, i.e., providing the best possible solution as quickly as necessary.
The solution provided solves the given problem in the area of scheduling and presentation planning. As a next step, I go beyond that and add situation awareness.
Situation Awareness, according to Endsley (see chapter 5), means "knowing what is going on around you." Although Endsley's definition primarily aims at the human operator's understanding of his surroundings, we can extend it to the system as well: One of the contributions of this thesis is the introduction of "System Situation Awareness," which is based on Endsley's model.
As the aim is to minimize distraction and thus to increase the performance and safety of the driver, the parts of the situation the system needs to be aware of are twofold:
1. What is the current cognitive load of the driver? Which cognitive capacities are available for him to process the next incoming information?
2. How complex, or to put it in another way, how cognitively demanding, is the information to be presented to the driver? How can it be modified to be either more detailed or easier to understand (in terms of the previous question)?
In order to process this information, we also have to include some means of formalizing the results of these questions, e.g., by annotating the presentation representation language used. Furthermore, information on the driver's state needs to be formalized as well and stored in the user model, e.g, we will have to extend the KAPcom described in [FE10]. Finally, we will wrap up the results to achieve the system situation awareness mentioned above. Figure 9.1 shows the two lines of research leading up to achieving situation awareness. At the same time, it summarizes how the remainder of this chapter is organized: The numbers on the left and right of the graph refer to the

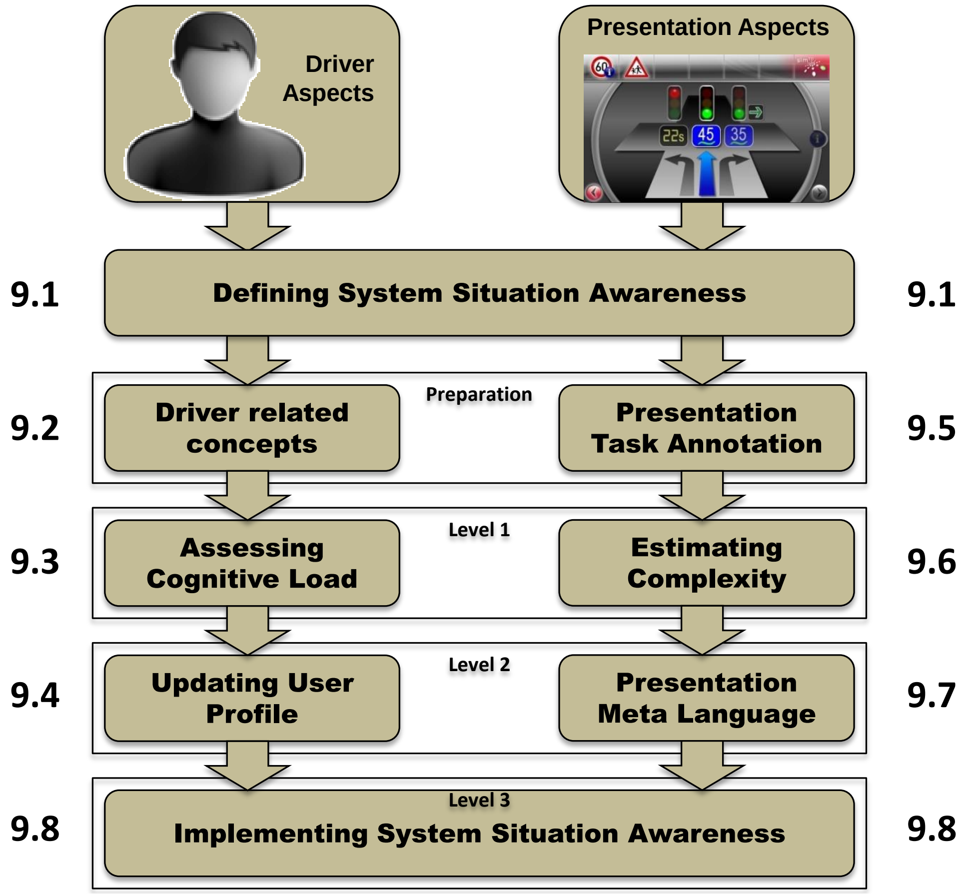


Figure 9.1: Paths leading to System Situation Awareness.

section where this part of the overall goal is achieved.

9.1 Defining System Situation Awareness

In chapter 5, I mentioned that Endsley's definition of situation awareness, although applied specifically as the operator's awareness of his environment, is sufficiently broad to be applied to a system as well. Taking up on that thought, I introduce the concept of *system situation awareness*, i.e., the awareness of the system of its environment or rather the extend to which the environment and the user are represented in and considered by the system.

Figure 9.2 shows an adapted version of Endsley's model. The three levels of situation awareness (perception, comprehension, projection) are replaced by the three levels of system situation awareness.

Level 1, assessment of user and context, in our case covers cognitive load assessment and presentation complexity estimation. The acquisition of data is the system's equiv-

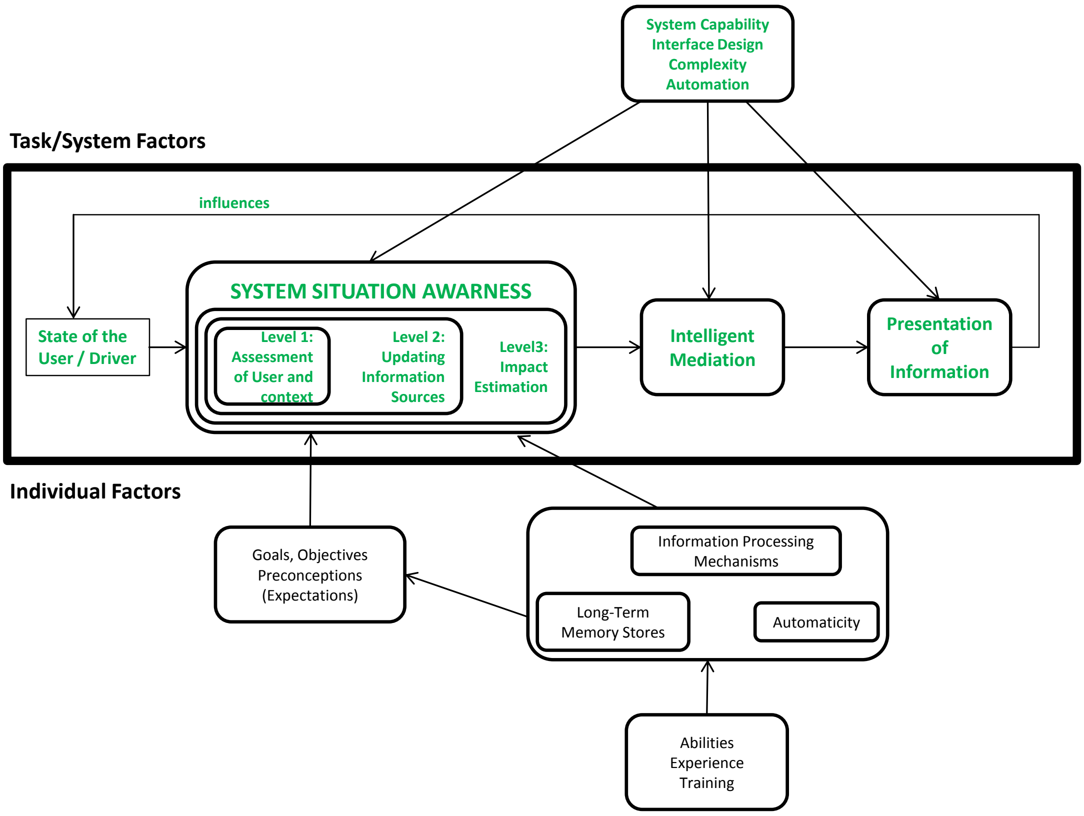


Figure 9.2: System Situation Awareness as a special case of Endsley's model of Situation Awareness.

alent to human perception.
Level 2 includes updating, storing, and annotating information obtained in level 1, in analogy to human comprehension.
Level 3, comprehension in the original model, is replaced by impact estimation.
In consequence, we see that human decision-making actually is replaced by the concept of intelligent mediation. Instead of performing actions, the presentation of the selected information takes place and influences the state of the user, closing the circle, as this is the state to be assessed by the system situation awareness.
The outer parts of the Endsley diagram stay mostly unchanged, except for removing the connections from individual factors to intelligent mediation and presentation of information, as these now are only indirect connections.

9.2 Formalizing Driver-Related Concepts

The key to processing information is an exact formal representation, especially when dealing with real world phenomena. Before we get to cognitive load estimation, an approach for quantifying the underlying concepts has to be defined.

9.2.1 Formalizing Cognitive Load

It is important to keep in mind that whenever we reduce a complex concept such as the cognitive state of a person or the effort required to steer a car to a single number, we lose qualitative information. In consequence, formalizing concepts has to be approached with care in order to obtain a model, which resembles reality as closely as possible.
As a start, I introduce a distinction between cognitive demand, cognitive load, and cognitive capacity (cf. figure 9.3).

Cognitive Demand

Cognitive demand encompasses all states, events and information which are perceived by a person and diminish his cognitive processing resources. Cognitive demand can be caused by primary or secondary tasks, environment, or system interactions of different information complexity. The task demands have a temporal dimension, and they can have cognitive or perceptual parts. Factors from the environment include information that it is either static or dynamic, i.e., changing more or less rapidly. Furthermore, the environment can be familiar or unfamiliar, and haas a certain visual complexity.
The term *cognitive demand* is used here for sources of cognitive load, to distinguish it from the aggregated cognitive load of the recipient. However, in literature, this concept is sometimes referred to as cognitive load as well.

Cognitive Load

Cognitive load–in my definition–refers to the person and is the aggregation of all the cognitive demands received by that person. Due to the complexity of human perception and its modeling, this aggregation should not be confused with a mathematical sum. Cognitive load is an influential factor for driving performance and in turn potentially critical, since it might affect passenger safety.

Cognitive Capacity

If we use a computer metaphor, the cognitive demand roughly corresponds with the load caused by a single process, and cognitive load is the current overall load of the machine. To further this metaphor, we might add that we do not know exactly how many CPUs the given machine has or how its load balancing works, i.e., we can not just simply add up the numbers of the cognitive demand to calculate the overall cognitive load. We can (and will) however use the sum of the single cognitive demands as an estimate for the upper bound of the cognitive load.

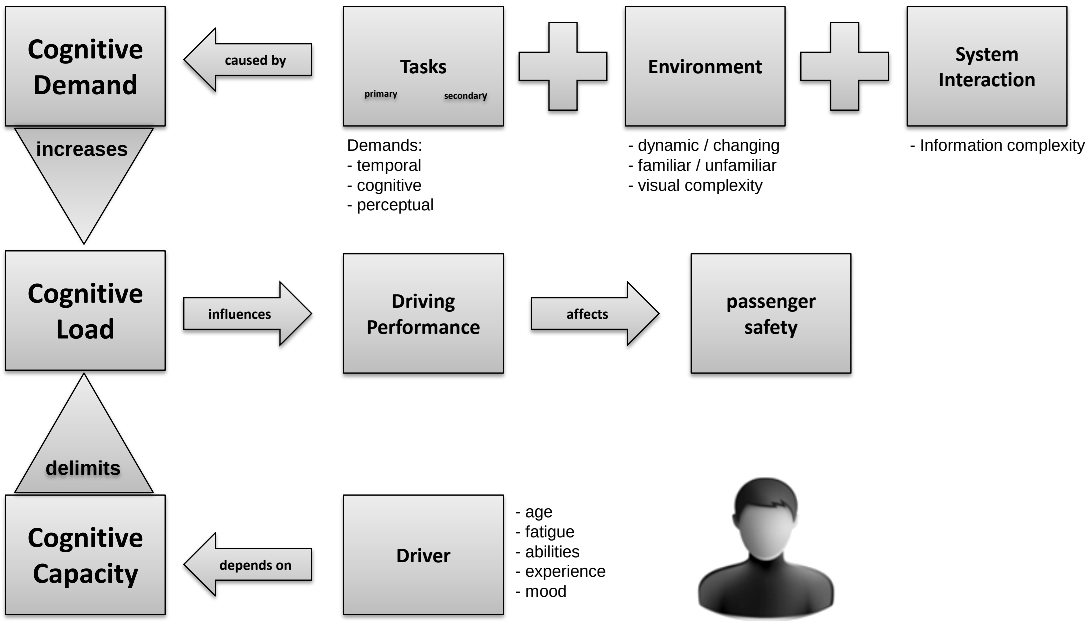


Figure 9.3: Connection between cognitive demand, cognitive capacity and cognitive load.

Cognitive capacity now refers to another potentially unknown variable, e.g., at what point of the scale the 100% load limit is reached. Due to individual differences, the cognitive capacity varies across individuals and different factors to be discussed later. Another important factor is the ability of human beings to adapt their performance level in times of higher demands.

9.2.2 Elements of the Driving Task

To formalize the driving task, we need to decompose it into its elements and analyze the single activities connected to it. [Sch93] derives five major elements of the driving task as an analogy to the basic aircraft crew workload functions described in Federal Aviation Regulation 25. These elements are vehicular guidance, navigation, communication and social skills, operating and monitoring systems, and command decisions. In figure 9.4, I refine this concept to a more detailed level of sub-elements.
Vehicular guidance in essence is determined by visual tracking and motoric/haptic reaction. During most of the driving time, maneuvering is performed, usually in lane keeping or lane changing, but also in turning or parking. The frequency of speed adjustments depends on the driving environment. It occurs less often on a quiet rural road than in busy urban traffic. Reaction to obstacles is also a part of maneuvering.
Navigation can achieve different levels of difficulty. In familiar areas, it becomes an automatism and does not require many resources. In unfamiliar areas, however, the complexity of navigation depends on the tools at hand, ranging from remembering a verbal description over written instructions and paper maps up to a navigation sys-

Driving Task			visual scanning, tracking, and monitoring	auditory tracking and monitoring	motoric tracking and haptic interaction	decision making	information recall	information processing / reaction	spatial estimation and interpretation	linguistic processing
Vehicular guidance	Maneuvering (lane keeping, turning, parking)		✓		✓	✓		✓	✓	
	Adjusting speed		✓		✓	✓		✓	✓	
	React to obstacles, traffic signs, other road users		✓	✓	✓	✓	✓	✓	✓	
Navigation	familiar area	memorized map	✓				✓		✓	
	unfamiliar area	paper map / written instructions	✓							✓
		navigation device	✓	✓	✓	✓		✓	✓	✓
Communication & Social Environment	Entertainment system		✓	✓	✓	✓	✓	✓		✓
	Co-driver and passengers		✓	✓		✓	✓	✓		✓
	Mobile communication (cell phone, etc.)		✓	✓	✓	✓	✓	✓		✓
Operating and monitoring systems	operating	critical (brake, steering wheel,...)	✓		✓	✓		✓	✓	
		uncritical (air conditioning, ...)	✓		✓					
	monitoring	critical (speed, ...)	✓	✓		✓	✓	✓		✓
		uncritical (mileage,...)	✓	✓		✓	✓	✓		✓
Command decisions	how to approach intersection		✓			✓		✓	✓	
	perform U-Turn		✓			✓		✓	✓	
	select route depending on traffic / time of day					✓	✓			
	passing maneuver		✓			✓		✓	✓	

Figure 9.4: Elements of the driving task, cognitive resources required, and their connection.

tem. [Sch93] states that the glance time at a navigation display is significantly longer than glancing on a printed map, which may also be attributed to the lower quality of navigation systems 20 years ago and the driver being unaccustomed to such a device [ADHW88].

Communication and social environment become increasingly important factors in the driving task. While previously limited to conversations with co-driver and other passengers or to using the car's entertainment system, over the last few years the use of mobile phones, text-messaging and social networks has become a serious issue [EBM11].

Operating and monitoring systems is the fourth component of the driving task. In both cases, critical and uncritical systems are distinguished.

Finally, command decisions are also a part of the driving task, although in comparison less prominent than for a military aircraft pilot. Examples here are passing maneuvers, the decision to perform a U-turn, or selecting a route depending on the current traffic density or the time of the day.

[Bub03] provides a more common distinction, which is not detailed enough for our purpose but due to its popularity nevertheless mentioned here. The author uses the analogy of a driver assistant system with a human assistant and classifies the elements of the driving task based on the specification of what one would consider as main- and side-tasks of this human assistant or of a co-driver in a car race:
The *primary driving task* consists in the driving process itself.
The *secondary driving task* covers all activities which are caused by driving or the traffic context but are not primarily aimed at keeping the car steady on the road. This encompasses the use of blinkers, light switches, etc., but also the control of ACC systems or distance control systems.
Tertiary tasks, according to Bubb, are not related to driving itself, but serve the satisfaction of the driver's need for entertainment and information. This covers seat heating, air conditioning, car stereo, telephone, navigation device and in the near future an increasing number of Internet services.

9.2.3 Processing Resources

[KSB87] defines mental workload as "the cost of performing a task in terms of a reduction in the capacity to perform additional tasks that use the same processing resource."
While early theories of human information processing assumed a single processing channel, more recent theories are based on the concept of a pool of different types of processing resources. In order to efficiently time-share two tasks, their required processing resources need to differ. The more the required resources overlap, the less efficient the attempted multitasking becomes.
Wickens [WH99] isolates three dimensions of processing resources:
1. *Processing modalities:* Auditory vs. visual perception, manual or vocal control, etc. This dimension explains why it is easier to share a visual and an auditory task than two visual tasks.
2. *Processing codes:* Verbal or spatial.
3. *Processing stages:* Beginning of task, in the middle of a task, almost finished.
In figure 9.4, I use a simplified model distinguishing eight processing resources, matching them to the elements of the driving task. This model reflects my estimate of the processing resources required for a driving task, and this view is likely to be shared by many experts.
The processing resources used in the model are:

- Visual scanning, tracking, and monitoring
- Auditory tracking and monitoring

- Motoric tracking and haptic interaction
- Decision making
- Information recall
- Information processing and reacting to information
- Spatial estimation and interpretation
- Linguistic processing

The same dimensions will later be used to define cognitive complexity and information complexity.

9.2.4 Driver Distractions

Distraction is commonly defined as the divided attention of an individual or group from the chosen object of attention onto the source of distraction. Causes for distraction are (a) a lack of ability to pay attention, (b) lack of interest in the object of attention, or (c) the intensity of the distracting object / event.

Causes (a) and (b) can be neglected in our context when assuming that a driver is both capable and responsible, respectively. More interesting is cause c, since it may affect any driver subconsciously.

To connect with the previous formalization, I define driver distraction as diversion of one or more cognitive processing resources away from the primary task, with the effect of diminishing the driver's driving performance. Or, in short, distraction is a cognitive demand which is not caused by the requirements of the primary task. Hence, we can classify distractions, in analogy to figure 9.4, according to the processing resources required, i.e., diminished.

Here, I introduce the four-dimensional POLI-classification as another point of view on driver distractions:

- **Perceptual channel:** Distractions can be auditory, visual or actual physical obstacles (change of flow).
- **Origin:** The origin of a distraction can be the driver himself, another person, a controlled part of our system, a technical part not in control of our system, or an event in the local context.
- **Location:** The distraction can be inside the car or outside the car.
- **Intention:** The distraction can be intended by the user or unintentional.

As an example: Loud music from the car stereo would be a distraction in the dimension <audio, driver, inside, intended>. A traffic jam belongs to <obstacle, event, outside, unintended>. A crying baby in the passenger seat is in the category <audio, person, inside, unintended>. Similarly, a ringing cell phone is categorized as <audio, person, inside, unintended>. Figure 9.5 shows a selection of distractions with their respective POLI classifications, overall cognitive demand, and required processing resources (i.e., cognitive demand).
Based on this classification, we can assign numerical levels to each distraction category and determine the total level of distraction as the sum of all occurring distractions in the implementation of PRESTK (see part IV).

	POLI classification					Required processing resources							
Distraction type (examples)	Perceptual Channel	Origin	Location	Intention	Overall cognitive demand	visual scanning, tracking, and monitoring	auditory tracking and monitoring	motoric tracking and haptic interaction	decision making	information recall	information processing / reaction	spatial estimation and interpretation	linguistic processing
outgoing phone call	auditory	self	inside	intentional	HIGH		✓	✓	✓	✓	✓		✓
incoming phone call	auditory	person	inside	unintentional	HIGH	✓	✓	✓	✓	✓	✓		✓
passenger conversation	auditory	person	inside	intentional	MEDIUM	✓	✓		✓	✓	✓		✓
crying baby	auditory	person	inside	unintentional	HIGH		✓		✓		✓		
honking car	auditory	person	outside	unintentional	MEDIUM		✓		✓		✓	✓	
shouting pedestrian	auditory	person	outside	unintentional	HIGH	✓	✓		✓		✓	✓	✓
system message (beep)	auditory	system	inside	unintentional	LOW	✓	✓		✓		✓		
car stereo	auditory	system	inside	intentional	MEDIUM		✓	✓					
ambulance siren	auditory	context	outside	unintentional	HIGH		✓	✓	✓		✓	✓	
Billboard	visual	context	outside	unintentional	LOW	✓				✓			✓
system message (text)	visual	system	inside	unintentional	MEDIUM	✓		✓	✓		✓		✓
traffic light	visual	context	outside	unintentional	MEDIUM	✓		✓	✓		✓	✓	
bicyclist	visual	context	outside	unintentional	MEDIUM	✓	✓	✓	✓		✓	✓	
ambulance	visual	context	outside	unintentional	HIGH	✓		✓	✓		✓	✓	
person on the street	obstacles	context	outside	unintentional	MEDIUM	✓		✓	✓		✓	✓	
traffic jam	obstacles	context	outside	unintentional	MEDIUM	✓		✓	✓		✓	✓	
construction site	obstacles	context	outside	unintentional	HIGH	✓		✓	✓	✓	✓	✓	
deviation	obstacles	context	outside	unintentional	HIGH	✓		✓	✓	✓	✓		
lost cargo	obstacles	context	outside	unintentional	HIGH	✓		✓			✓		

Figure 9.5: Selected distractions classified by POLI, overall cognitive demand, and required processing resources.

9.2.5 Individual Driver Differences

As with all skills, the ability for driving depends not only on the condition of the day but also on individual differences between the drivers. In the previous section, I formalized the elements of the driving task in accordance with and extending Schlegel's analogy between the driving task and the description of the flying task in Federal Aviation Regulation 25 [Sch93]. This analogy has its limits though. People who operate an aircraft are highly trained and carefully selected professionals, while acquiring a driver's license is feasible for a vast majority of the population. As a result, individual differences and skills vary more significantly.
The concepts *stress* and *strain*, originally used in the context of evaluating the quality of physical material, provide a good metaphor for describing the individual differences in the impact on cognitive demand for the individual cognitive capacity: the same level of stress does not result in the same level of strain for all drivers.
The *age* of the driver plays an important role. On one hand, we have young, unexperienced drivers with a higher tendency for incautious behavior, which increases the risk of ignoring safety precautions or warnings. On the other hand we have elderly drivers with diminished reaction time, more problems adapting to modern in-car information systems and potentially with limitations of hearing and eyesight [IWB93]. Even when compensating for vision problems with glasses, new problems can be introduced such as a slower change between far-sight and closely monitoring in-car systems while using bi-focal glasses. Overall, the expanding segment of elderly drivers is becoming a growing concern: Car manufacturers aim their higher priced premium cars at a target audience of 70+, 80+ or even older customers [EG10]. This is caused by the sociopolitical development and the higher purchasing power of this demographic segment. At the same time, the limitations of elder drivers have to be considered carefully. As a result of a study with more than 300 senior cititens, [FRC$^+$07] emphasize the declining ability to multitask at an older age. A demanding visual-spatial task affects motoric speed and reaction times. Another important factor is adult onset diabetes and related cardiovascular problems and/or sudden loss of conciousness. Countermeasures include an emergency assistant, as announced in 2009 by car manufacturer BMW, which detects a health problem such as unconciousness of the driver, safely parks the car at the side of the road and automatically calls an ambulance.
Fatigue affects reaction times and influences the ability to conduct a vehicle. While professional drivers usually have a legal obligation to take breaks and rest regularly, it is rather difficult to impose similar obligations on all drivers.
Drivers with *disabilities* have related differences, which have been a concern for several decades already. In 1966, [Ysa66] emphasized on safety issues for physically disabled drivers, and proposes measures aimed at "improving the compensatory technical modifications". Furthermore, activities of the European Union are aiming at "e-Inclusion", which is defined as an effort "that *no one is left behind* in enjoying the benefits of ICT. e-Inclusion means both inclusive ICT and the use of ICT to achieve

wider inclusion objectives. It focuses on participation of all individuals and communities in all aspects of the information society. e-Inclusion policy, therefore, aims at reducing gaps in ICT usage and promoting the use of ICT to overcome exclusion, and improve economic performance, employment opportunities, quality of life, social participation and cohesion."[1]. The e-Inclusion concept covers for instance speech handicaps [VSLR08] or learning difficulties [Abb07].
On a more general level, individual drivers' *abilities* also differ. Besides the obvious fact that different people are differently skilled at the driving task, other individual abilities or disabilities play an important role as well. This includes personal attention span and ability to concentrate on the task at hand, but also physical conditions such as color blindness.
While pilots meticulously keep track of their flight hours and regular practice, there is no such requirement for individual drivers. *Experience* differs significantly among drivers and is crucial for the driving performance.
The *mood* of the driver has also to be considered in detail. [JYW11] investigated the effects of specific negative emotions on risk perception, driving performance, and perceived workload. The authors argue that, for instance, an angry driver is not the same as a fearful driver, although both anger and fear are negative emotions. Fearful drivers for instance have a higher perceived workload than angry drivers. To consider individual differences, it is important to look not only at short-term emotions, but also at the general mood of the driver ("condition of the day").
Finally, *consumption of alcohol or drugs* (both prescription and illegal) has a significant impact on individual driving performance. Since this has been a well-known fact for a long time already,[2] legal regulations apply in most parts of the world.
Although the effects of the influence last only for a certain time, the tendency to consume alcohol/drugs or the need for prescription drugs can be considered individual properties of the driver.

9.3 Assessing Cognitive Load

In section 5.4, we discussed several ways known from literature to assess the cognitive load of the driver. We distinguish between self-assessment, performance-based measures, and physiological measures. Most of the known measures however are not suitable for everyday use on the road, partially because the values cannot be measured in real-time, and partially due to the intrusiveness of the method of measuring.
A potential alternative already mentioned previously is to refrain from measuring cognitive load and turn to estimate cognitive load by context (see section 5.4.4), which

[1] ec.europa.eu/information_society/activities/einclusion/index_en.htm
[2] The first documented arrest of a drunk driver dates back to September 1897 in Great Britain, see www1.wdr.de/themen/archiv/stichtag/stichtag2926.html

partially covers System Situation Awareness Level 1 proposed in figure 9.2. This connection is empirically evaluated in section 10.2. Figure 9.6 shows this correlation and a selection of environmental selections.
Another contextual distraction is provided by the system itself: Each piece of information presented itfelf adds a cognitive demand. We will see in section 9.6 how to estimate the complexity of presentations.

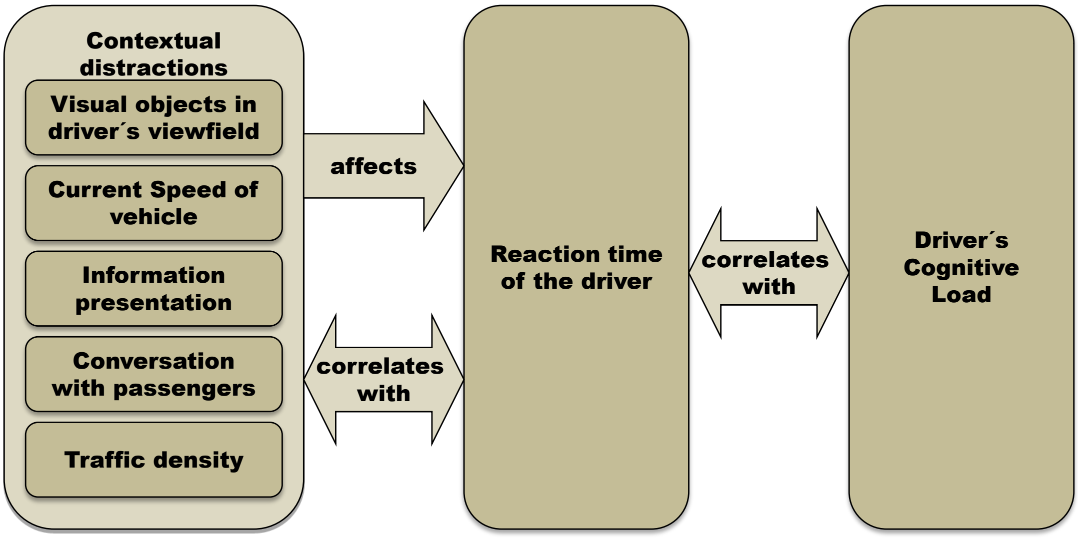


Figure 9.6: Correlation between contextual complexity and reaction time.

By keeping a history of these cognitive demands and providing a model of the decay of cognitive demand over time, the impact of system messages on the overall cognitive load of the driver can be estimated. [Geb07] provides various models of the temporal decay of emotions, which can also be applied to the decay of stress.
As with emotions, the temporal decay of cognitive demand is hard to predict; there is no known model in the literature. To overcome this problem, we assume an approximation with linear decay. Figure 9.7 shows an example of the decay of the cognitive demand of several consecutive presentations by the system. The overall distraction or cognitive workload is approximated using the sum of the individual demands at each point on the temporal axis as an upper bound. In a more detailed model, the overall cognitive demand can be replaced by the eight processing resource dimensions identified previously. The overall mechanism stays unchanged.
Calculating the impact of presented information can be applied not only retrospectively but also looking ahead to the effect planned presentations will have on the driver. This is System Situation Awareness level 3 in my model.

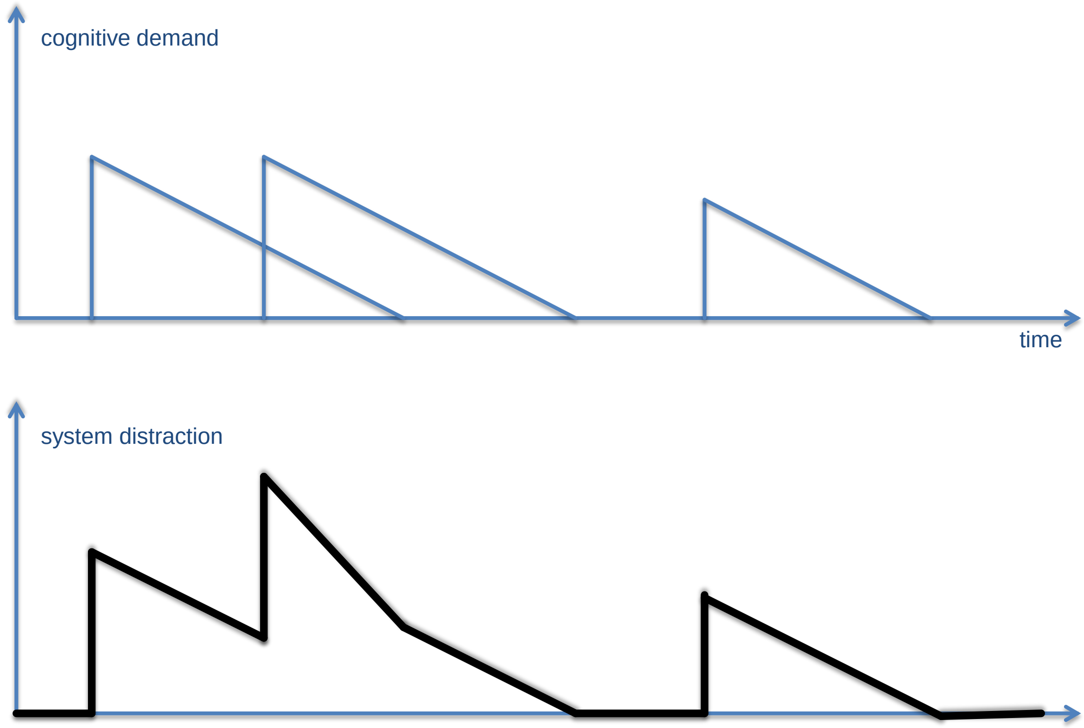


Figure 9.7: Linear temporal decay of cognitive load caused by distractions.

9.4 Updating the KAPcom User Profile

The KAPcom Automotive Ontology

[FE10] claim that the Human-Machine Interface (HMI) in vehicles is currently undergoing an evolutionary change: "While the focus is still on functions supporting the primary task, a new generation of more powerful interaction, service and entertainment concepts, which extend beyond driving and also involve non-driver passengers, is starting to surface." These developments lead to a greater degree of context adaptability and personalizability. To support this development, a new and open approach for sharing, maintaining, and exchanging information is required: A knowledge base to assemble important information on environment, vehicle, driver, passengers, and context. The Knowledge And Personalization component KAPcom is our approach to realizing this infrastructure.
In addition to this component, we also need a way to formalize this knowledge.
The KAPcom Automotive Ontology [FE10] was designed with the following benefits in mind:
1) Functions can easily be made context-adaptive, e.g., dependent on vehicle speed, traffic conditions or surroundings;
2) Applications can exchange knowledge and cooperate in new ways;

3) User models can be shared between vehicles (e.g., car and motorcycle);
4) Privacy features allow a fine-grained control over what is shared over a Car-2-Car channel.
KAPcom uses generic user properties and characteristics from [HSB+05] and supports time-based reasoning based on the methods described in [Kri10] and thus supports dynamic change of information and keeping a history of previous values/states. Figure 9.8 shows a part of the GUMO [HSM+07]. Although concepts such as cognitive load or physiological state are included, these conepts have not yet been defined in the original ontology.

Extensions for the Automotive Ontology

The extendable Automotive Ontology proposed in [FE10] did not yet contain any means for annotating the cognitive load of a person in detail. In this section we propose a way to extend it and include this information. As we have seen previously, cognitive load can be modeled at different levels of complexity. We can use a single value to annotate the overall complexity, or we can get in more detail into the different processing resources required. Listing 9.1 shows a simple annotation.

Listing 9.1: Simple annotation of Cognitive Load in the KAPcom Automotive Ontology

```
<kapcom>
  <user id="m.feld">
    <cognitiveload>
      <overall value="0.3" />
    </cognitiveload>
    [...]
  </user>
</kapcom>
```

To model the current cognitive load of the user in more detail, we will come back to the processing resources identified in section 9.2.3. Listing 9.2 shows an example.

Listing 9.2: Simple annotation of Cognitive Load in the KAPcom Automotive Ontology

```
<kapcom>
  <user id="m.feld">
    <cognitiveload>
      <overall value="0.3" method="mean" />
      <resources>
        <resource name="visual"     value = "0.2">
        <resource name="auditory"   value = "0.3">
        <resource name="motoric"    value = "0.4">
        <resource name="deciding"   value = "0.3">
        <resource name="recall"     value = "0.1">
        <resource name="reacting"   value = "0.5">
        <resource name="spatial"    value = "0.4">
        <resource name="linguistic" value = "0.2">
      </resources>
    </cognitiveload>
    [...]
  </user>
</kapcom>
```

This representation can answer questions both about single processing resources and about the overall value of the cognitive load. Please note that by stating the statistic mean of the values as a method for computing the overall value, we could have

Figure 9.8: The GUMO ontology [HSM+07]

skipped the value field there and leave it up to the processing component, i.e., KAPcom, to compute this value. Other methods for calculating the overall value are “min” (the minimum of all the known values) or “max” (the maximum of all known values). Furthermore, not all values for the processing resources have to be known. KAPcom skips over unknown values when computing the overall value.
This approach further adds flexibility: in case inclusion of additional processing resources is found to be necessary, no previously developed components have to be changed.
Updating KAPcom corresponds to system situation awareness level 2.

9.5 Annotating Presentations with Estimated Complexity Values

The presentation model concept in sim^{TD}, as described in detail in [Cas13], is based on a hierarchical model of XML descriptions of the presentation task (see figure 9.9).

The root is a general description of a presentation model (see listing 9.5). Descriptions based on this model can either inherit or overwrite parts of the description.
The basic presentation model consists of three main parts: OSGi parameters, internal parameters, and display strategies.
The OSGi parameters consist of general information on the presentation task, which can be updated after the creation of the task. The start and end time specified here are initial estimates which can be modified both by the application requesting the presentation and by the HMI bundle orchestrating the presentation tasks.[3] Local prioritization is also given by the requesting application and part of the sim^{TD} prioritization concept (see section 4.2).
Global prioritization on the other hand is part of the internal parameters and cannot be changed by the application. It provides the general range of priority for this application and was defined in a workshop with all application developers involved. This range represents how important and safety-critical each type of information is for the user from the perspective of the system designers. Further internal parameters define whether or not a change to the presention screen is necessary or what the minimal presentation span (and its tolerance in emergencies) is for a meaningful presentation of this information.
The display strategies portion of the general presentation model description starts out empty and filled by inheriting descriptions.

Listing 9.3: presentationModel.xml

```
<PresentationModel name="presentationModel" appId="0" basedOn="">
        <OsgiParameters>
                <Parameter name="timeStart" type="long" min="0" max="max" default="" />
                <Parameter name="timeEnd" type="long" min="0" max="max" default="" />
                <Parameter name="currentPrioritisation" type="int" min="-100" max="100" default="0"/>
                <Parameter name="presentationType" type="enum" values="0" default="0"/>
        </OsgiParameters>
        <InternalParameters>
                <Parameter name="globalPrioritisation" type="int" min="0" max="100" default="0"/>
                <Parameter name="show" type="int" min="0" max="1" default="1" />
                <Parameter name="pos" type="string" values="" default="0,1,2,3,4,5" />
                <!-- the minimal time in milliseconds a presentation should last.
                        Presentation request below this time will be rejected -->
                <Parameter name="minimalPresentationTime" type="int" min="0" max="max" default="4000" />
                <!-- A tolerance in milliseconds for the parameter "minimalPresentationTime" -->
                <Parameter name="toleranceTime" type="int" min="0" max="max" default="500" />
        </InternalParameters>
        <DisplayStrategies>
        </DisplayStrategies>
</PresentationModel>
```

To clarify the concept described here, we follow one path through the tree as an example. In listing 9.4 we see the concept *local danger warning* as extension of the *presentation model*. Of particular interest is the introduction of the parameters *distanceToEvent*, *warningLevel* and *typeOfWarning* as *OSGi parameters*, which refine the required information for this kind of warning. Also, as an internal parameter, the *minimal presentation duration* is set to 3 seconds.[4]

[3] Of course the start time cannot be modified after the presentation task started.

[4] Displaying information for too short a time, especially in potentially dangerous situations, would confuse the driver and is counterproductive for the purpose at hand.

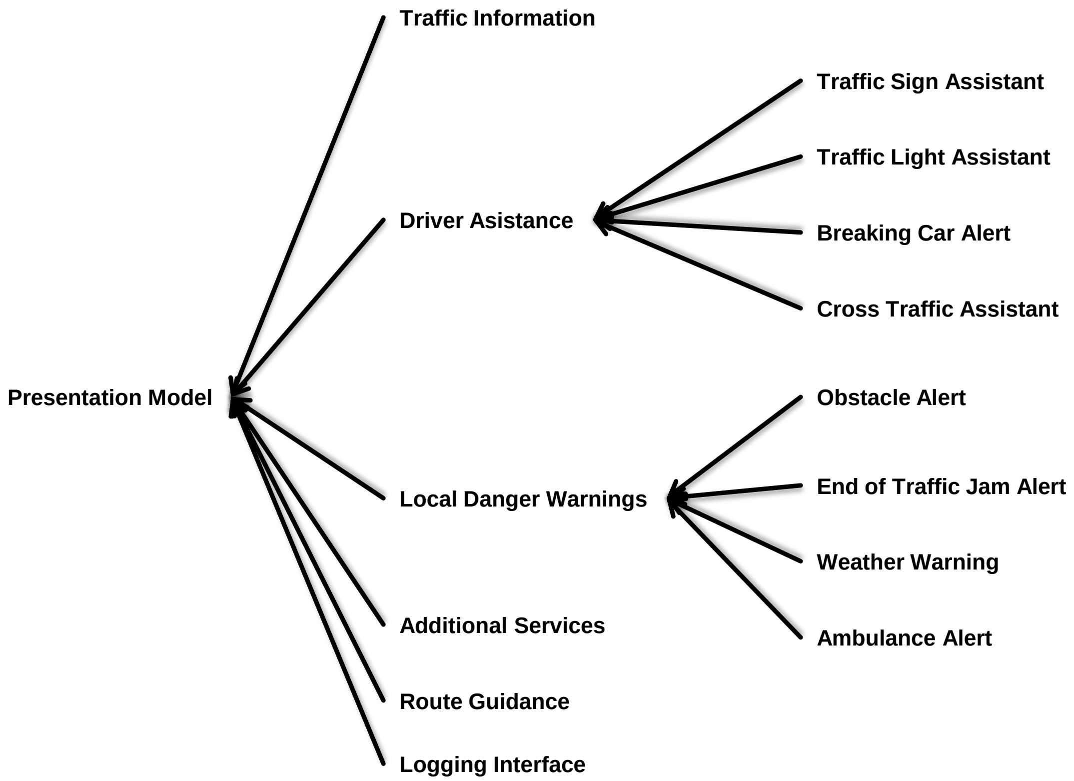


Figure 9.9: The sim^{TD} presentation model concept as hierarchical model of XML descriptions.

Two common *display strategies* for the *local danger warnings* are also introduced here. The preferred strategy (*id=0*) describes displaying the information both on the *main screen* and in the *symbol area*, whereas the alternative strategy (*id=1*) is limited to the *symbol area*. This strategy can be selected when more important information has to be displayed on the main screen.

Listing 9.4: LocalDangerWarnings.xml

```
<?xml version="1.0"?>
<PresentationModel name="localDangerWarnings" appId="21" basedOn="presentationModel.xml">
        <OsgiParameters>
                <Parameter name="distanceToEvent" type="int" min="0" max="max" default="300"/>
                <!--0: possible, 1: information, 2: warning-->
                <Parameter name="presentationType" type="enum" values="0,1,2" default="0"/>
                <!--0: possible, 1: information, 2: warning-->
                <Parameter name="warningLevel" type="enum" values="0,1,2" default="0"/>
                <Parameter name="typeOfWarning" type="enum" values="0" default="0"/>
        </OsgiParameters>
        <InternalParameters>
                <Parameter name="maxDistance" type="int" min="0" max="max" default="0" />
                <Parameter name="minimalPresentationTime" type="int" min="0" max="max" default="3000" />
        </InternalParameters>
        <DisplayStrategies>
                <DisplayStrategy id="0">
                        <displayTask>
                                <channel name="mainScreen">
                                        <layout name="F$appId$">
                                                <typeOfWarning>$typeOfWarning$</typeOfWarning>
                                                <distance progressType="to">$distanceToEvent$</distance>
                                                <maxDistance>$maxDistance$</maxDistance>
                                                <warningLevel>$warningLevel$</warningLevel>
                                        </layout>
                                        <show>$show$</show>
```

```
                    </channel>
                    <channel name="symbolArea" pos="$pos$">
                        <layout name="F$appId$">
                            <warningLevel>$warningLevel$</warningLevel>
                            <typeOfWarning>$typeOfWarning$</typeOfWarning>
                        </layout>
                    </channel>
                </displayTask>
            </DisplayStrategy>
            <DisplayStrategy id="1">
                <displayTask>
                    <channel name="symbolArea" pos="$pos$">
                        <layout name="F$appId$">
                            <warningLevel>$warningLevel$</warningLevel>
                            <typeOfWarning>$typeOfWarning$</typeOfWarning>
                        </layout>
                    </channel>
                </displayTask>
            </DisplayStrategy>
        </DisplayStrategies>
</PresentationModel>
```

Finally, on the third level, we have a specific presentation task, an alert about obstacles on the road. Most of the parameters are given already in the parent description, so we only need to define the type of obstacle and redefine the global priority range. The display strategies are inherited from the local danger warnings description and need not be redefined. As a result, we have a very compact description of the presentation task here.

Listing 9.5: obstaclesOnTheRoad.xml

```
<PresentationModel name="obstaclesOnTheRoad" appId="211" basedOn="localDangerWarnings.xml">
    <OsgiParameters>
        <!-- 0 = Liegengebl. Fahrzeug
            1 = Baustelle
                2 = Verlorene Ladung
                3 = Wanderbaustelle
                4 = Fussgaenger
                5 = Tiere
            6 = Gefahrenstelle
                7 = Unfall
                8 = Stehendes Einsatzfahrzeug
        -->
        <Parameter name="typeOfWarning" type="enum" values="0,1,2,3,4,5,6,7,8" default="0" />
    </OsgiParameters>
    <InternalParameters>
        <Parameter name="globalPrioritisation" type="int" min="70" max="100" default="85" />
    </InternalParameters>
    <DisplayStrategies>
    </DisplayStrategies>
</PresentationModel>
```

Please note that this XML representation is only used as a blueprint for validation of the incoming presentation tasks. Due to performance reasons, internal processing of instances of the presentation task as well as exchange of information between applications and the HMI bundle consists solely of efficiently coded Java objects.
In this representation defined by [Cas13], it would be easy to annotate (i.e., extend) the different display strategies with a value for presentation complexity: The tag *DisplayStrategy* can be enriched with a parameter *overall complexity* containing a numerical value to be considered by the presentation component. Alternatively, parts of the *display strategy* such as the *channel* can be annotated with a value describing the complexity of that part of the presentation to be aggregated on *displayStrategy*-level. However, since the PRESTK implementation will not be using this annotation format anymore, specific proposals are not necessary.

9.6 Introducing ACE: Annotated Complexity Estimation

In order to assess system-generated CL, we need to be aware of the impact of system-generated presentations to the driver, i.e., we need to estimate presentation complexity. Literature in this area has been discussed in detail in section 5.5.
Determining the complexity value depends on the availability of structured data. We distinguish three different cases:
1. Structured information about possible presentations is available as a blueprint in the system, or can be derived beforehand from unstructured presentation information.
2. The presentation is available at runtime and specified in a formal presentation markup language.
3. We obtain an unstructured presentation in form of an image or an audio file, or both.
In this section, all three cases are briefly discussed. However, I focus on case 2 in the discussion and in the implementation of the PRESTK system.
Case 1 does not provide much of a scientific challenge and is of little interest here. It merely reduces the question to an issue of solid design.
Case 3 is very hard to tackle at a level any better than by an educated guess. However, it is not completely impossible to reverse engineer the raw data received for presentation and obtain structured data to be subjected to analysis that we perform on structured data. [DSZ07] for instance used reverse engineering to transform existing GUIs into a generic description in the language XIML, which could in turn be used for rendering the same GUI on different platforms. The authors did however have the advantage of having a GUI to analyze, and not just a picture to be presented. Analyzing a picture of a presentation to obtain its structure is an interesting challenge, but out of the scope of this thesis.

Case 2 is where I focus my attention. Structured data for alternative presentation strategies of the same content can be encoded in the presentation container language defined in section 9.7. Here, I introduce the *Annotated Complexity Estimation* procedure *ACE* to formally analyze the complexity. Using this measure, we can provide a value for the cognitive complexity of a presentation at runtime.
The literature review in section 5.5 indicates clearly that considerable work has been put into analyzing parameters and conditions to streamline and improve the delivery of information to the driver of a vehicle. Especially [IWWC89] and [IWB93] probed every conceivable parameter of in-car display design very thoroughly and provided display designers with a detailed model of their impact on the driver's perception. Attempting to extend their work would not be of much avail.
On the other hand, their work is based on displays of the late eighties, and technological progress did not stop there. While the emphasis back then was on font size, color,

brightness, contrast and word complexity, we now also have to deal with sophisticated layouts, background patterns, icons, etc. Also, the use of a touch screen and virtual buttons on the screen was not considered then.

In this section, I will present my Annotated Complexity Estimation procedure ACE. It is evaluated in section 10.3. A shorter version of this description was already published in [EFM12a].

ACE

Layout is commonly defined as the part of graphic design that deals with the arrangement and stylistic treatment of elements. We distinguish between grid-layout and the more rigid template-layout. A programming interface for a user interface, such as for instance Java Swing [ELW98], provides several layout managers for the developer to choose from.
In defining the Annotated Complexity Estimation procedure ACE, I reverse the top-down layout manager process to a bottom up aggregating model of complexities. Figure 9.10 shows a sample screen from the sim^{TD} system, and figure 9.11 provides a schematic view on it.
The nested structure of a user interface can be represented as a tree structure, similar to the HMI Allocation Tree concept introduced in chapter 8. The main layout manager is located at the root of the tree. Other layout managers may be nested in it.
When analyzing the complexity of a layout, we start at the leaves of that tree and work up to the top, accumulating the complexity until we reach the top of that tree.
The simplest leaf (or more precisely: component) we encounter is, for instance, a label or an icon. A label has a text of a certain complexity, and it might contain an additional small icon making it visibly more complex. These rudimentary components can be grouped in a panel with identical or different elements. The panel has features like the size, in terms of the number of elements it contains, or it might have a visual boundary, such as a borderline, that makes it easier to perceive as a unit.
Panels again might be combined into a higher level panel. Following this combination of elements even further, we reach the root of the tree and the component that fills the whole screen. We are interested in obtaining a numerical value that describes the visual complexity of that root node. In figure 9.12, the structure of the HMI is transformed into a component tree. In order to apply the ACE procedure, the leaves of the tree have to be annotated with numerical values, and for all non-leaf nodes an aggregation formula has to be specified. In order to automate the procedure, we transform the tree into a machine-readable XML annotation. Listing 9.6 shows the XML representation of the tree.

Figure 9.10: Hierarchical panels in the options screen of the sim^{TD} HMI.

Listing 9.6: The structure of the sim^{TD} option screen in XML representation.

```
<ace name="simtd_options_screen">
        <root>
                <panel type="icon">
                  <icon type="empty" />
                  <icon type="empty" />
                  <icon type="empty" />
                  <icon type="empty" />
                  <icon type="empty" />
                  <icon type="empty" />
                        <icon type="static" />
                </panel>
                <panel type="label" metainfo="named" decoration="framed">
                        <label text="true" icon="true" />
                        <label text="true" icon="true" />
                        <label text="true" icon="true" />
                </panel>
                <panel type="label" metainfo="named" decoration="framed">
                        <label text="true" icon="true" />
                        <label text="true" icon="true" />
                </panel>
                <panel type="label" metainfo="none" decoration="framed">
                        <label text="true" icon="true" />
                </panel>
                <label text="true" icon="true" />
        </root>
</ace>
```

Component Values and Aggregation

An interim consent for the weights was achieved as shown in Table 9.1. It will be the objective of further experiments to determine its validity or to learn more accurate projections. The algorithm traverses the XML description, recursively assigning values to each component. The attribute cpx is used for annotation. In our example, this results in annotated XML shown in listing 9.7.

The overall complexity calculated for this example is 9.3.
Please note that this is based only on structural complexity by design of the user inter-

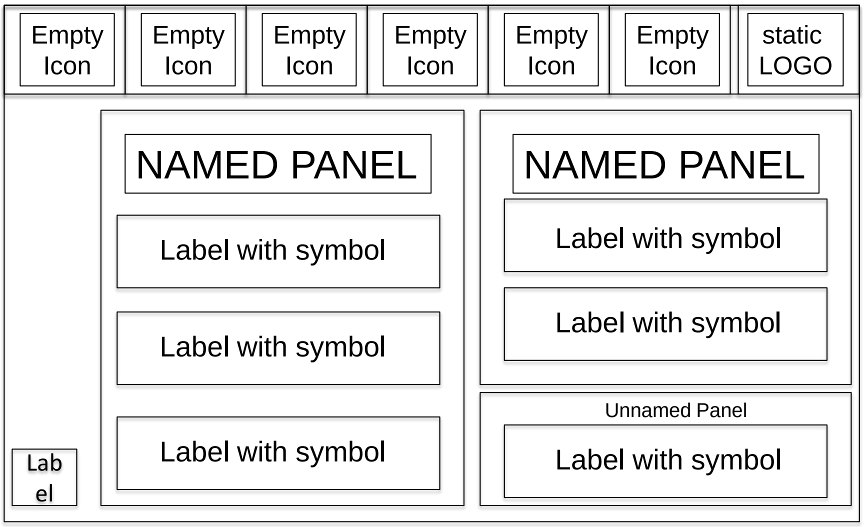


Figure 9.11: HMI structure as nested GUI components.

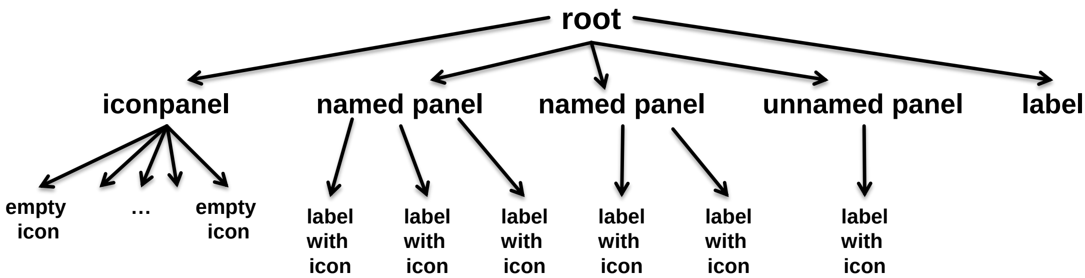


Figure 9.12: HMI as component structure tree.

face. In a more refined approach, all the parameters identified in [IWWC89] have to be considered as well.

Listing 9.7: Listing 9.6 with annotated complexity.

```
<ace name="simtd_options_screen" cpx="9.3">
        <root>
                <panel type="icon" cpx="0.8">
                  <icon type="empty" cpx="0.1" />
                  <icon type="empty" cpx="0.1" />
                  <icon type="empty" cpx="0.1" />
                  <icon type="empty" cpx="0.1" />
                  <icon type="empty" cpx="0.1" />
                  <icon type="empty" cpx="0.1" />
                        <icon type="static" cpx="0.2" />
                </panel>
                <panel type="label" metainfo="named" decoration="framed" cpx="3.4">
                        <label text="true" icon="true" cpx="1.0" />
                        <label text="true" icon="true" cpx="1.0" />
                        <label text="true" icon="true" cpx="1.0" />
                </panel>
                <panel type="label" metainfo="named" decoration="framed" cpx="2.4">
                        <label text="true" icon="true" cpx="1.0" />
                        <label text="true" icon="true" cpx="1.0" />
                </panel>
                <panel type="label" metainfo="none" decoration="framed" cpx="1.7">
                        <label text="true" icon="true" cpx="1.0" />
                </panel>
                <label text="true" icon="true" cpx="1.0" />
        </root>
</ace>
```

component	basic complexity value	feature	added complexity
label	0.1	text=true	+0.5
		icon=true	+0.4
icon	0.1	type=empty	+0.0
		type=icon	+0.5
		type=static	+0.2
		metainfo=text	+0.4
panel	$0 + \sum$ child nodes	decoration=framed	+0.2
		decoration=none	+0.5
		metainfo=named	+0.2
		metainfo=none	+0.5

Table 9.1: Calculating values for ACE evaluation

9.7 PTCL–The PRESTK Container-Language for Presentation Annotation

As shown in figure 9.1, we have discussed the driver-related contribution to situation awareness and now will take a look at presentation-related information, i.e., presentation complexity. By doing this, we start again at system situation awareness level 1 in our model.

In section 3.1 and appendix A, a broad survey of various presentation markup languages was presented. Despite standardization efforts, there is still a large amount of available languages. Imposing a new, additional standard on top of existing approaches is not necessary.

As an alternative, I propose a wrapper language as a container format for existing markups. Container languages are commonly defined as a meta file format whose specification describes how different data elements and meta data coexist in a computer file. The requirements for such a container language are simple:

1. It can contain different kinds of representation languages.

2. Meta-data on the presentation task necessary for the presentation toolkit can be stored in it.

3. Different alternative display strategies can be encoded (following the example of [Cas13]).

4. Each display strategy can be annotated with a metric for preference as well as with a cognitive demand value, either as an overall value or in a more detailed way.

As one of the results of the presentation language survey, a clear trend towards XML-based languages was identified. In accordance with that, the proposed meta-format introduced here–PTCL for PRESTK Container Language–is also XML-based. A simple example is shown in listing 9.8.

Listing 9.8: A simple PTCL example wrapping two alternative display strategies of one presentation task.

```
<ptcl>
        <meta>
                <overallPriority value=70 metric="percent" />
        </meta>
        <displayStrategies>
                <strategy>
                        <preference=1 />
                        <demand=0.8 />
                        <representation language="XY">
                                [first variant of presentation task in language XY]
                        </representation>
                </strategy>
                <strategy>
                        <preference=2 />
                        <demand=0.3 />
                        <representation language="XY">
                                [second variant of presentation task in language XY]
                        </representation>
                </strategy>
        </displayStrategies>
</ptcl>
```

This example can be extended in various ways. Meta-data of the *presentation task* may also include *minimal presentation duration* or similar processing information specifying whether or not the scheduler may prepone or postpone this presentation on the temporal axis. The *displayStrategies*-block may define some variables for recurring parts in more than one *display strategy*, and the strategies themselves can contain a more detailed demand definition as defined in section 9.4.

Formalizing complexity values by annotating presentation corresponds to System Situation Awareness level 2.

9.8 Implementing System Situation Awareness: Considering Cognitive Load and Presentation Complexity

How can we adequately utilize the previously collected information? Figure 9.13 shows the impact of a presentation manager on the driver's CL. By both assessing information complexity and measuring CL, presentations can be modified such that in high demand situations the additional cognitive workload is kept to a minimum. Complex presentations can be avoided or replaced by presentations with a simplified version of the same content, or, in cases of low priority, skipped altogether. If complex presentations have to be presented, we can make sure that the time for processing them is sufficiently long. If new and potentially difficult-to-grasp graphical concepts are used in the HMI, we may consider introducing them at low demand times. They would then be used in high demand times only after we can assume that the driver is familiar with them.

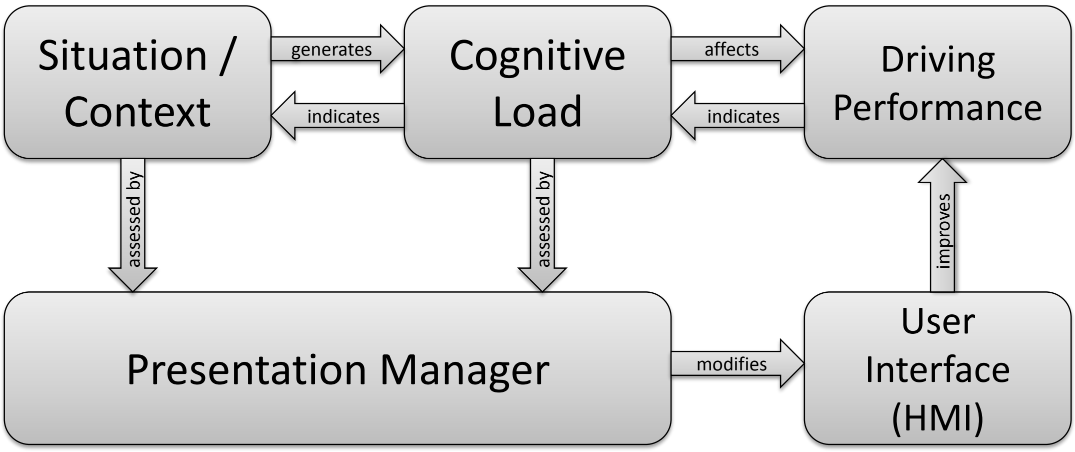


Figure 9.13: A situation-aware Presentation manager and its impact on driving performance.

Implementation

In chapter 8, the Resource-Constrained Scheduling Problem RCSP was introduced, and a graph-based approach to solving it proposed. This solution, however, did not include the system situation awareness I introduced in this chapter. Fortunately, the algorithm is sufficiently generic that the modifications required to make it situation-aware are possible.

The original tree-search algorithm introduced in chapter 8 constructs a tree with the given problem as its root. The child nodes are possible modifications of this problem, and the edges are labeled with the type of modification performed and the costs caused by it in terms of a penalty. A leaf is a conflict-free modification of the problem, i.e., a solution. Intelligent pruning removes duplicate solutions and all paths that are more expensive than the best solution found so far. Figure 9.14 recapitulates the concept. The implementation relies (among other things) on several problem-specific functions:

1. Each node can calculate which modifications are possible.
2. To each modification, a cost (penalty) can be assigned.
3. A node can determine whether it is a solution.
4. Each node knows its overall penalty.
5. Each node knows the series of modifications leading to it.

By modifying each of these points to include system situation awareness, or in some cases realizing that no modifications are necessary, the concepts introduced in this chapter can be implemented. I will discuss each of these points in detail now.

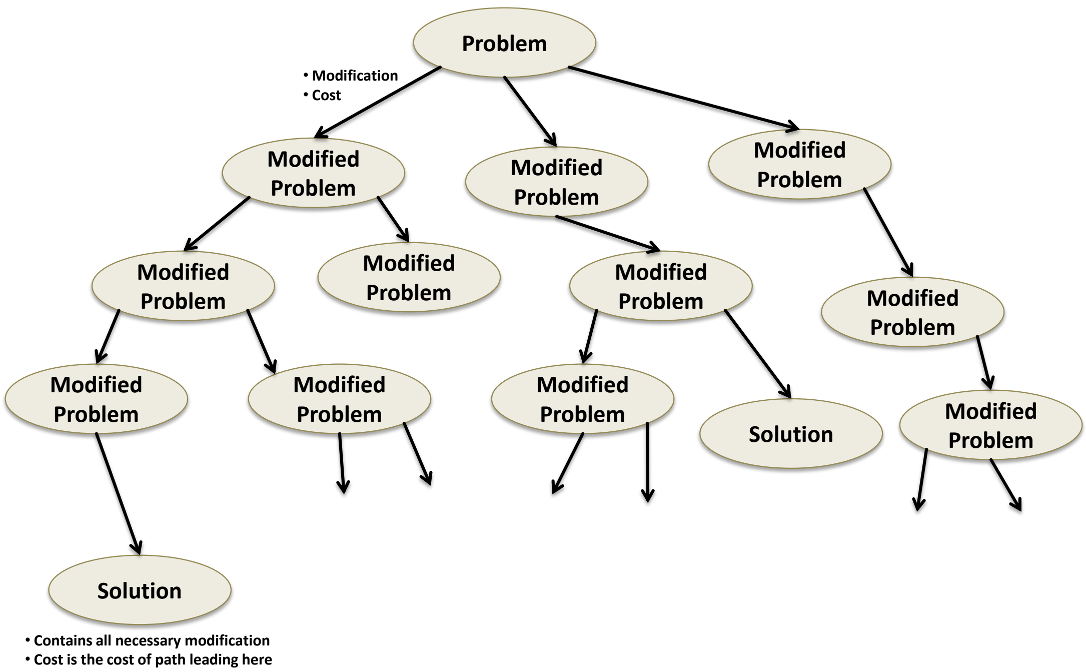


Figure 9.14: The underlying mechanism of the tree-search algorithm.

Possible Modifications

The possible modifications are encoded as constraints in the node implementation. The previous implementation already had the "change display strategy" modification included. This is sufficient for the system situation awareness as well.

Modification Penalties

The penalties for modifications have been previously determined using the type of modification, the extend of the modification, and the priority of the modified task. For situation awareness, we have to consider the change of complexity estimation when switching between different display strategies as an additional factor in the calculation.

Solution Detection

Detection of solution remains unchanged. A solution is a conflict-free modification of the problem. Of course it might be inappropriate due to an overall too high cognitive demand, but that is reflected in the penalty and the solution will be replaced by a better one.

Overall Penalty

The overall penalty for a solution is the sum of all penalties aggregated on the path leading to it. We have to add an additional criterion to the calculation here. As already described in section 9.3, consideration of the cognitive load induced by system messages is applicable not only in retrospect, but also on level 3 of my System Situation Awareness theory as part of the impact estimation: The system-induced cognitive demand from the proposed solution (cf. figure 9.7) has to be put in context with the projected cognitive load. Although there is some amount of uncertainty in the projected cognitive load due to possible sudden changes, in most cases the projection obtained using data from KAPcom will be sufficient.

Overall Modifications

The overall modifications are–as before–the modifications applied on the path from the tree root to the current leaf. No change is necessary here.

Personalization

Also, by using personalization, knowledge about the driver, especially about his expertise of the system, can be utilized. One important factor for the understanding of a presentation is the distinction between known and unknown, e.g., new, concepts appearing in it. In sim^{TD} for instance, concepts of the HMI design include symbol area, virtual screens, progress bar before an event, progress bar during an event, different audio signals, etc.
The list could be extended with other concepts, such as crossmodal referencing [EMM11] for instance.
Whether or not it is advisable to introduce a new concept in a presentation depends

mainly on two factors: The user's familiarity with the system and the amount of distractions he is currently encountering (as shown in Figure 9.15).

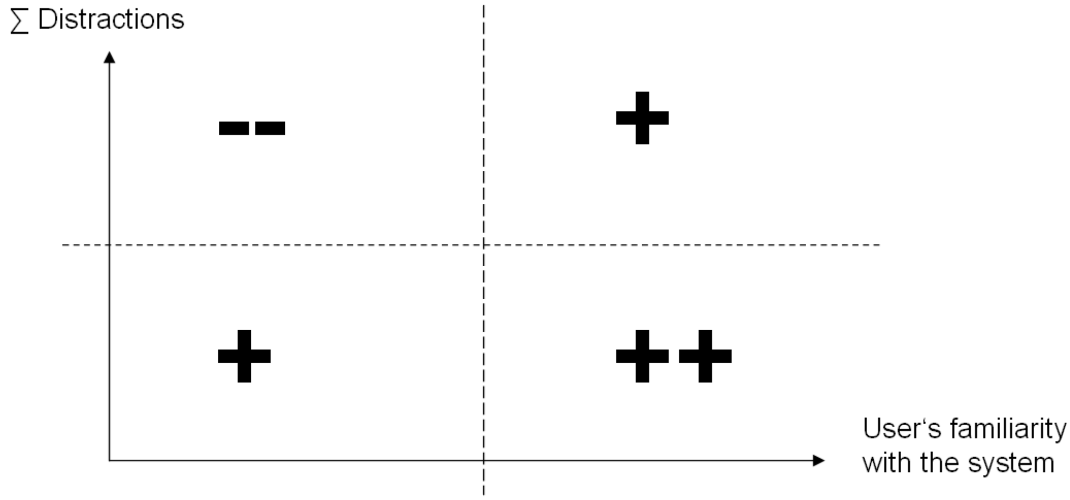


Figure 9.15: Recommendations for complexity of presentations shown.

9.9 Summary

In this chapter, the concept of the sim^{TD} presentation manager was extended with the newly introduced system situation awareness. In order to achieve system situation awareness, two paths have been followed and eventually combined. The necessary steps and achievements are summarized here.

1. **Cognitive Load**
   Required concepts for an implementation were formalized. The concepts of cognitive demand, cognitive load, and cognitive capacity were defined and explained. The driving task has been decomposed into its elements, in extension of [Sch93]. An eight-dimensional model of processing sources was introduced, defined and later on formalized to be used in the KAPcom component. The connection between driving task and processing resources was examined and defined. The POLI-classification for driver distractions was introduced and also connected to the processing resources concept. Individual driver differences resulting in varying levels of cognitive capacity were discussed. Endsley's model of situation awareness was adapted to the newly introduced concept of system situation awareness.
   Instead of using the cognitive load assessment techniques described in section 5.4, the approach of estimating cognitive load from context was followed and

detailed for the automotive environment. Two contextual conditions, visual complexity and speed, were picked for empirical evaluation in a simulator experiment. A third condition, presentation history, has been modeled according to the model of emotion decay in [Geb07].

2. **Complexity Estimation**
   Three varieties of complexity estimation have been introduced and discussed: Offline analysis, online analysis of structured data, and online analysis of unstructured data. Using the example of hierarchical sim$^{\mathrm{TD}}$ presentation tasks, a possible way to annotate those was shown. In a more general sense, the container language PTCL is introduced which can be used to annotate any presentation language for use with PRESTK .

By combining these two paths, system situation awareness is achieved. Modifications to the previous tree-search algorithm to be implemented in the PRESTK system are described.

Chapter 10

Experiments

The previous chapters introduced three concepts which require empirical evaluation.
In chapter 8, I proposed a tree search algorithm as a solution to the Resource-Constrained Scheduling Problem (RCSP) occurring when several mutually-independent information sources attempt to access a scarce resource simultaneously. Using a data-driven evaluation, the suitability of this approach will be shown in this chapter.
I postulated the connection between visual complexity of the environment and cognitive load of the driver in section 9.3. This connection is verified in a user study here. A short version of this evaluation was already published in [EMB12].
Finally, the Annotated Complexity Estimation (ACE) introduced in section 9.6 is evaluated here in a user study aimed both at verifying the general correctness of the procedure and at refining its parametrization.

10.1 Empirical Evaluation of the Tree-search Algorithm

In preparation of the sim^{TD} field test, applications are tested on the basis of prerecorded traces containing GPS positions, Car-2-Car communication messages, and vehicle data (velocity, heading, gas pedal status, brake pedal status, index lights, etc.). The scenarios lined-up at the X-axis in Figure 10.1 correspond to what we have seen earlier in Figures 8.2 and 8.3. Additionally, a scenario with hyper-realistic complexity was added (here denoted as Scenario V). Each scenario was further specified with three concrete evaluation cases (I.1, I.2, I.3, ... V.3). The dashed black line corresponds to the actual complexity of the evaluation cases:

Definition 7 (Complexity)**.** *The complexity of a conflict set T^C is*

$$cx(\mathcal{T}^C) = \frac{\sum ol(t_i^C, t_j^C)}{ms} \cdot |\mathcal{T}^C| \cdot |ol(t_i^C, t_j^C)| \cdot alloc(ms),$$

where ol is the temporal overlap of two conflicting tasks in milliseconds and alloc(ms) is the percentage of the makespan allocated by the unmodified tasks.

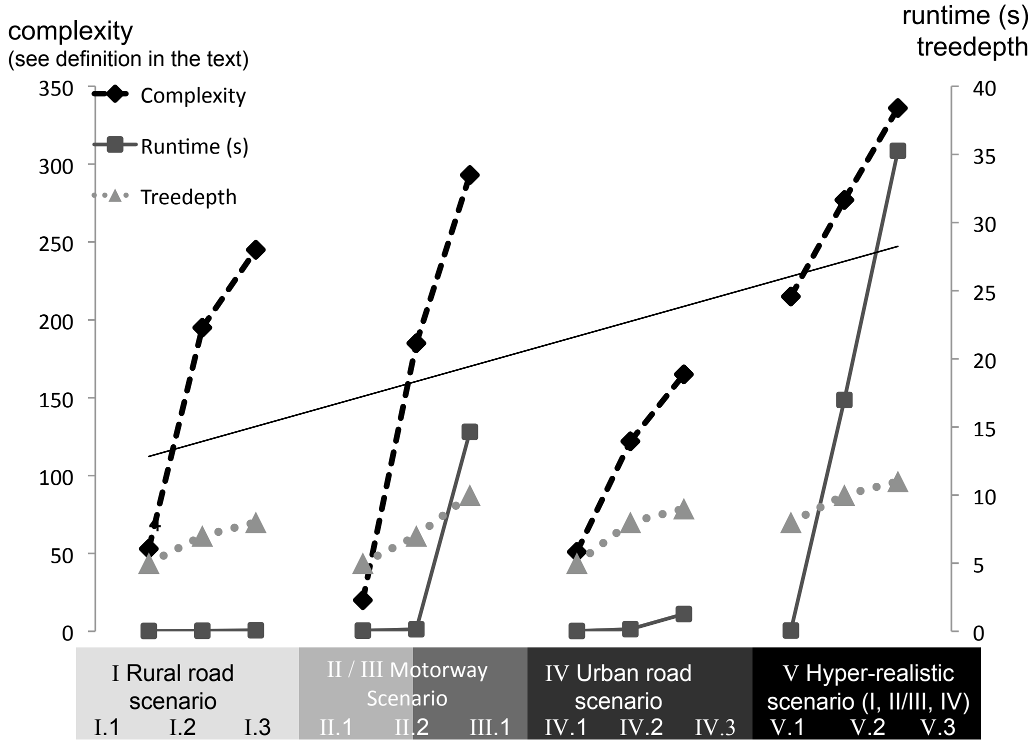


Figure 10.1: Result for test scenarios.

Figure 10.1 shows that the complexity of the test cases has a positive trend from left to right (see solid black trend line), which appeals to our intuition. Nevertheless, there is a significant case-specific complexity within each of the scenarios. We see, for example, that the variance of complexity in *Motorway scenario* is higher than both *Rural road scenario* and *Urban road scenario*. Note that the complexity of V.1 is lower than those of I.3 and III.1. This is because V.1 is a hyper-real *Rural road* test case, V.2 is hyper-real *Motorway* test case, and V.3 is a hyper-real *Urban road* test case.

Both run-time and tree-depth differed significantly, which can be seen by regarding the gray (square) solid lines, and, respectively, the dashed (triangle) lines. Runtime relates to the time needed to find the optimal solution. It was measured on a regular state-of-the-art desktop PC.

For each of the evaluation cases, a solution was found by the scheduling algorithm. In order to be able to distinguish between successful solutions of different quality, we define quality via a penalty for the modifying actions it involves: the fewer the modifications, the better the solution.

Definition 8 (Penalty of a RCSP solution)**.** *Let $\mathcal{M}$ be a set of modifications, $\mathcal{M} = \{m_1,...,m_N\}$ and t a task with given positive priority prio(t). The penalty p for m applied to t is $p(m,t) = d(m) \cdot prio(t) \cdot \Delta$, where d(m) defines the general desirability*

of an action as a positive integer value, with Δ specifying the temporal aspect of the modification (e.g., postponing or shortening).
The penalty of a solution is then defined as $\sum p(m,t)$ *of all (m,t) contained in it.*

The order of steps taken towards a certain solution is in our case not important. This is respected by pruning duplicate and redundant solutions. As a side effect, the penalty for a solution is well defined and independent of the way it was constructed.
Figure 10.2 shows the correlation between runtime and solution quality in terms of the penalty we defined. As stated before, the automotive context demands fast response times, so we need a good solution fast. Our algorithm provides a first solution typically in the first few milliseconds and then finds increasingly better solutions quickly. After 50–100 ms, we are very close to, or in some cases already at, the optimal solution. Note that the optimal solution is not at penalty zero, so the asymptotic behavior of the graph is not aimed towards the X-axis. Interrupting the scheduling process at this point leads to a satisfactory result. In Figure 10.3, the effects of pruning are depicted.

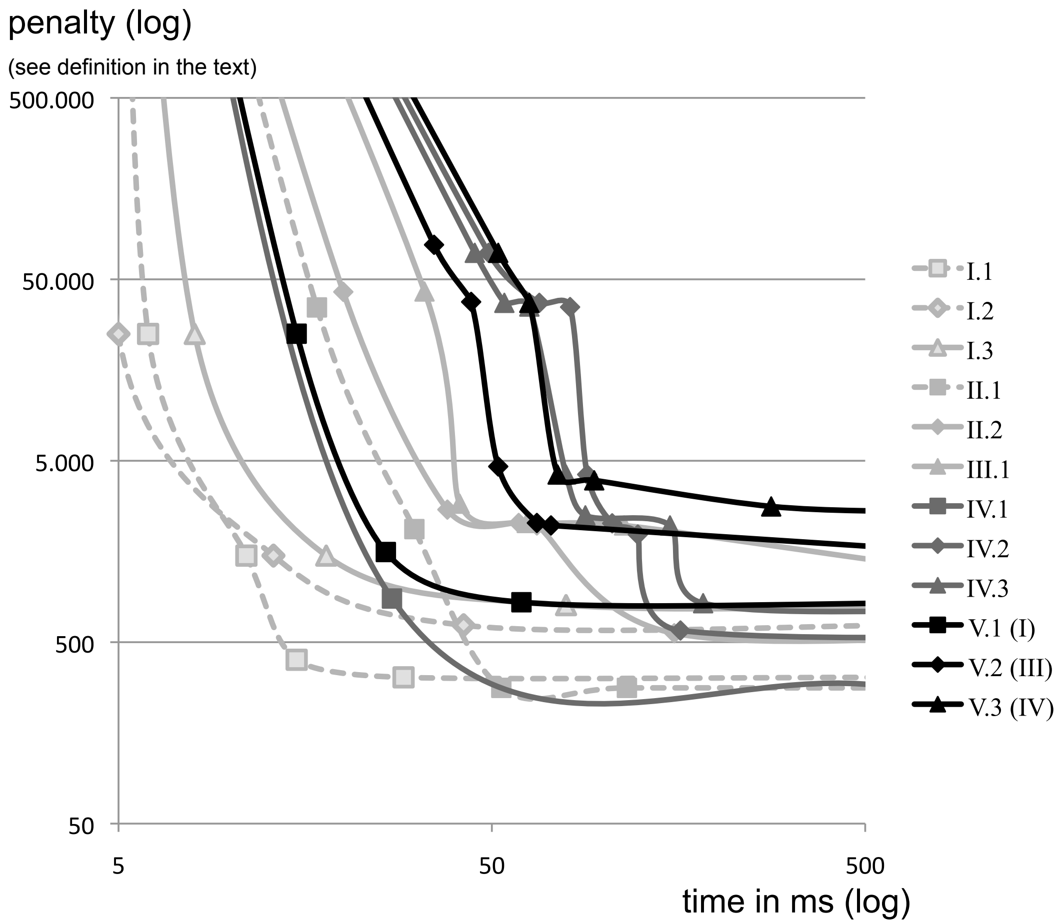


Figure 10.2: Improvement of solution quality over time.

The tree depth is correlated to the number of nodes calculated on that level using our BFS tree search. If no pruning had been applied, the graph would be a straight line on the logarithmic scale. Using the penalty of the best solution found so far (which drops rather rapidly as we saw in Figure 10.2) as an upper bound enables us to severely prune the tree quickly. The symbols correspond to the point at which the optimal solution was found.

It is still not advisable to let the algorithm finish its search. It is still very space-consuming and, in all our test cases, the optimal solution was found significantly before the peak in the node-curve. If we look at those examples that had a significant runtime (>2500 ms), we see that in all cases the best trade-off between runtime and quality was reached after 7% of the runtime, usually even in less than 1%. The best solution was found in less than 14% of the runtime, in almost all cases even less than 5%.

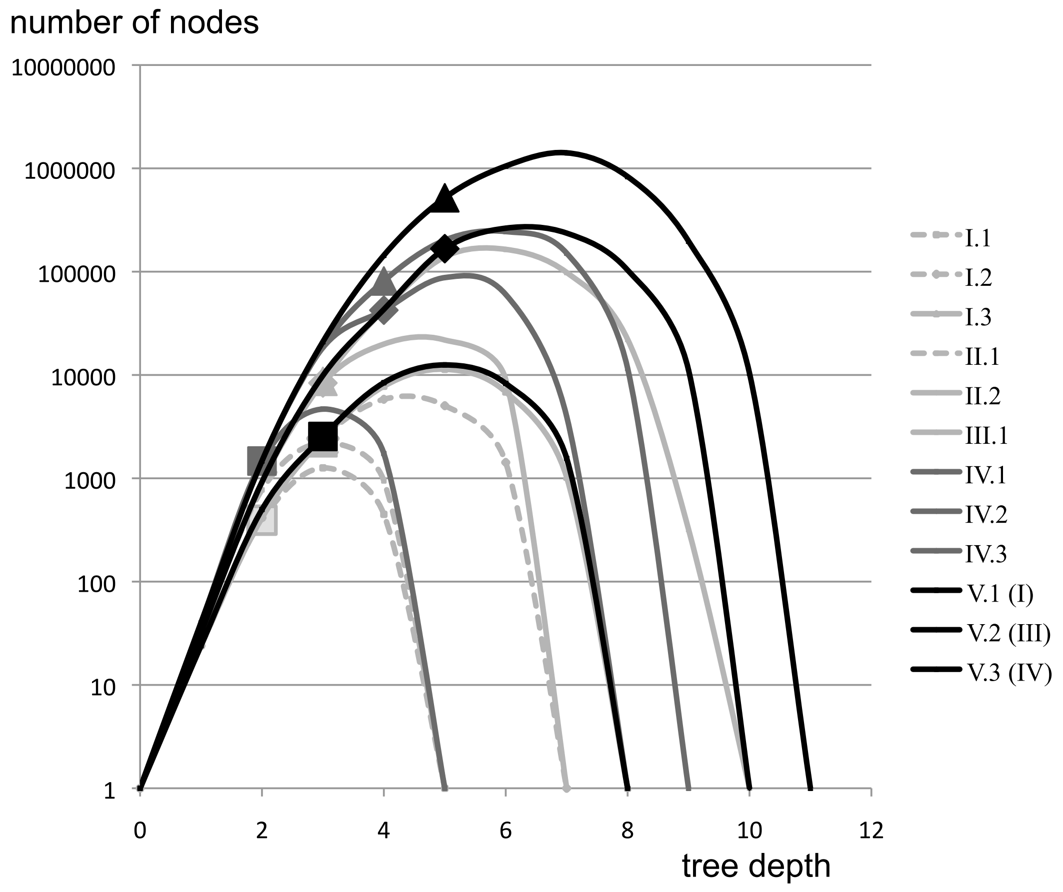


Figure 10.3: Effects of pruning.

To put these numbers in perspective of the overall timing of the information flow between calling function over processing in the HMI bundle to displaying the information on the screen: The function preparing a presentation request is outside of our implementation and the implementation of the functions is done by different developers and with different complexity. An exact timing is not available her, but it can be asumed that the proecess takes more than 25ms and less than 100ms. The connection between the function and the HMI Bundle is a function call, i.e., approximately 2ms at most. Processing the request in the bundle has been discussed in this section; 50-100ms are sufficient, however the anytime algorithm uses more time if available. The timing for sending the information from the bundle to the HMI, including rendering and displaying the presentation task has been measured by my colleague Sandro Castronovo. The values are usually under 16ms and always under 50ms.

10.2 Connection between Context and Cognitive Load

In section 9.3 I claimed that environmental factors correlate with driving performance, e.g., reaction time, and thus also serve as an indicator for the current cognitive load of the driver. In this section I will support this hypothesis with empirical data acquired in a driving simulator test.

The Lane Change Test (LCT)

Real-life field studies for evaluating driver distraction (e.g., observing accident data) are often inefficient or prohibitively intricate to accomplish. On the other hand, indirect laboratory methods measuring reaction times independent of the real driving context can be of limited validity. To overcome these shortcomings, [Mat03] introduced the Lane Change Test (LCT, cf. figure 10.4) as a measure of the influence of the secondary task on the driving performance in a simple simulator task: “It is sought to reach the reliability of a laboratory method while increasing the validity by making the cognitive requirements as similar as possible to real driving.”

The original LCT consists of a simple driving simulation at a regular consumer PC with steering wheel and foot pedals used for computer games. The subject is driving on a straight three-lane road with traffic signs indicating which lane to use (cf. figure 10.5). These traffic signs are used as stimuli, and the corresponding maneuver of the driver is the response. In between two responses, the subject still has to keep driving and stay in his current lane, which we have identified previously as a main component of the driving task. Hence, we have a significant advantage here over simple stimuli/response tests where the subject is idle in between two responses. In short: The LCT combines the advantages of a laboratory setting with the benefit of a realistic environment and task.

Three simulator runs of each subject were recorded to measure general driving performance, which was subsequently compared with driving performance while conducting a secondary task such as looking up an address from an address book, unfolding a Kleenex, tuning the radio, using a cell phone, etc.

Figure 10.4: Screenshot from original Lane Change Test (LCT).

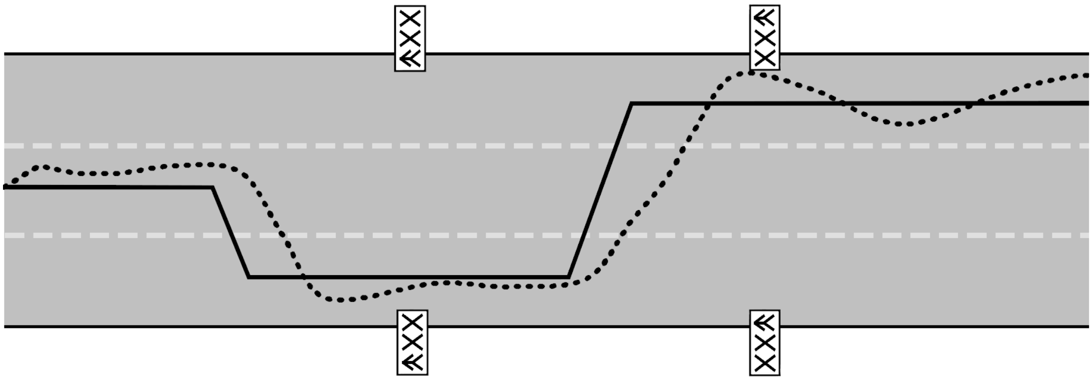


Figure 10.5: The comparison between the normative model and the driving data as a measure for distraction.

As a measure for performance, only the parts of the drive where a lane change occurs were considered. A comparison of the normative model and the driving data was conducted, and the average deviation from the optimal performance was measured and used as a measure of distraction. The difference between the average distraction without a secondary task (i.e., the driver's general performance level) and the distraction while performing a secondary task is proposed as a measure of how distracting the secondary task is.
The LCT was standardized as an ISO norm in 2008 [ISO08].

Figure 10.6: LCT Kit 4.0, based on the original Lane Change Test (LCT).

LCT Kit 4

The LCT Kit has been developed since 2009 at the Leibniz Research Centre for Working Environment and Human Factors (IfADo[1]) based on the original Lane Change Test (LCT). Its main purpose is to determine reaction times and also the connection between driver reaction times and the placement of the stimulus in the left or right visual half-field, such as [EBP+12].

In this simulation, the driver is placed in the middle lane of a straight road with a seemingly infinite number of lanes to the left and right. Except for seeing his own car interior, there are no visual distractions in the simulation (cf. figure 10.6). After a prompt, a short instruction to change lanes (either one or two lanes to the left or right) is displayed for 300 milliseconds on one side of the screen. This short time span is sufficiently long to decode the information (after a short training) but short enough to avoid saccades, which would add noise to the data to be observed. The reaction to the stimulus is measured as the time span between stimulus and a steering wheel angle outside of the ordinary lane keeping range. Furthermore, the task of changing the lane has to be completed in a certain amount of time.

[1] official German name: “Leibniz Institut für Arbeitsforschung an der TU Dortmund”

Open Drive Simulator OpenDS

OpenDS[2] is an open source driving simulator software developed in our automotive group at the German Research Center for Artificial Intelligence [MMMM12]. The lead developer of the effort is my colleague Rafael Math, who coordinates the open source project as part of the EU-project *GetHomeSafe* and with support of the *Intelligent Mobility and Transportation Systems action line* of EIT[3] ICT[4] Labs.
The driving simulator is based on the open source game engine jMonkey Engine (jME)[5]. jMonkeyEngine is an open-source game engine aimed at modern 3D game development, and it makes extensive use of shader technology. The engine is written in pure Java and supports OpenGL. It is released under the BSD[6] license and used in research/education as well as by commercial game studios.
The physics engine included in OpenDS is jME Physics 2[7], which is a wrapper for the Open Dynamics Engine (ODE). It is used to calculate relevant physical properties of the virtual world, such as velocity, acceleration, friction, etc.
The 3D models for OpenDS are generated using *Blender*,[8] an open-source 3D modeling software distributed under the GNU General Public License GPL.
The driving simulator OpenDS provides: speedometer and rev meter panel, rear-view mirror, and basic audio support for positional and directional sounds.
Going beyond game-like simulation, OpenDS provides not only an interface for game-controllers, but also a CAN interface, which we use to connect a real car and read out its state in real time, especially pedal states and steering wheel angle. In that way, a more realistic simulation is provided.
Multiple screens are also supported for a projection of up to a 180 degree viewing angle. Our current installation at the ZWM[9] uses three projectors and a viewing angle of 120 degrees (cf. figure 10.7).
OpenDs was developed with the intention of having a flexible driving simulation which can perform the standard Lane Change Test (LCT), but can easily be extended to other/similar/new testing tools beyond the possibilities of LCT, since the original LCT is very restricted and not extensible. The ConTRe task [MFMM12] was the first extension realized in OpenDS.

[2] www.gethomesafe-fp7.eu/index.php/menu-opends
[3] European Institute of Innovation and Technology, www.eit.europa.eu
[4] Information and Communication Technologies, eit.ictlabs.eu
[5] jmonkeyengine.com
[6] Berkeley Software Distribution
[7] code.google.com/p/jmephysics/
[8] www.blender.org
[9] Zukunkftswerkstatt Mobilität (future mobility lab)

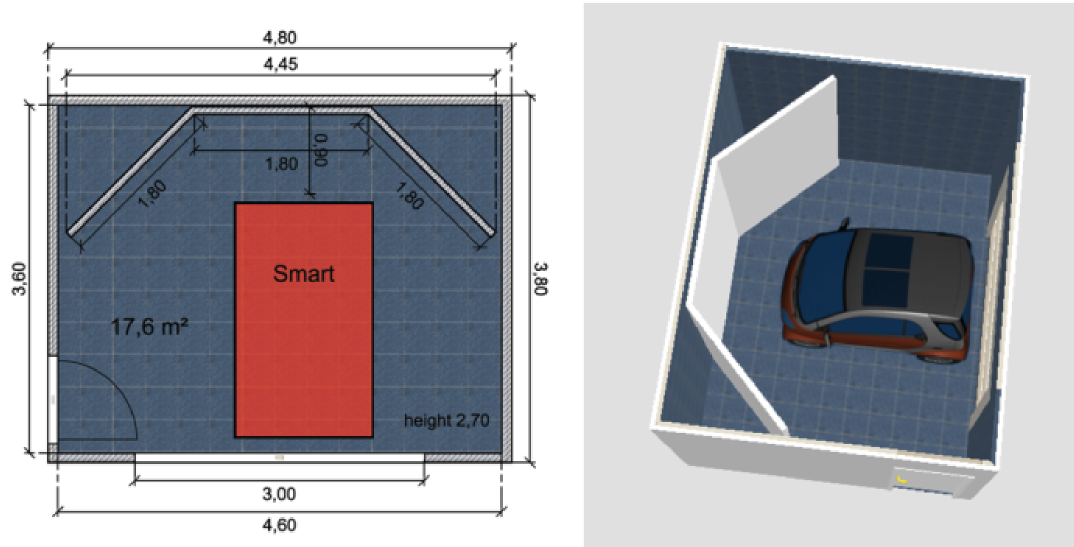


Figure 10.7: The OpenDS simulator lab at Saarland University.

Distraction Experiment

The LCT Kit used in training science was designed to measure the driver's reaction time to instructions shown briefly on either the right or left visual half field, short enough to avoid saccades, and to measure the difference in reaction time between information shown on the left or on the right.
As part of our collaboration, we ported this task to OpenDS, and we use it in our driving simulator installation. In order to use it here for the task at hand, some modifications have been made. Figure 10.8 shows the connection between the original Lane Change Test, the LCT Kit, our OpenDS simulator and the distraction experiment described here.

Objectives

As the aim is to show the correlation between contextual complexity and cognitive load, we change the parameters speed and visual complexity.
Visual Complexity. The original experiment uses infinite lanes and offers no visual distraction for the subject. In a first step, we reduce the number of lanes to a necessary minimum of five and fill the remainder of the visible plane with monochrome shading. To avoid predictability (e.g., the subject knows the next instruction must be left because he is on the far right lane), the driver is automatically centered again on the middle of 5 lanes by adding or removing lanes on the respective sides.
Now, we introduce visual complexity by adding objects to both sides of the street. To keep different experimental runs comparable, all objects are of the same size and

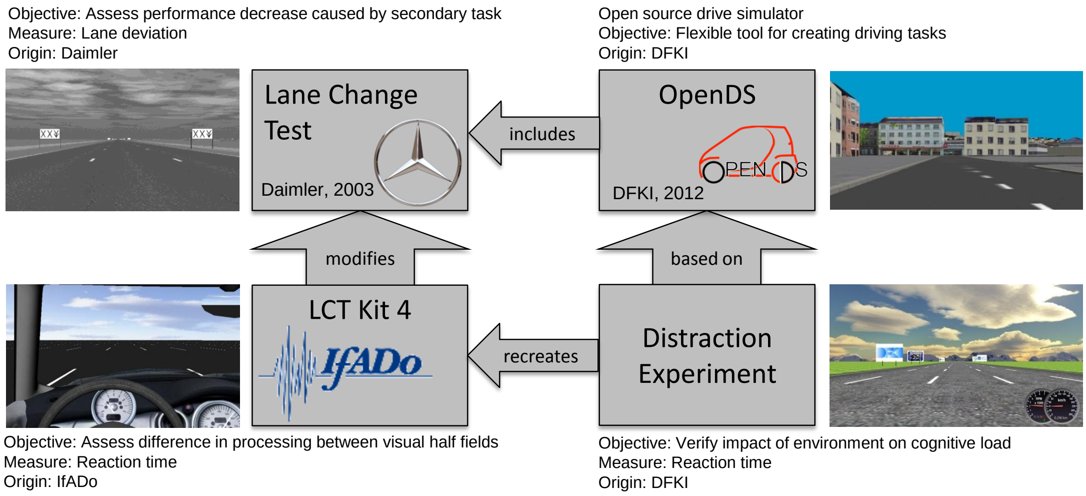


Figure 10.8: Connection between the distraction experiment and related software tools.

visual complexity. i.e., identical models. The hypothesis to be verified in the experiment is a positive correlation between visible objects in the subjects view field and the reaction time during the experiment.

Vehicle Speed. The second assumption to be verified here is the correlation between vehicle speed and reaction time. We extend the original experiment and run it at different speeds. This assumption is closely related to the previous assumption, as the higher vehicle speed results in faster changes to the visually perceived environment.

Setup

We used three variants of visual complexity (no distractions, some billboards on the side of the road, many billboards on the side of the road) and two different speeds (40 km/h and 120 km/h). The test took place in a real car positioned in front of three projection walls covering a visual range of 120 degrees.

Experiment

In the experiment (cf. figure 10.9), five subjects drove in the simulation under varying conditions. Their task was to react as quickly as possible to lane change commands displayed on the screen. In order to observe whether or not any training effect in getting used to our simulator occurred, three of the subjects were asked to perform the test twice in a row.
The results clearly confirmed our two hypotheses: (1) The average reaction time increases with increasing speed. (2) The average reaction time increases with the number of distracting objects (billboards) on the roadside. A more detailed look at the data is shown in table 10.1.

Figure 10.9: Experiment based on the Open source Driving Dimulator (OpenDS).

An interesting effect can be found when looking at the reaction time under both varying speed and varying number of distractions: While at high speed the reaction time increases with the number of distractions, the exact opposite can be observed at slow speed. Our first assumption that this can be attributed to training could not be confirmed, since the effect prevailed when considering only the three test runs of subjects performing the test for a second time.

	slow driving	**fast driving**
no distraction	1380 ms	1387 ms
medium distraction	1344 ms	1464 ms
high distraction	1126 ms	1783 ms

Table 10.1: Average reaction time in ms under varying conditions

Differences Between a Race Car Driver and the Average Driver

As a part of our collaboration with the institute of sports sciences, we started to investigate objective metrics for assessing the talent of aspiring race car drivers. Additional data collected in this experiment provided some evidence for one of the hypotheses:
I claim that trained, professional drivers focus more on the task at hand, and get less distracted by the environment. If a direct connection between the effect of distraction to the driver on one hand and the qualification of a driver for race driving on the other hand could be established, then this could be used as one component in a complex evaluation process.

Figure 10.10: Distraction experiment with young race car driver (courtesy of E. Messina).

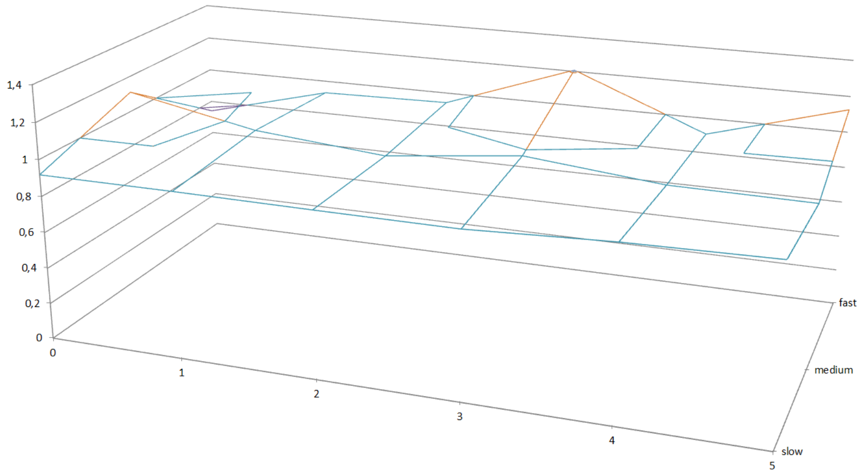


Figure 10.11: Evidence for a measurable difference between drivers ans race car drivers.

To gain a first insight in this area, we invited kart race talent Chiara Messina (cf. figure 10.10) to partake in the experiment. The results of this preliminary test are shown in figure 10.11: While the effect of higher speed levels to the reaction time is visible, the impact of the different levels of distraction is barely noticeable. It is however important to keep in mind that this is only first evidence and not sufficient yet for a scientific theory.

Resulting Model

The PRESTK implementation is based on a more detailed collection of data in the same setup. Using the eyeBox system developed in our group, the environment of the car can be assessed and the amount of visual distraction estimated. The current speed of the car is assessed using the car's CAN bus. These two data sources serve as the foundation of an extensible model to estimate the cognitive load of the driver in real time.

10.3 Annotated Complexity Estimation ACE

In section 9.6, the Annotated Complexity Estimation (ACE) procedure [EFM12a] has been introduced, but the empirical verification was yet pending. The aim of the ACE procedure is to provide a metric for the complexity of a presentation by assigning a numerical value based on the layout and the elements of the presentation. Under the assumption that the perceived complexity of a piece of information can be judged objectively by the recipient, I designed an online study attempting to verify the values calculated by ACE. In this study, 30 screenshots of the actual sim^{TD} system with varying complexity are presented to the participant in random order. The task is to assign a numerical value of complexity in a range from zero to ten to each screenshot (cf. figure 10.12). The average value assigned to a screen is used as ground truth and compared with the value calculated by ACE.
We distinguish two phases here, the development phase and the evaluation phase.
After the experiment, the screens are sorted by average user rating and devided into development data set and control data set. For the control data set, 10 evenly distributed screens are selected and set aside. The remaining 20 screens are used for development. In this first phase, these 20 screens are used to fine tune the values used by ACE. They are annotated according to the ACE notation and their complexity values are calculated using the parameters shown in table 9.1. These values are an interim consent and the aim of this experiment is to refine them. We first compare the results of the interim ACE parametrization to our ground truth given by the average user rating. Results are shown in figure 10.13 in the upper left graph. On the x-axis, the average user rating is displayed, and on the y-axis the calculated values.
Tweaking the ACE parameters for the development data set is achieved by optimizing the correlation coefficient (sometimes also called cross-correlation coefficient) of the two data sets $\mathcal{X} = \{x_i, i \in \{1,..,n\}\}$ of complexity values based on user opinion and $\mathcal{Y} = \{y_i, i \in \{1,..,n\}\}$ of values calculated by ACE, whereas n is 20 in this case[10].
The correlation coefficient is a value which describes the quality of the least square fitting of the scatterplot to the covariance.
In order to define the correlation coefficient, we first calculate the unnormalized forms of variances and covariance of $\mathcal{X}$ and $\mathcal{Y}$ as the sum of squared values:

$$var(\mathcal{X}) = ss_{xx} = \sum_{i=1}^{n} x^2 - n\bar{x}^2$$
$$var(\mathcal{Y}) = ss_{yy} = \sum_{i=1}^{n} y^2 - n\bar{y}^2$$
$$cov(\mathcal{X},\mathcal{Y}) = ss_{xy} = \sum_{i=1}^{n} xy - n\bar{x}\bar{y}$$

[10]More information on the theory of statistics described here can be found in numerous textbooks, cf. for instance [Bor85], or online at mathworld.wolfram.com/CorrelationCoefficient.html

Figure 10.12: Online study.

The correlation coefficient r can now be defined as

$$r^2 = \frac{ss_{xy}^2}{ss_{xx}ss_{yy}}$$

Optimizing the parameters defined in table 9.1 can now be reduced to maximizing r with the aim of getting as close as possible to 1.0. In case of the optimal value of 1.0, all data points would be on the same line. As we can see, this is not the case, but the scatter plot is not completely random: it tends toward a line, and the correlation coefficient already reaches a value of 0.828. This is a high value, but it is reasonable to believe that the weights based on expert opinion still can be improved. I use a custom implemented genetic algorithm for optimizing the parameters, which is adapted

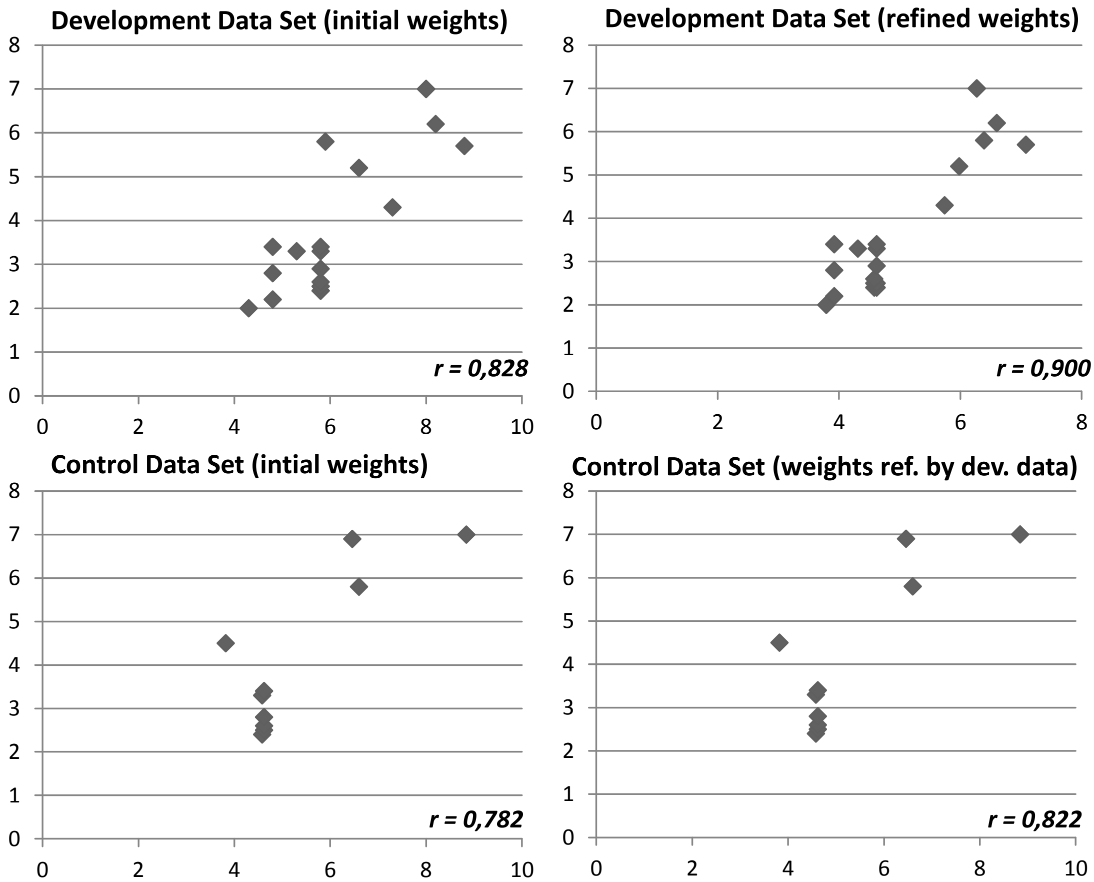


Figure 10.13: Comparison of the development data set with the values calculated.

to solve the problem at hand quickly (cf. algorithm 2).

In lines 8-10, each weight is consecutively modified by a given stepwidth. The modify-function (not shown here) calculates the correlation coefficient for the given weights, and also for two modified versions: in one, the weigth at position *idx* is increased by *step*, in the other version, the same parameter is decreased by *step*. The version with the highest correlation coefficient is returned. Meanwhile, the number of changes while modifying the weights is counted and the procedure repeated until no changes occur anymore (lines 5,6,11). After that, we go even deeper into finetuning and switch to half the stepsize repeatedly until we are below a certain treshhold (lines 4,12,13). As a result, we obtain a new set of weights for the ACE procedure. These weights are shown in table 10.2 . The changes compared to the initial values shown in table 9.1 (also in parentheses in table 10.2) are minimal, with one exception: our estimate of the weight of framed and unframed panels were reversed.

Using these refined values, the correlation coefficient can be increased to 0.9, and the resulting change in the scatterplot in figure 10.13 on the upper right diagram are an obvious improvement.

Algorithm 2 Optimizing parameters of the ACE procedure

initialweights $\leftarrow \{0.1, 0.5, 0.4, 0.1, 0.0, 0.5, 0.2, 0.4, 0.2, 0.5, 0.2, 0.5\}$
currentweights $\leftarrow$ initialweights
step $\leftarrow$ 0.1
while step $>$ 0.0001 **do**
    (static) changes $\leftarrow$ 1
    while changes $>$ 0 **do**
        changes $\leftarrow$ 0
        for all i $\in$ $\{0, |initialweights|\}$ **do**
            currentweights $\leftarrow$ modify(currentweights,i,step)
        end for
    end while
    step $\leftarrow$ step/2
end while

To show the general applicability of the newly obtained weights, we take a look at the control data set (cf. bottom graphs in figure 10.13). As this set is half the size of our development set, the correlation coefficient is not as high as in our development set, however we are still able to see a significant improvement between the initial and the weightsrefined by development data, which increases the correlation coefficient from 0.782 up to 0.822. This shows that we did not optimize any effects on the development data set but found a broadly applicable solution.

component	**basic complexity value**	**feature**	**added complexity**
label	0.1 (0.1)	text=true	+0.425 (0.5)
		icon=true	+0.400 (0.4)
icon	0.1 (0.1)	type=empty	+0.000 (0.0)
		type=icon	+0.388 (0.5)
		type=static	+0.250 (0.2)
		metainfo=text	+0.500 (0.4)
panel	0 + $\sum$ child nodes	decoration=framed	+0.622 (0.2)
		decoration=none	+0.325 (0.5)
		metainfo=named	+0.275 (0.2)
		metainfo=none	+0.300 (0.5)

Table 10.2: Empirical values for ACE evaluation based on online study.

Overall, we have shown that the ACE procedure yields values that are comparable to the subjective perception of the average user.

Now that I closed the remaining gaps in the theoretical contribution through experimental evaluation here, I will discuss some additional features in the next chapter.

Chapter 11

Discussion of Additional Features

The extensions of the presentation manager and the lessons learned from sim^{TD} are not limited to adding system situation awareness as a new feature as proposed in chapter 9. In this chapter, I briefly present three other system extensions and discuss their applicability. While some are based on practical experience in the testing area and should be included, others are theoretical concepts developed beforehand that turned out not to be useful.

11.1 Reactive Scheduling

The longest and most vivid discussion I had during the sim^{TD} project with developers of the several information functions was about the question of how far in advance a function should announce its request to be presented on the screen. Being responsible for the presentation management, my point of view was very firmly that–unless there is an emergency–the pre-announcement window of time should be the minimal presentation duration of a presentation that might just have been started (usually four seconds) plus some time for scheduling the new presentation. To be on the safe side, I assumed one second, although as shown in section 10.1 less time is needed. The resulting rule in the HMI bundle was: every function which is not safety-critical needs to announce a request for presentation five seconds beforehand.
It turned out that this requirement was impossible to fulfill for almost every function. In consequence, one of the main challenges in designing an HMI and its underlying logic is the presentation of spontaneously incoming warning messages without delay, while at the same time disrupting the current state as little as possible. Also, inconsistencies in the visualization and irritation of the driver have to be avoided. This is not always possible, for instance an incoming warning of high importance might need to override a presentation on the screen which just started half a second before. This causes an effect of "flickering", which is in the described case unfortunately inevitable. Still, the safety of the driver has priority over potential esthetical qualms.

But if the lack of advance notice becomes the standard case instead of the exception in critical situations, some ways to handle it need to be introduced.
I have already described in section 9.8 that the tree-search algorithm I am using is very flexible, and emphasized how easy it is to modify it to include new functionality. We can apply this here as well:
If sufficient time for planning ahead is not available, the planning process changes from looking at an interval of time in the future and resolving potential conflicts there to a reactive decision, regarding what to do with an incoming presentation request of a certain priority with some options for modification under the given condition, which is the current state of the display and its history. Formally, we have to change two things:
1. the problem definition, and
2. the calculation of possible modifications.
I will describe the necessary adaptations for both and thus provide the necessary foundation for adjusting the implementation.

Problem Definition

The tree search algorithm starts with the problem description as its root and successively makes all possible changes to all parts of the problem in order to find a modified version of the problem without a conflict, e.g., a solution. By evaluating the quality of the solution and successively replacing less optimal solutions with better ones, anytime functionality is achieved.
In the original implementation, the problem consisted of a set of potentially overlapping presentation tasks to be presented on one or more scarce resources, i.e., the screen. The possible modifications included temporal adjustments, change of display strategy, and canceling a task.
Here, the situation is different. We have an incoming presentation task requesting immediate presentation, and a current state of the screen (cf. HMIstate), along with its recent history.
The HMIState needs to keep references to the presentation task objects which are presented on its leaves, so that information such as when a currently displayed presentation started and how long its minimal presentation duration is can be accessed. Fortunately, this is already the case in the sim^{TD} system, so very little additional information has to be included, e.g., the timestamp of the last user interaction. Figure 11.1 shows the extended HMI state.

Possible Modifications

The set of available modification possibilities is guided by usability rules:
A *minimal presentation duration*, adjusted to the importance and complexity of the contained message is stored in the system for each function (cf. section 9.5). This

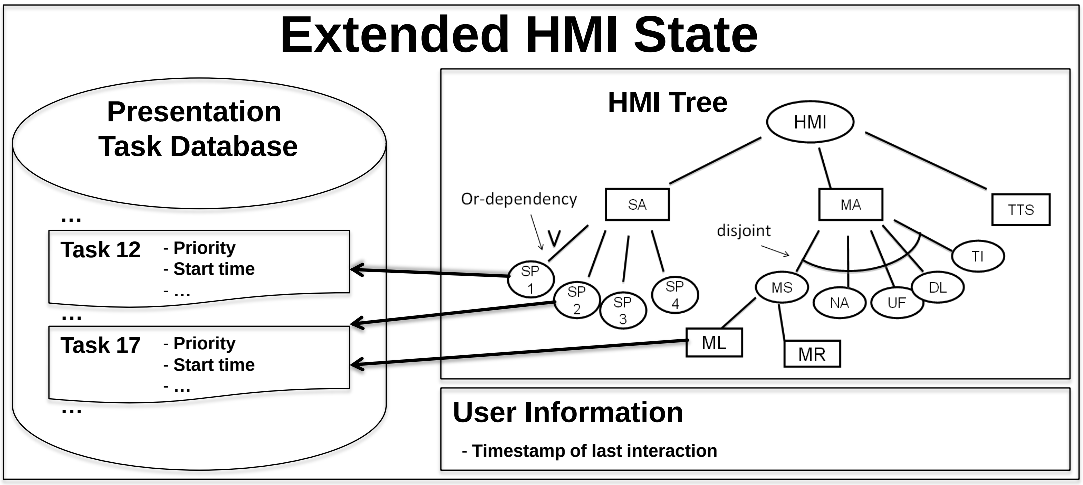


Figure 11.1: Extended version of HMI state for reactive planning.

prevents the driver from being irritated or distracted by a short message flashing up: The presentation of a piece of information for a time period too short for processing it should be avoided due to its possible negative impact on the driver's focus. However, even if a presentation task is accepted by the HMI as it fulfills the criterion of a sufficiently long presentation duration, the requesting function is still able to cancel it too early while on the screen. This also results in an undesired flickering, which is out of the scope of the HMI. In the sim^{TD} testing area, we displayed warning messages against undesired use of the HMI, in order to bring this problem to the attention of the function developers.
User interactions are respected by the HMI and only interrupted in emergencies. If the user selects a presentation task from the symbol area to be presented on the main screen, no interruption will take place for the next three seconds by an incoming presentation task of lower priority. If the incoming task requests the use of the main screen, it will still be moved to the symbol area only. If the incoming presentation task however has higher priority and is a warning message, the user interaction is ignored and the incoming message is shown on the main screen.
By considering these rules in the implementation of the possible modifications of a conflict set, reactive planning is realized.

11.2 Conflict Detection

The task to be solved by the presentation manager is a conflict-free positioning of potentially overlapping tasks on a timeline in a dynamic manner. In order to facilitate the example to discuss here, we assume only one resource, so that every overlap in the timeline is a conflict. Also, the presentation manager is assumed not to be situation-aware (as the example stems from the sim^{TD} development). The possible actions for

the tree search algorithm are also from the original sim^{TD} planner, i.e., presentations can only be shortened at the beginning or the end, moved forward or backward, or be skipped altogether.
As an example, we consider 8 presentation tasks of four seconds length in a 30 second interval starting 10 seconds in the future. Priorities of the tasks differ. Table 11.1 shows start time, end time, and priority of the tasks.

#	start time	end time	priority
1	10	14	40
2	14	18	40
3	18	22	40
4	22	26	40
5	26	30	80
6	26	30	80
7	33	37	60
8	38	42	80

Table 11.1: Scheduling problem example

Figure 11.2 shows the tasks on a timeline in the upper part. The different priorities of the tasks are encoded by the use of different colors.
The question arising here is: Which of these tasks are part of my problem, i.e., which tasks potentially need to be modified, and which ones definitely do not? As our approach to solving the problem is a tree-search algorithm with a underlying data-structure growing at the beginning exponentially by the number of tasks involved, it is advisable to avoid unnecessary inclusion of unrelated tasks.
It seems trivial at first, since, looking at the timeline, it is plainly visible that postponing task seven one second will suffice to make room for either task five or task six to be postponed by four seconds, which results in a conflict-free schedule. Thus, our problem set consists of tasks five, six, and seven.
However, if the solution had included preponing either tasks five or six, things would have gotten more complex, involving presentation tasks one to four as well.
I propose a heuristic to solve this problem.
In a first step, temporal areas of conflict are identified and the conflict level calculated. An area of conflict here is a section of the timeline where two or more tasks overlap. The conflict level is defined as the sum of the priorities of the overlapping tasks. The conflict level graph in this case is simple: We have an overlap of tasks five and six at [26,30] with a constant conflict level of 160, while everywhere else the conflict level is zero.
In the next step, a gauss curve (or in more complex cases: one gauss curve per conflict area) is fitted to the graph so that it covers the conflict area. This corresponds to the assumption that the probability of a task being involved in a conflict is 1.0 in the middle of a conflict region and declining according to a standard distribution with the

distance to the conflict. As a second graph, the time available for solving the conflict is applied. It consists of a straight line representing the identity function: The time remaining to solve the problem is identical to the time left until the conflict.
The intersections of this line with the gauss curve(s) determine the start and endpoint of the conflict. Presentations contained (even partially) are considered part of that conflict.
This provides a first approach for solving this problem, which needs to be refined and parametrized further. The gauss curve, for instance, can be flat or high (an alternative version is shown in the graph).
It turned out however that this problem is purely of academic interest in the given domain, since (a) usually not much time to plan ahead is available in real-life applications and (b) the density of tasks occurring in real-life is not anywhere close to requiring this solution.

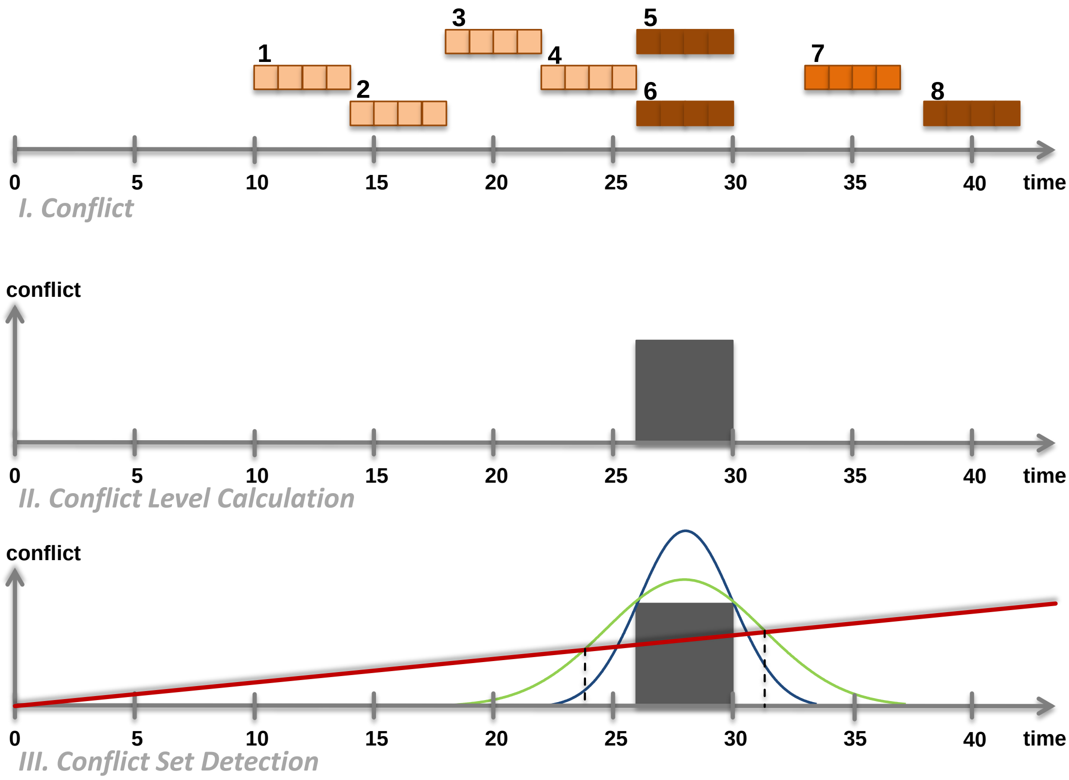


Figure 11.2: Conflict Detection Approach.

11.3 Low Cognitive Load Consideration

The Yerkes-Dodson-Law (cf. section 5.3.1) implies that individual performance is impaired both in case of high arousal and in case of low arousal. While the former case is covered by the PRESTK system described in this thesis, the latter has not yet been considered.

[DHM98] for instance reviewed the effect that too much automation in the car can lead to a significant performance decrease in case of a sudden system failure. The authors conclude that cognitve workload does not necessarily decrease with automation, since it is beyond a certain level likely to induce fatigue and decrease task motivation. [HHMK00] classified existing fatigue detection techniques and discussed the means to evaluate their effectiveness.

The method of estimating cognitive load proposed in this thesis entails a focus on the context/situation, e.g., things that occur or happen. The "left side" of the Yerkes Dodson Law however is concerned with the lack of things happening, which might induce boredom in the driver and result in decrease of performance.

While this might be seen as a drawback in my approach, I claim that it can also been regarded as an advantage: If we consider only the cognitive demands of the context, it becomes very easy to detect critically low demand times. A combination of a lack of social interaction (i.e., no co-driver or passengers), a high level of automation resulting in limited system interactions, and a quiet driving situation lead to a situation where the driver's lack of arousal could significantly impair his ability to handle a sudden system failure or unexpected event.

Counteracting measures proposed by [DHM98] imply that a limited need for system control can be balanced with an option for increased system monitoring: If not much is happening while driving, we can provide system information on the screen giving feedback on driving style, fuel consumption, or compliance with an ecological life style. [EMB10] for instance implemented a prototype of the youldeco[1]-system to make eco driving an entertaining and competitive game in a social network (cf. figure 11.3). The precondition for this is the availability of low cognitive demand times, which matches our given problem here very well.

But there are also other approaches to providing entertainment for the driver. Literature discussed in chapter 6 includes more approaches:

[AKS$^+$10] conducted a survey to determine which kinds of entertainment are most desirable for low demand times, and proposed an algorithm to detect such time windows. Their example was the detection of standing phases at red lights. Using the context evaluation described in this thesis, this could be extended to a more general approach for entertainment in low demand times. [TWV11] propose a speech-based system, which can be used for driver entertainment as well. [DS11] investigate the cognitive impact of linguistic complexity in dialog systems. This as well can be used as a foundation for an entertainment system.

In short: A lot of information which is contained in the various systems of the car, or can be obtained by communication means (e.g., news streams), can be put to use to prevent a driver from getting bored when driving alone on a uneventful drive.

[1] youldeco is an acronym for *You'll drive eco-friendly*.

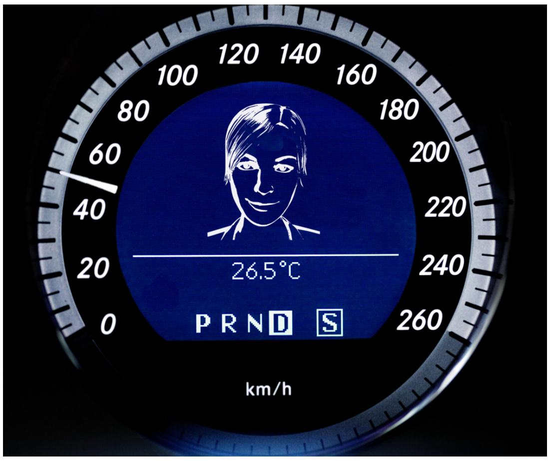


Figure 11.3: The youldeco approach for eco-friendly driving and avoiding performance decrease.

11.4 Summary

In this chapter, I proposed three extensions to the situation-aware approach presented previously:
Reactive scheduling is an important modification of the scheduler and is in part already included in the PRESTK system.
Conflict Detection, or rather, the detection of the subset of given tasks included in a problem, can get very complex. I propose a heuristic, which is not included in the implementation, since in real applications reactive scheduling is required instead of planning ahead with a large number of tasks on a timeline.
Preventing boredom is important since boredom decreases the performance of the driver. Countermeasures are discussed.

Chapter 12

Conceptual Achievements

I started this section by identifying the conceptual requirements for system situation awareness. I then presented the solution provided for the sim^{TD} project in chapter 8, and introduced system situation awareness for the more advanced PRESTK system in 9. Experiments in chapter 10 supported the claims I made. Some more features for PRESTK were discussed in 11.
Before I begin with the description of the implementation, I will go back to the beginning and take a look at the requirements formulated and describe what has been achieved. Evaluation of the single pieces will be summarized, and ways to demonstrate the achievements presented.

12.1 Revisiting Requirements

The main requirement formulated in chapter 7 concerns solving the search problem in the three dimensions time, modality, and complexity:

1. **Time:** By formulating the Resource-Constrained Scheduling Problem (RCSP) and the tree-search approach as a solution in the sim^{TD} project, management of presentations on the temporal axis has been achieved. Additional requirements regarding whether, to what extent, and how a task can be modified are considered.

2. **Modality:** Choice of modality is implicitly achieved as well by the tree-search solution by defining so-called display strategies.

3. **Complexity:** By using the Annotated Complexity Estimation (ACE) procedure defined in section 9.6 and modifying the algorithm according to section 9.8, presnetation complexity is taken into consideration.

Further requirements for the proposed presentation toolkit were:

- **Extensibility.** The claim of a system's extensibility to yet unknown features cannot be easily verified. It has been shown twice however, that the original tree search algorithm can be extended to new situations (system situation awareness in section 9.8 and reactive scheduling in section 11.1). Additionally, the model for how to obtain cognitive load from context information is open to inclusion of new parameters. By using the KAPcom database, data exchange with additional new systems is also possible. And furthermore, the ACE procedure for presentation complexity calculation is explicitly designed for being extended, both in the number of possible elements and in feature evaluation.

- **Dynamic.** Already in the first implementation for the sim^{TD} system, it was possible to change presentation time, duration or any other parameter of a presentation even after the presentation task was created. The only exceptions are obvious and inevitable, e.g., that the start time of a presentation can't be changed after the presentation has already started.

- **Anytime Functionality.** An anytime algorithm is commonly defined as an algorithm that can return a valid solution to a problem even if it's interrupted at any time before it ends. The algorithm is expected to find better and better solutions the more time it keeps running. The empirical evaluation in section 10.1 shows that a good solution to a problem can be found very quickly, and by iteratively refining it, the algorithm can be stopped at any time after the first few milliseconds and produce a reasonably good result.

- **System Situation Awareness.** In chapter 9, I defined system situation awareness (SSA) according to Endsley's model. By following the three levels of SSA both for the driver and for the information to be presented, an overall concept of how to implement this feature has been presented.

- **Cognitive Load (CL) Assessment.** I claimed that most CL assessment methods, despite being very useful for user studies and development of concepts, are in most cases not really desirable for everyday driving. An exception is, for instance, steering wheel angle observation, which can be performed unintrusively, but according to [SP11] produces suboptimal results. As an alternative, I propose deducing CL from contextual parameters. In a user study (cf. section 10.2), the general applicability of that approach has been shown for two selected parameters.

- **Presentation Complexity Estimation.** The Annotated Complexity Estimation (ACE) principle has been introduced in section 9.6 and empirically verified in section 10.3. Further refinement is possible and planned for the SiAM project.

- **Cognitive Load Consideration.** A method for considering the cognitive load of the driver is achieved by modifying the tree-search algorithm according to section 9.8.

- **Formalization of acquired information.** While defining and achieving system situation awareness, required concepts have been formalized. On the driver side, this includes cognitive load, elements of the driving task, processing resources, and possible distractions. The latter two concepts have been thoroughly classified in figure 9.4 and 9.5 respectively. The Automotive Ontology of KAPcom was extended to store required information about the cognitive state of the driver. On the presentation side, the container language PTCL was introduced, which can serve as a wrapper for a variety of formats and at the same time handle detailed complexity annotations.

12.2 New Concepts

The following concepts have been newly introduced and are part of my contribution to the research field:

- The Resource-Constrained Scheduling Problem (RCSP).
- A tree-search algorithm to solve the RCSP.
- System Situation Awareness, based on Endsley's model for Situation Awareness, including a description of how to implement it.
- Classification of the driving task, based on [Sch93].
- Classification of processing resources, based on [Sch93] as well.
- The connection between driving task and processing resources.
- The POLI classification for driver distractions.
- A model for cognitive load at different levels of detail for the Automotive Ontology used in KAPcom.
- The Annotated Complexity Estimation (ACE) principle for determining presentation complexity.
- The container language PTCL, which can handle complexity annotations in different presentation languages.

12.3 Overall Evaluation and Demonstration

The single parts of the PRESTK approach have been empirically evaluated in chapter 10. What is missing now is the overall evaluation of the single parts working together. Building a demonstrator for an empirical user experience is an appropriate means to do so.
In this section, I describe two demonstrators for PRESTK.

12.3.1 Demonstrator 1: Displaying the driver's state in the Simulator

The first demonstrator is based on the experiment described in section 10.2 and reverses the acquired model. In the experiment, I varied the number of visual distractions and the speed of the car, and measured the reaction time, which can be used as an indicator for cognitive load. In the demonstrator described here, the driver uses the simulator and drives on the same road as in the experiment, with a random and varying number of visual distractions, and is able to adjust the speed. On the screen in the car the current speed, number of distractions, and estimated reaction time is displayed continuously.
The model derived from the user study can be described with the following formula:

$$reactionTime = 1380 + (numBillboards * ((1.625 * speed) - 115))$$

Like the user study, the demonstrator is implemented using the OpenDS simulator developped in our automotive lab. Figure 12.1 shows a photo of the running demonstrator (left) and a screenshot of the screen displaying the driver's estimated reaction time based on speed and visual distractions (right).

Figure 12.1: Reaction time estimated by speed and visual distraction displayed in the car.

12.3.2 Demonstrator 2: Adaptive system on the road

The second demonstrator uses the EyeBox system [MM12, MFM12, Mon11] and extends it with PRESTK and especially the techniques shown in the first demonstrator towards situation-aware information presentation.
The demonstrator uses our test carrier vehicle BMW 535i with a built-in PC and two additional screens on the dashboard. The chosen location is the Saarland University campus site in Saarbrücken (cf. figure 12.2).

Figure 12.2: Route for the demo on campus site of Saarland University (source: Google Earth).

The overall architecture is shown in figure 12.3. The components involved are described below.

The cars built-in **GPS module** continuously provides information about the current position of the vehicle during several rounds on the campus. This information is fed into the EyeBox system.
EyeBox was originally developed for multimodal reference resolution and used as a proof of concept for mobile spatial interaction in urban environments.[1] Using data from an eyetracker along with information about the current location of the car and a detailed 3D-model of the surrounding, the EyeBox system is able to answer questions by the driver of the type “What is this building/object?”, referring to his current glance position. Even more sophisitcated: the position of the car and the direction of

[1]Please note: The quoted references [MM12, MFM12, Mon11] describe the EyeBox system without mentioning its name because they were published before the system was named.

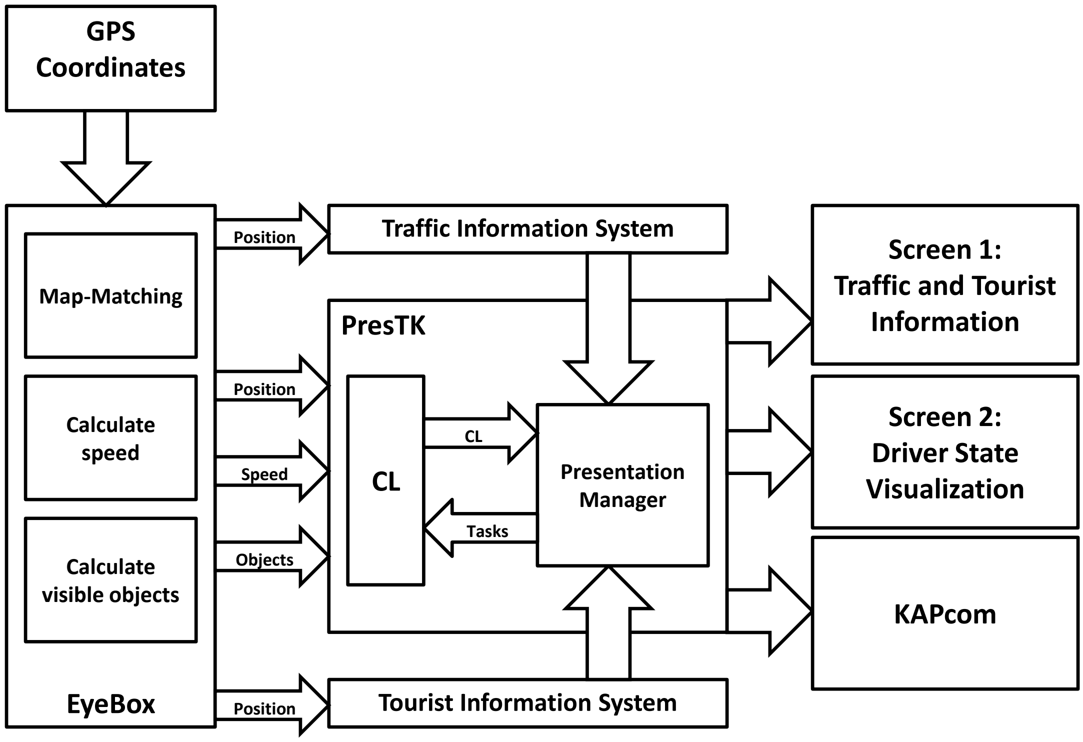


Figure 12.3: Adaptive system for second demonstrator

the driver's glance at the moment the question was uttered can be reconstructed afterwards. The desired information is then selected from a database and displayed on the screen.

In the demonstrator described here, the task of the EyeBox component is threefold: (1) provide current position of the car using GPS data and refine it using mapmatching techniques, (2) provide current speed based on the improved position data, and (3) constantly count the buildings in the viewfield of the driver. In analogy to demonstrator 1, we now use buildings as measure for visual distraction instead of billboards.

The **Traffic Information System** and the **Tourist Information System** are two independently running information sources generating conflicting presentation tasks. While the Traffic Information System uses important driving related presentation tasks from the sim^{TD} system such as construction site information or warnings about obstacles on the road, the Tourist Information System provides edutainment information about objects in the viewfield, such as the sculpture "Torque" by the famous american minimalist sculptor Richard Serra[2], which was errected on the campus in 1992. The system remembers which information has already been presented to avoid annoying the driver with repetitive presentations.

PRESTK uses the information received from EyeBox via remote procedure call (rpc)

[2] www.moma.org/serra

to orchestrate these two information systems. The driver's cognitive load is estimated based on the current speed, the amount of distractions in the viewfield, and a history of previously presented information and their complexity. For the cognitive load induced by the latter, a linear decay is assumed, while for the other two sources of distraction the momentary state is used as input to the calculation. Using this information, the presentation manager choses the appropriate information to present to the driver, based on information complexity and the driver's current cognitive state.
The output of the demonstrator is now used in three ways:
(1) The primary screen on the driver side shows the orchestrated presentation tasks.
(2) The secondary screen facing at the codriver's seat shows information about current position, calculated speed, number of buildings in the viewfield, estimated cognitive load of the driver, and status of currently shown (or canceled) presentations.
(3) Whenever the calculated estimation of the driver's cognitive load changes, the user modeling database KAPcom is updated with the current value.
KAPcom is used in this demo solely as information storage. To fully use the potential of this component, the demo could be extended to use personalized information about the driver to determine which of the over 50 presentations of building or object information might be of interest. At the moment, no such modeling of the driver is available as it is out of the scope of the concepts to be demonstrated here.

Figure 12.4: Demonstrator 2 inside the BMW.

Part IV

Implementing the Presentation Toolkit PRESTK

Chapter 13

The PRESTK Toolkit

13.1 Requirements

The requirements for the implementation of the presentation toolkit PRESTK can be summarized as “extensible prototype of a component-based presentation planner toolkit realizing the concepts defined in the previous part.” To be more precise, the reference implementation of PRESTK presented here contains:

- A selection of presentation managers. Different technologies, such as *rule-based* approaches or *genetic programming* were implemented and tested. Due to requirements such as anytime behavior, the *tree search* scheduler became the most advanced and most detailed tested scheduler for the toolkit. It exists in two different versions, one for planning ahead, and one for reactive planning, as described in section 11.1.

- The default tree-search presentation manager features *anytime behavior*, *intelligent pruning*, as well as *dynamic replanning*.

- PRESTK takes into consideration the *complexity* of presentations. As we have seen previously, three types of complexity assessment are possible: beforehand, at runtime using structured data, and at runtime using unstructured data. The first case does not require any implementation, since the presentations can be annotated beforehand by experts using the *PTCL* language defined in section 9.7. The third case is rather complex and out of the scope of this thesis. This leaves us with the annotating of structured data at runtime. An implementation of *ACE* is included in PRESTK in order to assess the complexity of a presentation at runtime.

- PRESTK takes into consideration the *cognitive load* of the driver. Based on the model derived from the user study in section 10.2, an extensible version of a component for estimating cognitive load from contextual parameters is implemented (as demonstrated in section 12.3).

- A presentation manager is a highly complex concept that requires extensive monitoring and supervision during the implementation phase. A simulation environment with visualization of the inner workings of the system as well as extensive logging possibilities is added to the PRESTK implementation.

In the next sections, I will describe the implementation of the PRESTK toolkit in more detail.

13.2 The Simulation Environment

A simulation environment is necessary during the development phase of a complex system such as a presentation manager. Since several components are working in parallel and the overall system behavior is working in real time, debugging becomes a real challenge. In this section, I will describe the PRESTK simulation environment, which is based on the simulation environment I implemented for the development phase of the sim^{TD} presentation manager.

Separation between Simulation and Application

The aim of the simulation is to facilitate and streamline the development and testing of the PRESTK toolkit. At the same time, the simulation should be cleanly separated from the system and not interfere with its inner workings. PRESTK is used but not modified (cf. figure 13.1). The simulation environment uses the prestk.jar implementation as an unmodified library.
The simulation has three tasks to perform:
(1) Simulate events and create/update/cancel presentations. This simulation can be performed manually or scripted. The aim is to get as close as possible to the mechanisms occurring in real usage.
(2) Monitor the processes inside PRESTK . Monitoring involves inspection of the presentation task database or visualizing the logging on a console window.
(3) Visualize the output of the presentation toolkit. Ideally, the GUI used inside the car is displayed.
As a result of these requirements, we obtain a GUI containing a large set of simulation, monitoring, and visualization components. As an interaction paradigm for the user, I chose the concept of a full-screen application frame containing separately selectable inner frames for each component (cf. figure 13.2).
In the next section, we will take a look at the single components of the simulation environment.

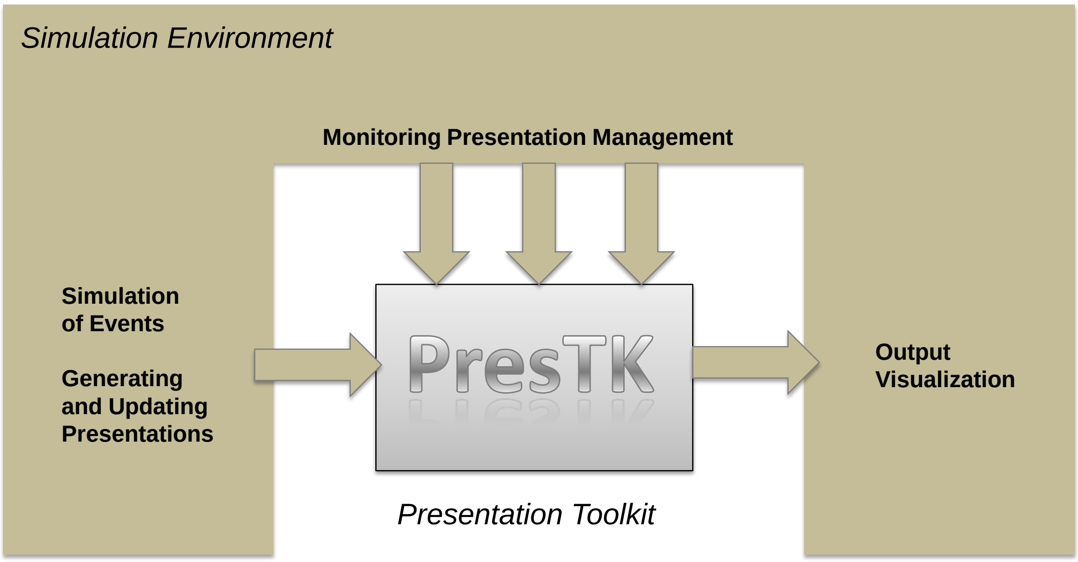


Figure 13.1: Clean separation of code base between simulation environment and PresTK system.

Components

A selection of the important components in the simulation environment is presented here.

Timer

The simulation environment has a class clock implemented as a public singleton, equipped with a listener interface. By that, we ensure that all components in the simulation listen to the same time. The advantage of this is that it can be easily replaced with a slowed down or sped up version without changing anything else in the code.

Console

The PRESTK system implements a logging interface. By providing an implementation of this interface to PRESTK, we can hook into the logging and visualize it in real time. The logging console is also realized as an internal frame, with scroll bars and a configurable scroll size. If necessary, logging can be not only shown but also stored in a logfile for persistance.

Presentation Task List Editor

This editor provides the option of generating a list of tasks by selecting a task type from a menu, and specifying its local priority, start time, and duration. The task can

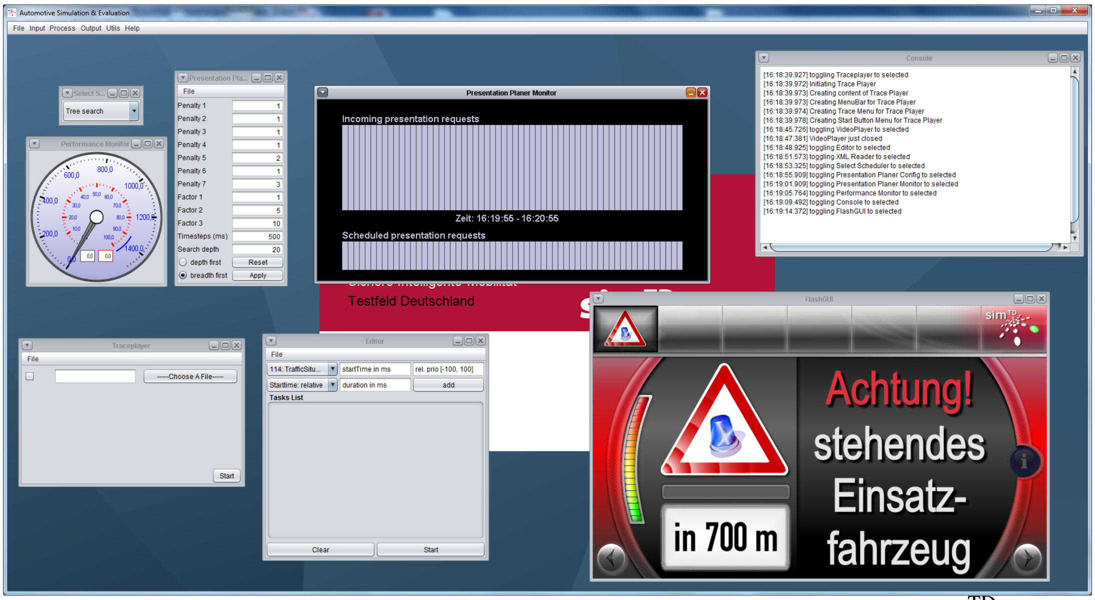


Figure 13.2: Screenshot of simulation environment, originally implemented for sim^{TD} system.

then be added to a task list. Task lists can be stored in a file and reloaded if necessary. In this way, test scripts can be generated and re-run under reproducible conditions.

Parametrization

The tree-search scheduler uses parameters based on an interim understanding of optimal configuration. These parameters include, among other things, the penalties for different actions. For debugging, an option to adjust these parameters is useful. A special configuration window is available for that.

Presentation Task Database Monitor

The task monitor is a time-line based visualization window split into two parts. On the upper part, the incoming presentation tasks with potential conflicts are shown on a timeline. On the lower part, the orchestrated conflict-free presentations are shown. The working of the scheduling process can be observed in real time.

Flash GUI Player

As mentioned previously, the best possible visualization is the user interface actually used with the system. For the sim^{TD} system, we included the Flash GUI used in the car. For displaying it inside a Java JInternalFrame, the free JFlashPlayer API[1] is used.

[1]Unfortunately the JFlashPlayer API is not supported anymore.

After looking at the simulation environment, we will discuss the PRESTK system implementation in the next section.

13.3 Workflow and Architecture

Workflow

From a workflow point of view (cf. figure 13.3), PRESTK is a layer or middleware between the user and applications attempting to present information. We assume a context in which different, sometimes mutually independent applications are generating information to be presented. This information comes in the form of presentation tasks. The minimal information contained in a presentation task is start time, end time (or duration), a numerical priority value, information to be presented and one or more presentation strategies. These presentation tasks are stored for further processing in the presentation task database. Due to the mutual independence of the applications, the presentation tasks might be in conflict. We consider a conflict the attempt of two or more applications to simultaneously access a limited resource. Several components can access and modify the content of the presentation task database:

- The presentation complexity estimation component analyzes incoming presentation tasks and for each presentation strategy involved calculates a numerical value describing the cognitive complexity resulting from presenting the given information in the specified way.
- The conflict detection component checks whether or not incoming presentation tasks cause conflicts with existing presentation tasks and triggers rescheduling, if necessary.
- The dynamic replanning assistant checks whether or not rescheduling is necessary for reasons not covered by the conflict detection component.

A scheduler ensures that conflictinging presentation tasks are modified in a way that the conflicts are resolved. After this resolution of technical limitations, the cognitive bottleneck of the user is addressed by the context and situation module, and the filtered (and possibly modified) presentation tasks are presented to the user.

Architecture

The architectural perspective (cf. figure 13.4) differs slightly. We still have the application layer sending presentation tasks to the presentation task database. The presentation complexity estimation component ACE (cf. section 9.6) picks incoming tasks from the database, analyzes, and modifies them. The dynamic conflict detection component checks for conflicts in the database and triggers rescheduling. Scheduling can

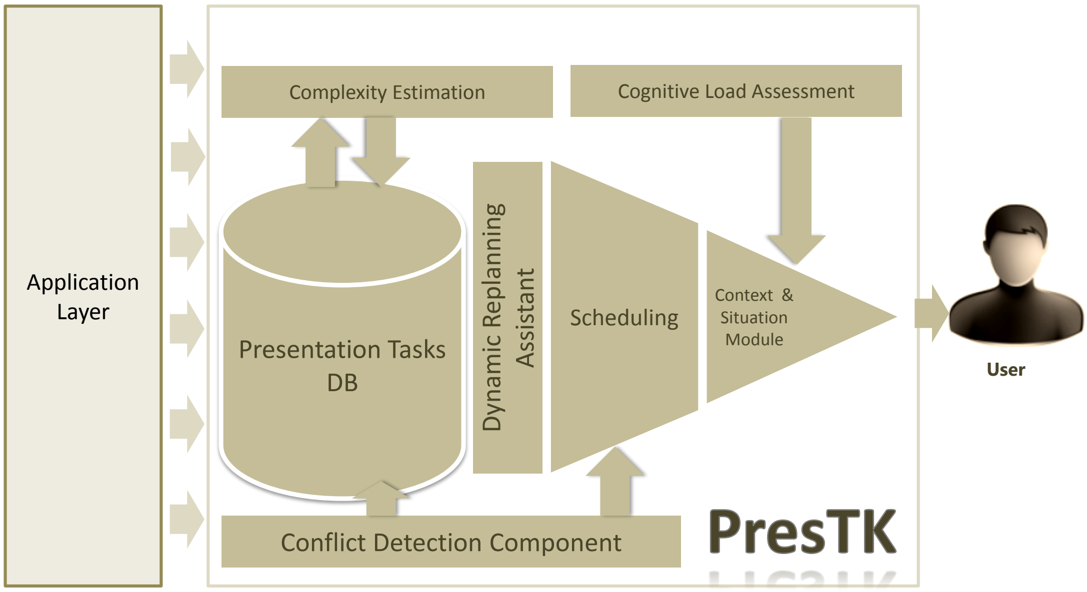


Figure 13.3: Workflow Perspective on PRESTK .

be performed by a selected manager; several choices are available. By default, tree-search is used. The manager takes presentation complexity and the user context into consideration. The user context is assessed by looking up the user's current assumed cognitive load from the user modeling database KAPcom. At the same time, PRESTK feeds back information about assumed cognitive load induced to the user through the presented information and the way it is presented. Finally, conflict-free information is presented to the user in accord with his current cogntive state.

13.4 Elements of the Toolkit

The architecture of PRESTK is based on connected components (cf. figure 13.5), which can be configured and adapted to the application requirements. At the current state of development, adaptation of the toolkit has to be done "manually", i.e., adapting the code is necessary for most but not all changes. Parametrization of the scheduler for instance can be achieved using the simulation environment.
Not all components are mandatory; for some we have a choice of different technologies. The basic functionality of all components is described in this section.

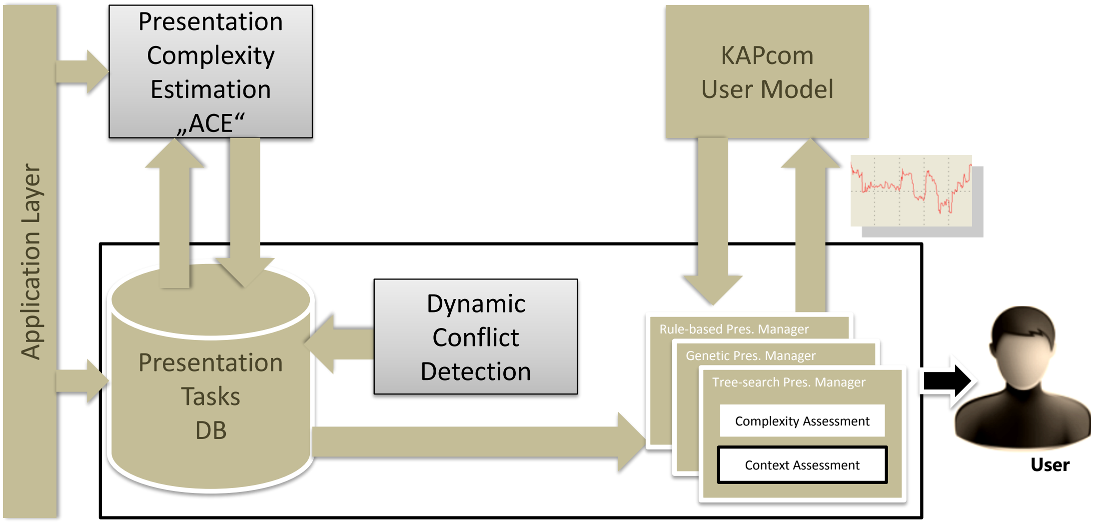


Figure 13.4: Architectural Perspective on PRESTK .

Core components

Presentation Task Database

The presentation task database is a non-optional core component of PRESTK . Incoming presentation tasks are stored here, and interfaces are provided for components working on the database. Access can be triggered by events such as arrival or modification of a presentation task, or performed asynchronously, e.g., triggered by a timer.

Dynamic Conflict Detection Component

A conflict occurs when two or more applications simultaneously attempt to access a limited resource. This can happen the moment a presentation task is created or updated. It can also occur when a presentation task is modified by another component, e.g., by the manager itself. A change in timing (start time, end time, duration) or display strategy might introduce a new conflict. A separate component is used to check, when necessary, whether or not the presentation task database is conflict free.

Selection of Presentation Managers

The advantage of a toolkit is that to a certain extent component redundancy is possible. Especially for presentation management, a choice of different techniques is available. We tested and evaluated several, as described here.

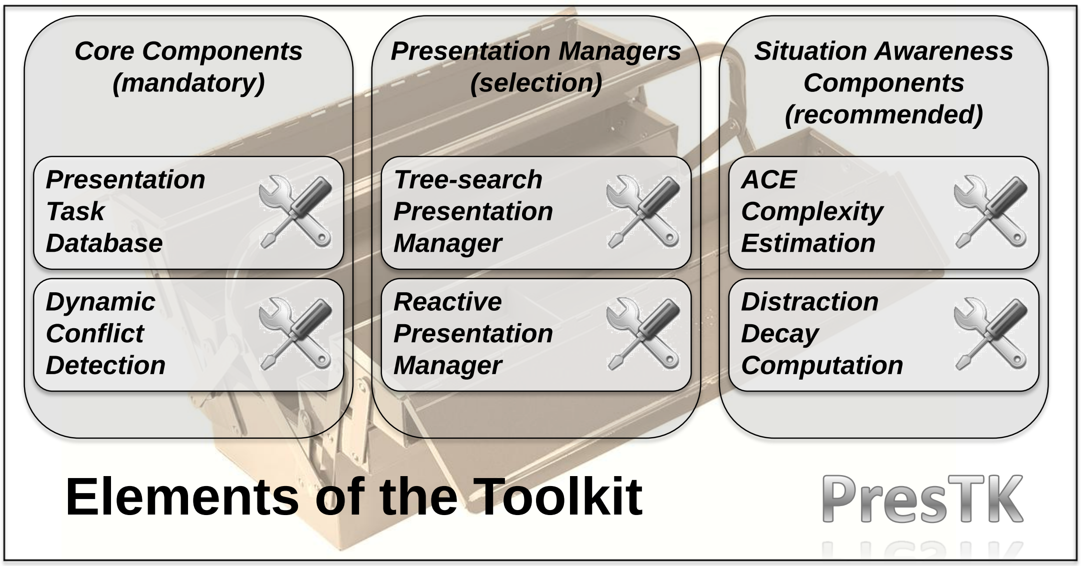


Figure 13.5: Elements of PresTK

Tree-search Presentation Manager: Planning ahead

In the sim^{TD} project, a tree search scheduler was developed and evaluated in a real life application, i.e., on the road. Based on the requirements of the project, this scheduler has a focus on a single screen setup. This scheduler was used as the foundation for the first PRESTK presentation manager. It is, along with the modifications described in section 9.8, included in the PRESTK implementation.

Reactive Tree-search Presentation Manager

Based on our experience with automotive OEMs, we learned that "in real life" the requirements for a scheduler can differ quite significantly from the theoretical approach. One main difference is the timing aspect. A lot of information is available only at the last moment, i.e., right at presentation time, which makes look-ahead scheduling impossible. With that in mind, I modified the tree scheduler such that it can deal with reactive decisions. The modifications are described in section 11.1.

Additional presentation managers such as a genetic version and a rule-based version [Mau11] were implemented and tested, but not included in the toolkit as they offered no advantage over the tree-search approach.

Additional Components

ACE Complexity Estimation

When a new presentation task is entered in the presentation task database, the presentation complexity estimation component is informed by an event listener. If the presentation task does not contain complexity estimations for all its presentation strategies, an analysis is started using the ACE complexity estimation procedure and the resulting complexity is added to the presentation task description, for further processing by the presentation management. The process of estimating a presentation complexity is described in more detail in 9.8.

Distraction Decay Calculation

The experiment described in section 10.2 no detailed information about the eye movement of the driver was used, and basically all visual distractions in his viewfield were considered. Using an eye-tracker, we are able to obtain more information about the glance time towards a distraction and can use a more detailed model of short-term distractions and their decaying effect on the driver. Decay functions are discussed in section 9.3. The current protoype of the decay function implementation uses parametrizable linear decay and does not yet consider the glance time but only the moment of glance and the amount of distractions in the viewfield.

13.5 Toolkit Adaptivity

The question arising immedeately from the word toolkit is how and to which extend it can be adapted to different applications. If we stay with the mobility context, other vehicles come to mind and should be discussed here.
Aircraft. The situation in an aircraft differs from a driving situation. While start and landing are very intense situations for the pilot, the main part of the flight is usually rather uneventful. Traffic is less dense, there are less distractions, more automation is available, there are two or more pilots on redundant controls, the training is by far more extensive than for an ordinary drivers license, and constant training is required. Due to higher travelling speed, changes in direction are preformed over a longer distance. For military aircraft, there might be additional tasks during the flight.
Ship. The travelling speed of a ship is comperatively slow, and due to the inertia caused by the water, steering is more complicated and more time-consuming than in a car. The effects of slow steering maneuvers have to be assessed carefully and planned with a longer time-to-effect. As with aircrafts, traffic is usually sparse (unless if in close proximity to a harbour) and many operations can be automated.
Motorcycle. The situation of a motorcycle is mostly comparable with a car, at least in terms of traffic density and driver training. The average travelling speed tends to be a

bit higher, and the driver belongs to the group of vulnerable road users with increased demand to cognitive ability and reaction time.
Motorsports. The situation in motorsports is similar to the motorcycle context, as far as speed and reaction time is concerned. Training of drivers however is here also very meticulous and there are usually no “seasonal drivers”. The demands in car sports and motorcycle sports differ again in terms os of cognitive demand.
Agricultural machinery and **heavy construction equipment.** When operating heavy equipment, the aim is usually not at driving as a means of transportation, but at the additional task to be performed. Depending on the point of view, one can argue that the driver has an additional primary task, or, taking that thought a bit further, that driving becomes a secondary task while operating and supervising the machine becomes the primary task, which may even add constraints to driving and route planning, e.g., when the filling level of a combine harvester forces the operator to be a a certain position for emptying. Additional demands such as hillside leveling could be used as an argument against considering driving a secondary task here.
Although all these different settings have different features and demands, in the end it is all just a matter of parametrizing and configuring: Operations need to be triggered earlier or reactions need to be faster, different information have to be displayed, less distraction is required, etc.
Using PRESTK in these different settings with slight modifications is feasible[2].

13.6 Connection to Other Systems

One of the aims when designing the PRESTK system was to keep the overall picture in mind and make it blend in smoothly with other research work at the same group as described in this section.

PADE

C2X PADE (**C**ar-**2**-**X** **P**latform for **A**pplication **De**velopment and **E**valuation) consists of several components simulating and executing the process of Car-2-Car and Car-2-X communication end-to-end, from low level communication protocols up to high level in-car presentations.
The software platform C2X PADE is the integration of research results of [Cas13] in order to systematically examine the applications and their mutual interdependencies both in a laboratory setting and on the road. C2X PADE supports the development of so-called Car-2-X pull services while considering human factors, and at the same time introduces work flow processes into the application development. C2X PADE consists of four main modules:

[2] The aim of the research presented here is to provide safety in traffic and to potentially save lives. The use of these concepts in a military context is neither intended nor supported by the author.

- *Communication and Simulation Module:* for the simulation of applications in a controlled laboratory setting.
- *Presentation Module:* for the optimal use of scarce output channel resources. One option of the system is to use PRESTK here.
- *Interaction Module:* for optimal and intuitive interaction with applications in every driving context.
- *Evaluation Module:* for evaluating the application before going on the road.

Overall, C2X PADE enables an easy transfer of applications from the laboratory to the road.

KAPcom

KAPcom (**K**nowledge management, **A**daptation and **P**ersonalization **com**ponent) is a central database for user modeling, designed in the automotive context. PRESTK connects to this database to look up and refine information about the current state of the user.
KAPcom and the necessary extensions for PRESTK are described in section 9.4. It is mainly used as a central storage facility for user related information.

SiAM

The dialog manager of the SiAM project (cf. figure 13.6) has a different view on presentation management than the sim^{TD} system: The case of multiple functions assessing a scarce output component is rare; the usual mode of operation consists of turn taking in input processing and output generation. Scheduling plays a less important role, but the presentation planning itself is of core interest. Presentation planning in SiAM is twofold: A preprocessor transforms abstract presentation descriptions into a set of meaningful alternatives, and the PRESTK system selects the most appropriate one.

13.7 Design features

Implementing algorithms which have been specified already in detail is mainly a straightforward engineering task. This does not imply that it is trivial, especially since the implementation has to be reliable and stable, and furthermore because the author takes special pride in writing high quality code. Nevertheless, transferring the procedures and algorithms described in chapter 8, section 9.8, section 9.6, and chapter 11 into code does not hold a lot of surprises for an experienced software engineer. In

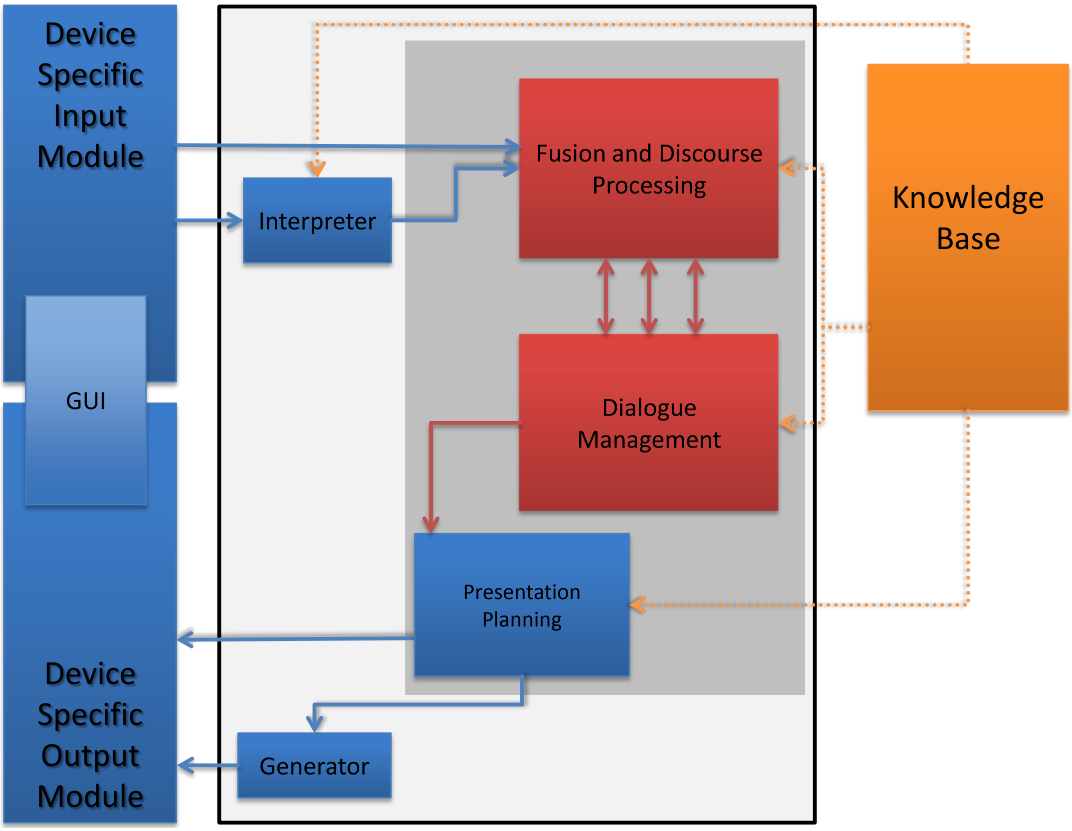


Figure 13.6: A simplified version of the SiAM architecture by M. Feld.

this section, I will detail only a few implementational features which I consider worth mentioning.

Minimizing memory requirements

Tree-search algorithms are very useful for solving problems with a large search space, but also come with some disadvantages: (1) extensive use of recursion is necessary which requires careful implementation, (2) trees have the inherent property of growing exponetially, (3) efficient pruning and effective termination conditions are necessary. Overall, special care has to be taken in memory management, even more so in an environment such as a car where computing resources usually are more scarce than on a high-end consumer PC.

In implementing the sim^{TD} system (and later on with PRESTK as well), I used the "small footprint"-paradigm: In order to minmize memory usage, the rather large "PrestentationTask" objects are transferred into "TaskDummy" objects containing only the most basic information about the presentation task and a reference back to the original object. This offers two advantages: On one hand, memory usage is kept to a minimum, which is important as the objects have to be cloned extensively while con-

structing the tree. On the other hand, replicating the objects can be preformed faster, as less information has to be copied. Once a solution is found, the dummy-object calls its referenced presentation task and writes back all information necessary for the solution.
Efficient pruning still plays a vital role; it is described in chapter 8 and evaluated in section 10.1.

Resource Allocation

The tree-search algorithm is a recursive implementation which by design is a long series of short atomic calculations. Also, if not interrupted, calculation may take a while. If this algorithm is implemented "greedy", i.e., taking up all the available resources, no other system thread would be able to function properly. By adding short calls to the Java sleep()-method, we offer a chance to other processes to get a share of the processing resources. By parametrizing the sleep-calls, it is possible to manage the amount of CPU resources used by the tree-search.

Duplicating the environment

The original sim^{TD} system runs on an OSGi platform, which is a very suitable environment for a system with multiple input sources that can dynamically sign on or sign off without interrupting the overall performance. In a testing environment however, the situation is a bit leaner as the incoming information is not actually triggered by complex mechanisms but instead manufactured or scripted based on empirical data. Setting up a complete OSGi environment for the simulation would be over-dimensioned. In designing the sim^{TD} HMI bundle, I introduced an abstract *HMIController* class, which can be instantiated with either the *OSGIController* object or a *GenericController* object. For the rest of the system, this decision has no effects on implementation, and the controller can be used as a "black box" without further consideration of the implementational details. Special features of the OSGi environment, such as logging, thread management, system listener, etc. are implemented with the same API, but depending on the selected environment either passed to OSGi or the generic implementation. By using this coding tweak, the simulation runs under the same conditions as the real system, and the produced bundle for OSGi is simply used as a library for the simulation environment.

Testing

The implementation is tested using JUnit tests. JUnit is a unit testing framework for test-driven development in the Java programming language. It is linked as a .jar file at runtime. In test-driven development, a testing method and the expected result is

specified. The tests can run autonomously, e.g., before an automated nightly build of a bundle, and a report is generated. If the result of the test method differs from the specified value, a warning is issued.
For developing the HMI bundle, the JUnit tests were used twofold: On one hand, the correct behaviour of the code can be tested automatically. On a more detailed level, the tester can visually confirm that the behaviour of the GUI matches the specified design.

13.8 Code Metrics

The amount of classes in object-oriented code as well as the lines of code are a metric for the size of a programming project. For the latter, we distinguish between lines of code (LOC), source lines of code (SLOC), and logical lines of code (LLOC). For ease of counting, the LOC measure is the most convenient. The measure is not undisputed, since it makes no statement about the efficiency of coding or the quality of the code, as large numbers can indicate both a large system as well as inefficient coding. However, lacking a better metric, it is customary to mention these numbers in a thesis. I use a (classes/LOC) annotation here.
The sim^{TD} HMIBundle (148/18085) provided the foundation for the PRESTK implementation.[3] It contains the core components and the tree-search presentation management. The simulationm environment (55/4810) uses this bundle as a .jar library. The extensions for situation awareness and its demonstrators described in section 12.3 add another (37/2117). The underlying OpenDS and EyeBox systems are under constant development and rapidly changing in size.

13.9 Summary

In this chapter, I have described the PRESTK implementation which is based on the theoretical contributions of this thesis. It is a small, prototypical implementation with room for further extension or added convenience such as adaptation based on configuration instead of code changes. Some parts, such as the tree-search presentation manager implemented for sim^{TD} have been tested extensively, while other parts are still in a more experimental state.

[3] I emphasize that this bundle was programmed by several software developers, most notably by my colleague Sandro Castronovo.

Part V

Conclusion

Chapter 14

Conclusion and Outlook

Short Summary

The advance of technology has gradually transformed the car into a sophisticated information hub. Additional information is helpful for the driver to make informed decisions. On the downside, too much information leads to a cognitive overload of the driver. Filtering and processing information is necessary in order to achieve meaningful assistance instead of an information avalanche.
In this thesis, I introduced my ideas for a situation-aware presentation toolkit in an in-car environment. In order to achieve system situation awareness, three research areas have to be combined: scheduling, presentation planning, and situation awareness theory.
A presentation manager combining scheduling and presentation planning has been introduced and evaluated in the context of the sim^{TD} system. As a next step, I introduced my concept of system situation awareness (SSA), based on Endsley's model. Achieving SSA is a problem that needs to be approached from two sides:
(1) The current cognitive load of the driver needs to be assessed. I argue that most approaches for assessing cognitive load are either of limited use for everyday driving or not sufficiently accurate (cf. [SP11] for instance). As an alternative, I propose the estimation of the driver's cognitive load using contextual parameters. As an example I examine the parameters of speed and visual distraction to verify the existence of an impact on the cognitive load of the driver.
(2) Information to be presented needs to be annotated with a complexity value. For cognitively demanding situations, a detailed model of the impact of a given piece of information on the cognitive state of the driver helps to choose between presentation alternatives.
Having provided these two foundations for system situation awareness, I explain how to extend the sim^{TD} presentation manager accordingly.
Finally, I describe the implementation of the extensible presentation toolkit PRESTK.

Key Research Questions

In the first chapter of this thesis, I introduced my research questions. At this point, I would like to get back to that point and shortly summarize the answers given in this thesis.

(1) What is the difference between cars and conventional ubiquitous intelligent environments and how can the specific challenges with respect to presentation planning arising from those differences be taken into account?

The car is in my opinion an intelligent environment and should be considered as such [EC10, EMM11]. At the same time, it is a moving environment with dynamically changing context which has to be taken into consideration. Presentation planning becomes time-critical and subject to dynamic changes. Information with high priority can turn up unexpectedly at any time. The car is, unlike traditional intelligent environments, not a closed system. Communication with other cars and even remote other environments such as the intelligent home, the intelligent office, and the social context is fostered by the advent of Car-2-Car and Car-2-X technology with increasing bandwidth and increasing possibilities. When implementing information management in a car, constant replanning has to be possible.
Furthermore, the management mechanism has to behave according to the paradigm of anytime algorithms, being interruptable and provide a usefull result even before terminating its calculation. In this thesis, I introduced a tree-search approach for managing potentially conflicting presentation requests based on the anytime paradigm (cf. chapter 8). The quality of the solution has been positively verified in a data-driven evaluation (cf. section 10.1).

(2) To what extent are traditional presentation planning mechanisms applicable to the domain and which changes have to be considered?

When driving a car, we usually distinguish between three prioritized types of tasks for the driver. The primary task is driving itself, the secondary tasks covers activities related to the driving context, and the teritiary task activities of the driver not directly related to driving (cf. section 9.2.2). The driving task has the highest relevance for the driver, in which his performance is extremely safety-critical. Decisions have to be achieved fast, and sometimes crucial information comes in literally at the last moment. We distinguish here between planning ahead and reactive planning. While planning ahead is the more desirable approach, in real life reactive planning plays an important role as well. The planning mechnisms presented in chapter 8 are taken up again in section 11.1, where reactive planning for emergency cases is discussed.
The dynamics of the driving context do not always leave room for meticulous prepa-

ration and fusion of presentation content at runtime. Presentation planning is usually achieved here by providing alternative presentation strategies beforehand and selecting the most appropriate presentation type at runtime.

(3) How can the cognitive complexity of a presentation be estimated and used for presentation planning with context assessment?

According to the Yerkes-Dodson law, performance of a task is best at a medium level of arousal and decreases in both directions of deviation. For in-car presentation management, it is advisable to keep the driver always at the peak of his driving performance. In order to do so, the impact of presented information to its cognitive state must be predicted. In this thesis, I present ACE, a layout-based estimation of presentation complexity (cf. section 9.6), which builds upon previous work, especially [IWB93], and combines it to an up-to-date approach for estimating the complexity of information on a display with a complex layout.
Empirical evaluation of the ACE procedure (cf. section 10.3) shows that the calculated results match the users' opinion to a very high degree. Further finetuning of the parameters in the algorithm by genetic programming improved the results even more. ACE provides a solid base for further refinement, inclusion of interaction items, and formalizing of the single elements of the GUI according to results known from literature.
Using this approach, the complexity of a GUI can be evaluated automatically, which minimizes the amount of required user studies.

(4) How can an explicit priority management be developed that takes into account both the driving situation and the cognitive limitations of the driver?

For the sim^{TD} system, I implemented an information management system which takes information priority and available resources in terms of output channels into account (cf. chapter 8). In this thesis, I propose to extend this approach with consideration of the cognitive load of the driver and the complexity of the information to be presented. The tree-search algorithm used for the sim^{TD} system is sufficiently flexible to be extended for it (cf. section 9.8). This leaves us with the problem of assessing the complexity of information to be presented and to constantly monitor the current cognitive load of the driver. For the former, I presented the Annotated Complexity Estimation (ACE) procedure. For the latter, I propose to deviate from traditional methods in literature: Most of them are unsuitable for everyday driving due to their obtrusiveness[1]. My approach is to build a model based on environmental parameters such as visual complexity, current speed, strain produced by the complexity of previously presented

[1]There are exceptions such as the use of steering wheel angle, which is completely unintrusive, but unfortunately also not very reliable according to [SP11]

messages, etc.
This approach is presented in section 9.3, evaluated in section 10.2 and demonstrated in section 12.3.2.

(5) How can we define situation awareness on a system level?

In this thesis, I claim that the concept of situation awareness according to Endsley needs to be extended in automotive research: Not only the driver has to be aware of its surroundings, but also the car, e.g., the system. Although Endsley's definition (cf. section 5.2) does not explicitly preclude this interpretation of situation awareness, her model is usually used solely for the situation awareness of the driver (or user, in a broader sense). I define the concept of *system situation awareness* (cf. section 9.1) in close analogy to Endsley's model. In the core part, the three levels of system situation awareness are: (1) assessment of user and context, (2) updating information sources, and (3) impact estimation.
How to achieve these three levels is discussed and described in detail in chapter 9: In a twofold approach, we assess the driver's current cognitive load on one hand (cf. research question 4) and the cognitive complexity of information to be presented on the other hand (cf. research question 3). Using these two pieces of information, the planning mechanism introduced in chapter 8 is refined in chapter 9 to a situation aware presentation system.

(6) Which are the core components of a flexibly applicable system for situation-aware information presentation and how can they be combined into a comprehensive toolkit?

In this thesis, several concepts were introduced that can be represented in code and assembled towards achieving the overall goal of system situation awareness. In part IV, the implementation of the presentation toolkit PRESTK is described. A choice of two presentation managers is included, and two experimental versions shortly discussed. As underlying tools, the mandatory modules "dynamic conflict detection" and the "presentation task database" are included already in the first version implemented for the sim^{TD} project.
For PRESTK, additional modules for situation awareness have been implemented: The ACE complexity estimation component can assess the cognitive demand of information being presented or to be presented. Furthermore, the "distraction decay component" calculates the impact of environmental factors on the cognitive load of the driver and its change over time.
The combination of these techniques already discussed in the previous research questions has been demonstrated successfully in section 12.3. Implementational details can be found in chapter 13.

Collaborations and Interdisciplinary Research

The literature discussed in this thesis is not limited to computer science and artificial intelligence papers only. Publications from other disciplines, especially psychology, but also medicine and training science are included. The origins of this work can be traced back to military aviation, but it has been adapted to automotive research gradually over the past two decades.
Another indicator for the interdisciplinary nature of this thesis is the choice of the examiners, Wolfgang Wahlster as expert for Artificial Intelligence, Stefan Panzer as expert for sports and training sciences, and Albrecht Schmidt as a second computer scientist and expert for Interactive Systems and Human Computer Interaction (HCI).
In collaboration with the institute of training science, the chair of construction sciences, and car racing talent Chiara Messina, a new line of research aimed at objective evaluation for young race drivers has been triggered.
With the World Wide Web Consortium (W3C), and most notably HTML architect Dave Raggett, ideas for an automotive presentation language have been discussed and requirements defined [EFM12b].
The first prototype of this thesis stems from the sim^{TD} project, a joint effort of all major players in the German automotive industry. The results of this thesis, especially the PRESTK system and the PTCL language, will become a foundation of the SiAM system to be developed at DFKI.

Main contributions of this thesis

The main contributions of my thesis are summarized in table 14.1. For the three main areas, as well as for underlying research, the following information is given:
- Theoretical concepts discussed,
- Own contribution to the theory in that field,
- Practical implementation,
- Own publications on the subject,
- Evaluation,
- Demonstration, and
- Connection to the research questions.

In the theoretical part (Part II), a survey of the literature is given and important concepts are discussed and described. My contribution to the theory is described in part III, mainly in chapters 8 and 9. The implementation is described in part IV. As the most important own publications in connection to this thesis, the papers [EFM12b], [EMB12], [EFM12a], [End12], [EM12], [EC10], [FE10], [EBM05], [KKE03], [End03], and [WKE03] are listed. My other publication on new in-car interaction modalities [EB10, ED10, ESM11, EBM11, MESM11], and automotive research in general [EFSM10, FE10, CME10, CE10, ES10, SCE10, EMB10, CEF^{+}11]

	Area	Topic	Subtopic	Theory	Theoretical contribution	Implementation	Own Publications	Evaluation	Demonstration	RQ
0. Basics	Intell. Env.	general	description & definition	Survey	"Car as IE"	(implicit)	[EC10] [EBM05] [KKE03] [End03] [WKE03]	N.A.	Foundation / Part of all demonstrations (ZWM, in-car)	1
		history	previous work	Survey						
	Automotive Domain	general	properties	Description			[EC10]			
		HMI	description	Survey						
	Searching	Tree-Search	Algorithms	Survey	Extension / new Approach (see RCSP)		[EM12]	data driven; presented in peer-reviewed paper		2
			Pruning	Survey						
		other approaches	Algorithms	Survey						
I. Scheduling	Scheduling	JSP		Description						
		RCPSP	different Approaches	Survey						
		New problem description			RCSP	Part of PresTK				
	simTD	Implementation		Description	Algorithm	sim^{TD} Pres. Manager		Part of simTD	sim^{TD}	
II. Presentation Planning		Lessons learned		Description	Formalizing requirements	Extension of sim^{TD}	[End12]	empirical		3
	Presentation Planning	History	Systems	Survey	PresTK					
		Presentation Languages	Existing languages	Survey	Requirement analysis, work with W3C	Definition of PTCL language	[EFM12b]		to be demonstrated in SiAM	
			Suitability for Autom.	Discussion						
III. Situation awareness	Cognitive Load	Cognitive Load Assessment	subjective measures	Survey	Discussion of suitability for Automotive Domain	KAPcom estension	[FE10] [BEM11] [EFM12a] [End12]	Experiment (Chapter 10.2)	N.A.	4 5
			performance based	Survey					Demo shows level of cognitive load estimated (ZWM)	
			physiological measures	Survey						
		Cognitive Load Estimation	by context	Survey	Combination and Extension of approaches in literature	EyeBox Data --> visual Objects				
		Consideration	Possibilities	Survey	New heuristics	Part of PresTK	[End12]	Empirical	System reacts according to current level of cognitive load (in-car)	
	Complexity Estimation	offline / before		Discussion	Annotation	Manual Annotation		Experiment (Chapter 10.3)		
		online / in real time	structured data	Discussion	heuristic	ACE procedure				
			unstructured data	Discussion	heuristic					

Table 14.1: Summary of the main contributions by this thesis

influenced this thesis as well. Evaluation was presented in chapter 10. Demonstrations of the concepts developed in this thesis are described in section 12.3. The last column assigns the research questions asked at the beginning and summarized in this chapter to the individual parts.
Research question 6 is not mentioned explicitly as it is implicitly included in all rows.

Conclusions and Outlook

I started the introduction of this thesis with the events of Bridget Driscoll's dead and the wish of coroner Percy Morrison that "such a thing would never happen again". A lot of effort has been made towards that goal over the past 116 years, and a single thesis can of course only have a humble impact and will not solve all the problems of car accidents and driver distractions. At the end of this document, I would like to summarize what has been achieved and which further steps are necessary in my opinion.

The definition analysis of System Situation Awareness (cf. chapter 9) and the implementation of PRESTK (cf. chapter 13) respectively fill a gap that is currently noticeable in the body of automotive research literature. In spite of the numerous efforts to assess the cognitive state of the driver, a hollistic approach how to integrate that information into an in-car system is still missing. Furthermore, the efforts in automated estimation using the newly defined ACE procedure were very successfull and unprecedented in literature.

To my knowledge, this thesis presents the first approach of combining scheduling, presentation planning, and situation awareness in an in-car context. Chapter 12 summarizes how the formulated requirements were achieved. The presentation management was formulated in the three dimensions time, modality, and complexity. The solution provided is adapted to the specific nature of the driving context; its features are: extensible, dynamic, anytime, situation-aware, assessment and consideration of cognitive load and presentation complexity, and formalization of all concepts involved. The ideas for an in-car presentation language have been discussed with the automotive research section of the World Wide Web Consortium (W3C).

A data-driven evaluation (cf. section 10.1) supports the claim that tree-search is an appropriate mechanism for the presentation management problem in an automotive environment. Two empirical studies (cf. section 10.2 and 10.3) verify the methodology of cognitive load estimation from the driving context and the layout-based estimation of presentation complexity respectively. The overall functionality is demonstrated both in a driving simulator (cf. section 12.3.1) as well as on the road (cf. section 12.3.2).

What are the next steps?

First of all, the PRESTK approach was developped after the presentation management system for the sim^{TD} project, and the changes and additions made towards situation awareness were not subjected to a field test. The European FOT-Net (Field Operational Test) initiative[2] emphasizes the importance of field tests as an essential step towards deploying ITS solutions. They define Field Operational Tests as "large-scale testing programmes aiming at a comprehensive assessment of the efficiency, quality, robustness and acceptance of ICT solutions used for smarter, safer, cleaner, and more comfortable transport solutions, such as navigation and traffic information, and ad-

[2] fot.net.eu

vanced driver assistance."[3]. This definition covers both stand-alone systems in the car as well as Car-2-X or similar cooperative systems.
Designing and conducting a large-scale field test is of course out of both the scope as well as the budget of a thesis. Nevertheless, the requirements for such a test should be discussed here.
Unlike the sim^{TD} field test, a test for PRESTK would need a strong focus on information presentation. I propose a test scenario in two parts: First, real-life traffic situations would have to be recorded and logged. Interesting traffic situations, i.e., situations with non-trivial conflicts, need to be identified and the system behaviour analyzed. In order to determine the quality of the solution provided by the system, expert opinions have to be collected as a ground truth in a second phase. As an expert in this context I would consider somebody who is driving professionally, e.g., a taxi driver or somebody travelling a lot by car on business. These experts could analyze the collected data and provide the desired solution to be compared with the actual solution. However, technical constraints have to be considered: The ground truth given by experts has to be feasible, e.g., it can not be expected for the system to provide information before it has been transmitted or detected by the sensors.
The availability of a large body of such data could be used to fine tune the presentation management system.
Another aspect that should be considered in future work is the inclusion of new output modalities. Recent efforts towards reducing driver distraction tend towards the inclusion of other communication channels to the driver. [AB10] for instance examined the use of tactile information transmitted by vibrating devices in the driver's seat for providing navigation instructions. These and similar upcoming communication means have not been considered in my thesis.
The focus of PRESTK is on information output. In order to achieve interaction, input modalities, their complexity and their distraction potential need to be examined as well. By including the PRESTK system and especially also extending the ACE procedure in the dialog manager of the SiAM [MM12], an important step towards a situation-aware dialog manager will be taken.
Interfacing with the KAPcom system provides more benefits than have been used yet. Taking advantage of a given user model for the driver extends the possibilities of an in-car information system. As I emphasized before, the requirements for obtaining a driver's license do not require very hard or intense training, and in result the skills of drivers on the road vary noticeably. A personalized mode for each driver could be established in terms of system configuration parameters and stored in the user profile. Especially for people who just started to learn how to drive, the system could provide the digital equivalent of the learner sign ("L") used in the United States (and similar in other countries).
But even experienced drivers sometimes can not drive at the peak of their skill-level,

[3] Official FOT-Net brochure.

especially when changing to a new car model. It might be easier for some people who frequently use rental cars, but a driver used solely to his own car for several years and buying a new and more advanced car can easily get overwhelmed. Here, two approaches could be used to make things easier: First of all, a more advanced mode of assistance could be used, similar to the learner mode described above. Secondly, the car could "introduce itself", which brings us back to the beginning of presentation planning research and the project WIP described in section 3.2. I stated previously that the presentation planning aspect in PRESTK does not include full presentation planning as the early systems with content generation and combination, but rather selects predefined presentation startegies. This system could be overlayed with a full presentation planner with a long term goal, i.e., introducing the driver to the full functionality of his new car. A simple version of how this could be done was shown in the second demonstrator, using the concept of the *tourist information system*, which has a similar objective.

I hope that this thesis will foster further work and discussion and proves to be a suitable foundation for follow-up research.

Part VI

Appendix

Appendix A

Presentation Languages

3DMLW - 3D Markup Language for Web

Table A.1: Classification sheet for 3DMLW

3DMLW - 3D Markup Language for Web	
Year:	2008–2011
Domain:	Web (2D and 3D)
Target Platform:	3DMLW platform (Windows, Mac, Linux)
Origin:	3D Technologies R&D
Stage of Development:	stable release
License:	Gnu Public License (GPL)
Compatibility:	3rd party XML editors
Tools:	own scripting language (LUA)
Language structure:	XML 1.0 based
Community/Impact:	Sourceforge
Media support:	3D Models (.3ds, .obj, .an8, .blend)
Webpage:	www.3dmlw.com, wiki.3dmlw.org

The term *3DMLW* refers to both an open source platform and to the accompanying markup language. According to 3D Technologies R&D, the technology consists of five parts: The markup language, scripting support, style sheets, browser plug-ins, and model viewer for the supported media formats. The advantage of the technology is the combination of 2D and 3D content. The rendering engine uses OpenGL (Open Graphics Library). The solution is cross-platform and can be extended through plugins. 2D and 3D content is handled independently from each other, but they are free to overlap. For animating 3D scenes and handling different events a Lua scripting facility is provided. The developers stopped developing this technology, "as there are other commercial 3D rendering engines for browsers and probably WebGL will also remove the need for this kind of 3D browser plug-ins"[1]. Components of 3DMLW are still used in other projects of the company, such as 3D Wayfinder, a digital building directory that helps visitors find their way in large public buildings[2].
Listing A.1 shows an example of combining 2D and 3D content in 3DMLW.

Listing A.1: 3DWML example code

```
<?xml version='1.0' standalone='no'?>
<document>
        <content2d>
                <area width='200' height='100' color='#C0C0C0FF' texture='flower.png' />
        </content2d>
        <content3d id='content' camera='{#cam}'>
                <camera id='cam' class='cam_rotation' y='10' z='40' viewy='10'/>
                <box name='ground' width='100' height='2' depth='100' color='green' class='ground' />
                <box name='dynamic' y='20' width='10' height='10' depth='10' color='blue' />
        </content3d>
</document>
```

[1]Email from Kaspar Koov, July 4th, 2012
[2]www.3dwayfinder.com

Android Drawable XML

Table A.2: Classification sheet for Android Drawable

Android Drawable XML	
Year:	2007
Domain:	Mobile Apps
Target Platform:	Android Operating System
Origin:	Google Inc.
Stage of Development:	product
License:	open
Compatibility:	
Tools:	Eclipse, any text editor, etc.
Language structure:	XML 1.0 based
Community/Impact:	Google
Media support:	Graphics, audio, video
Webpage:	android.com

Android OS is a Linux-based popular operating system for smartphones, tablet computers, and other mobile devices. It was originally developed by Android Inc., a company which was bought by Google Inc. in 2005. Applications ("Apps") for Android OS are written in Java. Although it is possible to implement the user interface purely in Java code, the usual way for Android applications is to specify the user interface using a XML-based layout description, which is placed in the res/layout folder of the programming project and called in the onCreate() method of the respective Java class [Mur08].
Android Drawable is used in connection with Android SDK and compiled into the application. The layout is adapted automatically to the screen size of the target device. Listing A.2 shows a "Hello world!" example in Android Drawable XML.

Listing A.2: Android Drawable XML code example

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent"
                                                    android:layout_height="match_parent">
        <TextView android:text="Hello World!" android:id="@+id/textView1" android:layout_width="wrap_content"
                                                    android:layout_height="wrap_content">
        </TextView>
</LinearLayout>
```

AUIML - Abstract User Interface Markup Language

Table A.3: Classification sheet for AUIML

AUIML - Abstract User Interface Markup Language	
Year:	2004
Domain:	GUI and web development
Target Platform:	Browser, Stand-alone Java Swing application
Origin:	IBM alphaworks group
Stage of Development:	abandoned
License:	
Compatibility:	
Tools:	AUIML Toolkit, Eclipse plugins
Language structure:	XML based plus properties file
Community/Impact:	abandoned language
Media support:	Java GUI elements
Webpage:	www.alphaworks.ibm.com/tech/auiml

AUIML was developed by IBM alphaworks as an attempt to facilitate rapid GUI development. The idea was to code one abstract description of a User Interface using the AUIML Toolkit and then export it for different target platforms. The GUI could be coded by editing the AUIML code or by using a visual builder.
Initially, two renderers were included: the HTML and the Java Swing renderer. AUIML code could be transformed for the target platform without any changes in the code. The development environment was Eclipse with several plugins, such as the respective renderers and the Visual Builder plugin.
Several other renderers were planned but not realized. By now, the development is abandoned.
According to [B'f05], AUIML is one of the efforts which eventually lead to the development of XForms.

DisplayML

Table A.4: Classification sheet for DisplayML

DisplayML	
Year:	2005 (webpage registered)
Domain:	Display information encoding language
Target Platform:	
Origin:	Swarco Mizar[3]
Stage of Development:	Webpage dead.
License:	open, free-to-use
Compatibility:	
Tools:	
Language structure:	
Community/Impact:	
Media support:	
Webpage:	DisplayML.org (dead), alternatively: en.wikipedia.org/wiki/DisplayML

The information available on DisplayML unfortunately is somewhat sparse, and the project's webpage is not available anymore. It is safe to assume that development has been discontinued. According to the limited information available on Wikipedia[4], DisplayML is "open, free-to-use protocol for encoding display information for display devices. It is based upon XML messages with a Request/Response model where the display device works as the server". The protocol supports file transfer, so that fonts, images or software can be transmitted to a display.

[3] http://www.swarco.se/?menu=3668&id=756

[4] I am aware that wikipedia is not the preferred source for a scientific publication. However, lacking alternatives, I prefer to gather the limited information I can get over excluding this language from the survey.

EMMA - Extensible MultiModal Annotation markup language

Table A.5: Classification sheet for EMMA

EMMA - Extensible MultiModal Annotation markup language	
Year:	2005; W3C specification 2009
Domain:	Dialog management, multimodal interaction annotation
Target Platform:	
Origin:	Multimodal Interaction Working Group, W3C
Stage of Development:	stable (W3C recommendation)
License:	open
Compatibility:	converters available
Tools:	
Language structure:	XML 1.0 based
Community/Impact:	
Media support:	
Webpage:	www.w3.org/TR/emma/

EMMA is a markup primarily used as a data interchange format between components of a multimodal system. The content is usually machine generated, not authored. The purpose of inter-component data exchange distinguishes this language from the others discussed in this section. EMMA is focused on semantic annotation of user input with automatically extracted information. According to the specification[5], an EMMA document holds three types of data: instance data, data model and meta data.
Listing A.3 shows an example of EMMA code with three levels of input data interpretation, labeled raw, better and best.
EMMA was extended at DFKI in the Smartweb project to SWEMMA - SmartWeb EMMA (see page 215).

Listing A.3: EMMA code example

```
<emma:emma version="1.0" xmlns:emma:=http://www.w3.org/2003/04/emma
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.w3.org/2003/04/emma http://www.w3.org/TR/emma/emma10.xsd"
          xmlns="http://www.example.com/example">
        <emma:derivation>
                <emma:interpretation id="raw">
                        <answer>From Boston to Denver tomorrow</answer>
                </emma:interpretation>
                <emma:interpretation id="better">
                        <emma:derived-from resource="#raw" composite="false"/>
                        <origin>Boston</origin>
                        <destination>Denver</destination>
                        <date>tomorrow</date>
                </emma:interpretation>
        </emma:derivation>
        <emma:interpretation id="best">
                <emma:derived-from resource="#better" composite="false"/>
                <origin>Boston</origin>
                <destination>Denver</destination>
                <date>20030315</date>
        </emma:interpretation>
</emma:emma>
```

[5] www.w3.org/TR/emma/

Flash SWF - Adobe Flash

Table A.6: Classification sheet for Flash SWF

Flash SWF - Adobe Flash	
Year:	1996 (FutureSplash)
Domain:	Interactive Web-Applications
Target Platform:	Stand-alone applications (Windows, Mac, Linux), Browser (support for mobile browsers ends 2012), mobile devices
Origin:	FutureWave software, macromedia, Adobe
Stage of Development:	product
License:	specification partially available under restrictive license
Compatibility:	none
Tools:	Adobe Flash, Flashbuilder, Flashdevelop
Language structure:	compiled format, using scripting language (actionscript), edited visually on an IDE
Community/Impact:	OpenSource community, tools both commercial and open source
Media support:	vector graphics, 3D graphics (with hardware accelaration), audio, video
Webpage:	http://www.adobe.com/de/products/flex.html

SWF is an Adobe Flash file format used for both multimedia and interaction scripting. Originally, it was limited to showing sequences of images and vector graphics. Audio was introduced in version 3, video in version 6, and it now supports a broad range of multimedia formats. SWF files can be played in a player (either inside a webbrowser or a player application) or be encapsulated with the player as a so-called "projector". Several players under GNU GPL are available, such as Gnash (GPL) or SWFdec (LGPL). SWF enables rapid prototyping and generation of interactive animations by providing an easy collaboration between designers and programmers. Code and graphics can be edited separately, using a visual editor (see Figure A.1). Using Adobe Air Runtime (AIR), swf enables the creation of sophisticated desktop applications. Multitouch and gestures are supported.

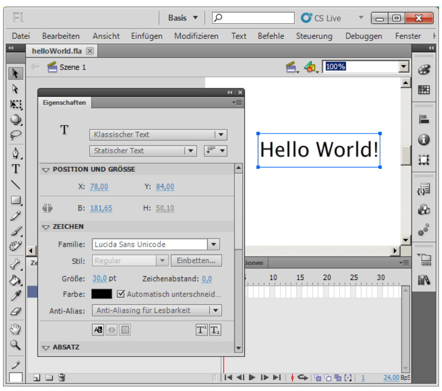


Figure A.1: Hello World in Flash SWF

FXML (ActionScript)

Table A.7: Classification sheet for FXML (Actionscript)

FXML (ActionScript) - FlashXML	
Year:	2009
Domain:	Interactive web-applications; possibly used for rich Internet applications
Target Platform:	Flash
Origin:	Jordan Doczy (initiator)
Stage of Development:	release
License:	open source
Compatibility:	works only with ActionScript 3 and a special .swf to be incorporated within the GUI declaration
Tools:	Adobe Flash / Flashbuilder / Flashdevelop
Language structure:	XML-based
Community/Impact:	Open source community
Media support:	possible due to Actionscript 3 and extensions
Webpage:	www.fxml.org/

As mentioned previously (see page 203), it is difficult to change a .swf file once it is compiled. The idea of FXML is to introduce a scripting language which changes given parameters of an animation at runtime. The elements of the selected user interface can be accessed directly on a tree structure (similar to DOM traversing). Unlike the older FXML (JavaFX), it is a pure command language without any graphical elements. Listing A.4 shows how to set the String “Hello World” in a given Textbox on a Flash GUI.

Listing A.4: Hello World in FXML

```
<data>
	<addChild>
		<child class="flash.text.TextField">
			<text>Hello World!</text>
		</child>
	</addChild>
</data>
```

FXML (JavaFX)

Table A.8: Classification sheet for FXML (JavaFX)

FXML (JavaFX)	
Year:	2007 (first announcement)
Domain:	Java FX application / Desktop
Target Platform:	Stand-alone applications (Windows, Mac, Linux)
Origin:	Sun Microsystems, Oracle
Stage of Development:	stable release
License:	open source
Compatibility:	none
Tools:	Java editor
Language structure:	XML 1.0 based
Community/Impact:	
Media support:	graphics, audio, video
Webpage:	www.oracle.com/technetwork/java/javafx/overview/

JavaFX is a software platform for creating cross-platform Rich Internet Applications (RIA) intended to run on a variety of devices. It currently supports applications for desktop, browser and mobile devices. JavaFX uses Java FX Script as a scripting language and FXML for modeling the user interface. This separation between design and logic is convenient for web developers, since it enables easy collaboration between graphic designers and software developers. FXML does not have a schema, but a basic predefined structure. The expressiveness of FXML and the construction of the scene graph depends on the design and the API of the underlying Java objects. Most JavaFX classes can be used as elements. FXML is not a compiled language; changes in the code are immedeately visible, which supports rapid prototyping. Unlike FXML (ActionScript) it is really a GUI description language. The underlying philosophy of FXML can later be found in Android Drawable XML (see page 199). Listing A.6 shows a “Hello World!” example in FXML, referring to the Java objects in the codesnippet of Listing A.5.

Listing A.5: Java code snippet referenced in Listing A.6

```
BorderPane border = new BorderPane();
Label toppanetext = new Label("Page Title");
border.setTop(toppanetext);
Label centerpanetext = new Label ("Some data here");
border.setCenter(centerpanetext);
```

Listing A.6: FXML (JavaFX) code example, referring to the Java code in Listing A.5

```
<BorderPane>
        <top>
                <Label text="Page Title"/>
        </top>
        <center>
                <Label text="Some data here"/>
        </center>
</BorderPane>
```

HTML - HyperText Markup Language

Table A.9: Classification sheet for HTML

HTML - HyperText Markup Language	
Year:	1991 (first specification)
Domain:	Hypertext documents / Web pages
Target Platform:	
Origin:	CERN, Tim Berners-Lee
Stage of Development:	W3C standard
License:	open
Compatibility:	
Tools:	HTML editors (open source or commercial)
Language structure:	tag-based
Community/Impact:	international community, very popular
Media support:	vector graphics, audio, video
Webpage:	http://www.w3.org/html/

The Hypertext Markup Language HTML was developped in an effort of Tim Berners-Lee at CERN to develop a hypertext system that can be used over the internet. Initially, his specification[6] consisted of 18 elements for a relatively simple design of documents; 11 of those proposed elements are still part of HTML.
The focus of HTML shifted later on from simple annotation of documents for rendering using a default renderer included in a browser to very sophisiticated design possibilities including Cascading Style Sheets (CSS) for rendering and script languages like JavaScript for interaction with HTML elements (directly accessible).
Due to conflicts between competing browser developers, the definition of HTML is managed by the World Wide Web Consortium (W3C) with input from commercial software vendors.
Listing A.7 shows a "Hello world!"-webppage written in HTML.

Listing A.7: HTML code example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
        <head>
                <title>Hello World</title>
        </head>
        <body>
                Hello World!
        </body>
</html>
```

[6] www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html

M3L - MultiModal Markup Language

Table A.10: Classification sheet for M3L

M3L - MultiModal Markup Language	
Year: 2003–2006	
Domain:	abstract multimodal presentations
Target Platform:	SmartKom presentation module
Origin:	German Research Center for Artificial Intelligence (DFKI GmbH)
Stage of Development:	finalized
License:	internal use only
Compatibility:	N/A
Tools:	OIL2XSD [GPS+03], M3L API, any XML or text editor
Language structure:	XML based
Community/Impact:	internal use only
Media support:	N/A (abstract description only)
Webpage:	smartkom.org

M3L was developed in the Smartkom project. The underlying idea is to cover all data interfaces in a complex dialog system in one single, coherent language. The language definition has been decomposed in 40 different schema specifications in order to provide thematic organization and make the specification process manageable. The data flow between user input and system output continuously adds information to the M3L expression and refines it. Important semantic aspects are encoded in specific element structures. Complementing the language definition, an API has been developped as a lightweight programming interface. The basic structure and underlying idea of M3L resemble PreML (see page 209). Listing A.8 shows a partial M3L structure from [HKM+03] as an example. The shown intention lattice represents the interpretation result for a multimodal user input. More detailed information on M3L and the SMARTKOM project can be found in [HKM+03, Wah06].

Listing A.8: M3L code example

```
<intentionLattice>
        [ ]
        <hypothesisSequences>
                <hypothesisSequence>
                        <score>
                                <source> acoustic </source>
                                <value> 0.96448 </value>
                        </score>
                        <score>
                                <source> understanding </source>
                                <value> 0.91667 </value>
                        </score>
                        <hypothesis>
                                <discourseStatus>
                                        <discourseAction> set </discourseAction>
                                        <discourseTopic><goal> epg_info </goal></discourseTopic>
                                        [ ]
                                [ ]
                </hypothesisSequence>
                [ ]
        </hypothesisSequences>
</intentionLattice>
```

MXML

Table A.11: Classification sheet for MXML

MXML (inofficial backronym: Magic eXtensible Markup Language)	
Year:	2004
Domain:	Web (2D and 3D), rich internet applications, User Interface, Business Logic, mobile devices
Target Platform:	Web and Rich internet applications
Origin:	Macromedia, Adobe
Stage of Development:	product
License:	open source
Compatibility:	none (proprietary)
Tools:	Flex Server, Adobe Flash Builder, PHP PEAR
Language structure:	XML based plus scriptlanguage (actionscript)
Community/Impact:	Open source community, tools bot commercial and open source
Media support:	Vector graphics, audio, video
Webpage:	www.adobe.com/de/products/

MXML was initially developed by Macromedia. It is an XML-based markup language for declarative layouting the user interfaces. It is used in connection with Actionscript, which adds the application logic to the user interface. In combination, they are used to develop rich internet applications (RIA). The Actionscript parts can be directly included in the MXML code. MXML can be coded using a graphical editor or manually by editing the source code.
MXML can be used in connection with Flex Server, which dynamically compiles it into standard binary SWF files. There are also several alternatives, such as the Adobe Flash Builder IDE and a free Flex SDK which can also compile MXML into SWF. MXML can also be compiled in Adobe AIR format using Adobe Integrated Runtime. MXML is–similar to XAML (see page 222)–considered a propriertary format. It is tightly integrated in Adobe technologies and no converters into other user interface languages exist.
Listing A.9 shows a “Hello World!”-example in MXML.

Listing A.9: MXML code example

```
<?xml version="1.0" encoding="utf-8"?>
<s:WindowedApplication xmlns:fx="http://ns.adobe.com/mxml/2009"
                       xmlns:s="library://ns.adobe.com/flex/spark"
                       xmlns:mx="library://ns.adobe.com/flex/mx">
<s:Label x="40" y="20" text="Hello World!"/>
</s:WindowedApplication>
```

PreML -Presentation Markup Language

Table A.12: Classification sheet for PreML

PreML - Presentation Markup Language	
Year:	2007
Domain:	Web, rich internet applications
Target Platform:	desktop and mobile touch surfaces
Origin:	German Research Center for Artificial Intelligence (DFKI GmbH)
Stage of Development:	internal support / usage for DFKI multimodal dialog system and semantic output representation framework
License:	(no official release)
Compatibility:	
Tools:	editable with any text/xml editor, in combination with a dialog shell and workbench.
Language structure:	XML based
Community/Impact:	internal use only
Media support:	(abstract presentation only)
Webpage:	no official page; references in THESEUS project [HW11, Son10]

The Presentation Markup language PreML was developed at DFKI and is–like M3L–used for description of multimedia content as well as for small-footprint inner-system communication protocol. It was developped in the context of the Theseus project [Son10] but not limited to it. Although not officially stated, it can be considered the successor of M3L (see page 207).
The expressivness of the language ranges from encoding presentation tasks from the system to be presented to the user up to transmitting events like pointing gestures from the user back through the multimodal dialog system. It is derived from an ontological representation of the presentation task.
Listing A.10 shows a "Hello World!"-example in PreML. More information on PreML can be found in [BNP+07].

Listing A.10: PreML example code

```
<preml>
    <premldata>
        <spotlet>
            <text>Hello World!</text>
        </spotlet>
    </premldata>
</preml>
```

S5 - Simple Standards-Based Slide Show System

Table A.13: Classification sheet for S5

S5 - Simple Standards-Based Slide Show System	
Year:	2004
Domain:	Slide show
Target Platform:	
Origin:	Eric Meyer
Stage of Development:	
License:	public domain
Compatibility:	
Tools:	
Language structure:	XHTML based
Community/Impact:	
Media support:	
Webpage:	meyerweb.com/eric/tools/s5/

The Simple Standards-Based Slide Show System S5 was developped by Eric Meyer as a browser-based alternative to commercial slide-show programs. It is based entirely on XHTML, CSS, and JavaScript and should run in any browser. Additionally, it provides a printer friendly version. Meyer describes it on his webpage: “The markup used for the slides is very simple, highly semantic, and completely accessible. Anyone with even a smidgen of familiarity with HTML or XHTML can look at the markup and figure out how to adapt it to their particular needs. Anyone familiar with CSS can create their own slide show theme. It’s totally simple, and it’s totally standards-driven.”
Listing A.11 shows a “Hello World!”-slide written in S5.

Listing A.11: S5 code example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
        <title>Hello World slide</title>
        <meta name="version" content="S5 1.0" />
        <link rel="stylesheet" href="ui/slides.css" type="text/css" media="projection" id="slideProj" />
        <link rel="stylesheet" href="ui/opera.css" type="text/css" media="projection" id="operaFix" />
        <link rel="stylesheet" href="ui/print.css" type="text/css" media="print" id="slidePrint" />
        <script src="ui/slides.js" type="text/javascript"></script>
</head>
<body>
        <div class="layout">
                <div id="currentSlide"></div>
                <div id="header"></div>
                <div id="footer">
                        <div id="controls"></div>
                </div>
        </div>
        <div class="presentation">
                <div class="slide">
                        <h1>Hello World!</h1>
                </div>
        </div>
</body>
</html>
```

SCORM - Shareable Content Object Reference Model

Table A.14: Classification sheet for SCROM

SCORM - Shareable Content Object Reference Model	
Year:	1996
Domain:	web-based e-learning
Target Platform:	Learning Management Systems (LMS)
Origin:	Advanced Distributed Learning (ADL) initiative (U.S. Secretary of Defence)
Stage of Development:	de-facto industry standard
License:	
Compatibility:	
Tools:	Several SCORM products
Language structure:	Colelction of standards, XML based, zip-compressed for packagin (“package interchange format”)
Community/Impact:	
Media support:	
Webpage:	http://scorm.com/

The Shareable Content Object Reference Model SCORM is a collection of standards for interchanging content between e-learning programs. Especially, it governs the communication between online learning content and Learning Management Systems (LMS). SCORM also defines the packaging of content in transferable archive (zip) files as so called “package interchange format”. The XML-based format basically describes constraints in the order in which a learner can work trough the presented material, with possibilities to set bookmarks, etc. Listing A.12 shows an seqeuncing example from the SCORM webpage[7]: “In this example, Module 1 is an activity with three child activities (each represented by an item). In the sequencing for Module 1, the limit condition restricts this activity to two attempts. ” [BSSW02] critically discusses the advantages and shortcomings of the approach.

Listing A.12: SCORM example code

```
<item identifier=ITEM-1? identifierref=RES-1?>
<title>Module 1</title>
<item identifier=ITEM-2? identifierref=RES-2?>
<item identifier=ITEM-3? identifierref=RES-3?>
<item identifier=ITEM-4? identifierref=RES-4?>
<imsss:sequencing>
<imsss:limitConditions attemptLimit= 2 ?/>
<imsss:rollupRules>
<imsss:rollupRule minimumCount= 1 ?>
<imsss:rollupConditions>
<imsss:rollupCondition condition= satisfied />
</imsss:rollupConditions>
<imsss:rollupAction action= completed />
</imsss:rollupRule>
</imsss:rollupRules>
<imsss:randomizationControls selectCount= 1 ? randomizationTiming= onEachNewAttempt />
</imsss:sequencing>
</item>
```

[7] scorm.com/scorm-explained/technical-scorm/scorm-2004-overview-for-developers/

Silverlight (Microsoft Silverlight)

Table A.15: Classification sheet for Silverlight

Silverlight (Microsoft Silverlight)	
Year:	2007
Domain:	Interactive web-applications
Target Platform:	rich internet applications
Origin:	Microsoft Corporation
Stage of Development:	stable release
License:	freeware
Compatibility:	
Tools:	several, e.g., Microsoft Expression Blend
Language structure:	
Community/Impact:	
Media support:	graphics, audio, video
Webpage:	silverlight.net

Microsoft Silverlight is an application framework with objectives similar to Adobe Flash. It's runtime is provided as plug-in for webbrowsers under Microsoft Windows and Mac OS X. It is also one of the application development platforms for Windows Phone. The free implementation Moonlight, developped by Novell in cooperation with Microsoft, is availabe to bring Silverlight to Linux, FreeBSd and other platforms. The user interface in Silverlight is described using XAML (see page 222) and can be programmed using a subset of the .NET platform. It supports a wide range of multimedia file formats.

SMIL - Synchronized Multimedia Integration Language

Table A.16: Classification sheet for SMIL

SMIL - Synchronized Multimedia Integration Language	
Year:	1998
Domain:	Multimedia presentations, Online speech / lectures
Target Platform:	SMIL-Player, e.g., RealPlayer
Origin:	W3C
Stage of Development:	W3C standard
License:	
Compatibility:	
Tools:	RealPlayer
Language structure:	XML
Community/Impact:	decreasing, still in use for online lectures
Media support:	Vector graphics (svg), VoiceXML, MusicXML
Webpage:	www.w3.org/AudioVideo/

The Synchronized Multimedia Integration Language SMIL is one of the earliest attempts to formalize presentation content in a language. It dates back to 1998. [RHVO99] discusses its suitability as multimedia language for the web.
SMIL is recommended by the W3C and very popular for online lectures and interactive presentations, since it defines all necessary elements such as markup for timing, layout, animations, visual transitions, and media embedding, etc. SMIL is also highly integrated with other W3C languages; for example SMIL can be used as a means for animating vectorgraphics in SVG (see page 214). It can also be integrated in VoiceXML, MusicXML or RSS.
Listing A.13 shows a “Hello World!” example in SMIL[8].

Listing A.13: SMIL code example

```
<smil>
<head>
<layout> <!--Create the canvas and two display regions -->
        <root-layout width="248" height="300" background-color="blue" />
             <region id="a" top="20" left="64" />
             <region id="b" top="120" left="20"/>
</layout>
</head>
<body>
<par>
        <img src="http://www.content-networking.com/smil/hello.jpg"
             region="a"
             begin="0s"
             dur="6s"/> <!--Display "Hello" image now for 6 seconds -->
        <img src="http://www.content-networking.com/smil/earthrise.jpg"
             region="b"
             begin="2s"
             end="8s"/> <!--Display the "World" image after 2 seconds -->
        <audio src="http://www.content-networking.com/smil/hello.wav"
             begin="4s"/> <!-- Begin the audio after 4 seconds -->
</par>
</body>
</smil>
```

[8]cited from [HB05], chapter 4, page 93

SVG - Scalable Vector Graphics

Table A.17: Classification sheet for SVG

SVG - Scalable Vector Graphics	
Year:	2001
Domain:	Vector graphics
Target Platform:	Browser, graphic programs
Origin:	W3C
Stage of Development:	standard
License:	open source
Compatibility:	converters available
Tools:	Graphic editors (commercial, open source), e.g., inkscape
Language structure:	XML
Community/Impact:	
Media support:	Animation
Webpage:	www.w3.org/Graphics/SVG/

The Scalable Vector Graphics SVG format is developped by the W3C as an open standard since 1999. It can describe both static and animated vector graphics.
The XML based format can be edited using various graphic editors, such as Inkscape or Adobe Illustrator. The format is supported natively in the current version of all major browsers.
Listing A.14 shows a "Hello World!" example in XML.

Listing A.14: SVG code example

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" width='250px' height='250px'>
	<title>Hello World</title>
	<text x='60' y='100' fill='black'>Hello World!</text>
</svg>
```

SWEMMA - SmartWeb EMMA

Table A.18: Classification sheet for SWEMMA

SWEMMA - SmartWeb EMMA	
Year:	2004
Domain:	Dialog management, multimodal interaction annotation
Target Platform:	
Origin:	German Research Center for Artificial Intelligence (DFKI)
Stage of Development:	
License:	
Compatibility:	
Tools:	internally used only
Language structure:	XMLbased
Community/Impact:	
Media support:	
Webpage:	smartweb.org

SWEMMA is an extension of the EMMA standard developped at DFKI in the context of the Smartweb project. In addition to EMMA, it provides tags for wrapping the result of processed information, the status of a process monitoring annotation, out of vocabulary words, and a turn-id associated with an element.
Listing A.15 shows an example of SWEMMA code.

Listing A.15: SWEMMA code example

```
<emma:emma version="1.0">
        <swemma:result emma:id="DIA123" emma:process="DIA#42" emma:turn-id="42" emma:lang="de"
                emma:start="1087995961542" emma:end="1087995963542" emma:mode="speech">
                <swemma:result emma:confidence="1.0">
                        <emma:derived-from emma:resource="#spin1"/>
                                <speak version="1.0" xsi:schemaLocation="www.w3c.org/....xsd" xml:lang="de">
                                        1990 war Deutschland Fussballweltmeister.
                                </speak>
                </swemma:result>
        </swemma:result>
</emma:emma>
```

sXBL - SVG's XML Binding Language

Table A.19: Classification sheet for sXBL

sXBL - SVG's XML Binding Language	
Year:	2003 (as RCC - Rendering Custom Content)
Domain:	Interaction specification
Target Platform:	SVG
Origin:	W3C
Stage of Development:	working draft, not updated since 2005
License:	open
Compatibility:	SVG
Tools:	
Language structure:	XML based
Community/Impact:	W3C
Media support:	vector graphics
Webpage:	www.w3.org/TR/sXBL

The SVG's XML Binding Language is a mechanism for defining the presentation and interactive behavior of elements described in a namespace other than SVG's. The idea is to enable user interaction in SVG graphics such as an interactive flow chart. The approach seems abandoned; the last version of a working draft dated august 2005 and has not been updated since.
Listing A.16 shows a "Hello World!"-example in SVG using sXBL.

Listing A.16: sXBL example code

```
<svg width="10cm" height="3cm" viewBox="0 0 200 60" xmlns="http://www.w3.org/2000/svg" version="1.2">
        <title>Example xbl01-equivalent.svg - equivalent rendering for "hello world" sample file</title>
        <rect x="1" y="1" width="198" height="58" fill="none" stroke="blue"/>
        <g font-size="14" font-family="Verdana" transform="translate(10,35)">
                <g>
                        <text>Hello, world, using sXBL</text>
                </g>
        </g>
</svg>
```

UIML - User Interface Markup Language

Table A.20: Classification sheet for UIML

UIML - User Interface Markup Language	
Year:	1997
Domain:	User Interface meta language
Target Platform:	cross platform
Origin:	User Interface Markup Language Technical Committee (UIMLTC) of the Organization for the Advancement of Structured Information Standards (OASIS)
Stage of Development:	Commitee draft of Version 4.0 availabe
License:	open
Compatibility:	using converters / rendering engines
Tools:	several rendering engines
Language structure:	XML based
Community/Impact:	probably abandoned (webpage dead)
Media support:	
Webpage:	uiml.org

According to its specification, the "design objective of the User Interface Markup Language (UIML) is to provide a vendor-neutral, canonical representation of any user interface (UI) suitable for mapping to existing languages" [HSL$^+$08]. The method of describing a user interface in UIML is device independent, and the UIML code can be transformed to a target platform using specialized rendering engines. The development of the language was guided by the following questions: What are the parts comprising the UI? What is the presentation used for the parts? What is the content used in the UI? What is the behavior of the UI? What is the mapping of the parts to UI controls? What is the API of the connected business logic? While the approach aimed at reducing the effort of implementing user interfaces for different platforms by decleratively describing the desired outcome is quite useful, it has limitations in practice due to the different capabilities and resulting incompatibilities of the target devices. Listing A.17 shows a "Hello World!" example in UIML.

Listing A.17: UIML code example

```
<?xml version="1.0"?>
<!DOCTYPE uiml PUBLIC "-//OASIS//DTD_UIML_3.1_Draft//EN" "http://uiml.org/dtds/UIML4_0a.dtd">
<uiml>
	<interface>
		<structure>
			<part id="TopHello">
				<part id="hello" class="helloC"/>
			</part>
		</structure>
		<style>
			<property part-name="TopHello" name="rendering">Container</property>
			<property part-name="TopHello" name="content">Hello</property>
			<property part-class="helloC" name="rendering">Text</property>
			<property part-name="hello" name="content">Hello World!</property>
		</style>
	</interface>
	<peers> ... </peers>
</uiml>
```

VRML - Virtual Reality Modeling Language

Table A.21: Classification sheet for VRML

VRML - Virtual Reality Modeling Language	
Year:	1994
Domain:	Interactive vector graphics / Virtual Reality "Worlds"
Target Platform:	Web
Origin:	Web3D consortium
Stage of Development:	stable, mostly replaced by successor X3D (see page 221)
License:	open
Compatibility:	
Tools:	
Language structure:	based on Open Inventor (proprietary format of Silicon Graphics[9])
Community/Impact:	declining but still in use
Media support:	
Webpage:	web3d.org/x3d/vrml/

The Virtual Reality Modeling Language (VRML) is a file format for describing so called "worlds", i.e., interactive 3D visualizationsn with integrated multimedia content. It was the first standardized major VR modelling language. The development is goverened by the Web3D consortium, which was founded for this purpose.
VRML is a text file format, unlike most of the other formats described in this section not XML-based, which can specify 3D polygon worlds along with color, textures, lighting, transparency and so on. URLs for web content can be attached to elements of the world, so that fetching a web page can be triggered by interacting with a component in the 3D world. The VRML specification also enables shared worlds to be used by several users simultaneously. VRML is supported by a wide variety of platforms.
The specification follows six design criteria: authorability, composability, extensibility, be capable of implementation, performance, and scalability.
VRML is still in use but is in large parts replaced by its successor X3D (see page 221).
Listing A.18 shows a "Hello World!" example in VRML.

Listing A.18: VRML code example

```
Shape {
    appearance Appearance {
        material Material {
            diffuseColor 0.6 0.4 0.0
        }
    } geometry Text {
        string "Hello World!"
    }
}
```

[9] oss.sgi.com/projects/inventor/

VXML - Voice XML

Table A.22: Classification sheet for Voice XML

VXML - Voice XML

Year:	1997
Domain:	Interactive voice dialog specification, voice input annotation
Target Platform:	
Origin:	AT&T, IBM, Lucent, and Motorola
Stage of Development:	W3C standard
License:	open
Compatibility:	
Tools:	voice browser
Language structure:	XML-based
Community/Impact:	
Media support:	
Webpage:	voicexml.org (forum), vxml.org (tutorials)

VoiceXML was developped by a consortium of four companies as a markup language for voice browsers. In analogy to HTML running in a webbrowser, VoiceXML supports presenting voice documents and reacting appropriately with a user. One of the main areas of usage is automated telephone systems.
The scope of VoiceXML encompasses: spoken prompts (synthetic speech), output of audio files and streams, recognition of spoken words and phrases, recognition of touch tone (DTMF) key presses, recording of spoken input, control of dialog flow, telephony control (call transfer and hangup) [Rag01].
VoiceXML is standardized since version 2.0 and makes it easy for developpers to rapidly build new applications without getting into low-level implementation details. Listing A.19 shows a “Hello World!” example in VoiceXML.

Listing A.19: Voice XML code example

```
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
  <form>
    <block>
      <prompt>
        Hello world!
      </prompt>
    </block>
  </form>
</vxml>
```

W3C MMI - W3C Multimodal Interaction Language

Table A.23: Classification sheet for W3C MMI

W3C MMI - W3C Multimodal Interaction Language	
Year:	2002
Domain:	Multimodal interaction annotation
Target Platform:	web applications
Origin:	W3C
Stage of Development:	candidate recommendation
License:	open
Compatibility:	to other W3C standards
Tools:	any XML editor
Language structure:	XML based
Community/Impact:	mainly W3C
Media support:	
Webpage:	w3.org/TR/mmi-arch/

The Multimodal Interaction Language MMI is an effort of W3C to provide means for annotating interaction, formalizing the interaction event lifecycle and to support multimodal interaction scenarios on the web. So far, a general framework for interaction has been produced, along with a set of use cases and core requirements. Other W3C languages in more advanced state already cover part of the requirements, e.g., EMMA (see page 202), InkML[10], or EmotionML[11].
Listing A.20

Listing A.20: MMI code example

```
<mmi:mmi xmlns:mmi="http://www.w3.org/2008/04/mmi-arch" version="1.0" xmlns:vxml="http://www.w3.org/2001/vxml">
    <mmi:prepareRequest mmi:source="someURI" mmi:target="someOtherURI"
        mmi:context="URI-1" mmi:requestID="request-1" >
        <mmi:content>
            <vxml:vxml version="2.0">
                <vxml:form>
                    <vxml:block>Hello World!</vxml:block>
                </vxml:form>
            </vxml:vxml>
        </mmi:content>
    </mmi:prepareRequest>
</mmi:mmi>
```

[10] www.w3.org/TR/InkML/
[11] www.w3.org/TR/emotionml/

X3D

Table A.24: Classification sheet for X3D

X3D	
Year:	1999
Domain:	Interactive vector graphics / Virtual Reality “Worlds”
Target Platform:	Web
Origin:	Web3D consortium
Stage of Development:	ISO standard, W3C liaison
License:	open
Compatibility:	other rendering standards
Tools:	
Language structure:	XML
Community/Impact:	
Media support:	
Webpage:	web3d.org/x3d

X3D is an open-standard extension of VRML (Section A). Unlike the Open Inventor-like VRML language, it is based on XML. It provides a system for the storage, retrieval and playback of real time graphics content embedded in applications in an open architecture. Due to its large variety of features it can be customized for a large range of usages, including CAD, medical visualizations, training, simulation. entertainment, education, and more.
X3D is more refined than its predecessor VRML.
Listing A.21 shows a “Hello World!” example in X3D.

Listing A.21: X3D code example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN" "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile='Immersive' version='3.0' xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
            xsd:noNamespaceSchemaLocation =' http://www.web3d.org/specifications/x3d-3.0.xsd '>
        <head>
                ...
        </head>
        <Scene>
                <Shape>
                        <Appearance>
                                <Material diffuseColor='0 0 .5'/>
                        </Appearance>
                        <Text string='"Hello World!"' solid='false'>
                                <FontStyle size='0.5'/>
                        </Text>
                </Shape>
        </Scene>
</X3D>
```

XAML - eXtensible Application Markup Language

Table A.25: Classification sheet for XAML

XAML - eXtensible Application Markup Language	
Year:	2009
Domain:	Web, rich internet applications, stand-alone applications
Target Platform:	Windows Applications (WPF / .Net) / Web (Silverlight)
Origin:	Microsoft Corporation
Stage of Development:	product
License:	open ("Microsoft Open Specification Promise")
Compatibility:	
Tools:	Microsoft Visual Studio for visual editing, or any XML/text editor, e.g., XAMLPad
Language structure:	
Community/Impact:	
Media support:	graphics (2D, 3D)
Webpage:	

Microsofts Extensible Application Markup Language XAML is a declarative XML-based language. It is not limited to describing user interfaces but can be used more generally for initalizing any kind of structured values and objects. XAML is used in a variety of Microsoft products and frameworks, such as .NET, Silverlight (see page 212, or Windows Presentation Framework (WPF).
One of the design objectives of XAML is the reduction of complexity, i.e., designing code in XAML is easier than coding the same application in another language like C# for instance.
XAML files can be compiled into a Binary Application Markup Language (BAML), which can be used as a resource in a .NET Framework assembly.
Listing A.22 shows a "Hello World!" example in XAML.

Listing A.22: XAML code example

```
<Canvas xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Width="640" Height="480" Background="White">
   <TextBlock x:Name="txtHW" Width="72" Height="24" Canvas.Left="136" Canvas.Top="336"
                                        Text="Hello World!" TextWrapping="Wrap"/>
</Canvas>
```

XForms

Table A.26: Classification sheet for XForms

XForms	
Year:	2003 (first specification)
Domain:	data processing / forms
Target Platform:	browser (via plugins) / standalone user interfaces
Origin:	W3C
Stage of Development:	stable / W3C recommendation
License:	open
Compatibility:	
Tools:	XML editors / text editors
Language structure:	XML based
Community/Impact:	
Media support:	
Webpage:	w3.org/MarkUp/Forms/

XForms was developped as an extension/improvement over HTML Forms. It is by design easy to understand for HTML developers, and offers advantages, e.g., being simpler and allowing beforehand checking of form entries and thus avoiding uneccessary requests to a webserver. In the official documentation[12], several primary benefits are mentioned, such as device-independence, ability to handle input in XML form, combining existing technilogies, being accessible, and internationalization. To be more precise, there are several features included in XForms that are not availbale in HTML forms. Data can be checked while being typed in, required fields can be indicated, the same form can be submitted to different servers, or values can be automatically calclated from other entered values, just to name a few. So far, not major browser supports XForms natively, but plugins or client-side extensions are available. XForms is also supported by LibreOffice[13]. Listing A.23 shows how to include an XForm for entering and submitting a search request into an HTML page.

Listing A.23: XForms code example

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:f="http://www.w3.org/2002/xforms">
<head>
    <title>Search</title>
    <f:model>
        <f:submission action="http://example.com/search" method="get" id="s"/>
    </f:model>
</head>
<body>
    <p>
        <f:input ref="q"><f:label>Find</f:label></f:input>
        <f:submit submission="s"><f:label>Go</f:label></f:submit>
    </p>
</body>
</html>
```

[12] w3.org/MarkUp/Forms/2003/xforms-faq.html

[13] wiki.services.openoffice.org/wiki/Documentation/OOoAuthors_User_Manual/Writer_Guide/XForms

XIML - eXtensible Interface Markup Language

Table A.27: Classification sheet for XIML

XIML - eXtensible Interface Markup Language	
Year:	2002
Domain:	User interface
Target Platform:	
Origin:	Red Whale Software
Stage of Development:	
License:	
Compatibility:	
Tools:	
Language structure:	XML based
Community/Impact:	
Media support:	
Webpage:	ximl.org (no longer available)

XIML is a markup language describing both the user interface and its interactions. It has been introduced in [PE02], but it seems that development has discontinued. The authors of that paper describe the requirements on which XIML is based on as: central repository of data, comprehensive lifecycle support, support for abstract and concrete elements, relational support, and that the underlying technology is based on industry standards such as XML and does not impose any particular methodologies on design, operation, and evaluation.

[DSZ07] used XIML in the process of reverse engineering existing GUIs to a generic description, which could in turn be used for rendering the same GUI on different platforms.

XML 3D

Table A.28: Classification sheet for XML 3D

XML 3D	
Year:	2010
Domain:	Web (3D)
Target Platform:	Mozilla Webbrowser / Chromium
Origin:	DFKI, Fraunhofer IGD and the Web3D Consortium.
Stage of Development:	under development
License:	
Compatibility:	existing web technologies
Tools:	
Language structure:	
Community/Impact:	
Media support:	
Webpage:	xml3d.org

XML is introduced by [SKR+10] and described as an addition to existing web technologies to support 3D graphics in web browsers with a "minimum set of additions that fully support interactive 3D content as an integral part of mixed 2D/3DWeb documents.". This approach is fully integrated in HTML, CSS, DOM and JavaScript, which distinguishes it from X3D (see page 221).
Listing A.24 shows an XML 3D example of a rectangular mesh with texture coordinates.

Listing A.24: XML 3D code example

```
<xml3d id="MyXml3d" style="width: 150px; height: 100px; border: 1px solid gray"
          xmlns="http://www.xml3d.org/2009/xml3d">
  <mesh type="triangles">
    <int name="index">0 1 2  1 2 3</int>
    <float3 name="position">-1 -1 -5 1 -1 -5 -1 1 -5 1 1 -5</float3>
    <float3 name="normal">0 0 1  0 0 1  0 0 1 0 0 1</float3>
    <float2 name="texcoord">0.0 0.0 1.0 0.0 0.0 1.0 1.0 1.0</float2>
  </mesh>
</xml3d>
```

XUL - XML User-Interface Language

Table A.29: Classification sheet for XUL

XUL - XML User-Interface Language	
Year:	
Domain:	Web, Mozilla-based applications
Target Platform:	Mozilla Products / Gecko Engine-based
Origin:	Mozilla Foundation
Stage of Development:	
License:	
Compatibility:	
Tools:	
Language structure:	HTML-based
Community/Impact:	Mozilla
Media support:	
Webpage:	

The XML User Interface Language XUL is a language developped by Mozilla with the aim of building “feature-rich cross platform applications that can run connected or disconnected from the Internet.”[14] One of the advantages is the easy customization, localization, and branding of these applications with alternate text and design.
Listing A.25 shows a “Hello World!”-example in XUL.

Listing A.25: XUL code example

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window id="main" title="Hello World" width="300" height="300"
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
   <caption label="Hello World"/>
</window>
```

[14]developer.mozilla.org/en/XUL

References

References

[AA10] Darçin Akin and Bülent Akba. A neural network (NN) model to predict intersection crashes based upon driver, vehicle and roadway surface characteristics. *Scientific Research and Essays*, 5(19):2837–2847, 2010.

[AAD+05] Luisa Andreone, Angelos Amditis, Enrica Deregibus, Sergio Damiani, Domenico Morreale, and Francesco Bellotti. Beyond Context-Awareness: Driver-Vehicle-Environment Adaptivity. From The Comunicar To The Aide Concept. *Proceedings of the 16th IFAC World Congress*, 2005.

[AB10] Amna Asif and Susanne Boll. Where to turn my car?: comparison of a tactile display and a conventional car navigation system under high load condition. In *Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 64–71. ACM, 2010.

[Abb07] Chris Abbott. *E-inclusion: Learning difficulties and digital technologies*. Futurelab Bristol, UK, 2007.

[ADHW88] J.F. Antin, T.A. Dingus, M.C. Hulse, and W.W. Wierwille. The effects of spatial ability on automobile navigation. *Trends in Ergonomics/Human Factors*, 5(24):1–248, 1988.

[ADN08] Christian Artigues, Sophie Demassey, and Emmanuel Neron. *Resource-constrained Project Scheduling. Models, Algorithms, Extensions and Applications.* John Wiley & Sons, March 2008.

[AFG+93] Elisabeth André, Wolfgang Finkler, Winfried Graf, Thomas Rist, Anne Schauder, and Wolfgang Wahlster. Wip: The automatic synthesis of multimodal presentations. *Intelligent multimedia interfaces*, 9:75–93, 1993.

[AID06] AIDE Deliverable D2.2.6: Subjective Assessment methods for Workload, 2006.

[AKS$^+$10] Florian Alt, Dagmar Kern, Fabian Schulte, Bastian Pfleging, Alireza Sahami Shirazi, and Albrecht Schmidt. Enabling micro-entertainment in vehicles based on context information. In *Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '10, pages 117–124, New York, NY, USA, 2010. ACM.

[ALDG12] Ignacio Alvarez, Karmele Lopez de Ipiña, Shaundra B. Daily, and Juan E. Gilbert. Emotional Adaptive Vehicle User Interfaces: moderating negative effects of failed technology interactions while driving. In *Adjunct Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)*, pages 57–60, Portsmouth, NH, USA, October 1719 2012.

[Ale08] Jan Alexandersson. i2home–towards a universal home environment for the elderly and disabled. *Künstliche Intelligenz*, 8(3):66–68, 2008.

[AMR97] Elisabeth André, Jochen Müller, and Thomas Rist. WIP/PPP: automatic generation of personalized multimedia presentations. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 407–408. ACM, 1997.

[And00] Elisabeth André. The generation of multimedia presentations. *A handbook of natural language processing: techniques and applications for the processing of language as text*, pages 305–327, 2000.

[APAB06] Angelos Amditis, Aris Polychronopoulos, Luisa Andreone, and Evangelos Bekiaris. Communication and interaction strategies in automotive adaptive interfaces. *Cognition, Technology and Work*, 8:193–199, 2006.

[AR93] Elisabeth André and Thomas Rist. The design of illustrated documents as a planning task. In *Intelligent multimedia interfaces*, pages 94–116. American Association for Artificial Intelligence, 1993.

[AR96] Elisabeth André and Thomas Rist. Coping with temporal constraints in multimedia presentation planning. In *Proceedings of the National Conference on Artificial Intelligence*, pages 142–147, 1996.

[BBM07] Jorge A. Baier, Fahiem Bacchus, and Sheila A. McIlraith. A heuristic search approach to planning with temporally extended preferences. In *Proce. 20th International Joint Conference on AI (IJCAI-07)*, pages 1808–1815, 2007.

[BBW09] Daniel Baldauf, Esther Burgard, and Marc Wittmann. Time perception as a workload measure in simulated car driving. *Applied ergonomics*, 40(5):929–935, 2009.

[BD98a] Mathias Bauer and Dietmar Dengler. Infobeansconfiguration of personalized information assistants. In *Proceedings of the 4th international conference on Intelligent user interfaces*, pages 153–156. ACM, 1998.

[BD98b] Mathias Bauer and Dietmar Dengler. Trias-an architecture for trainable information assistants. *Working Notes of the Autonomous Agents*, 98, 1998.

[BD99] Mathias Bauer and Dietmar Dengler. Trias: Trainable information assistants for cooperative problem solving. In *Proceedings of the third annual conference on Autonomous Agents*, pages 260–267. ACM, 1999.

[BDP00] Mathias Bauer, Dietmar Dengler, and Gabriele Paul. Instructible information agents for web mining. In *Proceedings of the 5th international conference on Intelligent user interfaces*, pages 21–28. ACM, 2000.

[BDP01] Mathias Bauer, Dietmar Dengler, and Gabriele Paul. Trainable information agents for the web. *Lieberman [14]*, pages 87–114, 2001.

[BEAM06] Robert Broström, Johan Engström, Anders Angvall, and Gustav Markkula. Towards the Next Generation Intelligent Driver Information System (IDIS): The Volvo Car Interaction Manager Concept. *Proceedings of the 2006 ITS World Congress London*, 2006.

[BEM11] Daniel Braun, Christoph Endres, and Christian Müller. Determination of Mobility Context using Low-Level Data. In *Adjunct Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2011)*, pages 41–42, Salzburg, Austria, November 2011.

[B'f05] Reza B'far. *Mobile Computing Principles: designing and developing mobile applications with UML and XML.* Cambridge University Press, 2005.

[BFF$^+$97] M. Bordegoni, G. Faconti, S. Feiner, M.T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson. A standard reference model for intelligent multimedia presentation systems. *Computer standards & interfaces*, 18(6):477–496, 1997.

[BJ99] André Berthold and Anthony Jameson. Interpreting symptoms of cognitive load in speech input. *Courses and Lectures*, pages 235–244, 1999.

[BJ05] Andreas Butz and Ralf Jung. Seamless user notification in ambient soundscapes. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 320–322. ACM, 2005.

[BK00] Peter Brucker and Sigrid Knust. A linear programming and constraint propagation-based lower bound for the rcpsp. *European Journal of Operational Research*, 127(2):355–362, 2000.

[BKST98] Peter Brucker, Sigrid Knust, Arno Schoo, and Olaf Thiele. A branch and bound algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 107(2):272–288, 1998.

[BLRK83] Jacek Blazewicz, Jan Lenstra, and Alexander Rinnooy Kan. Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, 5:11–24, 1983.

[BM91] Murray R. Barrick and Michael K. Mount. The big five personality dimensions and job performance: a meta-analysis. *Personnel psychology*, 44(1):1–26, 1991.

[BNP+07] Jana Besser, Robert Neßelrath, Alexander Pfalzgraf, Jan Schehl, Jochen Steigner, and Norbert Pfleger. D6. 1: Selection of technical components and system specification. i2home, 2007.

[Boe02] Paul Boersma. Praat, a system for doing phonetics by computer. *Glot international*, 5(9/10):341–345, 2002.

[Bor85] Jürgen Bortz. *Lehrbuch der Statistik*. Springer, 1985.

[Bos91] Bosch. Can specification, 2.0. http://www.semiconductors.bosch.de/media/pdf/canliteratur/ can2spec.pdf, 1991.

[Bro57] P.L. Broadhurst. Emotionality and the yerkes-dodson law. *Journal of Experimental Psychology*, 54(5):345, 1957.

[BS01] Stefan Berti and Erich Schröger. A comparison of auditory and visual distraction effects: behavioral and event-related indices. *Cognitive Brain Research*, 10(3):265–273, 2001.

[BSS04] Andreas Butz, Michael Schneider, and Mira Spassova. Searchlight–a lightweight search function for pervasive environments. *Pervasive Computing*, pages 351–356, 2004.

[BSSW02] Oliver Bohl, Jörg Scheuhase, Ruth Sengler, and Udo Winand. The sharable content object reference model (scorm)-a critical review. In *Computers in Education, 2002. Proceedings. International Conference on*, pages 950–951. IEEE, 2002.

[BSW11] Roland Bader, Oliver Siegmund, and Wolfgang Woerndl. A study on user acceptance of proactive in-vehiclerecommender systems. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications. International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI-11), November 29 - December 2, Salzburg, Austria*, pages 47–54. ACM, 2011.

[Bub03] Heiner Bubb. Fahrerassistenz–primär ein Beitrag zum Komfort oder für die Sicherheit? *VDI-Berichte*, pages 25–44, 2003.

[Cai07] Brad Cain. A Review of the Mental Workload Literature. Technical report, NATO, 2007.

[Cas13] Sandro Castronovo. *Beyond the Push-Paradigm: Enhanced forms of Information Access and Novel Application Areas for Vehicle-2X Communication Networks*. PhD thesis, Computer Science Institute, Saarland University, Saarbrücken, Germany, 2013. to appear.

[CAVT87] Nicos Christofides, Ramon. Alvarez-Valdes, and Jose Manuel Tamarit. Project scheduling with resource constraints: A branch and bound approach. *European Journal of Operational Research*, 29(3):262–273, 1987.

[CCMM09] Yujia Cao, Sandro Castronovo, Angela Mahr, and Christian Müller. On timing and modality choice with local danger warnings for drivers. In *Proc. of the 1st Intern. Conf. on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2009)*, pages 75–78, Essen, September 2009. ACM Press.

[CE10] Sandro Castronovo and Christoph Endres. Interface Outside: Extending the V2X Communication Framework for Vulnerable Road User Protection. In *Adjunct proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010)*, page 2, Pittsburgh, PA, USA, November 2010.

[CEF+11] Sandro Castronovo, Christoph Endres, Tobias Del Fabro, Nils Schnabel, and Christian Müller. Multimodal Conspicuity-Enhancement for E-Bikes (Demo): the Hybrid Reality Model of Environment Transformation. In *Proceedings of the International Conference on Intelligent User*

Interfaces (IUI 2011), pages 433–434, Palo Alto, CA, USA, February 2011.

[CEM+12] Sandro Castronovo, Christoph Endres, Angela Mahr, Christian Müller, Karl Naab, and Tim Schwartz. Working Document W22.2 – HMI Handbuch. Technical report, simTD, 2012.

[CLLH01] Jens Clausen, Jesper Larsen, Allen Larsen, and Jesper Hansen. Disruption management - operations research between planning and execution. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2001.

[CMC+10] Yujia Cao, Angela Mahr, Sandro Castronovo, Mariet Theune, Christoph Stahl, and Christian Müller. Local Danger Warnings for Drivers: The Effect of Modality and Level of Assistance on Driver Reaction. In *Proc. Intern. Conf. on Intelligent User Interfaces (IUI 2010)*, pages 239–248, Hong Kong, February 2010. ACM.

[CME10] Sandro Castronovo, Angela Mahr, and Christoph Endres. On In-Car User Interfaces for Car-2-X Pull-Applications: Design Considerations for HMIs. In *Adjunct proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010)*, page 1, Pittsburgh, PA, USA, November 2010.

[CRM11] Joseph F. Coughlin, Bryan Reimer, and Bruce Mehler. Monitoring, managing, and motivating driver safety and well-being. *Pervasive Computing, IEEE*, 10(3):14–21, 2011.

[CS10] Matthew W. Crocker and Jörg Siekmann. *Resource-adaptive cognitive processes*. Springer, 2010.

[CSGB+89] William H. Corwin, Diane L. Sandry-Garza, Michael H. Biferno, George P. Boucek Jr, and Aileen L. Logan. Assessment of Crew Workload Measurement Methods, Techniques and Procedures. Volume 1. Process, Methods and Results. Technical report, DTIC Document, 1989.

[CWSS08] Jeff K. Caird, Chelsea R. Willness, Piers Steel, and Chip Scialfa. A meta-analysis of the effects of cell phones on driver performance. *Accident Analysis & Prevention*, 40(4):1282–1293, 2008.

[DAM05] Sophie Demassey, Christian Artigues, and Philippe Michelon. Constraint-propagation-based cutting planes: An application to the resource-constrained project scheduling problem. *INFORMS JOURNAL ON COMPUTING*, 17:52–65, 2005.

[Dav89] Fred D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340, 1989.

[DB88] Thomas Dean and Mark Boddy. An analysis of time-dependent planning. In *Proceedings of the seventh national conference on artificial intelligence*, pages 49–54, 1988.

[DeB04] S.L. DeBoer. *Emergency Newborn Care*. Trafford Publishing, 2004.

[Del11] Friedrich Christian Delius. *Die Frau, für die ich den Computer erfand*. Rowohlt, 2011.

[DHM98] Paula A. Desmond, Peter A. Hancock, and Janelle L. Monette. Fatigue and automation-induced impairments in simulated driving performance. *Transportation Research Record: Journal of the Transportation Research Board*, 1628:8–14, 1998.

[DK09] Vera Demberg and Frank Keller. A computational model of prediction in human parsing: Unifying locality and surprisal effects. In *Proceedings of the 29th meeting of the Cognitive Science Society (CogSci-09)*, 2009.

[DK10] Veronika Dimitrova-Krause. Physiological Measurement of Driver Stress Induced by Car2X-Based Local Danger Warnings. Master's thesis, Saarland University, 2010.

[DS11] Vera Demberg and Asad B. Sayeed. Linguistic cognitive load: implications for automotive UIs. In *Adjunct Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2011)*, 2011.

[DSZ07] Giuseppe Di Santo and Eugenio Zimeo. Reversing guis to ximl descriptions for the adaptation to heterogeneous devices. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 1456–1460. ACM, 2007.

[EB10] Christoph Endres and Daniel Braun. Pleopatra: A Semi-Automatic Status-Posting Prototype For Future In-Car Use. In *Adjunct proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010)*, page 7, Pittsburgh, PA, USA, November 2010.

[EBM05] Christoph Endres, Andreas Butz, and Asa MacWilliams. A Survey of Software Infrastructures and Frameworks for Ubiquitous Computing. *Mobile Information Systems Journal*, 1(1):41–80, January–March 2005.

[EBM11] Christoph Endres, Daniel Braun, and Christian Müller. Prototyping a Semi-Automatic In-Car Texting Assistant. In *Proceedings of the 3rd Workshop on Multimodal Interfaces for Automotive Applications (MIAA 2011)*, pages 57–60, Palo Alto, CA, USA, February 2011.

[EBP+12] Thomas Ellenbuerger, Arnaud Boutin, Stefan Panzer, Yannick Blandin, Charles H. Shea, Lennart Fischer, and Joerg Schorer. Observational training in visual half-fields and the coding of movement sequences. *Human Movment Science*, 2012.

[EC10] Christoph Endres and Sandro Castronovo. Intelligent Environment Car: A New Perspective. In *Adjunct proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010)*, page 9, Pittsburgh, PA, USA, November 2010.

[ED10] Christoph Endres and Svilen Dimitrov. Using a Theremin for Micro-Gesture Recognition in an Automotive Environment. In *Adjunct proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010)*, page 8, Pittsburgh, PA, USA, November 2010.

[EFM12a] Christoph Endres, Michael Feld, and Christian Müller. A Layout-based Estimation of Presentation Complexity. In *Adjunct Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2012)*, pages 19–20, Portsmouth, New Hampshire, USA, October 2012.

[EFM12b] Christoph Endres, Michael Feld, and Christian Müller. Proposing a Meta-Language for Specifying Presentation Complexity in order to Support System Situation Awareness. In *W3C Web and Automotive Workshop*, 2012.

[EFSM10] Christoph Endres, Michael Feld, Tim Schwartz, and Christian Müller. Cinematic Analysis of Automotive Personalization. In *IUI 2010 Proceedings of the 14th International Conference on Intelligent User Interfaces*, Hong Kong, China, February 2010. ACM Press.

[EG10] Hilko Ehmen and Mehmet Gövercin. Deliverable D2.6.3: Methoden der Leistungsprädiktion; TP2: Notfallassistenz. Technical report, SmartSenior, 2010.

[EHN94] Kutluhan Erol, James Hendler, and Dana S. Nau. Umcp: A sound and complete procedure for hierarchical task-network planning. In *Proc. AIPS*, volume 94, pages 249–254, 1994.

[ELW98] Robert Eckstein, Marc Loy, and Dave Wood. *Java swing*. O'Reilly & Associates, Inc., 1998.

[EM12] Christoph Endres and Christian Müller. A Graph-Search Approach on Resource-Constrained Scheduling Problems and its Application to Advanced Driver Assistance Systems. In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART 2012)*, Vilamoura, Algarve, Portugal, February 2012.

[EMB10] Christoph Endres, Jan Miksatko, and Daniel Braun. Youldeco - Exploiting the Power of Online Social Networks for Eco-Friendly Driving. In *Adjunct proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010)*, page 5, Pittsburgh, PA, USA, November 2010.

[EMB12] Christoph Endres, Rafael Math, and Daniel Braun. Simulator-based Evaluation on the Impact of Visual Complexity and Speed on Drivers Cognitive Load. In *Adjunct Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2012)*, pages 30–31, Portsmouth, New Hampshire, USA, October 2012.

[EMM11] Christoph Endres, Gerrit Meixner, and Christian Müller. MIAA 2011: Multimodal Interaction for the Intelligent Environment Car. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI 2011)*, pages 473–474, Palo Alto, CA, USA, February 2011.

[EMW99] Christoph Endres, Markus Meyer, and Wolfgang Wahlster. Personal Picture Finder: Ein Internet-Agent zur wissensbasierten Suche nach Personenphotos. In *Proceedings of Online99*, pages 211–232. Velbert, 1999.

[End87] Mica R. Endsley. Sagat: A methodology for the measurement of situation awareness (nor doc 87-83). *Hawthorne, CA: Northrop Corporation*, 1987.

[End95] Mica R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.

[End99a] Christoph Endres. Personal Picture Finder – A Softbot for the World Wide Web. Diplomarbeit, Universität des Saarlandes, Saarbrücken, Germany, May 1999.

[End99b] Christoph Endres. The MultiHttpServer – A Parallel Pull Engine. Technical Memo TM-99-04, German Research Center for Artificial Intelligence (DFKI GmbH), Saarbrücken, Germany, April 1999.

[End00] Mica R. Endsley. Theoretical underpinnings of situation awareness: A critical review. *Situation awareness analysis and measurement*, pages 3–32, 2000.

[End03] Christoph Endres. Towards a Software Architecture for Device Management in Instrumented Environments. In *Fifth International Conference On Ubiquitous Computing (UbiComp) – Adjunct Proceedings*, pages 245–246, Seattle, October 2003.

[End12] Christoph Endres. Real-time Assessment of Driver Cognitive Load as a prerequisite for the situation-aware Presentation Toolkit PresTK. In *Adjunct Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2012)*, pages 76–79, Portsmouth, New Hampshire, USA, October 2012.

[ES10] Christoph Endres and Tim Schwartz. Crossmodal Referencing as Automotive Fission Concept. In *Adjunct proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010)*, page 6, Pittsburgh, PA, USA, November 2010.

[ESM11] Christoph Endres, Tim Schwartz, and Christian Müller. "Geremin": 2D Microgestures for Drivers Based on Electric Field Sensing. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI 2011)*, pages 327–330, Palo Alto, CA, USA, February 2011.

[FD12] Michael J. Flannagan and Joel M. Devonshire. Effects of automotive interior lighting on driver vision. *LEUKOS–The Journal of the Illuminating Engineering Society of North America*, 9(1):9–23, 2012.

[FE10] Michael Feld and Christoph Endres. Sharing User and Context Models in Automotive HMI. In *Adjunct proceedings of the 2nd Interna-*

tional Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010), page 10, Pittsburgh, PA, USA, November 2010.

[FII09] Sandrine Fischer, Makoto Itoh, and Toshiyuki Inagaki. A cognitive schema approach to diagnose intuitiveness: an application to onboard computers. In *Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '09, pages 35–42, New York, NY, USA, 2009. ACM.

[Fla95] John M. Flach. Situation awareness: Proceed with caution. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):149–157, 1995.

[FM91] Steven K. Feiner and Kathleen R. McKeown. Automating the generation of coordinated multimedia explanations. *Computer*, 24(10):33–41, 1991.

[FM11] Michael Feld and Christian Müller. The automotive ontology: Managing knowledge inside the vehicle and sharing it between cars. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications. International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI-11), November 29 - December 2, Salzburg, Austria*, pages 79–86. ACM, 11 2011.

[FRC$^+$07] Kimberley A. Faulkner, Mark S. Redfern, Jane A. Cauley, Douglas P. Landsittel, Stephanie A. Studenski, Caterina Rosano, Eleanor M. Simonsick, Tamara B. Harris, Ronald I. Shorr, Hilsa N. Ayonayon, and Anne B. Newman. Multitasking: association between poorer performance and a history of recurrent falls. *Journal of the American Geriatrics Society*, 55(4):570–576, 2007.

[Gar12] Sandro Rodriguez Garzon. Situation-Aware Personalization of Automotive User Interfaces. In *Adjunct Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)*, pages 15–16, Portsmouth, NH, USA, October 1719 2012.

[Geb07] Patrick Gebhard. *Emotionalisierung interaktiver Virtueller Charaktere– Ein mehrschichtiges Computermodell zur Erzeugung und Simulation von Gefühlen in Echtzeit*. PhD thesis, Saarland University, 2007.

[Gib00] Edward Gibson. The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain*, pages 95–126, 2000.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, first edition edition, January 1979.

[GKKR03] Patrick Gebhard, Michael Kipp, Martin Klesen, and Thomas Rist. Authoring scenes for adaptive, interactive performances. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 725–732. ACM, 2003.

[GMK11] Patrick Gebhard, Gregor Mehlmann, and Michael Kipp. Visual scenemakera tool for authoring interactive virtual characters. *Journal on Multimodal User Interfaces*, pages 1–9, 2011.

[GPS+03] Iryna Gurevych, Robert Porzel, Elena Slinko, Norbert Pfleger, Jan Alexandersson, and Stefan Merten. Less is more: Using a single knowledge representation in dialogue systems. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning-Volume 9*, pages 14–21. Association for Computational Linguistics, 2003.

[GSB+04] Christhard Gelau, Roland Schindhelm, Klaus Bengler, Andreas Engelsberg, Villy Portouli, Katia Pagle, and Harald Berninger. AIDE Deliverable 4.3.2: Recommendations for HMI Guidelines and Standards. Technical report, AIDE, Adaptive Integrated Driver-vehicle InterfacE, 2004.

[GSC+08] Patrick Gebhard, Marc Schröder, Marcela Charfuelan, Christoph Endres, Michael Kipp, Sathish Chandra Pammi, Martin Rumpler, and Oytun Türk. IDEAS4Games: Building Expressive Virtual Characters for Computer Games. In Helmut Prendinger, James C. Lester, and Mitsuru Ishizuka, editors, *Intelligent Virtual Agents, 8th International Conference Proceedings. Intelligent Virtual Agents, 8th International Conference (IVA-08), September 1-3, Tokyo, Japan*, volume 5208 of *Lecture Notes in Artificial Intelligence, LNAI*, pages 426–440. Springer, 2008.

[Hal01] John Hale. A probabilistic earley parser as a psycholinguistic model. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics, 2001.

[Hal08] Catalina Hallett. Multi-modal presentation of medical histories. In *Proceedings of the 13th international conference on Intelligent user interfaces*, IUI '08, pages 80–89, New York, NY, USA, 2008. ACM.

[Har06] Sandra G. Hart. NASA-task load index (NASA-TLX); 20 years later. *Human Factors and Ergonomics Society Annual Meeting Proceedings*, 50(9):904–908, 2006.

[HB05] Markus Hofmann and Leland R. Beaumont. *Content networking: architecture, protocols, and practice*. Morgan Kaufmann Pub, 2005.

[HFH+09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[HHMK00] Laurence Hartley, Tim Horberry, Nick Mabbott, and Gerald P. Krueger. *Review of fatigue detection and prediction technologies*. National Road Transport Commission, 2000.

[HJC10] Daz L. Hibberd, Samantha L. Jamson, and Oliver M. J. Carsten. Managing in-vehicle distractions: evidence from the psychological refractory period paradigm. In *Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '10, pages 4–11, New York, NY, USA, 2010. ACM.

[HKM+03] Gerd Herzog, Heinz Kirchmann, Stefan Merten, Alassane Ndiaye, and Peter Poller. Multiplatform testbed: An integration platform for multimodal dialog systems. In *Proceedings of the HLT-NAACL 2003 workshop on Software engineering and architecture of language technology systems-Volume 8*, pages 75–82. Association for Computational Linguistics, 2003.

[HNE02] J.L. Harbluk, Y.I. Noy, and M. Eizenman. The impact of cognitive distraction on driver visual behaviour and vehicle control. Technical report, Transport Canada, 2002.

[HNM+04] Gerd Herzog, Alassane Ndiaye, Stefan Merten, Heinz Kirchmann, Tilman Becker, and Peter Poller. Large-scale software integration for spoken language and multimodal dialog systems. *Natural Language Engineering*, 10(3-4):283–305, 2004.

[Hor11] Eric Horvitz. Bricks, arches, and cathedrals: reflections on paths to deeper human-computer symbioses. In *Proceedings of the 16th international conference on Intelligent user interfaces*, page 1. ACM, 2011.

[HS88] S.G. Hart and L.E. Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Human mental workload*, 1:139–183, 1988.

[HSB+05] Dominik Heckmann, Tim Schwartz, Boris Brandherm, Michael Schmitz, and Margeritta von Wilamowitz-Moellendorff. GUMO–the general user model ontology. *User modeling 2005*, pages 428–432, 2005.

[HSL+08] James Helms, Robbie Schaefer, Kris Luyten, Jean Vanderdonckt, Jo Vermeule, and Marc Abrams. User interface markup language (uiml) specification version 4.0. Technical report, Technical report, Oasis UIML TC, 2008.

[HSM+07] Dominik Heckmann, Eric Schwarzkopf, Junichiro Mori, Dietmar Dengler, and Alexander Kröner. The user model and context ontology gumo revisited for future web 2.0 extensions. *Contexts and Ontologies: Representation and Reasoning (2007)*, pages 37–46, 2007.

[HSP91] CE Hartel, K. Smith, and C. Prince. Defining aircrew coordination: Searching mishaps for meaning. In *Sixth International Symposium on Aviation Psychology, Columbus, OH*, 1991.

[HW11] Lutz Heuser and Wolfgang Wahlster. *Internet der Dienste*. Springer, 2011.

[HWTB07] Otmar Hilliges, Maria Wagner, Lucia Terrenghi, and Andreas Butz. The living-room: browsing, organizing and presenting digital image collections in interactive environments. In *Intelligent Environments, 2007. IE 07. 3rd IET International Conference on*, pages 552–559. IET, 2007.

[IBH10] Francisco Iacobelli, Larry Birnbaum, and Kristian J. Hammond. Tell me more, not just more of the same. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 81–90. ACM, 2010.

[IKL11] Tobias Islinger, Thorsten Köhler, and Bernd Ludwig. Driver Distraction Analysis based on FFT of steering wheel angle. In *Adjunct Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2011)*, page 70, Salzburg, Austria, November 2011.

[ISO08] Road vehicles–Ergonomic aspects of transport information and control systems–Simulated lane change test to assess invehicle secondary task demand. ISO Norm 26022, 2008.

[IWB93] Daniel Imbeau, Walter W. Wierwille, and Yves Beauchamp. Age, display design and driving perfomance. In Brian Peacock and Waldemar Karwowski, editors, *Automotive ergonomics*, chapter 16, pages 339–357. Taylor and Francis London, 1993.

[IWWC89] Daniel Imbeau, Walter W. Wierwille, Laurie D. Wolf, and Gail A. Chun. Effects of Instrument Panel Luminance and Chromaticity on Reading Performance and Preference in Simulated Driving. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 31(2):147–160, 1989.

[JB05] Ralf Jung and Andreas Butz. Effectiveness of user notification in ambient soundscapes. In *Proceedings of the workshop on Auditory Displays for Mobile Context-Aware Systems at Pervasive*, pages 47–56, 2005.

[Joh67] T. Johnson. *An algorithm for the resource-constrained project scheduling problem.* PhD thesis, M.I.T., Boston, 1967.

[JYW11] Myounghoon Jeon, Jung-Bin Yim, and Bruce N. Walker. An Angry Driver Is Not the Same As a Fearful Driver: Effects of Specific Negative Emotions on Risk Perception, Driving Performance, and Workload. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2011)*, pages 137–140, 2011.

[KJF07] Jürgen Kuster, Dietmar Jannach, and Gerhard Friedrich. Handling Alternative Activities in Resource-Constrained Project Scheduling Problems. In *Proc of 20th Intern. Joint Conference on AI (IJCAI-07)*, pages 1960–1965, 2007.

[KKE03] Christian Kray, Antonio Krüger, and Christoph Endres. Some Issues on Presentations in Intelligent Environments. In Emile H. L. Aarts, René Collier, Evert van Loenen, and Boris E. R. de Ruyter, editors, *Ambient Intelligence, First European Symposium, EUSAI 2003, Veldhoven, The Netherlands, November 3.-4, 2003, Proceedings*, volume 2875 of *Lecture Notes in Computer Science*, pages 15–26. Springer, November 2003.

[KMG+06] Roberta L. Klatzky, James R. Marston, Nicholas A. Giudice, Reginald G. Golledge, and Jack M. Loomis. Cognitive load of navigating without vision when guided by virtual sound versus spatial language. *Journal of Experimental Psychology: Applied*, 12(4):223, 2006.

[KMPH11] Andrew L. Kun, Zeljko Medenica, Oskar Palinko, and Peter A. Heeman. Utilizing Pupil Diameter to Estimate Cognitive Load Changes During Human Dialogue: A Preliminary Study. In *Adjunct Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2011)*, 2011.

[Kos04] Timo Kosch. Local Danger Warning based on Vehicle Ad-hoc Networks: Prototype and Simulation. In *1st International Workshop Intelligent Transportation (WIT 2004)*, 2004.

[KPR12] Andrew L. Kun, Oskar Palinko, and Ivan Razumenić. Exploring the Effects of Size and Luminance of Visual Targets on the Pupillary Light Reflex. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)*, pages 183–186, Portsmouth, NH, USA, October 1719 2012.

[Kri10] Hans-Ulrich Krieger. A General Methodology for Equipping Ontologies With Time. In *Proceedings of the 7th international conference on Language Resources and Evaluation (LREC'10), ELRA*, 2010.

[KSB87] Arthur F. Kramer, Erik J. Sirevaag, and Rolf Braune. A psychophysiological assessment of operator workload during simulated flight missions. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 29(2):145–160, 1987.

[Kun11] Björn Kunz. A Combined Uniform and Heuristic Search Algorithm for Finding Shortest Paths in Unknown Highly Dynamic Graphs . Master's thesis, Saarland University, 2011.

[LB10] John K. Lenneman and Richard W. Backs. Enhancing assessment of in-vehicle technology attention demands with cardiac measures. In *Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 20–21. ACM, 2010.

[LD97] Thomas K. Landauer and Susan T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review; Psychological Review*, 104(2):211, 1997.

[LEMN12] The Nhan Luong, Patrick Etcheverry, Christophe Marquesuzaà, and Thierry Nodenot. A visual programming language for designing interactions embedded in web-based geographic applications. In *Pro-*

ceedings of the 2012 ACM international conference on Intelligent User Interfaces, IUI '12, pages 207–216, New York, NY, USA, 2012. ACM.

[Lev08] Roger Levy. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177, 2008.

[Lie10] Cornelia Liegl. Evaluation multimodaler virtueller Agenten im Bereich E-Learning mit dem Beispiel eines interaktiven Sprachlernsystems. Master's thesis, Saarland University, 2010.

[LKLS99] Dave Lamble, Tatu Kauranen, Matti Laakso, and Heikki Summala. Cognitive load and detection thresholds in car following situations: safety implications for using mobile (cellular) telephones while driving. *Accident Analysis & Prevention*, 31(6):617–623, 1999.

[LN08] László Laufer and Bottyán Németh. Predicting user action from skin conductance. In *Proceedings of the 13th international conference on Intelligent user interfaces*, IUI '08, pages 357–360, New York, NY, USA, 2008. ACM.

[LPTD09] Wilburta Q. Lindh, Marilyn S. Pooler, Carol D. Tamparo, and Barbara M. Dahl. *Delmar's Comprehensive Medical Assisting: Administrative and Clinical Competencies*. Delmar's Comprehensive Medical Assisting. Delmar Cengage Learning, 2009.

[Mat03] Stefan Mattes. The Lane-Change-Task as a Tool for Driver Distraction Evaluation. In *Quality of Work and Products in Enterprises of the Future (Proceedings of the GfA/17th Annual Conference of the International Society for Occupational Ergonomics ans Safety, ISOES)*, pages 57–60, 2003.

[Mau11] Michael Maurer. Rule-based Presentation Management for Car2X-based Warnings and Infotainment. Master's thesis, Saarland University, 2011.

[MCT+10] Angela Mahr, Yujia Cao, Mariet Theune, Tim Schwartz, and Christian Müller. What if it Suddenly Fails? Behavioural Aspects of Advanced Driver Assistant Systems on the Example of Local Danger Alerts. In *Proc. 19th Europ. Conf. on AI (ECAI 2010)*, pages 1051–1052, Lisbon, August 2010 2010. IOS Press, Amsterdam.

[ME09] Atif Mehmood and Said M. Easa. Modeling reaction time in car-following behaviour based on human factors. *International Journal of Applied Science, Engineering and Technology*, 5(14):93–101, 2009.

[MESM11] Angela Mahr, Christoph Endres, Tanja Schneeberger, and Christian Müller. Determining Human-Centered Parameters of Ergonomic Micro-Gesture Interaction for Drivers Using the Theater Approach. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2011)*, pages 151–157, Salzburg, Austria, November 2011.

[MF08] Martin Molina and Victor Flores. A presentation model for multimedia summaries of behavior. In *Proceedings of the 13th international conference on Intelligent user interfaces*, IUI '08, pages 369–372, New York, NY, USA, 2008. ACM.

[MFM12] Mohammad Mehdi Moniri, Michael Feld, and Christian Müller. Personalized in-vehicle information systems: Building an application infrastructure for smart cars in smart spaces. In *Proceedings of the 8th International Conference on Intelligent Environments IE'12*, june 2012. (Best Video/Demo Award).

[MFMM12] Angela Mahr, Michael Feld, Mohammad Mehdi Moniri, and Rafael Math. The ConTRe (Continuous Tracking and Reaction) Task: A Flexible Approach for Assessing Driver Cognitive Workload with High Sensitivity. In *Adjunct Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)*, pages 88–91, Portsmouth, NH, USA, October 1719 2012.

[MGJ+01] Christian Müller, Barbara Großmann-Hutter, Anthony Jameson, Ralf Rummer, and Frank Wittig. Recognizing Time Pressure and Cognitive Load on the Basis of Speech: An Experimental Study. In *UM2001, User Modeling: Proceedings of the Eighth International Conference*, pages 24 – 33, New York - Berlin, 2001. Springer.

[MK12] Zeljko Medenica and Andrew L. Kun. Data Synchronization for Cognitive Load Estimation in Driving Simulator-based Experiments. In *Adjunct Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)*, pages 92–94, Portsmouth, NH, USA, October 1719 2012.

[MM12] Mohammad-Mehdi Moniri and Christian Müller. Multimodal reference resolution for mobile spatial interaction in urban environments. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, October 2012.

[MMFM07] Katrin Meinken, Roberto Montanari, Mark Fowkes, and Anny Mousadakou. Watch-over HMI for vulnerable road users' protection. In *UAHCI'07: Proceedings of the 4th international conference on Universal access in human-computer interaction*, pages 488–496, Berlin, Heidelberg, 2007. Springer-Verlag.

[MMG02] Jason P. Mitchell, C. Neil Macrae, and Iain D. Gilchrist. Working memory and the suppression of reflexive saccades. *Journal of Cognitive Neuroscience*, 14(1):95–103, 2002.

[MMMM12] Rafael Math, Angela Mahr, Mohammad M. Moniri, and Christian Müller. OpenDS: A new open-source driving simulator for research. In *Adjunct Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)*, pages 7–8, Portsmouth, NH, USA, October 1719 2012.

[Mon11] Mohammad-Mehdi Moniri. Multimodal reference resolution for mobile spatial interaction in urban environments. Master's thesis, Computer Science Institute, University of the Saarland, Germany, 2011.

[Mül00] Jochen Müller. *Persona: Ein anthropomorpher Präsentationsagent für Internet-Anwendungen*. PhD thesis, Saarland University, 2000.

[Mül05] Jörg Müller. Beanspruchungsschätzung im Automobil mit Bayes'schen Netzen. Master's thesis, Saarland University, 2005.

[Mur08] Mark L. Murphy. *The Busy Coder's Guide to Android Development*. CommonsWare, LLC., 2008.

[Nåb07] Arne Nåbo. Driver warnings in time and safety critical situations. Technical report, IVSS Project Report, Transport Research Laboratory, Wokingham, UK, 2007.

[OCC90] Andrew Ortony, Gerald L. Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge Univ Press, 1990.

[OH93] Russel Ovans and William S. Havens. Intelligent mediation: an architecture for the real-time allocation of interface resources. In *Proceedings of the 1st international conference on Intelligent user interfaces*, IUI '93, pages 55–61, New York, NY, USA, 1993. ACM.

[PE02] Angel Puerta and Jacob Eisenstein. Ximl: a common representation for interaction data. In *Proceedings of the 7th international conference on Intelligent user interfaces*, IUI '02, pages 214–215, New York, NY, USA, 2002. ACM.

[PKSH10] Oskar Palinko, Andrew L. Kun, Alexander Shyrokov, and Peter Heeman. Estimating cognitive load using remote eye tracking in a driving simulator. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, pages 141–144, New York, NY, USA, 2010. ACM.

[PR76] James H. Patterson and G.W. Roth. Scheduling a project under multiple resource constraints: a zero-one programming approach. *AIIE Transactions*, 8:449–455, 1976.

[PWW69] A. Alan B. Pritsker, Lawrence J. Watters, and Philip M. Wolfe. Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, 16:93–108, 1969.

[RAB$^+$03] Norbert Reithinger, Jan Alexandersson, Tilman Becker, Anselm Blocher, Ralf Engel, Markus Löckelt, Jochen Müller, Norbert Pfleger, Peter Poller, Michael Streit, and Valentin Tschernomas. Smartkom: adaptive and flexible multimodal access to multiple applications. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 101–108. ACM, 2003.

[Rag01] Dave Raggett. Getting started with VoiceXML 2.0. http://www.w3.org/Voice/Guide/, 2001.

[RAM97] Thomas Rist, Elisabeth André, and Jochen Müller. Adding animated presentation agents to the interface. In *Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 79–86. ACM, 1997.

[RBE$^+$05] Norbert Reithinger, Simon Bergweiler, Ralf Engel, Gerd Herzog, Norbert Pfleger, Massimo Romanelli, and Daniel Sonntag. A look under the hood: design and development of the first smartweb system demonstrator. In *Proceedings of the 7th international conference on Multimodal interfaces*, pages 159–166. ACM, 2005.

[RDMP04] Susana Rubio, Eva Díaz, Jesús Martín, and José M. Puente. Evaluation of Subjective Mental Workload: A Comparison of SWAT, NASA-TLX, and Workload Profile Methods. *Applied Psychology*, 53(1):61–86, 2004.

[RFA09] Andreas Riener, Alois Ferscha, and Mohamed Aly. Heart on the road: HRV analysis for monitoring a driver's affective state. In *Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '09, pages 99–106, New York, NY, USA, 2009. ACM, ACM.

[RHVO99] Lloyd Rutledge, Lynda Hardman, and Jacco Van Ossenbruggen. The use of smil: Multimedia research currently applied on a global scale. In *Proceedings of Multimedia Modeling*, volume 99, pages 1–17. Citeseer, 1999.

[RLH11] Barbara Rosario, Kent Lyons, and Jennifer Healey. A dynamic content summarization system foropportunistic driver infotainment. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications. International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI-11), November 29 - December 2, Salzburg, Austria*, pages 95–98, 2011.

[RMC$^+$09] Bryan Reimer, Bruce Mehler, Joseph F. Coughlin, Kathryn M. Godfrey, and Chuanzhong Tan. An on-road assessment of the impact of cognitive workload on physiological arousal in young adult drivers. In *Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '09, pages 115–118, New York, NY, USA, 2009. ACM.

[RMW$^+$12] Bryan Reimer, Bruce Mehler, Ying Wang, Alea Mehler, Hale McAnulty, Erin Mckissick, Joseph F. Coughlin, Steve Matteson, Vladimir Levantovsky, David Gould, Nadine Chahine, and Geoff Greve. An Exploratory Study on the Impact of Typeface Design in a Text Rich User Interface on Off-Road Glance Behavior. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)*, pages 25–32, Portsmouth, NH, USA, October 1719 2012.

[SCE10] Tim Schwartz, Sandro Castronovo, and Christoph Endres. A GPS-less Method to Find Your Parked Car. In *Adjunct proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2010)*, page 26, Pittsburgh, PA, USA, November 2010.

[Sch93] Robert E. Schlegel. Driver mental workload. In Brian Peacock and Waldemar Karwowski, editors, *Automotive ergonomics*, chapter 17, pages 359–382. Taylor and Francis London, 1993.

[SDK78] J. P. Stinson, E. W. Davis, and B. M. Khumawala. Multiple resource-constrained scheduling using branch and bound. *IIE Transactions*, 10:252–259, 1978.

[SEH+07] Daniel Sonntag, Ralf Engel, Gerd Herzog, Alexander Pfalzgraf, Norbert Pfleger, Massimo Romanelli, and Norbert Reithinger. Smartweb handheldmultimodal interaction with ontological knowledge bases and semantic web services. *Artifical Intelligence for Human Computing*, pages 272–295, 2007.

[SGC+08] Marc Schröder, Patrick Gebhard, Marcela Charfuelan, Christoph Endres, Michael Kipp, Sathish Chandra Pammi, Martin Rumpler, and Oytun Türk. Enhancing Animated Agents in an Instrumented Poker Game. In Andreas Dengel, K. Berns, Thomas Breuel, F. Bomarius, and Thomas Roth-Berghofer, editors, *KI 2008: Advances in Artificial Intelligence. September 23-26, Kaiserslautern, Germany*, volume 5243 of *Lecture Notes in Artificial Intelligence, LNAI*, pages 316–323. Springer, 2008.

[SH95] Kip Smith and P. A. Hancock. The risk space representation of commercial airspace. In *Proceedings of the 8th International Symposium on Aviation Psychology*, 1995.

[SKR+10] Kristian Sons, Felix Klein, Dmitri Rubinstein, Sergiy Byelozyorov, and Philipp Slusallek. Xml3d: interactive 3d graphics for the web. In *Web3D '10: Proceedings of the 15th International Conference on Web 3D Technology*, pages 175–184, New York, NY, USA, 2010. ACM.

[SMR+02] H. Storm, K. Myre, M. Rostrup, O. Stokland, MD Lien, and JC Raeder. Skin conductance correlates with perioperative stress. *Acta anaesthesiologica scandinavica*, 46(7):887–895, 2002.

[Son10] Daniel Sonntag. *Ontologies and Adaptivity in Dialogue for Question Answering*. AKA and IOS Press, Heidelberg, 2010.

[SP11] Joonwoo Son and Myoungouk Park. Estimating Cognitive Load Complexity Using Performance and Physiological Data in a Driving Simulator. In *Adjunct Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2011)*, 2011.

[SPBS95] E. Salas, C. Prince, D.P. Baker, and L. Shrestha. Situation awareness in team performance: Implications for measurement and training. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):123–136, 1995.

[SPKS11] Stefan Schneegaß, Bastian Pfleging, Dagmar Kern, and Albrecht Schmidt. Support for modeling interaction with automotive user interfaces. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications. International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI-11), November 29 - December 2, Salzburg, Austria*, pages 71–78, 2011.

[Sta04] Mark A. Staal. Stress, cognition, and human performance: A literature review and conceptual framework. *NASA technical memorandum*, 212824, 2004.

[THS+12] Udo Trutschel, Christian Heinze, Bill Sirois, Martin Golz, David Sommer, and David Edwards. Heart Rate Measures Reflect the Interaction of Low Mental Workload and Fatigue During Driving Simulation. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)*, pages 261–264, Portsmouth, NH, USA, October 1719 2012.

[TWV11] Patrick Tchankue, Janet Wesson, and Dieter Vogts. The Impact of an Adaptive User Interface on Reducing Driver Distraction. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2011)*, pages 87–94, 2011.

[VAB11] Yolanda Vazquez-Alvarez and Stephen A. Brewster. Eyes-free multitasking: the effect of cognitive load on mobile spatial audio interfaces. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 2173–2176. ACM, 2011.

[VDVM94] Michael Vidulich, Cynthia Dominguez, Eric Vogel, and Grant McMillan. Situation awareness: Papers and annotated bibliography. Technical report, DTIC Document, 1994.

[vM99] Susanne van Mulken. *User Modeling for Multimedia Interfaces*. PhD thesis, Saarland University, 1999.

[VSLR08] Carlos Vaquero, Oscar Saz, Eduardo Lleida, and W.-Ricardo Rodríguez. E-inclusion technologies for the speech handicapped. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 4509–4512. IEEE, 2008.

[VWS91] Michael A. Vidulich, G. Frederic Ward, and James Schueren. Using the subjective workload dominance (SWORD) technique for projective workload assessment. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 33(6):677–691, 1991.

[WAB+92] Wahlster Wahlster, Elisabeth André, Sanghamitra Bandyopadhyay, Winfried Graf, and Thomas Rist. Wip: The coordinated generation of multimodal presentations from a common representation. *Communication from an Artificial Intelligence Perspective*, pages 121–143, 1992.

[Wah92] Wolfgang Wahlster. Automatic Design of Multimodal Presentations. In T. Catarci, S. Levialdi, and M. Costabile, editors, *Advanced Visual Interfaces (Avi'92): Proceedings of the International Workshop*, pages 243–257. World Scientific Publishing Co., Inc., 1992.

[Wah00] Wolfgang Wahlster. *Verbmobil: foundations of speech-to-speech translation*. Springer Verlag, 2000.

[Wah03] Wolfgang Wahlster. Smartkom: Symmetric multimodality in an adaptive and reusable dialogue shell. In *Proceedings of the Human Computer Interaction Status Conference*, volume 3, pages 47–62. Berlin, Germany: DLR, 2003.

[Wah04] Wolfgang Wahlster. Smartweb: Mobile applications of the semantic web. *KI 2004: Advances in Artificial Intelligence*, pages 50–51, 2004.

[Wah06] Wolfgang Wahlster. *SmartKom: Foundations of Multimodal Dialogue Systems (Cognitive Technologies)*. Springer-Verlag New York, Inc., 2006.

[Wah07] Wolfgang Wahlster. Ein multimodales dialogsystem für das semantische web. *Informatikforschung in Deutschland, Heidelberg, Berlin, Springer*, 2007.

[Wei91] Mark Weiser. The computer for the 21st century. *Mobile Computing and Communications Review*, 1991.

[Wes99] Robert West. Visual distraction, working memory, and aging. *Memory & Cognition*, 27(6):1064–1072, 1999.

[WH99] Christopher D. Wickens and Justin G. Hollands. *Engineering Psychology and Human Performance*. Prentice Hall New Jersey, third edition, 1999.

[Wic02] Christopher D. Wickens. Multiple resources and performance prediction. *Theoretical issues in ergonomics science*, 3(2):159–177, 2002.

[WK99] Ewa Wojciulik and Nancy Kanwisher. The generality of parietal involvement in visual attention. *Neuron*, 23(4):747–764, 1999.

[WKE03] Rainer Wasinger, Christian Kray, and Christoph Endres. Controlling multiple devices. In *Workshop on Physical Interaction at Mobile HCI 2003*, Udine, Italy, September 2003.

[WRB01] Wolfgang Wahlster, Norbert Reithinger, and Anselm Blocher. Smartkom: Multimodal communication with a life-like character. In *Seventh European Conference on Speech Communication and Technology*, 2001.

[WSV83] Christopher D. Wickens, Diane L. Sandry, and Michael Vidulich. Compatibility and resource competition between modalities of input, central processing, and output. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 25(2):227–248, 1983.

[WT97] Wolfgang Wahlster and Werner Tack. SFB 378: Ressourcenadaptive Kognitive Prozesse. In Matthias Jarke, Klaus Pasedach, and Klaus Pohl, editors, *Informatik'97: Informatik als Innovationsmotor: 27. Jahrestagung der Gesellschaft für Informatik, Aachen, 24.-26. September 1997*, pages 51–57. Springer, 1997.

[YD08] Robert M. Yerkes and John D. Dodson. The relation of strength of stimulus to rapidity of habit-formation. *Journal of comparative neurology and psychology*, 18(5):459–482, 1908.

[YRM+12] Yan Yang, Bryan Reimer, Bruce Mehler, Alan Wong, and Mike McDonald. Exploring Differences in the Impact of Auditory and Visual Demands on Driver Behavior. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)*, pages 173–177, Portsmouth, NH, USA, October 1719 2012.

[Ysa66] Lars Ysander. The safety of physically disabled drivers. *British Journal of Industrial Medicine*, 23(3):173–180, 1966.

[ZHM+11] Jürgen Ziegler, Tim Hussein, Daniel Münter, Jens Hofmann, and Timm Linder. Generating route instructions with varying levels of detail. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications. International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI-11), November 29 - December 2, Salzburg, Austria*, pages 31–38. ACM, 2011.

[Zil96] Shlomo Zilberstein. Using anytime algorithms in intelligent systems. *AI magazine*, 17(3):73, 1996.