LMU

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Visual Analysis of In-Car Communication Networks

Michael Sedlmair

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt von
Michael Sedlmair
aus München

München, den 16. Juli 2010

Erstgutachter: Prof. Dr.-Ing. Andreas Butz
Zweitgutachter: Prof. Dr. Tamara Munzner
Tag der mündlichen Prüfung: 21. Oktober 2010

Abstract

Analyzing, understanding and working with complex systems and large datasets has become a familiar challenge in the information era. The explosion of data worldwide affects nearly every part of society, particularly the science, engineering, health, and financial domains. Looking, for instance at the automotive industry, engineers are confronted with the enormously increased complexity of vehicle electronics. Over the years, a large number of advanced functions, such as ACC (adaptive cruise control), rear seat entertainment systems or automatic start/stop engines, has been integrated into the vehicle. Thereby, the functions have been more and more distributed over the vehicle, leading to the introduction of several communication networks. Overlooking all relevant data facets, understanding dependencies, analyzing the flow of messages and tracking down problems in these networks has become a major challenge for automotive engineers.

Promising approaches to overcome information overload and to provide insight into complex data are Information Visualization (InfoVis) and Visual Analytics (VA). Over the last decades, these research communities spent much effort on developing new methods to help users obtain insight into complex data. However, few of these solutions have yet reached end users, and moving research into practice remains one of the great challenges in visual data analysis. This situation is particularly true for large company settings, where very little is known about additional challenges, obstacles and requirements in InfoVis/VA development and evaluation. Users have to be better integrated into our research processes in terms of adequate requirements analysis, understanding practices and challenges, developing well-directed, user-centered technologies and evaluating their value within a realistic context.

This dissertation explores a novel InfoVis/VA application area, namely in-car communication networks, and demonstrates how information visualization methods and techniques can help engineers to work with and better understand these networks. Based on a three-year internship with a large automotive company and the close cooperation with domain experts, I grounded a profound understanding of specific challenges, requirements and obstacles for InfoVis/VA application in this area and learned that “designing with not for the people” is highly important for successful solutions. The three main contributions of this dissertation are: (1) An empirical analysis of current working practices of automotive engineers and the derivation of specific design requirements for InfoVis/VA tools; (2) the successful application and evaluation of nine prototypes, including the deployment of five systems; and (3) based on the three-year experience, a set of recommendations for developing and evaluating InfoVis systems in large company settings.

I present ethnographic studies with more than 150 automotive engineers. These studies helped us to understand currently used tools, the underlying data, tasks as well as user groups and to categorize the field into application sub-domains. Based on these findings, we propose implications and recommendations for designing tools to support current practices of automotive network engineers with InfoVis/VA technologies. I also present nine InfoVis design studies that we built and evaluated with automotive domain experts and use them to systematically explore the design space of applying InfoVis to in-car communication networks. Each prototype was developed in a user-centered, participatory process, respectively with a focus on a specific sub-domain of target users with specific data and tasks. Experimental results from studies with real users are presented, that show that the visualization prototypes can improve the engineers' work in terms of working efficiency, better understanding and novel insights. Based on lessons learned from repeatedly designing and evaluating our tools together with domain experts at a large automotive company, I discuss challenges and present recommendations for deploying and evaluating VA/InfoVis tools in large company settings. I hope that these recommendations can guide other InfoVis researchers and practitioners in similar projects by providing them with new insights, such as the necessity for close integration with current tools and given processes, distributed knowledge and high degree of specialization, and the importance of addressing prevailing mental models and time restrictions. In general, I think that large company settings are a promising and fruitful field for novel InfoVis applications and expect our recommendations to be useful tools for other researchers and tool designers.

Zusammenfassung

Die weltweit ständig steigende Verfügbarkeit großer Datenmenge hat in den letzten Jahren zu immer neuen Herausforderungen in den Bereichen der Wissenschaft, aber auch im Ingenieurs-, Gesundheits- und Finanzwesen geführt. Beispielsweise sind Fahrzeugentwickler in der Automobilbranche mit einer immensen Datenkomplexität konfrontiert, welche durch die Integration immer neuer elektronischer Komponenten hervorgerufen wird. So enthalten moderne Fahrzeuge bis zu 100 verschiedene Steuergeräte, die miteinander in einem sogenannten Bordnetz vernetz sind und auf unterschiedlichen Wegen miteinander kommunizieren. Im Zuge dieser Entwicklung sind herkömmliche textuelle Beschreibungen der Kommunikationsarchitektur immer mehr an ihre Grenzen gestoßen, so dass aktuell viel Zeit und Erfahrung benötigt wird um beispielsweise Auswirkungen von Änderungen an einzelnen Komponenten auf die Gesamtarchitektur abzuschätzen, um mögliche Engpässe zu erkennen oder um Fehler in der Bordnetzkommunikation zu finden.

Methoden der *Informationsvisualisierung* und der *Visuellen Analyse* versprechen hierbei mögliche Ansätze um diese Herausforderungen zu bewältigen. In beiden Forschungsbereichen wurden in den vergangenen Jahren eine Vielzahl von neuartigen Visualisierungsmethoden und -techniken präsentiert, mit denen es möglich ist Verständnis über komplexe und große Datensätze zu erlangen. Aktuell sind jedoch noch wenige dieser Lösungen in praktische Anwendungen integriert worden und eine der größten Herausforderungen ist immer noch diese Forschungsergebnisse in die Praxis zu transferieren. Dies gilt vor allem für große Firmen, bei denen spezielle Anforderungen zusätzliche Hindernisse und Herausforderungen an die Entwicklung und Evaluierung von Visualisierungstools stellen.

Im Rahmen dieser Dissertation wird untersucht wie Methoden der Informationsvisualisierung dazu beitragen können die Entwicklung und Analyse von automobilen Bordnetzen zu unterstützen und wie diese in die Praxis integriert werden können. Die Arbeit wurde in enger Kooperation mit BMW, einem großen Automobilhersteller, durchgeführt und bot daher die Möglichkeit Visualisierungsanwendung unter realen Bedingungen zu entwickeln und zu testen. Es wurde klar, dass vor allem eine konstante und enge Zusammenarbeit mit den Endanwender, d. h. mit Entwicklungs- und Testingenieuren, eine zentrale Rolle für den Erfolg von Visualisierungsanwendung in diesem Bereich spielt.

Der wissenschaftliche Beitrag der vorliegenden Arbeit kann in drei Aspekte unterteilt werden: (1) Eine empirische Feldanalyse des Bereichs Bordnetzentwicklung und -absicherung, sowie Designanforderungen, die daraus abgeleitet wurde; (2) die Vorstellung und Evaluierung von neun Visualisierungsprototypen, von denen fünf in die aktuelle Arbeitsprozesse der Ingenieure integriert wurden; und (3) basierend auf den Erfahrungen, die über drei Jahre gesammelt wurden, eine Zusammenfassung von Herausforderungen und Empfehlungen zur Durchführung von Informationsvisualisierungsprojekten in großen Unternehmen.

In der Arbeit werden hierzu ethnografische Studien mit mehr als 150 Ingenieuren vorgestellt. Diese Studien wurden dazu genutzt aktuelle Werkzeuge besser zu verstehen, Arbeitsweisen zu analysieren aber auch um Nutzergruppen zu kategorisieren und Herausforderungen zu erkennen. Basierend auf diesen Studien wurden anwendungsgebiet-spezifische Anforderungen abgeleitet und als Grundlagen für die Entwicklung der Prototypen verwendet. Mit den Prototypen wurden dann schrittweise verschiedene Detailaspekte des Anwendungsgebietes untersucht. Hierbei wurden insbesondere Lösungen zur Unterstützung von Netzwerkarchitekten, Visualisierungsmethoden zur Trace-Analyse sowie die Verwendung von 3d Modellen für ein besseres Verständnis von Zusammenhängen von Elektronik und Mechanik untersucht. Die Tools wurden in einem benutzer-zentrierten Ansatz entwickelt und mit Endanwender evaluiert um Erkenntnisse über deren realen Einsatz zu erlangen. Die Ergebnisse zeigen, dass die vorgestellten Tools aktuelle Arbeitsprozesse beschleunigen konnten und neues und hilfreiches Verständnis über die Daten generiert wurde.

Während der drei Jahre, in denen die Prototypen entstanden sind, wurden viele Erfahrungen über die Entwickelung und Evaluierung von Visualisierungstools in einem großen Unternehmen gesammelt. Neben bekannten Designkriterien aus der Literatur gilt es in solchen Umgebungen darauf zu achten ein Tool in die vorhandenen Prozessketten zu integrieren und speziellen Anforderungen—wie beispielsweise verteiltes Wissen, hoher Spezialisierungsgrad, Zeitknappheit oder bereits vorhandene Tools—Rechnung zu tragen. Diese Kriterien wurden zusammengefasst um anderen Designern und Forschern bei ähnlichen Projekten zu helfen. Generell gilt anzumerken, dass die Erforschung von Visualisierungstools in großen Unternehmen ein sehr gute Gelegenheit bietet Tools zu integrieren und unter realistischen Bedingungen zu evaluieren.

Acknowledgements

I would like to thank all the people who have helped me on my way of doing this PhD and without them the work would not have been possible. As these acknowledgements are the very last part of my PhD left to write, I am starting to realize that it also might be one of the most difficult parts to write. Over the last three years, I collaborated with so many people who supported me in doing this research, who influenced me and my work, and who shaped my thinking and ideas. At this point, I want to thank all of them, including those not mentioned here by name.

First, I want to thank Andreas Butz, my supervisor, for opening my eyes in doing research, for being open to all my questions at every time, and especially for providing me with the great opportunity to visit conferences even in times of recession. I also want to thank Wolfgang Hintermaier who supervised my work at BMW for introducing me to a variety of people at the company and for having always valuable tips and tricks for doing research in this setting.

Special thanks also go to my committee members Tamara Munzner, Heinrich Hußmann, Hans Jürgen Ohlbach and Rolf Hennicker for their time and energy. In particular, I would like to thank Tamara for helping me to improve this document by providing excellent feedback.

Furthermore, I would like to thank all my university and research colleagues, especially Petra Isenberg, Dominikus Baur and Sebastian Boring, for many discussions, valuable feedback and publishing our ideas together. And, of course, to the number of students who have contributed to this work via their project and diploma theses: Annika Frank, Christian Bernhold, Korbinian Zollner, Ben Kunze, Wolfgang Jacobi, Alexandre Dürr, Michael Mauerer, Alex Messner, Fabian Hennecke, Alexander Kahl and Martin Knobel. They all investigated much more work than would have been required, some of them even co-authored conference papers, and I really enjoyed working with and supervising all of them.

Besides, I want to particularly mention Michael Schraut and Richard Bogenberger in whose research groups I worked at BMW, and all my BMW colleagues who supported me in many ways. The atmosphere has been excellent at every time and I always enjoyed working in this great environment. Of course, my thanks also go to the *BMW Group Research and Technology* in general for funding the work and, in doing so, to make this thesis possible.

I also want to thank all the automotive engineers participating in my studies (over 150!) for their valuable time and energy. While it is not possible to name each single person here, I want to emphasize some of them who were very closely involved in designing and building the tools presented in this thesis and who always took time for our research—even in situations where their time was scarce: Robert Schuster, Christian Pigorsch, Andreas Greiner, Thomas Würdinger, Jürgen Hoffmann, Daniel Herrscher, Florian Kuttig, Ben Krebs, and Martin Schlaugath.

Last but not least, I also want to give a very special thanks to my family and all my friends for their patience and support. My parents helped me with so many things and always understood when I had to work even during visits at home. Finally, I want to thank my wonderful girlfriend Hanna for her support and love through all stages that a PhD brings with it, for understanding stressful times, and for always managing to encourage me.

Contents

Chapter 1

Introduction

Due to enormous progresses in computer power and storage capacity over the last decades, our ability to collect data for engineering, scientific or commercial purposes has immensely increased. However, while storing (nearly) all available information without any restrictions is possible now, analyzing the huge amounts of data and deriving valuable information hidden within it poses great challenges. One approach to cope with this information overload challenge is Information Visualization (InfoVis) or Visual Analytics (VA) where researchers focus on the question how visual data representations can help people to better understand and gain insights into these large and complex datasets. The major objective of InfoVis/VA is amplifying cognition and not just—as misleadingly believed—drawing nice graphics. According to Card's famous InfoVis definition from 1999

> "[InfoVis is] the use of computer-supported, interactive, visual representations of abstract data to amplify cognition." [CMS99]

Over the last two decades, InfoVis researchers have shown that visual data representations have several benefits over exploring information in textual form. Visual displays can, for instance, help to reduce search time, enhance pattern detection, support the understanding of complex correlations and support hypothesis forming, evaluation and confirmation [CMS99, War04].

While InfoVis focuses more on interactive, visual representation itself, VA has a broader perspective on the entire analytics process. According to Keim's definition

> "Visual analytics is more than just visualization and can rather be seen as an integrated approach combining visualization, human factors and data analysis. [...] With respect to the field of visualization, visual analytics integrates methodology from information analytics, geospatial analytics, and scientific analytics. Especially human factors (e. g., interaction, cognition, perception, collaboration, presentation, and dissemination) play a key role in the communication between human and computer, as well as in the decisionmaking process." [KMSZ06]

These research communities have spent much effort on developing new methods to help users obtain insights into complex data. However, by observing the current situation in industrial settings—as I did for three years at a large automotive company—it becomes clear that only few of these solutions have yet reached end users and 'moving research into practice' remains one of the great future InfoVis/VA challenges [TC05].

In 2004, Plaisant noted that

> "We are still a long way from understanding how our research tools become products. Gathering struggle and success stories from research and product teams should help our community make more rapid progress." [Pla04]

According to my experience, this situation holds unchanged even six years later and is probably the strongest motivation behind the work in this thesis. To contribute to a better understanding of how we can apply InfoVis/VA technologies to solve real world problems, I concentrated on a specific and so far untreated real world use case and investigated how our tools can help automotive engineers to better understand complex in-car communication networks and what challenges we are confronted with when we want to apply and evaluate our technologies in large companies. I closely collaborated with engineers over three years in a large automotive company, learned about their problems and challenges in developing, understanding, and testing in-car communication networks, built a variety of InfoVis/VA prototypes and evaluated them in a real world context (nine of these design studies are presented in this thesis).

In this chapter, I motivate my thesis by introducing the challenges of developing and analyzing in-car communication networks, point out the scope and objectives within the InfoVis/VA research community and explain my methodological approach. After a brief overview of my research contributions and design studies, I outline the dissertation's entire structure at the end of this chapter.

1.1 Motivation: Complexity of In-Car Communication Networks

The amount of electronics and software in automobiles has increased enormously over the last years and has posed a variety of novel challenges to automotive engineers [Bro06, Gri03, Hei05, PBKS07].

More and more advanced functions such as airbag systems, adaptive cruise control, rear seat entertainment systems or automatic start/stop engines have been integrated with the car to enable safer, more enjoyable and efficient driving. This constant integration of new systems, however, also led to an enormously increased complexity of the vehicle's electronics.

For example, in order to determine whether an airbag should be triggered in a car, accelerometers send information to a microprocessor at 10 millisecond intervals and the evaluation of this

Figure 1.1: In-car communication network with ECUs, sensors and actuators.

data determines whether and how to inflate an airbag [BG07b]. In order to provide passengers safety, this process underlies strict real time regulations and has to be carefully implemented and tested by automotive engineers. Another prominent example is the adaptive cruise control system (ACC) which allows automatically adjusting the car's speed according to distance and speed of a vehicle ahead. Speed and position of the vehicle ahead are gathered by a radar sensor and are sent to a distant Electronic Control Unit (ECU). This ECU computes specific parameters such as acceleration or braking values and distributes them to several other systems such as engine management, the instrument cluster or to the ESC (Electronic Stability Control) system [BGB02]. Similar to these two examples, many novel vehicle functions are based on information combined from multiple sensors and actuators embedded within a highly distributed and cross-linked system in the car [BG07b] (see Figure 1.1).

When specifying such systems, integrating them in the car and testing them, automotive engineers have to take into account a variety of parameters including cross-correlations, appropriate timing of information distribution, handling interruptions or evaluating external conditions. Communication within this system is realized by several bus systems that provide specific technology and different types of bus protocols in order to fulfill varying specifications [LHD99], such as real-time communication (e. g., CAN or Flexray bus systems), bandwidth (e. g., MOST for distributing multimedia content), or pricing (e. g., LIN for a low-price bus system). These bus systems connect up to 100 ECUs on which the communication software is implemented and, in order to transport all relevant information, up to 15,000 messages per second are distributed over this network.

Overseeing all relevant information and testing these networked systems is one of the major challenges for automotive engineers [Bro06]. To get a grounded understanding of all relevant parameters in this network, a variety of data sources have to be taken into account, including network specification documents, information databases, network traces and CAD data.

Especially when analyzing the flow of messages in this network, the challenge of handling all relevant data and information has increased enormously. Based on recorded network trace files with up to 2GB of data and 15 million messages per hour, analysis experts have to track down errors in order to meet the strict specifications of these networks.

Providing effective tools for the development of the networks and the analysis of this data is of highest importance as the safety of the automobile and its passengers hinges on the ability of automotive engineers to understand and debug this sensor data (for a recent examples underlining the relevance of this topic see, for instance, [Aut10]). Current development and analysis tools, however, are based on purely textual representations. In consequence, much time and experience is needed to specify stable and efficient network configurations, to understand the processes and their complex correlations and to detect the sources of errors.

While recent research trends in developing in-car communication networks indicate a reduction of ECUs in these networks, however, at the same time they emphasize that the amount of network communication and cross-correlations in these networks will increase [Hei05]. Technologies such as x-by-wire[1] [ISS02], distribution of computing power over several ECUs [Wür10] or car-to-car/car-to-infrastructure [Kos05] communication will exacerbate this trend and I believe that challenges and problems addressed in this thesis will be of high importance for understanding, developing and analyzing future in-car networks.

1.2 Research Objectives

During my work for this thesis I followed two main objectives. The first, more from an automotive engineer's perspective, was how InfoVis/VA technologies can improve the understanding of and the work with in-car communication networks. The second, more from an InfoVis/VA researcher's perspective, was what we generally can learn by applying InfoVis/VA to this domain and in large company settings in general. In the following, I explain these two objectives in more detail and provide background information and motivation from literature.

Objective 1: Exploring how InfoVis/VA methods and technologies can help to gain insight into and improve the engineering work with in-car communication networks

Over the last decades, many promising techniques and tools have been developed in the InfoVis/VA research communities. Some of them are closely related to the first objective of this work in terms of network visualization (such as Henry and Fekete's Matrix Explorer [HF06]), that, for instance, can provide better insight into correlations and dependencies in network structures, or trace and time-based visualization (such as Pretorius and Wijk's work on trace visualization [PVW08]), which are helpful to analyze message recordings in large communication networks.

---

[1] x-by-wire is the general term to describe the displacement of traditional mechanical/hydraulic coupling by electronic communication, basically for high security functionality such as steering or braking. 'x' acts as a placeholder and generalizes terms as steer-by-wire, brake-by-wire, drive-by-wire, and so on.

Few of these solutions, however, addressed specific aspects, requirements and obstacles in automotive engineering.

Visualization in the automotive domain, on the other hand, is most commonly used in the context of computer-aided-design (CAD), virtual reality, and Scientific Visualization [Ste07]. Within scientific visualization, many techniques have focused on the analysis of physical simulations, such as the flow of particles for car body development [SRBE99]. Such techniques have also been integrated with InfoVis [DMG+04, KMG+07]. However, considerably less work has been dedicated to the support of electronic engineering for car development and testing. The focus of electronic engineering in cars is to ensure that a car's separate computer systems for controlling such things as the engine timing, brakes and the air bags are all properly integrated and functioning. While there exist various complex problems stemming from large and abstract datasets in this domain, InfoVis/VA solutions, however, are still rare.

Objective 2: Gain a better understanding of how to successfully apply, deploy and evaluate InfoVis/VA tools in large company environments.

As a second objective, this thesis more generally focuses on how we can closely integrate our solutions to solve real world problems, with real data, in a large company environment and how we can evaluate the benefits of our solutions in such environments.

Early InfoVis research was often conducted detached from real end-user environments and research cooperations were mainly restricted to colleagues from familiar departments. While these brought valuable insights into general aspects of perceptability, methodologies and a profound understanding of the design space, recently, it became clear that in order to broaden our knowledge about visualization and data analysis applications for real end users, a closer integration of these tools into user's working environments (e. g., in industry) as well as novel evaluation techniques beyond usability studies and controlled experiments are necessary [KHI+03, Pla04, TM04]. Therefore, 'moving research into practice' and 'Evaluating VA technology in the context of its intended use' were named as two grand challenges for future research [JMM+06, TC05]. By looking into current InfoVis and VA research these challenges get more and more attention: InfoVis/VA researchers started to establish close cooperations with various data-intensive application areas, e. g. [KBGE09, TDS08]. End users had been integrated into design processes by adopting user-centered [WKVD+08] and participatory approaches [HF06, SML+09]. Researchers also searched for reasons why InfoVis/VA solutions are not yet adopted widely and suggested solutions how to overcome them. Van Wijk [VW06], for instance, discussed a significant gap between visualization researchers' (find novel method, get a paper accepted) and the domain experts' goals (require a useful tool regardless of its novelty factor) and provided several approaches how to overcome this gap. Further limitations of current approaches are discussed in [AS05, JMM+06, Pla04].

However, success stories of adopted InfoVis/VA solutions are still rare. The underlying problem is that we do not yet completely understand how end users can benefit from our techniques nor do we have adequate evaluation techniques to measure this [Pla04, TM04]. Traditional usability studies fall short in terms of understanding the long-term, exploratory and complex nature of tasks aimed to be supported by our tools, such as detecting complex patterns over weeks or even months [And06, Car08b, Pla04, SP06]. Additionally, it became clear that it is necessary to evaluate VA technology in the context of its intended use. Therefore, researchers started to propose novel, basically ethnographic and insight-based evaluation techniques to better understand end-user efficacy, e. g., [IZCC08, SND05, SP06] and offered help when to use which evaluation technique [Mun09]. However, these works are just a first step into the direction of generally learning about the "real world" value of our tools. To do so, we have to continue the path of developing and evaluating tools within realistic environments [SP06] by establishing new and fruitful end-user cooperations with real driving problems [TC05].

The research described in this thesis was conducted in close cooperation with a large automotive company. Large companies are an important part of the real world—e. g., according to the German Federal Statistical Office in 2008 42% of company employees worked in large companies [Sta08]. These companies have many driving problems, large datasets and complex problems and provide a fruitful endeavour for applying InfoVis/VA. However, we currently know little about additional challenges and obstacles when designing, deploying and evaluating InfoVis/VA tools in such environments. The second goal of this thesis is therefore to identify these challenges and to derive recommendations of conducting InfoVis/VA research in large company settings.

To address these two objectives, I followed the methodological approach as outlined in the following section.

1.3 Research Approach, Process and Scope

The goal to support engineers in their work leads to a problem-driven, empirical research approach with an equal proportion of prototypes and explorative, formative as well as summative user studies. There are two general research approaches in InfoVis/VA: problem-driven and technique-driven (e. g. [Mun08, Mun09]). Problem-driven means that a problem (e. g., drive a nail into wood) is identified and a researcher looks for a proper solution for this problem (e. g., find a hammer to drive the nail into the wood). Technique-driven describes that (typically) there is an intrinsically clear problem, and the researcher tries to find a novel, better solution by focusing on technology (find a novel hammer).

To a large extent, this thesis is problem-driven: There are many real world problems and challenges in designing, understanding and testing in-car communication networks. Yet, it is not clear what exactly these problems (nails) and tasks (drive in) are, which of these problems and tasks can be supported by InfoVis/VA technologies (nails which can be driven by an InfoVis/VA hammer) and how InfoVis/VA solutions have to be designed in order to properly support these domain tasks/problems (find working hammers for these nails).

According to the problem-driven approach and to the objectives stated above, I divided the scope of this work into three stages:

Stage 1: Understand target users, data, tasks, problems and requirements

Understanding a target domain is essential for building valuable tools improving work and solving domain specific problems. Learning the target users' language, identifying real domain challenges and problems, and finding the right people to collaborate are highly important in order to design valuable solutions for a specific target domain [Mun09]. This approach has been successfully applied to other domains (e. g., [BOR09] in CSCW), and it also has become clear that only if these aspects are understood properly, effective visualization applications can be developed [VW06]. The first stage of my research approach was therefore, to get a clear understanding of the underlying data, the engineers' technical background, their current practices, tasks, problems, demands and challenges. Along with a literature review in automotive electronics and networks (see [BG07a] for a good overview), I conducted many ethnographic field studies and collaborated with domain experts in a variety of projects. The general approach to gain a systematic understanding of this field was based on grounded theory [GS77], a socio-scientific approach of combining various qualitative research methodologies in order to form a comprehensive understanding of the subject of study[2]. Conducting focus groups, user observations, contextual inquiries, interviews, questionnaires and informal discussions with more than 150 automotive engineers over a period of three years helped me to get a grounded understanding of this domain, to identify application areas within this domain, to initiate fruitful cooperations, and to derive InfoVis/VA challenges and requirements.

Stage 2: Explore the design space for InfoVis/VA

Based on this field analysis, I drilled down several concrete use cases, important aspects within the data and specific problems, for instance, browsing large network catalogs or tracking down network errors. In doing so, three important application areas were identified, (a) supporting network development engineers in understanding specification data, (b) supporting analysis experts in debugging traces and (c) a more technique-driven area, how virtual 3d models can contribute to a better understanding of in-car communication. Several high-fidelity visualization prototypes for each area were built and subsequently evaluated with domain experts. These prototypes can be seen as design studies for the focused target domain of in-car communication networks. All prototypes of areas (a) and (b) were developed using an iterative, participatory design approach [BKS04, KS98], involving end users, tool developers and decision makers alongside the entire process. The general guideline behind this approach was "designing it with the people not for the people". For these prototypes, I therefore disposed a problem-driven, close collaboration with domain experts with the goal to create solutions that directly can improve their work with in-car communication networks (see objective 1). Several formative (what can we make better?) and summative (What are the benefits?) user studies were conducted for each prototype. For formative evaluations especially heuristic evaluations and think-aloud protocols were used. The summative validation of the design studies focused on realism [McG95, Car08b] and domain value, and various strategies were used. Think-aloud studies provided feedback from various domain experts outlining potential benefits, estimations and drawbacks. Experimental setups al-

---

[2] We basically used the adapted Straussian grounded theory (SGT) for theme analysis and concept generation without an explicit focus on deriving a theory [SC90].

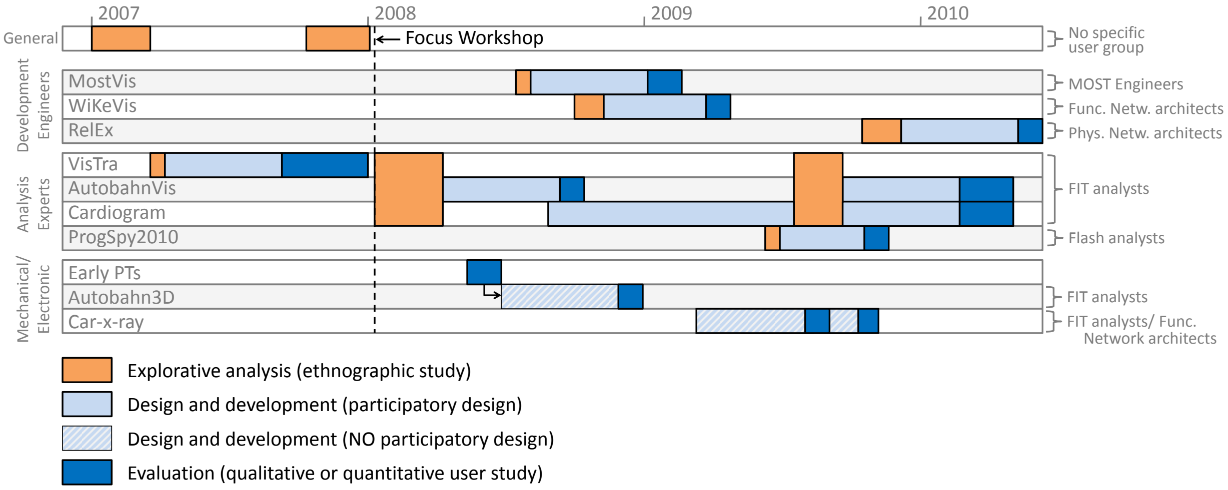


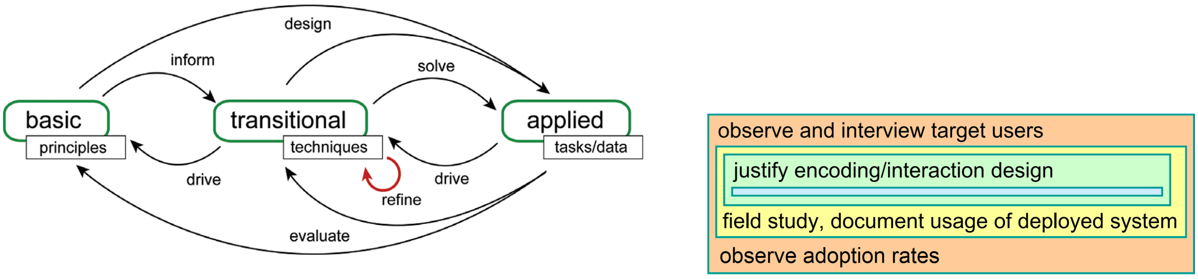
Figure 1.2: Temporal overview over the projects and their different development phases. The top row represents general explorative activities. Each row below represents activities on a specific prototype (optionally) designed for and with a specific user group (right-side notation) and belonging to one of the three application areas focused in this thesis (row clusters, left-side notation). The color coding marks different phases in the prototype development process.

lowed us to compare time and error rates compared to traditional tools. Furthermore, some of the tools were integrated as fully implemented systems into the target users' working environment in order to better address the long-term nature of analysis tasks and to validate their benefits under real circumstances, with real problems and data [SP06].

Stage 3: Generalization: Comprehensive aspects and lessons learned

Based on the exploration of the target domain (stage 1) and the design space exploration of applying InfoVis/VA to the in-car communication network domain (stage 2) the last stage is about taking a step back, investigating more comprehensive aspects and deriving general knowledge. By analyzing the lessons learned from 1 and 2, I was particularly interested in deriving a better understanding about conducting InfoVis/VA research in large company settings and especially about deploying and evaluating our tools in such environments. Summarizing experience and success stories into general challenges and recommendations can help other researchers in similar situations.

In practice, these three stages were not carried out in a strict temporal order, but rather in several iterations according to a prototype's development cycle. These resulted in a step-by-step refinement of hypothesis, domain understanding and lessons learned. Figure 1.2 shows all activities on a horizontal time line from left to right. Stage 1 is influenced by all explorative studies (orange boxes), each prototype of Stage 2 can be found on a specific horizontal line, clustered according to the three identified main application areas (c. f. Stage 2), and Stage 3 is influenced by experience gained over the entire process.



(a) (b)

Figure 1.3: Models for thesis' scope definition: **(a)** Diagram on interaction between basic, transitional and applied research [JMM+06]; and **(b)** Focus of this work in Munzner's Nested Model [Mun09].

In the remainder of this section I relate my work according to two InfoVis/VA models in order to provide the reader with a better understanding of the work's scope and of how it fits into the research domain of InfoVis/VA.

As already stated, this thesis was a highly collaborative work with domain engineers at a large automotive company. According to Johnson et al.'s model of roles in InfoVis collaborations (see [JMM+06] and Figure 1.3-a), this work can be seen as transitional/applied research. The scope of this thesis is not research in basic principles and methodologies, I rather analyzed driving domain problems by studying end users (applied, Stage 1), helped solving them by transferring, adapting and fine-tuning InfoVis/VA principles and methodologies (transitional, Stage 2) and by providing my lessons learned and study results, I hope to give back valuable input for basic research (Stage 2 and 3).

The second model I want to provide for scope definition purposes is Munzner's Nested Model [Mun09] for classifying steps in visualization design and validation. This model describes four stages, problem domain characterization (orange), data/operation abstraction design (yellow), encoding/interaction technique design (green) and algorithm design (blue). The main contributions of this thesis are located in (but not restricted to) the following layers: observing and interviewing target users (upstream orange), encoding/interaction design (upstream green), downstream validation on target users (downstream yellow) and observing adoption rates (downstream orange). Figure 1.3-b shows the graphical representation of this thesis's focused stages. For a more detailed description of the main contributions please see the next section.

1.4 Contributions and Target Audience

This thesis provides three main contributions: (1) an empirical analysis of current working practices of automotive engineers and the derivation of specific design recommendations for InfoVis/VA tools; (2) the successful application and evaluation of nine design studies, five of them closely integrated as fully implemented systems into the end users' environment; and (3) based on the three-year experience, a set of challenges and recommendations for deploying and evaluating InfoVis systems in large company settings.

1. Analysis of current working practices

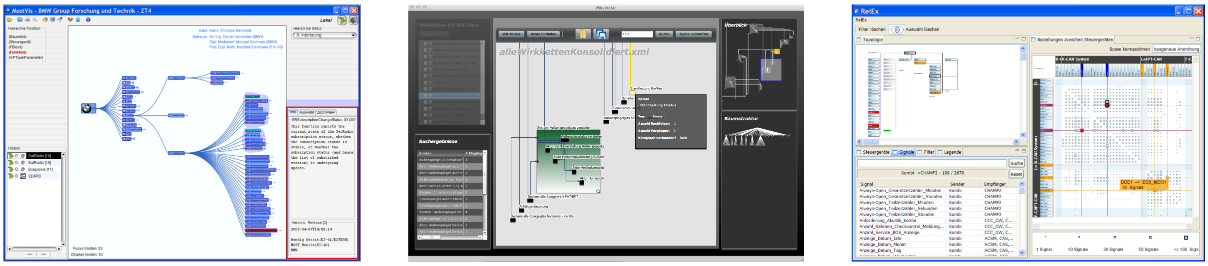
Engineering of in-car communication networks is a large and complex domain with a diversity of intricate datasets to handle. Current tools and practices are mostly text-based and technologies of visual analysis seem to be a fruitful endeavor (see Section 1.1). However, we currently do not know how exactly the challenges and problems in this domain look like nor do we know which tasks can be supported via InfoVis/VA approaches and which not. The first contribution of this thesis is therefore an in-depth field analysis of the automotive network engineering domain. Based on automotive literature review as well as on a variety ethnographic studies over three years including more than 150 domain experts, I describe datasets in use, analyze correlations, differences and dimensions of these datasets, and outline the usage of these datasets. Furthermore, I identified several user groups with driving problems, most importantly analysis experts and network architects, and by observing and interviewing them learned about tasks, problems and their currently used tools. The results of these explorative studies are summarized in general as well as several user group-/use case-specific design implications.

2. Nine design studies for in-car communication network visualization

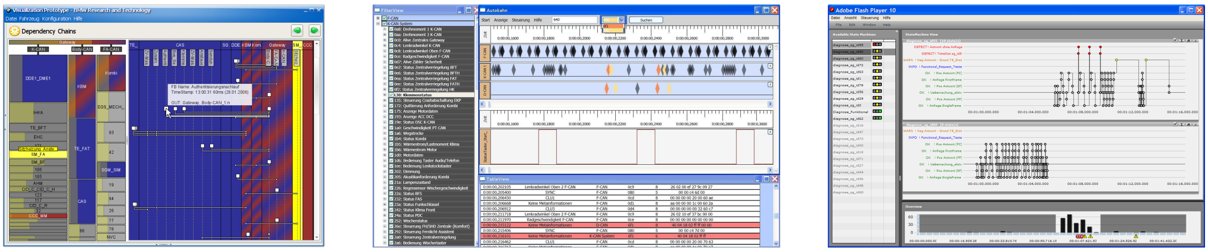
Based on the field analysis results, several visualization systems were developed, ranging from high-fidelity prototypes to fully implemented systems, visualizing different datasets, supporting varying domain tasks and suggesting diverse approaches. Each system was subsequently evaluated with domain experts in order to understand potentials, added value as well as limitations. In this thesis, I present the nine most important of these design studies providing both valuable lessons learned and best practice examples for the automotive electronics domain. The following table 1.1 provides an overview over all visualization prototypes presented in this thesis. Figure 1.4 shows miniature screenshots in order to allow the reader to come back at every time and to quickly reference the prototypes by their visual interfaces.

Name	Brief description	Section	Figure
MostVis	Supports browsing and exploring network specification catalogs.	4.1	1.4-a
WiKeVis	A network visualization for functional in-car network specifications.	4.2	1.4-b
RelEx	A network visualization for physical in-car network specifications.	4.3	1.4-c
VisTra	Trace visualization based on re-computation of dependencies.	5.2	1.4-d
AutobahnVis	Time-based trace visualization.	5.3	1.4-e
Cardiogram	Visualization of state-machine analyzed traces.	5.4	1.4-f
ProgSpy2010	Visualization for analyzing software deploying on ECUs.	5.5	1.4-g
Autobahn3D	An additional 3d view for visualizing in-car communication.	6.2	1.4-h
Car-x-ray	Combination of a 3d model visualization with abstract data representation.	6.3	1.4-i

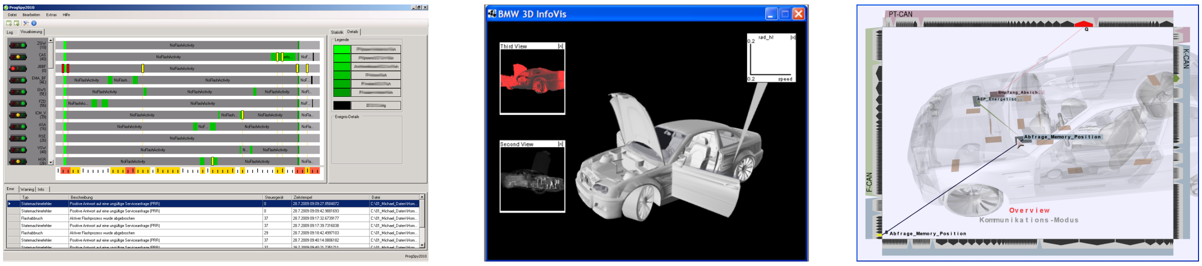
Table 1.1: Overview of the visualization prototypes presented in this thesis.



(a) MostVis (b) WiKeVis (c) RelEx



(d) VisTra (e) AutobahnVis (f) Cardiogram



(g) ProgSpy2010 (h) Autobahn3D (i) Car-x-ray

Figure 1.4: Miniature screenshots of the prototypes presented in this thesis.

3. Challenges, recommendations and orientation for developing and evaluating InfoVis/VA in large company settings

The research presented in this thesis was conducted in close cooperation with a large automotive company. While several researchers have addressed the difficulties of designing and evaluating information visualizations with regards to changing data, tasks, and visual encodings, considerably less work has been published on the difficulties within specific work contexts. Based on the three-year experience, I discuss additional challenges of conducting InfoVis/VA research within the context of a large automotive company, and provide a set of recommendations to deploy and evaluate our tools in such environments. On a first look one might argue that deploying tools is not of interest for research, however, as discussed above for InfoVis/VA tools it is invaluable to install them in real working environments in order to study the long-term nature of data analysis tasks. Challenges and recommendations, but also sharing our success stories, can aid other researchers and practitioners in deploying, preparing and conducting evaluations of their products within a large company setting.

The research presented in this thesis was conducted to contribute to two different communities:

Automotive Engineers: My research showed how we can successfully apply InfoVis/VA techniques in the automotive area, more specifically in in-car communication network engineering. Over the three years, it has become clear that a flood of domain problems and challenges exist which can be supported by harnessing InfoVis/VA methods and technologies. This thesis, provides several best practices examples how to apply these techniques to automotive challenges and how to develop and evaluate them in large company settings. By reading this thesis automotive engineers can adopt solutions, find inspiration and learn about challenges and criteria for integrating InfoVis/VA technologies into their own tools.

Reading recommendations: From an automotive engineers perspective, it might be particularly interesting to focus on reading the brief introduction of InfoVis/VA technologies in Chapter 2 and to refer to one or all chapters presenting the various design studies (Chapter 4 - 6). In this respect, I especially recommend development engineers dealing with specification data to read Chapter 4 and test engineers working with traces Chapter 5.

InfoVis/VA researchers and practitioners: For InfoVis/VA researchers and practitioners this thesis provides the preparation of a novel application area, namely engineering of in-car communication networks, in terms of specific challenges, tasks and requirements. The design studies presented in this thesis can be seen as best practice examples and as benchmark for future developments in this domain. Additionally, the lessons learned can be used for informing research about novel requirements. Moreover, the section about how to deploy and evaluate InfoVis/VA tools (see Chapter 7) in large company settings may be of broad interest, as it provides valuable knowledge for a variety of application areas.

Reading recommendations: While this target audience might skip the brief InfoVis introduction in Chapter 2, depending on the researcher's/practitioner's interest I suggest reading the chapter about the general field analysis (Chapter 3), the sections about the design studies (Chapter 4 - 6), and especially recommend reading the general implications about conducting InfoVis/VA research in large company settings (Chapter 7).

1.5 Structure of the Thesis

After this introductory chapter, the remaining chapters of the thesis are organized as follows.

Chapter 2: This chapter provides a brief introduction to the InfoVis techniques which are used in the design studies presented in this thesis.

Chapter 3: To understand problems, challenges as well as the prototypes, it is mandatory to have a solid background knowledge on in-car communication technologies. Based on a literature review and on results of a variety of ethnographic field studies with automotive engineers, this chapter provides insights into the technical background of in-car communication networks, their development process, visualization techniques currently used by engineers in this domain, tasks, datasets as well as challenges for InfoVis/VA. Based on these findings, a set of first design implications for visual analysis tools in this domain is derived.

Chapter 4: This is the first of three chapters presenting design studies for the domain of in-car communication engineering. In this chapter, three interactive visualization tools for supporting network architects are presented. These tools are based on graph and hierarchy visualization and were designed to support browsing and exploring large specification catalogs and to better understand correlations and dependencies within in-car cnetwork specifications.

Chapter 5: Next, the focus is on supporting analysis engineers in working with large network log files (traces) where one hour of recorded data can be as much as 2GB of data or 50 million messages. Along with the results from an intensive field study of this group and design recommendations we derived, three different visual analytics tools are introduced and discussed in terms of evaluating them with domain experts.

Chapter 6: After that, two design studies on harnessing a 3D model of a car for a better understanding of correlations between mechanical aspects and electronic information are presented.

Chapter 7: While the previous chapters dealt with the design of InfoVis/VA tools in the automotive engineering domain, Chapter 7 focuses on deploying and evaluating InfoVis/VA in large company settings in general. Based on our field experience, I present challenges as well as a set of recommendations how to overcome them.

Chapter 8: To conclude with, I summarize the research objectives and contributions of the thesis and shed light on future issues on InfoVis/VA in the automotive domain and in large company settings in general.

Chapter 2

Visualization Background and Related Techniques

This chapter gives a general introduction to information visualization and provides a brief literature background of related work and techniques from this area. This background is important to better understand and relate the design studies presented in this thesis (Chapter 4–6) to the larger area of information visualization and to learn about similar approaches. While for a reader with an InfoVis/VA background most of the aspects presented in this chapter might be clear, it particularly provides a valuable introduction for readers from the automotive domain. However, as describing all relevant techniques in depth would go far beyond the scope of this thesis, I only give a brief overview and recommend the interested reader the carefully chosen literature for more information.

The chapter is organized as follows: After giving an introduction to general goals and techniques of InfoVis, I present Multiple Coordinated View (MCV) techniques which are used by (nearly) all prototypes presented in this thesis. Then, I directly focus on sub-areas of InfoVis research related to the design studies presented in this thesis by distinguishing the type of data visualized. MostVis, the first prototype presented in Section 4.1 visualizes hierarchical structures of specification documents. WiKeVis and RelEx are based on network visualization, showing dependencies and signal paths (cf. Section 4.2 and 4.3). The trace visualization prototypes presented in Chapter 5, namely VisTra, AutobahnVis, Cardiogram and ProgSpy2010 are mostly based on time-based visualization techniques. Finally, Autobahn3D, Car-x-ray and the two early prototypes presented in Chapter 6 use 3d representations of a car to bridge the gap between physical and abstract information. To better classify and understand the prototypes within the field of information visualization, I give a short introduction to the research areas of tree visualization (background for MostVis), to graph drawing (WiKeVis and RelEx), to time-based visualization techniques (VisTra, AutobhanVis, Cardiogram and ProgSpy2010), and to the usage of 3d representations of physical entities within the field of InfoVis/VA (Autobahn3D and Car-x-ray).

2.1 General Visualization Principles

> "A picture is often cited to be worth a thousand words and, for some (but not all) tasks, it is clear that a visual presentation—such as a map or a photograph—is dramatically easier to use than is a textual description or a spoken report" [Shn96]

"A picture is worth a thousand words" is probably one of the most cited sayings in information visualization. It describes the fact that it is often much easier to communicate complex issues with an image or a figure instead of explaining it with detailed text. Shaped by Frederick R. Barnard in the 1920s, this saying can also be seen as the primal rationale of the dedicated research area of "Information Visualization", or short InfoVis, which actively started in the mid 1980s.

Since then, several different definitions and various descriptions of the field appeared. For the purpose of this thesis, I use the widely cited definition by Card et al. that also marked the very beginning of the thesis: "Information visualization is the use of computer-supported, interactive, visual representations of abstract data to amplify cognition." [CMS99]. This definition contains three major characteristics of the field and comprises a variety of aspects used by most other definitions and descriptions:

1. **Interactive systems:** InfoVis is about interactive tools, not about static information graphics. Harnessing interactive exploration of data should additionally help gaining insight into the data.
2. **Abstract data:** In contrast to scientific visualization the data has no or little direct relation to the physical world such as, e. g., a temperature measurement at a specific location has. According to Chen, abstract data typically describes nonspatial, high-dimensional information that is often hierarchical- or network-structured and contains multivariate parameters [Che04]. Typical examples can be found in social networks, financial data or in descriptive data bases, e. g., for car data (power, weight, maximum speed, etc.).
3. **Amplify cognition:** The major goal of information visualization is "amplify cognitive performance, not just to create pretty pictures. Information visualizations should do for the mind what automobiles do for the feet" [Car08a]. Differently phrased, this central objective can be found in (nearly) all definitions and descriptions of InfoVis, for instance: "Gaining insight into the data" [Kei02], "make new discoveries" [FvWSN08] or getting an "Ah-ha" effect when exploring the data [Spe07].

The field of information visualization comprises various research areas, most notably, computer graphics, human-computer-interaction (HCI) and cognitive psychology. Especially cognitive psychology plays an important role in InfoVis, for instance, in terms of creating, combining and perceiving different forms of data encodings [Ber83, Cle85, Mac88, War04], preattentive processing (extracting basic features such as color, shape or size without active awareness)

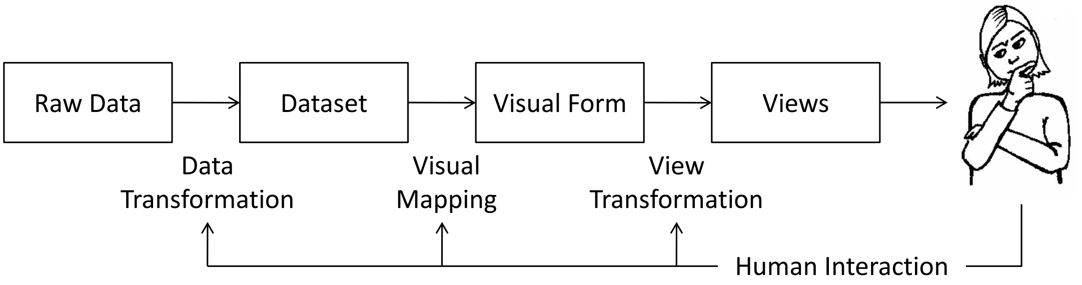


Figure 2.1: The visualization pipeline showing the path from raw data to visual representation (adopted from [CMS99]).

[Tre85, HBE93] or the research on gestalt psychology [Köh29, Kof35]. Additionally, specific interaction techniques such as focus and context [LRP95, MRC91], zooming and panning [FB95] or dynamic queries [Shn94, WS92] have received great attention in InfoVis research and practice. Another important technique commonly used in information visualization are Multiple Coordinated Views (MCVs), presented in more detail in the next section.

To convert raw data into a visual representation Card et al. suggested a model called the *information visualization reference model* or just *visualization pipeline* which is widely used by InfoVis practitioners ([CMS99], see also Chi et al.'s earlier *data state model* [CR98, Chi00]). Figure 2.1 illustrates the model and shows the major steps that are defined. These steps are:

1. **Data Transformation:** The first step is to convert the raw information to a well-organized data format. In this step, for instance, missing values can be interpolated, the data can be filtered, or erroneous information can be corrected. Additionally, data mining and clustering techniques can be applied at this stage in order to compose meaningful formats of the data (cf., for instance, [FGW02] or [Kei02]).
2. **Visual Mapping:** The second step is mapping the dataset to a visual form and is the heart of the visualization process. The data items specified in the previous step are mapped to geometric primitives such as points or lines, and to their visual attributes such as color, position or size.
3. **View Transformation:** In the third step the visual forms are integrated into views which are shown on the screen and which can provide view transformations such as user navigation.

After the raw data has been transformed into visual views, the user can interpret and reconstruct information by looking at the view(s). By interacting with the steps listed above, the user, then, can alter the visualization to further explore the data.

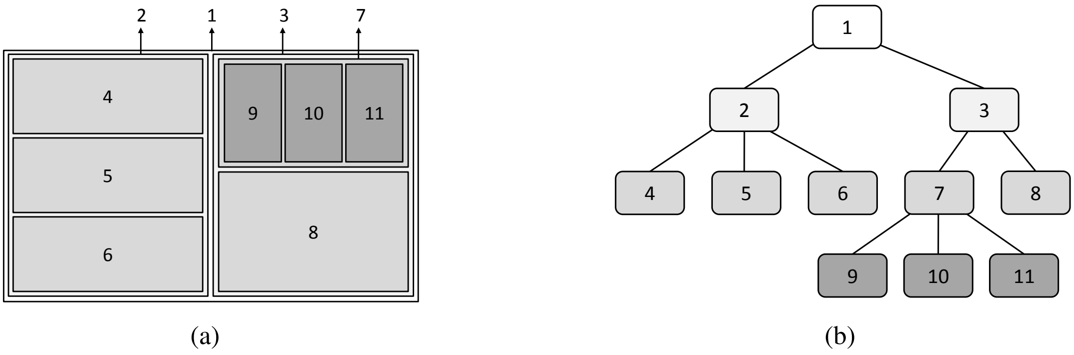
While these general techniques, models and findings definitely influenced the design decisions on the prototypes described in Chapter 4–6, each chapter is allocated to specific data structures within in-car network data, namely hierarchical and network structures of specification documents, time-based structures of traces, and the implicit physical correlation to the vehicle. According to this classification, Section 2.3–2.6 provide more specific background on the visualization of each of these fields within the greater research area of information visualization.

2.2 Multiple Coordinated Views

Multiple Coordinated Views (MCVs) are a frequently used technique in information visualization and are also used in most of the design studies presented in this thesis. Similar to Tufte's small multiples approach where several thumbnail-sized representations of multiple images, e. g., a scatterplot, are displayed in parallel [Tuf83], the basic idea behind MCVs is to use multiple views providing different perspectives of either a single dataset or allowing to compare different datasets via two or more views [NS97, BWK00]. Coordinating these multiple views in terms of user interaction has turned out to be highly beneficial (cf., for instance, [NS00b]). Most important coordination techniques are (cf. North and Shneiderman [NS00a]):

- **Overview and detail:** In addition to a global view on the data, e. g., a map, another view provides a detailed subsection, e. g., a close-up view of the map.
- **Brushing and linking:** When the user hovers over a data item in one view (all) corresponding data elements in other views are accordingly highlighted.
- **Drill down:** After selecting a data item, this particular item is shown in more detail in another view. E. g., after selecting a point in a scatterplot a textual list of all its details is shown in another view.
- **Synchronized scrolling:** Scrolling in one view scrolls another view accordingly (Selection and navigation can be coordinated as well, cf. [NS00a]).

In some situations, however, multiple views can also be of disadvantage in terms of additionally burdening the user with context switching between views [CCY+03] or just by wasting valuable screen space if inattentively used [LMK07]. Therefore, the usage of MCVs should be carefully weighed. Baldonado et al. proposed several helpful guidelines for this decision [BWK00], a cognitive study and a critical discussion of these guidelines can be found in Convertino et al. [CCY+03]. Furthermore, Lam et al. provided a good discussion about separating overviews versus directly embedding them in a focus-and-context manner. In information visualization MCVs have been used for a variety of applications, for instance, to explore high dimensional data spaces [MDH+03], to provide insight into historic hotel visitation patterns [WFR+06] or for visualizing network attacks [NJKJ05]. Also, different frameworks, toolkits, models and applications supporting the design and implementation of MCVs have been suggested, such as Weaver's Improvise [Wea04], North's and Shneiderman's Snap-together [NS00a], Jern et al.'s GAV toolkit [JJJF07] or Boukehelifa's model for coordinating views [BR03].



(a) (b)

Figure 2.2: Visualization of hierarchies: **(a)** Space-filling approach in the form of a treemap dividing a rectangle iteratively into child elements and finally showing rectangles of all leaf nodes; and **(b)** the same hierarchy visualized as a node-link representation.

2.3 Visualization of Trees/Hierarchies

Large hierarchies, or tree structures, appear in many different forms and environments: In file systems, in social structures and also in specification documents for in-car communication networks. Visualizing such hierarchical datasets is an important topic in the domain of information visualization and several tree visualization solutions have emerged over the years. Generally, tree visualizations can be classified into node-link representations and space-filling approaches [SS06]. In both areas a variety of layouts has been investigated. Prominent space-filling approaches are treemaps [Shn96] with their several design variations [BD05, BCS04] and radial ring layouts [SCGM00, YWR02]. Likewise, in the area of node-link representations, many different 2d [LR96, LPP$^+$06, WI90, BJL02] and 3d [Mun97] layouts have been developed and evaluated. A combination of both approaches can be found in [ZMC05]. Figure 2.2-a shows a treemap as an example for a space-filling approach, Figure 2.2-b shows the same hierarchy as top-down node-link representation.

Throughout the process of finding novel layouts, much effort was spent to make tree visualizations scalable to large datasets, e.g., [FP02, Mun97]. User studies about how and when to use which layout have been investigated in [BN01, SCGM00]. Because of the automatically increasing display space for each layer of the hierarchy, radial layouts, for instance, turned out to be advantageous for the representation of large and broad hierarchies. However, on the other hand, they often lead to reduced readability in return.

While widespread approaches representing hierarchies such as the Microsoft Windows Explorer are mostly text based, there are also several tree visualization based tools for browsing large hierarchical data. The PDQ Tree-browser [KPS97], for instance, is an overview and detail tool based on a node-link representation for visualizing large tree structures. Further application examples from different domains can be found in [Bed01, IC07, PGB03].

2.4 Visualization of Networks/Graphs

As the name “in-car communication networks” already implies, the work in this thesis will definitely address network/graph-structured information. From a mathematical point of view, a graph can be described as a collection of *vertices* (also called *nodes*) and a collection of *edges* pairwise connecting vertices. Edges can be directed or undirected as well as optionally weighted. Vertices of large graphs often are additionally ordered in a hierarchical structure. Such graphs are called clustered graphs (or compound graphs). Similar to trees—which are a specific form of graphs, namely connected graphs without cycles—networks can be found in a variety of real world applications, such as social networks, transportation networks, the Internet, telephone, but also in-car communication networks.

The usual representation form of a graph/network is the node-link diagram. The research area of Graph Drawing explicitly focuses on how graph structures can visually be mapped to the 2d plane (and also 3d). Over the years, several different layout strategies such as planar layouts, orthogonal layouts or force-directed layouts have been introduced and optimized regarding perceptability, task completion and computational time aspects. For these layouts, a variety of aesthetics metrics such as minimizing edge crossings, minimizing edge length, maximizing symmetries, or optimizing node distances have been proposed and empirically tested, e. g., finding that line crossing minimization is most important [Pur97] or suggesting a priority of geometric length of the path, continuity and the number of edge crossings [WPCM02]. Optimizing one criterion though often results in worsening other criteria. One of the most widespread application of visual graph representations (and also one of the most cited early examples in InfoVis) are transportation networks with dots for stations and colored lines, e. g., for a specific train line, connecting the stations. For a more detailed overview of graph drawing please refer to [DBETT98], [KW01] or [BW03].

Alternatively to node-link representations, graph structures can be represented with adjacency matrices [Ber83]. In doing so, n nodes are (redundantly) represented in n lines and n columns resulting in n^2 cells of all possible connections (directed). Marking these cells in the matrix is used to specify the edges, usually ‘0’ for ‘no edge’ and ‘1’ for ‘edge’ or appropriate numbers for weighted edges. For visualization purposes, however, cells are more likely coded with symbols, small rectangles and/or colors. Especially for large and dense graphs, it has been shown that a matrix representation outperforms a node-link representation (for some low-level reading tasks) [GFC05]. However, if the task is to find a specific path, node-link diagrams are the better choice regardless of the size of the graph [GFC05].

In information visualization, a variety of applications exist for visualizing graph/network-structured data, such as social network visualization, e. g., [Fre00, HB05], the visualization of the world wide web, e. g., [MB95], or for diagnosing wireless mesh networks [JSH+08]. InfoVis most importantly focuses on the problem of how to visualize very large graphs and how to interactively explore them. Over the last years, a variety of design solutions such as interactive matrix browsers [ZKB02], edge bundling in circular layouts [Hol06] and spatially ordered designs [HvW09] have been proposed and evaluated. While these solutions basically use or vary either node-link or adjacency matrix approaches, combined solutions have been suggested based on multiple coordinated views [HF06] and integrated into the same visual representation [HFM07].

2.5 Visualization of Time-based Data

While in-car network specifications to a large extent are hierarchical- and network-structured, traces—i. e., recordings of communication processes in these networks—are time-based. As derivable from the name, time-based data is characterized by an implicit or explicit chronology in the dataset and usually results from measuring a specific data variable (e. g., stock prices, but also in-car signal values) over time. By default, time-based information is represented in 2d line graph plots and can be found in a variety of popular applications—according to Tufte 75% of graphics drawn in newspapers and magazines between 1974–1980 visualized time-series data [Tuf83].

In the area of information visualization, much work has been done to find categorizations and taxonomies for time-based data, tasks and representation techniques. According to Frank, time-based data can be distinguished in terms of discrete points vs. intervals, linear vs. cyclic time, ordinal vs. continuous time, and ordered vs. branching vs. time with multiple perspectives [Fra98]. Based on this taxonomy, time in traces can be described as discrete (messages recorded at a specific time point), basically linear but also with cyclic messaging, continuous and usually ordered but also with the possibility of parallel events (for a closer description of traces please see Section 5.1.2). Frequent tasks which are conducted on time-series data are detecting whether an element exists at a point of time or not, investigating how long it exists or which other elements exist in parallel, but also non-trivial tasks such as detecting patterns and trends in order to derive deeper insights and knowledge about the data [Mac95, AA06]. For visualizing time-based data, a variety of different interactive visualization techniques has been proposed (for an overview, see, for instance, Aigner et al. [AMM+08]). Influential techniques include ThemeRiver [HHW+02], history flow visualization [VWD04], spiral based layouts for periodic data [CK98], time series bitmaps [KLK+05] and arc diagrams [Wat02]. Good examples for interactive applications of time-based data visualization can be found in LifeLines [PMR+96], Calendar View [VWVS99], VizTree [LKL05] or LiveRAC [MMKN08].

Time-based visualization techniques have also been adopted for network analysis and similar application areas closely related to trace analysis of in-car communication networks. Malony et al., for instance, introduced a system called Traceview supporting visual analysis of time-based network traces [MHJ91]. More recently, Holten et al. showed how massive message sequence charts can contribute to trace analysis [HCvW07]. Pretorius and van Wijk provided several further ideas and systems designed for trace analysis using time-based visualization techniques [PvW05, PVW08], and Hackenberg et al. introduced a techniques for event tracing and visualization for multi-core processor communication [HBN08]. Furthermore, much work has been done to visualize time-based network traffic in order to guarantee security and to detect malicious network activities (cf., for instance, [WMM06], [McR08] or [Mar08]).

2.6 Visualization of Spatial Aspects: Towards SciVis

The last data aspect related to in-car communication networks is the implicit spatiality it adheres by being installed in a vehicle. While InfoVis rather deals with abstract data having little or no physical correlations (cf. Section 2.1) the border to scientific visualization (SciVis) representing data with a strong physical correlation is often seamless [Rhy03, RTM+03]. Many InfoVis tools elucidate relationships between the abstract data and its reference to real world entities by integrating 2d or 3d representations of these entities—usually within a MCV system. Such applications typically depict medical data [NSP97] or geo-spatial information [RLS+96, JJJF07] with additional 2d displays. Techniques harnessing virtual 3d models of physical entities have been, for instance, proposed by Doleisch et al. combining 2d and 3d scatterplots [DGH03] or by Gresh and Rogowitz using solid 3d representations together with 2d scatterplots [GRW+00]. Further examples of combining abstract representations with virtual 3d model entities can be found in Goel et al.'s VizCraft system for supporting air craft designers [GBS+01], or in Ruhland et al.'s approach utilizing a 3d library model for accessing measured and abstract data [RSBO09].

While in the automotive domain much attention has been paid to SciVis focusing mechanical simulations—some of them also combined with InfoVis technologies (cf. Section 3.4.1 for more information about visualization in the automotive domain)—to our knowledge, currently no solutions exist using InfoVis/SciVis techniques for representing in-car communication networks. While the main focus of this thesis is definitely on abstract InfoVis solutions for the hierarchical, network, and time-based aspects in the data, I will also present several solutions for combining 3d models with InfoVis techniques (cf. Chapter 6).

Chapter 3

Understanding the Domain: In-Car Network Engineering

This thesis aims at investigating how InfoVis/VA can support engineering work with in-car communication networks. To do so, it is essential to get a grounded understanding of the target domain and to understand where and how InfoVis/VA can improve current practices. Working with in-car communication networks is a broad and many-sided domain in which thousands of employees work and a diversity of tasks, tools and needs exist. Based on automotive literature review and more importantly on our intensive exploratory field studies over a period of three years, this chapter aims at providing a holistic understanding of the in-car network engineering domain and discusses it from an InfoVis/VA perspective. It presents insights into the technical aspects of in-car networks, development processes, different datasets, user groups, use cases, current working practices and tools, established visualization techniques, and challenges and problems in understanding in-car communication networks.

The chapter is organized as follows: Section 3.1 gives a brief introduction and an overview of the methodology we[1] used for our exploratory field analysis. Section 3.2 continues with explaining the technical background of in-car communication networks. Section 3.3 sheds light on the development process of such systems within automotive manufacturers, defines target user groups and discusses data sources, tools and tasks. Section 3.4 gives a general overview of currently used visualization and of previous attempts to integrate InfoVis/VA with this domain. Primarily based on domain expert interviews, Section 3.5 presents engineers' opinions about potential use cases applying InfoVis/VA to the target domain, their estimations about current lacks in understanding in-car networks and describes how we jointly drilled down the three most important domain challenges for InfoVis/VA in a focus group. Section 3.6 summarizes the chapter, discusses how the findings presented led to the thesis' in-depth focus areas and provides a first set of general design implications for InfoVis/VA application in our domain.

---

[1] The use of "we" in this chapter refers to Michael Sedlmair and in parts additionally to Annika Frank, Martin Knobel and Benjamin Kunze.

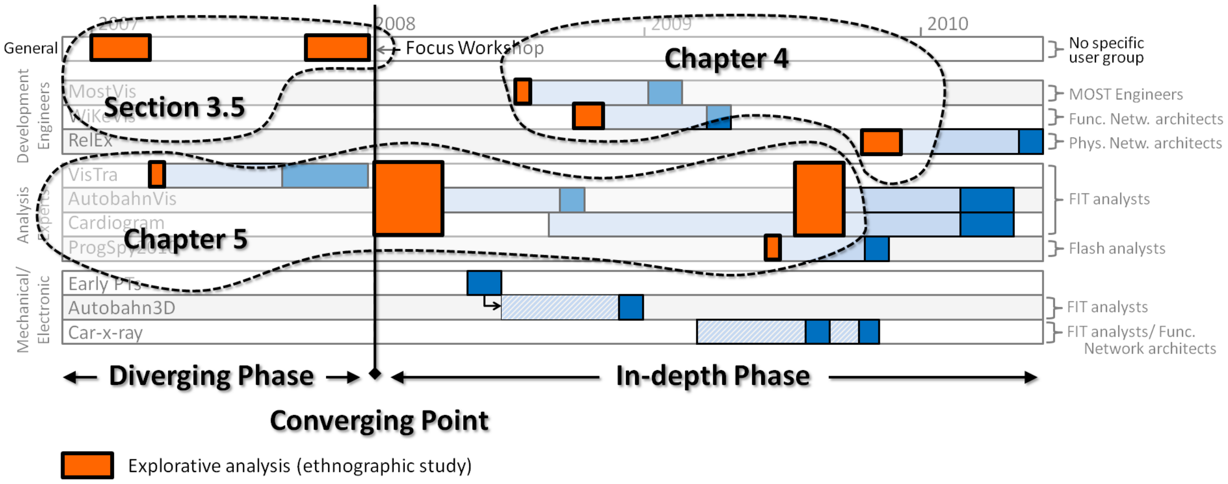


Figure 3.1: Explorative studies which were used as basic information source for this chapter. In general, these studies were divided into three parts: diverging phase, converging point and in-depth phase. A closer description of the single studies can be found in the indicated section/chapters.

3.1 Methodology Outline

To understand our target domain, we conducted a variety of formal and informal ethnographic studies with automotive engineers over a period of three years and involved some of them in participatory and user-centered design processes. Our general approach was based on grounded theory, a systematic approach from social sciences combining different mainly qualitative research methodologies in order to form a realistic theory and a "grounded" domain understanding [GS77]. In this line, we used a variety of qualitative data collection methods and iteratively derived a grounded understanding of the in-car network development and analysis domain. Specifically, we used: observational methods, contextual inquiries, guided interviews and focus groups. In addition, we engaged in frequent informal conversations with engineers who were more closely involved in our user-centered design processes. These conversations helped us to iteratively refine our domain understanding and our knowledge about requirements for visual analysis tools. Overall, more than 150 different domain experts were involved in this explorative field analysis. Our explorative studies were globally organized in three stages:

1. A *diverging phase* to get a broad domain understanding,

2. a *converging poin*t where the thesis was focused on three major directions: visualizing specification data to support development engineers, visualizing traces to support test engineers and providing a better understanding of correlations between mechanical and electronic information, and

3. an *in-depth phase* where we intensively studied specific user groups of development and test engineers in order to derive requirements for designing InfoVis/VA tools for them.

While the methodological description and detailed results of each individual study are distributed over the thesis in order to closely connect them to their projects, this chapter provides a résumé and discussion based on our studies and field experience working together with domain experts over three years. Figure 3.1 highlights all explorative studies (orange) which were used as primary input for this chapter and highlights the three different stages discussed above. A detailed description of the diverging and converging studies can be found in Section 3.5, details about the in-depth studies can be found in Chapter 4 and 5.

3.2 Technical Background

> "Just as LANs connect computers, [in-car] control networks connect a vehicle's electronic equipment." [LH02]

Back in the 1970s, an electronic revolution within the automobile sector started and resulted in a still ongoing process of electronic system integration into automobiles. These systems complemented but also replaced mechanical and hydraulic systems and step-by-step helped to increase vehicles' comfort, safety and efficiency. One of the main purposes of integrating electronics in the car was and still is to better assist the driver, for instance, in steering, accelerating or braking, such as the antilock braking system (ABS), electronic stability program (ESP), electric power steering (EPS) or recently the intelligent parking assist system (IPAS). Another reason for using electronic systems is to control devices incorporated in the vehicle such as lights, windows, or, recently, entertainment, communication and navigation equipment [NSSLW05]. Nowadays, the costs of electronics in high-end vehicles is more than 35% of the total manufacturing costs and experts estimate that 90% of all automotive innovations stem from electronics [BG07a].

In the early days, each electronic function was implemented on a single, dedicated stand-alone electronic control unit (ECU) that individually connected and controlled a set of sensors and actuators. To distribute information such as the vehicle speed over several ECUs point-to-point connections between these ECUs were established. However, soon it became clear that this approach was not scalable and highly inefficient regarding costs, weight, complexity and reliability. Therefore in the 1990s, automotive manufacturers started to integrate in-car (or: in-vehicle) communication networks multiplexing communication over a shared medium for distributing information between ECUs. Since then, different automotive applications posed different requirements to these networks regarding parameters such as bandwidth, reliability or costs and led to the development of several automotive specific in-car communication network technologies [LHD99, BG07a], most importantly:

- CAN (Control Area Network, see [BG91]), the first in-car network technology introduced in 1991, for distributing real-time engine control messages, with high reliability on an average bandwidth (up to 1 Megabits per second);

- LIN (Local Interconnect Network, see [GvdW05]), a low-price, low bandwidth (20 kilobits per second), for safety-uncritical, local connections between ECUs and sensors/actuators such as power window regulation;
- Flexray (see [MHB+01]), high performance, real-time and highly reliable bus system constructed for safety critical x-by-wire functions;
- MOST (Media Oriented Systems Transport, see [Grz07]) with very high bandwidth but low safety requirements for multimedia applications such as, video, audio or navigation; and
- recently also adopting Ethernet for application in automotive networks is intensively discussed (see [DAES06]).

Current in-car communication networks of upper class vehicles are composed of up to thirteen of such bus systems[2] forming a larger complex network. 20–100 ECUs per vehicle are connected in such networks and distribute more than 8,000 different signals [Alb04, BG07a]. In order to update information, individual signals are packed together into messages and are sent between ECUs via one or several bus systems (connected via a central gateway). This leads up to 15,000 messages per second in an in-car communication network.

From an architectural perspective, in-car communication networks are divided into two main views: The physical architecture and the functional (also logical) architecture.

The *physical architecture* describes the network from a hardware perspective, namely ECUs, Gateways and bus systems but also how these components communicate via messages and signals. This is the perspective we outlined above and which is, according to our findings from observations and interviews, the perspective engineers use most commonly when talking about in-car communication networks. Therefore, it is often used synonymously for in-car networks in general.

The *functional architecture* on the other hand is used in early development phases (and also later for deriving the software architecture) to structure the entire network based on functional aspects. Usually, this is done by applying a formalized, object-oriented modeling language such as UML or COLA (a specific modeling language for the design and development of automotive embedded systems [KTB+08]). To do so, each vehicle's system, such as ACC (Adaptive cruise control), air conditioning or radio, is step-by-step broken down to small and reusable functional units, called functional blocks (FBs). A FB, for instance, "Audio Amplifier", in turn is broken down to the functions it should provide, such as "Audio Mute". Functional blocks therefore could be compared to a class in object-oriented programming, while functions would correlate to methods. Interfaces and correlations describe communication paths between FBs and, if applicable, are specified based solely on physical parameters in order to be independent of the actual implementation [Kra06].

---

[2] This does not implicitly mean that there are 13 different basic technologies. Some technologies have been fine-tuned for specific application domains, e. g., K-CAN (a slow CAN bus system for comfort and car body technique) or PT-CAN (a fast CAN bus system drive control).

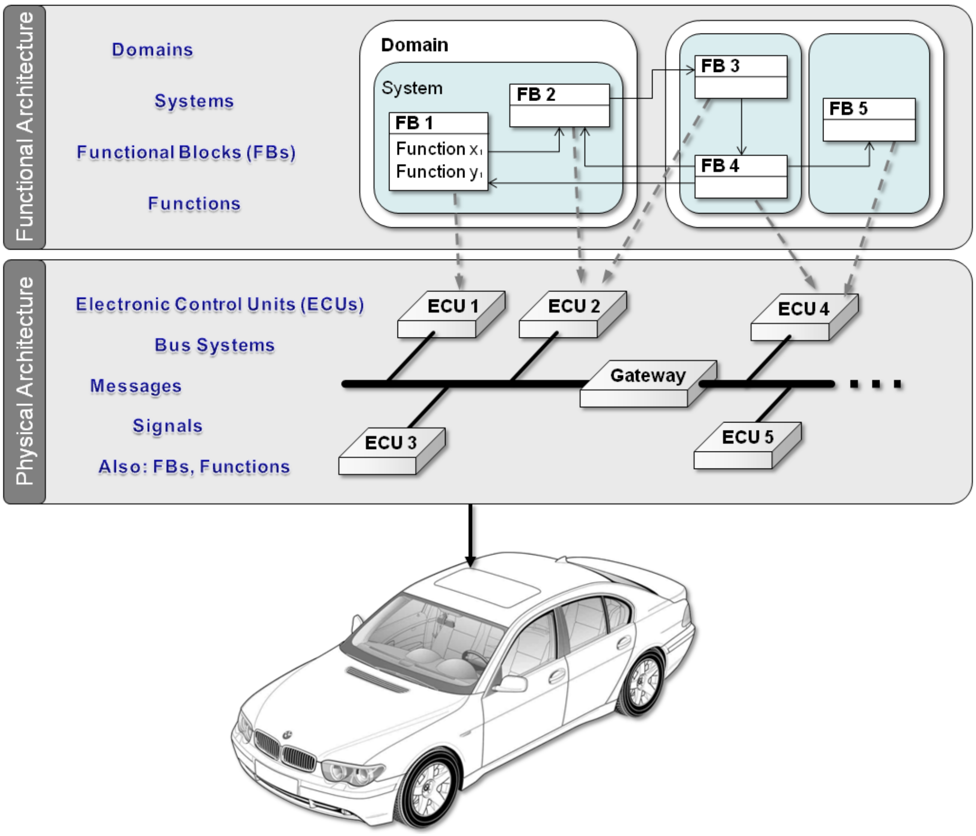


Figure 3.2: Top: Functional/logical network architecture with functional blocks (FBs) clustered in systems and domains. Bottom: Physical network architecture showing Electronic Control Units (ECUs) connected via different bus systems. Mapping FBs onto ECUs is an important part of partitioning the functional to the physical network. The blue terms provide an overview over the basic vocabulary used on each layer.

After the functional network for a car series has been specified, in a next step FBs are mapped to ECUs and the information exchange between FBs is mapped to signals and messages sent over the vehicle's bus systems. FBs of one logical system (e. g., ACC) are not necessarily mapped to one specific ECU but are often distributed over several ECUs. Hence, such systems are called distributed systems (or distributed functions). Engineers' decisions of mapping FBs to ECUs depend on a variety of functional requirements (e. g., control path) and non-functional qualities (e. g., costs). The mapping process is called partitioning and the result is the physical network architecture [Kra06].

Figure 3.2 gives a schematic overview over the two architectural layers, shows how they correlate and highlights which terms are used in this thesis to describe in-car communication networks. Unfortunately, terms and definitions about in-car communication network components (especially on the functional layer) are not consistently used in literature, nor over different automotive companies and even not always within one company. Based on Bosch's Cartronic [Kra06], the MOST Specification [Grz07], Autosar [Aut], and on our own experience working in this domain, in the following I summarize all relevant terms and clearly define their meaning within this thesis. For each term a brief summary is given which can be used as a quick reference on technical aspects for better understanding the prototypes presented in Chapter 4 – 6:

ECU: Abbreviation for Electronic Control Unit. ECUs are the electronic hardware components and processing units of a vehicle. An ECU is composed of an embedded computing unit—ranging from simple microcontrollers to complex multi-core processors with a dedicated graphic system—and usually controls a set of sensors and/or actuators. Typical ECUs are Airbag Control Units (determining whether an airbag is inflated or not), Seat Control Unit (controlling electronic seat adjustment), Engine Control Unit (controlling the engine behavior) or Car Infotainment Computer (controlling, for instance, audio, video and navigation systems).

Sensor/Actuator: Sensors and Actuators are either directly linked with a specific ECU or connected via a local subsystem (LIN). Sensors measure qualitative or quantitative variables, such as speed, temperature but, e. g., can also deliver a camera stream. The processing of signal values is conducted in ECUs. Actuators are the counterparts of sensors and convert electronic signals into mechanical motion, or other physical actions, such as controlling the suspension of a vehicle or moving a window.

Bus system: Each ECU is connected to one or several bus systems. Bus systems serve as shared message carrier systems for multiplex communication between ECUs with specific protocols for managing the communication and, in particular, for granting bus access (such as CSMA/CR on a CAN bus). According to different cost, bandwidth and reliability requirements varying technologies exist, most importantly: CAN (with sub-specifications such as K-CAN or PT-CAN), MOST, LIN, Flexray and Ethernet (see above for more details). Typically, a high-performance vehicle contains several of these bus technologies which are then interconnected via one or several Gateway Control Units.

Signal: Signals are the elementary information and communication units. Signals are sent between different ECUs as well as between ECUs and their actuators/sensors. Signals contain, for instance, measured variables such as temperature but also abstract software information such as the specific value of a parameter.

Message: In order to send signals over a bus system, several of them are packed together in a message (e. g., for CAN up to eight signals per message). Similar to computer networks each message gets a specific header with additional metadata such as sender, receiver or check information. Before packed to a message, signals are usually pre-clusters in protocol data units (PDUs) which then in return are packed to messages. As this concept is only of peripheral importance for this thesis, it is mentioned here just for completeness purposes.

Functional block (FB): Functional blocks (FBs) are software units similar to classes in object oriented programming. Each FB is implemented on a specific ECU and communication between FBs is operated by signals. An ECU usually holds up to 100 FBs.

Function: Each FB consists of several functions equivalent to functions and methods in object oriented programming.

System: The term system is used to describe a specific functionality of the car (and therefore is often misleadingly called function). Typical examples of systems are adaptive cruise control (ACC), ABS brake but also air conditioning or radio. A system is realized by mechanical, hydraulic/pneumatic and increasingly by electronic components in the car. Each system with an electronic component is based on a set of FBs which are mapped to one or more ECUs. Large systems are often additionally clustered in sub-systems.

Distributed System: If a system's FBs are distributed over two or more ECUs this system is called a distributed system. A typical example is ACC which combines input from several distributed sensors and controls gear-, brake- and engine-control units in return.

Domain: As there can be several hundred systems in a vehicle configuration, systems are clustered together to domains such as driving assistance, communication or motor. Domains are additionally used as a basic organizational structure in automotive companies.

Partitioning: The process of mapping the functional/logical architecture to the physical architecture, namely FBs onto ECUs and correlations onto signals and messages, is called partitioning.

3.3 Development Process and Data Sources

The previous section provided a brief introduction to historical and technical backgrounds of in-car communication networks. In this section, I focus on the question of how in-vehicle electronics are developed in a company and explain what steps and tasks are necessary during the development process and which datasets and tools are used.

3.3.1 Development Process: The V-Model

Along with the increasing importance of software within modern vehicles, developing in-car electronics and networks has become a long and highly collaborative process. In the 1990s, it became clear that for guaranteeing a high product quality and for increasing planning reliability a novel, structured and robust development process is necessary. Therefore, large (German) automotive manufacturers started to apply a process management tool standardized by the German government, called V-Model [DW99, BR05], and adapted it to the specific requirements of automotive electronic development [SZ06, WR06]. The model divides the entire process into smaller parts which can be executed in a 'divide and conquer' manner. The goal of the model is not to exactly reflect reality and force the development process into an inflexible schedule but rather to help structuring the entire progress and to provide a standardized "discussion basis" to help making important steps more concrete [WR06].

Figure 3.3 shows a simplified version of the V-Model used today in the automotive domain for developing in-car electronics and networks. The project's progress develops from the left to the right by following the 'V' form. The 'V' form is clustered into several steps and horizontal back

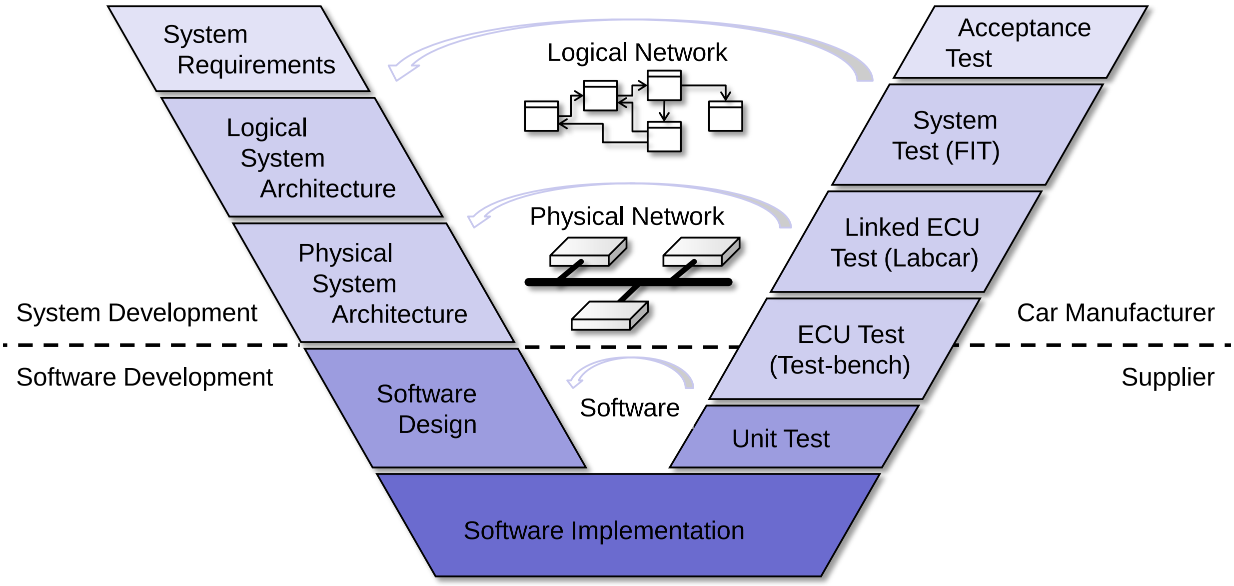


Figure 3.3: Simplified and schematic view of the V-Model used in the automotive industry derived from Schäuffele and Zurawka [SZ06] and Wallentowitz and Reif [WR06]. The V-Model separates the entire development in several larger steps of system specification and testing, shows iterative cycles, describes the layers of functional and physical networks during the development process and highlights the border between car manufacturers (system development) and suppliers (software development).

links between these steps mark iteration cycles in the development process. The following description gives a brief and simplified outline of the process steps in automotive electronic development: Starting with analyzing user requirements and mapping them to system requirements, the entire car electronic is modeled and specified first on a functional/logical layer and then mapped to the actual physical network (see previous section for more details). At that point the work is passed over from car manufactures to supplier companies which design and implement the software (and also the hardware which is usually not shown in the automotive electronic V-Model) for a specified sub-system, i. e., one, several or parts of one or several ECUs. The upside-edge of the 'V' concentrates on testing the system. The ordering is arranged accordingly to steps taken in the developing phase: After software unit testing, single ECUs are tested detached from the larger system, e. g., by using test benches (see Figure 3.4-a). During that phase the work is handed over from the suppliers to the manufacturers again. Then, ECUs are step-by-step linked together and checked in test installations such as labcars (see Figure 3.4-b). After integrating the electronic systems into the vehicle, an extensive series of final test runs with specially equipped test vehicles is conducted to verify a series' safety, security, and functionality (see Figure 3.4-c). These tests are called 'intensive vehicle tests', or in short FIT[3]. Finally, customer acceptance tests are conducted.

---

[3] Derived from the German translation: 'Fahrzeug Intensiv Test'.

(a) (b) (c)

Figure 3.4: Different test hardware: **(a)** A test bench for testing single ECUs, **(b)** a labcar reproducing the in-car network on a rack, and **(c)** measuring hardware installed into the trunk of a FIT car.

In reality, this process is not linear from left to right but characterized by many iteration cycles which appear most likely on a horizontal back link: if a unit test fails software design must be corrected, if a ECU integration test fails the physical network specification must be checked or if a FIT test fails an error in the functional network design is often the cause. It is important to notice, that the later an error occurs in this process the more expensive and unpleasant it is to fix it. The reasons are simple: (a) Late errors usually require going back far in the development process and re-running all steps in-between is expensive, and (b) usually the time left until start of production (SOP) is short. Therefore, finding quick, working solutions is highly urgent.

3.3.2 Target Groups According to the V-Model

The automotive V-Model provides two basic distinctions which are important for this thesis. First, there is a clear distinction in the model between system- and software development and along with that an implicit distinction between vehicle manufacturers (basically system development) and suppliers (basically software development). As this work was conducted in cooperation with a large car manufacturer, this thesis focuses on system development and testing and not on software design, implementation and testing. Second, the two wings of the ‘V’ divide the engineering of in-car networks into two main groups: development engineering and test engineering (see Figure 3.5). While development engineers are basically designing and specifying the functional as well as the physical network, test engineers analyze the correctness of implemented systems and feed back errors to development engineers. This basic distinction of user groups is also used for organizing this thesis and is reflected in Chapter 4 which presents solutions for development engineers, and in Chapter 5 presenting solutions for test engineers.

3.3.3 Data Sources, Tasks and Tools

InfoVis and VA are about visualizing large and complex datasets. Therefore, an important aspect we particularly were interested in was the question about commonly used data sources with in-car communication networks. After identifying these data sources, it was important for us to

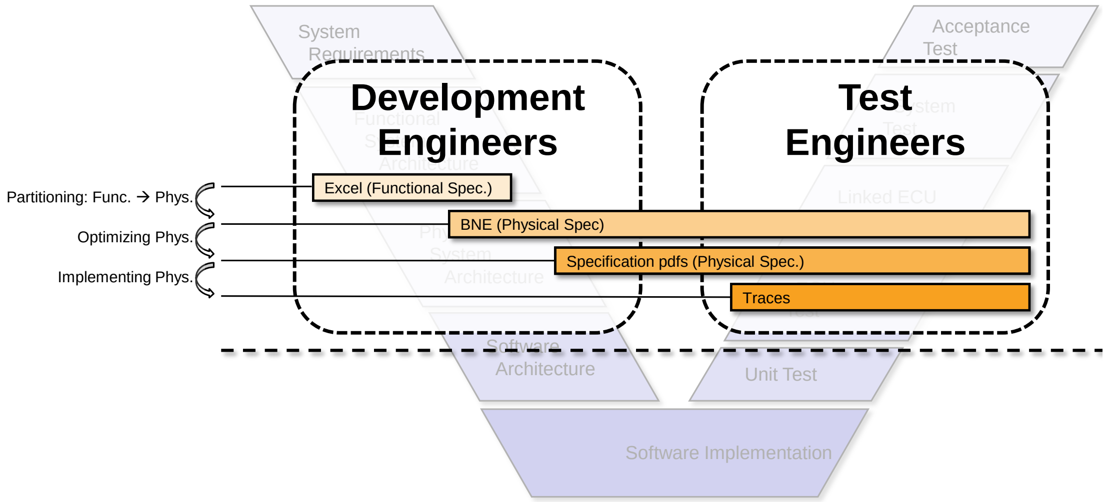


Figure 3.5: According to the V-Model two basic in-car network groups of engineers exist: development engineers and test engineers. Along the development process different data sources are used. The arrows on the left show the evolution of the data along the development process.

understand what domain engineers do with these datasets and what tools they currently use for exploring it.

The data used by engineers generally can be classified—according to the development process introduced above—in three groups: (1) Early/functional specification data, i. e., domains, systems, subsystems, functional blocks, functions, (2) physical specification data, i. e., bus systems, ECUs, functional blocks, functions, messages, signals, and (3) recordings of real communication processes for testing purposes (list of message instances). We found four main sources of these datasets:

Excel sheets—for functional specification: In early development phases during which the functional in-car communication network is specified, we frequently observed engineers using Microsoft Excel sheets to store their data. As at the time we conducted our studies, the official in-car network database BNE (see below) did not allow for filing domain, system, and subsystem information, engineers applied this workaround instead. They used Excel sheets for storing, sharing and collaborating with their data during the design of functional network specifications (see Section 4.2 for an example and for more information). Depending on the purpose, the Excel sheets got relatively large with more than thousand lines and tens to hundreds of columns and therefore working with them was a challenge. As the process of manually and repeatedly generating local Excel sheets for specifying functional networks is error-prone and inefficient, current efforts increasingly head for model-based automotive software development [BBR$^+$05], for the usage of CASE-tools[4] such as PREEvision [Aqu10], and for a close integration of functional specification in the official specification database BNE.

---

[4] Computer-Aided Software Engineering, see [Fug93].

BNE—for (temporal and final) physical specification: The BNE[5] is a large, company-wide available database that stores all specified information about physical in-car communication networks for all the company's vehicle series, types and variants. This includes information on ECUs, bus systems, messages and signals but also on functional blocks and functions that have been partitioned to the physical components (see Section 3.2 for a closer description). The BNE supports various import/export formats, most importantly fibex, an XML exchange format for data of message-oriented bus communication systems [Ass10, ZS06] and comes with a windows explorer-like, text-based interface allowing users to directly explore and/or manipulate all information stored in the database (see Figure 3.6-a). By using this interface, engineers, for instance, can search and explore signals and their senders/receivers, messages and the bus systems via they are transported, or ECUs and their associated functional blocks.

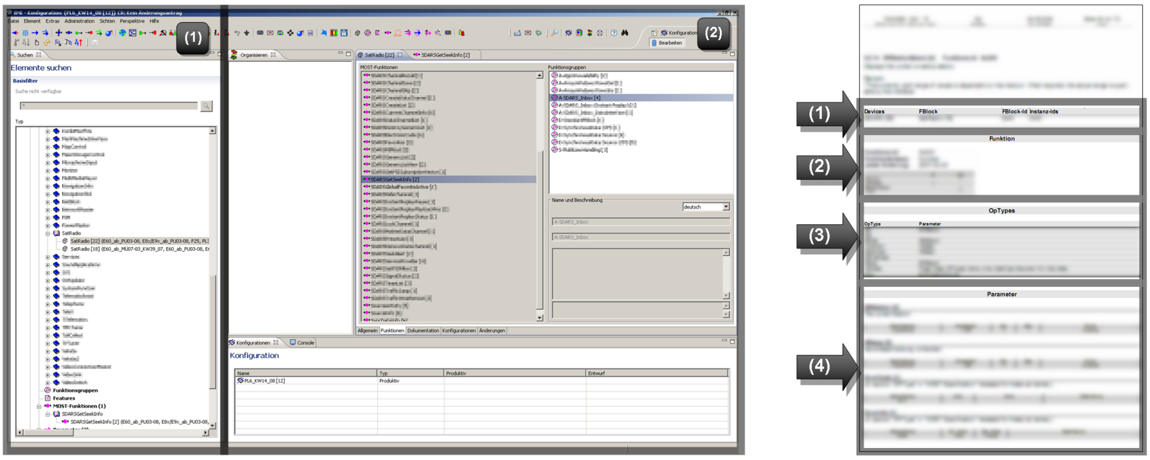
The BNE is used for several purposes: First, it is used for specifying and describing components during partitioning the physical network, and in doing so to collect, enter and maintain all relevant data (network architects with write access, cf. Section 4.1 and 4.3). Second, nearly all engineers working with in-car communication networks use the database at least occasionally for information investigation (development and test engineers with read access). Inquiries to the BNE are usually about finding specific parameters, e. g., 'finding a specific signal value and its meaning', or understanding interactions between elements, e. g., 'which functional block belongs this function to', or 'which signals does ECU A send/receive'. The intrinsic goals of engineers, however, are mostly on a higher level, such as, 'which other system will be influenced when I change this/my function' or 'what are dependencies of this ECU within the entire network'. These goals, however, have to be translated by the engineers to the more elementary queries stated above and the higher level task has to be accomplished by manually combining these information snippets. Other purposes of the BNE are specification generation (see next paragraph) and interpretation of raw data (see below, paragraph about traces).

Specification documents—for (final) physical specification: Specification documents are formalized requirement descriptions of a system, such as an ECU device or a bus system. After a "feature freeze" for a specific vehicle, PDF dumps of the specification are generated from the BNE and are given to the respective device suppliers as contract agreement. In addition to that, the PDF dumps are widely used throughout the company as quick reference, especially for users without access to or experience with the BNE. With the increasing functionality of modern vehicles, however, the specification documents have become very large [WW03] making browsing and deriving information from these documents a challenging task. Figure 3.6-b schematically illustrates one page of a MOST functional specification document, showing the definition of a functional block with its included functions and parameters (for more information see Section 4.1).

Traces—recordings from the implemented network: To analyze and debug in-car communication networks, automotive test engineers record traces, large temporally ordered lists containing all messages sent over one, several or all bus systems. For this purpose, usually specific recording devices are installed in test cars, logging the communication by wiretapping bus systems and

---

[5] Short for 'Bordnetz Engineer', 'Bordnetz' is the German word for in-car communication network.



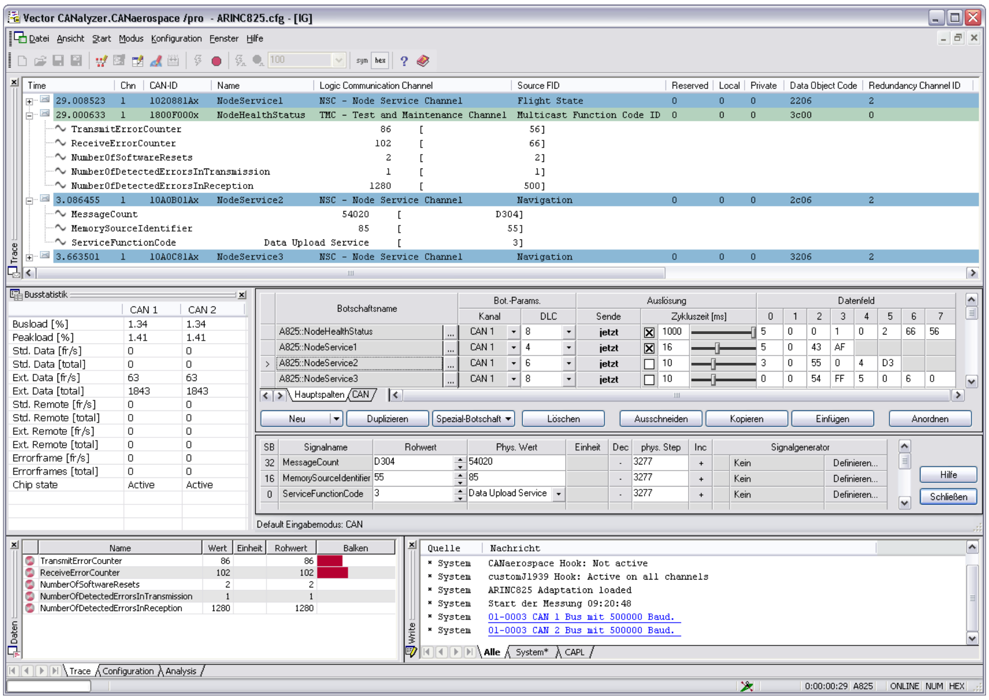
(a) BNE interface

(b) Specification pdf

Figure 3.6: Tools for working with specification data: **(a)** The BNE interface with (1) an overview and navigation frame and (2) a frame for showing all detailed information within different sub-frames; and **(b)** an example page of a PDF specification from a MOST functional catalog with information about (1) a specific FB including (2) its functions, (3) optypes (specific MOST formulation for data types) and (4) parameters; Altogether, there are over 4,000 pages in this catalog (Details have been blurred due to IPR restrictions).

storing all messages on the device's hard drive (see Figure 3.4-c). The resulting trace lists can become tremendously large as logging one hour on all bus systems of a vehicle, for instance, results in approximately 2GB of data and roughly 50 million recorded messages. To work with traces, engineers use special-purpose software such as Canalyzer [Vec] (see Figure 3.7), or in-house tools such as Carmen. These tools interpret the data (using specification exports from the BNE) and present it usually in textual lists, simple signal plots and virtual gauges (for more information on traces and analysis tools, please see Section 5.1.2).

The data sources, tools and tasks described above are the ones which we encountered during our studies and which were frequently mentioned by our subjects as important sources posing challenges in terms of data size and complexity. We are aware that there might be additional sources, such as statistical data from simulation tests or other databases, which were not specifically mentioned or used by our participants during our studies. Figure 3.5 (see page 32) shows the datasets we found along with the V-Model and the user groups, and clarifies what data is used at which time of the development process.



```
Time       Chn  CAN-ID     Name               Logic Communication Channel           Source FID                     Reserved  Local  Private  Data Object Code  Redundancy Channel ID
29.008523  1    1020881Ax  NodeService1       NSC - Node Service Channel            Flight State                   0         0      0        2206              2
29.000633  1    1800F000x  NodeHealthStatus   TMC - Test and Maintenance Channel    Multicast Function Code ID     0         0      0        3c00              0
   TransmitErrorCounter                              86      [             56]
   ReceiveErrorCounter                              102      [             66]
   NumberOfSoftwareResets                             2      [              2]
   NumberOfDetectedErrorsInTransmission               1      [              1]
   NumberOfDetectedErrorsInReception               1280      [            500]
3.086455   1    10A0B01Ax  NodeService2       NSC - Node Service Channel            Navigation                     0         0      0        2c06              2
   MessageCount                                54020     [            D304]
   MemorySourceIdentifier                         85     [              55]
   ServiceFunctionCode          Data Upload Service      [               3]
3.663501   1    10A0C81Ax  NodeService3       NSC - Node Service Channel            Navigation                     0         0      0        3206              2
```

	CAN 1	CAN 2
Busload [%]	1.34	1.34
Peakload [%]	1.41	1.41
Std. Data [fr/s]	0	0
Std. Data [total]	0	0
Ext. Data [fr/s]	63	63
Ext. Data [total]	1843	1843
Std. Remote [fr/s]	0	0
Std. Remote [total]	0	0
Ext. Remote [fr/s]	0	0
Ext. Remote [total]	0	0
Errorframe [fr/s]	0	0
Errorframes [total]	0	0
Chip state	Active	Active

Botschaftsname	Bot.-Params.		Auslösung			Datenfeld							
	Kanal	DLC	Sende	Zykluszeit [ms]		0	1	2	3	4	5	6	7
A825::NodeHealthStatus	CAN 1	8	jetzt	☒	1000	5	0	0	1	0	2	66	56
A825::NodeService1	CAN 1	4	jetzt	☒	16	5	0	43	AF				
A825::NodeService2	CAN 1	6	jetzt	☐	10	3	0	55	0	4	D3		
A825::NodeService3	CAN 1	8	jetzt	☐	10	0	0	54	FF	5	0	6	0

SB	Signalname	Rohwert	Phys. Wert	Einheit	Dec	phys. Step	Inc	Signalgenerator	
32	MessageCount	D304	54020		-	3277	+	Kein	Definieren...
16	MemorySourceIdentifier	55	85		-	3277	+	Kein	Definieren...
0	ServiceFunctionCode	3	Data Upload Service		-	3277	+	Kein	Definieren...

Default Eingabemodus: CAN

Name	Wert	Einheit	Rohwert	Balken
TransmitErrorCounter	86		86	🟥
ReceiveErrorCounter	102		102	🟥
NumberOfSoftwareResets	2		2	
NumberOfDetectedErrorsInTransmission	1		1	
NumberOfDetectedErrorsInReception	1280		1280	

```
Quelle    Nachricht
* System  CANaerospace Hook: Not active
* System  customJ1939 Hook: Active on all channels
* System  ARINC825 Adaptation loaded
* System  Start der Messung 09:20:48
* System  01-0003 CAN 1 Bus mit 500000 Baud.
* System  01-0003 CAN 2 Bus mit 500000 Baud.
```

Figure 3.7: Screenshot of Canalyzer [Vec], an example for a trace analysis tool.

3.3.4 Discussion: Perspectives on the data

In-car communication networks data comes in many guises. The appearance varies over the development process and depends on the engineers' tasks. For InfoVis/VA, it is crucial to understand who you are designing for and what aspects of the data are important for them.

The first important discrimination is between physical and functional network and was intensively discussed in Section 3.2. We call this the architectural perspective.

Second, there is a difference between specified network data and the real configuration/communication processes. Specification documents of an in-car network such as the BNE or the specification PDFs describe all elements (ECUs, FBs, mesagges, and so on) that exist in networks of all car variants of a series. The automotive jargon often refers to that as the "150-percent specification" since it includes all specified components though they never can be installed simultaneously in one car. Its counterpart is the "100-percent configuration" describing an actual setup of one specific vehicle. Furthermore, it is important to notice that the actual communication (trace recordings, see Chapter 5) differs from the specified information (network specifications, see Chapter 4), or real dynamic behavior and ideal static description. For instance: (a) The specification defines a maximum response time for messages, in the real communication the actual value, however, is most likely higher or smaller; (b) the specification defines all possible dependencies and message paths, in a real communication process, however, only a subset of these dependencies appear; or (c) the specification defines each message once, in the actual com-

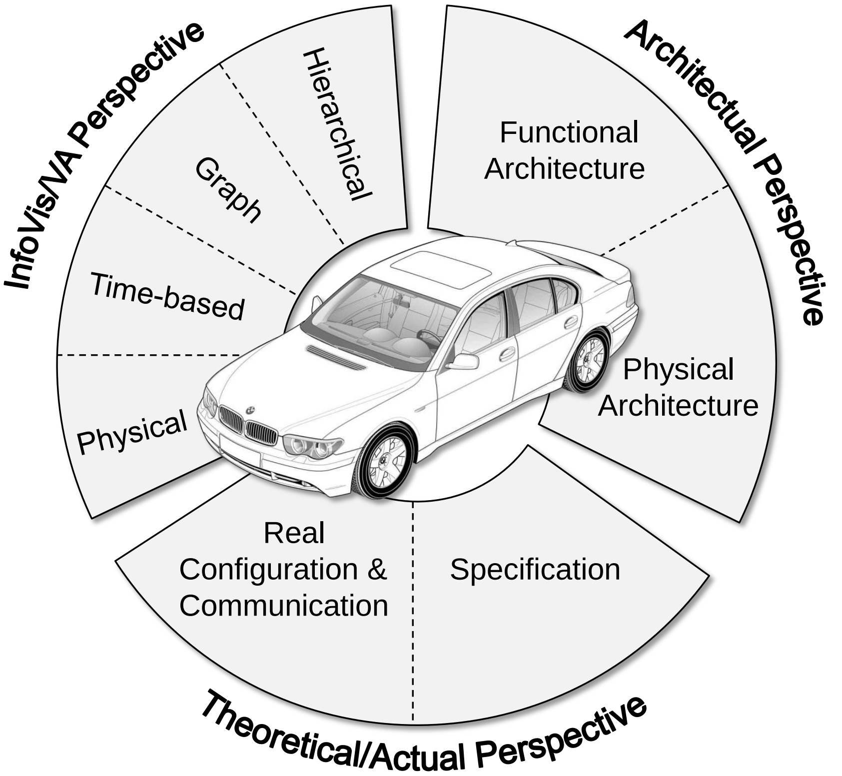
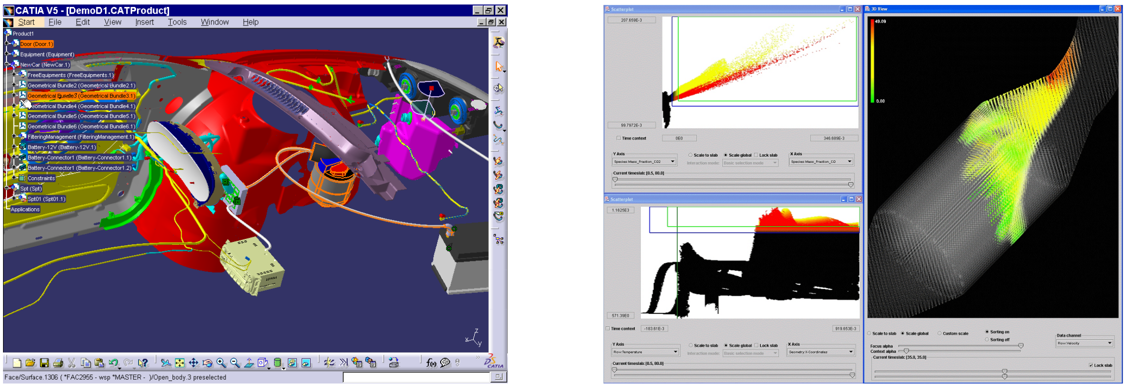


Figure 3.8: There are different perspectives on the data which can be taken into account when designing InfoVis/VA tools for the in-car communication domain.

munication instances of this message usually appears a thousandfold. We call this perspective therefore specification/actual perspective.

Finally, there are various aspects and structures within the data which we as InfoVis/VA researchers/practitioners might find interesting to support with proper visualization techniques. First, there are hierarchical structures in the data. For instance, the specification files are organized hierarchically, ECUs have functional blocks, functional blocks have functions, functions have parameter, and so on. Section 4.1 presents a visualization of MOST specifications that visualizes these hierarchical structures in the data. Second, networks intrinsically provide graph structured data, e. g., ECU communication (nodes: ECUs, edges: messages or signals) or FB communication (nodes: FBs, edges: specified correlations or signals). Section 4.2 and 4.3 shows two network visualizations based on node-link and matrix representations of such graphs. Third, traces provide time-based information in form of chronically ordered lists of messages. Chapter 5 provides several time-based visualizations. Last, the data always has an implicit connection to the physical entity car. ECUs and bus systems are carefully positioned within the vehicle, messages are sent ’through’ the car, or specific signal values cause mechanical components to react, for instance, a window to move. Chapter 6 provides our ideas how to integrate these physical aspects into visualizations.



(a)

(b)

Figure 3.9: Screenshots of **(a)** CATIA, a CAD tool widely used in the automotive domain [Das]; and **(b)** SimVis, a multiple coordinated view approach combining the benefits of SciVis and InfoVis to support the analysis of Diesel exhaust systems [DMG+04].

While it is important to be aware of each single dimension in the data, it is equally worth to think about combining them, for instance, visualizing the mapping between functional and physical network, providing a comparison between specified and actual communication data, or to visualize hierarchical clustered graphs. Which combination is sensible and useful, has to be figured out in close collaboration with domain experts. Several of our approaches are based on such combinations.

Figure 3.8 presents a summary of all the perspectives described above.

3.4 Visualization in the Automotive Domain

After understanding datasets, tasks and tools, we were specifically interested in the current use of visual representations in order to better understand engineers' usage of graphical data representations and existing mental models about the data. Based on a literature review, this section starts with a general overview over related work from visualization in automotive mechanical engineering. After that, I present static visual representations of vehicle electronics that we found engineers using during our observational studies. The section concludes with some attempts of visualizing in-car networks we encountered, a discussion of problems with these approaches and a derivation of lessons learned for future InfoVis/VA tools in this domain.

3.4.1 Related Work: Visualization in the Automotive Domain

In contrast to automotive electronic engineering, mechanical vehicle engineering such as car body design or engine construction is characterized by a high utilization of interactive visualization

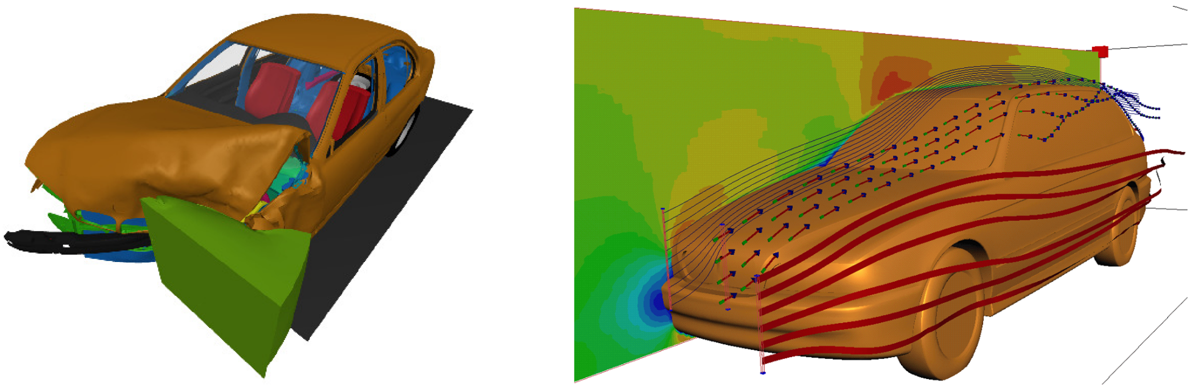

(a) (b)

Figure 3.10: Two examples of SciVis used in the automotive domain: **(a)** Visualization of a car crash simulation[SRE98]; and **(b)** flow visualization of a particle trace with lines, ribbons, glyphs and an isosurface [SRBE99].

and much scientific work has been published in this scope. These visualization techniques are usually based on a 3d representation of the vehicle or parts of the vehicle and are most commonly used in the context of computer-aided-design (CAD), virtual reality, and scientific visualization (SciVis) [Ste07]. When automotive engineers talk about visualization, they often refer to these approaches as they are already embedded in daily working practices and rarely think of abstract representations as used in InfoVis.

Figure 3.9-a shows a screenshot of CATIA [Das] as an example of a widely used CAD tool. CAD tools provide engineers support in designing, compiling and reviewing mechanical components, as well as in simulating and analyzing behavioral aspects of these mechanical components [Lee99]. Over the last decades, mechanical simulations and SciVis of the resulting data have become highly valuable tools in the automotive domain. They allow engineers to predict physical behavior of complex technical systems and to re-engineer mechanical components without implementing them [SRE98]. These techniques are often referred to as virtual prototyping [DFF+96] and most commonly harness CFD-based (Computational Fluid Dynamics) simulation of aerodynamics, acoustics or vibration (e. g., [SRE98, SRBE99, Dha96]), or FEM-based (Finite Element Method) crash simulations (e. g., [RE, KHSE98, SWS07]). For visualization purposes usually color coding of vehicle components is used. CFD simulations additionally use (optionally color-coded and animated) glyphs, ribbons, streamlines or isosurfaces for flow representation purposes. Figure 3.10 shows two typical examples of SciVis usage for simulated automotive data.

Recently, such techniques have also been combined with InfoVis techniques such as scatterplots and histograms for the analysis of, for example, a Diesel exhaust system [DMG+04, DGH03]. Figure 3.9-b shows a screenshot of Doleisch et al.'s system. A larger focus on InfoVis was also applied by Konyha et al. for the analysis of simulation data for car engine development [KMG+07].

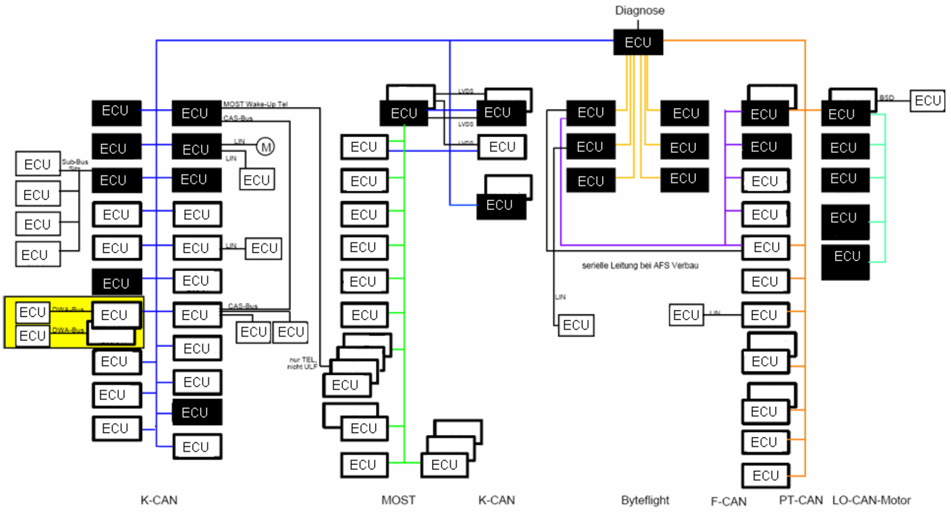


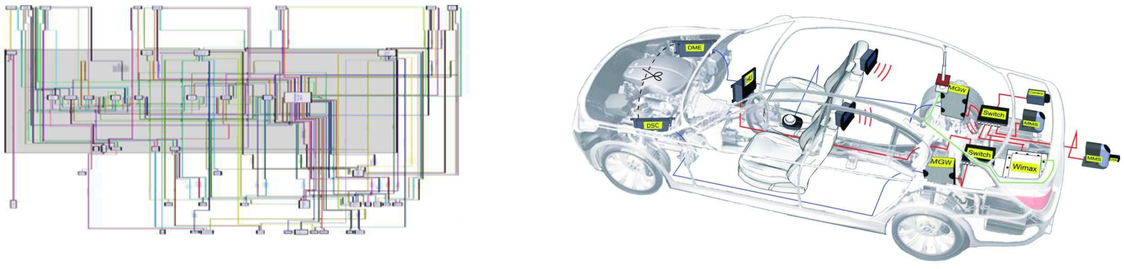
Figure 3.11: Most frequently used visual representation during in-car network engineering: A topology map of an in-car communication network with ECUs as rectangles and bus systems as connection lines (Original labels have been changed due to IPR restrictions).

However, despite the growing importance of electronic engineering compared to mechanical engineering [BG07a], still considerably less work has been done to visually support electronic development and testing of automobiles.

3.4.2 Currently Used Visual Representations of In-car Networks

While the examples above gave a general introduction to literature about CAD, SciVis and InfoVis approaches in automotive mechanical engineering, we found no particular literature about visualization of in-car communication networks. However, in-car network engineers indeed use visual representations in their daily work environments. Based on our observations and interviews, as a next step I discuss current working practices of engineers in terms of frequently used static and interactive visualizations. This knowledge is worth taking into account when designing novel visualizations as these representations show the current way engineers think about the domain (mental models) and how they communicate information. In the following, I present visual representations frequently used by in-car network engineers:

Topology maps: Topology maps (see Figure 3.11) are static, node-link representations of the physical in-car network showing ECUs (nodes) and bus systems (links). Typically, they are manually prepared with Microsoft Powerpoint by a dozen of engineers after but also during the specification of physical in-car networks for a novel car series. Topology maps represent the entire network including ECUs and bus systems of all possible variants (150-percent specification) and are widely used over the company by nearly all engineers dealing with in-car network development and analysis, but also by engineers developing novel systems or suppliers implementing software for a specific ECU. Two of our observed participants even used the topology maps as



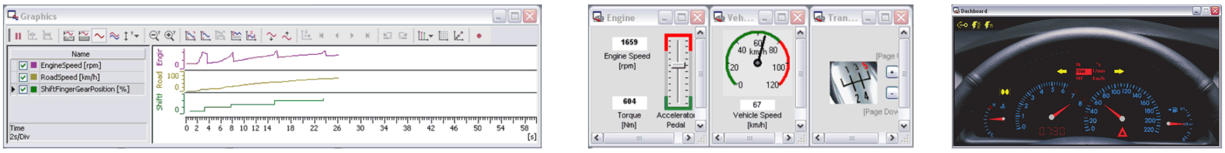
(a) (b)

Figure 3.12: Other visual representations of in-car networks: **(a)** A connection map of a subset of functional blocks; and **(b)** ECUs and bus systems embedded in a 3d model of the vehicle.

wallpapers on their laptops for quickly accessing the map at any time. Topology maps are used for getting an overview of the entire physical network, understanding the wiring of ECUs to bus systems, and to reproduce cross-correlations within the system.

There are no strict layout rules for topology maps. However, over the years several implied guidelines for drawing these graphs have been arranged: Based on an orthogonal graph layout, each ECU is schematically represented by a rectangular box and bus systems by different colored, orthogonal lines connecting the ECUs. The colors typically code the type of a bus system, e. g., blue for K-CAN bus or green for MOST. Optionally, ECUs are color-coded according to basic (black) or optional equipment (white). Mutually exclusive alternatives of an ECU are represented by stacking them. Similar to a top-down hierarchy the layout is roughly organized as follows. Usually, the central gateway ECU is positioned at the top center. Below, based on a specific order from left to right, all ECUs are clustered together into (double-)columns according to their bus connectivity. Then, point-to-point connections and local bus systems such as LIN bus systems are added. Labels of the ECUs are put into the particular boxes, labels of bus systems are put at the very bottom. During the design process, aesthetic criteria, such as minimizing line crossings, bends and area and maximizing symmetries, are manually adapted by the topology map designer.

Other orthogonal diagrams: According to the topology map, which we found the most important static network representation in our studies, several similar approaches of manually generated connection diagrams exist. Figure 3.12-a, for instance, shows a connection diagram of dependencies between a subset of functional blocks within the greater network. Very similar to topology maps, these diagrams are typically characterized by an orthogonal layout with color-coded lines and are used for documenting, communicating and specifying system behavior. For the same reason many engineers use standardized visual modeling and specification techniques and languages, such as UML Communication Diagrams, Sequence Diagrams or State Machine Diagrams. Also, famous automotive engineering tools widely use such visual representation techniques, for instance, Matlab Simulink [The] or PREEvision [Aqu10]. Engineers are very familiar with this kind of representations. However, these systems are more used by system developers to specify local behavior rather than mapping the entire network on a visual representation.



(a) (b) (c)

Figure 3.13: Typical visual representations in trace analysis software: **(a)** signal plots showing the change of three signal values over time; **(b)** gadgets for engine speed, vehicle speed and current gear; and **(c)** a virtual replication of the instrument cluster with speedometer, fuel gauge, blinker, etc.

3D-Model network visualizations: Another way of visually representing the in-car network is a static 3D representation where a quasi-realistic physical network is embedded in a fully or partially semitransparent 3D vehicle representation (see Figure 3.12-b). These pictures are always manually generated with specific 2D and 3D drawing programs and were very popular among our study participants but rarely used in the daily work. While many engineers argued in our studies that the basic benefit is the realistic positioning of the network structure within the car, in the majority of cases, the actual reason, however, was a nice and easy communicable representation used for publication, presentation or advertising purposes. Abstract representations of the network (topology maps) were used far more frequently because they code more relevant information, are arranged more clearly and are directly tailored to support engineers visual tasks [War08]. Furthermore, the manual generation of an orthogonal graph is much faster and easier than a comparable 3D representation.

Interactive visualizations: All visualizations described so far were mostly static representations and therefore did not harness the power of interactive data exploration—one of the basic advantages of InfoVis/VA tools. During our studies we found only a few tools using interactive visualization to support engineers in data exploration and understanding.

In addition to the increasing spread of CASE-tools using UML-like graphical representations for functional network specification (see above), the tools using interactive visualization most intensively were trace analysis tools. Current analysis tools are highly sophisticated in using multiple (partially coordinated) views in order to provide the analysis engineers with varying perspectives on the data. Visual approaches, supporting engineers in investigating and understanding traces, however, are still rare. They are basically restricted to (a) zoomable signal plots showing value changes of one or several signal(s) over time (see Figure 3.13-a) and (b) small, animated virtual gadgets representing real gauges such as speed indicators and updating according to navigation in or replay of time (see Figure 3.13-b and c). While these visualization approaches turned out to be very helpful for retrace specific signal changes over time and also to draw correlations between several signals, there is still much information hidden beyond these representations, for instance: What about timing behavior of messages, what about message distribution, what about cross-correlations and global dependencies within the network?

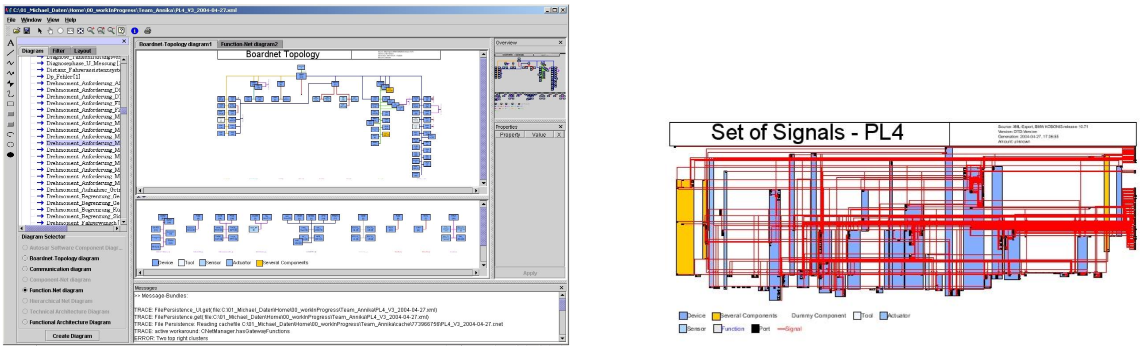


Diagram Selector
○ Autosar Software Component Diagr...
○ Boardnet-Topology diagram
○ Communication diagram
○ Component-Net diagram
● Function-Net diagram
○ Hierarchical Net Diagram
○ Technical Architecture Diagram
○ Functional Architecture Diagram
Create Diagram

```
>> Message-Bundles:

TRACE: FilePersistence_UI.get( file:C:\01_Michael_Daten\Home\00_workInProgress\Team_Annika\PL4_V3_2004-04-27.xml)
TRACE: FilePersistence.get( file:C:\01_Michael_Daten\Home\00_workInProgress\Team_Annika\PL4_V3_2004-04-27.xml)
TRACE: File Persistence: Reading cachefile C:\01_Michael_Daten\Home\00_workInProgress\Team_Annika\cache\773966756\PL4_V3_2004-04-27.cnet
TRACE: active workaround: CNetManager.hasGatewayFunctions
ERROR: Two top right clusters
```

(a) (b)

Figure 3.14: Screenshots of BNVis: **(a)** A Screenshot with a semi-automatically generated topology map; and **(b)** a signal diagram connecting several functional blocks automatically generated with BNVis.

3.4.3 Examples of Previous Visualization Attempts

While visualization is known as a highly valuable tool supporting automotive mechanical engineering for a long time, with increasing complexity of networks, recently it also gained increasing attention in the electronic engineering domain. According to our experience, the sparse diffusion of visualization tools and techniques in this domain (see Section 3.4.2) does not necessarily reflect the current interests and efforts in visual based solutions. Indeed, on the one hand there are still many engineers sticking to traditional text-based techniques. However, on the other hand many of them now start to rethink traditional practices and judge visualization as a promising approach to master the complexity of modern vehicle electronics. A variety of research and development projects on novel tool chains for automotive electronic and network engineering have been conducted, some of them directly addressing visualization as one important aspect in their agendas. However, as stated above, few of these solutions, especially interactive and automated solutions, have yet found their way into the daily practice of engineers.

During our three-year field experience we found several of these attempts of integrating visualization tools into daily working practices of engineers. By providing three of these examples and the reason why they have not been adopted to daily work practices, we want to demonstrate threats and pitfalls in designing InfoVis/VA tools in this domain. In doing, so we want to support designers of future InfoVis/VA tools (both coming from an automotive background and from an InfoVis/VA background) by helping them to understand specific circumstances and to avoid them from running into similar problems (see also next subsection).

1. Example: BNVis—Semi-Automatically generated connection maps

A general drawback of manual generated information graphics, such as the topology maps, is the fact that they have to be redrawn each time the underlying data changes. An obvious approach to counterbalance this drawback is automatically generating static representations. Alongside our pre-design studies for RelEx (see Section 4.3) we found out about a former in-house tool called BNVis that automated the layout process of visual connection maps. BNVis was built by an external software development company in collaboration with the BNE engineers at BMW. It is a standalone tool using exported data from the BNE to generate topology maps, functional network diagrams and other static diagrams.

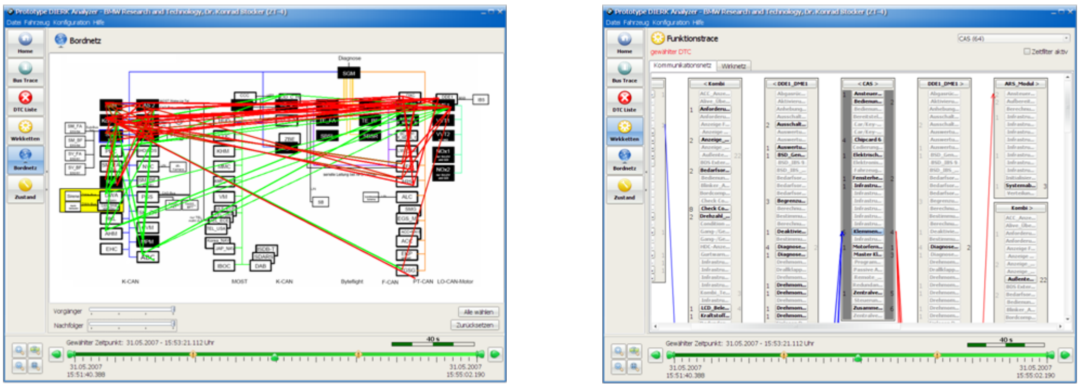
The first step using BNVis is to load a specific data export file from the BNE. For topology maps, then a set of graph layout rules (similar to the ones described above) wis applied to automatically compute a draft of the map. After that, the user has the opportunity to fine-tune the layout by applying common graph editing technologies, filtering techniques and finally exporting a static topology map (see Figure 3.14-a). For functional network diagrams (and all other representation techniques), the user selects a sub-set of the imported data, e. g., functional blocks, and BNVis fully-automatically converts it into a static graphical representation (see Figure 3.14-b).

A change in BNE data export formats in 2008, however, resulted in incompatibility problems between BNE and BNVis. Due to this technical problem BNVis could not be used any longer. Interestingly, asking engineers about BNVis, often revealed positive feedback first, especially about the automatic topology map creation. However, by asking more deeply about why then technical obstacles have not been overcome in order to pursuing BNVis, it turned out that BNVis actually had been rarely used—infrequently for topology map generation and virtually not at all for fully-automatic diagram generation—and therefore was not missed in daily working practices after the BNE interface changes. There were several reasons for this: (a) The impact of automatic topology map generation was little as changes to them are often of minor nature and as the constructing entirely new maps is rather infrequent; (b) Creating topology maps manually was only slightly more time-consuming than automatically generating and fine-tuning the layout with BNVis; (c) manual changes to topology maps could not be preserved when reloading data; and (d) fully automatic generated diagrams (e.g, functional network images) were hard to use and to understand as the representation of a useful amount of information (e. g., functional blocks) resulted in high visual clutter (see Figure 3.14-b). These representations therefore had virtually never been used by engineers.

2. Example: An interactive visualization prototype

Figure 3.15-a and b show two examples of interactive visualizations which we found when starting our very first cooperation with engineers in a research project on future analysis tools (for more information about our follow-up visualization please see Section 5.2). Already before we started to work on this project, information visualization was one of the project goals and therefore along with a novel computation algorithm two visualizations had been implemented.

Both visualizations were designed to gain insight into dependencies between different ECUs as well as dependencies between functional blocks. The general idea was to let user select an ECU or a FB and subsequently to show all it’s predecessors and successors (cf. Section 5.2.1). Although the general idea behind the visualization approaches was good, there were major shortcomings in design and implementation. Visualization a (see Figure 3.15-a) was based on a topology map and tried to show predecessor and successor ECUs by connecting them with red and green lines overlaid on the map. Some of the apparent shortcomings were: (1) Red/green color



(a) (b)

Figure 3.15: Visualization approaches in a previous automotive research project: **(a)** Topology map with overlaid connection lines to predeseccors (red) and followers (green); and **(b)** parallel coordinates approach showing dependencies between functional blocks.

coding will cause problems for color blind people (especially in a domain where more than 90% of target users are male) [War04], (2) color coding of lines in topology map and the overlay visualization interferes, and (3) high clutter by overlying lines. Visualization b (see Figure 3.15-b) took a parallel coordinate oriented approach, redundantly putting (all) FBs in parallel, vertical lists and interactively connecting them according to their correlations with blue (predecessors) and red lines (successors). The most obvious problems with this visualization were large lists of up to 3500 FBs per column resulting in endless vertical scrolling when trying to learn about dependencies. Based on these visual design shortcomings, both visualizations never found their way into productive tools.

3. Example: An alternative approach for topology maps

Also during our pre-design studies on RelEx (see Section 4.3), we talked to several engineers reporting of a failed project which proposed an alternative topology map representation claiming a more efficient data representation. The general idea behind this visualization approach was to invert the visual representations of ECUs and bus systems: Rectangles became bus systems and labeled lines became ECUs. As there are usually up to ten times more ECUs than bus systems, the designers wanted to save valuable space and to reduce visual clutter with this novel layout. When the designers presented their approach to end users (the engineers we later collaborated with), it turned out that they strongly disliked it: *“ECUs are rectangles and bus systems are lines! It had always been like that!”*. After that presentation the project was canceled ahead of schedule. The basic mistake in this example was that engineers’, i. e., end users’ practices, implicit conventions and their mental models were completely neglected. Indeed, from a pure InfoVis perspective the designers approach was valid—according to Tufte’s data density measure [Tuf83] it could even be argued to be better than traditional topology maps (less area for representing the same data)—however, disarranging prevalent mental models made it unacceptable for automotive engineers.

3.4.4 Threats: What can we Learn from these Examples?

To summarize what we can learn from the examples above, we derived three kinds of threats: InfoVis-design threats, domain-design threats and domain-problem threats. To develop successful InfoVis/VA tools in our domain it is necessary to be aware of each kind of these threats.

InfoVis-design threats: InfoVis-design threats describe shortcomings in visualization, interaction and interface design that could be counterbalanced by InfoVis expertise. In the examples above we saw several of them, e. g., red-green color coding, usability and scalability problems or visual cluttered displays. As visualization tools usually are recognized by domain experts as “just additional supporters” which are not intrinsically and crucially needed for their everyday work, such shortcomings consequently resulted in very low tool acceptance rates. We encountered two basic reasons for InfoVis-design threats:

1. *Disregard of InfoVis and HCI in software projects:* From HCI is known, that especially in-house software development often lacks in sufficiently addressing usability testing [HBH09, BÅPL03]. Based on our experience, this also holds unchanged for information visualization. In the projects we encountered, often the core components of a software tool were assessed many times higher as user interface and visualization components. Postponing visualization and interface design to the very end of a project often led to a mismatch between software core components and visual representation resulting in imperfect visualization solutions.

2. *Missing visualization expertise in software projects:* Even if there was the will for a stronger focus on visualization, often InfoVis/VA expertise was missing in projects. Instead of InfoVis or HCI practitioners, either software engineers designed and implemented the visualization in addition to their regular tasks or the visualization was directly designed by domain engineers. Though knowing the problem very well, however, their expertise in visualization design (in both cases) was usually rather low.

Much has been written about designing good InfoVis tools (for general introductions, see, for instance, [CMS99, War04, Che04, Spe07]). Either taking this literature into account, or having an InfoVis/VA expert in a project team, will help to design better InfoVis tools.

Domain-design threats: Domain-design threats can be best explained by the third example from above (an alternative approach for topology maps). In contrast to the previous threat, from an InfoVis point of view no wrong design decision had been made. However, the visualization failed because end users’ practices and conventions have not been sufficiently taken into account. Solely being an InfoVis expert therefore is not enough. Additionally, it is highly important to understand users’ mental models and to derive the right design decisions accordingly [Pal09]. To overcome this threat a close cooperation with end users during the design process is invaluable. Much work has been done on close user integration in user-centered, contextual and participatory design research [ND86, BKS04, BH98] and have also been adopted to InfoVis [WKVD^{+}08].

Domain-problem threats: The last type of threat we observed with our examples are domain-problem threats. This category is derived from Munzner's domain threats [Mun09], describing that an InfoVis tool was designed to solve the wrong problem:

> "[...] the assertion is that particular problems of the target audience would benefit from visualization tool support. The primary threat is that the problem is mischaracterized: the target users do not in fact have these problems." [Mun09]

To overcome this problem Munzner suggested applying participatory and user-centered design approaches. In the automotive domain, applying such techniques is even more crucial as the degree of expertise is very high and conducting them must be very carefully planned and designed. With BNVis, for instance, the semiautomatic generation of topology maps was highly demanded and also designed in close cooperation with automotive domain experts in iterative design cycles. However, although carefully designed (and in terms of the semi-automatic topology map generation also well implemented), BNVis has not been adopted by target users (downstream validation, according to Munzner). This fact poses additional questions beyond solving the "the wrong problem" about the "quality of the domain problem":

- How important is the domain problem?
- Which are the important facets of a problem?
- How much benefit can a novel solution add over traditional solutions (delta)?
- Who is interested in solving this problem (stakeholder, end user)?
- Who are the right end users to ask?
- Is the opinion of the end users I collaborate with shared by many users, or is it just an individual opinion or the opinion of a small group?
- Can I believe what the end users say to me?

In order to design successful visualization tools in the automotive domain (and also in large company settings in general), carefully thinking about these questions is essential. Cooperation with domain experts does not implicitly guarantee the right answers nor does it guarantee a successful visualization tool design. Often, much energy is spent in building a sophisticated tool that in the end solves—not necessarily the wrong problem, but—not important enough problems to be adopted by end users. Carefully designed pre-design studies such as conducted in ethnography or social-science [IZCC08] can help to actively address these challenges. Understanding a problem and assessing its importance/quality in the automotive domain, however, remains challenging.

3.5 Expert Opinions about InfoVis/VA

For us it was very important to design InfoVis/VA solutions not just for but together with domain experts (see previous section and Section 1.3). To do so, it was necessary along with understanding domain challenges to gradually introduce InfoVis/VA technologies and to learn about domain experts' estimations on the potential value and imaginable use cases. We also used this InfoVis/VA introduction to trigger fruitful discussions about current gaps in understanding in-car networks in order to better understand starting points for our tools. In doing so, we gathered a list of potential use cases and a list of current gaps in understanding in-car networks. Then, we conducted a focus group with five carefully chosen participants and broke down these lists to the three most important, InfoVis/VA related domain challenges in order to clearly focus the thesis scope within the larger area of in-car network engineering.

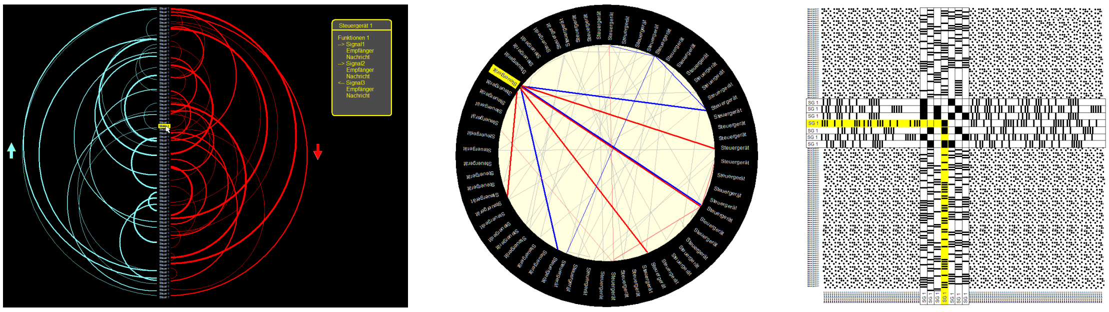
3.5.1 Methodology

The results described below are directly derived (a) from our diverging studies during the first year and (b) from a converging focus group discussion. For the temporal context of these studies please refer to Figure 3.1 on page 24. We used the following methodologies:

Diverging Phase (Interviews, Observations, Presentations, Informal Discussions): During our diverging studies the goal was to talk to a variety of automotive engineers with different backgrounds. This phase was split in two sub-phases, the initial phase where we communicated InfoVis basics and goals and the idea phase where we tried to trigger additional ideas by presenting concrete InfoVis/VA approaches for the identified problems.

In the initial phase, we conducted guided interviews with four automotive engineers working at the company's research department. Each interview took approximately one hour, and was structured according to the following aspects: (1) Introduction to InfoVis/VA techniques, interview about (2) previous experience with visualization, (3) potential application scenarios and user groups for visualization, and (4) requirements for visualization tools. Additionally, informal discussion with another twelve automotive researchers with backgrounds in different in-car network engineering domains helped us to form our initial understanding.

In the idea phase, building up onto the identified problems, we sketched different InfoVis/VA drafts and in doing so tried to induce new ideas in discussions with domain experts. Additionally, we intended to promote InfoVis/VA to a broader audience of automotive engineers in the company. As artifacts, several static sketches (see Figure 3.16 for three examples) were designed and a first working prototype was implemented (see Section 5.2 for more details). These artifacts were used in several studies gathering additional input on use cases and understanding challenges. These studies in particular were: (1) A paper prototype user study with three lead users (1.5 hours per study, discussion of 11 different design sketches, first working prototype has not been finished at that time), (2) four presentations with subsequent group discussions (ranging between 15-45 min; 6-12 participants with automotive researcher and/or engineering



(a) (b) (c)

Figure 3.16: Three early idea sketches to visualize in-car communication networks and used for explaining the ideas of InfoVis/VA: **(a)** An arc diagram [Wat02] showing all ECUs in a vertical list with blue arrows for predecessors (left) and red ones for followers (right); **(b)** a circular layout derived from [Hol06] showing dependencies between ECUs; and **(c)** a Matrix approach to visualize the network similar to [HF06].

background, with sketches and prototype), and (3) again, informal discussions with seventeen additional automotive engineers with different backgrounds.

During the studies we collected a variety of notes serving as a basis for the results described in the following sections. We intentionally dispensed with audio and video tracking for several reasons: First, according to potential risks of industrial espionage most automotive companies have strict IPR restrictions. Conducting a user study with audio or video recording therefore requires to apply for a specific license upfront. Depending on where and with whom you want to conduct a study, this can be a tedious undertaking. Second, for the same reason participants may be reserved when audio or video tracking them. Our experience showed, that depending on your participants recording technology can pose artificial situations, for instance, as stated by a participant in a later study, *"oh, audio recording, so I have to be careful what I'll tell you"*. Third, many of our studies were informal and spontaneous, e. g., meetings at the cafeteria or having a joint lunch. This casual atmosphere provided us a good platform for reaching numerous, usually time-restricted engineers, however, it obviously hinders recording conditions. A detailed and general discussion about note taking vs. audio/video tracking can be found in Dix et al. [DFA04]. As a basic results two lists were derived, a list of potential use cases and a list of current understanding gaps, which are presented in Section 3.5.2 and 3.5.3.

Converging Point (A Focus Group): The basic goal at the converging point was to break down the collected information to the three most important InfoVis/VA challenges within the larger scope of in-car network engineering. To do so, a two-hour focus group with five team leaders (all male) was conducted. The participants were carefully selected according to several factors: (a) Their future-orientation (at that time, all of the participants worked at company's research departments), (b) their experience in the field (working at least 6 years at the company, heading teams of at least 5 engineers, four participants worked at company's engineering departments before moving to research), and (c) their varying background in automotive electronics (Five different backgrounds: future network design, future software architecture, future analysis methodologies, design of networked camera systems, design of novel navigation systems). During the focus group first a general introduction to InfoVis/VA and a presentation of the sketches and the first prototype was given. Subsequently, all use cases and understanding gaps found in the diverging phase were presented to the participants, discussed and then pre-evaluated according to their importance for the company and to their prospective impact. Finally, the three major domain challenges were voted (each participant had five votes) and discussed subsequently. During the focus group we took notes and also encouraged our participants to write down comments which we collected afterwards. The results of the focus group are presented in Section 3.5.4

3.5.2 Results of Diverging Phase: Potential Use Cases

During our diverging studies we discussed intensively with our subjects about what we can do with and who we can support by using InfoVis/VA. The following list shows the ideas and wishes of our participating engineers to use visualization in the in-car network engineering domain:

Diagnosis/Analysis: An idea frequently mentioned was to use InfoVis/VA to support analysis and diagnosis engineers in debugging and tracking down errors in large network traces. Traces can be become tremendously large (see last section) and especially when errors are more complex and distributed finding them can take much time and effort. Current analysis tools are basically text-based and lack in providing insight into correlational and more global aspects. Our subjects estimated InfoVis and especially VA as highly valuable approaches to provide new perspectives on traces and in doing so to support analysis engineers in tracking down errors.

Onboard-monitoring: Similarly, several of our participants mentioned that they would like to use InfoVis/VA technologies for onboard-monitoring. In contrast to the use case above, the trace is not recorded and analyzed afterwards but is directly and live visualized in the car either by using a laptop connected to the external bus connection or by utilizing the vehicle's display properties.

Simulation: During the research on and the development of novel in-car network technologies, parts of the network or the entire network topology can be simulated upfront in order to learn about real system behavior without implementing its hardware components. Simulations produce basically statistical data but also trace files as described in the previous example. According to our subjects' opinions, InfoVis/VA would be helpful to analyze and understand these datasets. Yet, few of these network simulation techniques are integrated into the development processes, however, especially to the estimation of one of our participants this *"will become tremendously important in future in-car network development"*.

Database investigation and exploration: Many of our study participants complained about having difficulties when working with the BNE interface. Frequently mentioned objections were problems to find the right information in the right moment and problems in understanding correlations between elements beyond their trivial connection (e. g., finding out about function block on an ECU was judged easy, finding out about the relation between several different ECUs was mentioned to be complex). According to their estimation, having a more visual and easier to use interface would help to overcome these problems.

Collaboration/ Communication: Another use case for applying InfoVis/VA in this domain is to support collaboration and communication. The development of in-car communication networks is characterized by a high degree of expert specialism and collaboration between experts. Therefore, engineers very frequently have to communicate within and beyond departments and misunderstandings in these communications often cause expensive errors. InfoVis/VA can be used to prevent misunderstandings in conversations between development engineers by providing clear representations and common understanding.

Presentation/ Communication: Especially when talking to engineers who design and develop novel electronic system or network technologies, presentation and communication becomes important. There are many novel creative ideas and developments in the automotive area, however, only a subset of them makes it into the final product. Several of our participants mentioned that they would find InfoVis/VA tools helpful to properly communicate but also promote their ideas. Therefore, this use case refers back to the previous one, however, with a different focus on communication between engineers and decision makers and therefore a higher interest in striking and promoting representations.

Management of changes: Developing and designing novel network architectures means evolutionary development and therefore numerous changes on the system over time. As the complete system is very large, looking over all changes made by several hundreds of engineers is challenging. According to our subjects' opinions, InfoVis/VA tools can help to better reconstruct changes and to understand the history of such processes.

Management of variants: Modern vehicles are characterized by a high degree of functional personalization. Each customer can configure her/his car's functionality individually. According to a statement of an analysis engineer only 3 out of 1,000 produced cars today may have exactly the same electronic configuration (except vehicles fully equipped with all options). The different configurations of functionality poses different requirements to in-car communication networks and can lead to unpredictable behavior or even errors. The influence of the different variants on in-car network requirements is not yet fully understood and poses numerous challenges to development and analysis engineers. Engineers estimate InfoVis/VA a potential candidate to help shedding more light at these aspects.

Training/Instruction: Working in the area of in-car communication engineering requires a high degree of specific expertise. Learning and understanding all relevant facets of the field is challenging and often tedious for novice employees. Most of the time, this highly practical knowledge is not covered by university education and therefore company internal trainings and more importantly face-to-face communication between experts and novice employees are used to pass on the knowledge. According to our study outcomes, InfoVis/VA tools could potentially help to shorten this process and to simplify understanding complex issues by making data more graspable and by externalizing knowledge of expert employees into visual representations.

3.5.3 Results of Diverging Phase: Lacks in Understanding In-Car Communication Networks

As a second result from our diverging studies, we derived a list of lacks in engineers' current understanding of in-car communication networks. We were particularly interested in directions where InfoVis/VA can help to shed more light on currently unknown aspects or to aspects which can hardly be derived with current, text-based technologies. To do so, we asked our participants which information would be helpful for their work currently not or not sufficiently available, and more generally about their opinion where understanding of in-car communication networks lacks with present practices. The following list shows the results:

Understanding more comprehensive aspects: This was the most frequent answer of our participants and can be seen as a superordinate concept for all lacks listed below. Finding a specific value in the BNE or in the specification PDF, finding a defined bit error in a trace or looking up a single cell in an Excel sheet is rather simple. However, the text-based representations meet their limits when trying to understand more comprehensive aspects in the data such as correlations, clusters or dispersion—typical situations, however, where InfoVis/VA can provide novel insights beyond textual representations.

Understanding entire functional network: Understanding the structure of the physical network can easily be conducted via topology maps (see Figure 3.11). However, there is currently no comparable and working approach for functional networks. Our participants mentioned that it is hard to get a big picture of the functional network, to gain insights into global correlations on this layer, to better understand distributed functions or to evaluate the role of such functions within the entire system. Recent attempts such as BNVis (see Section 3.4.3) tried to tackle this problem basically by providing semi-automatic generation of static visualizations, however, failed due to cluttered representations and poorly interactive, hard-to-use interfaces.

Understanding dependencies and correlations in the network: Current tools and representations lack to show dependencies and correlations in the functional (see also previous aspect) as well as in the physical network. Considering the physical network, the topology map is highly adequate to show how ECUs and bus systems are connected together. However, based on current data representations it is hardly possible to learn about the actual correlations and dependencies between the network components such as: Which ECU communicates with which ECU in which intensity using which signals/messages? Or: Which messages/signals are sent via which bus? While the topology map is a very helpful top-level representation, easy to read, commonly known and well understood, it, however, cannot provide insights into the variety of aspects of correlations and dependencies in this network. It should not be the goal to replace the topology map but to complement it with other interactive, more powerful tools.

Understanding communication processes beyond trace list and signal plots: Talking about the actual communication processes in the vehicle is intrinsically tied to the discussion of analyzing traces (see Chapter 5) which are most commonly represented in message lists, signal plots or virtual gauges (see Section 3.4.2). While these representations are good for scanning single values and for tracking changes in signals over time, talking to analysis engineers, however, showed a variety of aspects beyond that, especially more global and comprehensive aspects, where gaps in understanding exist. Thus, by using current technologies it is hard to understand how signals and information spread out over the system, how messages inter-correlate and which timing combinations cause erroneous communication situations. Also, aspects such as cyclic messaging or shifts in distributing messages over several bus systems become important but are not completely understood yet. Our analysis expert participants stated that providing novel views on traces for a deeper understanding of these aspects would help them *"enormously in tracking down complex errors beyond [these] simple single bit errors"*.

Understanding differences between actual network behavior and specified behavior: Another lack in understanding which was frequently mentioned by analysis experts in our studies is the gap between specified system and the actual behavior in the network. An implicit approach of this is implemented in the car where conditions for automatic error detection are specified based on the vehicle specification. However, if manually detected errors occur (e. g., a window do not open, but there is no explicit error entry) it is very hard to draw a line between the specification and what actually happened in the car. Currently, all connections between these two data sources have to be done in the heads of engineers by reading and combining information from both worlds. This additional workload, however, handicaps understanding all relevant differences which in turn could help better and faster detecting errors in vehicle electronics but also in specifying documents.

Understanding relation between mechanical and electronic information: The last aspect which was mentioned frequently during our studies was a lack of understanding of correlations between mechanical and electronic information. Observing the current situation in our automotive company, showed that mechanical engineers (CAD, etc.) and electronic engineers (e. g., BNE, traces) worked mostly separately. Each of these areas has distinct problems and challenges to solve, and 'divide and conquer' is the aspiration. Different data bases are used focusing on different aspects and interoperability is usually not guaranteed. This situation, however, makes understanding correlations between mechanical and electronic information rather hard. Bridging this gap of understanding could help, for instance, to better correlate messages to their vehicle behavior and in doing so address mechanical aspects in trace analysis. Additionally, a well-grounded understanding of this connection can help to better integrate mechanical constraints into the specification process of electronic systems.

3.5.4 Results of Converging Point: Three Major InfoVis/VA Challenges

While the diverging phase aimed at getting a broad understanding of the field and gathering potential application areas for InfoVis/VA, the converging point's goal was to evaluate the potentials of these findings and to focus the thesis accordingly. Therefore, we took the results from the diverging studies, namely the lists of uses cases and current understanding gaps, into a focus group discussion with five carefully chosen participants (see Section 3.5.1). The central result of this focus group was a set of the three most important, InfoVis/VA domain challenges. The first of these challenges was derived from the use case list, while the second two were taken from the understanding gaps list. The three challenges and the arguments for this choice are summarized and discussed in the following:

1. Challenge—Support (FIT) Trace Analysis: Trace analysis was evaluated by all five participants as one of the most important use cases for applying visualization to. With the increasing complexity of in-car networks, traces got enormously large over the last years and with current, text-based technologies it is highly challenging for test engineers to track down errors, as, for instance, stated by one of our participants: *"[Trace] analysts have kilometer-long trace lists, I'm sure that there are much better ways than just putting every message in a text-based list!"*. Within the area of network testing FIT tests were estimated to be most challenging, as a participant stated: *"[...] the FIT guys have solved the problem of storing all traces. But now they have the problem that they cannot analyze all the traces they store anymore"*. Therefore, the first (and most important) challenge for prospective visualization support was stated to be trace analysis. Within trace analysis, the most pressing need is supporting FIT analysis

2. Challenge—Support Understanding of Dependencies: The second major challenge evaluated within the focus group was the lack of understanding of dependencies and correlations in in-car networks. According to a participant's opinion: *"Current techniques heavily lack in providing insights into dependencies in in-car networks. [...] What we do today is manually drawing images with Powerpoint, if at all."* During the focus group several important examples were listed, which can be summarized in this category. For instance, change management would heavily benefit by a clear representation of network dependencies: *"As a system designer, I must be aware of what I can change and what I rather not change [...] It is crucial to have a global understanding of the [functional] network in order to make the correct design decisions. [...] Currently this is only in the heads of the engineers"*. Discussing about drawbacks of current techniques, another participant mentioned: *"Imagine I have two functions*[6] *and I want to know how they communicate [...], with the BNE it is really hard to find out, in my opinion a visualization would be much more helpful."* On the other hand, a better understanding of dependencies was also estimated to help test engineers in tracking down errors: *"[...] the point where an error is detected, often is not the real source of the error. It has appeared somewhere much earlier [...] currently tracking down such errors is more a brute-force activity. Knowing exactly how the information spread out, or the path an erroneous message tooks through the network, would help enormously to improve testing practices."*

3. Challenge—Support Understanding of Correlations between mechanical and Electronic Information: The last major challenge worked out during the focus group was 'bridging the gap between mechanical and electronic information'. Currently, *"the three worlds [i. e., mechanical, electric and electronic engineering] are completely autonomous. We have different data sources, we use different tools and we have different practices [...] A central question for the future will*

---

[6] The more precise term would be functional block, however, the term function is often used ambiguous, see Section 3.2.

be, how we can build a common ground of understanding and how we can bridge this gap." According to four focus group participants these techniques should be based on 3d models of a vehicle. A variety of use cases for such 3d visualizations were mentioned in the focus group, e. g., *"If I press this button here, it would be interesting to see what's mechanically and electronically happening in the car then?"* Or for error diagnosis: *"errors are not always of purely electronic nature, e. g., [...] a screw driven through a specific cable [...] if I had a 3d visualization showing me the path the message took through the car I would sooner take into account whether this might be the reason."* Or for specification purposes: *"Currently information is distributed widely over the car, e. g., a camera sends its information first to an ECU in the back and then to the front again and [...] mechanical information is not really taken into account when designing functional networks, but I am sure this can help in early stages to better understand physical constraints and to design better network configurations based on this knowledge."*

There have also been other challenges discussed and estimated as important such as visual specification documentation, visualization of change histories or application flow visualization. However, in order to get a clear focus for the thesis the main goal of the focus group was to narrow down the three most important challenges and not more.

3.6 Summary, Discussion and Outlook

To this point, I introduced the domain of in-car network engineering and presented the general results of the explorative field studies in terms of data sources, user groups, current tools and use of visualization. The studies particularly showed that current representation techniques in both network development and analysis lack in providing insights into more comprehensive patterns and correlations within the data. By discussing InfoVis/VA technologies with automotive engineers, we found them estimating these techniques as valuable tools for a variety of use cases and subsequently evaluated the three most important use cases with five carefully chosen lead users. To conclude this chapter, I describe how this finally led me to the focus and structure of my thesis and which first design implications I derived from the studies.

3.6.1 Thesis's Main Application Areas

After the focus group discussion, I broke down the thesis subsequent focus to three main application areas (in-depth phase). The decision was primarily derived from the results of the focus group (see Section 3.5.4), however, was also strongly influenced by the classification of user groups with their assigned datasets (see Figure 3.5 on page 32) and our data perspective classification (see Figure 3.8 on page 36). In doing so, the challenges from the focus group was adapted to be more concrete in terms of relation to specific datasets and user groups and led to three main application areas for this thesis:

1. Visualizing traces—to support error tracking (Chapter 5): This application area was directly derived from the first challenge identified in the focus group—support trace analysis. There is a clear underlying data source, trace files, a clear target group, test engineers, and a clear task, tracking down errors. We built several different PTs in cooperation with FIT analysts and I will present the four most important of them in this thesis (Section 5.2, 5.3, 5.4 and 5.5).

2. Visualizing specification data—to support dependency understanding and browsing large specification datasets (Chapter 4): This application area focuses on development engineers working with functional and physical specification data (Excel sheets, BNE, specification documents). Two of the prototypes were designed for a better understanding of dependencies and therefore directly map the second challenge from the focus group (see Section 4.2 and 4.3). The third prototype in this category, MostVis, again was not scheduled in the original project plan but was a engineer-initiated solution[7], collaboratively built with a group of MOST engineers (see Section 4.1). Instead of focusing on dependencies this prototype took the approach of a visual specification document and supports faster browsing and exploring this data.

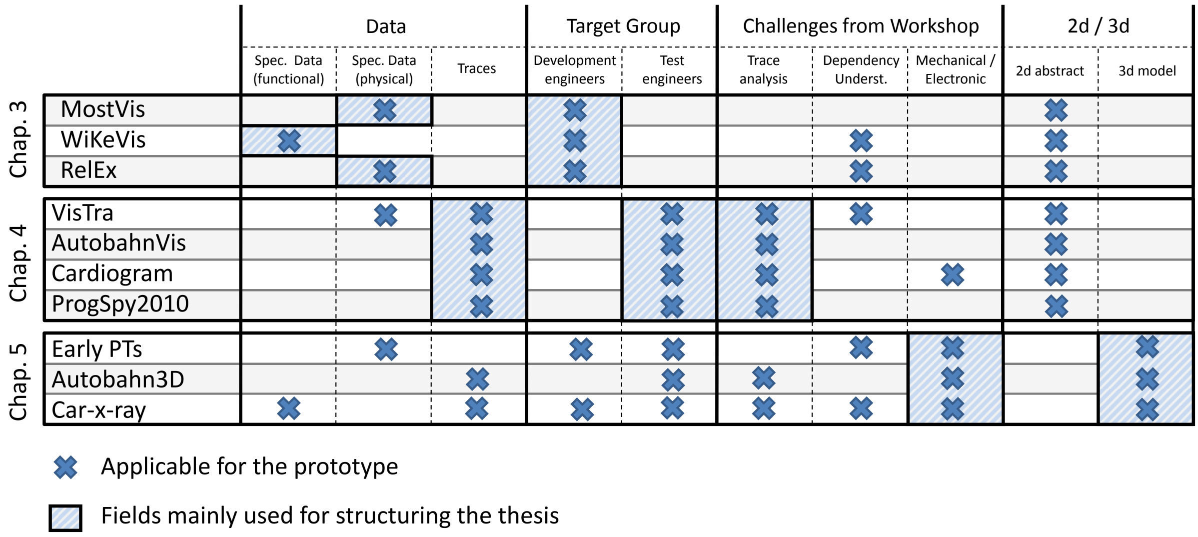
3. Visualizing in-car network data with 3d models—to better understand correlations between mechanical and electronic information (Chapter 6): This application area is very closely related to the third challenge from the focus group. In the focus group as well as during our diverging studies, bridging the mechanical/electronic gap was frequently cited together with using 3d models for visualizing the data to be of high importance for the domain. Based on this strong desire for 3d visualization and on the fact that this thesis was funded to research on this topic, I focused on this application area. To explore the design space, again a set of prototypes was implemented and evaluated. Section 6.1 describes two early approaches which have been developed by BMW before my thesis started, and Section 6.2 and Section 6.3 describe two of our design studies.

Figure 3.17 summarizes all prototypes that will be presented in this thesis and shows the parameters used for focusing and structuring them along this thesis, namely:

- *Data:* Marks which data we visualized in our prototypes: Functional (Excel sheets) or physical specification data (BNE, specification documents) or traces (cf. Section 3.3.3).
- *Target group:* This field shows the target group—development engineers and/or test engineers—we collaborate with during our participatory design approaches and/or which evaluated our tools (cf. Section 3.3.2).
- *Focus Group Challenges:* Shows which of the three main challenges from the focus group is addressed by the particular prototype (cf. Section 3.6.1).
- *2d/3d:* Refers to the question how a prototype visualizes the data, 2d or using a 3d vehicle representation.

---

[7] Solutions initiated and requested by end user contacting us for help.



		Data			Target Group		Challenges from Workshop			2d / 3d	
		Spec. Data (functional)	Spec. Data (physical)	Traces	Development engineers	Test engineers	Trace analysis	Dependency Underst.	Mechanical / Electronic	2d abstract	3d model
Chap. 3	MostVis		▨ ✖		▨ ✖					✖	
	WiKeVis	▨ ✖			▨ ✖			✖		✖	
	RelEx		▨ ✖		▨ ✖			✖		✖	
Chap. 4	VisTra		✖	▨ ✖		▨ ✖	▨ ✖	✖		✖	
	AutobahnVis			▨ ✖		▨ ✖	▨ ✖			✖	
	Cardiogram			▨ ✖		▨ ✖	▨ ✖		✖	✖	
	ProgSpy2010			▨ ✖		▨ ✖	▨ ✖			✖	
Chap. 5	Early PTs		✖		✖	✖		✖	▨ ✖		▨ ✖
	Autobahn3D			✖		✖	✖		▨ ✖		▨ ✖
	Car-x-ray	✖		✖	✖	✖	✖	✖	▨ ✖		▨ ✖

✖ Applicable for the prototype

▨ Fields mainly used for structuring the thesis

Figure 3.17: Matrix showing parameters of the prototypes derived from the findings in this chapter. We took several of them into account to cluster the thesis in three main application areas (blue-striped).

3.6.2 General Implications for Design

Based on the findings of our field studies, finally I want to summarize them in a first set of six general design implications for InfoVis/VA tools for in-car network engineering:

Implication 1: Harness Engineers' Familiarity and Mental Models
Even if currently there are just a few techniques using interactive visualization, engineers utilize many static graphical representations in their daily working practice. Message sequence charts, state machine diagrams, as well as (usually orthogonal) network representations such as topology maps are frequently used for system specification, communication between different groups of engineers and are also supported by frequently used tools such as Matlab Simulink or PREEvision. Engineers also use this kind of representations intensively for sketching and expressing their ideas, e. g., in face-to-face communications, group discussions or in presentations. During our studies we encountered these behavior very frequently and in doing so learned much about the (graphical) mental models of engineers. To make a visualization tool easy to understand and to support communication with and between engineers, it is advantageous to reuse these common representation techniques as they are mental models and metaphors from the engineering domain. Similarly, supporting known interactions and work flows can help engineers to adopt a new tool.

Implication 2: Provide Comprehensive Understanding of the Data
Current techniques tend to lack in the amount of information they represent. On the one hand, there are static high level representations, such as topology maps. While these representations are valuable tools for getting a top level understanding, they cannot provide information into the large specification and trace datasets. On the other hand, this raw data is mostly shown in textual lists, failing to gain insights into correlations, clusters or dispersion. InfoVis/VA technologies provide

adequate tools for overview and interactive exploration of this data. Considering Shneiderman's popular information seeking Mantra: "Overview first, zoom and filter, then details-on-demand" [Shn96], the current situation of representating in-car network data could be described as: "Top level overviews, nearly nothing in between and no direct connection, details en masse". Section 3.5.3 provided a list of understanding gaps we found during our studies, all of them related to the missing understanding of comprehensive aspects of the raw data. InfoVis/VA tools should take these aspects into account and provide task-oriented solutions for better understanding correlations, clusters or other more comprehensive aspects in the data.

Implication 3: The Power of Interactivity—Design beyond Automatic Image Generation
Automatically generated static images are not interactive visualizations! Recent approaches in visualizing in-car networks took this approach and often failed as the added value was little compared to manually drawn images. While certainly cases exist where these automatic graph drawing approaches might be helpful, the real strength of InfoVis/VA, however, is to support interactive exploration of the data [War04]. What seems obvious for an InfoVis researcher or practitioner, however, might not always be clear to a designer or developer of automotive software tools, therefore, we included this rather general implication at this point. During our studies we encountered many sticking misconceptions about information visualization, most frequently, *"InfoVis is 3d visualization"*, *"InfoVis is doing graphics in Powerpoint"*, *‘‘InfoVis is human-car interface design"*, *"InfoVis is UI design"* or *"InfoVis is painting nice pictures"*. Before starting to design a visualization tool for automotive electronics, especially when the designers background is not InfoVis/VA, it might be helpful to remember the InfoVis foundations: "[...] computer-supported, interactive, visual representations [...] abstract data [...] amplify cognition." [CMS99]

Implication 4: Design together: Domain Experts and InfoVis/VA Experts
To support specific domain experts with InfoVis/VA it is important to get a clear understanding of their problem domain [Pal09]. Automotive electronic engineering, however, is a complex area that requires expertise and much background knowledge, and for an InfoVis designer/practitioner it is mostly not possible to understand all relevant details. Therefore, it is invaluable to counterbalance the incomplete domain knowledge through an exploratory study of domain practices and a user-centered or participatory design process. Allowing end users to closely participate in the design process can help designing tailored, well-directed, and valuable solutions. From an automotive tool designer's perspective, on the other hand, it is important to collaborate with InfoVis experts, too. The examples we showed in Section 3.4.3 revealed some fundamental information visualization shortcomings during projects with little or no specific expertise in InfoVis. A constant, close cooperation between domain experts and InfoVis/VA practitioners can help to overcome the threats we derived (see Section 3.4.4) as well as some of the pitfalls outlined in [Mun09].

Implication 5: Take Diversity into Account
Our target group is located in a large company setting where thousands of employees work on highly specified and diverse tasks. This poses many challenges to an InfoVis/VA tool designer. First, it is challenging to find the right people to collaborate with. It is not intrinsically clear

whether your collaborators are representative for a larger group or if their problems are highly specific. A promising approach to counterbalance this is to talk to engineers from different departments. This leads to the second challenge of forming a larger knowledge based on highly specific, even inconsistent domain expert statements. And finally, the designer has to derive concrete design decisions based on this knowledge. For tool designers, it is invaluable to be aware of this diversity in order to take appropriate steps rising to these challenges.

Implication 6: Support Collaboration

As described in the previous implication, our target domain is characterized by a high degree of task specialization and diversity where each single employee acts as “a small cog in a big wheel”. To form a greater understanding based on individual expertise and to master comprehensive challenges collaboration is often indispensable. Being aware of this highly cooperative working environment is important for tool designers in this domain. Already simple design decisions such as allowing to store a specific view status or to annotate a visualization can help to better support collaboration.[8]

---

[8] In the scope of this thesis we only marginally touch the field of collaborative visualization. For more in-depth information on collaboration in InfoVis please refer, for instance, to [HA07] or to [Ise09].

Chapter 4

Visualizing Specification Data

In this chapter, I present three design studies for visualizing specification data and their summative evaluations. Each system was designed in collaboration with a specific target group of development engineers, i. e., engineers working on the left wing of the V-Model (cf. Section 3.3.1) and focuses on different perspectives of the data. The first system I introduce is MostVis, a tool for visualizing large, hierarchically ordered catalogs of MOST specifications supporting engineers in browsing, searching and exploring these catalogs. Based on successful results of our user study, MostVis was closely integrated with current software and deployed by the company. Then, I present WiKeVis that consolidates small parts of the functional network (called "dependency chains") to a larger, hierarchically clustered graph and in doing so provides insights into more comprehensive aspects of functional dependencies and correlations. Last, I introduce RelEx which visualizes signal paths in the physical network based on combining an interactive topology map with a matrix visualization. While MostVis is a design study for visualizing hierarchical data (cf. Section 2.3), WiKeViz and RelEx are examples for graph visualization (cf. Section 2.4).

4.1 *MostVis*: Browsing MOST Function Catalogs

In this section, I introduce MostVis[1], a tool for visualizing hierarchical structured specification catalogs of physical in-car networks. Over the last years, these catalogs have become increasingly large and, thus, challenge automotive engineers in terms of browsing, searching and deriving insights. Our[2] tool, MostVis, was initiated by MOST development engineers who work with one specific type of such specification catalogs, called MOST functional catalog (for more background information on the MOST bus system please see Section 3.2). For MostVis' design we carefully adapted hierarchical visualization techniques and combined them in a multiple coordinated view (MCV) approach to satisfy the specific needs of our target group. The basic goal of

---

[1] The name was derived from: MOST functional catalog VISualization.

[2] Portions of this section have been published in [SBH$^+$09]. Thus, any use of "we" in this section refers to Michael Sedlmair, Christian Bernhold, Daniel Herrscher, Sebastian Boring and Andreas Butz.

MostVis was to support engineers in faster access to and exploration of specification data. We evaluated MostVis in a user experiment with 14 domain experts and showed that it was significantly faster than current tools for browsing and searching MOST functional catalogs. Based on these successful results, MostVis was additionally funded, transferred to an external software company and is currently being further developed to closely integrate into BMW's central specification data base tool, BNE.

4.1.1 Problem and Requirements Analysis

After the engineers asked us to help with the problem as briefly stated above, we started our work on MostVis by conducting two explorative user studies: Contextual inquiries [HJ93] with six engineers working in the MOST domain and, second, a focus group with five MOST domain experts (two of them already participated in the contextual inquiries). Both the interviews and focus group lasted an average of one hour per session. The contextual inquiries were used to gather a profound understanding of practices, tasks and problems. We used a focus group approach to collaboratively finalized a set of concrete requirements for MostVis. The results are described as follows:

MostVis' primary target users are MOST development engineers dealing with one specific bus system, MOST, and all ECUs that are connected to this bus system. The major tasks of these engineers are:

- designing and implementing novel MOST systems,
- adjusting existing systems and adapting them to novel developments, and
- specifying these novel systems or system changes in the BNE.

To accomplish these tasks, they more or less frequently use the BNE client to browse and search MOST components, more specifically ECUs connected to the MOST bus system (6–10 in a specific vehicle configuration, up to 20 in the 150-percent specification), functional blocks (FBs < 100 per ECU), functions (functions < 100 per FB) as well as the various parameters of these functions. For browsing without direct BNE access, e. g., at supplier companies, engineers also often use pdf dumps of the data, i. e., the pdf specification documents, for these purposes (see Section 3.3.3, Figure 3.6-a and b show screenshots of the BNE interface and an example page from a MOST specification pdf). However, along with the increasing amount of specification data, frequent activities such as exploring, browsing and overseeing all relevant details have become tedious and laborious with these text-based techniques. This is particularly true for MOST engineers due to the explosion of innovations in vehicle multimedia—i. e., MOST related—systems such as television and internet in the car, rear seat entertainment, surround sound systems or interfaces for various consumer electronics devices. The MOST specification documents and the data stored in the BNE have become unmanageably large. While, for instance, a printed version of the MOST functional catalog from 2001 was comprised of 1500 pages, the equivalent catalog from 2008 we worked with already had 4,000 pages.

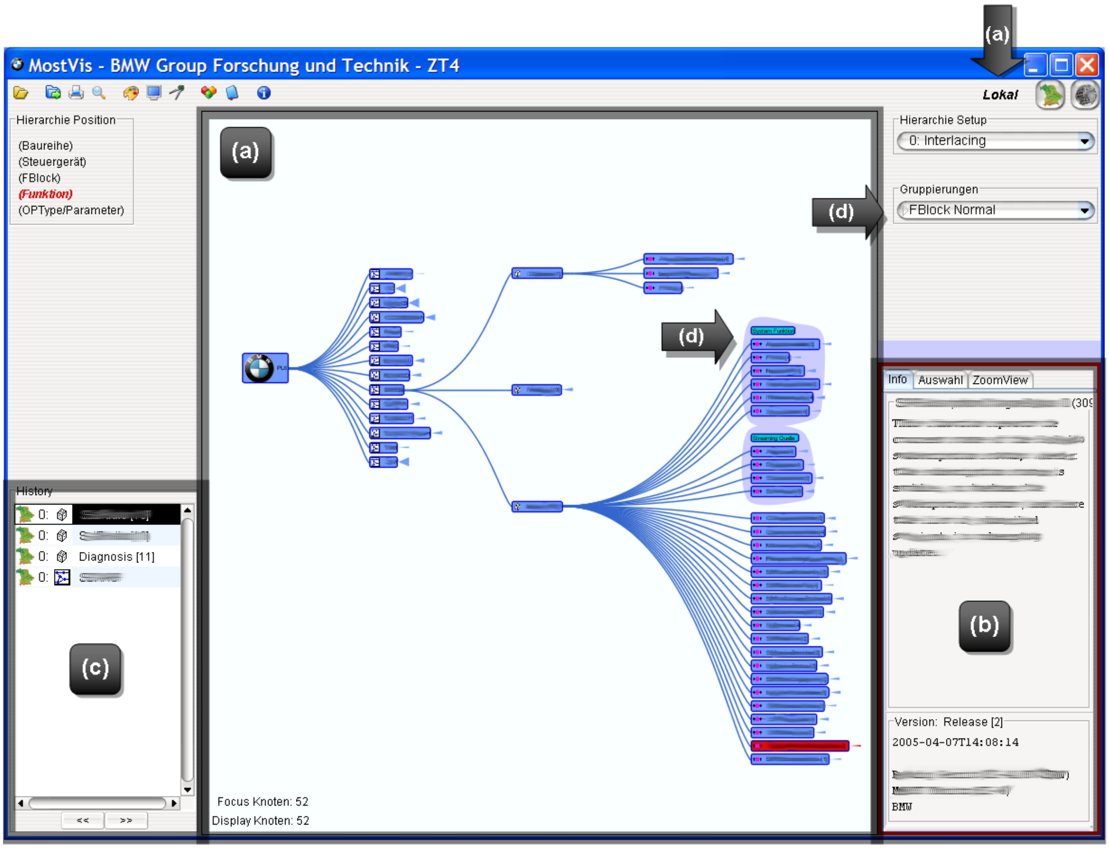
From our contextual studies, we derived a distinction between two sub-groups within our target user -group: (1) MOST-Experts who are engaged in MOST specific tasks more than 50% of their working time and regularly use the BNE (3 interviewees), and (2) MOST-Non-Experts more occasionally working on directly MOST related topics (the other 3 interviewees). Both groups worked with MOST functional catalogs, namely pdf specification documents and the BNE. However, especially MOST-Non-Experts complained about problems with current tools. The textual database front-end overburdens most of these users with its complexity and functional richness. The pdf catalog on the other hand lacks features for interactive browsing, advanced searching and gaining insight into relational or overview aspects. Not surprisingly, due to their long-time experience, MOST-Experts reported not to have major problems in working with the currently available tools. Our interviews, however, revealed that the experts demanded a higher degree of visualization of correlations and also desire supplementary support for an easier and faster, even more lightweight way to browse and search the MOST function catalog. Additionally, MOST-Experts demanded new functions not supported by current tools, such as aggregated search queries, customizable grouping or the possibility to get insight into overview aspects.

We used this input to trigger discussion in our focus group and collaboratively with the engineers finalized the following list of design requirements (DR)—requirements that we derived from our findings—and novel feature requests (FR)—features that were explicitly requested by our end users:

- Representation of all MOST components for browsing and searching purposes (DR)
- Visualization of hierarchical structures between the components, e. g., which ECU holds which FBs (DR)
- Providing all necessary detail for each single component, e. g., the concrete description of a function (DR)
- Increased usability compared to current tools (DR)
- Explicit search for data types, e. g., strings, as this is not possible with current tools (FR)
- Interactive, personal and flexible grouping functionality, also not possible with current tools (FR)

4.1.2 Design

In order to meet the engineers' requirements with an adequate system design we started a user-centered design approach with five MOST engineers who were willing to collaborate on this project. We prepared a variety of paper mockups with several design ideas and variations of



Hierarchie Position
(Baureihe)
(Steuergerät)
(FBlock)
(Funktion)
(OPType/Parameter)

Hierarchie Setup: 0: Interlacing

Gruppierungen: FBlock Normal

Focus Knoten: 52
Display Knoten: 52

Version: Release [2]
2005-04-07T14:08:14
BMW

Figure 4.1: MostVis with **(a)** Local View and its indication, **(b)** Detail View, **(c)** History, **(d)** Grouped elements / Grouping dialog (Some details have been blurred due to IPR restrictions).

them, discussed these with our participants, and iteratively adapted a visualization concept until it fit the engineers' needs. After a final validation of the concept in a focus group with four MOST engineers, we implemented the final concept as a Java prototype using the prefuse framework[3] [HCL05].

MostVis is based on multiple coordinated views (MCV) and visualizes a MOST function catalog of a specified vehicle series. Each function catalog can be selected by the user and then is loaded from a BNE xml export. The xml can contain several different hierarchies which are supported by MostVis and which can be swapped in MostVis during runtime. For a better understanding, however, I will solely stick to the basic MOST hierarchy, car series–ECUs–FBs–functions–parameters, in the reminder of this section.

According to our design considerations during the paper prototyping phase, the main view (cf. Figure 4.1-a and 4.2-a) of MostVis represents the MOST function catalog data by means of a node-link tree visualization and is positioned in the center of the application. Each element of the MOST function catalog is represented as a node, and the edges depict the hierarchical correlations. Besides a textual label, each node incorporates an additional icon representing

---

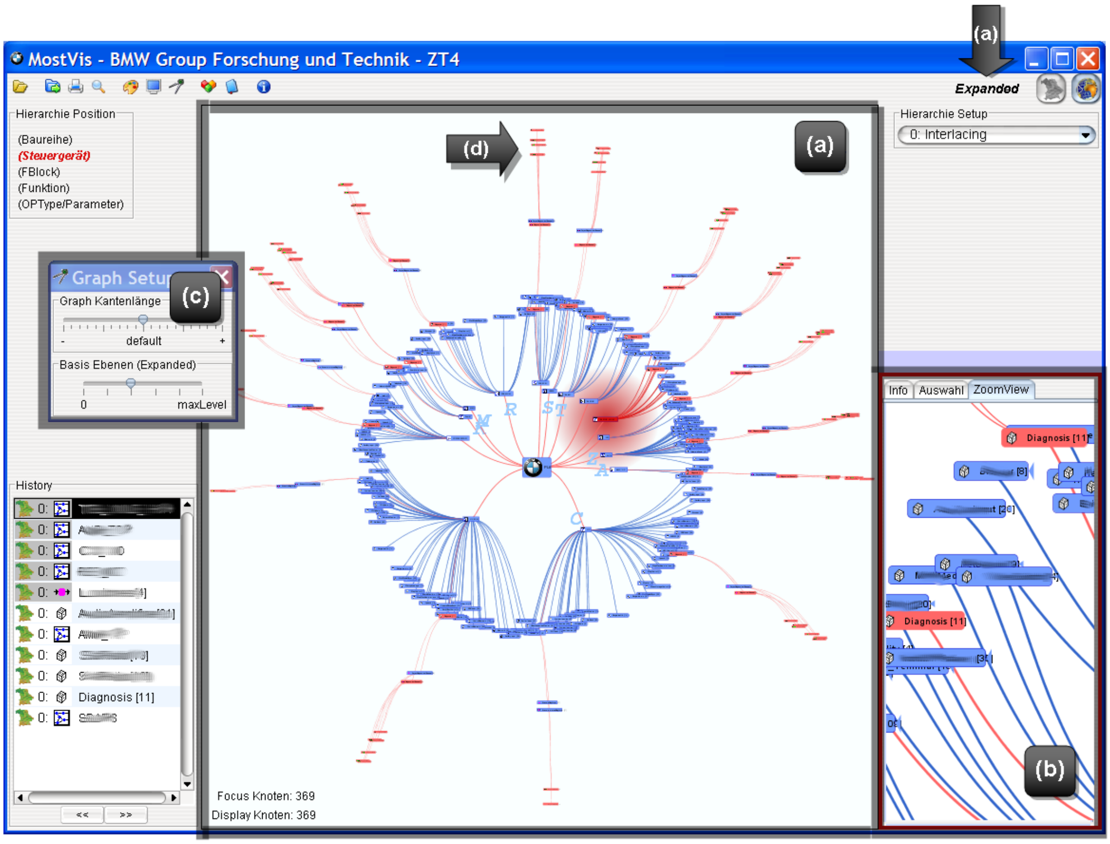
[3] `http://prefuse.org/`

the element's type. These icons are inherited from the BNE interface make use of domain-specific knowledge and to support established recognition patterns (cf. familiarity implication in Section 3.6.2). The general navigation is realized via zooming and panning.

Based on the needs of our target users, the main view provides two different modes: *Local View* and *Expanded View*. The Local View (cf. Figure 4.1-a) is based on a horizontal, left-to-right tree layout. By default, the tool starts with this view showing the root node (car series) and its children (ECUs). Now the user can navigate the tree by selecting and expanding nodes using single-click and double-click actions respectively. A selected node is highlighted and additional detailed information is shown in an extra *Detail View* (cf. Figure 4.1-b). Expanding a node also selects it and, additionally, reveals its children with a short animation. According to the outcomes of our early design studies with MOST engineers, we integrated both, (1) a mechanism which automatically reduces "old" branches when a "new" node is expanded, i. e., all branches are collapsed that do not hold nodes of the path from the selected node to the root node, and also (2) a mechanism for browsing multiple branches manually and in parallel (holding the ctrl-key).

By combining both techniques, our goal was to give users flexibility when exploring the hierarchy without overwhelming them by the entire information or distracting them with unnecessary interactions: The user starts with a rough overview showing only the first level of the hierarchy. Interactively unfolding and folding sub-trees enables step-by-step navigation into regions of interest. Details can be found for each node by looking into the Detail View (cf. Shneiderman's Mantra [Shn96]). Influenced by Plaisant et al.'s SpaceTree project [PGB03], we provide additional angle visualizations to help the user to get a better orientation in the Local View. Each folded (inner) node includes a preview representation of its outgoing edges in the form of a right-side attached triangle. The opening angle of this triangle gives the user a rough estimation of the number of child elements and in doing so also an implicit measurement for importance. If a node is extended the triangle disappears and is replaced by the edges connecting the child nodes.

Switching to the *Expanded View* (cf. Figure 4.2-a) enables the user to gain an overview of the underlying MOST function catalog and to examine elements in a more global context. The Expanded View is based on a hyperbolic 2D tree layout [LR96] which is—due to its radial layout—more adequate for an overview representation of the broad and flat MOST hierarchies [DBETT98]. This view enables a display of the entire catalog. Due to the large amount of data items in the dataset we used (all-in-all approximately 40,000 nodes in the dataset we used), we decided to additionally allow users to filter the information by restricting the displayed layers of the hierarchy (e. g., only root node, ECUs and FBs; cf. Figure 4.2-c). This turned out to be very useful for the engineers because they require overview information mostly for lower layer information, such as showing all ECUs with their associated FBs. The filtered overviews still are very big which complicates the readability of labels. We, thus, integrated a *Zoom View*. The Zoom View is either available via an integrated tab in the lower right corner (cf. Figure 4.2-b) or within an additional extra window. It provides an adaptive magnification of the current mouse pointer's surroundings. This helps the user to identify and select single nodes or links of the graph even at a low level of detail.



MostVis - BMW Group Forschung und Technik - ZT4

Hierarchie Setup: 0: Interlacing

Focus Knoten: 369

Display Knoten: 369

Figure 4.2: MostVis with **(a)** Expanded View and its indication, **(b)** Zoom View, **(c)** Base level (and edge length dialog), currently two levels are completely shown, **(d)** Highlighted search results in red (Some details have been blurred due to IPR restrictions).

Switching between the Local and Expanded View is challenging for the user because the different tree layout requires context switching. However, the different layouts are reasonable due to their different intentions. To counterbalance potential loss of context, we integrated a mapping functionality that retains selected elements (and also search queries, see below) during transitions between the Local and Expanded Views (and also between different hierarchies). Highlighted elements in one view will also be highlighted in another view.

In addition, to the basic visualization we integrated a set of useful functionalities which we derived from our pre-design studies and from the meetings with engineers during our participatory design process. Probably the most important feature is search. A *Search Dialog* (cf. Figure 4.3-a) allows for conducting different kinds of search queries. So, the user, for instance, can either search the entire catalog or constrain the query to subtrees, or alternatively chose a specific data type to search for. After conducting a search, the results are presented and highlighted in several views. First, an additional *List View* (cf. Figure 4.3-b) pops up and textually lists all found elements together with their path within the hierarchy. Second, the results are highlighted in the tree visualization by coloring the found nodes and the paths leading to them, and by automatically expanding the branches to hidden search results. While in the Expanded View all hidden search

Suche

Beginn der Suche:
● Wurzel ○ akt. Knoten

● Label diagnos*
○ Datentyp BitField

(a)

Suchergebnis für: diagnos* (118 Einträge)

Such Knoten	(0) Pfad	(1) -->	(2) -->	(3) -->
Diagnosis [11]	PL6	[redacted]		
Stream DiagnosticsData	PL6	[redacted]	Diagnosis [11]	AsyncDiagno
Number DiagnosticsSA	PL6	[redacted]	Diagnosis [11]	AsyncDiagno
Number DiagnosticsTA	PL6	[redacted]	Diagnosis [11]	AsyncDiagno
DiagnosticsChannel [4]	PL6	[redacted]	Diagnosis [11]	
Stream DiagnosticsData	PL6	[redacted]	Diagnosis [11]	DiagnosticsC
Number DiagnosticsSA	PL6	[redacted]	Diagnosis [11]	DiagnosticsC
Number DiagnosticsTA	PL6	[redacted]	Diagnosis [11]	DiagnosticsC
Diagnosis [11]	PL6	[redacted]		

(b)

Figure 4.3: Additional search views in MostVis: **(a)** Search Dialog; and **(b)** List View of search results (Some details have been blurred due to IPR restrictions).

results are expanded immediately (cf. Figure 4.2-d), in the Local View, due to clarity reasons, only the first search result is automatically expanded. Subsequent results can be expanded either by selecting them in the List View, or by directly following marked edges and manually unfolding branches.

Another feature which was requested in the pre-design studies is the grouping functionality. Therefore, we implemented a feature, that allows the user to cluster child elements according to a previously specified xml grouping pattern (cf. Figure 4.1-d). These grouping patterns can either automatically be generated during database exports or manually using meta-knowledge. Stored as xml files, groupings can be imported into MostVis, a direct generation of grouping files within MostVis, however, was not supported at that time. As stated during the pre-design studies, the grouping feature was highly demanded by the MOST-Experts. It addresses one currently unsolved problem for long, ungrouped lists of elements. Current tools simply list all elements in alphabetical order which hinders the user in quickly finding, re-finding and correlating necessary information. On the other hand, there already exists rich knowledge on how to semantically cluster FBs, functions, etc. Such semantic groups, can be, for instance, management functions that belong to each FB, digital audio sources or sinks, or a variety of application-specific functions. At present, this information, however, is only in the heads of engineers and cannot be specified within the BNE. Making the knowledge explicit can improve orientation and navigation but also support active use of this information for design decisions.

We also integrated a one-dimensional, browser-like history. Every time an expansion occurs, a history log is created and is added to the history view in the lower left corner (cf. Figure 4.1-c). The user can then either directly jump to a desired point or can navigate step-by-step back and forth through the history. Each history log holds a “shortcut visualization” of the corresponding application’s state. A conducted search query in the log is represented via a grayish rectangle, particular icons show whether the Local or the Expanded View was opened and a label and icon of the selected element are provided. This helps the user to get a better orientation within the history.

According to the technical requirements, features for exporting and printing were added to MostVis. The export was demanded by the engineers to support reintegration of information into other tools or to get fast access to representation material. Therefore, we support csv exports (e.g., for search results) and image exports of each tree view. Especially for the latter feature, it is helpful that the layouts are customizable. Edge length, colors, etc. can be freely changed by the user.

4.1.3 Evaluation

We conducted an expert user study with our target group engineers to evaluate MostVis in terms of performance, understandability as well as subjective preference. In this subsection, I first describe the study design and the users' tasks. Then, I show both qualitative and quantitative results of our study followed by a discussion.

Study Design and Tasks We conducted the study with 14 participants (aged 20 to 39 years, all male) that are all automotive engineers with varying levels of expert knowledge regarding MOST (average of 2.5 years, eight MOST-Non-Experts and six MOST-Non-Experts). All studies were conducted at the desks of our participants. The study was composed of five parts:

1. A pre-questionnaire to gather information about the subject's current working practices and level of MOST expertise;
2. An introduction of MostVis and its features;
3. A practice phase where the users accomplished nine example tasks with MostVis in order to verify that they entirely understood both the application and the interaction techniques;
4. The main part of the user study where we measured task completion time of our participants with MostVis and with their favored current tool both running on their one machine
5. A post-questionnaire with qualitative questions regarding the acceptance and likability of MostVis.

The tasks we used in the main part of the study (4) were derived from our contextual inquiries and were designed to reflect every-day work practices of MOST engineers (cf. Section 4.1.1) dealing with exploration of and browsing/searching in MOST function catalogs. Additionally, all tasks were pre-evaluated in a pilot study with a MOST-Expert as well as a MOST-Non-Expert. We classified the tasks in three categories, (a) *lightweight tasks* (searching and browsing elements/information), (b) *advanced tasks* (aggregated search queries; in general, tasks which were estimated by our lead users to be important but rather hardly realizable with current tools), and (c) *novel tasks* (grouping, overview; tasks which were estimated by our lead users not to be realizable with current tools) . Table 4.1 shows a list of all tasks we tested in our study.

Nr.	Task	Classification
1.	Find all functions of the FB *FB-a*!	lightweight
2.	Sort the following three ECUs by means of their complexity (equal to number of FBs): *ECU-b, ECU-c, ECU-d*!	lightweight
3.	Name 3 functions of the FB *FB-e* that contain *booleans*!	advanced
4.	Show the description of the *FB-f* that is located in the *ECU-g FB-h*!	lightweight
5.	How many *enums* are located in the *ECU-i*?	advanced
6.	Return to task 4!	lightweight
7.	Name 3 functions from the FB *FB-j* which are assigned to its *sinks* and 3 functions which are *sources*!	advanced
8.	What is the element *Element-k* and where can you find it?	lightweight
9.	Is there an available function in the *FB-l* that gives you information about the currently selected radio station?	lightweight
10.	Give an overview over all elements contained in *ECU-m* (FBs, functions, OpTypes, parameters)!	novel
11.	Name 3 ECUs that are member of the group *Group-n* as well as 3 of the *Group-o*	novel

Table 4.1: Tasks (paraphrased from German), gray marked rows show “novel” tasks which were excluded from the statistical analysis.

The participants had to pass through these eleven tasks twice, on the one hand with MostVis and on the other hand with their preferred current MOST tool(s) (pdf and/or the BNE interface, specified in the pre-questionnaire: nine participants used the BNE only, five used both BNE and pdf). While the order of the tasks was constant, the order of the tools was alternated between participants in order to minimize fatigue effects. With both tools, we measured the task completion time for each task, beginning after the participants were instructed with their task (i.e., after reading the description and starting to interact with the tool) until they completed it (e.g., pointing out the correct element in a browsing task). If a task lasted longer than three minutes, participants proceeded to the next task (due to time restrictions). Furthermore, if a user had the impression, that the task was not feasible (especially for advanced tasks with traditional tools), he could also proceed to the next task. However, to avoid distorting the results these options were allowed only with current MOST tools and therefore could not benefit MostVis for purpose of this study.

We used a repeated measures within subject factorial design with the independent variables *Task* and *Tool* (*preferred MOST tool* vs. *MostVis*) and measured the time for each *Task* with both *Tools*. In addition, we took the participants' *MOST-Expertise* (*MOST-Non-Experts* and *MOST-Experts*) into account derived from the amount of time per day our participants worked with MOST functional catalogs (derived from the pre-questionnaire). The total time of the study for each participant varied between one and 1.5 hours.

Despite measuring the task completion time, we also recorded spontaneous feedback during the session by taking notes. In order not to distort time measures, however, we advised our participants not to give any feedback during task completion.

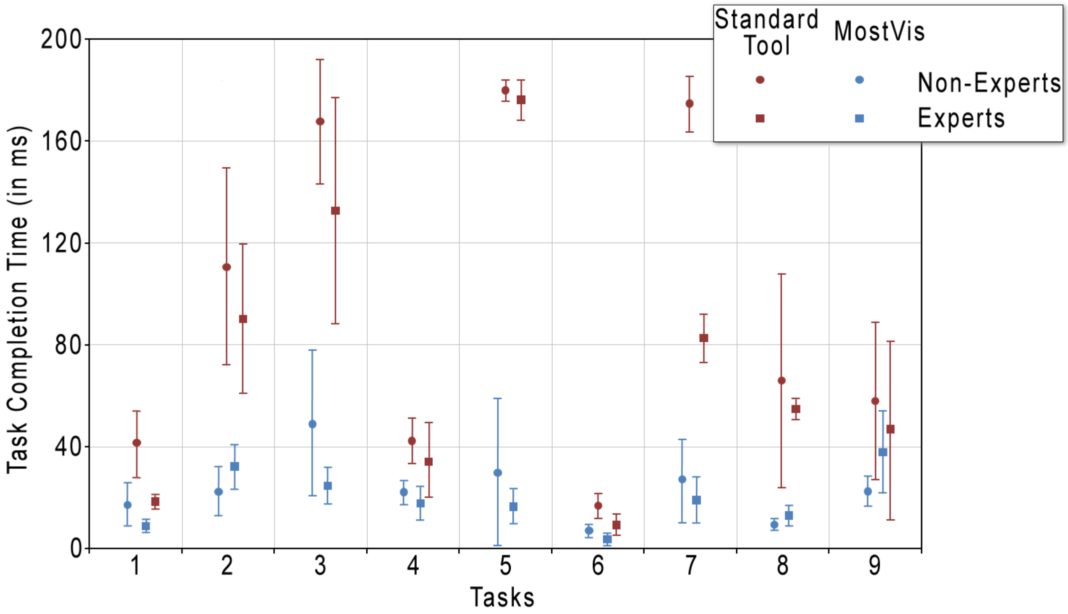


Figure 4.4: Average selection times for MOST-Expert users versus MOST-Non-Experts depending on the used tool.

Hypotheses Based on our knowledge from the pre-design studies, we had four hypotheses:

1. MostVis will outperform current tools with MOST-Non-Experts independent of the task (H1).
2. For MOST-Experts MostVis will be better for advanced tasks and novel tasks (H2), because they initially were stated by the engineers to be hardly or even not realizable with current tools. MostVis, on the other hand, was designed to support them.
3. For the same reason, MostVis will have a similar performance for all of our tasks while current tools will strongly depend on the task, lightweight or advanced (H3).
4. The ratings of MOST-Non-Experts will be considerably better for MostVis compared to current tools (H4).

Results Repeated measures analysis of variance showed no significant effect on task completion time when presenting the two *Tools* in different order, indicating the adequacy of a within-subject design. Subsequent analysis revealed that two *Tasks* (no. 10 and 11, cf. tab. 4.1) were not realizable by a large amount of participants using their preferred MOST tool(s) (42.9% of MOST-Experts and 57.1% of MOST-Non-Experts respectively for task no. 10, and no one was able to realize task no. 11). Looking at Table 4.1 shows that these tasks are the novel tasks which were mentioned by engineers to be not realizable with current tools and therefore this result was predictable. Interestingly, however, some of our participants managed to execute the overview task no. 10. Though, in these cases they had high task completion times (averages of 161.4 seconds for MOST-Experts and 134.3 seconds for MOST-Non-Experts respectively). Using MostVis, on the other hand, the average task completion times were much lower (14.6 seconds for MOST-Experts and 27.4 seconds for MOST-Non-Experts respectively). In order not to benefit MostVis with its novel features that are not provided or only very hardly realizable with current tools, we excluded them from our statistical analysis but included them them again in the discussion section.

Selection Time: For the remaining 9 *Tasks*, we analyzed the measured task completion times. We performed a 9 × 2 × 2 (*Task* × *Tool* × *Expertise*) within subjects analysis of variance and found significant main effects for all independent variables: *Task* ($F_{8,48} = 33.533, p \ll .001$), *Tool* ($F_{1,6} = 191.719, p \ll .001$) and *Expertise* ($F_{1,6} = 15.319, p = .008$). However, most interesting is the significant interaction of *Task* × *Tool* ($F_{8,48} = 37.336, p \ll .001$). Post-hoc multiple means comparison tests showed that especially the identified advanced *Tasks* (i.e., tasks 3, 5 and 7) had significantly better task completion times using MostVis regardless of the participants' expertise (all $p \ll .001$). Additionally, we found better performance for lightweight tasks for both groups (only *Task* 9 was not significant with $p = .097$, all other $p < .01$). This indicates that MostVis helps MOST-Experts as well as MOST-Non-Experts to accomplish both, advanced and lightweight tasks, with nearly the same performance (cf. fig. 4.4). Therefore, our hypotheses H1, H2 and H3 are supported.

Besides achieving significantly better results with MostVis, the saved time especially for advanced tasks is of great importance. As shown in Table 4.2 the average saved time of completing advanced tasks was 138.3 seconds for MOST-Non-Experts and 110.3 seconds for MOST-Experts. Hence, MostVis' performance increase is even stronger for MOST-Non-Experts. Generally, while the average task time was 83.3 seconds using the standard tools, MostVis allowed users to perform the same tasks in an average time of 21.1 seconds resulting in saved time of 62.2 seconds.

Subjective Ratings: In post-study questionnaires, participants were asked several questions regarding visualization and interaction using MostVis. They had to rank these aspects on a Likert-Scale from 1 to 5 where 1 equals best and 5 equals worst. When asked about the visualization techniques generally used in MostVis, participants ranked them 1.33 on the mentioned scale. This supports our hypothesis H4. For the different views, the Local View got slightly better ratings than the Expanded View (1.27 compared to 2.0). This could be explained by the large dataset that partly made the Expanded View confusing for the users. These findings, however, also indicate room for slight improvements with the Expanded Views. Further ratings for built-in mapping, history and group functionality were consistently good with 1.4, 1.8 and 1.47 respectively.

We also verbally asked the subjects to give general feedback and criticism regarding the MostVis prototype. We received very encouraging feedback such as: (a) "*When can we use MostVis*" (or similar by 4 subjects), (b) "*when working with the database interface I am not as productive as with MostVis*" (or similar by 2 subjects), (c) "*I was not able to handle the current database interface after one week as good as MostVis after 15 minutes*" (1 subject) or (d) "*such a nice*

	Std. Tool	MostVis	Diff.
Lightweight Tasks/MOST-Non-Experts	55.4s	16.6s	38.8s
Advanced Tasks/MOST-Non-Experts	174.0s	35.7s	138.3s
Lightweight Tasks/MOST-Experts	42.2s	18.7s	23.5s
Advanced Tasks/MOST-Experts	130.4s	20.1s	110.3s

Table 4.2: Improvement of selection times using MostVis (without novel tasks).

overview is not available in current tools" (1 subject). In addition to the general positive feedback and indication of acceptance (see a), the subjective ratings indicted improved productivity (see b), that the interface could be quickly learned (see c) and that overviews were well supported (see d). Additionally, we got several inspirations for future improvements and extensions for MostVis, most importantly the direct integration of the grouping functionality, the usage of *"google hints"*, i. e., an autocompletion feature for additionally supporting and speeding up MostVis' search, and the possibility to alternatively sort children based on revision numbers instead of alphabetically.

Discussion of our Results As already shown in the results, MostVis performed better than currently used tools. This can be understood by providing better browsing and searching functionality needed in large datasets which is insufficiently supported by current tools. While the results with current tools significantly depend on the *Tasks*, MostVis additionally allows similar performance for various tasks (H3). This can be explained with MostVis' direct support of the novel, demanded features, namely, aggregated search queries, grouping and overview leading to much better completion times with advanced tasks (cf. Figure 4.4) and with novel tasks which were not realizable by most of our participants using current tools (see above). However, it is important to note that also for lightweight tasks—which are very frequently performed by the engineers—MostVis performed better, for both MOST-Non-Experts and MOST-Experts. The little influence of the participants' *Expertise* was surprising. We hypothesized that MOST-Non-Experts will better perform with MostVis regardless of the task (H1), MOST-Experts, however, just for advanced and novel tasks (H2). Indeed, MOST-Experts unexpectedly also were faster with MostVis performing lightweight tasks. According to these results, a tool such as a MostVis could be applied to all levels of expertise without forfeiting performance in subgroups (which is especially important for an industrial company).

In addition, the subjective ratings for MostVis were consistently good. While unsurprisingly the MOST-Non-Experts responded very positively to the tool(H4), we did not expect these results were from MOST-Experts as we assumed that they are highly familiar with current tools and would prefer them. From verbal feedback, we learned that especially providing the novel features (grouping and overview) and the better support of advanced tasks (aggregated search queries) were very popular with MOST-Experts and that these aspects played an important role the good subjective ratings.

The fact that similar tasks are frequently performed by engineers day-to-day, demonstrates that concepts of MostVis have the potential to provide a benefit in work performance, cost reductions and additionally in better understanding and communication. It is obvious that due to richness of functionality (e. g., editing functions in the database application) and to the intention of documentations (e. g., pdf as official contract with supplier companies), currently used tools will persist. However, MostVis seems to be a well-performing alternative for frequent searching and browsing tasks and is worth using as an extension of the current tool chain.

Impact of our Results: Integration with BNE 2.0 After the user study, we presented MostVis to a larger group of decision makers at BMW. Based on our successful evaluation results, this committee decided to additionally fund MostVis in order to closely integrate it with the upcoming version of the BNE, BNE 2.0. In doing so, they wanted to make MostVis available for a wider range of employees at BMW and to validate the benefits of MostVis under real conditions. MostVis was transferred to an external software engineering company. This company (a) ensured software quality for company-wide smooth deployment, (b) closely integrated MostVis with the BNE 2.0, and (c) implemented additional features based on the formative feedback from our user studies, each of these tasks in close cooperation with us. Most important changes, are the adaption of MostVis from a stand-alone application to an eclipse plugin [Ecl] in order to meet BNE 2.0's requirements, the close integration of group editing with MostVis' interface and the direct connection to the BNE 2.0 backbone database. At the moment of writing this thesis, MostVis has not been rolled out in the company yet due to the fact that the larger project's deployment, BNE 2.0, was postponed to beginning of 2011 and with it also the deployment of MostVis.

4.1.4 Discussion

In this section, I introduced MostVis a tool for visualizing MOST functional catalogs supporting development engineers in browsing, searching and exploring these catalogs. In a quantitative user study we showed that MostVis is significantly faster than current techniques for a set of typical tasks derived from engineers working practices. These results allowed us to deploy MostVis with current software environment and processes (BNE 2.0).

As outlined in Section 1.2, one of the main objectives of this thesis is to better understand how we can apply, deploy and evaluate InfoVis/VA tools in large company environments. MostVis provides a best practice example showing how a research project successfully ended up in a system closely integrated in current working practices. We see the two main reasons for this in: (a) The iteratively refined design resulting from close cooperation between interested end users and InfoVis designers; and (b) validating the tools benefits with a clear and generalizable measure of improved efficiency—time—and using these results to convince decision makers. From an InfoVis/VA point of view, especially the second aspect, the evaluation, was interesting. From literature (e. g., [IZCC08, Pla04]) it is known that quantitative studies often fail to provide a proper understanding of how InfoVis tools are used and what the actual reasons for success or failure are. To better understand the long term nature of data exploration tasks and the process of getting insight into data, qualitative methods often turned out to be more adequate as they help to better understand the users' thoughts (e. g., [IZCC08, SND05]). This holds also true for our quantitative evaluation of MostVis. While we showed that MostVis was significantly faster for tasks we derived from current practices, we could provide only little insights about the whys: Was MostVis faster because of the hierarchical visualization? Was it a specific combination of views? Would we get the same results if we change the layout of the visualization? Did we just have a better system performance? Comparing two entirely different tools as we did, imposes too many uncontrolled variables in order to derive clear conclusions for such questions or implications for general design guidelines.

On the other hand, the results are clearly understandable, and easy to communicate. Time improvements shown through a statistical analysis, helped us to convince decision makers at the company and finally will lead to a close integration of our tool into end users daily working practices. Having such a closely integrated InfoVis/VA tool then, however, provides an excellent platform for conducting long-term studies in real environments, with real tasks under real circumstances (e. g., MILCs [SP06]). In our setting, usually doing studies with prototypes not integrated with current processes (e. g., no direct data connection, no coexistence with other tools, etc.) leads to artificial study conditions and distorts findings in terms of reality.

During this process we, however, also had to learn that deploying a tool in a large company setting can be a long-time undertaking. At the point of writing this thesis, the start of the MostVis project dated back two years but the integration process had not yet been finished. Integrating a tool into company's processes often hinges on factors which are hard to influence by researchers. For us, the deployment of the platform we integrated MostVis into (BNE 2.0) was postponed by one year so that long-term user studies could not been conducted within the scope of this thesis. Key future work for MostVis would therefore be to conduct such long-term user studies in order to derive a better understanding of how it is actually used under real circumstances and how it provides novel insights into the data (this is already scheduled by BMW). Furthermore, we implemented MostVis specifically for MOST functional bus systems. However, other specification documents are often organized similarly and usually are even smaller in the amount of information they store. We therefore believe that a fine-tuned version of MostVis will likely work for other specification documents as well. Further research in this direction is necessary.

4.2 *WiKeVis*: Visualizing Functional Networks

In this section, I introduce WiKeVis a design study for the visualization of functional in-car networks (cf. Section 3.2 for a closer description of functional in-car networks). While MostVis was designed for supporting engineers in browsing specification catalogs, WiKeVis is closely related to the current lack of understanding comprehensive aspects in functional networks I identified and outlined in Section 3.5.3.

WiKeVis was a engineer-initiated solution triggered (and also organizationally led) by an engineer working on novel development processes for automotive electronics (WiKeVis' stakeholder). Based on his contacts within the company, we[4] set up a cooperation with a specific target group of network engineers, namely engineers who are responsible to specify and maintain dependencies during functional network design (cf. V-Model top left, cf. Section 3.3.1). For this purpose, they use "dependency chains", local specifications of communication paths through the functional network. After specifying these functional dependencies, they are used to design and optimize the functional network architecture and influence decision making during partitioning of the functional network to the physical network. We found that current forms of representing these dependencies are restricted to local correlations of 5–50 functional blocks and strongly lack in providing insights into comprehensive dependencies and to correlations within the entire functional network (up to 2500 FBs).

WiKeVis provides a more global visualization of the functional dependencies in the network based on a hierarchically clustered, multi-focus node-link representation and allows the user to interactively explore various dependencies between elements of the entire functional network. Our approach was strongly influenced by Schaffer et al.'s method for navigating hierarchically clustered networks [SZG+96]. Section 2.4 gives a general introduction to graph drawing which is useful to better understand the design of WiKeVis.

WiKeVis was evaluated in a qualitative user study, focusing on usability and on potential domain value of such an approach. While the approach to more comprehensively represent the functional network was found to be beneficial, we learned that especially combining our approach with editing techniques to edit dependency chains and an adaption of our graph layout would be necessary for our target users.

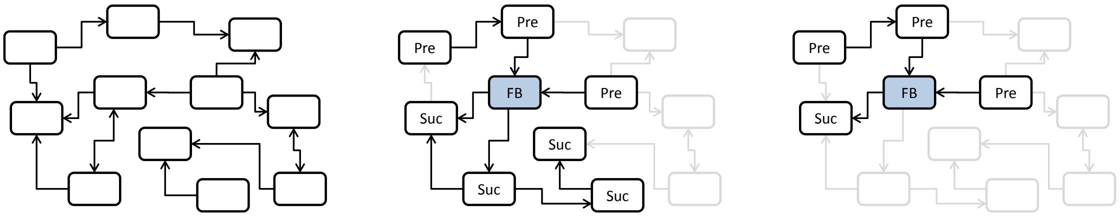
4.2.1 Dependency Chains

Over the last years, the usage of *dependency chains*[5] has become popular in the automotive electronics domain. Loosely spoken, the term is used to describe dependencies of a specific vehicle's functionality within the larger network, meaning all FBs influencing this functionality and FBs influenced by this functionality. Such dependency chains are used to specify the functional network (this section), and recently also for network diagnosis (cf. [BSN07] or our approach introduced in Section 5.2). To the best of our knowledge, currently there exists no specific definition and the term is used ambiguously and rather loosely by engineers of different departments. For the purpose of this thesis, we distinguish between two frequently referred descriptions of "dependency chains" and provide a clear definition for both of them. Our definitions are based on describing the functional network as a directed, clustered graph with FBs as vertices and directed communication exchange between FBs as edges, each FB therefore has dedicated ins (direct predecessors) and outs (direct successors). The graph is clustered either by domains, systems and subsystems (functional network, in this section) or ECUs (physical network) and does not necessarily refer to a completely specified network but also can describe a work-in-progress version. We refer

---

[4] Any use of "we" in this section refers to Michael Sedlmair and Korbinian Zollner.

[5] Translation from the German domain term "Wirkkette", therefore the name of our tool: WiKeVis.



(a) (b) (c)

Figure 4.5: Functional graphs: **(a)** Schematic representation of the entire functional dependency graph; **(b)** a transitive chain holding all reachable elements around an selected FB; and **(c)** a dependency chain showing the interaction of an FB how it is manually specified by an engineer (The dependency chain shown in this example is chosen arbitrary for the purpose of clarifying the concept).

to such graphs as *functional dependency graphs* (cf. Figure 4.5-a). Based on such a graph we define:

1. *Transitive chain (Mathematically derived):* We define a transitive chain around a specific FB as the set of all directly and indirectly reachable successors and all directly and indirectly reachable predecessors in a functional dependency graph (similar to the mathematical concept of transitive closure). Transitive chains are directed sub-graphs of the functional dependency graph and exhibit all possible relations of a FB in question. They are usually used in later development phases where the network is mostly or fully specified (cf. Figure 4.5-b and Section 5.2.1 where used them for re-computing dependencies).

2. *Dependency chain (Manually specified):* Dependency chains are similar to transitive chains, i. e., directed sub-graphs of the functional dependency graph, however, with the basic difference that they are manually specified by engineers and not mathematically derived from a larger graph. A dependency chain therefore does not necessarily include all possibly reachable FBs, such as transitive chains, but only the ones that are specified by the responsible development engineer (cf. Figure 4.5-c, used in this section).

In this thesis, we use the terms as defined above to explicitly refer to one of the concepts described. To generally refer to all of them (in the loose way as it is often done in the field), we use the term functional dependencies. This section deals with dependency chains, manually specified by engineers, while Section 5.2 deals with transitive chains, derived from a fully specified system. Figure 4.5 illustrates the two definitions in a graphical way and helps to distinguish between the ambiguous meanings of the term “dependency chain”. Set-theoretic the relationship between the three concepts presented can be described as:

$$Functional\ Dependency\ Graph \subseteq Transitive\ chain_{fb} \subseteq Dependency\ chain_{fb}$$

4.2.2 Problem and Requirements Analysis

For problem and requirements analysis purposes, we collaborated with three domain experts: An engineer working on improvement of development processes (the stakeholder of this project), a group leader of nine engineers working with dependency specifications, and a system developer (the latter two based on the contacts from the stakeholder). We had three pre-design meetings (focus groups) with these engineers with the goals (a) to better understand domain practices, problems and challenges, (b) to collaboratively derive design requirements, and (c) to discuss and evaluate early design sketches of WiKeVis. The results of our studies can be summarized as follows:

Current Practices and Problems At the time of our pre-design studies on WiKeViz, the specification of functional dependencies was done via Excel sheets. Each row in such an Excel sheet represented a dependency chain represented by a name, ID and a set of functional blocks respectively with all direct ins and outs to describe the correlations. Each sheet was concurrently edited by several (usually five to six engineers). The Excel sheet we got for using with WiKeVis contained 412 dependency chains (rows) with a maximum of 50 FBs. An average dependency chain, however, contained between 5 and 20 functional blocks.

In addition to these Excel sheets, engineers prepared graphical representations for chosen dependency chains. For this purpose, they often used a tool called PREEvision [Aqu10] that allowed them to import the raw data from Excel and to manually design an orthogonal node-link network representation of a particular dependency chain. For this purpose, PREEvision provides several (automotive electronic specific and basically orthogonal) diagram templates that can be chosen, edited and arranged in a very similar way as it is, for instance, possible with Microsoft Visio (i. e., manually dragging nodes and connection lines with connectors that stay connected). Figure 4.6 shows an example of such a graphical dependency chain representation in PREEvision. According to the description of our focus group participants, Excel sheets and PREEvision drawings are used in a next step to design, reorganize and optimize specification documents of the functional network. During our focus groups, we also learned about several drawbacks with current practices. First, concurrent editing in Excel sheets often results in inconsistencies and redundancies in the data. Second, the manual creation of graphical representations of dependency chains is time consuming and error prone. However, most importantly current working practices lack in providing sufficient insight into functional dependencies beyond single specified chains. Our collaborators' stated that a thoughtful graphical representation of comprehensive network dependencies, would be highly instrumental in detecting mistakes and suboptimal functional structures earlier on.

Design Requirements and Challenges Based on these findings, we collaboratively derived three basic design requirements for WiKeVis in the last workshop:

Figure 4.6: Manually generated graphical representation of a dependency chain in PREEvision.

1. *Support automatic graphical representation:* Currently, manual preparation of graphical dependency chain representations is tedious and time-consuming. To save valuable time of domain experts, novel visualization tools therefore should provide as much automation as possible[6]

2. *Showing the entire "wallpaper":* The metaphor was introduced by the group leader and adopted over the three meetings by the other two participants. According to the group leader *"showing the entire functional network would be really beneficial [...] however, showing it with current techniques would end up with a 100 square meter 'wallpaper' [...]"*. Currently there is no information available about the entire structure of the functional network, however, according to our participants *"this could be very helpful for redesigning systems or for optimizing communication paths"*. The most promising approach was seen in *"making the 'wallpaper' interactive to master its complexity."*

3. *Showing traditional dependency chains:* Though the representation of a comprehensive functional network was estimated to be beneficial, on the other hand each tool for dependency chain engineers should still provide a traditional representation of single, manually specified dependency chains. This is of high importance for quickly referring back to see how one or somebody else specified a specific chain.

Beside this basic requirements, we collaboratively decided not to address the aspect of editor functionality in the work on WiKeVis, as this would bring along a set of other problems which should be addressed in follow up projects. We also excluded another challenge, namely overcoming redundancies and inconsistencies in data management as out of scope for a visualization research project.

---

[6] Comment: A tool focusing on this aspect should also take into account the lessons learned from BNVis (cf. Section 3.4.3) which we discovered during our pre-design research on RelEx (cf. Section 4.3), i. e., after the WiKeVis project.

4.2.3 Design

After the pre-design analysis, we triggered a participatory design process in close cooperation with two of the three engineers already participating in our focus groups (stakeholder and group leader) and iteratively implemented an Adobe Flex[7] software prototype.

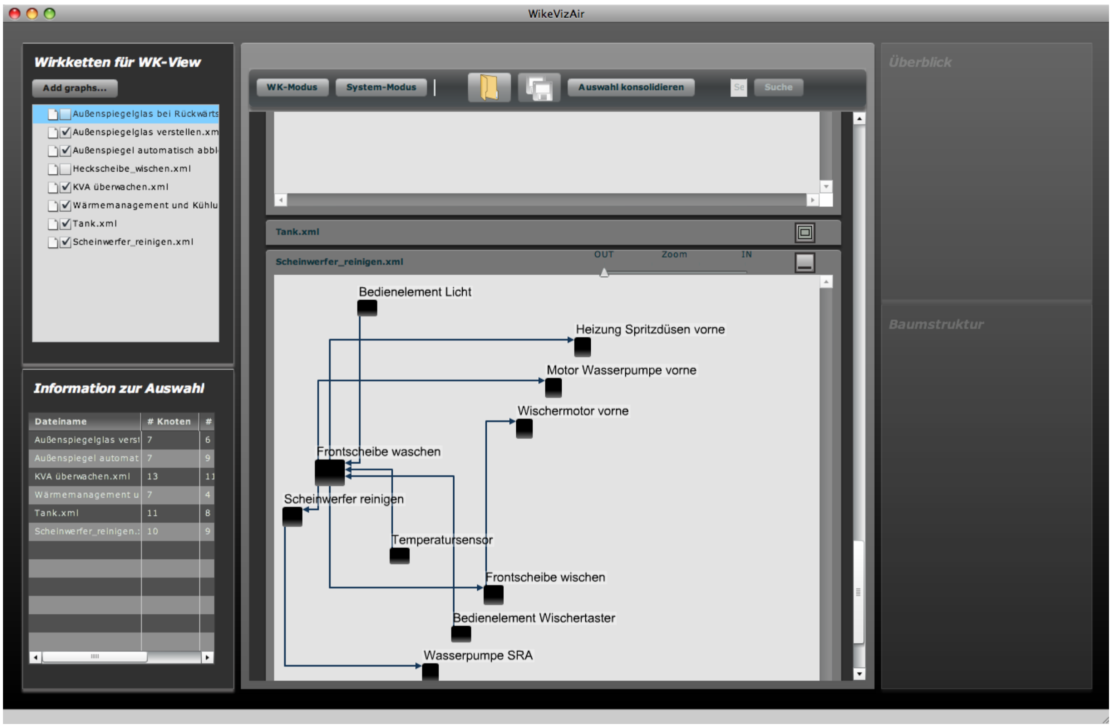
The resulting application WiKeVis is based on multiple coordinated views similar to MostVis (see Section 4.1), with a central view for representing the data for exploration and several secondary views supporting overview, additional perspectives, search and navigation. We used GraphML [BEH+01] for data storage and provided a parser to easily transfer data from Excel exports to GraphML. WiKeVis uses a strict distinction between two modes: dependency chain mode (d-mode) and consolidated mode (c-mode). In line with our design requirements, we wanted to support (and slightly improve) on the one hand traditional working practices with single dependency chains similar to PREEVision (d-mode). On the other hand, the c-mode should provide engineers with a novel perspective on global structures of the functional dependencies. For this purpose, we decided together with our collaborators to cluster the functional network in systems and subsystems (domains were estimated as being too rough) and to use this information to scale the entire graph to a readable form. Both modes are based an orthogonal, grid-based graph layout algorithm derived form Biedl et al.'s three-phase method [BMT97]. According to Biedl's method, nodes are represented via small rectangles ordered onto a grid, and arrows orthogonally connect the nodes with at most one bend per edge. For us, we defined functional blocks to be black rectangles, added a label to them and decided to add ins and outs to all four edges of a node as it is, for instance, done in PREEvision. Furthermore, we—collaboratively with our main stakeholder—decided to position all nodes' ins at the top/bottom and all outs on the sides as it seemed us a good possibility for a clear arrangement (what turned out not to be true, see Section 4.2.4). Figure 4.7 shows an example how the final graph layout looked like.

In the following, I describe the tool by explaining the two modes as well as the process of consolidating chains to a comprehensive functional graph in more detail.

d-mode According to our requirements, the d-mode provides the user with a traditional representation of single dependency chains, with the difference that nodes and edges are automatically layed out. To show the graphical representation of a specific dependency chain, the user can select them from a *List View* containing all dependency chains imported. After selecting a chain, the user initiates to show its graphical node-link representation in the *Main View*. Selecting additional dependency chains leads to a vertical, scrollable set of panels in the main view, each of them showing the graph representation of a selected dependency chain. To support usable navigation and to allow side-by-side comparison each dependency chain panel can be closed, moved (drag'n'drop) or minimized/maximized. For quick reference purpose, an additional *Information View* shows all currently selected dependency chains and provides statistical information such as number of nodes or edges. Figure 4.7 shows a screenshot of the d-mode.

---

[7] `http://www.adobe.com/`



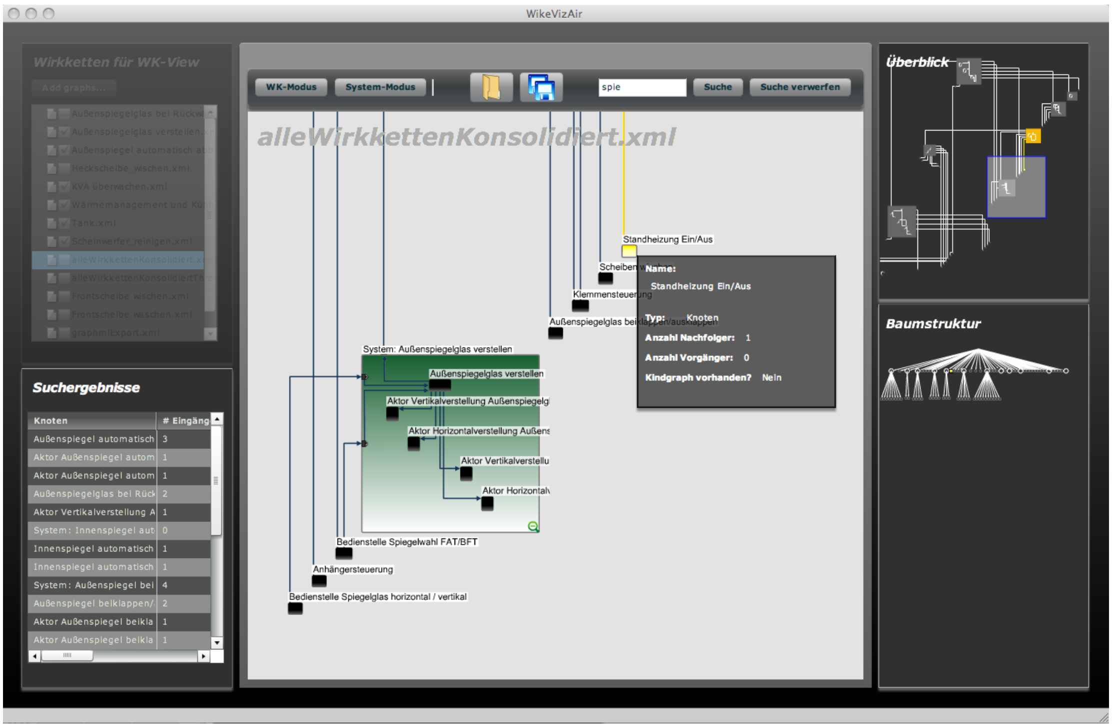
☐ Außenspiegelglas bei Rückwärts
☑ Außenspiegelglas verstellen.xm
☑ Außenspiegel automatisch abbl
☐ Heckscheibe_wischen.xml
☑ KVA überwachen.xml
☑ Wärmemanagement und Kühlu
☑ Tank.xml
☑ Scheinwerfer_reinigen.xml

Dateiname	# Knoten	#
Außenspiegelglas verst	7	6
Außenspiegel automat	7	9
KVA überwachen.xml	13	11
Wärmemanagement u	7	4
Tank.xml	11	8
Scheinwerfer_reinigen.:	10	9

Figure 4.7: WiKeVis in d-mode: Top left shows the List View of currently loaded dependency chains. With the ‘add graphs’ button the user can dynamically add other dependency chains to the Main View in the center where the seperate chains are shown in an orthogonal graph.

Consolidation To show the greater functional network the user can consolidate all or alternatively a selection of dependency chains. After selecting all chains of interest and clicking the “consolidation” button, WiKeVis computes the unified graph, adds clusters of systems and subsystems accordingly and automatically switches to the c-mode where the resulting clustered graph is represented. Unfortunately, no specific cluster information was available with our data and also was not—as previously arranged—delivered by our collaborators during the project. We therefore provided two alternative possibilities for clustering the graph:

1. *Manual clustering:* The user can add her/his own semantic knowledge about sub/systems by specifying them in the GraphML file.
2. *Automatic clustering:* We provided a defined programming interface for engineers to specify automatic rules and algorithms to cluster FBs. Due to the reason that we did not get any input on this aspect during the development of WiKeVis (this can only be done be domain experts!), we implemented a showcase clustering algorithm based on FB connectivity within the graph and used it for tool presentation and evaluation purposes.



Name:
Standheizung Ein/Aus
Typ: Knoten
Anzahl Nachfolger: 1
Anzahl Vorgänger: 0
Kindgraph vorhanden? Nein

Suchergebnisse

Knoten	# Eingäng
Außenspiegel automatisch	3
Aktor Außenspiegel autom	1
Aktor Außenspiegel autom	1
Außenspiegelglas bei Rück	2
Aktor Vertikalverstellung A	1
System: Innenspiegel aut	0
Innenspiegel automatisch	1
Innenspiegel automatisch	1
System: Außenspiegel bei	4
Außenspiegel beiklappen/	2
Aktor Außenspiegel beikla	1
Aktor Außenspiegel beikla	1

Figure 4.8: WiKeVis in c-mode: Bottom left shows search results, the main view in the center represents the consolidated graph, top right gives an overview of the entire graph and outlines the current segment shown in the main view, and the bottom right shows the hierarchy which is used for clustering the graph. All views are coordinated.

c-mode The consolidated graph shown in the c-mode is usually much larger as a single dependency chain (cf. the “wallpaper”) and can contain as much as 2500 FB nodes. To make the representation usable on a normal laptop/desktop screen we therefore used superimposed zoomable clusters to scale the graph and to provide the user with an opportunity to interactively explore interesting regions of the graph. For this purpose, we used the variable zoom display method for 2d networks proposed by Schaffer et al. [SZG$^+$96] utilizing a fisheye focus and context technique for interactively folding and unfolding clusters in the graph (in our case, sub/system-clusters). For user awareness reasons, we retained black for “normal” FB-nodes and used green to indicate sub/system-cluster nodes. In addition, cluster-nodes got a magnifying glass icon suggesting that they can be zoomed by the user.

Initially, the consolidated graph is represented entirely folded to provide adequate overview, the user then can explore it by interactively unfolding sub/systems (clicking on the magnifying glass or using the mouse wheel while over a clustered node), by panning and zooming, and by using additional views that we implemented for navigation, orientation and search purposes. An *Overview* provides context awareness by indicating the current section shown in the Main View correlated to the entire graph (overview and detail, cf. Section 2.2). An extra *Tree View* visualizes the underlying cluster hierarchy using a top-down node-link tree, i. e., sub/systems are the inner nodes, FBs are the child nodes (Our studies showed that Overview and Tree View, however, might not be worth the screen space we invested for showing them, cf. Section 4.2.4). Main View, Overview and Tree View are closely coordinated by using linking and brushing, synchronous scrolling and panning, but also coordinated interaction for un/folding clusters (cf. Section 2.2). Depending on the tool setup hovering also highlights direct predecessors and successors as well as the paths to them in order to better support users in following these connections. In addition, we provided a search function that highlights all results in the respective views and additionally presents them in a traditional text-based *Search List*. Clicking on a specific search result in the list automatically navigates and zooms to the selected object. Another feature which was requested during the implementation by one of our collaborators was the *interactive flagging* of ins and outs of a sub/system. The interactive flagging allows for marking one or several ins (a small port where the lines hit the boxes) of a sub/system with little flag symbols and results in highlighting all paths to reachable outs. This helps the user to better and faster understand the behavior of a sub/system depending on a specific communication input. Accordingly, this can be done vice versa for marking outs and receiving dependent ins. Figure 4.8 shows how the c-mode in WiKeVis.

4.2.4 Evaluation

We conducted a qualitative user study with five domain experts working with dependency chains (not the participant of our pre-design and design studies) and five graduate HCI/InfoVis students. Our general goal was twofold, on the one hand we wanted to get usability feedback and on the other hand we aimed at expert estimations about the potential value of our approach for their daily work.

Methodology Our study used a think-aloud protocol where we let our subjects conduct short tasks and encouraged them to give both positive and critical feedback. We used a task set of 14 tasks for our students, and a subset of nine tasks for domain experts in order to have more time to freely discuss our solution with them. The engineers' task-subset has also been fine-tuned as we were allowed to use "real" data in this study setup (see below). The tasks included exercises in using the tool (tool tasks) and graph problem solving tasks such as finding a specific node, edge, or path (graph tasks). Table 4.3 shows the set of 14 tasks and marks which of them we tuned for using with our domain expert participants[8].

We audio-recorded the sessions with students and due to IPR restrictions used note-taking with experts. Subsequently, we provided our subjects with a questionnaire (5-point likert scales where 1 equals best and 5 equals worst) for gathering subjective ratings. For testing our tool with students we created an synthetic dataset with 30 dependency chains (IPR restrictions!). For the

---

[8] Due to IPR restrictions, we are not allowed to print the domain specific tasks: However, we usually only changed the names of the components to their original names and therefore the student tasks can be seen as meaningful substitutes.

Nr.	Task	Category	Engineer Task
1.	Load the dataset ‘test.xml’ and freely explore it with the tool!	tool	yes
2.	Which graph in the d-mode holds the most nodes?	graph	no
3.	Select three graphs in the d-mode and consolidate them!	tool	yes
4.	Open the nodes ‘System A’, ‘System B’ and ‘Subsystem AA’!	graph/tool	yes
5.	Additionally import the dataset ‘test2.xml’ and re-consolidate the graph so that it includes the dependency chain ‘test2–chain2’!	tool	no
6.	Find the node ‘FB-lights-out’!	graph/tool	yes
7.	How many outs and ins has this FB?	graph	yes
8.	With which other FBs does this FB communicate?	graph	yes
9.	Find all nodes with “is” in their label!	tool	no
10.	Is there a direct connection between the node ‘FB-key-pressed’ and ‘FB-window-open’?	graph	yes
11.	Which nodes are between the ‘FB-key-pressed’ and ‘FB-lights-out’?	graph	yes
12.	Show a node of your choice in all views of the tool!	tool	no
13.	On which hierarchy level are the nodes ‘node-a’, ‘node-b’ and ‘node-c’?	graph/tool	yes
14.	To conclude, please export the consolidated graph!	tool	no

Table 4.3: Tasks (paraphrased from German), the rightmost column indicates whether we also tuned the task for our domain experts or not.

studies with experts we used the Excel sheet we got from our design collaborators. Unfortunately, it turned out that this dataset was highly incomplete and, thus, not directly usable for our study. By manually preparing the data we got 39 dependency chains (out of 412)[9]. Due to the smaller, manually prepared datasets we therefore cannot claim any implications about our solution’s scalability from our studies. Previous to the actual study, we conducted a pilot study with one automotive engineer and refined the study accordingly. Each study took approximately one hour.

Results The general feedback from students and engineers was good. Especially, using clustering with variable zoom-levels (cf. Schaffner et al. [SZG+96], average ranking 1.0), the general interaction concept of the tool (pan+zoom: 1.4, clustering navigation: 1.2, brushing+linking: 1.6, labeling: 1.6) and the search+autopan (average rating 1.2) were evaluated with very good ratings regarding usability. The consolidated graph view (in the main view) was evaluated to be the most important view (average rating 1.0), while interestingly the overview was rated as least important (3.0) and also the Tree View was rated as rather unimportant (2.8). This may be explained by the size of our dataset which was much smaller than a real dataset where an overview and an additional Tree View might be of higher value. On the other hand, there have also been several critical comments on the usability, particularly regarding the graph layout we had chosen (cf. Biedl et al. [BMT97]). Most notably (seven times), the subjects complained to have problems with follow-

---

[9] In spite of multiple inquiries and promises of our collaborators to deliver correct data at the latest for the user study, we did not get a usable version of a larger dataset.

ing long paths with the graph layout we had chosen. Our participants suggested choosing another graph layout, providing additional brushing of paths or allowing the user to adapt the graph layout manually (e. g., by dragging and dropping nodes and edges). Another aspect of critique was the magnifying glass interaction due to its fixed position in the bottom-right position of cluster nodes.

Generally, the experts' estimation about the suitability of our approach for the domain tasks at hand was good. Especially the separation between d- and c-mode, the novel comprehensive perspective on the data and the way to explore it were pointed out as adequate solutions, as, for instance, stated by one participant: *"[WiKeVis] gives me an overview but at the same time allows me to exactly zoom into areas which might be interesting to have in more details. That's good!"*. However, based on the engineers feedback we identified two major factors that would hinder a usage of WiKeVis for daily practices. First, the graph layout was particularly criticized by the engineers group as not practicable for their work. While they honored the usages of boxes and orthogonal lines, the basic layout problem turned out to be positioning of nodes. Instead of the layout we had chosen, all domain expert participants would strongly prefer a "flow-like" layout ordering nodes from left to right, as stated by one dependency chain designer *"we all use a left-right layout, by now this is quasi an implicit standard for us"*. While for the d-mode this was estimated to be crucial, two of our participants, however, doubted whether a clustered, left-right layout would scale in the c-mode, as mentioned, for instance: *"However, maybe for the consolidated graph this layout [i. e., Biedl's layout] is the better choice. But I'm not sure if this would be really accepted [...]"*. Second, WiKeVis as it stands does not support all necessary tasks when working with dependency chains. In this line, engineers most importantly missed an editor functionality to design new and change existing chains (which we excluded by purpose in our requirement analysis): *"We always need to quickly change between editing chains and representing them"*.

4.2.5 Discussion

In this section, I described WiKeVis, a tool for visualizing dependencies in functional networks in a more comprehensive way than current tools. We cooperated with two engineers working in the area of functional network design and collaboratively developed a tool for visualizing "dependency chains". A qualitative user study indicated the general suitability of our approach consolidating these chains to a larger functional dependency graphs and allowing the user to interactively expand and collapse clusters in this graph. However, we also got critical feedback most importantly on the graph layout that should be changed to a left-right orientated "flow-layout". Furthermore, explicit interaction techniques supporting users in tracing long communication paths should be considered in future designs. And last, we learned about the importance to combine data editing and visualization concepts in early development phases. As data in these phases is rapidly adapted and changed, for engineers a seamless switch between data inspection/exploration and data entering/changing is crucial. Closely combining and integrating Info-

Vis/VA techniques with proper techniques for editing the data therefore is a promising approach to support network development engineers with their tasks at hand.

From an development process perspective, the project on WiKeVis also provided us with some lessons learned. WiKeVis was initiated, actively supported and promoted by one single engineer, the stakeholder of this engineer-initiated solution. This stakeholder had already a very specific idea of a solution and therefore disapproved user-centered design techniques that we suggested. For our pre-design studies (cf. Section 4.2.2), for instance, we planned to additionally conduct observational studies and/or contextual inquiries with end users, however, we unfortunately encountered negative attitude towards our plans, as stated by the stakeholder: *"[...] we discussed everything important in our meetings. I am definitely sure this [observational studies/ contextual inquiries] won't provide any additional benefit"*. Instead, a collaboration with a group leader of the prospective end users and another stakeholder's colleague were selected for collaboration purposes. At the project's retrospective this setup, however, led to several disadvantages for WiKeVis' design. Though the group leader worked in the target domain, he was rather concerned with project management tasks and actually was not directly involved in the work with dependency chain engineering. Furthermore, due to his position he was very restricted in availability. The additional engineer also was no explicit expert/end user in working with dependency chains, however, our concerns about that aspect were not taken into account. The result was, that several important design requirements, such as the predominant graph layout preferences, revealed not until the summative user studies at the very end of the project. While also important at that moment, by immediate integration of end users into design processes some of them would have definitely emerged earlier on and could have been integrated and tested with WiKeVis' design study. Another obstacle in this project was acquiring adequate datasets. Contradictory to the warranty of our collaborators, we did not get any usable data to test our approach with.

Based on this experience and in line with findings from user-centered and participatory design research [BKS04], we want to summarize our lessons learned as follows: First, we recommend collaborating with end users as early as possible. This implies to carefully plan and conduct pre-design studies that can help enormously to get a better design (cf., for instance, MostVis). Second, end users cannot be replaced by collaborating with stakeholders, what stakeholders tell you can differ from what end users really do. Setting up collaborations with end users might not always be easy and obstacles such as time restrictions have to be overcome. However, we argue that it will pay off in a better visualization design. Last, if possible, data in usable form should be available right from the start of the project. Otherwise, working with synthetic data throughout the project holds the risk of designing a tool that does not scale to real world data and naturally will weaken a study aimed at realistic conditions.

4.3 *RelEx*: Visualizing Physical Networks

While WiKeVis dealt with functional networks, in this section I introduce RelEx, a visualization system to explore various relations in physical network specifications. To better understand these connections, currently automotive engineers mostly use static topology maps and textual interfaces such as the BNE client (cf. Section 3.3.3 and 3.4.2) as their most important data analysis tools. While these tools are helpful for understanding trivial connections between two entities such as ECU connections to bus systems (topology map), signals in a message or FBs of an ECU (both BNE), they fail in providing more complex connections such as signal-paths (which route does a specific signal take in the network) and lack in providing overview of communication relations including signals and/or messages. For the design of RelEx we[10] collaborated with in-car network architects, analyzed their working practices and derived a set of design implications. Subsequently, we designed RelEx with the goal to visualize signal communication and to provide a richer understanding of connections and dependencies in the physical network. Most importantly influenced by Henry's and Fekete's Matrix Explorer [HF06], RelEx also uses a multiple coordinated view approach and carefully combines traditional automotive representation techniques (an interactive topology map) with an adjacency matrix and a node-link-representation for signal paths. In doing so, on the one hand we hoped to retain current mental models of engineers and on the other hand to extend working practices with novel representation techniques. Thoughtfully linking the different perspectives with MCV interaction techniques furthermore supports the exploration of various connections within the data. We closely integrated RelEx with the BNE 2.0 and conducted several user studies, most notably a study with domain experts using the tool over several weeks in their daily practice. In doing so, we collected examples where RelEx was successfully used to provide novel insights into connections or to simplify work compared to current practices.

4.3.1 Pre-Design Studies and Requirements Analysis

The goal of our pre-design studies was to get a good understanding of the our target group's, i. e., network architects' tasks, their perspective on in-car networks, and to discover challenges and problems with their current practices.

Methodology To achieve our objectives, we used contextual inquiries [HJ93], a focus group and semi-structured interviews. For the contextual inquires we engaged five domain experts, all of them working in the area of network architecture and conducted between two and ten sessions per participant where each session took between five minutes and one hour. During the sessions the participants were asked questions regarding their field of activity, the tools they use for their daily activities, and challenges they are confronted with. Then, we asked them to demonstrate typical and important tasks, observed them in conducting these tasks. We interrupted only for clarifying

---

[10] Any use of "we" in this section refers to Michael Sedlmair and Annika Frank.

connections and activities. Luckily, we were located very close to the end-users working place, so we also asked them to spontaneously inform us when they had to conduct typical tasks that might be interesting for us to understand. This was very helpful for us in order to observe tasks under realistic conditions. In addition, we conducted a focus group study with four domain experts aiming at a better understanding of goals and challenges in working with signal paths. Providing signal path information is a relatively new concept (cf. similar to dependency chains, see Section 4.2.1) describing how a signal spreads over the physical system. As this concept, however, was not yet integrated into current working practices (not supported in BNE, but it will be supported by BNE 2.0) we actively looked for engineers involved in this innovation and engaged them in our focus group discussion. Finally, we conducted guided interviews with three carefully selected lead users in order to focus and evaluate our findings. In all studies we took notes to log information and statements.

Results of User Studies The main task at hand of the target users we observed was network change management, i. e., adding, changing, removing components to the physical network specification and re-configuring communication processes respectively. After the functional network had been partitioned to the physical specification (cf. Section 3.2), our target users were basically responsible (a) for optimizing the this network configuration, e. g., reducing cross-correlations between ECUs, and (b) for implementing change requests put by other engineers or development teams. After breaking down the necessary changes to concrete modifications (e. g., adapting a signal, adding new senders/receivers to an ECU, or changing a signals routing), our target users integrated them into the BNE, intensively validated network correctness and finally updated the network specification by releasing a new version. During this process, they frequently were engaged in exploring both the current network specification but also the temporarily changed data. By inquiring and observing our users, we found that during these exploration tasks, they particularly focused on understanding connections between ECUs, bus systems and signals: Which ECU is communicating with which ECU? Which signals do they exchange? Is a specific signal available on a bus system? What is the path a signal takes? For this purpose, our participants most importantly used three tools—topology maps, the textual BNE client (cf. Section 3.4.2) and occasionally a tool called BN-Communicator. The topology map was used for understanding connections between ECUs—similar to how a public transportation map, for instance, is used to find a way between two stations: Identify start/destination station/ECU, trace lines to learn about the connections between the two nodes, investigate where to change/investigate gateways (cf. Ware's description of visual tasks for reading an underground map [War08]). Second, the BNE client provided our users with information about all kinds of trivial connections, e. g., signals received by an ECU, signals of a message, or bus system over which a signal is sent. To investigate signal paths, i. e., how they spread over the physical network, our participants usually combined the topology map with trivial connections they derived from the BNE client and often manually sketched signal paths to make their knowledge explicit. Alternatively, some of our participants additionally used a BNE-plugin called BN-Communicator that allows the user to add sender ECUs to a list on the left side and receivers ECUs accordingly to the right (see Figure 4.9).

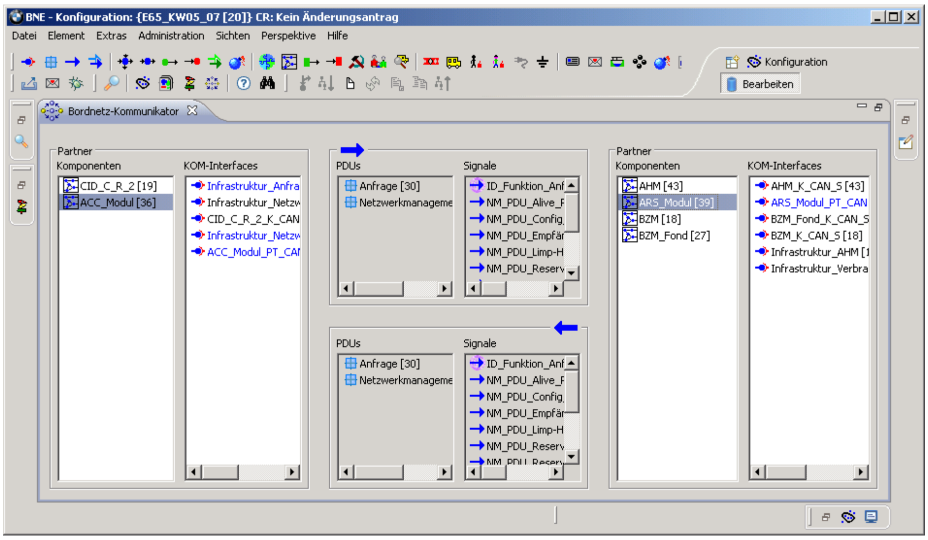


Figure 4.9: Screenshot of BN-Communicator, a module of the BNE allowing the user to explore communication between two ECUs.

Selecting a specific sender-receiver pair from these lists then shows involved communication elements (i. e., PDUs and signals) between them in the center of the application.

Interviewing our target users revealed that for them understanding connections in the physical network were of highest importance. However, most of them felt restricted in exploring these connections with current tools. According to our participants statements, a main current limitation is the restriction to showing trivial connections between the networks' components, i. e., direct connections between two elements, missing overview techniques including communication elements such as signals and messages, and the absence of suitable representation techniques for signal paths. In discussing our findings with the lead users in the final interviews, we collaboratively decided to initially focus on the visualization of connections between signals, bus systems and ECUs/gateways. We had two basic reasons for this decision. First, these components are the fundamental components of in-car communication and were cited by all our participants as most important for their work, as an engineer, for instance, said: *"A good solution for better showing signal communication in the physical network [i. e., in relation to ECUs and bus systems] is most urgent"*. Second, our lead users stated that other connections such as signal packing to messages or PDUs, while also important, would additionally pose complexity to the visualization that is not necessary/desired to address in a first design study. All lead users argued for starting with a focus on the fundamental components (i. e., ECUs/gateways, bus systems and signals) and validate a visualization approach for them before extending it to other components.

Classification of Connections Based on this focus, in a next step we described the connections between signals, ECUs/gateways and bus systems—the elements that will be visualized—systematically in order to classify current tools (topology map, BNE client and BN-

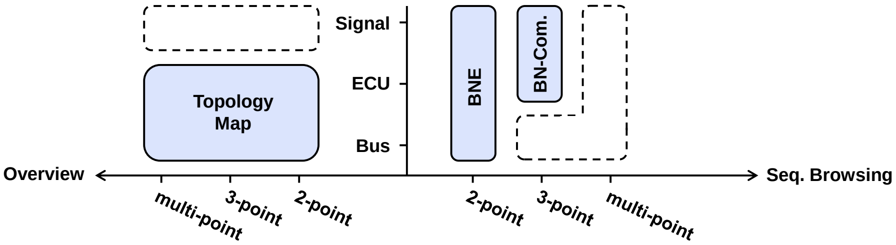


Figure 4.10: Categorization of current tools according to the grammar we used for describing all connections between ECUs/gateways, bus systems and signals (cf. Paragraph ‘Classification of Connections’ in 4.3.1). The white rectangles show lacks of current techniques.

Communicator) and to identify their gaps in providing insights into these connections. For this purpose, we used a simple grammar based on (a) counting the elements (signal, ECU, bus system) involved in a specific connection, (b) distinguishing the kinds of elements involved in a connection, and (c) the capability to provide overview of connections. To do so, we defined trivial connections as 2-point connections comprising two different elements, e. g., the connection between an ECU and its receiving signals (ECU–SIGNAL). The BNE client is designed to explore all kinds of 2-point connections. In this line, the BN-Communicator provides insights into 3-point connections between two element groups, ECUs and signals (ECU–SIGNAL–ECU), and with the topology map even multi-point connections between bus systems and ECUs are possible (ECU–BUS–ECU/GATEWAY–bus–ECU/GATEWAY–...). In addition, we differentiate whether all possible connections are shown at once (overview) or have to be sequentially explored by the user (seq. browsing). While the topology map, for instance, provides an overview over all ECUs and bus systems, neither the BNE client nor the BN-Communicator have the capability to completely represent more then one element group at once. Figure 4.10 graphically illustrates how current tools can be classified according to our grammar and shows their strengths but also gaps that currently exist in the representation of connections. The graphic particularly reveals that connections underestimated in current tools are multi-point connections, connections between all three categories and overviews involving signals. For multi-point connections our findings showed that especially signal paths are of high importance (i. e., ECU–SIGNAL–ECU–SIGNAL–ECU–...).

Design Implications Based on the results of our field studies, the focusing aspects of the final interviews and our connection analysis discussed above, we summarized the following design implications for our tool:

1. *Focus on connections between ECUs, bus systems and signals:* According to the recommendation of the three lead users we decided to focus on the three most important aspects for our target group: ECUs (including gateways), bus systems and signals, and to intentionally abandon representing messages and PDUs.

2. *Maintain topology map:* Understanding physical aspects of the network, namely ECUs, bus systems and how they are linked together is highly important for our target group. For this purpose, engineers currently use topology maps and very frequently consulted them during the tasks we observed. As far as possible, a topology map representation should be reused for that as they are familiar and efficient tools for supporting engineers' tasks.

3. *Support overview including signals:* This implication can be derived by looking at our systematic inspection of current representations and was confirmed by engineers by frequently mentioning it during our expert interviews. Though, connections between signals and other elements (ECUs, bus systems) are supported by current tools, they merely can be explored sequentially and there is currently no representation that provides an overview of signals and their connections to other elements.

4. *Support exploration of multi-point connections, most importantly signal paths:* This was one of the initial project ideas and therefore should be definitely supported. Signal paths are becoming more and more important in the work of our target group as vehicles' functionality is increasingly distributed. The next generation in-car network database BNE 2.0 will also support signal path's storage leading to their integration into formal specification documents (see also Section 4.2.1, dependency chains). However, currently there is no adequate visual representation of signal paths available.

5. *Support exploring connections between all (i. e., three) element categories:* This implication is derived from our systematic inspection of the data and its current representation techniques. The inspection showed that current techniques are mostly restricted to show connections only between two categories of elements (e. g., topology map: bus systems and ECUs, or BN-Communicator: ECUs and signals). Overcoming these restriction might provide engineers with a better understanding of comprehensive correlations.

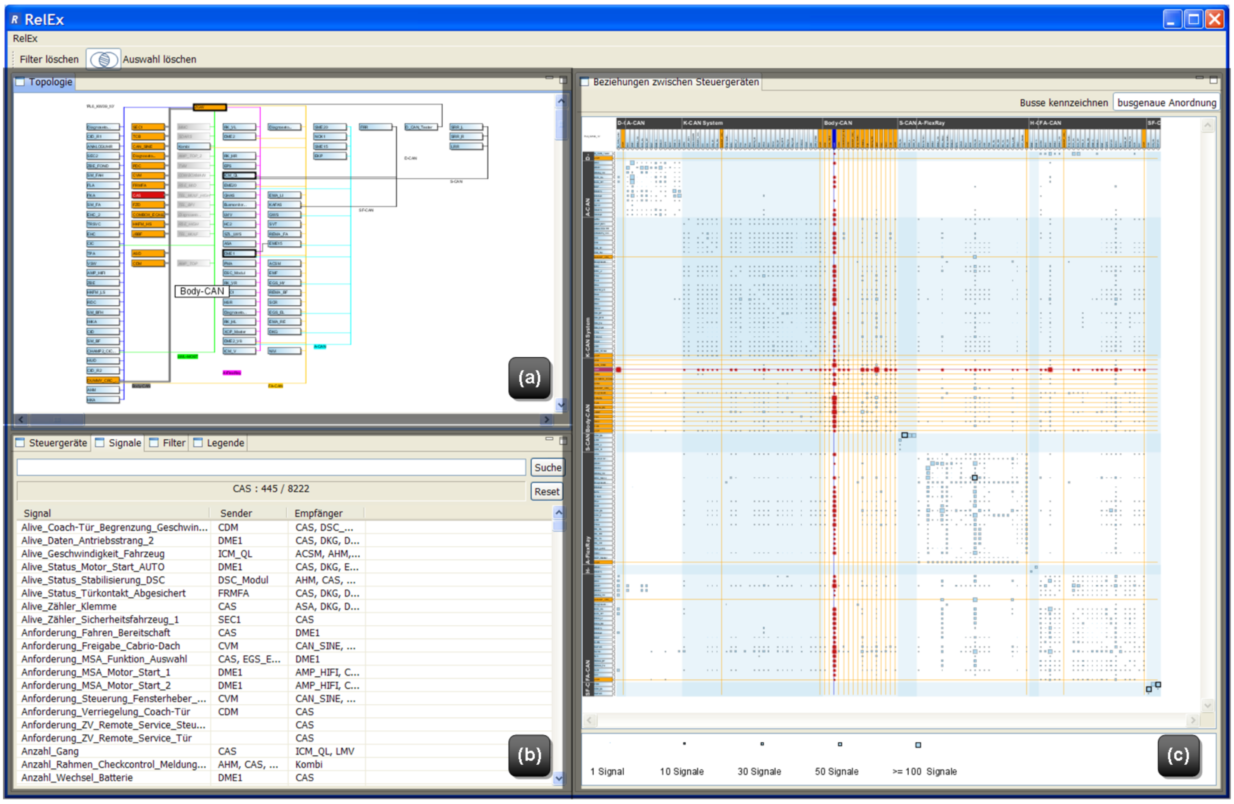
4.3.2 Design and Prototype Description

Based on the implications from our pre-design studies, we designed and implemented a prototype called RelEx (Relation Explorer). In line with the previous prototypes, we used multiple coordinated views: (a) A view virtually replicating a topology map, (b) a view providing overview over all ECU–SIGNAL and BUS–SIGANL connections based on an adjacency matrix, (c) a node-link diagram showing signal paths of a specific signals, and (d) List Views to provide all elements additionally in traditional text form. We iteratively designed RelEx with three domain experts (lead users) and discussed design decision and alternatives with these engineers. Based on the final concept, we implemented a software system using Java and the piccolo framework[11]. Fortunately, one of the lead users we collaborated with was closely involved in the development of the BNE 2.0[12] and, thus, we got a special permit for using the BNE 2.0 data connection and for closely integrating our tool with its user interface. As during our project the release date

---

[11] `http://www.cs.umd.edu/hcil/jazz/`

[12] See Section 4.1.3 about MostVis for a closer description of BNE 2.0.



CAS : 445 / 8222

Signal	Sender	Empfänger
Alive_Coach-Tür_Begrenzung_Geschwin...	CDM	CAS, DSC_...
Alive_Daten_Antriebsstrang_2	DME1	CAS, DKG, D...
Alive_Geschwindigkeit_Fahrzeug	ICM_QL	ACSM, AHM,...
Alive_Status_Motor_Start_AUTO	DME1	CAS, DKG, E...
Alive_Status_Stabilisierung_DSC	DSC_Modul	AHM, CAS, ...
Alive_Status_Türkontakt_Abgesichert	FRMFA	CAS, DKG, D...
Alive_Zähler_Klemme	CAS	ASA, DKG, D...
Alive_Zähler_Sicherheitsfahrzeug_1	SEC1	CAS
Anforderung_Fahren_Bereitschaft	CAS	DME1
Anforderung_Freigabe_Cabrio-Dach	CVM	CAN_SINE, ...
Anforderung_MSA_Funktion_Auswahl	CAS, EGS_E...	DME1
Anforderung_MSA_Motor_Start_1	DME1	AMP_HIFI, C...
Anforderung_MSA_Motor_Start_2	DME1	AMP_HIFI, C...
Anforderung_Steuerung_Fensterheber_...	CVM	CAN_SINE, ...
Anforderung_Verriegelung_Coach-Tür	CDM	CAS
Anforderung_ZV_Remote_Service_Steu...		CAS
Anforderung_ZV_Remote_Service_Tür		CAS
Anzahl_Gang	CAS	ICM_QL, LMV
Anzahl_Rahmen_Checkcontrol_Meldung...	AHM, CAS, ...	Kombi
Anzahl_Wechsel_Batterie	DME1	CAS

Figure 4.11: Screenshot of RelEx: **(a)** The topology Map View; **(b)** the List View; and **(c)** the Matrix View showing the signal communication between all ECUs of the network. Below the matrix view is the legend of communication-boxes' size coding.

of the BNE 2.0 was postponed, we decided to provide our RelEx system both as an BNE 2.0 eclipse plugin (for a close integration in the future) and as a standalone tool in order to allow users to work with export data in the meanwhile. Figure Figure 4.11 shows a screenshot of the final system.

Next, I describe the design of RelEx in more detail and discuss our design decisions as well as alternatives which we considered during this process.

Topology View: Showing physical connections Along with findings from our previous project on WiKeVis underlining how important it is in our domain to retain predominant representation techniques (Section 4.2.5), our first view is directly derived from the frequently used topology maps. By retaining this highly familiar representation, we aimed at supporting the engineers' mental model in order to increase our tool's usability, learnability and user acceptance. Initially we considered two kinds of solutions for representing the topology map in RelEx: (a) Using a static image and overlay all additional information and interactions, or (b) generating an automatic layout solely based on the data. A definite advantage of using static images would be that they exactly look like the ones engineers designed. However, on the downside overlaying information can quickly result in high clutter (cf., for instance, Section 3.4.3), requires manually reloading them, and implies changing the tool with each change in topology map's layout (the layout is not provided in a machine-readable way, e. g., xml). Automatic generation on the other hand allows for directly computing the layout from any data without additional effort for engineers. Besides, it is much easier to make an automatic generated tool interactive and to include dynamic information, e. g., for highlighting ECUs or bus systems. However, as there is no defined layout algorithm for topology maps, the final layout will most likely vary from manually drawn layouts (cf. BNVis, Section 3.4.3). A valuable alternative is to automatically lay out the design and thereafter let the user manually adapt the layout (cf. BNVis). Finally, we decided to implement an automatic layout and to approximate the manual design as closely as possible. Our main reason was our experience that additional time costs for engineers—e. g., for loading a topology map update—usually outweighs all other criteria and often results in non-usage of a tool.

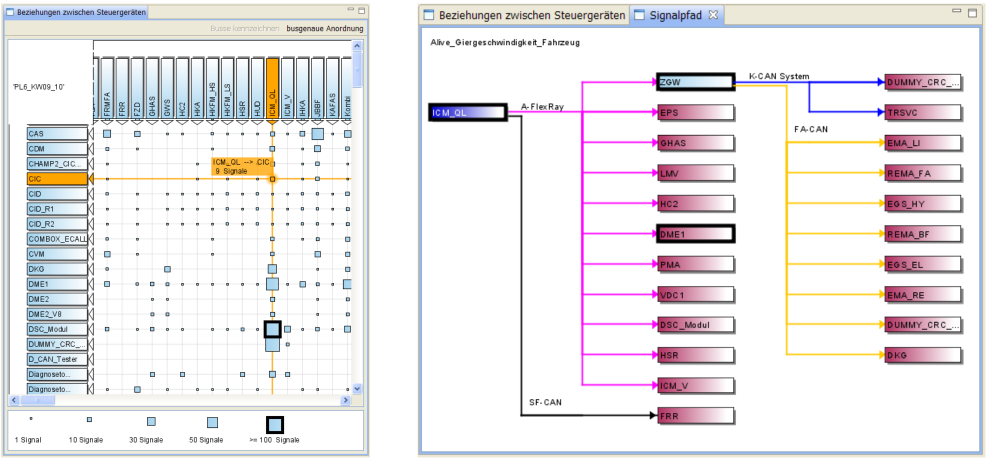
Based on our decisions, we came up the following layout: For the automatic topology map we used labeled rectangles representing ECUs, and for bus systems orthogonal lines color-coded according to standard topology maps. As required by our lead users, Gateway ECUs, i. e., ECUs that connect more than one bus, additionally receive a thick black frame. In short, the layout algorithm locates the central gateway ECU in the top center and then horizontally sorts all other ECUs according to their bus connectivity. After that, the resulting vertical stacks of ECUs are connected with orthogonal bus-lines labeled at the bottom of each line. Similar to manually drawn topology maps (cf., for instance, Figure 3.2 on page 27) we located outgoing lines on the right side of the ECU boxes (for simplicity reasons, we did not use double stacks as often utilized in manually drawn maps). Finally, stacks and lines are re-ordered to avoid crossings of bus-lines and ECU-nodes (avoid node tunneling) and to minimize line crossings. Figure 4.11-a shows the resulting layout from an in-car network specification with approximately 100 ECUs and nine different bus systems. To allow the user, to focus on specific sub-sections of the network with maximizing this view, we implemented the Topology View as a zoom and pan interface.

Signal Matrix: A novel overview In our pre-design studies, engineers expressed a strong desire to get an overview over all signals and their communication between ECUs. From an InfoVis perspective this can be described as a graph where the ECUs are the nodes and the exchanged signals are the directed edges in the graph. As two ECUs can exchange more than one different signal (usually up to 100, sometimes more), the edges can be seen as weighted edges where the number of exchanged signals determines the weight.

We already discussed the visualization of graphs in Section 2.4 and in the previous section we used a graph in WiKeVis. Compared to WiKeVis where it was clear right from the start that we will use a node-link representation, our work on RelEx was strongly influenced by evaluating pros and cons of node-link (e. g., such as in [HB05, NSGS07]) and adjacency matrix (e. g., such as in [ZKB02, EDG$^+$08]) representations and how their benefits can be combined (e. g., such as in [HF06, HFM07]). A node-link representation would be beneficial for tracing signal paths, however, showing the entire graph would result in high clutter as the ECU/signal graph is relatively large and dense (e. g., 100 nodes with 3,000 single-directed edges). Based on Ghoniem et al.'s findings that visualizing large and dense graphs profits by using a matrix representation [GFC05] we, therefore, decided to show the signal overview with an adjacency matrix. However, as signal path tracing was another design goal of ours, we did not want to entirely abandon node-link representation as it is important for this task [GFC05]. Hence, we took a multiple coordinated view approach similar to Henry's and Fekete's Matrix Explorer [HF06], visualized signal paths in a node-link view and coordinated the views accordingly (see below for a closer description).

Figure 4.11-c shows what the final design of the signal matrix looked like. We positioned sender ECUs at the top row and all receiver ECUs at the left. For consistency purpose, the representation of the ECUs is equivalent to the one in the Topology View. In addition, we added small arrows to sender and receiver ECUs clarifying the direction of communication. At each crossing of a 'sender column' and a 'receiver row' the signals which are exchanged between this ECU pair in the specified direction are represented by a square (in the following we refer to them as 'communication-boxes'). The size of these rectangles codes the number of exchanged signals and all pairs that exchange more than 100 signals additionally get a thick black frame, to better support preattentive recognition of "big players". A legend right below the Matrix View provides the user with information about the size coding. To allow the user for additionally exploring bus systems in this view, we integrated a possibility to re-order the ECUs in the matrix. While initially the ordering is alphabetical, the user can interactively switch to ordering ECUs based on their bus connectivity. To support visual distinction between respective bus systems we added light blue colored "bus-stripes" to the matrix's background. As several ECUs (gateways), however, are connected to more than one bus, we had to decide between representing them redundantly or leaving out all but one bus connection. Based on our conversation with lead users, we decided to redundantly show ECUs in this mode and to generate awareness by simultaneously highlighting all appearances when hovered or selected (see below, for another example see also the selective highlighting concept introduced in Munzner et al.'s Constellation system [MGR]). Similar to the topology map, the Matrix View provides a zoom and pan interface in order to allow the user to investigate extracts in more detail. Not to lose context, the labels at the top and the left side stay visible all the time. Figure 4.12-a shows a screenshot of a zoomed Matrix View.

Signal Path View As stated above, another major goal of RelEx was to visualize signal paths. Due to the facts, (a) that path tracing is not well supported by matrix displays [GFC05] and (b) that our graph's density is too high for a comprehensive representation of all elements, we abandoned an overview representation and decided to provide signal paths sequentially on user interaction. For this purpose, users can select a specific signal to show the signal's path in an additional view. To derive the selected signal path from the data, we identify all edges of the ECU/signal graph that contain the signal as well as all ECU-nodes that are connected to these edges, resulting in a sub-graph similar to the transitive chain described in Section 4.2.1. The layout of a signal path representation is based on a left-right, orthogonal node-link representation which we derived from our pre-design observations of how engineers sketched them on paper



(a) (b)

Figure 4.12: Further screenshots of RelEx: **(a)** Zoomed-in matrix; and **(b)** the Signal Path View showing how a specific signal spreads over ECUs and bus systems.

and which also was influenced by the lessons learned from our project on WiKeVis (see previous section). The layout of the view is consistent with the style of the other views (ECU-rectangles, bus-colors, labeling, etc.). Figure 4.12-b shows a screenshot of a typical signal path represented in RelEx.

List Views, interaction and coordination In addition to the visualization views, we added several List Views for showing all ECUs and all signals respectively in the traditional way. Each List View includes a search box. A search request updates the lists and highlights all search elements in the visualization views. The other way round, selecting, i. e., clicking on an element in one of the visualization views also automatically initiates a search request and updates the list views. In doing so, we allow the user to freely explore all kinds of 2- and 3-point connections, e. g., selecting a bus system in the topology map automatically shows all signals that are sent via this bus in the signal list or all connected ECUs in the ECU list. Selecting a communication-box in the Matrix View updates the lists with sending and receiving ECUs as well as all signals that are sent between these ECUs.

Additionally, we carefully designed a brushing and linking concept which supports our target group's goals to explore connections. For highlighting elements, we generally used orange for hovering, a light red for selection, and blue/red in the case of sender ECUs (blue) and receiver ECUs (red) distinction. We chose blue and red because they were named by engineers as typical domain colors to code this information. Highlighting bus systems is done via slightly increasing the line strength in order to retain their color information. Both hovering and selecting an element highlights the element in question but in addition correlated elements in the same and in other views. Along with brushing the same elements in other views (1-point connections according to our definition), we provided several concepts to highlight correlated elements. The following list describes what additional connections hovering/selection in various views can reveal:

- *Hovering/selecting ECUs (Topology/Signal Path/Matrix/List):* Hovering or selecting ECUs in any view results in the Matrix View showing an additional cross line(s) vertically and horizontally "beaming" from the the hovered/selected ECU(s). These lines support users in tracing all signals, i. e., communication-boxes, relevant for the ECU(s). For further visual support, all communication-boxes on the cross lines are highlighted (cf. Figure 4.15 at the end of this section). If the matrix is bus-sorted and a hovered/selected ECU appears redundantly, all appearances are highlighted accordingly.
- *Hovering/selecting bus systems (Topology/Signal Path):* If a bus system is hovered or selected all connected ECUs in the Topology View are highlighted also. This supports faster detection of all connected ECUs, especially if the ECUs of a bus are connected to more than one bus and therefore located in different columns (cf. Figure 4.11).
- *Selecting signals (List):* Along with opening the Signal Path View, selecting a signal from the list highlights involved elements in all other views, i. e., all ECUs that send or receive this signal, all bus systems that transport this signal, and all communication-boxes (with cross lines) that incorporate this signal (cf., for instance, Figure 4.16 at the end of this section). This is also helpful, as in doing so a signal path is additionally highlighted within the familiar topology map representation.
- *Hovering/selecting communication-boxes (Matrix):* In this case the sending ECU A and the receiving ECU B are highlighted and the respective cross lines are shown. Additionally, the inverse communication, i. e., the symmetric box in the matrix (ECU B sends and ECU A receives) is highlighted as well. We integrated this based on our lead users' feedback in order to allow for the investigation of communication in both directions.

Also based on our lead users' feedback and requests during our user-centered approach, we integrated several other features, most importantly the capability for multiple selections (similar to the one in MostVis, cf. Section 4.1.2), and for interactively filtering bus systems and ECUs. Additionally, we integrated the possibility to create and specify new elements, i. e., ECUs/gateways, bus systems, and to edit/remove existing ones in order to allow our users to conduct specification changes directly in RelEx without changing to another tool first.

4.3.3 Evaluation

For the validation of RelEx's utility, we especially concentrated on the benefits of the matrix overview, the exploration of all kinds of connections via our multiple coordinated view approach and the comparison of RelEx to currently used techniques. For this purpose, we conducted a fine-tuned and down-scaled version of the MILC method proposed by Shneiderman and Plaisant

[SP06]. In doing so, we derived suggestions for improving RelEx (and integrated them if possible), got a variety of expert's estimations, comments and opinions, and most importantly found several usage examples where our participants successfully used RelEx for their daily tasks.

Methodology After conducting a heuristic evaluation with five graduate students and integrating their feedback, we provided RelEx to a group of seven automotive network architects who used the tool for a period of five weeks. Starting with an introduction at the beginning of the study, we scheduled weekly meetings of one hour length. In practice, we engaged in 3–9 rather spontaneous meetings per participant (due to participants' time restrictions) which lasted between ten minutes and one hour. In these meetings, we interviewed the participants about their experiences with the tool and encouraged them to give formative (suggestions for improvement) and summative (usage examples) feedback—we also gave them logbooks for that purpose, however, participants almost never used them. Based on the formative feedback, we iteratively refined, adapted or added additional features to RelEx, and provided our participants with weekly tool releases.

Additionally, we conducted a study using a think-aloud protocol with 10 domain experts. This study particularly focused on qualitatively comparing RelEx with current tools, namely (a) the visual differences between our Topology View and static topology maps, (b) comparing connection exploration tasks conducted with BNE/BN-Communicator and with RelEx, and (c) gathering subjective ratings. To do so, we prepared a set of representative tasks along with a questionnaire to evaluate RelEx (5-point likert scales, 1 best, 5 worst) and to rate the differences to current tools (3-point scale: worse–comparable–better). As we were rather interested in the qualitative feedback we abstained from measuring task completion time but intensively discussed benefits, drawbacks and opinions. In all our studies, we took notes for logging verbal feedback.

Usage Examples From our studies, we derived several usage examples that show how RelEx helped our participants in conducting daily tasks, or how they gained novel insights into the data by using it. Three engineers, for instance, used RelEx for replacing ECUs with other ECUs. For this task, RelEx turned out to be helpful in identifying and adapting all communication partners, deciding whether two ECUs might be consolidated or not by investigating differences and connections between the ECUs in question, or for finding out to which bus to connect a novel ECU. Another engineer who was relatively new in the area of in-car network engineering (two months) used RelEx frequently to familiarize himself with the network topology, communication processes in the network and especially to understand the intensity of communication between particular ECU pairs. Furthermore, RelEx was particularly used for exploring the current network specification and to derive room for improvement by gaining new insights into the data (see below). Finally, our prototype was used for ordinary tasks such as to identify all communication partners of an ECU, to investigate gateway communication, to explain decisions and facts to colleagues, or simply to refer to the topology map if a printed version of it was not available at that moment.

Based on these anecdotes, in the following I show how particularly the Matrix View—as it is the most innovative view in our tool—was used to gain novel insights into the data during the tasks described above. The examples can be classified into four groups:

- *Communication hot spots:* Due to the size coding and the additional black frame for very large communication-boxes, communication hot spots were well recognizable in the Matrix View and helped engineers in quickly detecting "communication hot spots" (cf. Figure 4.14-a at the end of this chapter).
- *Local/global communicator ECUs:* Together with the hovering and selection highlighting, the matrix was used to understand the amount of communication partners of a specific ECU. We distinguish between local communicators that have just one or several communication partners and global communicators exchanging signals with nearly all ECUs in the network (cf. Figure 4.15).
- *Introvert/extrovert bus systems:* Similar to the example above, the matrix was used to understand communication structures beyond bus systems. Sorting the Matrix View by bus systems helped engineers to distinguish between introvert bus systems—that have no or little communication with other network sub-systems—and extrovert bus systems—where much communication is forwarded and/or received from other bus systems (cf. Figure 4.14-b and c). This information was frequently referred to as being important for re-structuring and optimizing the network configuration.
- *Well-directed and wide-spread signals:* The Matrix View also helped to quickly get an impression of the importance of a selected signal. While well-directed signals exchanging information between two dedicated ECUs are represented by a single highlighted communication-box (or two: send and receive), wide-spread signals are immediately perceivable by multiple communication-boxes highlighted (cf. Figure 4.16).

Formative Feedback and Subjective Ratings During our think-aloud studies, we explicitly focused on comparing RelEx with current techniques (topology maps and BNE interface) and in doing so triggered more discussions about room for future improvement and possible extensions of our approach. First, we were particularly interested in comparing our automatic topology map layout to manually drawn maps. While the general feedback was better than we expected (similarity was ranked with 1.8; clear arrangement 2.1), we also got various suggestions for improvement. Some of our participants complained that the position of our ECUs is not exactly the same as the position in the original maps. For gateways it was suggested to position them more centric similar to the central gateway unit and the layout should support the representation of mutually exclusive ECUs as manually drawn layouts do. Regarding the comparison of RelEx to the BNE interface we received very good feedback (better ratings for 95% of all tasks). This result is not astonishing, as the work on RelEx explicitly focused on supporting connection exploration while the BNE was developed for data management purposes and simple exploration tasks, and

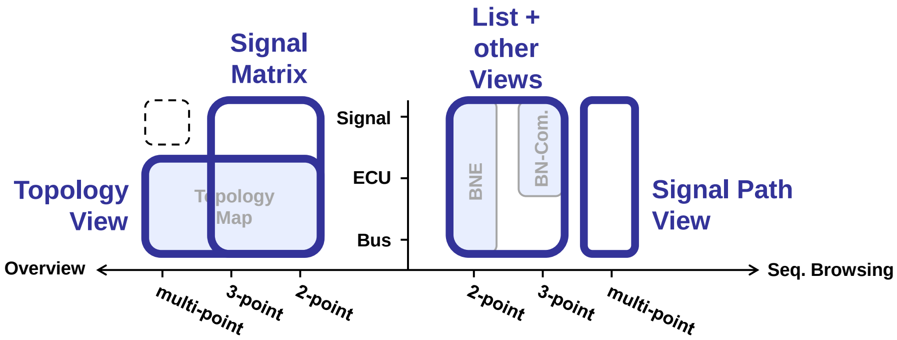


Figure 4.13: Categorization of RelEx split by its views according to the grammar we introduced in Section 4.3.1. In the background the classification of current tools is shown. The white rectangles in the top left corner shows the limitations of RelEx.

BNE's dedicated communication tool BN-Communicator lacks in richness of exploration opportunities. As major benefits of RelEx, our participants most frequently named the overview of the data RelEx provides, the richness of exploration opportunities, its comprehensibility and the fact to easily communicate aspects with RelEx. Four of our participants also particularly stressed the value of the Signal Path View as it shows all relevant elements of signal paths in an easy understandable and usable way. With current tools this information has to be derived manually and step-by-step. Several of our participants even argued to replace the fully integrated BNE tool, BN-Communicator with RelEx as soon as possible, i. e., once a solution for PDUs and messages has been integrated with RelEx. Especially integrating PDUs was frequently mentioned by our participants to be necessary to make RelEx a fully operational system, and therefore should definitely be addressed in future projects. Along with the extension to PDUs, two of our participants additionally suggested to improve the representation of sender/receiver ECUs in the matrix which is currently done via color coding and little arrows.

4.3.4 Discussion and Adoption

In this section, I presented RelEx, a multiple coordinated view visualization for exploring connections in and providing a novel overview of physical in-car network specifications. For categorizing connections in the network, we introduced a simple grammar of connections between signals, ECUs and bus systems and used it to classify current tools (Section 4.3.1). Figure 4.13 shows how RelEx can be classified according to this grammar and outlines how it unifies and extends connection representation of the current tools topology map, BNE and BN-Communicator. Along with allowing the user to sequentially explore 2-point and 3-point connections by the combination of List Views with the other visualization views, multiple-point connections, i. e., signal paths can be explored by using a specific view designed for this purpose, the Signal Path View. Finally, the Matrix View provides a novel technique to overview the entire specified signal communication in a physical network and to correlate them to sending/receiving ECUs and to transporting bus systems. While RelEx extends current techniques in these ways, our work did not focus on providing an overview of all multi-point connections, i. e., over all signal paths in the network (cf. Figure 4.13). For this purpose, a node-link diagram showing the entire network would be practicable.

Most notably, we evaluated RelEx in a user study with seven domain experts over a period of five weeks. We found several examples where RelEx successfully supported our participants in their daily work practices and where novel insights into the data could be gained. In general, the feedback we received was very positive and especially the combination and rich coordination of traditional techniques (topology map, lists) with novel visualization approaches (Matrix View, Signal Path View) turned out to be valuable in daily working practice. Given the strong time restrictions of engineers, we were rather surprised how often our participants used RelEx during the study. Usually, it is hard in our domain to convince users as they are accustomed to and effective with current tools and therefore convincing them of a novel solution could be hard. During the studies with RelEx each participant, however, frequently used RelEx in their daily practices. Even after our study, most of the participants continued to use our tool and—moreover—recommended it to colleagues for usage. To the best of our knowledge, currently approximately 15 engineers use the tool. Furthermore, the engineers we collaborated with in our user-centered design process have planned to overcome the current limitations we described above in the near future. By extending RelEx, they want to make it applicable to a wider range of problems and also useful for a broader audience.

RelEx can be seen as a best practice example how we successfully combined and fine-tuned visualization techniques from both the automotive domain and the InfoVis domain. Currently, RelEx is restricted by solely showing connections between signals, ECUs/gateways and bus systems. For future work, it is important to build up the insights we gained and to extend the system to show additional connections of other objects, most importantly of PDUs and messages. We also found room for improvement for the automatic design of our Topology View. Finally, the deployment of the BNE 2.0 in the near future will provide a good platform for conducting more user testing as we already designed RelEx as a direct plugin for that. With BNE 2.0 users will be able to use RelEx without any additional costs (i. e., by exporting data before using RelEx).

4.4 Summary

In this chapter, I introduced three systems for visualizing in-car specifications, all of them design studies for a specific domain use case, user group and dataset. **MostVis**, the first system, visualizes hierarchically structured specification catalogs of MOST functionality. In a controlled experiment with 14 domain experts we compared MostVis with current tools and showed that our visualization was significantly faster for search and browsing tasks that we derived from a pre-design field analysis. Based on these successful results, we had the opportunity to deploy MostVis with the companies most important software environment for storing, handling and exploring all kinds of specification data. The integration is currently conducted by an external company and the final, embedded tool will be available by the end of this/ beginning of the next year.

WiKeVis, the second system, was a design study to visualize dependencies in functional networks. We collaborated with two engineers working in early development phases where the functional network is specified via “dependency chains”, small pieces of correlations used to design the communication between functional blocks. WiKeVis consolidates dependency chains to a larger, hierarchical clustered graph and visualizes it in an interactively zoomable node-link diagram. A user study with five domain experts revealed positive feedback to our general approach but also suggested room for improvement especially in terms of the graph layout we had chosen. During the WiKeVis project, we also encountered several obstacles in our user-centered design process that finally led to a collaboration breakdown, however, that also provided us with several helpful lessons learned for our future projects.

Finally, I have presented **RelEx**, a system that allows exploring connections between signals, ECUs and bus systems. We carefully combined and coordinated representation techniques used in our domain with InfoVis techniques. A five week usage of our tools by seven domain experts revealed several examples of successful application in daily practices and especially our Matrix View provided novel insights into the data that could be used for optimizing the physical network specification. After our study, engineers continued using RelEx. Currently, we know from 15 engineers using the tool on an occasional and situational basis. In addition, our collaborators have planned to further extend RelEx in the near future.

To conclude with, I particularly want to emphasize the two design studies MostVis and RelEx. Both turned out to be helpful and successful applications in the automotive electronic engineering domain and can be referred to by other tool designers as successful examples. Especially, the strictly user-centered approach during the entire development process starting already with intensive pre-design studies and continuing with participatory design techniques turned out to be invaluable in a domain characterized by a very high degree of expertise. For the design of our tools, I believe that especially the careful combination of prevalent, domain specific representation techniques with novel approaches adopted from the research area of information visualization is a crucial factor for user acceptance and success. To design systems supporting users in an area where already a variety of other tools exist, it is important to understand strengths and weaknesses of these tools, not to abolish them but to try to find solutions combining the best of both worlds. Furthermore, we used two completely different ways to validate these design studies. Both, quantitative and qualitative studies turned out to be valuable in our domain, as they either helped us in recommending our tool for a company-wide deployment or in gaining insights into how the tool was used by engineers to solve real world problems.

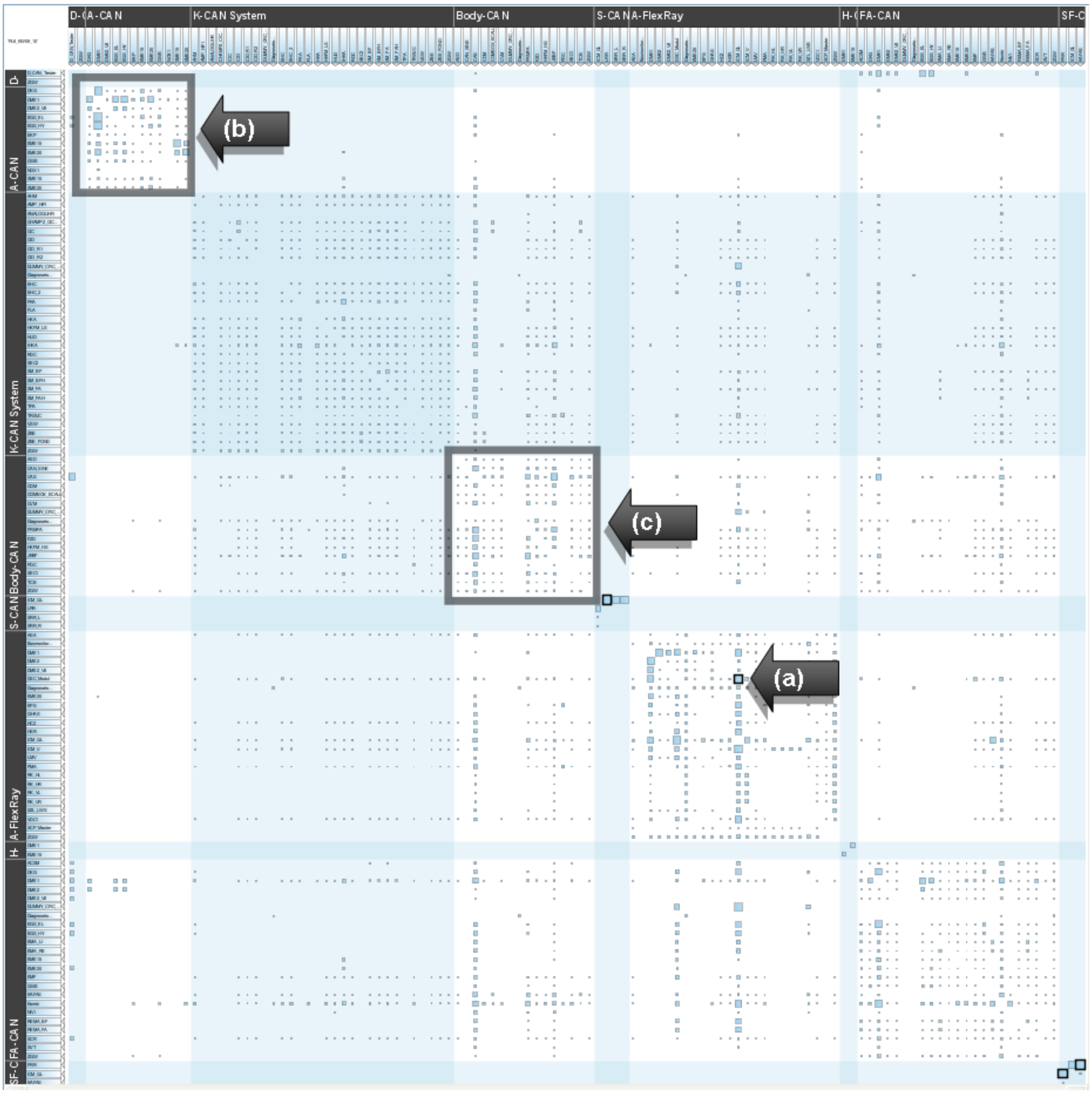


Figure 4.14: RelEx's Matrix View sorted by bus systems revealing **(a)** communication hot spots, as well as **(b)** introvert and **(c)** extrovert bus systems.

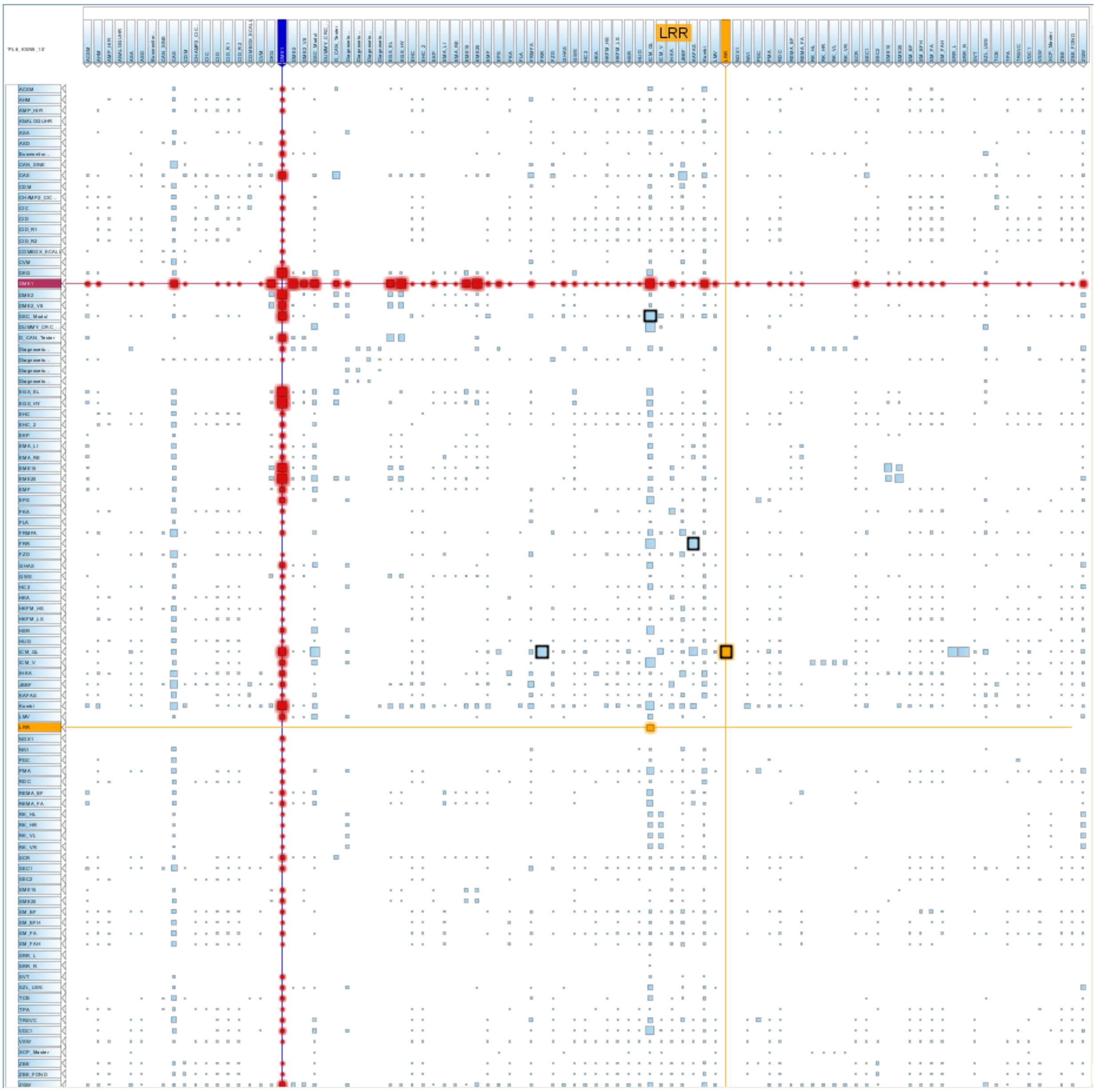


Figure 4.15: Selecting (red) and hovering (orange) in RelEx's Matrix View (sorted alphabetically). The red selection shows a global communicator ECUs, the hovered orange ECU is a local communicator exchanging signals with just one other ECU.

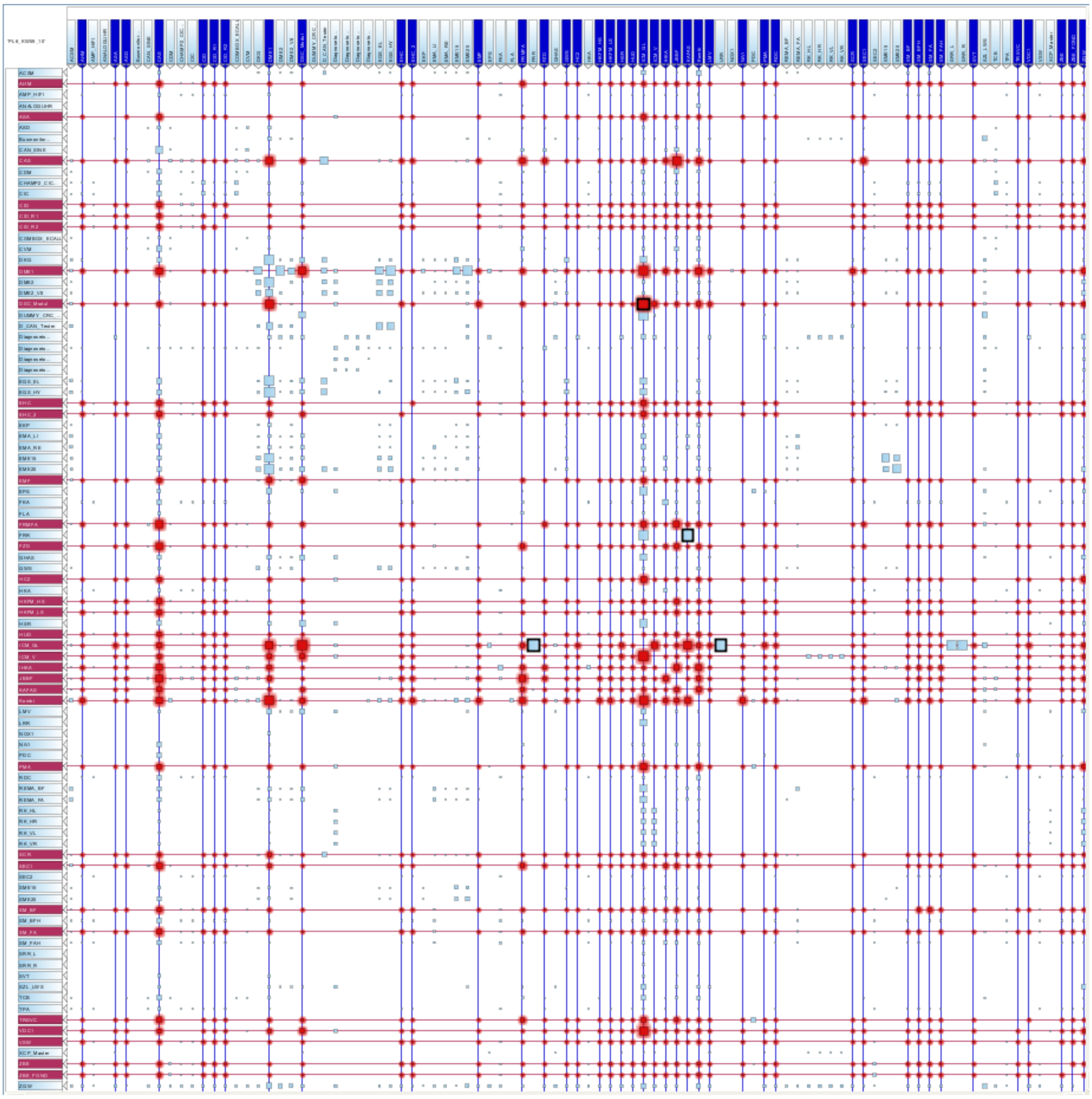


Figure 4.16: RelEx's Matrix View showing a wide-spread signal that is exchanged between a large part of the ECUs in the network.

Chapter 5

Visualizing Network Traces

The previous chapter focused on how in-car network specification data can be visualized and how development engineers (left wing of the V-Model) can be supported by these techniques. In this chapter, I introduce approaches for visualizing traces, i. e., large recordings of network communication logs that are used by electronic test engineers to track down errors in in-car networks (right-wing of the V-Model). During my early exploratory studies (cf. Section 3.5) it became clear that particularly the challenges of test engineers have risen enormously with the increased network complexity. Current in-car networks carry up to 15,000 messages per second, logging one hour of this data results in approximately 2GB and roughly 50 million recorded messages. Depending on the engineer's intention, traces normally vary from approximately 10s to up to 8h of data and therefore can become tremendously large. The studies showed that current tools for tracking errors are mostly text-based and particularly challenge engineers in terms of understanding the masses of data and to interpret more comprehensive aspects (cf. Section 3.5.3). As a consequence, much time and experience is needed to understand in-car communication processes and their complex correlations in order to detect sources of errors.

Within this thesis, I mainly focused on a specific group of analysts, namely **FIT analysts**. These engineers were the target users I collaborated with most intensively during my thesis, in total over a period of three years. FIT analysts are responsible for testing fully-built cars before start of production and are the largest and most important group of trace analyst experts at the company I worked. More specifically, I chose this user group for the following reasons: (a) In our converging focus group, FIT engineers were named as the most important group in trace analysis (cf. Section 3.5.4); (b) FIT analysts usually deal with the largest amount of data, i. e., many and huge traces. I therefore hope that our solutions might be adoptable for other fields of trace analysis as scaling down is usually easier than scaling up; (c) The need for finding novel solutions in this sub-domain is most pressing. First, as stated above traces are very large and numerous. Second, FIT is the last vehicle test before bringing the cars to market. Therefore, it is important to reliably detect all errors in a limited amount of time; (d) A large extent of FIT analysts' daily tasks deal directly with trace analysis. This provides a good platform for studying working practices in this domain as other groups of test engineers at the company often were simultaneously and strongly engaged in non-analysis tasks such as organizing and management duties; and (e) the expertise in trace analysis is highest with this group. Other engineers often contacted FIT analysts in terms of trace analysis related questions. During the three-year, close collaboration with FIT analysts, we[1] developed several visualization prototypes and systems, all of them in a strong participatory design manner and three of them presented in this chapter—VisTra, AutobahnVis and Cardiogram. While the systems introduced in the previous chapter were designed for separate use cases and we collaborated with different user groups, our systems for FIT analysts all focus on one specific group of tasks, problems and challenges. In exchange, the development processes of our FIT-systems were longer, more intensive and based on iteratively refining our understanding over a period of three years. We analyzed the field, built a prototype, evaluated it and learned something, analyzed the field again, built another prototype or adapted our prototype, and so on. Therefore, most of the specific insights we gained often did not appear during an explicit user study but rather developed and strengthened over time (cf. grounded theory [GS77]).

In addition to our close cooperation with FIT analysts, I introduce ProgSpy2010, a tool that we collaboratively redesigned with and for **flash analysts**. Flash analysts are responsible for analyzing and optimizing the process of uploading software onto the ECUs of the car. As the software is deployed via the in-car communication networks these engineers also use trace analysis techniques for diagnosis purposes. During a redesign and re-implementation of their text-based inhouse analysis tool ProgSpy, we used this opportunity to integrate a visualization approach similar to the one used in Cardiogram into the new tool version, ProgSpy2010. Our ProgSpy2010 project, again, can be seen as a focused design study where we built one single solution for a specific user and task group, similar to the design studies presented in the previous chapter.

As traces, the underlying data, are time-structured, our tools mostly rely on time-based visualization techniques. For a general background I therefore refer to Section 2.5. Besides, our solutions were strongly influenced by several specific InfoVis design studies and by work from the Model-based Software Development domain. In particular, our solutions were inspired by early work [MHJ91] on trace visualization, a work on multiple coordinated views [PVW08] for the analysis of multivariate simulation traces, and a system [HCvW07] for the visualization of software traces. Similar to these solutions we chose to focus on the temporal aspect of trace data. In addition, visualization approaches from temporal visualization in other domains influenced our work. The LifeLines project [PMR+96] is similar to our AutobahnVis approach in the vertical arrangement of temporal data to aid the detection of data correlations (Section 5.3). The dependency re-computation we present in Section 5.2 is similar to Balzer et al.'s approach [BSN07] and the state machine analysis technique we used for Cardiogram (cf. Section 5.4) is derived from model-based testing [UL07, GHP02] and is related to Bringmann's application of these methodologies to automotive software testing [BK08].

In this chapter, I first summarize design recommendations which we derived from our three year collaboration with FIT analysts (Section 5.1). Then, I present three visual analytics tools for this

---

[1] Parts of this chapter have recently be prepared for publication. Therefore, if not specified otherwise, any use of "we" in this chapter refers to Michael Sedlmair, Petra Isenberg, Dominikus Baur, Christian Pigorsch, Wolfgang Jacobi, Michael Mauerer and Andreas Butz.

user group, starting with VisTra the very first prototype developed during this thesis which helped us to develop our design recommendations (Section 5.2). After that, I introduce AutobahnVis and Cardiogram which were developed and iteratively refined over a period of two years and subsequently installed into engineers' daily working environments. After that, I present ProgSpy2010 a trace visualization that we developed for another target group, namely flash analysts (Section 5.5). I discuss how our recommendations influenced the tools' design and provide validation by presenting successful examples from domain engineers using our tools with real data under real circumstances (for AutobahnVis, Cardiogram and ProgSpy2010). The chapter ends with a summary and a discussion of limitations and future work.

5.1 Exploratory Studies with FIT Analysts

In this section, I discuss the results of our field analysis of automotive trace analysts. While Chapter 3 already gave a brief overview of work practices of analysis engineers, this section provides concrete and detailed findings from studies with FIT analysts and outlines the design recommendations which we derived. Both are important to understand the design decisions for our prototypes.

5.1.1 Methodology

Over a period of three years, we conducted various formal and informal ethnographic studies with more than 50 FIT analysts and collaborated with some of them in user-centered design approaches (see also Section 3.1). In addition to FIT analysts (end users), we also involved a group of eight FIT tool, i. e., Carmen's, developers (see also Section 3.3.3) in order (a) to get insights into the problem domain from a software developers perspective and (b) to learn about technical obstacles, challenges and about how they have previously been tackled. In the following, I summarize the methods we used, outline how the typical setup of such a study was and briefly talk about the experience we had with them.

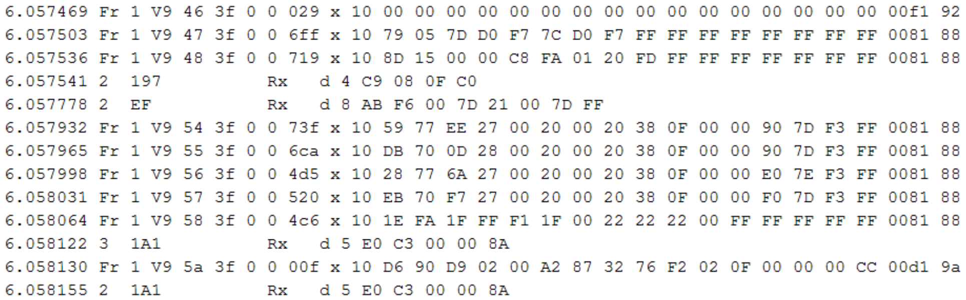
Interviews: Especially in the early phase, we used loosely-structured interviews to derive a better understanding of our complex target domain. These interviews helped us to get insights into technical aspects beyond that what you can find in books, provided us with an idea of user practices, and showed engineers' estimations on challenges and problems as well as their ideas and approaches to overcome them. Usually, an interview took an hour or less and was conducted in a meeting room at the location of the participant. Additionally, we asked participants to bring along their own laptops. In such cases, they could show us tools, problems or processes directly on their own equipment. Typical questions of interest were: “What are typical tasks?”, “What tools do you use”, “Where do you see the main challenges in working with traces”, or “How would you overcome these problems”. Most of the time, however, our loosely structured interviews were an open discussion between an InfoVis expert and a trace analysis expert, both with the goal to improve current working processes. We usually used note taking for logging the interviews. In some of the interviews, however, we additionally used audio-recording (for this interviews we had to get an IPR license). All in all, we conducted approximately 30 interviews with both, FIT analysts and FIT tool designers.

User observation and contextual inquiries: As interviews provide only information about what people say and not about how they actually do it (except from the interviews where participants brought along their laptop for task presentation purposes) we additionally conducted observational studies with end users. First, we started with purely observational studies with the goal not to interfere daily practices of the engineers. However, soon it became clear that due to complexity and high expertise in our domain it is not possible for a non-expert to derive meaningful results by just observing end users. Therefore, we desisted from purely observational studies and rather used contextual inquiries [HJ93]. We visited end users, observed their work and asked questions if something was unclear to us. All in all, we conducted contextual interviews with 14 FIT analysts in up to five sessions per participant. Each session took not more than one hour as we did not want to/could not interfere the engineers work for a longer time. We solely used note-taking for data logging purposes. In most studies, however, we had 2–3 observers/note takers in order to log the rich information.

Focus groups: Especially in the later phases, when we already had concrete findings, design mockups or prototypes, we discussed them in focus groups with 3–10 participants. The focus groups particularly helped us to discuss, evaluate and improve our findings of the exploratory studies (see below) and our various design ideas. Overall, we conducted approximately 20 focus groups with varying lineups of FIT analysts, FIT tool designers and also invited decision makers and engineers with slightly different backgrounds to these discussions. For logging our focus groups, we used both note taking and/or audio-recording (with IPR license). If we used note taking only, we tried to counterbalance this limitation by having at least two note takers.

Informal collaboration: In addition to the formal user studies, we engaged in frequent informal conversations with engineers who were more closely involved in our user-centered design process. For each of our FIT-design studies these were between three to six lead users. These conversations were, for instance, meetings at the company's cafeteria, having a joint lunch or engaging in casual discussions at the working place. To our experience, these informal conversations were highly valuable for understanding the various facets of our target domain, to iteratively refine our understanding and knowledge about requirements for visual analysis tools, and last but not least to establish fruitful relationships for a successful collaboration. A definite drawback of informal conversations, however, is the fact of restricted opportunities to log the information in order to meet scientific rigor. We tried to counteract by always having a notepad and a pencil along for note taking (seamless transition to interviews) or in case we could not take notes, e. g., at lunch, writing down the information immediately after the studies when retention was easier and richer.

In the following, I summarize the results from these studies.



```
6.057469 Fr 1 V9 46 3f 0 0 029 x 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00f1 92
6.057503 Fr 1 V9 47 3f 0 0 6ff x 10 79 05 7D D0 F7 7C D0 F7 FF FF FF FF FF FF FF FF 0081 88
6.057536 Fr 1 V9 48 3f 0 0 719 x 10 8D 15 00 00 C8 FA 01 20 FD FF FF FF FF FF FF FF 0081 88
6.057541 2  197          Rx   d 4 C9 08 0F C0
6.057778 2  EF           Rx   d 8 AB F6 00 7D 21 00 7D FF
6.057932 Fr 1 V9 54 3f 0 0 73f x 10 59 77 EE 27 00 20 00 20 38 0F 00 00 90 7D F3 FF 0081 88
6.057965 Fr 1 V9 55 3f 0 0 6ca x 10 DB 70 0D 28 00 20 00 20 38 0F 00 00 90 7D F3 FF 0081 88
6.057998 Fr 1 V9 56 3f 0 0 4d5 x 10 28 77 6A 27 00 20 00 20 38 0F 00 00 E0 7E F3 FF 0081 88
6.058031 Fr 1 V9 57 3f 0 0 520 x 10 EB 70 F7 27 00 20 00 20 38 0F 00 00 F0 7D F3 FF 0081 88
6.058064 Fr 1 V9 58 3f 0 0 4c6 x 10 1E FA 1F FF F1 1F 00 22 22 22 00 FF FF FF FF FF 0081 88
6.058122 3  1A1          Rx   d 5 E0 C3 00 00 8A
6.058130 Fr 1 V9 5a 3f 0 0 00f x 10 D6 90 D9 02 00 A2 87 32 76 F2 02 0F 00 00 00 CC 00d1 9a
6.058155 2  1A1          Rx   d 5 E0 C3 00 00 8A
```

Figure 5.1: Extract from an .asc trace file. Each row shows the time stamp and raw values (hexadecimal) of recorded messages. Usual trace files contain 100,000 to 50 million messages.

5.1.2 Background: Data, Tasks and Tools

As stated in the introduction of this chapter, our main target user group consists of analysis experts involved in final vehicle testing phases, namely FIT analysts. FIT stands for the German term “Fahrzeug Intensiv Test” which can be translated as “intensive vehicle tests”. FIT is used to describe a specific analysis process at the very end of a vehicle’s development process (cf. Section 3.3). In the following, I describe data, tasks and tools used during FIT analysts and give background information to better understand the entire process.

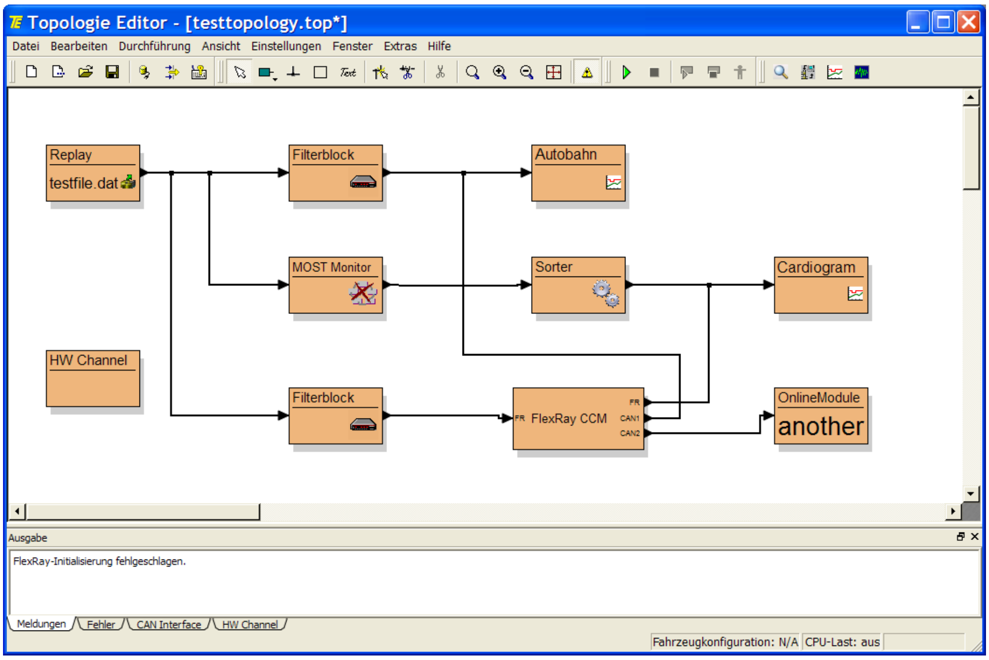
Typically, 1.5 years before start of production (SOP) for each novel car series, a test fleet of 200 vehicles is produced. These cars are then driven by employees, for example on weekends, to produce test data in ordinary situations. Additionally, specific test drivers simulate extreme situations and attempt to produce erroneous behavior. As the software of the vehicle’s ECUs is a black box system for car manufacturers—the implementation is done by suppliers and is kept secret due to protectionism—test engineers rely on the information they can see and measure, namely the communication on bus systems. For this purpose, all electronic communication is recorded during the test runs. This information together with a verbal, written and/or automatic error report is transferred to FIT analysts, our target group. Their job then is to analyze the traces, to detect errors and their sources, and in doing so to verify a vehicle series’ safety, security, and functionality. Finding critical errors in this phase is extremely important to ensure customer safety, but also to avoid additional costs or image damages by potentially calling back vehicles from the field. To record the data, specific hardware (for an example please refer to Figure 3.4 on page 31) is installed in the test cars that produces test files called traces each with its own proprietary data formats that analysis tools need to be able to read. Traces contain temporally ordered lists of all messages sent by one, several or all bus systems. Each message item is typically composed of two numeric values: (1) time at which the message was either sent or received, (2) the raw message data. Figure 5.1 shows a small excerpt from such a trace.

In addition, traces are usually enriched with error logs. There are two main types of error logs: automatic and manual. Automatic error logs are created by the electronics in the car (DTC - data trouble code) from a predefined error condition. These logs are very common and either are directly embedded with the trace or stored in a separate error file. Manual logs on the other hand are defined by the driver. They can either be recorded by pressing a specific trigger hardware installation and annotating the reason afterwards to the trace, or the transfer can be more informal using emails, written notes, or just through a verbal report to an analysis engineer. To reduce the amount of data recorded, engineers use “journals” as specific, pre-filtered formats that reduce the trace data to a few specific message types such as error frames or fault memory entries, and manually added information such as markers, triggers or predefined events.

If an error occurs and a log file is produced, it is the task of the FIT analyst to locate the error source (usually in form of a reliable ECU) and to initiate further steps to solve the problem. Such a step could for example involve contacting an ECU development engineer with an adequate description of the error source such as a screenshot of the analysis tool showing the error. During our studies, we found FIT analysts using several special-purpose analysis tools to analyze the data. The most frequently used were two in-house tools called Carmen and Ediabas as well as Canalyzer [Vec] and more occasionally Tracerunner [Sof] (see also Section 3.3.3 and 3.4.2). The most important feature of all these tools is the interpretation of the raw data to human readable form. For this purpose, the tools either use their connection to a vehicle specification database or import a network specification file and extract sending ECU, transporting bus, message name, as well as signal names and values from the raw data. Our participants found both Carmen and Canalyzer most relevant and powerful due to their scalability and compatibility to various data formats and the availability of special-purpose plugins and data interpreters. Both tools are based on the interactive combination of different digital modules that allows engineers to individually configure a tailored measuring setup. Figure 5.2 shows a screenshot of such a test setup in Carmen where several modules are visually linked together (among them two we integrated). Available modules include those for data loading, interpretation, filtering, or visual representation. Typical representation modules can show dynamically interpreted lists of messages, temporal signal plots, or rudimentary overview techniques. To analyze traces, our participants often used one or several of these tools in combination with general-purpose tools. Especially common text editors were frequently used to explore interpreted traces but also for directly investigating raw data (reading hexadecimal code!).

5.1.3 Data Analysis: Practices, Problems and Challenges

In our studies, we found that engineers typically began their error analysis with a first hypothesis about the error source. Using their analysis tools they then attempted to (a) verify this hypothesis, (b) iteratively refine the hypothesis, or (c) dismiss the hypothesis and start anew. Based on their initial hypothesis, the analysts took different approaches to finding an error. If a clear hypothesis about the error source existed, engineers commonly started to check interpreted or even raw values directly within a trace. If the hypothesis was not solid from the beginning, the error description was rather vague or if the error source was estimated to be more complex, our participants preferably started with an overview using journals and then iteratively filtered and analyzed interpreted message lists and signal plots.



FlexRay-Initialisierung fehlgeschlagen.

Fahrzeugkonfiguration: N/A

CPU-Last: aus

Figure 5.2: Example of an analysis setup with several modules in Carmen. The basic paradigm is representing modules as boxes and connect them via orthogonal arrows. The left-most box is a reply module where a raw data trace can be loaded and subsequently replayed. The arrows indicate the flow of the messages to other modules such as: A *Filterblock* for filering bus systems, ECUs, messages and/or signals, a *Counter Module* representing the number of messages per second passed through this module, or the *Online module* for interpreting and representing messages on-the-fly in a list-based interface. The modules AutobahnVis and Cardiogram were written by us (see below).

Our studies showed that engineers had to track errors of varying degrees of complexity which tremendously influenced the process of finding these errors in terms of processing time, costs, and engineers involved. Simple errors outnumbered complex errors but complex errors could take weeks or even months to find and solving them was often a highly collaborative undertaking. One engineer commented: *"I can track down a simple error in several minutes, but solving a complex problem can take weeks or even months"*. In three two-hour long focus groups we discussed error complexities with 3–6 engineers and identified three main origins of complexity:

Reproducibility: If an error appears only sporadically, cause-effect relationships are hard to define. Reasons for low reproducibility include external circumstances such as temperature, imprecise verbal error descriptions, special test cases, extreme driving situations, or incorrectly specified error conditions. For example: After ending a test run, our engineers detected that all car windows would open unexpectedly. Engineers spent several weeks analyzing and trying to reproduce this error. The actual reason was a specific test case in which all four doors were simultaneously slammed shut which activated an overpressure sensor that opened the windows.

Dispersion: Many highly distributed and inter-related hard- and software systems exist in a vehicle and errors often propagate over several ECUs before they are automatically detected and logged. Additionally, the interplay between two or more intrinsically and separately correct subsystems can lead to complex, unpredictable errors. The more dispersed errors or involved systems are, the higher the chance that they might be complex. This is also true for the above example where the actual reason was not located in the window system but in the accident security system.

Degree of trace preparation: Not every bus system or recording hardware supports journals (Section 3.2) to reduce and abstract the recorded data. Without any abstraction it can become complex and laborious to analyze traces especially when exact error timings are not available, for example for manually recorded errors. Additionally, engineers have to be aware of the fact that measurement hardware can be the source of errors and inaccuracies in the data. As a result of these complexities, our engineers relied on an array of different tools. One engineer used 14 different tools in an hour-long analysis session. He liked all of these tools but was burdened by the additional work of switching between them: *"Analyzing alone is a complex task, but handling the entire overhead of using so many incoherent tools is overkill [...] each tool is a valuable part of my work but they are not well coordinated and integrated, this means a lot of additional, redundant work to me [...]"*.

5.1.4 Design Recommendations

Based on our understanding of the domain challenges, we derived several design recommendations for building InfoVis/VA tools in this area. These recommendations have been developed over a period of three years by (a) iteratively conducting studies which results I described above, (b) by our increasing understanding of both building prototypes in this area (for instance, VisTra that I present in this thesis, cf. Section 5.2) and InfoVis/VA in general, and (c) by our general studies presented in Chapter 3. For completeness purposes, we also considered the general implications suggested in Section 3.6.2 and fine-tuned or echoed some of them respectively. We used this set of recommendations as a basis for the later prototypes presented in Section 5.3, 5.4 and in parts also for 5.5. They turned out to be very helpful for this undertaking and we hope that other tool designers might also find guidance in these aspects.

New Perspectives on Complex Errors The detection of complex errors (see Section 5.1.3) requires dedicated data representations to show correlations between error sources.

R-1: Visual Overview Techniques
Most current overviews are restricted to the representation of journal data. To provide overviews of traces without journals, to broaden the understanding of global aspects in general, and to support handling of complete traces, novel overview techniques based on time, messages, and signal values are necessary (see also implication 2 in Section 3.6.2).

R-2: Perspectives Beyond Raw Data and Signal Plots
In our study, most representation techniques were based on lists and simple signal value plots. Beyond what these tools can provide, engineers need to analyze timing aspects, detect outliers, see correlation between messages and between mechanical behavior and electronic data, as well as message propagation.

R-3: Equal Representation of Time and Logic
This recommendation is an important sub-component of R-2. For engineers it was extremely important to see correlations between the temporal (when has a message been sent) and the logical layer (who sent it, who received it, what software components were involved, etc.). Current techniques could not support this requirement.

R-4: Multiple, Modular, and Coordinated Solutions
Engineers require multiple different perspectives on the data to detect complex errors. Which perspectives were most relevant in our study relied on an engineer's knowledge, preferences, as well as the underlying problem. Therefore, an unrestricted and modular combination of perspectives is useful. Perspectives should support coordination over time and data linking according to known techniques (e. g., as in [NS00a]), but also the opportunity to work without coordination (e. g., for comparing behavior at different time stamps).

R-5: Fast Access to Raw Data
Engineers were used to working with hexadecimals and regularly had to check single bytes and bits during their work. For them, raw data must always be ready at hand in order to immediately prove or discard hypotheses based on raw values. "Fast access to raw data" was one of our most requested requirements (see also Section 5.2.3).

Handling the Masses of Data Handling the masses of data produced in automobile testing is a huge practical challenge. While data storage is no longer a pressing problem, analyzing all data in detail is nearly impossible for engineers. Designing data abstraction techniques, automated and semi-automated error detection techniques, and supporting fast and interactive data reduction is necessary to support engineers in recognizing and handling all relevant information.

R-6: Data Abstraction and Automated Filtering
Understanding comprehensive aspects in traces is essential for complex error detection but difficult to retrieve directly from raw data. Novel data abstraction techniques are required for reproducing behavioral aspects, comprehensive correlations and functional dependencies in the data. Additionally, automated filtering of data is desirable as often only a very small subset of the data is involved in the error finding process. The reduced data can then be used as input for novel representation techniques (see R-1and R-2).

R-7: Support for Automated Error Detection
Error analysis depends in large parts on an engineer's expertise as common error sources are often checked manually by one specific person. Current analysis procedures rely on sample testing and many recorded traces are never analyzed. Trying to automate this process would help (a) to

rapidly test a set of hypotheses, (b) to speed up the detection of common errors, and (c) to allow analyzing much more data than is achievable via manual inspection.

R-8: Avoid Repetitive Work and Unnecessary Iterations
Due to the size and complexity of recorded trace data, engineers used a large array of tools that each only supported parts of the analysis. Unnecessary time was spent converting data manually and importing/exporting formats. Missing features are often tedious and annoying and hinder the acceptance of novel solutions. Engineers frequently demanded a powerful basic tool for handling all the configuration tasks such as data loading and interpretation plus a collection of embedded modules for specific problems in order to avoid repeating analysis steps (see also Section 5.2.3).

Engineer-centered Solutions From our research on VisTra (see next section) we learned that a novel solution's value is directly correlated to its interplay with current technologies, and its close integration into the current engineering work flow. Additionally, from our general field analysis we learned about the importance of familiarity and collaboration in our domain (cf. Section 3.6.2).

R-9: Embed Solutions in Current Work Environments
Tools currently used by engineers can be very powerful in terms of flexibility, compatibility, scalability, and in providing specific features for specific problems. Re-implementing all of these features for a new visual analytics system is usually not possible within realistic time and budget requirements (and also often not wanted by the researcher). Instead, closely integrated solutions can be immediately used by engineers in the context of their familiar work environments, take advantage of already supported data formats, be combined with conventional solutions without any extra costs, and better extend engineers' current work processes. This may have additional benefits in terms of adoption (see also Section 5.2.3).

R-10: Take Familiarity into Account
Echo from implication 1 in Section 3.6.2—Additionally, our design studies in Chapter 4 already showed the utility of this guideline.

R-11: Support Collaboration
Echo from implication 6 in Section 3.6.2—Our explorative studies with FIT analysts showed that supporting collaboration is particularly important for complex error analysis where many test (and development) engineers are involved in detecting the source of an error. Tools in this application area should therefore actively take this into account.

In the following sections one the hand I describe how we used these recommendations but on the other hand also explain how earlier prototypes were involved in revealing and refining these recommendations.

5.2 *VisTra:* Using Transitive Chains for Analysis

In this section, I present the very first visualization tool that was developed in the scope of this work[2]. VisTra utilizes knowledge from functional network specifications to re-compute dependencies in traces, use this information to pre-filter them, and subsequently provides a visual interface for exploring transitive chains (cf. Section 4.2.1) in the prepared data. Naturally, at the point we[3] built the system we did not know about most of the aspects which appeared during the process of building this and other prototypes and by doing more intensive field analysis studies. Retrospectively, however, we already applied some of the recommendations we formulated later on intuitively and others solidified during evaluating and discussing VisTra with test engineers. Furthermore, it is also clear that from a today's perspective we would do several design decisions differently (e. g., color coding). Nevertheless, our user studies revealed good feedback, underlined the potentials of the approach we suggested, but also helped us much in forming the recommendations stated above. I therefore decided that it is definitely worth to introduce and discuss our ideas and experience and to share them with other tool designers and researchers.

5.2.1 Approach

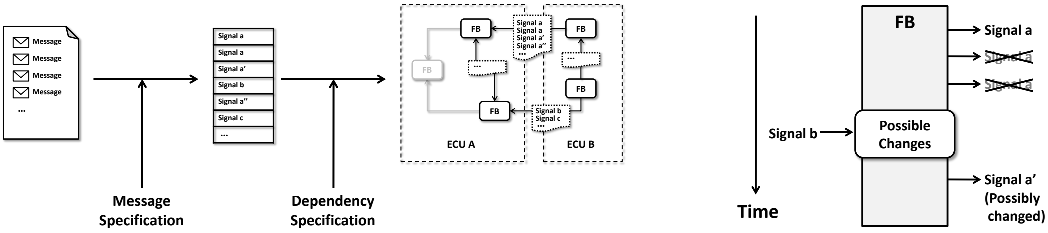
As stated above, the largest challenges of trace analysis today are to deal the masses of messages in a trace and to track down complex errors in a highly distributed system. Oftentimes, the location where an error is detected is not necessarily the location where the error really occurred. Furthermore, it can be important to see how information spreads from an error source in order to understand dependencies between different errors. However, current analysis techniques are purely message-oriented and provide little insight into correlations and causality within the network. The reason is that traces are merely time-based, list message for message, but do not contain explicit information about cross-correlations in the network (messages contain either information about receiver or sender, not both).

To fill this gap, our basic idea with VisTra was to re-compute dependencies in traces by using dependency specifications. In doing so, we want to provide test engineers with an opportunity to analyze traces from a more functional and dependency oriented perspective and to extend current working practices of going through traces merely message by message. Our rationales for that were twofold. First, we wanted to better support tracking down errors in highly distributed systems and second, we tried to address the challenge of loosely, non-automatically and verbally described errors where a functional-oriented exploration technique might be more suitable than a message-based one. VisTra's approach for data preparation and exploration is organized in three major steps:

---

[2] For a detailed description of VisTra's development process please see [Sed07].

[3] Portions of this section have been published in [SHS+08] and in [Sed08]. Thus, any use of "we" in this section refers to Michael Sedlmair, Wolfgang Hintermaier, Konrad Stocker, Thorsten Büring, and Andreas Butz.



(a) (b)

Figure 5.3: Schematic representation of the data preparation process in VisTra: **(a)** Interpretation of the trace and re-computation of functional dependencies both based on specification data. The result is a hierarchical clustered, directed functional graph. Some specified edges (signal communication) and/or nodes (FBs) might not appear in a real communication record (gray marked); and **(b)** filtering redundant signal communication.

1. Re-computing dependencies by combining traces with network specification data (R-6): As traces currently do not provided any information about dependencies or communication paths, VisTra initially transforms the raw values into a data format that supports this information (cf. Data Transformation step in Section 2.1). To re-compute functional correlations, we use specification models from earlier development stages (see Balzer et al. [BSN07] for a similar approach). For this purpose, we first interpret trace messages in terms of signals and transporting bus system as it is commonly done by analysis tools. Then however, we additionally map the interpreted traces to the according network specification[4]. The result is a clustered graph with ECUs as clusters, FBs as nodes and directed edges representing a set of temporarily ordered signals exchanged between pairs of functional blocks. Depending on the traces' size these edges can include thousands or even tens of thousands exchanged signals. However, on the other hand there might also be specified edges between two functional blocks that did not exchange any signals during a particular trace recording at all. The same applies for FBs that potentially are inactive during a certain driving/recording situation. We therefore distinguish between the real communication graph (what we compute) and the specified functional/physical network (what is specified). Figure 5.3-a illustrates the process of re-computing dependencies.

2. Pre-filtering traces by excluding redundant communication paths(R-6):
80% of the communication in current in-car networks is cyclic. This means that a majority of signals are sent in certain time intervals regardless if there has been a change in the signal's value or not. For instance, the window shifting button sends every 10ms a signal indicating whether the button has been pressed (up/down) or not. This design implies that much information is sent redundantly. Our approach harness this fact, analyzes redundancies in signal communication and automatically pre-filters the trace accordingly. Based on engineers' feedback we learned that it is additionally necessary to retain signals that another signal might have influenced, i. e., signals sent directly after another signal has been received at the FB in question. We therefore use both,

---

[4] We use the physical specification that also includes parts of the functional specification to derive a functional FB graph. In doing so, the data can be exported from the BNE (see Section 3.3.3).

redundancy information as well as other in-coming signals, to pre-filter the data. Figure 5.3-b illustrates our filtering approach in a graphical way.

3. Exploring transitive chains around functional blocks (R-2):
The first two steps transform a time-based trace into a clustered graph with ECUs and FBs, and with filtered, temporally ordered, directed signal edges. The dependencies of a specific functional block then can be described by a specific sub-graph, more precisely by the transitive chain around this particular FB as we defined it in Section 4.2.1[5]. The transitive chain provides all reachable predecessors as well as all reachable successors and might be valuable for better understanding real error sources and propagation based on FB dependencies. The visualization we designed provide a platform for browsing FBs along with their transitive chains in order to gain insight into elements which actually influenced and/or have been influenced by FBs (and not which potentially could be influenced as defined in specifications). The design of our visualization is described in more depth in the next section.

5.2.2 Visualization

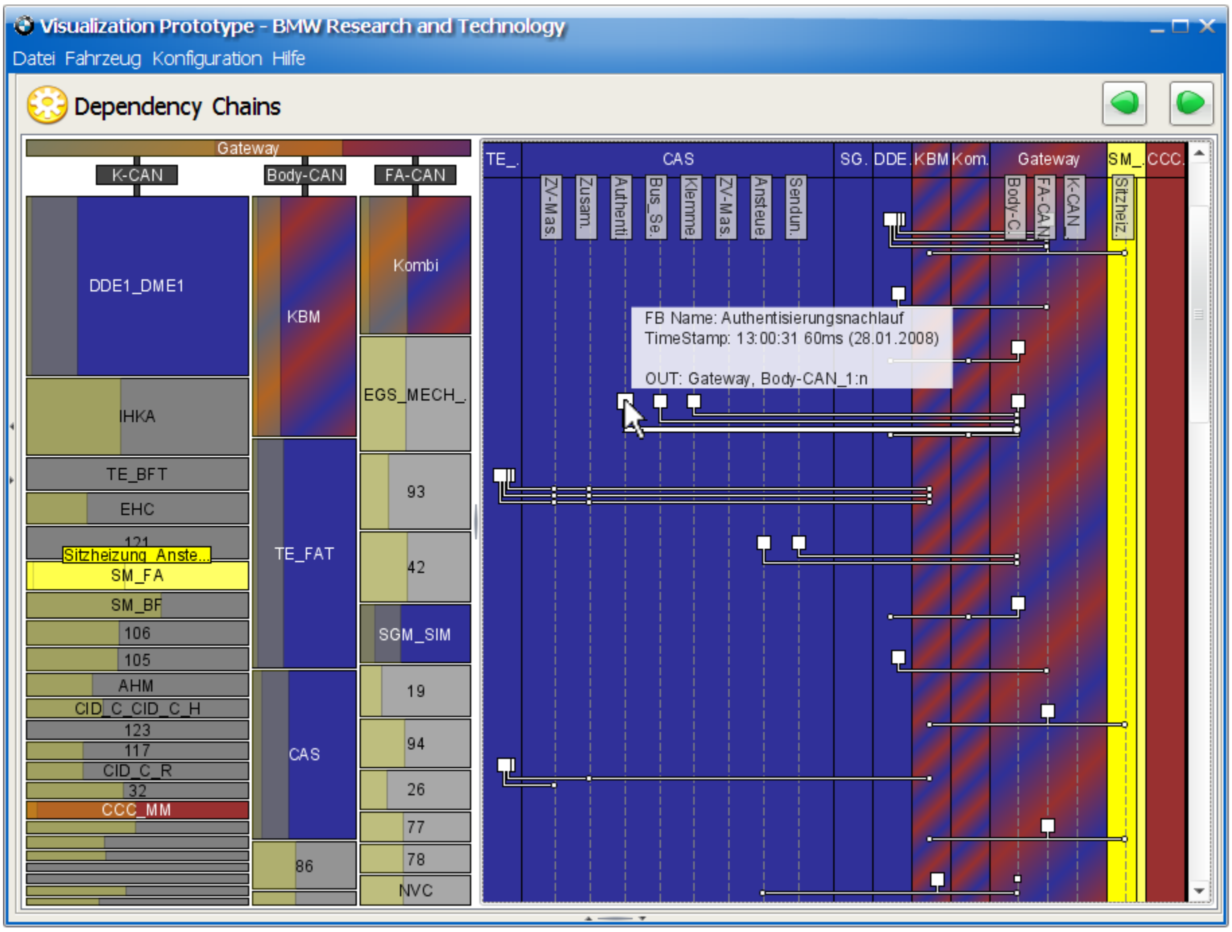
For designing VisTra's visual interface we closely collaborated with three domain experts, all researchers working on novel diagnosis methods. Our collaborators stated that it is invaluable to clearly present timing information along with transitive chains (R-3). For representation purposes, two of them suggested using message sequence charts as they are already very common in early development phases (R-10) and as they provide a good opportunity to show communication correlations over time. We therefore abstained from our initial idea of using a node-link diagram to represent the dependencies and rather explored how we could trim message sequence charts to our needs in terms of additional information and scalability.

Based on iterative refinement with our three domain collaborators, we came up with a vertically separated dual-view visualization (see Figure 5.4) where the left view was designed to provide an overview over the physical network with ECUs and bus systems (*ECU-Map View*) and the right view sequentially showed transitive chains in the requested message sequence chart-like representation (*Sequence Chart View*). To explore a specific transitive chain the user can select a FB, we call it the trigger-FB, and in doing so initiates a color-coded presentation in the ECU-Map View, and an update of the Sequence Chart View with the transitive chain around the selected trigger-FB. The design of the two views can be described as follows:

Views The *ECU-Map View* is a mixture of a common topology map (cf. Section 3.4.2, R-10) and a treemap (cf. Section 2.3, R-2) initialized with the two-level hierarchy of bus system and ECUs (segmentation of columns and lines) and the number of FBs to assign the aspect ratios. According to common topology maps the central gateway has a specific position and is located at the very top of the ECU-Map and ECUs of the same bus systems are grouped in

---

[5] We derived the name VisTra from this concept: VISualizing TRAnsitive chains.



FB Name: Authentisierungsnachlauf
TimeStamp: 13:00:31 60ms (28.01.2008)

OUT: Gateway, Body-CAN_1:n

Figure 5.4: Screenshot of VisTra showing the ECU-Map View on the left and the Sequence Chart View on the right. Within the Sequence Chart View, three ECUs are zoomed in, one connection is highlighted via mouse-over, and additional detail is shown in a semitransparent tooltip.

columns. The size of each ECU rectangle codes the number of FBs contained and therefore provides an implicit engineers' metric for ECUs' importance. In order not to abandon our size coding approach, ECUs with multiple bus connections are not presented redundantly but are connected to the most corresponded bus system, i. e., over which they sent the most signals. Additionally each ECU rectangle has a "status bar" to indicate the number of active FBs within an ECU, i. e., FBs that actively communicated in the trace that is shown. The status bar which is represented as a semitransparent greenish overlay on the gray ECU rectangles is fully included in the treemap approach so that the area of active FBs directly correlates with the entire system. The user can click on an ECU rectangle which in turn displays an alphabetical list of the included active FBs (and also—grayed out—inactive FBs for context retaining reasons). After a trigger-FB has been selected, both views are updated. The ECU-Map thereafter highlights all ECUs involved in the triggered transitive chain by changing their color and doing so allows for quickly detecting all physical components involved reachable from the trigger-FB. The selected ECU turns into a light yellow and gets an additional header in a more saturated yellow to label the trigger-FB. Predecessor ECUs turn blue, successor ECUs red, and ECUs being both are drawn in blue and red stripes (again, we took blue and red based on domain conventions). After a trigger-FB has been selected, the *Sequence Chart View* updates in order to show the transitive chain around the selected FB. To do so, the ECUs involved in the current chain are arranged in horizontally rectangles colored equally to the ECU-Map in yellow/blue/red. Within these ECU rectangles, vertical dashed lines represent the FBs involved within the respective ECU. Given a time line from top to bottom, the (filtered) signal communication between FBs is presented by horizontal, temporally-ordered “communication lines” between the respective FBs. The resulting visualization is a color-coded sequence chart with a temporal representation of the triggered transitive chain ordered by causality (Figure 5.4, right view).

Interaction The most important user activity in VisTra is exploring FBs’ transitive chains. For this purpose, the user can either click on ECU rectangles in the ECU-Map, or can directly select elements in the Sequence Chart View (FB-labels, -rectangles, -dots) in order to navigate to interrelated chains. We integrated a browser-like history, which allows the user to easily navigate back and forward in the transitive chains s/he explored so far. In order to provide a better orientation in highly complex transitive chains, we highlight the communication connections by enlarging it when the user hovers over it. Additionally, hovering over elements reveals detailed information about sender and receiver FBs/ECUs, exact timing information and exchanged signals in a semi-transparent text box. Besides, we added a semantic focus and context zoom to the Sequence Chart View. This solved the problem that some transitive chains are extremely complex and too large for showing them properly on the available horizontal space. If a triggered transitive chain exceeds a certain size (we initially defined 25 FBs, but this can interactively be adopted by the user), all ECU rectangles in the sequence chart except the ECU, which contains the trigger-FB, are semantically downscaled and the incorporated FBs are hidden. To explore these FBs, the user just clicks on an ECU to expand the downscaled ECU to the entire functional view. Clicking again collapses the ECU respectively (cf. Figure 5.4).

5.2.3 Evaluation

In the following, I provide qualitative feedback that we gathered during think-aloud studies with eight domain experts, five focus group workshops with 3–8 engineers and from various informal discussions with prospect end users. In the one-hour long think-aloud studies we encouraged our participants to bring along their own traces to analyze current problems of their daily work. Five of them used this opportunity, for the remaining three we used an example trace prepared by us and let them conduct a set of predefined tasks. By conducting all these studies, we wanted to get experts’ estimations on our approach in terms of domain utility and potentials, but also evaluate understandability, usability and current restrictions of both visualization and data preparation technique. Especially, understanding the limitations helped us very much in building later tools for trace analysis.

Visualization, Features and Usability The overall feedback to the visualization was positive. Our general approach, the way we encoded the information and the interaction concept was judged to be easy understandable and usable. While in general the dual-view concept was also judged positively, by observing our participants using the tool, however, revealed that the ECU-Map was rarely used compared to the Sequence Chart View. With the sequence chart, especially the path highlighting via hovering and the semantic zoom was liked by our participants such as one engineer cited: *‘‘oh good, this [path highlighting] is practical, by using it I do not have to follow the lines with my fingertip”*, and another: *‘‘opening and closing of the ECUs is enormously helpful and understandable, especially when I have two ECUs in mind that I want to explore in more depth”*. A frequent point of critique was the color coding in terms of using saturated colors for large areas. While this originally was an artifact of the participatory design process we conducted, we definitely agree with our participants (and also with literature, e. g., Ware [War08]). Furthermore, two of the automotive engineers mentioned that representing DTCs (errors) with VisTra would benefit in order to directly correlate dependencies with errors that have been occurred. Another missing feature frequently mentioned by engineers was filtering transitive chains in terms of time and shown levels of dependencies.

Data Preparation, Potentials and Limitations In addition to this, we intensively discussed our participant’s estimations about the prospected utility but also the limitations of our approach and in doing so gathered feedback and suggestions about what to address with future tools.

Regarding the potentials of VisTra, all participants named the opportunity to better understand and faster track down complex errors by providing a novel perspective on the data (R-2). Furthermore, two trace analysts stated that especially *“bundling the signals”* is helpful as redundant information is automatically filtered in order to provide a more comprehensive view on dependencies in the network (R-1, R-6). Another two engineers underlined the importance of showing both time and logic in parallel and evaluated VisTra as a *“first good step into this direction”* (R-3). While we generally got good feedback regarding our visual interface and data preparation, we, however, also learned about various technical limitations our first approach had and why these restrictions impeded a practical application in daily practices. These limitations strongly influenced the recommendations we introduced above and could be clustered in three categories, additional time costs/repetitive work (R-8, R-9), additional views/raw data (R-4, R-5, R-9) and personal preferences (cf. diversity implication in Section 3.6.2).

Additional time costs: This was the most important limitation mentioned by nearly all participants. VisTra is a standalone tool that uses exported trace files with a specific format (.asc), can handle traces with a maximum size of approximately 100MB, and additionally requires BNE exports for pre-computing dependencies. This hinders engineers using our tool in many ways. First, if a trace in question is not delivered in the correct data format it must be transformed beforehand. Second, if traces are too large (i. e., >100MB, what they often are) they have to be split and explored separately in the tool. Third, finding and exporting the right network configuration takes time and can be a potential source of mistakes. Last, as VisTra is a standalone that is not well-integrated with the engineers' current tool chains, switching between it and other analysis tools requires to repeatedly importing data, restoring context and manually correlating the information presented in the different tools. The basic problem with all these supplementary tasks is that they took additional time that the engineers, however, cannot afford. The situation becomes even more problematic as it is very often necessary to analyze many different traces in order to analyze a specific error (e. g., from different data loggers, from different parts of longer test runs, from different test cars showing the same behavior, or to refer to other traces in order to consult references). Manual and repeated preparation of all required traces is impractical (R-8). For using VisTra or other novel trace analysis tools in every day work, therefore all our participants argued for closely integrating them with current software environments. One engineer formulated it this way: *"The most important thing for us [FIT analysts] is to get a tool that supports our work. For this purpose it is crucial to integrate it in Carmen [the analysis environment most frequently used by his group] otherwise we cannot benefit from any new tool [...] there have been so many good ideas, but all of them just had been nice examples and we never could use one single of these tools productively"* (R-9).

Additional views/raw data: More than 90% of our participants furthermore noted that VisTra as it stands cannot be used for daily work because it lacks in providing additional perspectives on the data. Cited by nearly all engineers was a textual representation of either the interpreted trace or the raw data (R-5), as one engineer stated: *"This list is the trace! We need it all the time for quickly referencing back"*, or another one: *"every visualization is a potential source of errors, therefore I always need the trace [in textual form] to check my hypotheses with the real data"*. During our studies we also found that participants demanded other special views for specific needs (R-4). Two engineers said they would need a view for additionally showing dtc files, another three argued that a signal plot would be helpful, and one said he would need an additional view for showing vehicle status. While it might be too time- and cost-intensive to re-implement all these views, a complete and close integration of a tool in prevalent software environments would implicitly provide these additional views (R-9).

Personal preferences: In line with the demands for highly task- and user-specific views, our participants mentioned a variety of other requirements which they stated to be crucial for them, such as constantly highlighting cyclic messages, or showing the sequence chart horizontally rather than vertically. These aspects, however, rather indicate personal preferences and specific needs of a single participant or smaller sub-groups than providing generalizable implications. Asking other participants often revealed that for them a particular aspect or an additional view (except the view for the raw data!) would not be necessary, should be waived, or even solved differently (cf. diversity implication in Section 3.6.2).

5.2.4 Discussion

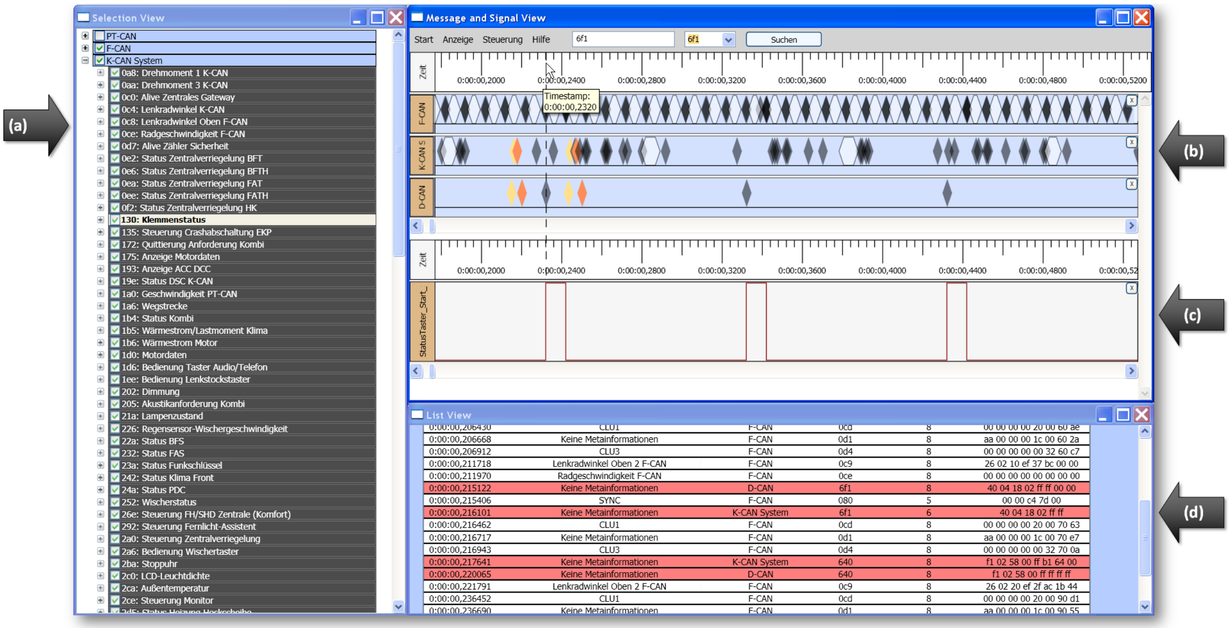
VisTra, the prototype presented in this section, is based on pre-computation and -filtering of traces utilizing dependency specifications from earlier development phases. Compared to traditional message based analysis techniques, VisTra allows the user to explore traces starting from functional blocks and sequentially examine transitive chains in the physical (ECU-Map) and the functional network (Sequence Chart). We conducted a variety of qualitative user studies with trace analysis experts including think-aloud studies, focus groups and informal discussions. While the basic approach was quite liked, the studies revealed several obstacles that would hinder VisTra's usage in daily practices. These findings helped us in better understanding domain requirements and especially in formulating the recommendations R-4, R-5, R-8 and R-9 (cf. Section 5.1.4). First, we learned that repetitive work and unnecessary iterations is a major reason for not using a solution (R-8). For practical application, novel tools need to be closely integrated with prevalent analysis environments (R-9). We also found that it is invaluable for nearly all engineers to have fast access to a textual representation of the trace in question (R-5) and also that other traditional perspectives on the data would be beneficial (R-4, R-9). Finally, we encountered a variety of specific needs differing between sub-groups and even single engineers (cf. diversity implication in Section 3.6.2). Usually however, it is neither possible nor advisable to address all these personal, even conflicting requirements at once in a novel tool design. Based on our experience, we think and recommend to start designing novel tools by focusing on a specific target group with similar needs and requirements, and to engage them in a participatory design approach. While we strongly that it is valuable to have a broad understanding of different fields in mind when designing the tool (i. e., doing pre-design studies with various user groups), during designing phase it might be easier and more practicable to have clear requirements from a specific user group. If this sub-group has carefully be chosen in terms of representativeness and a design has validated for them, in a next step the tool could be tested with other sub-groups and adapted to their specific needs.

5.3 *AutobahnVis*: Visualizing Messages

Based on the lessons learned from VisTra and from our explorative field studies we implemented two visual analytics modules that we closely connected to Carmen, the most frequently used trace analysis tool at the company we worked at. In this section, I will describe AutobahnVis, the first of these tools, Cardiogram, the second one, is described in the following section. AutobahnVis is a tool that we designed to extend the current state-of-the-art technology for raw data analysis. While VisTra tried to follow an entirely new approach for trace exploration based on dependency re-computation, with AutobahnVis we concentrated on traditional message-oriented browsing and tried to contribute to overcome shortcomings of the popular text-based representations and to enrich them by providing new perspectives of timing aspects within traces' raw data (R-2).

5.3.1 Visualization

The heart of AutobahnVis is a network visualization which represents raw messages according to two main dimensions: time on the x-axis and the transporting bus system or sending ECU on the y-axis (R-3). A scatterplot-like layout is thus created with horizontal "lanes" enclosing all



☐ PT-CAN
☑ F-CAN
☑ K-CAN System
  ☑ 0a8: Drehmoment 1 K-CAN
  ☑ 0aa: Drehmoment 3 K-CAN
  ☑ 0c0: Alive Zentrales Gateway
  ☑ 0c4: Lenkradwinkel K-CAN
  ☑ 0c8: Lenkradwinkel Oben F-CAN
  ☑ 0ce: Radgeschwindigkeit F-CAN
  ☑ 0d7: Alive Zähler Sicherheit
  ☑ 0e2: Status Zentralverriegelung BFT
  ☑ 0e6: Status Zentralverriegelung BFTH
  ☑ 0ea: Status Zentralverriegelung FAT
  ☑ 0ee: Status Zentralverriegelung FATH
  ☑ 0f2: Status Zentralverriegelung HK
  ☑ **130: Klemmenstatus**
  ☑ 135: Steuerung Crashabschaltung EKP
  ☑ 172: Quittierung Anforderung Kombi
  ☑ 175: Anzeige Motordaten
  ☑ 193: Anzeige ACC DCC
  ☑ 19e: Status DSC K-CAN
  ☑ 1a0: Geschwindigkeit PT-CAN
  ☑ 1a6: Wegstrecke
  ☑ 1b4: Status Kombi
  ☑ 1b5: Wärmestrom/Lastmoment Klima
  ☑ 1b6: Wärmestrom Motor
  ☑ 1d0: Motordaten
  ☑ 1d6: Bedienung Taster Audio/Telefon
  ☑ 1ee: Bedienung Lenkstockstaster
  ☑ 202: Dimmung
  ☑ 205: Akustikanforderung Kombi
  ☑ 21a: Lampenzustand
  ☑ 226: Regensensor-Wischergeschwindigkeit
  ☑ 22a: Status BFS
  ☑ 232: Status FAS
  ☑ 23a: Status Funkschlüssel
  ☑ 242: Status Klima Front
  ☑ 24a: Status PDC
  ☑ 252: Wischerstatus
  ☑ 26e: Steuerung FH/SHD Zentrale (Komfort)
  ☑ 292: Steuerung Fernlicht-Assistent
  ☑ 2a0: Steuerung Zentralverriegelung
  ☑ 2a6: Bedienung Wischertaster
  ☑ 2ba: Stoppuhr
  ☑ 2c0: LCD-Leuchtdichte
  ☑ 2ca: Außentemperatur
  ☑ 2ce: Steuerung Monitor
  ☑ 2d5: Status Heizung Heckscheibe

0:00:00,206430	CLU1	F-CAN	0cd	8	00 00 00 00 20 00 60 ae
0:00:00,206668	Keine Metainformationen	F-CAN	0d1	8	aa 00 00 00 1c 00 60 2a
0:00:00,206912	CLU3	F-CAN	0d4	8	00 00 00 00 00 32 60 c7
0:00:00,211718	Lenkradwinkel Oben 2 F-CAN	F-CAN	0c9	8	26 02 10 ef 37 bc 00 00
0:00:00,211970	Radgeschwindigkeit F-CAN	F-CAN	0ce	8	00 00 00 00 00 00 00 00
0:00:00,215122	Keine Metainformationen	D-CAN	6f1	8	40 04 18 02 ff ff 00 00
0:00:00,215406	SYNC	F-CAN	080	5	00 00 c4 7d 00
0:00:00,216101	Keine Metainformationen	K-CAN System	6f1	6	40 04 18 02 ff ff
0:00:00,216462	CLU1	F-CAN	0cd	8	00 00 00 00 20 00 70 63
0:00:00,216717	Keine Metainformationen	F-CAN	0d1	8	aa 00 00 00 1c 00 70 e7
0:00:00,216943	CLU3	F-CAN	0d4	8	00 00 00 00 00 32 70 0a
0:00:00,217641	Keine Metainformationen	K-CAN System	640	8	f1 02 58 00 ff b1 64 00
0:00:00,220065	Keine Metainformationen	D-CAN	640	8	f1 02 58 00 ff ff ff ff
0:00:00,221791	Lenkradwinkel Oben 2 F-CAN	F-CAN	0c9	8	26 02 20 ef 2f ac 1b 44
0:00:00,236452	CLU1	F-CAN	0cd	8	00 00 00 00 20 00 90 d1
0:00:00,236690	Keine Metainformationen	F-CAN	0d1	8	aa 00 00 00 1c 00 90 55

Figure 5.5: Screenshot of AutobahnVis: **(a)** Selection View: allows to select and filter messages, bus systems and signals; **(b)** Message View: shows messages sent over the bus systems ordered by time, a Multi-Color search shows four request/response pairs; **(c)** Signal View: shows signal value plots for selected signals; and **(d)** List View: provides a traditional text based representation of message details.

messages of one bus system (or one ECU) in the form of small black semi-transparent diamonds, reminiscent of cars on an Autobahn, as seen in Figure 5.5. AutobahnVis uses a zoomable interface that allows an analyst to see an entire trace or to focus on single messages at 100 μs/pixel resolution. We used diamonds to represent messages as their peaks indicate an exact time-point and used a semi-transparent fill to allow for better visibility of intersecting messages. To further increase the visibility of messages in crowded areas, in particular for zoomed-out views, we introduced another set of icons which represent groups of messages as white hexagons (R-6). These hexagons are presented when more than four messages fall within a range of four pixels between each other. The width of this aggregate icon depends on the first and last message in the group and its transparency encodes the number of messages per second within the timeframe represented by the icon. Per zoom level each message group icon holds a maximum number of messages in order to prevent the construction of very large hexagons; depending on the zoom level we used heuristics e. g., ten per hexagon on the lowest zoom-level.

From our exploratory field study, we learned that productive work with AutobahnVis would require a close connection to textual trace representations, message filters, and signal plots. For fast access (R-5) to this data, we rebuilt these views and functions in resemblance to other tools used by engineers and directly integrated them with the AutobahnVis module. Together with the central *Message View* described above, we thus added three additional views to AutobahnVis (see Figure 5.5). In the *Selection View* on the left side the user can select/deselect—and thus filter—bus systems, ECUs, messages and signals for a closer examination in the other views. The *Signal View* visualizes signals using a value/time plot (similar to traditional visualization techniques), and the *List View* represents the raw data in the usual textual form (R-10). Both views can either be coordinated with the Message View's timeline through synchronous scrolling or independently navigated on a detached timeline (R-4). Linking and brushing is supported in both modes. Detailed information can either be accessed by hovering the mouse over elements in the Message and Signal View or by looking at the List View at the bottom (R-5). According to engineers' requests during our user-centered design process, we integrated several additional usability features which we corporately developed and discussed together with our target users. Of those, the most important, frequently demanded, feature was the Multi-Color Search which allows the engineers to concurrently search for varying messages, to brush them with different colors in the Message View and therefore allows for systematic comparison of message timings.

5.3.2 Evaluation

To evaluate our design of AutobahnVis, we collected informal feedback during our participatory design approach and frequently engaged in discussions with domain experts. Five automotive analysis engineers (lead analysts) used AutobahnVis for six weeks and provided us with feedback via telephone or in personal meetings. During this time, the engineers used the tool once or twice a week for analyzing traces from their daily work. The usage duration of the tools ranged between five minutes and one hour. Additionally, we received qualitative feedback from seven domain experts during a one-hour test session with our tool. Based on these evaluations, we improved our tool and discuss its six main benefits:

Reproducing Message Dispersion: Wide distribution of information and high interconnectivity of functions are a common reason for complex errors (see Section 5.1.3). Three of our lead analysts used AutobahnVis to track messages sent over various bus systems. Differing bus characteristics often caused errors and AutobahnVis helped them to better understand the correlations between time shifts over gateways and dispersion of messages over several bus systems. A similar tracking of errors across bus systems had not been possible before.

Close Correlation Between Signal and Message Timing: Two of our lead analysts used AutobahnVis to establish a direct and close link between signal value changes and messages. The side-by-side presentation of the Signal and Message Views, together with the close coordination of their timelines turned out to be very helpful to correlate signal value changes and reliable messages in the trace. With traditional tools this important correlation was not as immediately evident but had to be manually drawn by sequentially jumping between list and signal views. Engineers appreciated the reduced work and attention shifts required to perform the analysis.

Tracking Request/Response Pairs: One lead analyst used AutobahnVis to validate request/response message pairs for correctness and timing conditions. He used the Multi-Color search and filters to highlight all messages of interest and to subsequently explore timing conditions along the timeline (see Figure 5.5).

Figure 5.6: AutobahnVis' entirely zoomed-out Message View showing an overview over a 16-minute trace. The communication gap on all bus systems indicates a measuring hardware defect.

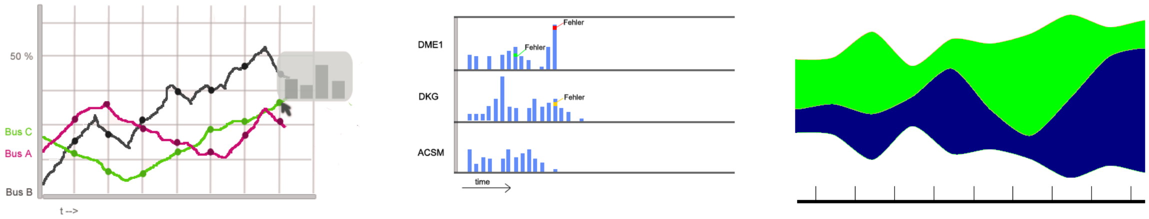
Tracking Cyclic Messaging: Another lead analyst used AutobahnVis to track cyclic messaging. Cyclic messaging composes 80% of in-car bus communication. Cross-correlations between multiple message cycles often cause time shifts which in turn can lead to errors. He used filtering, (Multi-Color) searching and zooming to track cyclic behavior of varying messages. This aspect was also highly valued by three of our test users.

Fast Detection of Measuring Hardware Breakdowns: One lead analyst successfully applied AutobahnVis to identify a measuring hardware breakdown. Several automatic errors had been indicated and important messages were missing. By zooming out in the Message View a break on all bus systems was recognized immediately and the error could be assigned to a measuring hardware defect. Figure 5.6 shows a screenshot of the visualization of that erroneous trace.

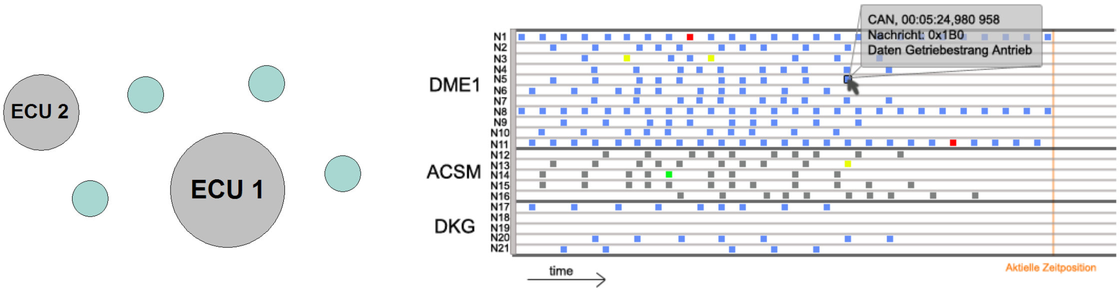
Detection of Message Bursts: Four of our test users stated that they saw a major advantage of AutobahnVis in gaining novel insights on message bursts. Filtering and zooming can help to detect when one or more ECUs "spam" a bus system and other messages might have been displaced. This behavior is a well-known source for complex errors.

5.3.3 User-centered Development: From Paper Mockups to an Integrated Tool

The development process we used for developing AutobahnVis is a very good example of how we in general proceeded in developing our tools in close collaboration with automotive engineers. By providing this representative example in more detail, I want to give the reader the opportunity to learn more about the steps we took, which kind of prototyping techniques we used, and how this influenced us in coming up with a final solution. All work on AutobahnVis was conducted



(a) Line Chart (b) Bar Chart (c) ThemeRiver



(d) Bubble View (e) Message View

Figure 5.7: Early visualization mockups in the AutobahnVis project.

in close collaboration with five trace analysts and three analysis tool developers (Carmen). Before we developed the final version presented above, we involved them but also other engineers and external testers over a period of two years in a variety of studies with prototypes of different fidelity. Our approach is similar to the staged design and development process applied by McLachlan et al. in their LiveRAC project [MMKN08] and we hope that our approach might serve as another best practice example in this context. In general, our process could be divided into three major steps:

1. Paper Prototyping As a very first step, we[6] used paper mockups in order to discuss possible solutions and features with our target users. We created a catalog of mockups which contained existing visualizations from the InfoVis domain, newly developed visualization concepts and adaptations of traditional automotive solutions. The catalog most importantly included:

- A traditional *List View* (no figure).
- *Traditional Views* such as line and bar charts (cf. Figure 5.7-a and b).
- A *ThemeRiver* [HHW+02] which we thought might be useful for showing trends in the communication process, for instance, showing the number of messages sent over the various bus systems over time (cf. Figure 5.7-c).

---

[6] Early work on AutobahnVis has been published in [SKHB09]. Any use of “we” in this context therefore refers to Michael Sedlmair, Benjamin Kunze, Wolfgang Hintermaier and Andreas Butz.

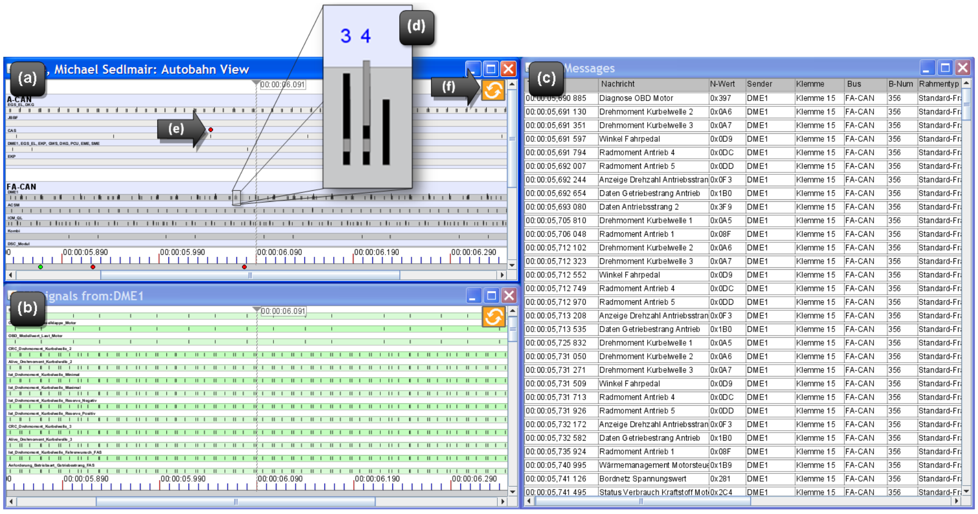
View	Figure	Desire	Reasons	Mapped use cases
List View	*no figure*	Strongly requested	The List View is irreplaceable to show the detailed data in tabulated format (cf. R-5)	- Showing precise detailed information - Exploring data - Monitoring data - Analyze data
Traditional Views (line and bar charts)	5.7-a 5.7-b	Optionally requested	These visualization forms are well known and easy to understand. Experts like using them but demand a higher degree of interactivity.	- Showing state of the components - Showing activity history of components - Showing traffic volume - Finding transition states
ThemeRiver	5.7-c	Not requested	High abstraction level with less level of detail was assessed to be not applicable to the in-car communication domain.	- Showing combined trends
Bubble View	5.7-d	Optionally requested	Although this view had no direct use case mapping it was well liked in discussions because of its innovative character.	*No use cases found*
Message View	5.7-e	Strongly requested	This visualization reached a wide acceptance by the expert users because it supported a common mental model with a simple and pleasant visual vocabulary.	- Finding Errors - Monitoring Cyclic Traffic - Monitoring the In-Car Communication - Getting familiar with the car network domain - Explore Cause-Effect relations

Table 5.1: Results of evaluating early design mockups in the AutobahnVis project.

- A design idea which we called the *Bubble View* which was based on scale-free networks [BO04]. We suggested this with the idea to visualize different points of an in-car network, for instance, ECUs or functional blocks, and draw them as bubbles and directed graphs with arrows between them. The basic idea was to adapt the bubbles' sizes, whenever the activity of its component increases in order to present a current state of the system and to show the "big players" at a specific time (cf. Figure 5.7-d).
- The initial design sketch of our *Message View* concept (cf. Figure 5.7-e).

To evaluate our design ideas, we discussed printed versions (cf. Figure 5.7) with our eight domain collaborators. Their input narrowed the visualization concepts down very quickly and led to a short list of pragmatic design solutions. We asked them to assign use cases to the visualization concepts and to imagine and illustrate in each case an example how it might be used to visualize the underlying data, such as: *"In my opinion the Message View could be used to show message bursts and to investigate frequencies in sending actions"*. Based on this evaluation, we classified the visualization concepts into three categories: Strongly requested, optional, or not requested. The results of this classification as well as the reasons and the mapped use cases can be found in Table 5.1.

High-fidelity Prototype In order to turn the basic concepts into a high-fidelity prototype, we took the most requested visualization concepts, namely the Message View and the List View and implemented a first stand-alone version that was not integrated into engineers' analysis software (cf. Figure 5.8). We basically used this version as a proof of concept and as a test environment for doing several usability and utility studies—mostly think-aloud studies—both with engineers



	Nachricht	N-Wert	Sender	Klemme	Bus	B-Num	Rahmentyp
00:00:05,690 885	Diagnose OBD Motor	0x397	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,691 130	Drehmoment Kurbelwelle 2	0x0A6	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,691 351	Drehmoment Kurbelwelle 3	0x0A7	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,691 597	Winkel Fahrpedal	0x0D9	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,691 794	Radmoment Antrieb 4	0x0DC	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,692 007	Radmoment Antrieb 5	0x0DD	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,692 244	Anzeige Drehzahl Antriebsstran	0x0F3	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,692 654	Daten Getriebestrang Antrieb	0x1B0	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,693 080	Daten Antriebsstrang 2	0x3F9	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,705 810	Drehmoment Kurbelwelle 1	0x0A5	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,706 048	Radmoment Antrieb 1	0x08F	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,712 102	Drehmoment Kurbelwelle 2	0x0A6	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,712 323	Drehmoment Kurbelwelle 3	0x0A7	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,712 552	Winkel Fahrpedal	0x0D9	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,712 749	Radmoment Antrieb 4	0x0DC	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,712 970	Radmoment Antrieb 5	0x0DD	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,713 208	Anzeige Drehzahl Antriebsstran	0x0F3	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,713 535	Daten Getriebestrang Antrieb	0x1B0	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,725 832	Drehmoment Kurbelwelle 1	0x0A5	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,731 050	Drehmoment Kurbelwelle 2	0x0A6	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,731 271	Drehmoment Kurbelwelle 3	0x0A7	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,731 509	Winkel Fahrpedal	0x0D9	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,731 713	Radmoment Antrieb 4	0x0DC	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,731 926	Radmoment Antrieb 5	0x0DD	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,732 172	Anzeige Drehzahl Antriebsstran	0x0F3	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,732 582	Daten Getriebestrang Antrieb	0x1B0	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,735 924	Radmoment Antrieb 1	0x08F	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,740 995	Wärmemanagement Motorsteu	0x1B9	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,741 126	Bordnetz Spannungswert	0x281	DME1	Klemme 15	FA-CAN	356	Standard-Fr
00:00:05,741 495	Status Verbrauch Kraftstoff Mot	0x2C4	DME1	Klemme 15	FA-CAN	356	Standard-Fr

Figure 5.8: Screenshot of the early AutobahnVis high-fidelity prototype with: **(a)** Message View; **(b)** Signal View which is similar to the Message View but shows signals instead of messages; **(c)** a Message List; **(d)** a magnified excerpt from the Message View illustrating the concept of message stacking we used for representing messages that were sent very closely in time; **(e)** a POI that can be set by the user; and **(f)** the Sync Button.

.

and with outside testers. These early studies, provided us with many interesting insights which strongly influenced (a) us in continuing the project, (b) our stakeholders' interests in our work, and (c) our design decisions of the final version of AutobahnVis. In the following, I briefly discuss design variations and the results of our earlier studies with the high-fidelity prototype.

The most interesting **summative result** we derived from our early studies with domain experts was the indication of a novel mental model that appeared for thinking and discussing about trace data. In a think-aloud study with five domain experts, it was very interesting to see them starting to explain things directly by means of the novel Autobahn metaphor, such as: *"As you can see, we have lots of traffic on this ECU's lane"*, *"Oh, what does that burst of message rectangles [—we initially used rectangles instead of diamonds—] mean?"*, or *"On the road you can perfectly track cyclic messages"*. Note, that we already found indications in these studies for the potential value of detecting message bursts and better understanding cyclic messaging (see above for more details). Another hint for the appearance of a novel mental model was an observation we made during a presentation with a live demo of our high-fidelity prototype in a meeting of analysis experts. During the live demo, the attendees suddenly started to discuss a known problem of a specific ECU by means of the Autobahn representation. The problem was about the ECU's *"spamming"* activities on the bus and the engineers started arguing: *"Does anyone know why the LRR ECU sends message bursts in this compressed cycle?"*, *"In my opinion that has to be the reason for the bus spam."*, *"No, the other ECUs seem to work normally"*, etc.

On the other hand, we also got much **formative feedback** which was very helpful for redesigning the final tool. In our early prototype we, for instance, used a Signal View that showed signals in the same way as the Message View showed messages, i. e., with lanes and rectangles (cf. Figure 5.8-b). However, all our participants mentioned that this view does not provide any further value beyond the message view. We therefore excluded it from the final tool. With the earlier prototype we also stacked messages that appeared nearly simultaneously on the bus system and outlined the number of these messages above the stack (see Figure 5.8-d). In our think-aloud studies we found that this stacking metaphor was misunderstood by three out of five participants. Instead of representing multiple messages, it was interpreted as a length coding of information and guessed to be the messages' byte length. Inquiring this in detail, it showed that coding the messages' length would not be beneficial for the engineers at all. While after resolving the misconception, they had no further problems in understanding it, we took this initial misunderstanding very seriously and refused message stacking in our final concept. Another point of criticism was the understanding of the coordination feature. In our early version of AutobahnVis we allowed the user to open an arbitrary number of Message, Signal and List Views. Each view held a synchronize icon placed in the upper right corner. By pressing, the user added this particular view to a list of views that were synchronized in time. Two of our study subjects, however, wondered that they had to select more than one view to start the synchronization action. Their current mental model matched more an "all-or-none" synchronization feature and the synchronization of sub-groups of views was not self-explanatory. We addressed this aspect in our final tool and left out group synchronization.

Finally, we also presented our prototype to Carmen's tool developers and outlined the potentials we found in our user studies. This helped us to convince them to allow us connecting our later approach closely to their software platform.

In general, however, there were two basic technical drawbacks of the first high-fidelity prototype that hindered our target engineers in productively using the tool and us in studying the tool under real condition. These aspects were the missing integration into daily working practices (cf. R-9 and R-8) and scalability restrictions as our tool could only work with traces up to five seconds.

Final Tool, Integration, Limitations We therefore discarded the first prototype and used the lessons learned to implement a second, closely integrated (R-9) system of AutobahnVis that I have described above. For our novel tool, we, on the one hand, slightly improved and adapted the design according to the results of our early user studies: We, for instance, used diamonds instead of rectangles for messages as they indicate an exact point in time, we introduced group hexagons for better scalability, we changed the Signal View to a line plot representation, provided more and better filter possibilities, and changed the coordination paradigm. On the other hand, we invested a significant amount of work in integrating the tool and in making it scalable to real traces. Both aspects required us to overcome a variety of technical, non InfoVis/VA specific challenges, such as handling the proprietary and restricted interfaces of Carmen we were allowed to use.

The current version of AutobahnVis is available as a block-module in Carmen's visual test configurator (cf. Figure 5.2) and, in doing so, can closely interact with several trace analysis modules, most importantly the Replay Module and diverse filter modules (R-4). Via the Replay Module an engineer can load a trace file and replay the file either in real time or up to 10^5 times faster/slower.

AutobahnVis then can be directly connected to this module and uses the data currently worked on by an engineer to fill the visualization with plain messages (R-8). In-between the Replay Module and AutobahnVis the engineer can use common modules for filtering the data, for instance, in terms of relevant bus systems, ECUs and/or messages.

While we generally succeeded in making AutobahnVis available as a module in Carmen, there are still some minor restrictions based on Carmen's connection technology for external modules we were allowed to use. Compared to the first prototype, the final version of AutobahnVis is restricted in showing messages solely separated by bus systems but not by ECUs. This functionality is not supported by the connection technology we used. For the same reason, we currently cannot distinguish between incoming and outgoing messages on bus systems which would—according to lead analysts' statements—further improve the Message View.

5.3.4 Discussion and Adoption

AutobahnVis is a tool that we built to visually support current techniques of message-based browsing and analyzing raw data. Due to the fact that we incorporated AutobahnVis into Carmen, a widely-used and accepted automotive software analysis platform used in our company, we had the chance to evaluate our tool under real conditions. In general, the feedback we got in our studies was very positive such as stated by one of our lead analysts: *"AutobahnVis heralds the future of trace analysis tools' interfaces!"* Following our initial design recommendation, we also could show that our design decisions did indeed support engineers with several of their analysis needs using AutobahnVis (most importantly, R-1–5, R-9 and R-10).

Even several months after conducting the studies, asking our participants revealed that they are all still using AutobahnVis on an occasional basis in a similar frequency as they used it during our studies (once or twice a week). One engineer stated: *"I would use it even more often, but you know there are still these little technical restrictions [...] when these limitations will be overcome, I guess I would use it everyday!"*. To make this possible, we recently transferred our final software to the Carmen tool developers who plan to overcome the very last technical hurdles and directly integrate our concept into the Carmen core tool and *"not just as a plugin"*.

5.4 *Cardiogram*: Using State Machines for Analysis

The intention of our second Carmen module, Cardiogram, was to provide a tool that would support engineers' analysis process with core data reduction and transformation components and with a final visual analysis component. Our main goal with Cardiogram was to reduce the amount of work and the information load engineers had to handle by providing automation techniques (R-7), and to provide new perspectives and novel insights on particular states of a vehicle during testing (R-2). The general concept behind Cardiogram was to use state machine analysis from model-based testing techniques [UL07, GHP02, BK08]. In doing so, we want to allow for the

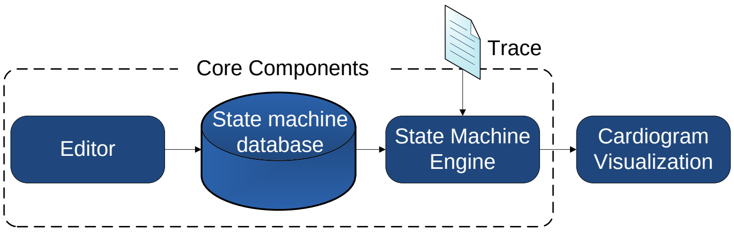


Figure 5.9: Workflow of our Cardiogram approach.

analysis of predefined vehicle states in communication traces, to automatically filter the data accordingly (R-6), and to provide semi-automatic analysis of large amounts of trace files (R-7).

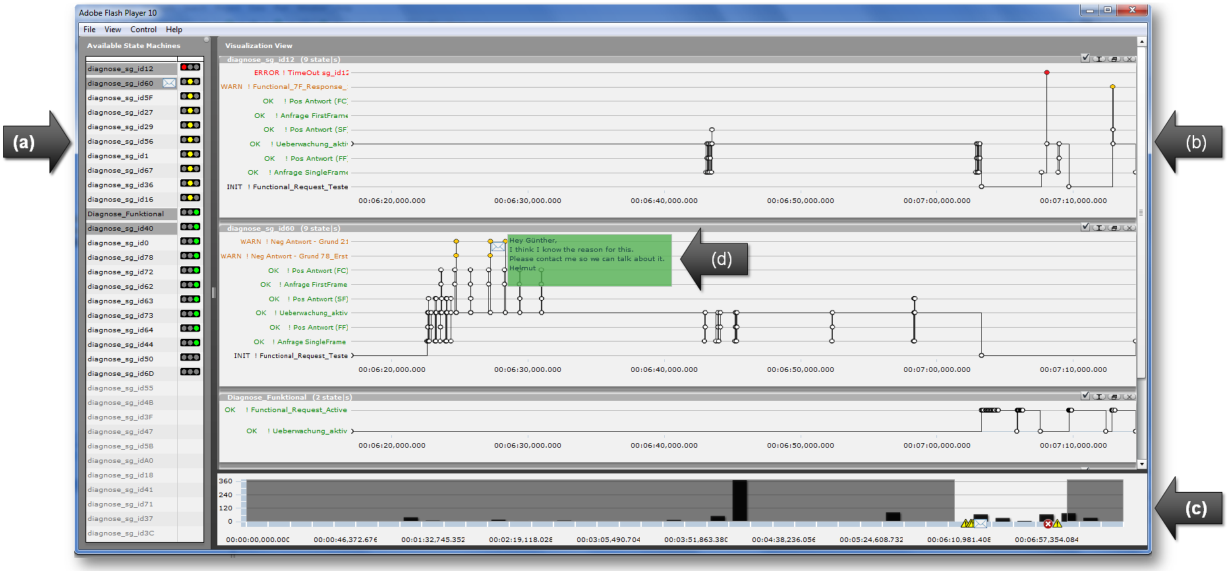
5.4.1 Approach

Data analysis with Cardiogram involves four main steps. These are described next and schematically represented in Figure 5.9.

Editor—Defining State Machines: By means of external graphical and text editors, including an Eclipse plugin we designed, engineers can specify state machines which define a set of vehicle states such as *"window front left open"* and transitions between states such as *"open window."* Engineers define these states in XML using a specific DTD. Two main kinds of state machine designs can be specified: Verification State Machines and Context Information State Machines. Verification State Machines test a predefined situation, e. g., if a specific error condition is met or not. Context Information State Machines are more general and represent generic vehicle information such as mechanical activities (e. g., monitoring window opening) or ECU behavior. In the following, we refer to both simply as state machines. A set of predefined prefixes allow engineers to annotate states in the XML files: Each state machine has one specific INIT state, for all other states engineers can add ERROR, WARNING or OKAY to characterize and to emphasize these states. This annotation is an important first step for analyzing trace data using these states machines.

Database—Making State Machines Available: Together with a description, all state machines are stored in a central database using a common XML format, and are subsequently available to all analysis engineers. This supports collaborative data analysis (R-11) and reduces redundant work (R-8).

Engine—Pre-Analyzing Traces Using State Machines: The State Machine Evaluation Engine is integrated as a module in Carmen. It supports loading specific state machines for analysis and imports traces by utilizing other modules in Carmen, such as a Replay Module (R-8). The engine then automatically computes a *transition table* containing all state changes with exact timing information for all loaded state machines. Additionally, a global tag indicates the occurrences of ERROR and WARNING states for each state machine which can be used for later analysis. The



Hey Günther,
I think I know the reason for this.
Please contact me so we can talk about it.
Helmut

Figure 5.10: Screenshot of the Cardiogram Visualization: **(a)** State Machine View showing all tested state machines ordered by relevance for bugfixing; **(b)** Visualization View with several detailed state/time plots showing transitions via vertical and horizontal lines and additional glyphs at target states; **(c)** a combined range slider/ overview bar showing the sum of all transitions within discrete time intervals; **(d)** shows an annotation which can be used for collaboration purposes.

engine thus reduces and preprocesses the data to be analyzed by an engineer (R-6) and makes novel overview techniques possible (R-1).

Visualization—Exploring Transition Tables on Demand: The purpose of the Cardiogram Visualization is to support the exploration of errors, warnings, or other hints which may require further inspection. Cardiogram visualizes all state transitions and helps to provide insight into incorrect vehicle states and into timing correlations between state machines. Together with the three data preparation steps presented above, this tool allows engineers to gain a novel perspective on complex dependencies of in-car networks (R-2) and especially helps to correlate logical with timing aspects (R-3). Next I describe the design decision around the Cardiogram visualization in more detail in relation to the design recommendations introduced in Section 5.1.4.

5.4.2 Visualization Design

The Cardiogram visualization module represents transition files computed by the three core data preparation components. It consists of two main views (see Figure 5.10). Similar to AutobahnVis, on the left side the *State Machine View* lists all state machines tested on a specific trace by the engine. As requested by our target group, this list is sorted according to priority based on aggregated state machine results. First state machines are shown that hit at least one ERROR state, then the ones with at least one WARNING (and no ERROR), then at least one OKAY (no ERROR, no WARNING) and last the state machines without any of the three. Due to their familiarity (R-

10), we used traffic light icons next to each state machine entry that encoded the aggregated state machine outcomes ERROR, WARNING, OKAY using the colors red, yellow, green, and no color coding for other outcomes (R-6). Inactive state machines (no transitions, only INIT state) are set at the very end of the list, have no traffic lights, and are grayed out. From this list, engineers can select one or more state machines for detailed analysis. For each selected state machine a familiar state/time plot is shown in the *Visualization View* showing all transitions of this state machine according to its transition table. Time is plotted on the x-axis while the y-axis holds all states sorted by annotation in the following order: ERROR, WARNING, OKAY, INIT, and then states with no annotation. States are connected by vertical and horizontal lines which give an indication of transition changes and which states were active at a given time. Transitions are additionally marked with small white dots at the target state for a better visibility in dense areas and for special cases such as fast sequences of transitions resulting in (nearly) vertical lines (see Figure 5.10). Detailed information about transitions can be retrieved by hovering the mouse pointer over these dots and a fast access to the underlying trace file is given by an integrated backlink to Carmen's list presentation of the trace (R-5).

At startup, the x-axis of the Visualization View presents the entire recording time and provides an overview of all transitions in the selected state machine (R-1). However, as our plots have to hold hundreds to tens of thousands of transitions it is, similar to AutobahnVis, implemented as a horizontally zoomable interface. To explore timings of transitions in more detail, engineers can zoom into the global timeline using the mouse wheel or a range slider at the bottom of the visualization panel. Along with indicating the current zoom region, the range slider bar is enhanced by an overview visualization using a bar chart that shows the sum of all transitions on the global timeline (R-1). By default all selected state machines are subsumed according to a time slot size pre-defined by the user. Therefore, each time slot shows a bar whose height encodes the overall number of transitions in the selected state machines and in doing so provides an indication about busy, calm, steady and void areas (R-6). Moreover, changes in ERROR and WARNING states are indicated with red and yellow dots in time/state plots and with accordingly colored, domain specific symbols in the overview bar. This supports fast readability of transitions that could be relevant for bugfixing. Interactive annotation of the data is also supported. Each analyst can freely add colored notes directly into a state/time plot. Once an annotation is set, it will be displayed at the exact timestamp and state it was created at. Symbols are also shown in the state machine list and in the overview bar where they indicate the position within the global timeline. These annotations can be exported together with the data and sent to a colleague for further inspection or inquiry (R-8 and R-11), see Figure 5.10).

Similar to AutobahnVis, we integrated a variety of other interactive features, including: keyboard short-cuts for all features, unrestricted vertical scaling of state machine plots, minimization and closing of state machine plots, drag and drop positioning of the state machine plots to allow side-by-side comparison, dynamic adding or subtractions of state machines to or from the overview bar, and the free configuration of nearly all system features and settings.

5.4.3 Evaluation

Cardiogram was developed and evaluated in two separate phases for (1) the core components and (2) the Cardiogram visualization (see Figure 5.9). First, we implemented and tested the core components (editor, database, and engine) in order to evaluate the novel automation and abstraction approach together with a textual representation of statistical results (number of errors/warnings occurred) and transition lists for detailed inspection on demand. This approach was validated in a field study with 15 domain experts over a period of twelve months—what is comparatively long by InfoVis/VA standards. In these twelve months, experts used the tool situationally during their daily activities, some of them nearly every day and others just once or twice a month. They created state machines on their own, and included the tool in their data analysis procedure. The usage length of such a single, situational session varied from several minutes up to three hours. Due to IPR restrictions and our stakeholder's opinions we did not automatically log usage data but referred to debriefings with our participants. For this purpose, we arranged bi-weekly meetings, discussed our users' experiences with the tool and elicited feedback on benefits and areas for improvement. The results of the study uncovered two main categories of benefits for our new approach:

Externalization of Expert Knowledge: Analysis experts created state machines to capture their expertise for verification and abstraction of complex behavior. Many of the state machines were specified to reproduce highly distributed procedures such as booting a car, starting the motor, or shutting down the vehicle. These state machines included up to 25 different states as well as clusters of sub-state machines. Each state in turn abstracted up to 15 signals in order to form combined and interpreted information about specific vehicle behavior. Externalizing this knowledge into state machines made it widely available for other engineers who benefited even without specific knowledge about this particular behavior.

Mass Analysis Instead of Sample-Tests: Our abstraction and automation techniques facilitated a broad analysis of a great number of traces. One engineer used the core components to automatically analyze 12,000 traces with 50,000 messages on average within one day. Based on the global result tags of state machines, he could isolate three important traces which he examined in more depth to verify a specific hypothesis. Previously, testing of this data relied on analyzing and debugging samples of the data, our approach, however, allowed the analysis of hundreds to tens of thousands of traces, and to test or verify hypotheses on a broad testing basis.

After building the core components and simultaneously to evaluating them, we developed the Cardiogram visualization by using our usual development process, i. e., participatory design with lead users, paper mockups, iterative refinement of releases, heuristic evaluations with outside testers, etc. Then we qualitatively evaluated the visualization with six domain experts during a one-hour session in which they used the visualization on their own data and/or on test datasets we provided. Additionally we got feedback from two test users who used Cardiogram for a eight week period. This evaluation showed three main benefits of the visualization:

Correlation Between State Machines: All of our participants stated that the Cardiogram visualization was enormously helpful to understand and explore correlations between dependent state machines. For example, we saw the Cardiogram visualization being used to explore several parallel procedures involved in shutting down a car. Correct shutdowns are of high importance as errors can lead to high consumption of electricity and load on the car's battery. Shutting down a car first involves the shutdown of all relevant subsystems. One of our participants used a set of twelve state machines together with our visualization and verified the shut-down behavior of all sub-systems as well as that of the entire car. Using the visualization allowed him to compare timings, to verify correctness of temporal order, and to correlate the transitions of the state machines. This in-depth analysis was not possible previously or with the core components alone.

Trace-Related Overview of State Machine Activities: Four of our participants mentioned that the Cardiogram visualization provided a good overview over all trace-related, logical, and temporal activities. The list of all state machines showed valuable information to them on which tests had been conducted and whether they had been successful or erroneous. Freely selecting, combining, and repositioning of state machines helped them to further explore erroneous state machines and to derive correlations between them. Additionally, three participants mentioned that when zooming out, the time line provided a valuable overview over a state machine's global activities and helped to quickly detect transition peaks.

Verification and Re-Engineering of State Machines: Two of our participants used the Cardiogram visualization to verify state machines currently under construction. Loading these state machines together with a known test trace helped them to validate the correctness of transitions and to estimate possible interaction with other state machines.

5.4.4 Discussion and Adoption

In this section, I introduced Cardiogram, a visual analytics approach combining automation and visualization techniques in order to provide test engineers with a richer and more efficient methodology for analyzing many and large traces. Similar to AutobahnVis, Cardiogram was integrated into Carmen and subsequently tested over a longer period of time. In summary, the analysis of the Cardiogram core components and visualization showed that the tool successfully supported the engineer's analysis requirements and addressed known challenges, most importantly abstraction (R-6), automation (R-7), collaboration (R-11) and novel perspectives (R-1, R-2, R-3). Cardiogram's abstraction and automation techniques were valuable tools for reducing and preprocessing the data and Cardiogram's visualization allowed for the analysis of comprehensive and global aspects.

The visualization is currently limited to a specific amount of information. Especially, when a large number of different states had to be taken into account, it became hard for our analysts to deduce global cross-correlations from our visualization design. To counterbalance this limitation future refinements should consider to integrate interactive filtering of states. In contrast to Cardiogram's valuable abstraction and automation approach, however, it is also often still necessary to directly analyze the raw information in order to be able to define useful data abstractions and to verify hypotheses on unknown aspects. For this purpose engineers, however, can use traditional tools for raw data inspection or our novel AutobahnVis approach we introduced in the previous section.

At the time of writing this thesis, the Cardiogram project was still actively carried on by BMW. To date, even more engineers than participated in our studies (approximately 30 engineers) use the core components (cf. Section 5.4.1) and also more and more of them start to use the visualization as well. Similar to AutobahnVis, we recently transferred our software to the Carmen tool developers who will address the limitations named above, further extend the tool by closely cooperating with the end users and directly embed our solution in Carmen.

5.5 *ProgSpy2010*: Visualizing Flash Activities

The final tool I present in this chapter is similar to Cardiogram's visualization (see previous section) but was designed with and for another target group, namely flash diagnosis engineers. While FIT analysts, the target group we focused on with the previous three tools, analyze fully-built cars to verify their correctness, flash analysts test, diagnose and optimize the process of uploading software or software updates onto ECUs. Just like FIT analysts these engineers use trace analysis for their purpose and therefore are often confronted with similar problems and challenges. Coincidentally, we[7] heard about plans to redesign their main analysis software, ProgSpy, due to novel requirements stemming from technical changes of the vehicle's hardware. In this context, we contacted an engineer from the responsible department, offered our help, and finally ended up in collaboratively redesigning and re-implementing the ProgSpy tool.

In the following, I first provide some background information to flash analysts' work based on our pre-design studies with them. Then, I show how we used our ideas from the Cardiogram project to redesign ProgSpy and describe the resulting tool ProgSpy2010. The section concludes with findings from a small-scale user study and a discussion of the results.

5.5.1 Problem and Requirements Analysis

In line with our usual design and development approach, we started our project by conducting pre-design studies in order to better understand our target users' tasks, tools, practices and challenges. To do so, we closely collaborated with the flash analyst who we contacted and who took responsibility for the project from the end-user side. We interviewed him, asked to introduce us into the current ProgSpy tool and learned in particular by informally discussing flash analysis practices with this lead user. Additionally, we conducted one-hour long contextual inquiries with further four target users, observed them working with ProgSpy and asked questions for clarifica-

---

[7] Any use of "we" in this section refers to Michael Sedlmair and Alex Messner.

tion and illustrating typical tasks with the tool. The findings of our studies can be summarized as follows:

Technical Background As already stated above, flash analysts are responsible for analysing the process of deploying software on ECUs. In the automotive domain, this process is referred to as *flashing ECUs*, hence our target users were called flash analysts. In earlier days, flashing ECUs was done via point-to-point connections between the computer holding the software source and the respective ECU. However, due to the increased number of ECUs as well as the increased amount of software in a car, this process became unmanageable as ECUs had to be manually removed each time a software update was necessary. Therefore, the strategy was changed and ECUs nowadays are flashed directly via the communication networks which are available in the car. For this purpose, the computer holding the software source is connected to a bus system in the same way diagnosis hardware is connected and the software/software updates are directly sent over one or several bus systems to the ECUs in question. During the software update process, each ECU runs thorough several *programming phases* (usually 5–10) which are defined in a state machine[8]. For diagnosing purposes, each ECU sends specific *diagnosis messages* back onto the bus systems (UDS messages [ISO04]). These diagnosis messages include information about an ECU's current programming phase, about errors and warnings which appeared within an ECU's flash activity, and about the success of the process in general.

Current Practices For analyzing flash processes, our target users recorded complete traces with usual recording hardware (cf. Section 5.1.2) and used a tool called ProgSpy to automatically pre-filter the trace so that only diagnosis messages remain. These diagnosis messages—usually ranging from hundreds to thousands—are interpreted and represented in a common text-based list. The resulting lists are then manually scrolled in order to identify errors, to check timing conditions, and to validate the correctness of programming phases. To support flash engineers with aggregated information, ProgSpy provides two other views holding (a) some basic statistical information, such as the number of occurred warnings, the number of state changes per ECU, or the duration of programming phases in seconds, and (b) a pre-filtered list of errors and warnings (cf. Figure 5.11).

Discussion and Design Requirements Asking our target users about the textual interface usually revealed that they are relatively happy with it and actually want to retain a similar interface with the newer version of ProgSpy. Indeed, we did not observe major problems in detecting single errors and warnings from the lists—which was mentioned from our engineers as the most important task for their work. In contrast to this we, however, observed that it often took them long to understand and correlate the timing of programming phases, though asking about the relevance of such tasks revealed that they are also of high importance for their daily work. Asking about further possible improvements, revealed a desire for speeding up the work with lists and

---

[8] Due to IPR restrictions we are not allowed to explain these state machines in more detail.

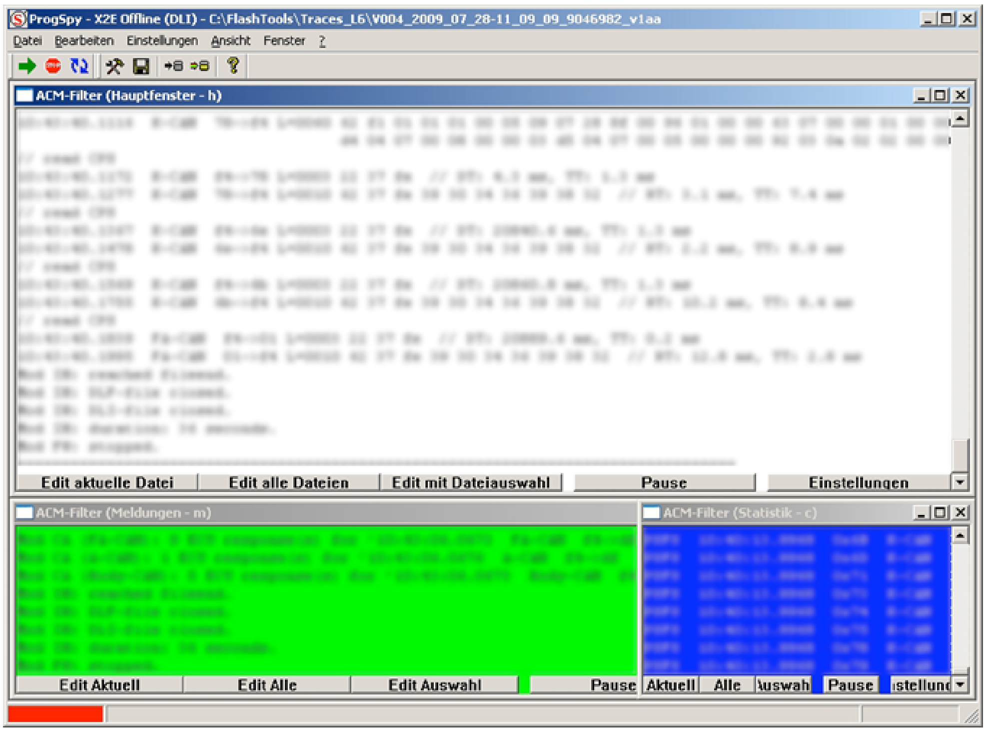
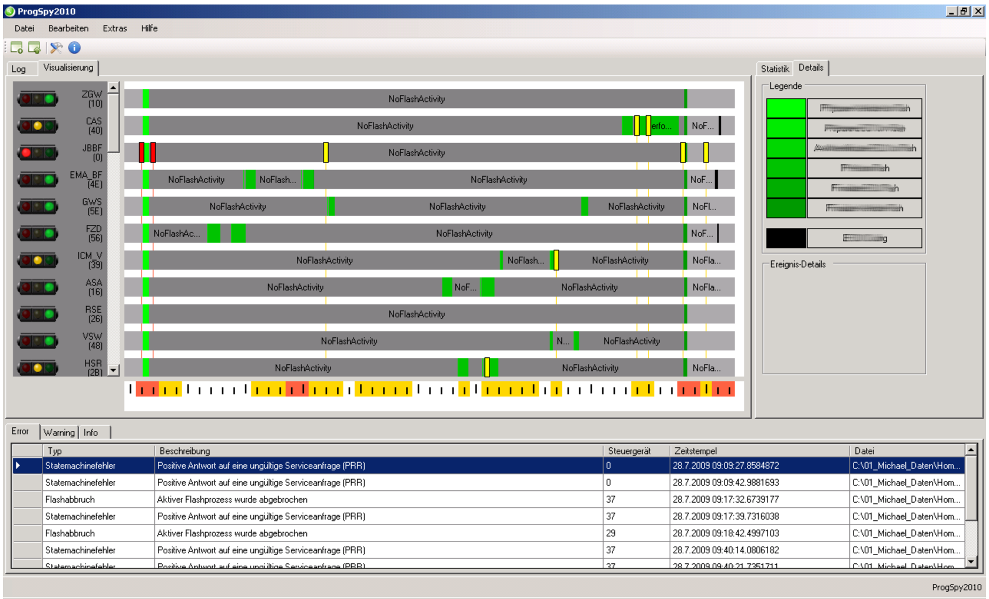


Figure 5.11: Screenshot of the former ProgSpy tool (Details have been blurred due to IPR restrictions).

for providing more overview, as stated by one of our participants: *"the list is good for analyzing all details, but it took me long to go through all the messages and it would be great if I could get a faster impression what's happening in the trace"*. Based on our findings and together with our lead user, we derived three basic design requirements. These requirements are very similar to the recommendations we provided for FIT analysts and can be seen as sub-domain specific interpretations of some of the more general design recommendations we listed above. The requirements are:

1. Provide an overview over an entire flash process (R-1).
2. Provide a clearer representation of programming phases' timing and ECU affiliation (R-2 and R-3).
3. Retain traditional list views for diagnosis messages, errors, warnings and statistical information should be retained (R-5).



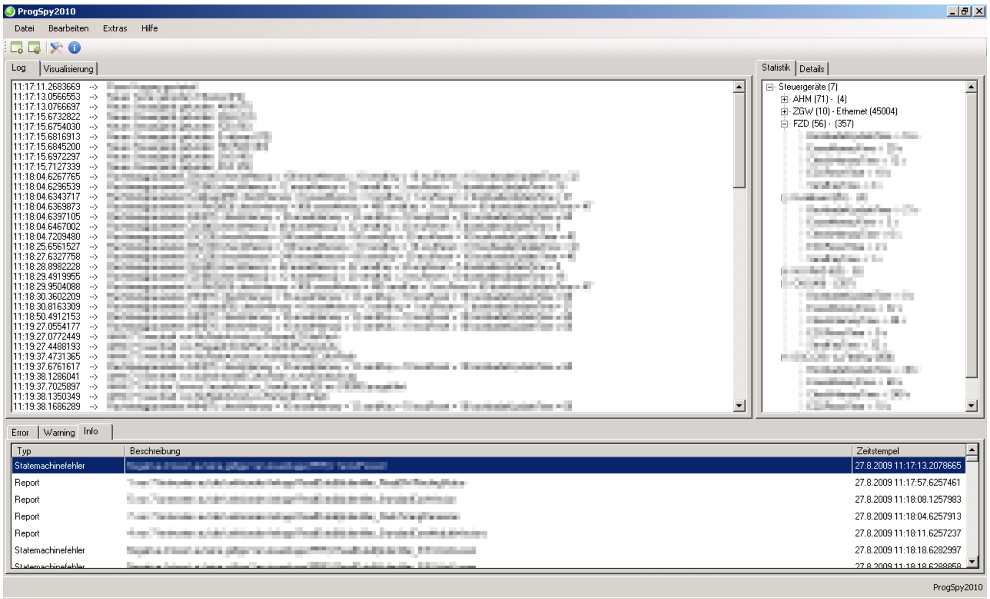
Typ	Beschreibung	Steuergerät	Zeitstempel	Datei
Statemachinefehler	Positive Antwort auf eine ungültige Serviceanfrage (PRR)	0	28.7.2009 09:09:27.8584872	C:\01_Michael_Daten\Hom...
Statemachinefehler	Positive Antwort auf eine ungültige Serviceanfrage (PRR)	0	28.7.2009 09:09:42.9881693	C:\01_Michael_Daten\Hom...
Flashabbruch	Aktiver Flashprozess wurde abgebrochen	37	28.7.2009 09:17:32.6739177	C:\01_Michael_Daten\Hom...
Statemachinefehler	Positive Antwort auf eine ungültige Serviceanfrage (PRR)	37	28.7.2009 09:17:39.7316038	C:\01_Michael_Daten\Hom...
Flashabbruch	Aktiver Flashprozess wurde abgebrochen	29	28.7.2009 09:18:42.4997103	C:\01_Michael_Daten\Hom...
Statemachinefehler	Positive Antwort auf eine ungültige Serviceanfrage (PRR)	37	28.7.2009 09:40:14.0806182	C:\01_Michael_Daten\Hom...
Statemachinefehler	Positive Antwort auf eine ungültige Serviceanfrage (PRR)	37	28.7.2009 09:40:21.7351711	C:\01_Michael_Daten\Hom...

Figure 5.12: Screenshot of ProgSpy2010 with the visualization tabs opened (Some details have been blurred due to IPR restrictions).

5.5.2 Design: From Cardiogram to ProgSpy2010

Based on our pre-design study participants' feedback and on our lead user's advice, we geared our approach towards the traditional version of ProgSpy. We provided a central view showing a list of all diagnosis messages and retained the additional views for statistical information, errors and warnings. However, as an alternative to the central list representation, we also provided a novel visualization. In doing so, we tried to provide the users with their common tool but, additionally, with the possibility to use the visualization on demand without any extra costs (R-8, R-9).

For ProgSpy's visualization we draw on our experience and design considerations from the Cardiogram project as the main task at hand—i. e., inspecting timing of state changes with possible errors and warnings—is basically the same. The general idea therefore was to allocate each ECU with an additional horizontal time line on which changes between programming phases can be shown. Compared to Cardiogram, we, however, added a few changes to the design in order to address the specific and slightly different requirements of flash analysts. Most importantly, we decided to use color coding to indicate programming phases instead of line plots as in Cardiogram. We had two basic reasons for that: First, using colors might be better for correlating the same programming phases between different ECUs—each ECU runs through the same programming phases but in different timings. Second, flash engineers desired an overview over all ECU's programming phases. Showing an overview with, e. g., 70 ECUs each with 5–10 states in turn would end up in very tiny line plots and we therefore thought that using colors might be better for supporting the flash analysts' overview demands. For the decision which color to take for which programming phase, we contacted our lead user and finally came up with a green color



```
11:17:11.2683669  -->
11:17:13.0566553  -->
11:17:13.0766697  -->
11:17:15.6732822  -->
11:17:15.6754030  -->
11:17:15.6816913  -->
11:17:15.6845200  -->
11:17:15.6972297  -->
11:17:15.7127339  -->
11:18:04.6267765  -->
11:18:04.6296539  -->
11:18:04.6343717  -->
11:18:04.6369873  -->
11:18:04.6397105  -->
11:18:04.6467002  -->
11:18:04.7209480  -->
11:18:25.6561527  -->
11:18:27.6327758  -->
11:18:28.8982228  -->
11:18:29.4919955  -->
11:18:29.9504088  -->
11:18:30.3602209  -->
11:18:30.8163309  -->
11:18:50.4912153  -->
11:19:27.0554177  -->
11:19:27.0772449  -->
11:19:27.4488193  -->
11:19:37.4731365  -->
11:19:37.6761617  -->
11:19:38.1286041  -->
11:19:37.7025897  -->
11:19:38.1350349  -->
11:19:38.1686289  -->
```

Steuergeräte (7)
- AHM (71) - (4)
- ZGW (10) - Ethernet (45004)
- FZD (56) - (357)

Typ	Beschreibung	Zeitstempel
Statemachinefehler		27.8.2009 11:17:13.2078665
Report		27.8.2009 11:17:57.6257461
Report		27.8.2009 11:18:08.1257983
Report		27.8.2009 11:18:04.6257913
Report		27.8.2009 11:18:11.6257237
Statemachinefehler		27.8.2009 11:18:18.6282997

Figure 5.13: Screenshot of ProgSpy2010 with the traditional text-based tabs opened (Details have been blurred due to IPR restrictions).

scale starting with light green for early programming phases and dark green for late phases. We chose green as an indicator for 'Okay', while red and yellow are reserved for error and warnings (see also color coding in Cardiogram). Gray is used for showing idle times, i. e., phases where an ECU is not in any of the active programming phases.

The remaining design aspects are very similar to Cardiogram. ProgSpy2010 is based on a zoomable timeline. Errors and warnings are shown on the ECUs' time lines and additionally on the global timeline at the bottom of the visualization in order to provide a quick overview over all ECUs (R-1). Each ECU is labeled on the left side and gets an additional traffic light visualization which provides a fast indication about the "overall outcome" of this ECU's flash process (R-6). Due to simplicity reasons and time restrictions in the project, we abandoned from retaining Cardiogram's overview timeline, the collaborative note adding feature and the interactive opening and closing of state machine/ECU lines.

Figure 5.13 shows ProgSpy2010 with all tabs opened in that way that it equals the traditional text-based ProgSpy tool. Figure 5.12 shows the visualization tab opened and another tab that provides a legend of our color coding.

5.5.3 Evaluation

To test and evaluate ProgSpy2010's usability and to improve the interface, we iteratively conducted expert reviews with three external testers with an HCI/InfoVis background (1–2 sessions per tester) and with our lead domain expert (four sessions). Each study took approximately one hour and we used note taking for logging the qualitative feedback. During the studies we just instructed our participants to use the tool with several example traces and encouraged them to criticize usability aspects which we then used for further improvements of our tool. In case of the studies with our lead analyst, we used traces from his daily work and additionally focused on gathering examples where he used our visualization to gain novel and helpful insights for his daily work. We especially found two interesting examples where he derived novel insights from the visual data representation and where he immediately could derive advantages for his work. These two examples underline the potential value that the visualization approach we proposed might add to flash analysts' work.

1. Anecdote: Our lead analyst used ProgSpy2010 on a test trace from flashing seven ECUs. He immediately stated that *"everything was okay"* by referring to the traffic light visualizations that showed green lights for all seven ECUs. From our pre-design observations we knew that with the traditional text-based data representation deriving this information took our participants longer as they first had to visually scan through the diagnosis message list **(Potential benefit: time savings)**. Despite this positive top-level information, our lead user started to inspect each ECU's programming phase timings in more detail in the main visualization view. He was surprised about the long gray, idle phases between the green, active programming phases and instantly ascribed this fact to a potential bug in the visualization software. However, by checking the timings from the raw values—what, by the way, took him much longer than deriving it from the visualization—it became clear that the long idle phases were not ascribable to a software bug in our tool but that they really reflected the flash process under inspection. Our lead analyst stated that these uncommonly long idle phases are not specified as erroneous flash processes, however, he mentioned that this behavior must definitely be addressed as it would lead to unintentional extra time for flashing vehicles. He immediately took action, contacted an engineering colleague and triggered an elimination of this unforeseen problem **(Potential benefit: novel insights leading to optimization of flash processes)**.

2. Anecdote: In another session, our lead user brought along several test traces from flashing a complex ECU which always ended up in throwing an exception error. By using our visualization, he understood that every time before the error occurred the ECU in question had been in an incorrect programming phase. He again started switching between the textual representation and the visualization in order to verify the correctness of this novel insight. Finally, he could lead back the error not to incorrect behavior in the ECU–as initially accepted—but to an incorrect specification of the flash process for this particular ECU **(Potential benefit: novel insights revealing errors in the flash specifications)**.

5.5.4 Discussion and Adoption

ProgSpy2010 is a tool for supporting flash analysts working with specific pre-filtered traces that are used for diagnosing ECU software deployment (flash processes). ProgSpy2010 is the direct successor of a former in-house analysis tool called ProgSpy and has recently superseded it. This means that our tool is now used by a group of approximately 10–15 engineers in their daily work.

According to our target users' demands, we initially implemented a text-based tool very similar to its predecessor ProgSpy. While retaining benefits and known features definitely increased the adoption of our tool, we also tried to carefully integrate an additional, easy usable and understandable visualization approach. We particularly paid attention not to impose the visualization to our users but rather to provide an visualization feature which they can use on demand but which they can also ignore if not required—the visualization can even be switched off entirely. Two examples that we gathered from observing our lead analyst using the tool, however, concretely showed how our visualization approach led to novel insights and how these insights in turn were used for optimizing flash processes or readjusting specifications. These examples also drastically changed our lead users mind from 'visualization as a nice-to-have feature' to 'visualization as a valuable and powerful tool for his daily working practices'. He was really impressed how the visualization revealed novel and important insights that had not been detected over weeks before and even did not believe them in the beginning—*"that must be a visualization bug"* (see above). Based on his positive experience, he also recommended ProgSpy2010's visualization to colleagues and several of them started using it on an occasional and situational basis (based on our last information, the visualization is used by five more experts now).

Due to time restrictions and a non-exclusive visualization focus in this project, the current visualization of ProgSpy2010 is very simple and we definitely think that enrichments and extensions can add further value. As a starting point, we would recommend to investigate how the features we used in Cardiogram could be adapted as we made good experience with them (e. g., overview timeline, extended interactivity, collaborative note taking, etc., cf. Section 5.4.2 for more information). For the same reasons, our current studies were limited in time and participants. We hope that future researchers might use the opportunity we established, extend the tool and do more in-depth research. In conclusion, we hope that ProgSpy2010 provide other researchers, designers and practitioners with a helpful example showing (a) how visualization can quickly and successfully be introduced in projects where a novel software tool has to be built anyway and (b) how even simple visualization approaches can add value to end users' work by gaining novel insights.

5.6 Summary

In this chapter, I discussed and exemplified how visual analytics technologies can be used to improve in-car network diagnosis. I introduced a set of eleven design recommendations that we derived from intensive ethnographic field studies and from evaluating an early design prototype, VisTra, which is also presented in this chapter. Based on these recommendations, we built two visual analytics Carmen modules for FIT trace analysis (AutobahnVis, Cardiogram) and one tool for flash analysts (ProgSpy2010), closely installed them in our target users' working environment and evaluated them under real circumstances revealing valuable information on the performance of the tools in real world analysis situations.

Overall, the recommendations from our exploratory field analysis together with a participatory design process helped us to design tools that found their way into everyday routines of our analysis experts. Along with the strong user integration throughout the entire design process (see also previous chapter), I think that the core ingredients to a successful deployment of our visual analytics techniques supporting trace analysts in particular were the visualizations' simplicity, and a tight integration into existing tools and workflows.

Our target users had demanded tools that *"simplify [my] work, not complicate it with intricate visualizations that have to be learned upfront."* Preferred were solutions with high **automation and simple, easy to understand representations** with an immediately apparent benefit that were explicitly tailored to their needs. Therefore, we found it important (a) to start with easy to understand solutions, (b) to iteratively extend them, and (c) to verify the value of a novel approach as soon as possible (e.g., through prototyping). This approach also generally integrated very well with the modular working practice of automotive engineers.

We also learned that in the domain of trace analysis **tight integration** of final tools with domain data and process is a crucial factor to success, adoption and an essential requirement for better understanding the value of our tools. Evaluating earlier prototypes that have not been integrated into daily working practices (such as VisTra) had always been restricted to expert estimation within mostly artificial conditions. Indeed, this provided valuable and helpful feedback and also let us better understand the recommendations we outlined in Section 5.1.4, however, it did not help us to gain insight into the long term nature of analysis processes. Our new approach to providing visual analytics solution in this domain, however, lead to new and extended insights and better adoption rates. Most of our test engineers continued using our tools AutobahnVis, Cardiogram and ProgSpy2010 after the evaluation phase was completed and to some extent also recommended them to their colleagues (for numbers and a detailed description see Section 5.3.4, 5.4.4 and 5.5.4).

All FIT analysis design studies introduced in this chapter, VisTra, AutobahnVis and Cardiogram, were designed for trace file analysis. A general limitation of these approaches is the fact that we solely "observe" the bus-systems. We do not allow engineers to actively simulate communication processes or imitate erroneous behavior by adding extra information to a bus system, as it is, for instance, conducted during flashing ECUs (cf. ProgSpy). For all our solutions it additionally holds true that correlations between mechanical and electronic information, dependencies between software components or inner-ECU behavior cannot always be analyzed by solely looking at trace files. In order to represent such aspects new modules must be made available that allow the analysis of additional types of datasets. One of the current obstacles to adding this support, however, is the fact that much of this data is widely distributed over manufacturers' suppliers who implement the software for ECUs. Security policies and protectionism then lead to black boxes in the vehicle's electronic system.

Chapter 6

Visualizing In-car Communication in 3d

The previous two chapters focused on the first two challenges outlined in Section 3.6.1 and introduced problem-driven design studies for visualizing specification data for development engineers (Chapter 4) and trace data for test engineers (Chapter 5). In this final design study chapter, I describe our[1] technique-driven research addressing the third and last focus area, namely ‘visualizing in-car network data with 3d models (cf. Section 3.6.1).

Already outlined in Section 3.4.2 and 3.5.4, the technique of 3d visualization[2]—i. e., the usage of virtual, semi-transparent 3d models for representing in-car communication—is highly demanded by our target users and stakeholders and great potentials are ascribed in terms of better understanding correlations between mechanical components and electronic communication. Initially, I—as an InfoVis researcher—was skeptical about using 3d techniques as they pose additional obstacles such as information occlusion or high navigation costs, and usually are only justifiable by an intrinsic and important role of 3-dimensional spatiality as it is, for instance, given in SciVis applications (cf., e. g., Ware’s discussion about 2.05-d representations where he argues that the actual added value of a third dimension is worth 0.05 of the other two [War08], or Smallman et al.’s comparison of 2d and 3d representations for air-traffic visualization [SCP95]). Despite my initial concerns, however, I decided to investigate this area for several solid reasons:

Intrinsic spatiality: First, in-car communication data inherits an intrinsic spatiality due to its implicit connection to the real network installed in vehicles and we do not yet completely understand if and how visualizing these spatial aspects can contribute to automotive electronic development and testing.

---

[1] If not specified otherwise, the use of “we” in this chapter refers to Michael Sedlmair, Fabian Hennicke and Alexander Kahl.

[2] For simplicity reasons, in the following the general term 3d visualization is used for referring to visualization approaches where a 3d model of a virtual vehicle entity is utilized for representing the data on a 2d screen.

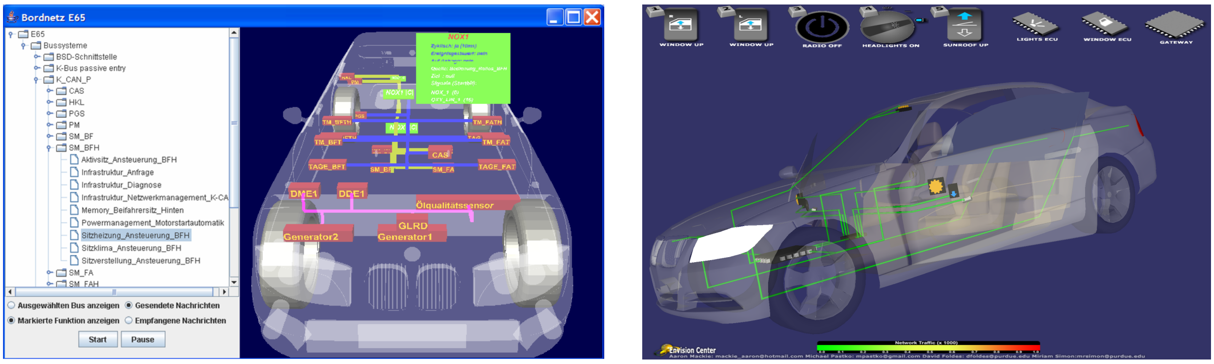
Very strong interest: Second, in line with the converging focus group's outcomes (cf. Section 3.5.4), all along the three years I worked at the automotive company, I very frequently encountered electronic engineers having strong interest in and estimating high potentials to 3d visualization of in-car network communication. While certainly technical fascination and a strong affinity to success stories from automotive SciVis and CAD applications have major impact on their interests and estimations (cf. Section 3.4), we do not know whether there are really valuable application areas for 3d visualization in the automotive electronic domain or whether the estimations solely rely on fascination. In case there are potential application areas, we furthermore know little about how a 3d visualization should be designed in order to support engineers with their tasks.

Familiar mental model: Third, a 3d representation of the vehicle is definitely a familiar and meaningful view for automotive engineers and we currently do not know if the visualization of in-car communication processes benefit by matching this mental model.

Funding: Last but not least, the funding of this thesis played an important role for the decision of researching on 3d visualization approaches as it was part of the contract.

To address the questions stated above, in this chapter, I present four prototypes using a (semi-transparent) 3d model. Two of them (early prototypes) have been already implemented by or for BMW at the time this thesis started. These prototypes were neither designed for a particular target group nor properly evaluated with domain experts. Therefore our first step was evaluating them and learning about several considerations important to address when designing 3d visualizations in our area. To have a more concrete focus for our own work and to gather explicit feedback from potential end users, we referred to our various collaborations with automotive engineers, and focused on two specific user groups, FIT analysts and network architects. We designed and implemented two further prototypes based on the considerations we derived from studying the early solutions, and evaluated them with domain experts subsequently. Our first approach, Autobahn3D, is based on enriching abstract multiple coordinated view representations with an additional 3d model view in order to allow the user understanding correlations between electronic communication processes and mechanical behavior. The second prototype, Car-x-ray, looks at how abstract data and spatial information can be visualized in one single view. We implemented two case studies with Car-x-ray and tested each of them with domain experts. First, we showed how our concept can be used to visualize communication paths and transitive chains (cf., for instance, WiKeVis and VisTra) within a semi-transparent 3d model. With our second case study, we introduced an approach for visualizing spatiality in early network specifications to support network architects in partitioning the functional to the physical network.

The early prototypes, the user evaluation we conducted with them and the design considerations we derived are presented in Section 6.1. Section 6.2 introduces and discusses Autobahn3D and Section 6.3 presents Car-x-ray. The chapter ends with a summary and a discussion of our findings.



(a) (b)

Figure 6.1: Screenshots of early 3d prototypes: **(a)** PT1 visualizing animated messages boxes "moving" through a virtual car [Moz07]; and **(b)** PT2 shows signal animations, color codes ECU-load and animates mechanical behavior such as moving a window [PMS07].

6.1 Pre-Considerations

In this section, I briefly introduce two prototypes that have been implemented at or for BMW before my thesis had started. We evaluated these prototypes and used them for triggering further discussions with a variety of automotive electronic engineers, mostly test engineers. I discuss the results and then outline several considerations which we derived by our studies and which should be taken into account when designing a 3d visualization for visualizing in-car communication networks—and might also provide valuable guidance for other application areas.

6.1.1 Two Early Prototypes

Figure 6.1-a and b show screenshots of the two prototypes we found already implemented at the start of this thesis (in the following simply referred as PT1 and PT2). PT1 [Moz07] is based on two views, a semitransparent 3d model of a vehicle that can be zoomed, panned and rotated by the user, and a hierarchically ordered list showing bus systems, ECUs and messages. The user starts by selecting one or more bus systems from the list in order to display them schematically in the 3d model. All ECUs connected to the selected bus systems are represented as labeled, red boxes, and the bus systems themselves are schematically drawn via orthogonal, color-coded lines. Subsequently, the user can select a message from the list view and in doing so initiates an animation of small, green message rectangles sent from one or several sender-ECUs (usually one) to one or several receiver-ECUs (usually several), resulting in one or several green message icons "moving" over the bus system. Clicking on such a moving message box, opens an additional, also green and moving, box showing the message's signals (cf. Figure 6.1-a). In doing so, the designers of PT1 wanted to provide a better understanding of how messages and signals distribute in the car's network. The data visualized in PT1 was an export from the database

BNE. Positioning of ECUs and bus systems was done manually based on automotive background knowledge (ECUs) and aesthetic layout criteria for the bus systems.

The second prototype we found already existing, PT2 [PMS07], is also based on a navigatable, semi-transparent 3d vehicle model and on a schematic representation of some ECUs, sensors, actuators and connection lines (cf. Figure 6.1-b). By clicking on one of five predefined buttons above the 3d model (excluded representatives for window/sunroof regulators, and switches for radio and lights), the user initiates an animation of a virtual signal icon that moves through the 3d model. This signal icon moves from a single initiating sensor, e. g., the window regulator button, over one or several bus systems passing through gateways where it pauses for a second, to a single actuator where it triggers some mechanical activity, e. g., opening/closing the window. Compared to PT1, each user interaction initiates only the animation of one single element, a single signal that follows a specified path in the network. At the time this animated signal element passes through a gateway, the gateway ECU object is highlighted in green as long as the signal remains in this object (indicating low load). By concurrently triggering more than one signal, i. e., clicking another button during an animation is still going on, the user can create situations where multiple signal representatives "pass through" a gateway. Concurrent transit of two signals results in yellow color coding of the gateway ECU (low–medium load), three in orange (medium–high load) and more than three at a time results in red brushing (indicating high load). Compared to PT1, PT2, however, does not rely on any real data, nor are realistic network traffic simulations used. All data is fictive and hardcoded. Therefore, the tool can just be seen as an example for different visualization techniques that may potentially be used rather than a visualization system where real data can be loaded and represented. The physical network is a minimal example of 3 ECUs and 3 actuators, the connection lines' layout is based on visualization aesthetic criteria and the signal paths are chosen arbitrary and simplified (Just 1:1 communication instead of usual 1:n communication, the signal path shown is not the real path). Nevertheless, the authors of this tool claimed statistical evidence based on a user test with 30 students, that – compared to the BNE—their visualization "definitely improve[s] perception of the information being presented" [PMS07].

6.1.2 Evaluation and Discussion

Rather than showing statistical evidence, we used these tools for conducting several qualitative user studies with automotive engineers in order (a) to better understand the potential domain value 3d visualizations can add, (b) to trigger new perspectives about utility as well as usability by letting engineers use the tools—again: what people do often differs from what they say or think—and (c) in doing so to learn about design criteria that are important to take into account for future developments. We did formal think-aloud user studies with three development and four analysis engineers. To better understand the tools potentials, we let our participants conduct a set of loosely defined tasks related to understanding spatial correlations, to communication between mechanical components and electronic information, and some just for exploring each tool's features. During and after the participants conducted these tasks, we encouraged discussions about their estimations as well as positive and negative critique. Similar to VisTra (cf. Section 5.2), we also used these prototypes for additional informal user studies and in doing so reached approximately 15 more engineers. If possible, we let the engineers also conduct several of the tasks we tested in the formal studies, or otherwise just let them try out the tools on their own and discussed it along the way. Our findings can be summarized as follows:

Usability of the Tools The general feedback by using and discussing PT1 and PT2 with our participants was quite good, however, indicated as we hypothesized a strong affinity to fascination underlined by statements such as *"it looks just good"* or *"that's cool and fun"*. By observing our users conducting the tasks and using the early prototypes, we encountered on the other hand several problems that could be set in correlation to recent findings from literature.

First, we observed our participants having several difficulties with representing messages or signals as animated icons "moving through" the virtual car. While these animations in general were often evaluated to be *"good looking"*, *"self-evident"* or *"well suited for presentation communication processes in the car"*, our participants, however, had problems in deriving sender-receiver correlations especially if there were more than one animated message/signal at a time. Furthermore our participants complained that for daily work sequentially following animations signal by signal would be tedious (cf. Ware et al.'s project on visualizing underwater behavior of whales [WAPW06]). Second, we frequently observed problems in reading PT1's labels of ECUs due to text-distortion, and for messages and signal text boxes additionally due to moving targets (cf., for instance, Grossman et al.'s findings on text readability in 3d [GWB07]). Third, for PT2 we found that our participants in several cases missed mechanical behavior animations such as moving a window which could be either a result of occlusion and/or of inattentional blindness [Mac03]. On the upside, a majority of our participants liked the possibility of filtering the in-car network components in PT1 and we encountered no major problems with navigating the 3d-models in both prototypes. The opinions about the degree of abstraction in represenating the in-car network diverged. While in general all engineers agreed in trying to be realistic in positioning ECUs and bus systems is invaluable, some argued for having just abstracted boxes representing ECUs and orthogonal lines for bus systems due to clarity reasons, others, however, argued for an exact positioning and layout of all hardware and wiring components to get valuable information from the visualization. All participants again agreed that filtering irrelevant mechanical components such as screws is definitively valuable and necessary, as a fully equipped car with all its *"nearly 10,000 components [...] makes it impossible to distinguish between relevant and irrelevant information"*. However, which mechanical components are relevant and which not heavily relies on the task at hand.

Estimated Utility of the Tools For all participants it was clear that both tools as they stand are not applicable for their daily working practice. The obvious reasons are the usability shortcomings of PT1 and the fact that PT2 does not rely on any real data. However, beyond that we discussed with engineers much about potentials and prospective use cases which such or similar solutions could support. These discussions revealed that most of our engineers estimated 3d visualizations good for communication (especially to their bosses), presentations and education, because they *"represent in-car communication in an intuitive way"*. Asking if such a tool would be a valuable helper for their own daily tasks we got twofold answers. The first half argued that they would rather stay with abstract representations and that they do not see any obvious benefits beyond the use cases named above. The other group argued that carefully adapting and extending the designs presented indeed would add value to their daily work. For using such tools beyond the use cases mentioned above, however, several aspects have to be reconsidered:

Scalability to real data: Both tools show extremely slowed down animations of one or several messages/signals through the system. While for education and presentation purposes this extremely downscaled version might be useful, for engineering purposes 3d visualizations must scale to real datasets and show them task-centered in an efficient way. This raises the question of how much information reasonably can be visualized in a 3d model view. Obviously, showing all messages in real-time, i. e., up to 15,000 per second is useless as it will not be perceptible, the same holds true for showing entire specification documents (e. g., all specified dependencies).

Meaningful ex- and abstractions of the data: Due to the restricted scalability, meaningful ex- or abstractions of the information have to be found. These ex-/abstractions one the one hand must be insusceptible against the restriction posed by a 3d representation but on the other hand provide enough spatial correlation to be worth representing in 3d. We collaboratively with our participants identified and discussed three scenarios where a 3d model view might add (more or less) additional value to engineering work:

1. *Signal and message path:* In line with the two early prototypes, several of our participants argued that it might be interesting for their work to see how information "moves" through the network. Scenarios mentioned were, for instance, detecting external error sources for analysts, e. g., a screw driven through a cable leading to communication breakdown, or to optimize system layout and improving reliability by additionally considering spatiality aspects, e. g., for deciding about alternative communication paths for crash safety communication.

2. *Electronic communication and mechanical behavior:* In line with PT2, our participants argued that for several tasks it is definitely important to understand cross-correlations between mechanical behavior of the car and the electronic communication. For trace analysts, for instance, it can be beneficial to see the real context of the test drive. For this purpose, today there are some tools available that allow for closely integrating video-taping with trace analysis. However, these recordings are usually restricted to one or several specific camera positions and it is also not possible to "look into" the vehicle as it would be possible with a virtual model. On the other hand, development engineers frequently mentioned simulation and virtual prototyping of vehicle electronics. The vision behind this idea is to start much earlier in the process of developing vehicle electronic to test its interactivity with mechanical components. Similar to mechanical simulations such as CFD (cf. Section 3.4.1) a 3d model could potentially serve as virtual platform for conducting test drives and to analyze the results subsequently.

3. *Working load:* This topic was discussed controversial. While several engineers quite liked the idea of representing working within the vehicle, others (including us) did not find any additional value which showing the load of ECUs and/or bus systems in its spatial context might add compared to a simple and abstract 2d representation (e. g., using a topology map). One of our participants stated, for instance: *"I wonder what additional benefit I will have by showing this in the 3d-model instead of, for instance, just using a topology map in the same way [i. e., color coding ECUs as in PT2]"*.

Missing correlation to other tools/views: As still large parts of the electronic data is abstract (correlations, timings, etc.) nearly all our participants argued that a prospective 3d tool work definitely must be combined with other text-based or abstract, traditional or novel representation techniques. Two of our trace analysts could well imagine to integrate and combine such a view, for instance, with Carmen (assumed all technical restrictions have been overcome before, see below).

6.1.3 Design Considerations

Based on the evaluation and discussion of the two early prototypes, we formulated a set of design considerations that tool designers, researchers but also decision makers can take into account when they think about 3d visualization in this or similar domains. For each consideration we provide a little discussion reflecting our thoughts about what might be critical points to address and important aspects to include. We divided our considerations in two categories, (1) *pre-design*, i. e., what should be considered before starting a 3d-visualization project, and (2) *design*, i. e., thoughts about what might be important when designing and realizing a concept. The considerations helped us for our own 3d studies and we hope that sharing our experience also provides other designers/researchers with valuable information and new inspiration.

Pre-Design Considerations Before deciding to visualize in-car network communication in 3d the following questions should be addressed, clarified and decided whether 3d is the preferable solution or not.

DC-1: Data Visualization or Aesthetic Concept Communication
The most important question to address is "What and who are you designing for?", i. e., what is the use case you want to support and who are your target users. Our studies with the early prototypes revealed that most of the participants considered a 3d representation good for communication, presentation and education rather than for *"serious engineering work"* as the data coded in the representations hardly provides any new insights into the data (PT1) or is completely detached from real data (PT2). While these soft-use cases are definitely valuable application areas, from a data visualization point of view, however, we have to ask how we can harness a 3d representation to amplify cognition and gain novel insights. As this thesis deals with information and data visualization all following considerations as well as the prototypes I will present focus on data representation rather than on aesthetic concept illustrations.

DC-2: Spatiality Counts
For data visualization purposes, the question about the importance of 3-dimensional spatiality must be carefully considered. Various examples showed that simple 2d visualizations can outperform a fancy 3d visualization due to benefits in navigation, information occlusion and overview aspects (see, for instance, van Wijk's and van Selow's Calendar View [VWVS99], or Smallman et al.'s air-traffic visualization [SJOC01]). For in-car communication networks these can, for instance, be abstract visualizations, e. g., using topology maps, or simple plan views of a vehicle.

DC-3: Paths and Mechanical Behavior
In discussions with our study participants we found that particularly two use cases for using 3d model visualization were frequently mentioned and estimated to be valuable for daily engineering work: (a) showing information paths "through" the vehicle, e. g., for better understanding environmental aspects in analysis or involve spatiality into designing and developing communication specifications; and (b) the visualization of correlations between electronic messaging and mechanical behavior, e. g., for virtual prototyping purposes. While these use cases appeared with our studies, there certainly are also other reasonable application areas. In this case, however, DC-2 should carefully be considered.

Design Considerations With the early prototypes we identified several design drawbacks that we summarize at this point. In doing so, we want to provide other designers help in addressing them in their tools and in finding further literature discussing similar aspects.

DC-4: Carefully Consider Animations of Messages and Signals
The metaphor of animated messages and/or signals "moving" over the bus systems is self-evident and might be reasonable for presentation, communication and education purposes. However, to code information relevant for engineers' daily work such animations come with two basic drawbacks. First, following animations cost time, but engineers' time is usually rare and valuable. Second, our studies indicated that following a path by reading an animation causes problems in recognizing and/or remembering the path. The problem escalates when more than one message at a time is "sent" over the network. In their whale behavior project, Ware et al. had similar findings regarding animated paths and suggested rather than mapping temporal behavior on time to map it on spatial patterns [WAPW06].

DC-5: Additionally Highlight (Mechanical) Behavior
By observing our participants conducting tasks, we also found that mechanical behavior was missed due to occlusion or inattentional blindness. There have been much work done on how to stress elements in 3d scenes which can be utilized for this purpose, such as simply coloring animated elements, using interactive shadows [HZR$^+$92], or reducing transparency of adjacent elements [VCWP96, KSW06] (see also Elmqvist and Tsigas' taxonomy of 3d occlusion management techniques [ET08] and the 3d highlighting technique overview by Preim and Ritter [PR02]).

DC-6: Different Perspectives and Automatic Camera Planning
Navigating through a 3d model can be fun but for time-critical tasks in the engineering domain navigating a 3d model can pose additional and annoying work. Providing shortcuts for pre- or

user-defined perspectives on the model and/or techniques for automatic camera planning can provide support for fast navigation (see, for instance, Christie et al.'s work, [CON08]).

DC-7: Filtering Mechanical Components
Modern vehicles have up to 10,000 single mechanical components. Which of these components are necessary and valuable to show, and whether interactive filtering of the components is useful or not, has to be decided according to the task at hand.

DC-8: Real Positions, Abstracted Forms
For representing the 3d model components the designer can either try to be as close as possible to reality in order to ideally capitalize spatiality—the 3d models basic strength—or to abstract elements in favor of clarity and/or better recognition of important elements (e. g., ECUs, bus systems). While our studies indicated, that engineers found it important to have "real" positions of ECUs and bus systems (real wiring paths), most of them argued that having abstract designs for components, e. g., cuboids for ECUs and lines for bus systems, is fine or even preferred. When thinking about adding labels to real or abstracted components, Grossman et al.'s findings [GWB07] should be taken into account and distorted labels should be avoided. Finally, the degree of necessary realism and useful abstraction usually varies from task to task and therefore should be clarified after consulting target domain experts.

DC-9: Combining 3d with other Representation Techniques
Nearly all engineers in our studies mentioned that a reasonable utilization of 3d models must be combined with other representation techniques such as lists or abstract visualizations. While for education and presentation purposes, a full-screen, immersive 3d vehicle—or even a stereoscopic representation—might be eligible, 3d visualizations will most probably not entirely replace current working practices. A useful 3d visualization therefore should be combined and coordinated with current or other techniques presenting the data (see also recommendation R-4 in Section 5.1.4). This strategy might also help to counterbalance restrictions of a 3d visualization (occlusion, additional navigation) by allowing the users to work as usual and just refer to spatial information on demand when it is necessary. A good example showing the value of combined abstract/3d visualization approaches can be found in Doleisch et al. [DMG$^+$04].

6.2 *Autobahn3D*: 3d and MCV

The basic idea of the prototype presented in this section is strongly orientated on the idea of using a 3d view embedded into a multiple coordinated view system rather than having a stand-alone application (cf. DC-9). For this purpose, we[3] designed and implemented a freely configurable *3d-Vehicle View* that can be used and controlled by other applications. To further illustrate our ideas, we implemented a case study where we combined the 3d-Vehicle View with our tool AutobahnVis (cf. Section 5.3). We evaluated and intensively discussed our case study with

---

[3] Portions of this section have been published in [SRH$^+$09]. Thus, any use of "we" in this section refers to Michael Sedlmair, Kerstin Ruhland, Fabian Hennecke, Andreas Butz, Susan Bioletti, and Carol O'Sullivan.

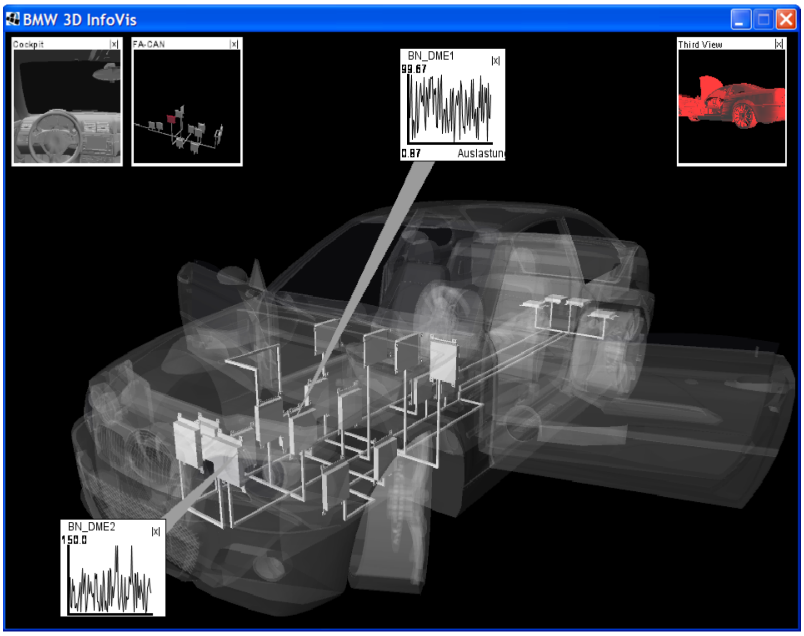


Figure 6.2: Screenshot of our freely configurable 3d-Vehicle View showing. Along with the main view several miniature views are shown providing either alternative perspectives or extra information.

domain experts and found several examples how engineers might profit by linking an additional 3d view to (traditional) abstract representations. In the following I describe the design of our 3d-Vehicle View, the case study Autobahn3D, show how the considerations presented in the last section influenced our decisions and finally discuss the results of our user studies.

6.2.1 A Remote-Controllable 3d View

Our first intention in this project was to find an adequate 3d vehicle model that on the one hand provides enough details to meet reality requirements (DC-8) and on the other hand is still manageable in size and complexity for realizing our prototype without investing too much energy in low-level performance aspects. To do so, we initially contacted mechanical engineers asking for CAD models. Due to information security reasons, though we were not allowed to use these models for our purpose. Next, we contacted a group of designers at the company and found them working with highly detailed, computing intensive 3d models. However, the major with these models turned out to be the absence of inner components—including ECUs and bus systems, the most relevant information for us. Similar to the two early prototypes described above, we therefore were forced to draw on freely available 3d vehicle models and on manually re-constructing the in-car network based on the information we could derive (a) from BNE for rough positions of ECUs, and (b) domain experts' knowledge about wiring information. While we tried to position elements as realistic as possible, we abstracted the forms, i. e., simplified, typical ECU representations and pipes for bus systems (DC-8).

The resulting 3d-Vehicle View is available in a separate application window and can be navigated by the user with conventional first-person 6-DOF-navigation techniques (cf. Figure 6.2). We provide a client-server communication interface (3d-Vehicle View is the client) that has to be implemented by the application that wants to communicate with the 3d-Vehicle View. The application then can send instructions to the 3d-Vehicle View and in doing so remote-control its parameters. The remote-controllable parameters are:

1. *Animating component behavior:* Similar to PT2, we manually defined a set of 74 animatable components—compared to seven in PT2—reproducing mechanical vehicle behavior, such as window moving, doors opening/closing, steering and turning wheels, or turning on/off lights (DC-3). Other applications can control these animations either by directly addressing components or by semantic instructions, e. g., ‘steering x degrees left’ resulting in turning both front wheels as well as the steering wheel.

2. *Controlling colors and transparency:* All mechanical and electronic components can be adjusted regarding their transparency and color. This allows for setting the car body’s transparency as required, for filtering elements by setting their transparency to 100% (DC-7) and for highlighting elements via color coding (e. g., DC-5).

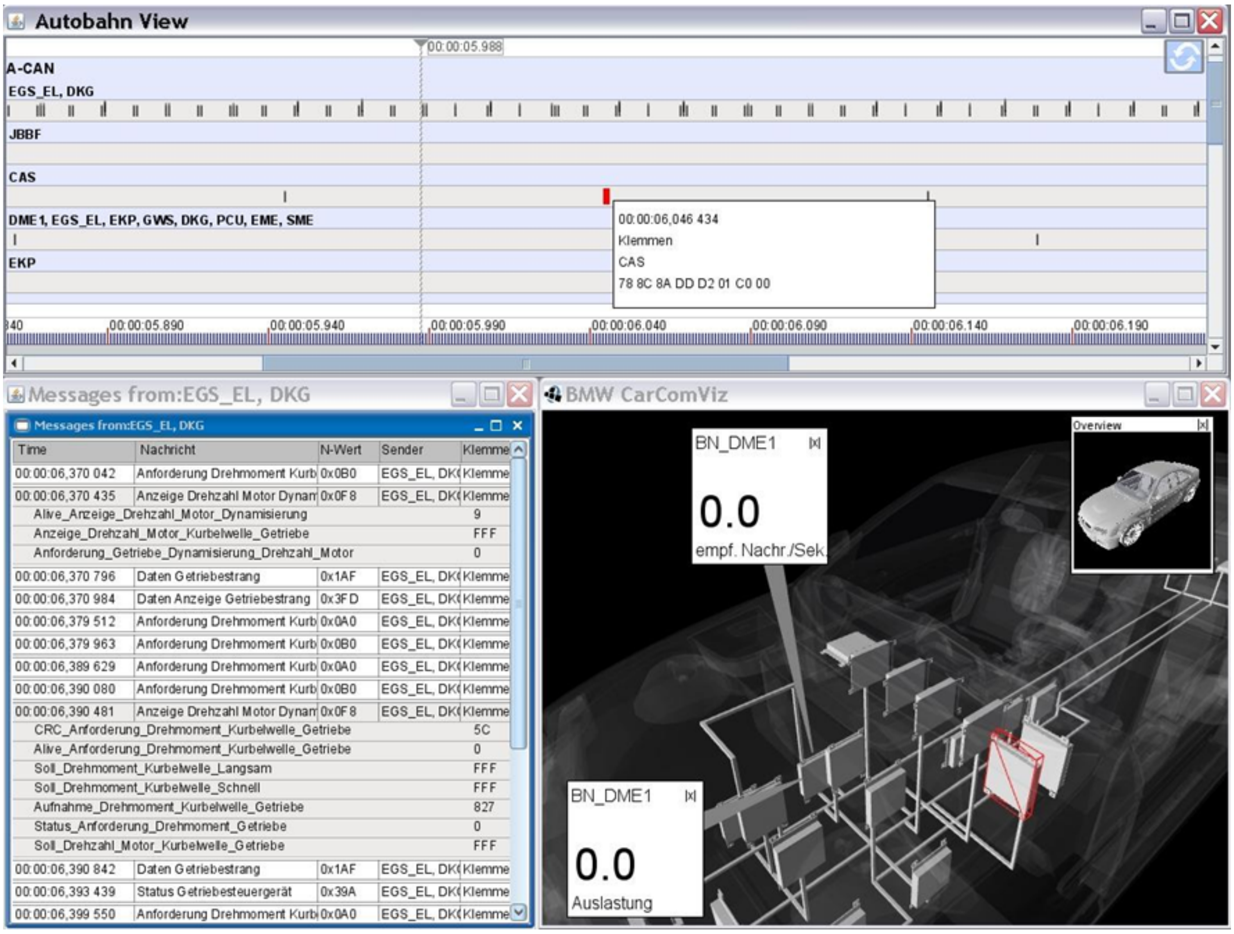
3. *Additional miniature views*: Our 3d-Vehicle View provides the opportunity to show additional views in a PIP-manner (picture in picture, see also [SCP95]). There are two types of additional views: First, the user can specify *various perspectives* of the 3d model into different views (DC-6) and by clicking on them switch the perspective in the main view. The remote-controlling application then can either influence all views at once (e. g., animating a door opening in all views) or separated (e. g., just adapt the transparency in the current perspective). Second, we integrated additional abstract *information boxes*. These boxes can, for instance, be used to show the load of an ECU or the driving speed of a wheel. They are billboarded to the 2d camera plane (cf. [GWB07]) and are connected via semi-transparent arrows to the object they are specified with. The additional information can either be shown in numbers or in a simple plot. In order to avoid unintentional occlusion, all miniature views can be freely drag-and-dropped by the user.

6.2.2 Case Study: Autobahn3D

In a next step, we used this 3d-Vehicle View in a case study to show how it can be combined and coordinated with other views (DC-9). For this purpose, we used an earlier version of AutobahnVis[4] (cf. Section 5.3) and visualized trace files in our case study (DC-1). In our case study Autobahn3D, we added the 3d-Vehicle View as an additional view to Autobahn View and Message View and prototypically implemented two coordination techniques:

---

[4] We decided to use AutobahnVis rather than other tools as we had full access to the source code without organizational and technical restrictions.



00:00:06,046 434
Klemmen
CAS
78 8C 8A DD D2 01 C0 00

Messages from:EGS_EL, DKG

Time	Nachricht	N-Wert	Sender	Klemme
00:00:06,370 042	Anforderung Drehmoment Kurb	0x0B0	EGS_EL, DK	Klemme
00:00:06,370 435	Anzeige Drehzahl Motor Dynam	0x0F8	EGS_EL, DK	Klemme
Alive_Anzeige_Drehzahl_Motor_Dynamisierung				9
Anzeige_Drehzahl_Motor_Kurbelwelle_Getriebe				FFF
Anforderung_Getriebe_Dynamisierung_Drehzahl_Motor				0
00:00:06,370 796	Daten Getriebestrang	0x1AF	EGS_EL, DK	Klemme
00:00:06,370 984	Daten Anzeige Getriebestrang	0x3FD	EGS_EL, DK	Klemme
00:00:06,379 512	Anforderung Drehmoment Kurb	0x0A0	EGS_EL, DK	Klemme
00:00:06,379 963	Anforderung Drehmoment Kurb	0x0B0	EGS_EL, DK	Klemme
00:00:06,389 629	Anforderung Drehmoment Kurb	0x0A0	EGS_EL, DK	Klemme
00:00:06,390 080	Anforderung Drehmoment Kurb	0x0B0	EGS_EL, DK	Klemme
00:00:06,390 481	Anzeige Drehzahl Motor Dynam	0x0F8	EGS_EL, DK	Klemme
CRC_Anforderung_Drehmoment_Kurbelwelle_Getriebe				5C
Alive_Anforderung_Drehmoment_Kurbelwelle_Getriebe				0
Soll_Drehmoment_Kurbelwelle_Langsam				FFF
Soll_Drehmoment_Kurbelwelle_Schnell				FFF
Aufnahme_Drehmoment_Kurbelwelle_Getriebe				827
Status_Anforderung_Drehmoment_Getriebe				0
Soll_Drehzahl_Motor_Kurbelwelle_Getriebe				FFF
00:00:06,390 842	Daten Getriebestrang	0x1AF	EGS_EL, DK	Klemme
00:00:06,393 439	Status Getriebesteuergerät	0x39A	EGS_EL, DK	Klemme
00:00:06,399 550	Anforderung Drehmoment Kurb	0x0A0	EGS_EL, DK	Klemme

Figure 6.3: The Autobahn3D prototype connected to a former version of the AutobahnVis tool (cf. Section 5.3.3): **(a)** Autobahn View; **(b)** Message View; and **(c)** the additionally added 3d-Vehicle View.

1. *Linking and Brushing*: Hovering over a message in the Autobahn or Message View highlights the sending ECU[5] in the 3d-Vehicle View and reveals it installation position in the vehicle (DC-2, cf. also [DGH03])

2. *Semantic Linking*: Navigating, i. e., panning, over time in the Autobahn View results in replaying mechanical behavior in the 3d-Vehicle View (DC-3). For this purpose, we added a dashed, vertical line in the Autobahn View indicating an accurate point in time that is used for coordinating and replaying behavior in the 3d model.

In addition, we provided a setup window where the user can manually configure all components transparencies, colors but also set up miniature and info-box views. Figure 6.3 shows a screenshot of our prototype Autobahn3D.

---

[5] The earlier version of AutobahnVis used interpreted traces. While this restricted scalability and hindered integration into engineers working practices, it allowed us for interpreting messages in terms of sending ECUS and therefore to show the messages on “ECU-lanes” instead of “bus-lanes”.

6.2.3 Evaluation

We qualitatively evaluated our case study Autobahn3D in terms of potential domain value with four analysis experts (tool-tests with four predefined tasks plus think-aloud, interviews) and additional informal discussions with approximately 15–20 automotive engineers (either group-presentation and discussion, or tool-test plus interview). In doing so, we gathered several examples where engineers evaluated Autobahn3D potentially adding value to their daily work practices. Due to incomplete data restrictions (see below), we had to manually prepare a test dataset and therefore could not study Autobahn3D under realistic circumstances integrated into daily working practices of engineers. Hence, the examples presented in the following are rather engineers' estimations on the concept's and tool's capabilities than real usage examples as provided, for instance, in Section 4.3.3, 5.3.2, or 5.4.3.

Detection of complex errors with mechanical/electric correlations: In Section 5.1.3, I outlined that one reason of error complexity is the difficultly to identify correlations between vehicle's mechanical behavior and electronic communication. In this section, I also introduced an example error where it took engineers several weeks to identify the error's source resulting from simultaneously slamming all four doors. Two of our participants, remembered this example during our studies and stated that in this context a 3d view would have been very useful because it provides a different perspective on the data (cf. R-2 in Section 5.1.4), as stated by one engineer: *"we solely—and very intensively indeed—looked at what happened in the trace, [...] but in doing so, we missed what mechanically happened in the test drive"*. While a video view also might have been gainful for detecting this hardly reproducible error, videos are usually only available if particularly requested and furthermore lack in possible perspectives (cf. Section 6.1.2).

Mechanical trace translation vs. video: In line with our findings from the pre-design studies, several of our participants emphasized the 3d-Vehicle View's potential value to *"translate a trace to mechanical behavior"*. This 'mechanical translation' has not necessarily to be identical with the real behavior—which, for instance, can be traced by video-taping test drives. However, by comparing both, i. e., real (video) and translated (3d-Vehicle View), discrepancies can be tracked and ascribed to hardware-related errors, for instance, when a trace indicates a specific behavior (e. g., a window should open), however, in reality this behavior did not occur (window did not open), or vice versa. According to our participants' estimations, the 3d-Vehicle View therefore could especially profitable for field analysis and repairs where errors often hinge on hardware breakdowns. In this case an online-visualization, i. e., representing the information during the trace is recorded, can be helpful in order to immediately compare the real car's behavior with its virtual representative.

Reading and navigating traces: Several of our participants mentioned that the 3d-Vehicle View in coordination with time-based representation such as the Autobahn View, can be used for speeding up reading and navigating traces. Navigating over time and seeing what mechanically happened in the car provides the engineers valuable orientation points about activities during a test drive. This in turn can help to faster identify relevant sections of the trace, especially if an error description is only delivered verbally and/or semantically, such as: “At the moment I turned on the light, *xy* happened”.

Referring to spatiality: Last, several engineers mentioned that our linking and brushing approach (i. e., messages to sending ECUs, see above) would be helpful for them to quickly refer to mechanical components and in doing so, keep in mind potential error sources ranging from hardware changes or braekdowns. One engineer stated, for instance: *“That can be good to return to mind that a colleague recently repaired something in this particular region of the car. [...] Maybe his changes have something to do with my error!”* Furthermore, most of these participants argued for extending this feature and to provide richer information for a message, most notably the transporting bus systems and sending as well as receiving ECUs (see Car-x-ray for our solution to this, Section 6.3).

6.2.4 Discussion and Limitations

With Autobahn3D we implemented two basic coordination techniques, linking and brushing and semantic linking, between a 3d-Vehicle View and other abstract views. Our user studies indicated that there might be several situations in which analysis engineering can profit by integrating such a 3d view together with the coordination techniques proposed into their analysis environment. However, we are aware that we did not yet fully utilize the capabilities our configurable 3d-Vehicle View provides. First, we did not implement an automatic highlighting technique for animated elements in order to counteract missing relevant behavior (DC-5). Second, as outlined by participants in our studies, linking and brushing between components can be further enriched by showing other relations between the components, e. g., messages and transporting bus systems, or the actual path a message took (DC-3). Third, we solely coordinate our 3d-Vehicle View based on other view but not the other way round, i. e., interacting with the 3d-Vehicle View and coordinating abstract view. For this purpose, other applications would have to offer adequate interfaces for remote controlling them as well.

Besides further utilizing the 3d-Vehicle View’s capabilities, there are two more general restrictions with our trace analysis case study. First, a 3d view translating traces into mechanical behavior can only show what is sent over a bus system. While this might have several benefits (see above), on the other hand there is definitly much mechanical information that will get lost. For instance, by translating the trace it can be distinguished between an opened and a closed door, however, there is no electronic information available on ‘how far’ the door is opened. Also, the information whether a screw has been driven through a line cannot (explicitly) be recognized by looking at a trace. A 3d view can only be as good as the underlying trace data allows and therefore cannot exactly reflect reality. For this purpose, however, video-taping can be utilized as I described above. Furthermore, our study is definitely restricted by the manually created/adapted data we had to use. On the one hand, this is based on the simplified 3d model with the manually designed in-car network. On the other hand, we also had to translate messages semantics manually into mechanical behavior as this information is not available with current databases. Overcoming this restriction, however, exceeds the scope of a PhD thesis and also of a research project in general as it requires actively changing data storing and working processes as well as overcoming barriers between different departments, and therefore can only be addressed by the company itself. However, not until these technical restrictions are overcome will it be possible to provide engineers with closely integrated solutions that can productively be used in daily practices and studied under real circumstances.

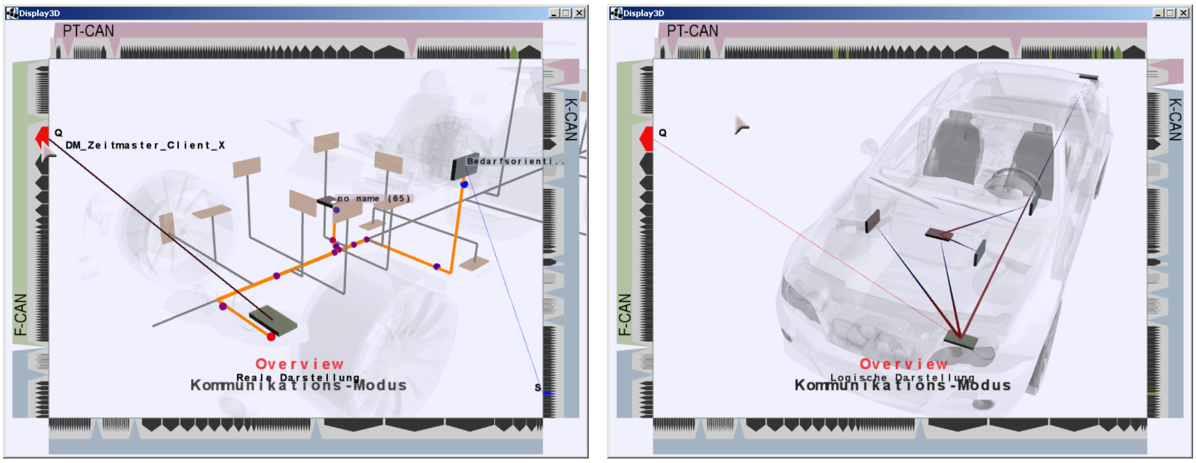
6.3 *Car-x-ray*: Closely Integrating Both Worlds

In the previous section I discussed our ideas on integrating a rather simple 3d model view in a multiple coordinated view environment (DC-9). In this section, in contrast, I focus on how the 3d model view itself can further and directly be enriched with abstract information. We implemented a prototype, Car-x-ray, that uses an additional *2d frame* around the 3d vehicle model to represent hierarchically clustered, abstract information about in-car networks. Lines between this frame and the 3d model indicate correlations between mechanical location and abstract information and help users to seamlessly bridge the gap between these two worlds. We used Car-x-ray for showing two different use cases. Our initial version aimed at showing real communication path “through” the vehicle (DC-3) and at supporting analysis engineers with a novel perspective on their data (cf. R-2 in Section 5.1.4). Inspired by a development engineer seeing Car-x-ray, we adapted and extended our prototype for a second use case and used it for visualizing early feature catalogs (DC-1). Our adapted version of Car-x-ray prototypically showed how spatiality can be utilized for browsing these catalogs (DC-2) and how this might be a valuable support for development engineers in better factoring in spatial parameters for partitioning the functional to the physical network. To learn about the prospective domain value of our ideas, we presented and tested both case studies with potential end users and discussed their estimations and opinions.

6.3.1 Case Study 1: Visualizing Communication Paths

While Autobahn3D focused on how a 3d view could be used for showing mechanical behavior, in line with our design consideration DC-3 the initial idea with Car-x-ray was to address the second use case estimated as relevant by engineers, namely using a 3d representation to show communication paths “through” the vehicle. According to automotive analysis engineers, this could especially be useful for diagnosis of errors where possible hardware changes or defects might have an influence on the error appearance (cf. Section 5.1.3). Similar to Autobahn3D, we therefore initially focused on solutions for analysis experts and on visualizing traces (DC-1).

Data In order to show communication paths of traces, first it was necessary to derive this information from traces as it is not explicitly available with them. To do so, we used the approach introduced for VisTra allowing us for translating traces to clustered, directed graphs with bus



(a) (b)

Figure 6.4: Screenshots of Car-x-ray in case study 1: **(a)** Highlighting the physical path between two FBs with additional flow animation using spheres changing there color from red (source) to blue (sink); and **(b)** abstract representation of a transitive chain of successor ECUs.

systems as clusters, ECUs as sub-clusters, FBs as nodes and exchanged signals defining edges (cf. Section 5.2.1).

Design As described above, Car-x-ray uses two basic components to visualize this information, (a) an abstract 2d frame surrounding (b) a semitransparent 3d model of the vehicle. The frame visualizes the data's hierarchical structure in a similar way as the treemap in VisTra did (cf. Section 5.2.2), however, uses nested, color- and size-coded trapezoids instead of nested rectangles and arranges them along the frame's borders. We used domain-specific colors for distinguishing the different bus systems and the width of the "FB-leaf-trapezoids" to code number of incoming and outgoing signal communication, i. e., large trapezoids were more communicative than small ones. The size of ECU- and bus-trapezoids directly results from the sum of FB-trapezoids. To focus on specific parts of the network, the user also can interactively fold and unfold bus systems. Within this frame, the virtual 3d model is shown, can be navigated by the user, controlled in terms of components transparency and provides an abstracted in-car network similar to the one we used with Autobahn3D.

By selecting a FB trapezoid from the 2d frame the user can initiate a representation of how communication had spread in the network. To do so, s/he defines a FB either to be a source-FB and in doing so highlighting the selected element plus all reachable successors (transitive chain of successors) or to be a sink-FB showing all predecessors respectively along with the selected FB (transitive chain of predecessors). After the user has selected a FB—for instance, a source-FB—the following elements are highlighted: All involved FB trapezoids are highlighted in red (source), blue (sink) or green (in-between or unspecified nodes)[6]. Folded bus systems that contain involved elements are automatically expanded. Within the 3d model, involved ECUs as well as the involved bus system sections are highlighted by color and enlargement. Finally, a line between the selected FB trapezoid and the appendant ECU in the 3d model is shown. In the same way, the user now can specify one of the involved FB-trapezoids as sink element and in doing so, initiates representing the exact communication path between source- and sink-FB with all involved elements (cf. Figure 6.4-a). As the described highlighting technique does not necessarily reveal the exact path that the communication took (e. g., is a specific gateway only involved in one path or in several different path), we designed two additional representation techniques for enriching the approach we described. First, we allow the user for initiating an animation by hovering over involved elements. Instead of using one single representative as in the in the early prototypes, we used a stream-like animation of many representatives changing their color from red (source) to blue (sink) in order to indicate the direction of communication flow rather than following a single item over time (DC-4). A second alternative that can be initialized by the user, is the abstract visualization of communication paths via direct arrows between the involved ECUs (cf. Figure 6.4-b). In doing so, all ambiguousness is eliminated at the expense of reduced spatial correlation.

To avoid unnecessary navigation, we also integrated an automatic camera planning strategy that is triggered after a user selects an element (DC-6). Our strategy automatically rotates the 3d model based on minimizing the distance of all involved ECUs to the camera and on avoiding occlusion of involved ECUs. Finally, we added a traditional, hierarchical sorted *List View* and a search, as interacting with the 2d frame where most labels are just shown on mouse over is tedious. Equally to the 2d frame, this view can be used to select source- and sink-FBs.

Evaluation After evaluating the usability with four external testers and fine-tuning the tool thereupon, we conducted a user study with seven domain experts (four analysis experts, one analysis tool developer and two researchers on novel analysis methods), in order to (a) validate our design considerations and (b) to discuss potential domain utility of our approach. We used our typical setup for qualitative "domain-estimation" studies, i. e., we used the dataset we manually had prepared, let our participants conduct a set of tasks similar to the one used for early prototypes (understanding correlations and communication), encouraged them to comment their thoughts during conducting the tasks, and subsequently discussed the provided solution in terms of potential domain value. Each study took one hour and we used note taking to track communication and tool usage.

By observing our participants conducting the tasks and by analyzing our think-aloud protocols, we found several indications, that the features we added helped in overcoming the problems of the early prototypes. Mapping communication path to spatial information rather than to animated messages/signals led to a better recognition of correlations and less errors in detecting them. Besides, it was preferred over animation by the subjects who had already participated in the first study, as it *"does not impose any additional time costs"* (DC-4). While the additional stream

---

[6] Cf. Section 4.3.2 and 5.2.2 for color conventions.

animation we added was evaluated *"[...] unnecessary, [as] it provides no additional value"*, however, especially the possibility to alternatively show correlations with abstract arrows was liked very much. Furthermore, we got good feedback to our design decisions of integrating an automatic camera planning strategy (DC-6) and on providing an additional list and search view which was basically used for selecting elements (DC-9). Similar to our previous studies, our participants often stated to *"have fun"* by using Car-x-ray, and frequently started to encourage the integration of other features that might improve the tool, such as additionally showing bus loads, providing ECU details on demand, and various other features known from their current analysis tools. The strongest point of critique—that we, however, were already aware of before conducting the study due to the reasons described in the previous section—concerned the representation of ECUs and bus systems as positions and wiring paths did not (exactly) map real positions.The abstract representation, however, was not seen as a problem by any participant. This strongly underlines our design consideration " Real positions, abstracted forms" (DC-8).

In terms of potentially added domain value, most of our participants argued (again) for using such solutions for educating novice users and for communication purposes. However, beyond that several engineers provided us with potential reasons why such a solution might benefit their own work. One analysis engineer even stated: *"[If embedded with current software environments] this tool would be used in my department straightaway"*. Along with the potentials of showing transitive chains in general (cf. results of our studies with VisTra, Section 5.2.3), most importantly the potential to provide a novel perspective on the data showing how information dispersion directly correlates with its spatial positions in the vehicle was estimated highly valuable (cf. R-2 in Section 5.1.4). Furthermore, two of our participants underlined that they would use the tool for collaborative analysis tasks: *"for me the highest value of such a tool would be that I can easily analyze traces together with my colleagues"* (cf. R-11 in Section 5.1.4).

In line with our findings from previous studies, however, it is clear that this Car-x-ray in this application scenario is not yet applicable (and testable) under real circumstances. For a productive application in daily work, overcoming current technical restrictions of embedding mechanical data in analysis tools (cf. Section 6.2.4), format compatibility and automatic connection to real data (3d models and traces, cf. Section 6.2.1), seamless integration with other trace analysis software (cf. R-9 in Section 5.1.4), and a conceptual coordination with these tools (cf. R-4 in Section 5.1.4) is crucial.

6.3.2 Case Study 2: Visualizing Early Network Specifications

Along the work on Car-x-ray, we frequently discussed our approach with automotive engineers, among them also an interested development engineer. While this engineer actually worked in the area of network architecture and therefore rarely was confronted with trace analysis, he was enthusiastic about our approach and provided us with several ideas how Car-x-ray could be adapted to support engineers responsible for partitioning functional to physical networks. In collaboration with him, we therefore adapted and extended Car-x-ray in order to visualize early network specification data and to provide novel perspectives in the partitioning process. In the following, I briefly outline our rationales for this usage scenario, show how we prepared the data, explain design adaptations and add-ons, and summarize engineers' qualitative feedback on our second case study with Car-x-ray.

Rationale and basic idea In early development phases of in-car communication networks, engineers start with functionally describing a new car series' electronics with functional blocks which they cluster to systems, and systems again to domains[7] (cf. Section 4.2.1). After the functional specification has been completed, a next crucial step is to map these functional specifications to the physical network (called 'partitioning'). Decisions how to ideally partition the network rely on a variety of different parameters, including communication between FBs (see, for instance, our solution on RelEx), computing resources, but also on spatiality, i. e., where necessary information is measured and where mechanical action is accomplished (in engineers jargon, these information is called "effect positions"[8]). Effect position information is essential for optimizing wiring and to reduce costs and risks of long communication paths. However, currently this information is neither explicitly available with the data nor with the text-based tools, and reliable decisions hinge on implicit knowledge and estimations of engineers about spatial correlations and distribution. Our basic idea therefore was to make this implicit knowledge about effect positions available and explorable in a 3d interface in order to explicitly factor in spatiality for partitioning decisions.

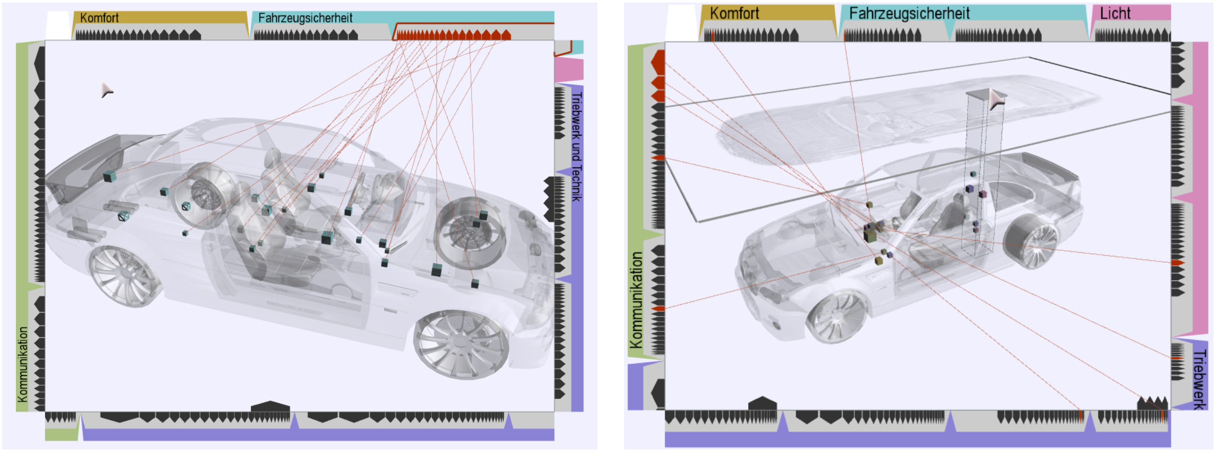
Data preparation As early functional network description data with explicit spatial information does currently not exist (see above), our first step was to manually adapt an example dataset with 280 FBs by adding spatial effect positions for each FB. For this purpose, we contacted several experienced in-car network architects and collaboratively derived effect positions for 54 FBs. A functional block "front lights" for example got the effect positions front left and right, as well as the position where the light switch is installed at. For the rest of the FBs we tried define effect positions best to our own knowledge. The final dataset was stored in an xml file.

Design modifications In order to meet the different needs of network architects, we slightly adapted and extended the design of Car-x-ray. First, the abstract 2d frame now represents the early network specification data described above, i. e., color-coded domains, systems and FBs. We used the width of the trapezoids for coding available system capacity (an important current metric for partitioning FBs) and allowed the user for moving around FBs between systems and domains in order to simulate partitioning tasks. As editing the specification, however, can result in over-filled (sum of FB capacity > system capacity) and under-filled (vice versa) systems, we fine-tuned our 2d frame and represented FBs that do not "fit" any more into a system by flipping their trapezoids to the frame's inside (over-filled system), and allowed for free "capacity-space" in under-filled systems.

---

[7] Often they additionally use further subsystems for more fine-granular clustering.

[8] Translation from the German term "Wirkposition".



(a) (b)

Figure 6.5: Screenshots of Car-x-ray in case study 2: **(a)** Selecting a complete system in the 2d frame reveals all its effect positions in the 3d model; and **(b)** the magic ray (right) and a spatial selection (left) done with the magic ray showing influenced domains.

Second, we removed the virtual in-car network as it is not available at the process stage our new target users worked at. Instead we added for each FB's effect positions small colored cubes (color of the domain) to the 3d model. As this, however, would have resulted in simultaneously showing over 500 *effect position boxes*, we initially set these boxes invisible and allow the user for interactively exploring them. To do so, on the one hand s/he can select one or more FBs from the 2d frame (or from the search/list view respectively) resulting in highlighting trapezoids in the 2d frame and effect boxes in the 3d model, and in connecting them with a correlation line. By selecting a complete system or even an entire domain, the user therefore is able to explore the dispersion of this system's/domain's effect positions over the vehicle—valuable information that can potentially be used for reorganizing partitioning of FBs (cf. Figure 6.5-a). On the other hand, the user can browse effect positions directly in the 3d model and in doing so investigate which areas are influenced by which domains/systems/FBs. As it turned out that manually navigating and trying to find a "invisible" effect boxes in a 3d model on a 2d screen is difficult and tedious, we integrated a browsing technique based on magic lenses [VCWP96] and object shadows [PR02]. To do so, we added a 2-dimensional plane above the 3d model showing a semitransparent shadow of the vehicle model. The plane can be vertically adjusted by the user and by hovering over it with the mouse a "magic ray" down through the car is shown. Each effect box in the model intersected by this magic ray then becomes visible and the user can browse all effect boxes by simply hovering over the 2d plane. By using the scroll wheel s/he also can interactively adjust the size of the magic ray. Finally clicking on the surface selects all currently intersected effect boxes and shows their correlations to the abstract frame elements in the usual way (connection lines, automatic camera planning, highlighting of all involved elements in the frame, etc.). Figure 6.5-b shows our magic ray technique and the selection of several effect boxes with their dispersion of various systems and domains.

Last, we retained the on-demand opportunity to show correlations between FBs, i. e., between their effect positions with abstract arrows, as this might also be an important information source when partitioning networks.

Evaluation We conducted a think-aloud study with five automotive network architects in a similar setup to Car-x-ray's first study, however, with a different dataset (see above) and tasks related to network partitioning (browsing and modifying specifications). Beyond the general results we found in the first study (cf. Section 6.3.1), we especially got several interesting comments about the use case we adapted Car-x-ray for. All participants were very open-minded about our approach and liked several of our ideas presented with this second case study. First, four participants emphasized the size-coding approach in the 2d fame as a good overview technique for learning about capacity ratio. Second, three engineers stated that using spatial information to browse the functional specification is intuitive, e. g., *"I intuitively know where I have to grasp the data"* and that this might particularly be beneficial for communication and collaboration purposes. Last, all engineers agreed with us that it is important to factor in spatiality for good partitioning decisions, that making this information explicitly available is necessary and that embedding Car-x-ray or a similar tool into their daily working practices might help to better address this aspect. Before, however, in line with the results from Autobahn3D's and Car-x-ray's first study a variety of technical obstacles have to be overcome. Beyond these already known aspects, we particularly learned that our approach did not address ambiguous opinions between engineers about effect positions of FBs. We observed three of our participants, expecting the effect position of one and the same at three different locations in the car. This is a serious problem that definitely should be addressed in future work.

6.4 Summary

In this chapter, I presented our ideas on utilizing virtual 3d models for visualizing in-car communication network data. Initially, we conducted qualitative user studies with two prototypes that we found already existing and derived several design considerations to address the questions when to use and how to design 3d model visualizations in our domain. Subsequently, we implemented two further 3d prototypes, Autobahn3D and Car-x-ray, showed how we applied our considerations in their designs, and discussed them in three case studies with potential end users from the automotive domain. Along with the question of how to design 3d visualizations, we were additionally and particularly interested in the question if there are valuable application areas for 3d tools in daily work of development and test engineers. While the early prototypes—which have not been developed for a specific target group or use case—were deemed by our study participants to be useful rather for education and presentation tasks than for "serious engineering work", we showed several examples judged by engineers to be potentially useful for their daily work. In particular, we focused on supporting trace analysts in understanding correlations between electronic information and mechanical behavior and investigating communication paths and transitive chains (cf. Section 5.2.1) in their spatial context. Furthermore, we visualized early network specification documents in order to exemplify how spatial aspects can better be addressed for partitioning the functional to the physical network.

While our studies indicated several potentials that 3d visualization might add to automotive electronic engineering, yet, we cannot provide any concrete validation for daily engineering work. Unavailable and incomplete datasets (3d models, missing information about correlations between mechanics and electronics) and—along with this—the absent opportunity to automatically and closely integrate our tools with current practices (cf. R-8 and R-9 in section Section 5.1.4) hinders studying the systems under realistic circumstances and restricts the results to estimations of prospective end users. Proving the value of such tools under real circumstances adheres to overcoming these technical restrictions first. In this chapter, we furthermore solely focused on 3d techniques. In a next step, it would be interesting to compare such 3d solutions with equivalent 2d representations such as plan views in order to investigate whether they can provide the same richness of information without posing additional occlusion and navigation costs (see, for instance, [VWVS99] and [SJOC01]).

Chapter 7

Evaluating InfoVis in Large Company Settings

In the previous chapters, I presented several design studies and showed how we collaboratively developed and evaluated them with automotive engineers. We[1] learned about data, tasks and challenges in the automotive electronic domain and about various requirements and implications for designing InfoVis/VA tools in this area. During this process, on the other hand, we also gained much experience in conducting field studies in a large automotive company, encompassing (a) explorative studies, i. e., evaluating current practice within the company setting, and (b) formative/summative studies, i. e., studying developed visualization tools in their accustomed working environment. We learned that applying and evaluating information visualizations directly within such a large company context is a fruitful endeavor and can produce valuable insights for the field of information visualization as a wide range of real data analysis problems, tasks, and datasets are available. However, on the other hand, we also found unique challenges and requirements stemming from structural differences to small organizations, such as a higher degree of organizational complexity, more specialization, formalization and decentralization [Daf95].

Considerably less work has been published on these difficulties of evaluating InfoVis within such a work context. In this last main chapter, I therefore want to summarize and categorize our three-year experience of conducting InfoVis/VA field research at BMW—a company having 100,000 employees. Derived from this experience, I provide a concrete set of field evaluation challenges as well as a collection of recommendations for applying and evaluating InfoVis/VA tools in large company settings. Challenges and recommendations include both aspects specific for InfoVis/VA evaluation but also more generic considerations which turned out to be no less important for our research. In doing so, we want to help other researchers and practitioners with useful guidance in preparing and conducting evaluations of their products within a large company setting.

---

[1] Portions of this chapter have been published in [SIBB10]. Thus, any use of “we” in this chapter refers to Michael Sedlmair, Petra Isenberg, Dominikus Baur and Andreas Butz.

The chapter is organized as follows: After this brief introduction, I start with relating our work to other approaches of field evaluation in InfoVis but also to known organizational obstacles from the HCI domain. After that, I introduce our categorization of nine challenges and fifteen recommendations, both derived from the three-year field experience. The chapter concludes with a summary and discussion of current limitations.

7.1 Background and Related Work

All challenges and recommendations that will be presented in this chapter focus on field evaluation strategies (cf. Section 1.3). Existing types of research strategies within the field can be roughly categorized as field studies and field experiments [McG95]. Field studies are described by McGrath [McG95] as direct observations with minimal possible intrusion within "natural" work environments, whereas field experiments are a compromise strategy where features of the system are manipulated in order to gain more precise results. The work presented in this thesis, concentrated on applying both field strategies within one specific type of field—that of large industrial companies of several thousand employees. In the following, I start by discussing previous field strategies that were conducted with information visualization tools and then go into more detail on obstacles of field research in large company settings as discussed in the general HCI literature.

7.1.1 InfoVis Evaluation in the Field

Evaluation in the context of specific data and task sets is a fundamental part of information visualization research [Pla04, TC05] as systems and techniques developed by researchers are often intended to support everyday work activities for domain-specific tasks and data. In order to more clearly understand and assess "real world" data analysis problems and the use of our tools within a specific work context, a close collaboration with domain experts is often instrumental [ED06, PRS01, SND05, TM04]. When working with domain experts on their own data and tasks, it is helpful or even necessary to study their data analysis habits, requirements, goals, and tool use within their respective work context, or "field" [Pal09, IZCC08, Mun09]. While still a considerably large amount of evaluations for information visualization tools are conducted in lab settings [ED06], more and more researchers start to invite their target audience to participate in user studies and to conduct evaluations in the field: Perer et al. [PS08], for example, studied their social network tool with several experts from different fields of data analysis. Ethnographic studies have also been used within a user centric design process with domain experts and have been shown valuable as a formative part of the design process: Tory et al. [TSF08], for example, documented the results of a qualitative analysis in the building design field and concluded that their structured analysis of qualitative study data provided deep insight on the work processes with visualization. Long-term studies [SP06] are another type of field strategy that offers the chance for deep insight and learning of the workings of a field and possible merits of visualization use. Unfortunately, they are laborious and only few have been reported on in the literature so far (e. g., [GK03b, MMKN08, SNLD06]). The work by González and Kobsa, for example, describes the adoption of an information visualization tool by data analysts over a longer period of time [GK03b]. In a follow-up paper, they describe further observations on the merits of such tools in the workplace [GK03a]. While these examples are promising steps towards more evaluation in close contact to domain experts, more insight is needed on the challenges of conducting information visualization evaluation within specific work contexts. Based on our experiences, I therefore want to list a first set of challenges and recommendations for deploying and evaluating information visualization within a large industrial company.

7.1.2 Organizational Obstacles Known from HCI

In the area of HCI, more precisely in Participatory and Contextual User-Centered Design (UCD), a considerable amount of previous work exists on how to meet usability evaluation and user needs by actively involving all stakeholders (e. g., end users, management, decision makers). Much of this research has been conducted in industry settings [BNRS08, MVSC05]. Grudin [Gru93] explicitly discusses obstacles encountered in large companies such as finding "representative" participants and crossing organizational barriers during a UCD process. Poltrock and Grudin [PG94] conducted two observational studies in large companies and reported how several organizational factors (e. g., missing commitment, unsatisfying training) can block UCD. Jeffries et al. [JMWU91] provided a comparison of four formative usability studies in real world environments and recommended heuristic evaluation and usability testing methods for evaluation when considering the number of found problems, UI expertise, and costs.

The main difference of our work to most of these approaches, however, is that we do not examine business-to-customer situations: While much of the previous work was concerned with employing UCD to develop tools for expert users on the outside, we are interested in designing information visualization tools for use within a large company to improve the work processes of its employees. While novel requirements and challenges applying UCD for in-house tools, such as platform and application buying concerns, change management, or the IT life cycle, have been previously discussed [BÅPL03, HBH09] related work in this specific area is still rare. In particular, the challenges of information visualization evaluation—as opposed to general usability evaluation—have not received much attention in this context. The contribution provided in this chapter is a first collection of challenges and recommendations for applying evaluation within a large company context. We hope that this collection will be expanded and modified as more evaluations of information visualizations will be conducted in such work contexts.

7.2 Problems and Challenges

While designing and evaluating information visualizations within a large company for the past three years, we have experienced several field characteristics that pose particular challenges to evaluation. These challenges basically arose due to the large company setting where workflow, bureaucracy, or hierarchical structures can be quite differently defined compared to smaller companies [Daf95, PHH69]. For instance, large industrial companies are often characterized by a high degree of collaboration and specification. A single employee often is highly specialized and responsible for a small subset of a highly specific collaborative task set (cf. [Daf95], or our experience on user diversity described in Section 3.6.2). Therefore, the know-how in a company is often widely distributed and a single person is not always able to understand all facets of the entire task domain [Dru88]. In a small company or research lab—where up to now most InfoVis field evaluations have been conducted in—a problem domain is usually more specific and employees may be able to maintain a comprehensive understanding of their work context and may even be able to deal with many tasks personally.

When attempting to evaluate information visualization within a large company context, it is imperative to understand the characteristics of this specific evaluation field in order to be prepared for the challenges that may arise in planning and conducting a study and finally analyzing and disseminating the results. In the following section, I describe nine specific challenges to evaluation of information visualization in large companies, categorized along the typical flow of a user study: study design, participant gathering, data collection, data analysis, and result generation. The challenges ground on our own understanding of conducting the field studies which have been described earlier on in this thesis (both qualitative and quantitative) but also on general lessons learned from HCI and sociology literature. For our categorization, we did not focus on a specific evaluation methodology, data collection or analysis method, but consider challenges both of studying already developed visualization tools as well as challenges when attempting to evaluate current practice within the company setting. Additionally, we focused rather on utility evaluations which are more holistic in nature than on usability evaluation assessing how well the interaction with an information visualization tool is designed.

Finally, we also included challenges related to tool deployment within the company setting which has been both extremely valuable, as a prerequisite to longer-term studies, as well as challenging for us (cf., for instance, RelEx, AutobahnVis and Cardiogram).

7.2.1 Study/Application Design

EC-1: Integrating Tools in Daily Work Processes
In order to conduct studies under real conditions it is often indispensable to integrate information visualization tools in daily work practice (see above, and [SP06]) what is a labor-intensive process, not only in large companies. Tools have to be stable, robust to changing datasets and tasks, and—if they replace previous tools—should support the functionalities of the tools being replaced. Besides these common challenges, due to our experience there are two additional critical aspects to consider in large company settings:
(a) Technical Issues: Task specialization is common in large industrial companies. Therefore, many specific data analysis tasks exist and most of these will likely already be supported with a variety of different analysis tools (see, for instance, the variety of currently available trace anal-

ysis tools, Section 5.1.2). These tools are often well integrated to perform within a chain of other tools so that they together provide more encompassing analysis solutions. Under these circumstances the integration of a new visualization tool may be quite challenging as it may break the chain of analysis processes that are already supported by existing solutions. Furthermore, you may be confronted with overcoming technical challenges such as incomplete and proprietary interfaces—especially if the existing tools are in-house software (cf., for instance, solutions in Chapter 5). However, the integration of a novel tool is usually indispensible when one wants to study its usage within a specific established work context [SND05].
(b) Political and Organizational Issues: Many large companies require the authorization of software or software components upfront. Initially, this may not seem complicated, however, depending on the amount of bureaucracy involved, this process may require highly collaborative synchronization efforts and may become long and exhausting (see, for instance, our integration process of MostVis with the company's official in-car network database, Section 4.1.4).

EC-2: Getting the Data
Not only the tools and techniques but also the domain-specific data itself will likely be distributed across different work groups within large companies. Your novel visualization approach, however, might have been designed to improve work with combined and aggregated sources of data. To evaluate your tool with these data sources you may have to deal with issues of interoperability between different data sources on different machines and within different work groups. Unavailability, different data versions, different or inappropriate format, unmaintained sources, and most importantly security restrictions can issue additional challenges to you (for a good examples, see our proposed solutions for 3d visualization in Chapter 6, WiKeVis in Section 4.2, or VisTra in Section 5.2). However, being able to evaluate visualizations with the data used and created by your participants in their everyday work practices can be critical—not only in evaluating how your tool is used with real world data characteristics, but also in order to convince the participants or stakeholders that your tool may actually improve everyday work.

EC-3: Choosing an Evaluation Context
Large companies have employees with varying goals, views, and work habits all working together (cf. [Dru88], or our results on user diversity in Section 3.6.2). In large industrial companies you will encounter a variety of personalities and opinions. This is particularly important to keep in mind when you are planning to conduct qualitative work such as interviews, observations, and focus groups with or without information visualization tools. There may be many teams with similar data analysis tasks and data types across a company that you can collaborate with but the qualitative results you may collect during a study in these teams can be vastly different.

7.2.2 Participants

EC-4: Finding Domain Expert Participants
It is very common that employees in large industrial companies are working under heavy time pressure and are bound to strict deadlines. Having to revise a deadline often leads to a considerable loss of revenue (see, for instance, postponing SOP, Section 3.3.1). These pressures result

in specific challenges for evaluation in general and for evaluation with significant participant involvement in particular: (a) Getting domain experts for studies is generally difficult. Time = money! Every hour you want participants to work with you is an extraordinary task without direct evidence of impact on their actual work tasks. (b) Under these circumstances, it becomes difficult to argue for long-term studies (e. g., MILCs [SP06]) without any kind of "pre-evidence" that the required involvement will result in qualitative or quantitative improvements to future work processes.

EC-5: Attachment to Conventional Techniques
Even if your tool may be designed to improve conventional tools, experts may be very accustomed to and effective with them. This effectiveness may lead to a certain amount of attachment to the traditional tool and may result in a certain reluctance to learn a new system. By working with their traditional tools over a long period of time people will likely have developed skills to estimate the effort and time required for a specific analysis and can factor this knowledge in when planning upcoming deadlines. It may be difficult for them to estimate this with a novel tool. In addition, some domain experts may have learned to master complex tools and data analysis tasks over the years. If you managed to design a tool that significantly simplifies a specific data analysis compared to a previous tool you may strip these experts from their respected expert status and allow others to also conduct the same tasks [Pal09]. In our own studies with trace analysts (cf. Section 5.1), we, for instance, ran into several participants working directly on the raw data and making clear to us that visualization would not be useful for them as it *"is just a potential source of error"*. These participants had learned over the years to read the raw hexadecimal data and were referenced by other analysts as "absolute analysis experts". Such issues, however, can complicate both acquiring participants for your studies (see also EC-4) and conducting and evaluating comparative studies.

7.2.3 Data Collection

EC-6: Confidentiality of Information
Video-, audio- and screen-recording can be useful data collection tools during evaluation. Especially for qualitative evaluation such data collection helps to capture participants' actions, conversations and responses and allows systematic coding and analysis of the data in retrospect [Ber00]. However, large companies often have confidentiality guidelines and restriction policies—Intellectual Property Rights (IPR) security requirements—that might forbid certain recording techniques (cf., for instance, Section 3.5). Before conducting a user study where some kind of recording is necessary, these policies have to be checked and permission may have to be sought. In addition, being discreet about collected data is important. Internal work processes are often secret. This, in particular, means that you may not be allowed to share your data with others (e. g., with a second coder or in online tools), you may only be allowed to discuss anonymized results, and that you will have to deal with publication restrictions—not only about the results of your study but also when talking about the data analysis characteristics of the tool you may have studied (see EC-9).

EC-7: Complex Work Processes
One important goal in information visualization is to support people in solving complex tasks. For this purpose, an important first step is to understand current data analysis problems with pre-design evaluation [IZCC08]. For us, this type of evaluation has been a very important step in order to focus our work on solving the right real world problems (cf., for instance, Section 3.5, 4.3.1 or 5.1). Pre-design studies, however, become additionally challenging in large companies where complex problems are often split among several, highly specific sub-problems. Understanding the specifics of both the overarching problem solving process (macro challenges) and the individual (micro) challenges may be difficult for an outsider (see EC-5). When observing different employee groups, it should not surprise that some may have built their own work processes or tools around their work tasks or datasets and that other groups and employees who may have similar data, may have come up with different solutions while being unaware of solutions from other groups (see EC-3). Additionally, experts in large companies often have varying tasks and not all of them may be relevant for the observer and neither do the domain experts want to be observed in every situation. Finding the appropriate balance between unobtrusive observation and intervention when observing work processes requires skill and tact on the side of the evaluator. Also notice that work tasks may not be understandable by solely observing them as much domain and company background knowledge might be necessary (cf. our strategy adaptation from pure observational studies to contextual interviews, Section 5.1.1). On the other side, simply talking to participants in pre-scheduled appointments is also often not sufficient: "What people tell you is not always the same as what people do" [BH98].

7.2.4 Results

EC-8: Convincing the Stakeholders
An important evaluation goal in information visualization is to understand how people use your visualizations to solve real world problems. This goal does not necessarily align well with the goals of stakeholders whose task it is to maximize profit for the company. Therefore, they are more interested in tools that help to save money and improve the effectiveness and efficiency of their employees (again, time = money). Another goal of the company is speeding up current work practices (e. g., more insights/time [MA08]) while we as researchers may be more interested in factors that influence or improve qualitative aspects of the work or the specific factors that may have led to improvements (e. g., how insights were achieved [SNLD06]).

EC-9: Publishing
To allow information visualization to grow as a field and to share and discuss your results with the larger research community they have to be made public. Due to competitive reasons, large companies, however, often have restrictions on what can be published, in particular if your work leads to a competitive advantage. You may have to expect a various bureaucratic hurdles.

7.3 Recommendations

To meet the problems and challenges described above, in this section, we provide a set of concrete recommendations for other information visualization researchers who are planning to conduct evaluations within a large company setting. We ground our recommendations on the three-year experience in conducting explorative, formative and summative field studies at BMW. The organization of our recommendations reflects the main categories of challenges in the previous section. Some of our recommendations are specific to working with data and data analysis tasks with information visualizations and some apply to evaluation in this field more generally.

7.3.1 Study/Application Design

ER-1: Overcome Technical Obstacles of Data and Tool Integration
To evaluate the full working process of domain experts an information visualization tool should be integrated and coordinated with current domain specific techniques and tools to operate in an entire analysis environment (see, for instance, R-9 in Section 5.1.4). Many of the existing tools in a work environment, however, have often been worked on and extended over the years and an integration may be a considerable software engineering challenge. In our projects on AutobahnVis and Cardiogram, for instance, we spent more time on overcoming technical restrictions than on designing and implementing the actual visualizations. In the end, however, in both projects this paid off in the opportunity to evaluate our tools over a longer period of time under realistic circumstances. Instead of an integration one can consider to extend the features of a new visualization tool to unite the capabilities of a previous tool chain. Depending on the amount of previous work this could be a valid solution for small projects. The costs of either solution should be considered based on the goal of the evaluation. Supplementing existing tools is often the cheapest and most effective way [GK03b]. Eventually, however, it is invaluable to have tools that do not require additional steps to work with domain-specific data. Such tools will not be accepted in everyday work (see, for instance, our 3d tools in Chapter 6, or our discussion on VisTra in Section 5.2.4). Tightly integrating your tool to work with only a subset of the data may be more important than supporting wide applicability. While this factor may not be important in research departments—where insight may outweigh time—the obstacle of additional time requirements is crucial in industrial environments. Having to convert data manually should be a last resort [SP06].

ER-2: Choose your Study Environment with Care
Obstacles for studying work environments, studying your solution or deploying your tool often result not only from technical challenges but from political or organizational requirements. To conduct evaluations you need permissions and committed collaborators. In order to receive permission it is imperative that you find employees who will support your project and that you convince your stakeholders (see ER-13 for further recommendations). You may encounter similar data analysis tasks and data across different groups within a large company. It takes skill as a researcher to generalize from the individual opinions and views encountered to find the right

target group and work environment for the tool you built or are interested in building. When conducting pre-design studies, connect with motivated domain experts and start with identifying and understanding different sub-problems and sub-groups in your problem domain. Talk to various people and be open-minded towards existing solutions from other people beyond your target group. Use this knowledge to become an expert in this domain and try to validate the importance of the problem you want to solve (see also our discussion about domain-problem threats in Section 3.4.4). However, do not try to solve everyone's problems at first hand. Rather try to find a motivated sub-target group with specific and concrete problems and with interest in your work. After researching specific solutions and validating sub-domain specific solutions, try to abstract your lessons-learned to a more general approach (see also our discussion on VisTra in Section 5.2.4).

ER-3: Consider both Employee- and Researcher-initiated Solutions
Generally, we distinguish between two kinds of solutions: *(a) employee-initiated solutions* (such as the solutions presented in Chapter 4) where one or several employees request for a specific tool, and *(b) researcher-initiated solutions* (such as the solutions presented in Chapter 5 and 6) where you, i. e., the visualization expert, advertise a tool. Both approaches may be successful: Employee-initiated-solutions are often easier because employees can argue that your tool may address a recognized analysis problem. However, they are not an inevitable factor for success, as this also depends on who you are collaborating with (see, for instance, our project on WiKeVis in Section 4.2.5), how important the problem you solved is (see, for instance, BNVis in Section 3.4.3), and of course on the quality of the solution (see InfoVis- and domain-design threats in Section 3.4.4). Researcher-initiated solutions, on the other hand, may require very tactful negotiation but are no less important. Specific work practices may have become established over the years and employees may be satisfied with improvable solutions. In these cases, a push from an outsider can help to provide a new perspective on more advanced data analysis options. For a good example for an researcher-initiated solution please see our project on Cardiogram (cf. Section 5.4).

ER-4: Delight with Usability and Aesthetics, Avoid Window-dressing
Do not underestimate the value of usability and aesthetics. In in-house tools these aspects are often neglected [HBH09]. Usability and aesthetics are important distinctive features you can use to gain acceptance of novel tools or to convince stakeholders. During the development of MostVis, for example, we intensively focused on usability engineering by doing several usability studies. The higher usability compared to current tools definitely was a major reason for our good results in the comparative study which finally led to the tool being integrated in a larger work context. During our studies with 3d tools, on the other hand, (cf. Chapter 6) our subjects frequently pointed out the aesthetics of the solutions and mentioned that it would be much easier to convince decision makers with such solutions. But be also careful not to exaggerate "aesthetics" and consider how much is accepted and required by your users. Our experiences with trace analysis engineers, for example, showed that rather simple and easy understandable solutions were strongly preferred (cf. Section 5.6)—however, this does not exclude aesthetic solutions, of course.

ER-5: Installability and Absolute Support
In large company settings you might be confronted with employees' computers having restricted user accounts, fine-tuned operating systems or specific security policies. You have to clarify all relevant technical aspects upfront and guarantee smooth operation and easy installation of your tool on employees' computers. The latter is particularly important when you plan for conducting long-term studies such as MILCs (cf. [SP06]) where you may provide participants with frequent tool updates. For other questions, provide good and fast technical support. Be always aware that due to time pressure (EC-4) even small technical barriers can lead to forfeiting all conceptual benefits your tool provides.

7.3.2 Participants

ER-6: The Magic One Hour Limit
Our experience showed that recruiting participants for one hour or less is significantly easier than for longer time periods. Employees are occupied with meetings, appointments, and deadlines and additional involvement in user studies just adds to this work load. Be prepared and professional in recruiting and conducting the study and stick to your suggested time limit.

ER-7: Convince your Target Audience
Even though participants may be very attached and used to their current tools there are some things you can do to convince them of your solution. Try to solve real problems of your target group even if these first-hand solutions are small and actually not the main focus of your work! People become immediately interested if you present solutions which they can use immediately with their own data. Your participants will be much more motivated to attend your studies when they know they will be remunerated by working on solutions of their current problems. One way to achieve this, is to integrate some simple but highly desired functions not available with current tools, as we did, for instance, with MostVis (grouping function, cf. Section 4.1.2) or in RelEx (signal path visualization, cf. Section 4.3.2). Even outlining solutions, e. g., presenting mockups after exploratory studies, was very valuable to convince our participants of the potential value of our work. However, be careful with outlining ideas which you might not be able to implement (e. g., due to data restrictions, EC-2).

ER-8: Learn from the Experts
Identify experts in your problem domain. You can learn much by interviewing and observing their practices. Often, they may not be interested in your solutions because they have mastered problems already using their own approach. Try to identify why their practices are effective and efficient and think about how you can use this knowledge in your tool to make it available to a wider range of people. During our exploratory studies with trace analysis engineers (cf. Section 5.1), for example, especially talking to three specific and long-time experts helped us enormously in understanding the variety of potential error sources and the importance of a hexadecimal representation. Both AutobahnVis' and Cardiogram's designs benefited much from their experience.

ER-9: Conduct Usability Studies with Outside Testers
In all our projects we conducted various heuristic and think-aloud studies alongside the development process. These studies helped us to focus on usability issues and we often conducted them with students and external testers with a usability background [TM05]. Usually this will not entirely supersede usability studies with target users—as they are the only ones who can provide domain expert feedback—but it definitely helped us to save valuable experts' time (EC-4).

ER-10: Gentle Reminders
Especially in long-term studies it might be valuable to gently remind your participants of the existence of your tool. Employees usually have a variety of different, not only analysis-related tasks and your tool may therefore only support a fraction of these tasks. Time-pressure and long periods between analysis tasks may lead to reverting back to confirmed habits regardless of your tool's quality. Gently remembering participants of the benefits can help you both, in slightly integrating your tool with daily working practices and in studying the utility of your tool. Informal venues can be a good opportunity for such reminders.

7.3.3 Data Collection

ER-11: Try to Get a License, Do Studies in any Case
Check IPR policies (see EC-6) and, if required, try to get permission to video or audio tape. We agree with Dix el al. [DFA04] that the analysis of recorded video or audio will allow you to gain a much deeper understanding of the scenario under study. If a permission was received, equipment has to be carefully installed. It is imperative that participants know about recording devices and that privacy concerns are thoroughly discussed. Particularly, in large companies, employees may be concerned about the company "watching" them. In such cases and in areas where IPR restrictions are strictly forbid you to digitally record study sessions, do qualitative user studies anyway and counterbalance the loss of documentation with more than one observer and with immediate notes and a summary. Especially in secure areas this methodology additionally may allow participants to be more open about their work processes, data, and tasks (see, for instance, Section 3.5.1).

ER-12: Be in Constant, Close Cooperation
To support specific domain experts with information visualization it is important to get a clear understanding of their problem domain [Pal09]. We have made good experiences with informal collaborations that helped us to get a very well-grounded and detailed knowledge about our target group: over the last three years we have talked to almost 150 domain experts, we conducted several types of studies (from pre- to post-design) and we directly worked together with the domain experts. We refer to such a process as "constant, close cooperation." The ambitious goal was to gain a deep understanding of our problem domain. From our experience, this kind of constant, close cooperation is valuable especially in large industries where problems are often highly diverse and complex (cf. Section 3.6.2). If possible, try also to be flexible and spontaneous in time in order to counterbalance busyness and time-restrictions (EC-4). Especially for investigating daily practices this helped us to observe "real" situations, both in exploratory studies and in sum-

mative long-term studies (see Section 4.3.1 and 4.3.3). We are aware that understanding all facets of a problem domain is time-intensive, however, we think that this approach of 'designing with not for the people' helps to clearly tailor solutions to the needs of a target group and to develop effective and efficient tools. Being in constant, close cooperation can help to overcome some of the pitfalls of evaluation as outlined in [Mun09] (see also our discussions in Section 3.6.2 and our various descriptions of informal studies, e. g., in Section 5.1.1).

7.3.4 Results

ER-13: The Magic Metric: Money
In industrial settings the benefits of a new tool are often measured in terms of cost savings. These savings are closely related to other metrics used in information visualization evaluation such as insights [SNLD06] or errors [CC00]. However, in an industry setting, the most related one may be time (again, time = money). Important quality metrics for stakeholders include such things as decisions per hour [MA08] or found errors per day (cf. our solutions on trace analysis in Chapter 5). Reporting the results of your study and presenting evidence that your tool can lead to measurable benefits in terms of such metrics may be very important if you want to convince the stakeholders (see EC-8). While studies that measure these metrics may not always be able to get at the research questions you are interested in, they could be a ticket for reaching more domain experts and studying your solutions in-depth in real working environments. As a best practice example you can refer to our comparative study between MostVis and engineers' current tools for browsing specification catalogs (cf. Section 4.1.3). Proving that MostVis was significantly faster for a set of predefined user-tasks was very convincing to our stakeholders. In doing so, we got the opportunity to tightly integrate MostVis into a current software environment that is subject to strict access regulations (EC-8). Conducting a quantitative user study was therefore our ticket to reaching many end users (EC-4) with real data (EC-2) in real environments (EC-1) and opened new possibilities for future long-term and more in-depth studies.

ER-14: Factor in High Skill with Current Techniques
When comparing traditional to new tools, one must consider that participants may have become very skilled with current techniques (see EC-5) and factor in learning time and potential reluctantness towards a new tool as these factors can initially distort a comparative evaluation [SP06].

ER-15: Clarify Publishing Conditions Upfront
If your main goal is to publish your work, make concrete agreements with your company upfront and preferably not just verbally. Make clear what you are allowed to write about, how or if you need to anonymize your results, what pictures (if any) you are allowed to include, and find out if the company requires you to submit your write-up for internal review first.

7.4 Summary and Discussion

In this chapter, I summarized some challenges and recommendations for working with a large company on information visualization evaluation. Both challenges and recommendations are derived from our three-year body of work involving prototypes presented in this thesis and their development processes.

In general, the chapter did not focus on the various advantageous aspects that conducting research in cooperation with a large company can exhibit but provides other researchers with the more valuable challenges and recommendations. Nevertheless, we definitely want to encourage research cooperations with large companies for several reasons: As already mentioned, large companies provide a many interesting challenges and complex real world datasets for information visualization research. In addition, although deployment and evaluation might be a long and laborious process, there are good chances that valuable solutions will be approved and integrated into real working environments (cf., for instance, MostVis). Thus, domain experts can benefit from dedicated information visualization solutions and researchers in return can investigate their systems under realistic circumstances [SP06]. Eventually, ‘moving research into practice’ remains one of our grand challenges [TC05]. We are convinced that closely cooperating with large companies will help us to better understand the value of information visualization.

Large parts of the challenges and recommendations presented in this chapter are based on our own experiences, i. e., experiences from BMW—one example of a large company setting. Experiences in other companies might differ or go beyond the ones we made. While the lessons we learned can serve as a reference for others who are planning information visualization evaluations within a large company context, I want to encourage other researchers to modify and extend our work through further research within this work context.

Chapter 8

Conclusions

The last chapter provides concluding remarks about the work presented in this dissertation. After a brief summary, I take a final look at the main contributions addressed in the preceding chapters and the publications evolved during this thesis. I also discuss aspects of generalizability and conclude with outlining directions for future work.

8.1 Summary

In 2006, Broy wrote: *"The increase of software and functionality in cars is not close to an end, in the contrary. We can expect a substantial growth in the future"* [Bro06]. This holds unchanged even four years later and is probably the strongest motivation behind the work presented in this dissertation. To cope with this challenge, I concentrated on the application of information visualization and visual analytics techniques and explored how we can harness them for gaining insights into in-car networks' complexity. All work was conducted in the context of a large automotive company, the BMW group, and can be divided into three basic components:

First, we[1] reviewed automotive engineering literature and conducted a variety of field studies in order to get a grounded understanding of in-car network engineering, of daily practices and tasks in this domain, data analysis challenges and problems, as well as potential application areas for InfoVis/VA (Chapter 3). During the studies, we particularly identified challenges in understanding and working with large network specification datasets and with network traces, communication recordings used for analyzing and debugging in-car networks. In both cases, the basic problem is that current text-based tools lack in gaining insight into the various correlations and patterns hidden in the data. This information, however, is often necessary for decision making in network architecture but also in finding errors of implemented in-car networks.

---

[1] The usage of "we" in this final chapter loosely refers to all people mentioned in the previous chapters.

Second, based on this analysis we identified different groups of network architects and trace analysts at BMW and collaboratively designed a variety of visualization prototypes, five of them resulting in closely integrated systems (MostVis, RelEx, AutobahnVis, and Cardiogram). Within the scope of this thesis, I discussed nine of these prototypes as design studies, with a focus on the domain utility they can add to current automotive engineering practices. We particularly visualized hierarchical and network structures of in-car specification data (Chapter 4), time-based information in traces (Chapter 5) and investigated how 3d visualizations can contribute to a better understanding of spatial aspects (Chapter 6). By evaluating our prototypes with domain experts we found that our approaches were able to speed up searching and browsing large specification documents (MostVis), support engineers in handling masses of traces (Cardiogram) or gaining novel and valuable insights into both specification data (in particular, RelEx) and trace files (in particular, AutobahnVis). The presented prototypes are:

- **MostVis (hierarchical visualization, integrated, Section 4.1):** MostVis is a tool representing MOST specification catalogs containing usually up to 40,000 hierarchically ordered entries. A comparative study with current tools showed that MostVis is significantly faster for searching and browsing these catalogs. MostVis will be incorporated in the next version of the company's official in-car network database BNE 2.0.

- **WiKeVis (network visualization, Section 4.2):** WiKeVis is a tool providing insights into correlations in functional network specifications. It is based on hierarchical clustered network visualization. During the WiKeVis project, we experienced several obstacles in user-centered design processes in our domain that finally led to a collaboration breakdown. We used the lessons learned for designing later tools and to inform our challenges and recommendations in Chapter 7.

- **RelEx (network visualization, integrated, Section 4.3):** RelEx is a network visualization of physical specifications based on a combination of matrix, node-link and traditional automotive engineering representations. We evaluated RelEx with domain experts over a period of five weeks and showed the tool's strong benefits by examples that helped to gain insights supporting decision making in physical network partitioning. RelEx is still used by our participants and by colleagues of them, will be extended in the near future to support more features and is planned to be integrated in the BNE 2.0 similar to MostVis.

- **VisTra (time-based visualization, Section 5.2):** VisTra is a visualization tool for representing trace files and was the very first visualization tool we built. VisTra utilizes a re-computation of correlations in the network in order to pre-filter traces and subsequently represents the results in a treemap/messages sequence chart dual view approach. Designing, evaluating and discussing VisTra with domain engineers, provided us with a variety of valuable insights. These insights helped us to come up with the design recommendations presented in Section 5.1.4 and in designing later tools.

- **AutobahnVis (time-based visualization, integrated, Section 5.3):** AutobahnVis is a scatterplot-like visualization showing all recorded messages of a trace on a zoomable timeline. In a six-week test phase with automotive network analysis experts we found anecdotal evidence that AutobahnVis can provide novel insights into traces and help engineers in tracking down complex errors. After the study, our participants kept using the tool and the core developers of the company's main trace analysis tool, Carmen, currently plan to directly embed AutobahnVis with their tool.

- **Cardiogram (state-machine analysis and time-based visualization, integrated, Section 5.4):** Cardiogram is an approach based on externalizing engineers' analysis knowledge into state machines and in doing so making it available for a wide range of users. Our one-year test phase showed that using this approach for automatic trace analysis can help to analyze the enormous masses of traces instead of just pick out sample tests. An additional time-based visualization that we provided helps to explore the results of the automatic data analysis and revealed also strong benefits in a eight-week user study. Similar to AutobahnVis, Cardiogram is still used by our participants, has started to be used by other test engineers in the company, and will be incorporated into the next version of the Carmen software by its developers.

- **ProgSpy2010 (time-based visualization, integrated, Section 5.5):** ProgSpy2010 is the direct predecessor of a tool, ProgSpy, used by engineers to analyze the process of uploading software to ECUs. It recently superseded the old tool and is now used by a group of approximately 15 engineers. Along with a textual interface that we designed exactly according to the demands of our target users, we also integrated a novel visual representation similar to Cardiogram's visualization. We evaluated ProgSpy2010 with our lead analyst and found him gaining novel insights that turned out to be of high value for his work. These examples also changed his mind on visualization.

- **Autobahn3D (3d visualization, Section 6.2):** Autobahn3D is an extension of our AutobahnVis approach with an additional 3d-vehicle view. Qualitative feedback of trace analysts indicated that an additional 3d view can help to correlate electronic traces to mechanical behavior. For a close and productive integration into daily work practices, however, current restrictions in available datasets have to be overcome first.

- **Car-x-ray (3d visualization, Section 6.3):** Car-x-ray, our second 3d prototype, bridges the gap between mechanical and electronic in-car network information by surrounding a 3d vehicle model with a frame representing abstract, hierarchical clustered information. With two case studies, we showed and evaluated how this approach can be used to visualize functional network specifications integrating spatial aspects and trace data revealing real communication paths "through" the car.

By studying all our tools with automotive experts, we particularly have learned that combining InfoVis techniques with familiar representation techniques from the engineering domain can be a fruitful approach. Retaining existing mental models helps engineers in understanding and using the tools and can lead to higher acceptance (see also [Spe07]). Furthermore, we have learned that simple visualizations with an immediately apparent benefit are preferred. As the work of engineers is already characterized by high complexity and time-pressure, solutions should not additionally complicate work with intricate visualizations. A crucial factor for all our prototypes was the tight integration of the tools with available data sources, other tools and existing working processes in general. Without a close integration, engineers are usually not able to use tools productively and thus, studying the long-term nature of analysis tasks is not possible. Finally, from a development process point of view, we have learned that applying user-centered and participatory design techniques in our domain is invaluable for developing tools accepted by engineers.

As a third and last part, we used our three-year experience of conducting InfoVis/VA research within a large company setting and formulated general implications for InfoVis/VA research in such a context. We found this kind of collaboration very fruitful as we got the opportunity to work with real data, solve real world problems and study our solutions under realistic circumstances. During this process, however, we also encountered various challenges in evaluating our tools stemming from the specific characteristics a large company context entails. To share our experience with other researchers and practitioners, we categorized large company-specific aspects of evaluating InfoVis/VA tools into a set of challenges as well as a collection of recommendations that can help conducting research in similar contexts (Chapter 7).

8.2 Main Contributions

According to the previous section and to our outlines in Section 1.4, the dissertation's main contributions can be summarized as following:

1. **Field analysis and problem characterization:** We analyzed the field of in-car communication network engineering from an InfoVis/VA point of view. We discussed datasets used by engineers, related tasks and challenges in working with them (cf., for instance, Munzner's call for more problem characterization contributions in InfoVis/VA [Mun09]).
2. **Design implications:** Based on our analysis, we derived several implications for designing and developing InfoVis/VA tools in this domain, including general implications (cf. Section 3.6.2) but also sub-domain specific recommendations (cf., most importantly, Section 5.1.4 and Section 6.1.3). We hope that other researchers find guidance in these considerations when they plan to develop InfoVis/VA tools in this but also in similar or other domains (see below).
3. **Nine design studies:** The nine design studies presented in this dissertation (cf. Chapter 4–6) can be seen as a first set of visualization studies in the domain of in-car network engineering. Especially our systems MostVis, RelEx, AutobahnVis and Cardiogram shall serve as best practice examples that helped us to solve real world, automotive electronic engineering problems with the help of InfoVis/VA techniques.
4. **Evaluating InfoVis/VA in a large company context:** The set of nine challenges and fifteen recommendations we have provided in Section 7.2 and 7.3 can serve as a reference for other researchers and practitioners planning to study their InfoVis/VA tools in large company settings in general.

8.3 Main Publications

Key aspects of this dissertation have been published or submitted for review in the peer reviewed papers below (Table 8.1). After each reference, I note the prototype(s) which are described in the publication and the thesis' section(s) in which material is used.

Publication	Prototype	Section
M. Sedlmair, C. Bernhold, D. Herrscher, S. Boring, and A. Butz. Mostvis: An interactive visualization supporting automotive engineers in most catalog exploration. In *Proceedings of the International Conference Information Visualisation (IV'09)*, pages 173–182, Los Alamitos, CA, USA, 2009. IEEE Computer Society.	MostVis	4.1
M. Sedlmair, W. Hintermaier, K. Stocker, T. Büring, and A. Butz. A dual-view visualization of in-car communication processes. In *Proceedings of the International Conference on Information Visualization (IV'08)*, pages 157–162, Los Alamitos, CA, USA, 2008. IEEE Computer Society.	VisTra	5.2
M. Sedlmair. MSCar: Enhancing Message Sequence Charts with Interactivity for Analysing (Automotive) Communication Sequences. In *Proceedings of the International Workshop on the Layout of (Software) Engineering Diagrams (LED'08)*. 2008. Electronic Communications of the EASST Vol. 13. Article 7.	VisTra (parts)	5.2
M. Sedlmair, B. Kunze, W. Hintermaier, and A. Butz. User-centered Development of a Visual Exploration System for In-Car Communication. In *Proceedings of the International Symposium on Smart Graphics (SG'09)*, pages 105–116, Berlin, Germany, 2009. Springer-Verlag.	AutobahnVis (Early)	5.3
M. Sedlmair, P. Isenberg, D. Baur, M. Mauerer, C. Pigorsch, and A. Butz. Cardiogram: Visual Analytics for Automotive Engineers. To appear in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*, 2011. *To appear.*	Cardiogram	5.4
M. Sedlmair, K. Ruhland, F. Hennecke, A. Butz, S. Bioletti, and C. O'Sullivan. Towards the Big Picture: Enriching 3D Models with Information Visualisation and Vice Versa. In *Proceedings of the International Symposium on Smart Graphics (SG'09)*, pages 27–39, Berlin, Germany, 2009. Springer-Verlag.	Autobahn3D	6.2
M. Sedlmair, P. Isenberg, D. Baur, and A. Butz. Evaluating Information Visualization in Large Companies: Challenges, Experiences and Recommendations. In *Proceedings of CHI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV'10)*, pages 79–86, New York, NY, USA, 2010. ACM Press. *Best Paper Award, Extension for Journal Paper (Information Visualization, Palgrave Macmillan) in submission.*	AutobahnVis & Autobahn3D (as examples)	7

Table 8.1: Main publications describing key aspects of this thesis.

8.4 Generalizability

Over the last years, an increasing number of InfoVis/VA researchers called for studying our tools under more realistic circumstances in order to better understand their "real world" utility (cf., for instance, [Pla04, TC05, SP06, Car08b, Mun09]). In this dissertation, I followed these calls and focused on one specific "real world" context, namely studying InfoVis/VA in a large automotive company. To address reality aspects, most of our studies focused on getting a qualitative, rich and domain-specific understanding and not on deriving statistically significant, i. e., generalizable, results (cf. [McG95]).

Nevertheless, we definitely think that parts of our results can help other researchers, designers and practitioners beyond the field of automotive in-car network engineering. Chapter 7, for instance, clearly followed this line and provide our experience on deploying and evaluating InfoVis/VA to large company settings in general (contribution 4). While our experience solely relies on conducting research at one such large company, we hope that others will try out our recommendations in different settings and modify and extend our work in order to generalize findings. Moreover, we also think that our design implications and studies (contribution 2 and 3) can be interesting and valuable references for a broader audience. From collaborative work with the aircraft industry we know, for instance, that challenges and problems in this domain are very similar to the ones addressed in this thesis. The same may be true for all kinds of engineering systems with embedded sensor networks, such as trains, ships, but also intelligent houses. Furthermore, we also think that—similar to our evaluation considerations—our design implications and recommendations may provide helpful guidance for others planning to develop InfoVis/VA in large company settings in general.

8.5 Future Work

There are many interesting directions for future work stemming from this thesis. In the following, I first look at future directions of applying InfoVis/VA within the domain of automotive electronic engineering and, then, more generally at future work of conducting InfoVis/VA research in large company settings.

8.5.1 Within the Automotive Domain

Enhancing solutions: The design studies presented in this thesis (Chapter 4–6) are prototypes which we built for the purpose of studying InfoVis/VA within our target domain. Obviously, this implies room for improvement and we definitely believe that our prototypes can benefit from further design iterations and extensions. Each design study section ended with a discussion providing our ideas on further improvements.

Overcoming technical and organizational restrictions: During our studies we were confronted with several technical and organizational restrictions such as incomplete datasets and non-automated processes. To allow for a seamless integration of visualization approaches and for even more powerful approaches, overcoming these restrictions is essential. However, this is future work that cannot be done by researchers but must be addressed by the companies themselves. For a detailed discussion of current restrictions please refer to the respective design study sections and to Section 5.6 and 6.4.

Long-term studies: In this thesis we followed the approach of designing, deploying and integrating visualization software into engineers' daily work practices. We conducted a variety of user studies with our target group and in doing learned much about applying InfoVis/VA in our domain. We also did several long-term studies with our tools, however, except from Cardiogram's automation technique, these studies usually lasted only several weeks. In line with

Shneiderman and Plaisant [SP06, Pla04], we think that longer studies over months or even years can provide richer insights into how our tools are actually used by target users. The fact that some of the design studies presented in this thesis have already been installed in engineers' daily practices—which is an inevitable requirement for doing long-term studies in our domain—allows for studying them over a longer period of time. We hope that other researchers may take this opportunity and study the proposed solutions or adaptations of them in more depth.

Novel systems and comparison of approaches: The prototypes presented in this thesis are only a first collection of design studies in this particular domain. We addressed a sub-set of problems in our domain, closely collaborated with several interested engineers and provided a set of visualizations in a field where InfoVis/VA was virtually not existent before. Based on this limitation, we want to encourage other researchers and designers for two kinds of future work within our domain. First, we hope that they will build upon our domain analysis and problem characterization (Chapter 3, contribution 1) to design their own tools and to compare their solutions with ours. Second, as the automotive electronics domain is a large business, we definitely think that there are other use-cases, challenges and/or target groups that can benefit from InfoVis/VA technologies but that we have not addressed yet. For instance, an aspect beyond the scope of this thesis but frequently addressed by our engineers, is the comparison of specified network (theoretical perspective) and actual behavior (traces, actual perspective).

From point-solutions to general-purpose systems: As described above, in the scope of this thesis, we have concentrated on a first set of best practice examples harnessing InfoVis/VA techniques to support in-car network engineering. These design studies are selective solutions for problems we identified. In a next step, it would be worth to look at how our findings and prototypes complement and how they could be combined in larger, even more powerful tools. A first idea, for instance, would be to investigate how AutobahnVis (Section 5.3) and Cardiogram ((Section 5.4)) can be integrated into one single tool. Feedback of our lead users indicated that these two different approaches may complement well and—as both visual interfaces are based on a horizontal time line—may provide valuable insights of connections between timings in state machine analysis and raw data (cf. R-2 in Section 5.1.4).

8.5.2 Beyond the Automotive Domain

Continue InfoVis/VA research in large company settings: Beyond the scope of the automotive electronics domain, we have started with categorizing our experience into a first set of challenges and recommendations of conducting InfoVis/VA research within the context of large companies (Chapter 7). At this point we want to echo calls of other researchers for more InfoVis/VA research under realistic circumstances (cf., for instance, [Pla04, TC05, SP06, Car08b, Mun09, SND05, IZCC08]) and in particular want to recommend collaborations with large companies. Large companies are a major part of our real world and provide a variety of real world datasets, tasks and challenges that can benefit from InfoVis/VA techniques. We hope that other researchers will continue our path and study their tools in large company contexts. We are curious about how their experiences might differ from our work and how the challenges and recommendations presented in this thesis can be modified and extended accordingly.

8.6 Conclusion

With this dissertation, I hope to have provided novel insights into one of our major challenges: “moving research into practice” [TC05]. During the three years I worked at BMW, I observed many automotive engineers changing their mind on InfoVis/VA from “drawing nice pictures” to a valuable tool for their daily work. Several of the tools presented in this thesis are still used by our target users, are currently adopted and/or funded for close integration or will be used for conceptually re-designing novel domain software. To conclude with, I again want to encourage other InfoVis/VA researchers to try this form of field research. Even though it may be more difficult to do than lab-based research, I am convinced that it can provide us with very rich, realistic insights on the value of our tools and I am sure that this type of applied work can be very rewarding for the InfoVis/VA community in general.

Bibliography

[AA06] N. Andrienko and G. Andrienko. *Exploratory analysis of spatial and temporal data: a systematic approach.* Springer-Verlag, Berlin, Germany, 2006.

[Alb04] A. Albert. Comparison of event-triggered and time-triggered concepts with regard to distributed control systems. In *Embedded World*, pages 235–252, 2004.

[AMM$^+$08] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):47–60, 2008.

[And06] K. Andrews. Evaluating information visualisations. In *Proceedings of the AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV'06)*, pages 1–5, New York, NY, USA, 2006. ACM Press.

[Aqu10] Aquintos GmbH. PREEvision, 2010. http://www.aquintos.com/, last visited: 05/10.

[AS05] R.A. Amar and J.T. Stasko. Knowledge precepts for design and evaluation of information visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):432–442, 2005.

[Ass10] Association for Standardisation of Automation and Measuring Systems (ASAM). Field Bus Exchange Format (FIBEX), 2010. http://www.asam.net, last visited: 05/10.

[Aut] Autosar Development Partnership. Autosar: Automotive open system architecture. http://www.autosar.org/, last visited: 05/10.

[Aut10] Autobild Online. Toyota Rückruf, 2010. http://www.autobild.de/toyota-rueckruf_1123062.html, last visited: 05/10.

[BÅPL03] I. Boivie, C. Åborg, J. Persson, and M. Loefberg. Why usability gets lost or usability in in-house software development. *Interacting with Computers*, 15(4):623–639, 2003.

[BBR$^+$05] A. Bauer, M. Broy, J. Romberg, B. Schätz, P. Braun, U. Freund, N. Mata, R. Sandner, and D. Ziegenbein. AutoMoDe – notations, methods, and tools for model-based development of automotive software. In *Proceedings of the SAE 2005 World Congress*, Detroit, MI, USA, 2005. Society of Automotive Engineers.

[BCS04] T. Bladh, D.A. Carr, and J. Scholl. Extending Tree-Maps to Three Dimensions: A Comparative Study. In *Proceedings of the Asia Pacific Conference on Computer-Human Interaction (APCHI'04)*, pages 50–59, Berlin, Germany, 2004. Springer-Verlag.

[BD05] M. Balzer and O. Deussen. Voronoi treemaps. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'05)*, pages 7–14, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[Bed01] B.B. Bederson. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'01)*, pages 71–80, New York, NY, USA, 2001. ACM Press.

[BEH$^+$01] U. Brandes, M. Eiglsperger, I. Herman, M. Himsolt, and M. Marshall. GraphML Progress Report. In *Proceedings of the International Symposium on Graph Drawing (GD'01)*, pages 109–112, Berlin, Germany, 2001. Springer-Verlag.

[Ber83] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps, (Translation from French 1967 edition: W.J. Berg)*. University of Wisconsin Press, Madison, WI, USA, 1983.

[Ber00] H.R. Bernard. *Social Research Methods: Qualitative and Quantitative Approaches*. Sage, Thousand Oaks, CA, USA, 2000.

[BG91] R. Bosch GmbH. Control Area Network specification version 2, 1991. http://www.semiconductors.bosch.de/pdf/can2spec.pdf, last visited: 07/10.

[BG07a] R. Bosch GmbH. *Autoelektrik/Autoelektronik, 5. Auflage*. Vieweg Verlag, Braunschweig, Germany, 2007.

[BG07b] R. Bosch GmbH. *Kraftfahrzeugtechnisches Taschenbuch. 26. Auflage*. Vieweg Verlag, Braunschweig, Germany, 2007.

[BGB02] R. Bosch GmbH and H. Bauer. Adaptive Fahrgeschwindigkeitsregelung ACC. *R. Bosch GmbH, Technische Unterrichtungen*, 2002.

[BH98] H. Beyer and K. Holtzblatt. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann, San Francisco, CA, USA, 1998.

[BJL02] C. Buchheim, M. Jünger, and S. Leipert. Improving Walker's Algorithm to Run in Linear Time. In *Proceedings of the International Symposium on Graph Drawing (GD'02)*, volume 2528 of *LNCS*, Berlin, Germany, 2002. Springer-Verlag.

[BK08] E. Bringmann and A. Krämer. Model-based Testing of Automotive Systems. In *Proceedings of the International Conference on Software Testing, Verification, and Validation*, pages 485–493, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[BKS04] K. Bødker, F. Kensing, and J. Simonsen. *Participatory IT design: designing for business and workplace realities.* The MIT Press, Cambridge, MA, USA, 2004.

[BMT97] T. Biedl, B. Madden, and I. Tollis. The three-phase method: A unified approach to orthogonal graph drawing. In *Proceedings of the International Symposium on Graph Drawing (GD'97)*, pages 391–402, Berlin, Germany, 1997. Springer-Verlag.

[BN01] T. Barlow and P. Neville. A Comparison of 2-D Visualizations of Hierarchies. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'01)*, pages 131–138, Los Alamitos, CA, USA, 2001. IEEE Computer Society.

[BNRS08] J.O. Bak, K. Nguyen, P. Risgaard, and J. Stage. Obstacles to usability evaluation in practice: A survey of software development organizations. In *Proceedings of the Nordic Conference on Human-Computer Interaction (NordiCHI'08)*, pages 23–32, New York, NY, USA, 2008. ACM Press.

[BO04] A.L. Barabasi and Z.N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

[BOR09] M. Büscher, J. O'Neill, and J. Rooksby. Designing for Diagnosing: Introduction to the Special Issue on Diagnostic Work. *Computer Supported Cooperative Work (CSCW)*, 18(2-3):109–128, 2009. Springer.

[BR03] N. Boukhelifa and P.J. Rodgers. A model and software system for coordinated and multiple views in exploratory visualization. *Information Visualization*, 2(4):258–269, 2003. Palgrave Macmillan.

[BR05] M. Broy and A. Rausch. Das neue V-Modell® XT. *Informatik-Spektrum*, 28(3):220–229, 2005.

[Bro06] M. Broy. Challenges in automotive software engineering. In *Proceedings of the International Conference on Software Engineering (ICSE'06)*, pages 33–42, New York, NY, USA, 2006. ACM Press.

[BSN07] H. Balzer, B. Stein, and O. Niggemann. Diagnose in verteilten automotiven Systemen. In *Paderborner Workshop über Entwurf mechatronischer Systeme*, pages 243–254, 2007.

[BW03] U. Brandes and D. Wagner. Analysis and Visualization of Social Networks. *Special Issue on Graph Drawing Software, Springer Series in Mathematics and Visualization*, pages 321–340, 2003.

[BWK00] M.Q.W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Conference on Advanced Visual Interfaces (AVI'00)*, pages 110–119, New York, NY, USA, 2000. ACM Press.

[Car08a] S.K. Card. Information Visualization. In *A. Sears, J.A. Jacko (eds.): The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*, pages 509–543. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 2008.

[Car08b] S. Carpendale. Evaluating information visualizations. In *A. Kerren, J. Stasko, J.D. Fekete, C. North (eds.): Information Visualization: Human-Centered Issues and Perspectives*, volume 4950, pages 19–45. Berlin, Germany, 2008. Springer LNCS Volume 4950.

[CC00] C. Chen and M.P. Czerwinski. Empirical evaluation of information visualizations: an introduction. *International Journal on Human-Computer Studies*, 53(5):631–635, 2000. ELsevier.

[CCY$^+$03] G. Convertino, J. Chen, B. Yost, Y.S. Ryu, and C. North. Exploring context switching and cognition in dual-view coordinated visualizations. In *Proceedings of the International Conference on Coordinated and Multiple Views In Exploratory Visualization (CMV'03)*, pages 57–66, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[Che04] C. Chen. *Information visualization: Beyond the horizon*. Springer-Verlag, Berlin, Germany, 2004.

[Chi00] E.H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'00)*, pages 69–75, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

[CK98] J.V. Carlis and J.A. Konstan. Interactive visualization of serial periodic data. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'98)*, pages 29–38, New York, NY, USA, 1998. ACM Press.

[Cle85] W.S. Cleveland. *The elements of graphing data*. Wadsworth Publications, Belmont, CA, USA, 1985.

[CMS99] S.K. Card, J.D. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, San Francisco, CA, USA, 1999.

[CON08] M. Christie, P. Olivier, and J.M. Normand. Camera control in computer graphics. *Computer Graphics Forum (Proc. EuroVis'08)*, 27(8):2197–2218, 2008.

[CR98] E.H. Chi and J.T. Riedl. An operator interaction framework for visualization systems. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'98)*, pages 63–70, Los Alamitos, CA, USA, 1998. IEEE Computer Society.

[DAES06] R.M. Daoud, H.H. Amer, H.M. Elsayed, and Y. Sallez. Ethernet-based car control network. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE'06)*, pages 1031–1034, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[Daf95] R.L. Daft. *Organization Theory and Design*. West St. Paul, Minneapolis, MN, USA, 1995.

[Das] Dassault Systèmes. Catia. http://www.3ds.com/de/products/catia/, last visited: 05/10.

[DBETT98] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall, Upper Saddle River, NJ, USA, 1998.

[DFA04] A. Dix, J. Finlay, and G.D. Abowd. *Human-computer interaction*. Prentice Hall, London, New York, and others, 2004.

[DFF$^+$96] F. Dai, W. Felger, T. Frühauf, M. Göbel, D. Reiners, and G. Zachmann. Virtual prototyping examples for automotive industries. In *Proceedings of the Virtual Reality World Conference*, pages 13–15, Stuttgart, Germany, 1996.

[DGH03] H. Doleisch, M. Gasser, and H. Hauser. Interactive Feature Specification for Focus+Context Visualization of Complex Simulation Data. In *Proceedings of the IEEE TCVG-Eurographics Symposium on Visualization (VisSym'03)*, pages 239–248, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[Dha96] M.N. Dhaubhadel. Review: CFD Applications in the Automotive Industry. *Journal of Fluids Engineering*, 118(4):647–653, 1996.

[DMG$^+$04] H. Doleisch, M. Mayer, M. Gasser, R. Wanker, and H. Hauser. Case Study: Visual Analysis of Complex, Time-Dependent Simulation Results of a Diesel Exhaust System. In *Proceedings of the IEEE TCVG-Eurographics Symposium on Visualization (VisSym'04)*, pages 91–96, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[Dru88] P.F. Drucker. The coming of the new organization. *Harvard Business Review*, 66(1):45–53, 1988.

[DW99] W. Dröschel and M. Wiemers. *Das V-Modell 97*. Oldenbourg Verlag, München, Germany, 1999.

[Ecl] Eclipse Foundation. Eclipse. http://www.eclipse.org/, last visited: 05/10.

[ED06] G. Ellis and A. Dix. An explorative analysis of user evaluation studies in information visualisation. In *Proceedings of the AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV'06)*, New York, NY, USA, 2006. ACM Press.

[EDG+08] N. Elmqvist, T.N. Do, H. Goodell, N. Henry, and J.D. Fekete. ZAME: Interactive large-scale graph visualization. In *Proceedings of the IEEE Pacific Visualization Symposium*, pages 215–222, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[ET08] N. Elmqvist and P. Tsigas. A taxonomy of 3d occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1095–1109, 2008.

[FB95] G.W. Furnas and B.B. Bederson. Space-scale diagrams: Understanding multiscale interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'95)*, pages 234–241, New York, NY, USA, 1995. ACM Press.

[FGW02] U.M. Fayyad, G.G. Grinstein, and A. Wierse. *Information visualization in data mining and knowledge discovery*. Morgan Kaufmann, San Francisco, CA, USA, 2002.

[FP02] J.D. Fekete and C. Plaisant. Interactive information visualization of a million items. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, pages 117–124, Los Alamitos, CA, USA, 2002. IEEE Computer Society.

[Fra98] A.U. Frank. Different types of "Times" in GIS. In *M.J. Egenhofer, R.G. Golledge (eds.): Spatial and temporal reasoning in geographic information systems*, pages 40–62. Oxford University Press, New York, NY, USA, 1998.

[Fre00] L.C. Freeman. Visualizing social networks. *Journal of social structure*, 1(1):4, 2000.

[Fug93] A. Fuggetta. A classification of case technology. *IEEE Computer*, 26(12):25–38, 1993.

[FvWSN08] J.D. Fekete, J. van Wijk, J. Stasko, and C. North. The value of information visualization. In *A. Kerren, J. Stasko, J.D. Fekete, C. North (eds.): Information Visualization: Human-Centered Issues and Perspectives*, volume 4950, pages 1–18. Berlin, Germany, 2008. Springer LNCS Volume 4950.

[GBS+01] A. Goel, C.A. Baker, C.A. Shaffer, B. Grossman, W.H. Mason, L.T. Watson, and R.T. Haftka. VizCraft: A Problem-Solving Environment for Aircraft Configuration Design. *Computing in Science & Engineering*, 3(1):56–66, 2001.

[GFC05] M. Ghoniem, J.D. Fekete, and P. Castagliola. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization*, 4(2):114–135, 2005. Palgrave Macmillan.

[GHP02] E. Gery, D. Harel, and E. Palachi. Rhapsody: A complete life-cycle model-based development system. In *Integrated Formal Methods*, pages 1–10, Berlin, Germany, 2002. Springer-Verlag.

[GK03a] V. González and A. Kobsa. Benefits of Information Visualization Systems for Administrative Data Analysts. In *Proceedings of the International Conference on Information Visualization (IV'03)*, pages 331–336, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[GK03b] V González and A. Kobsa. A workplace study of the adoption of information visualization systems. In *Proceedings of the International Conference on Knowledge Management (I-KNOW'03)*, pages 92–102, 2003.

[Gri03] K. Grimm. Software technology in an automotive company: major challenges. In *Proceedings of the International Conference on Software Engineering (ICSE'03)*, pages 498–503, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[Gru93] J. Grudin. Obstacles to participatory design in large product development organizations. In *D. Schuler, A. Namioka (eds.): Participatory Design: Principles and Practices*, pages 99–119. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1993.

[GRW$^+$00] D.L. Gresh, B.E. Rogowitz, R.L. Winslow, D.F. Scollan, and C.K. Yung. WEAVE: a system for visually linking 3-D and statistical visualizations, applied to cardiac simulation and measurement data. In *Proceedings of the IEEE Conference on Visualization (VIS'00)*, pages 489–492, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

[Grz07] A. Grzemba. *MOST: The Automotive Multimedia Network.* Franzis Verlag, Poing, Germany, 2007.

[GS77] B.G. Glaser and A.L. Strauss. *The discovery of grounded theory: Strategies for qualitative research.* Aldine, Chicago, IL, USA, 1977.

[GvdW05] A. Grzemba and H.C. von der Wense. *Lin-bus.* Franzis Verlag, Poing, Germany, 2005.

[GWB07] T. Grossman, D. Wigdor, and R. Balakrishnan. Exploring and reducing the effects of orientation on text readability in volumetric displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*, pages 483–492, New York, NY, USA, 2007. ACM Press.

[HA07] Jeffrey Heer and Maneesh Agrawala. Design considerations for collaborative visual analytics. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST'07)*, pages 171–178, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[HB05] J. Heer and D. Boyd. Vizster: Visualizing online social networks. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'05)*, pages 32–39, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[HBE93] C.G. Healey, K.S. Booth, and J.T. Enns. Harnessing Preattentive Processes for Multivariate Data Visualization. In *Proceedings Graphics Interface (GI'93)*, pages 107–117, Toronto, Canada, 1993.

[HBH09] K. Holtzblatt, J. Barr, and L. Holtzblatt. Driving user centered design into it organizations: is it possible? In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI'09)*, pages 2727–2730, New York, NY, USA, 2009. ACM Press.

[HBN08] D. Hackenberg, H. Brunst, and W. Nagel. Event tracing and visualization for Cell Broadband Engine systems. In *Proceedings of the International Euro-Par Conference on Parallel Processing (Euro-Par'08)*, pages 172–181, Berlin, Germany, 2008. Springer-Verlag.

[HCL05] J. Heer, S.K. Card, and J.A. Landay. Prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'05)*, pages 421–430, New York, NY, USA, 2005. ACM Press.

[HCvW07] D. Holten, B. Cornelissen, and JJ. van Wijk. Trace Visualization Using Hierarchical Edge Bundles and Massive Sequence Views. In *Proceedings of the International Workshop on Visualizing Software for Understanding and Analysis (VisSoft'07)*, pages 47–54, 2007.

[Hei05] H. Heinecke. Automotive System Design–Challenges and Potential. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '05)*, pages 656–657, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[HF06] N. Henry and J.D. Fekete. Matrixexplorer: a dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):677–684, 2006.

[HFM07] N. Henry, J.D. Fekete, and M.J. McGuffin. NodeTrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309, 2007.

[HHW$^+$02] S. Havre, E. Hetzler, P. Whitney, L. Nowell, B.P.N. Div, and W. A. Richland. ThemeRiver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[HJ93] K. Holtzblatt and S. Jones. Contextual inquiry: A participatory technique for system design. In *D. Schuler, A. Namioka (eds.): Participatory Design: Principles and Practices*, pages 177–210. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1993.

[Hol06] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[HvW09] D. Holten and J.J. van Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum (Proc. EuroVis'09)*, 28(3):983–990, 2009. John Wiley & Sons.

[HZR$^+$92] K.P. Herndon, R.C. Zeleznik, D.C. Robbins, D.B. Conner, S.S. Snibbe, and A. Van Dam. Interactive shadows. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'92)*, pages 1–6, New York, NY, USA, 1992. ACM Press.

[IC07] P. Isenberg and S. Carpendale. Interactive Tree Comparison for Co-located Collaborative Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1232–1239, 2007.

[Ise09] P. Isenberg. *Collaborative Information Visualization in Co-located Environments*. PhD thesis, University of Calgary, Canada, 2009.

[ISO04] ISO (International Organization for Standardization). Road Vehicles—Unified Diagnostic Services (UDS), Part 1, 2004. Geneva, Switzerland, Tech. Rep. ISO/DIS 14229-1.

[ISS02] R. Isermann, R. Schwarz, and S. Stolzl. Fault-tolerant drive-by-wire systems. *IEEE Control Systems Magazine*, 22(5):64–81, 2002.

[IZCC08] P. Isenberg, T. Zuk, C. Collins, and S. Carpendale. Grounded evaluation of information visualizations. In *Proceedings of the CHI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV'08)*, pages 55–60, New York, NY, USA, 2008. ACM Press.

[JJJF07] M. Jern, S. Johansson, J. Johansson, and J. Franzen. The GAV Toolkit for Multiple Linked Views. In *Proceedings of the International Conference on Coordinated and Multiple Views In Exploratory Visualization (CMV'07)*, pages 85–97, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[JMM^{+}06] C. Johnson, R. Moorhead, T. Munzner, H. Pfister, P. Rheingans, and T.S. Yoo. NIH-NSF Visualization Research Challenges Report. 2006. IEEE Press.

[JMWU91] R. Jeffries, J.R. Miller, C. Wharton, and K. Uyeda. User interface evaluation in the real world: a comparison of four techniques. In *Proceeding of the SIGCHI Conference on Human Factors in Computing Systems (CHI'91)*, pages 119–124, New York, NY, USA, 1991. ACM Press.

[JSH^{+}08] A. Jardosh, P. Suwannatat, T. Höllerer, E. Belding, and K. Almeroth. SCUBA: Focus and Context for Real-time Mesh Network Health Diagnosis. In *Proceedings of the International Conference on Passive and Active Network Measurement (PAM'08)*, pages 162–171, Berlin, Germany, 2008. Springer-Verlag.

[KBGE09] S. Koch, H. Bosch, M. Giereth, and T. Ertl. Iterative Integration of Visual Insights during Patent Search and Analysis. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST'09)*, pages 203–210, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[Kei02] D.A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[KHI^{+}03] R. Kosara, C.G. Healey, V. Interrante, D.H. Laidlaw, and C. Ware. Thoughts on user studies: Why, how, and when. *IEEE Computer Graphics and Applications*, 23(4):20–25, 2003.

[KHSE98] S. Kuschfeldt, M. Holzner, O. Sommer, and T. Ertl. Efficient visualization of crash-worthiness simulations. *IEEE Computer Graphics and Applications*, 18(4):60–65, 1998.

[KLK^{+}05] N. Kumar, N. Lolla, E. Keogh, S. Lonardi, C. Ratanamahatana, and L. Wei. Time-series bitmaps: a practical visualization tool for working with large time series databases. In *Proceedings of the SIAM Conference on Data Mining (SDM'05)*, pages 531–535. SIAM, 2005.

[KMG^{+}07] Z. Konyha, K. Matkovix, D. Graxanin, M. Duras, J. Juric, and H. Hauser. Interactive visual analysis of a timing chain drive using segmented curve view and other coordinated views. In *Proceedings of the International Conference on Coordinated and Multiple Views In Exploratory Visualization (CMV'07)*, pages 3–15, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[KMSZ06] D.A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. Challenges in visual data analysis. In *Proceedings of the International Conference on Information Visualization (IV'06)*, pages 9–16, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[Kof35] K. Koffka. *Principles of Gestalt psychology*. Routledge, London, England, 1935.

[Köh29] W. Köhler. *Gestalt psychology*. Liveright, New York, NY, USA, 1929.

[Kos05] T. Kosch. Technical concept and prerequisites of car-to-car communication. In *Proceedings of the European Congress and Exhibition on Intelligent Transport Systems and Services (ITS'05)*, pages 1–12, 2005.

[KPS97] H.P. Kumar, C. Plaisant, and B. Shneiderman. Browsing hierarchical data with multi-level dynamic queries and pruning. *International Journal of Human-Computers Studies*, 46(1):103–124, 1997.

[Kra06] D. Kraft. Cartronic. In *H.J. Gevatter (editor): Handbuch der Mess- und Automatisierungstechnik im Automobil*, pages 517–524. Springer-Verlag, Berlin, Germany, 2006.

[KS98] F. Kensing and J. Simonsen. MUST: A method for participatory design. *Human-Computer Interaction*, 13(2):167–198, 1998. Lawrence Erlbaum Associates.

[KSW06] J. Kruger, J. Schneider, and R. Westermann. Clearview: An interactive context preserving hotspot visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):941–948, 2006.

[KTB$^+$08] S. Kugele, M. Tautschnig, A. Bauer, C. Schallhart, S. Merenda, W. Haberl, C. Kühnel, F. Müller, Z. Wang, D. Wild, et al. COLA–The component language. *Technical Report TUM-I0714, Institut für Informatik, Technische Universität München*, 2008.

[KW01] M. Kaufmann and D. Wagner. *Drawing graphs: methods and models*. Springer-Verlag, Berlin, Germany, 2001.

[Lee99] K. Lee. *Principles of CAD/CAM/CAE Systems*. Addison-Wesley, Boston, MA, USA, 1999.

[LH02] G. Leen and D. Heffernan. Expanding automotive electronic systems. *Computer*, 35(1):88–93, 2002. IEEE Press.

[LHD99] G. Leen, D. Heffernan, and A. Dunne. Digital networks in the automotive vehicle. *Computing & Control Engineering Journal*, 10(6):257–266, 1999.

[LKL05] J. Lin, E. Keogh, and S. Lonardi. Visualizing and discovering non-trivial patterns in large time series databases. *Information Visualization*, 4(2):61–82, 2005. Palgrave Macmillan.

[LMK07] H. Lam, T. Munzner, and R. Kincaid. Overview use in multiple visual information resolution interfaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1278–1285, 2007.

[LPP+06] B. Lee, C.S. Parr, C. Plaisant, B.B. Bederson, V.D. Veksler, W.D. Gray, and C. Kotfila. TreePlus: Interactive Exploration of Networks with Enhanced Tree Layouts. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1414–1426, 2006.

[LR96] J. Lamping and R. Rao. The Hyperbolic Browser: A Focus+ Context Technique for Visualizing Large Hierarchies. *Journal of Visual Languages and Computing*, 7(1):33–55, 1996.

[LRP95] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI'95)*, pages 401–408, New York, NY, USA, 1995. ACM Press.

[MA08] S.M. McNee and B. Arnette. Productivity as a metric for visual analytics: reflections on e-discovery. In *Proceedings of the CHI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV'08)*, New York, NY, USA, 2008. ACM Press.

[Mac88] J.D. Mackinlay. Applying a theory of graphical presentation to the graphic design of user interfaces. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'88)*, pages 179–189, New York, NY, USA, 1988. ACM Press.

[Mac95] A.M. MacEachren. *How maps work*. Guilford Press, New York, NY, USA, 1995.

[Mac03] A. Mack. Inattentional blindness. *Current Directions in Psychological Science*, 12(5):180, 2003.

[Mar08] R. Marty. *Applied security visualization*. Addison-Wesley, Boston, MA, USA, 2008.

[MB95] T. Munzner and P. Burchard. Visualizing the structure of the world wide web in 3d hyperbolic space. In *Proceedings of the Symposium on Virtual Reality Modeling Language (VRML'95)*, pages 33–38, New York, NY, USA, 1995. ACM Press.

[McG95] J.E. McGrath. Methodology Matters: Doing Research in the Behavioral and Social Sciences. In *Human-Computer Interaction: Toward the Year 2000*, pages 152–169. Morgan Kaufmann, San Francisco, CA, USA, 1995.

[McR08] Russ McRee. Security Visualization: What you don't see can hurt you. *Information System Security Association Journal (ISSA)*, 6(6):38–41, 2008.

[MDH+03] A. MacEachren, X. Dai, F. Hardisty, D. Guo, and G. Lengerich. Exploring high-D spaces with multiform matrices and small multiples. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'03)*, pages 31–38, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[MGR] T. Munzner, F. Guimbretiere, and G. Robertson. Constellation: a visualization tool for linguistic queries from MindNet. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'99)*, pages 132–135, Los Alamitos, CA, USA. IEEE Computer Society.

[MHB$^+$01] R. Mores, G. Hay, R. Belschner, J. Berwanger, C. Ebner, S. Fluhrer, E. Fuchs, B. Hedenetz, W. Kuffner, A. Krüger, et al. FlexRay-The Communication System for Advanced Automotive Control Systems. SAE International, 2001.

[MHJ91] A. Malony, D. Hammerslag, and D. Jablonowski. Traceview: A trace visualization tool. *IEEE Software*, 8(5):19–28, 1991.

[MMKN08] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. LiveRAC: Interactive Visual Exploration of System Management Time-series Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*, pages 1483–1492, New York, NY, USA, 2008. ACM Press.

[Moz07] M. Mozdari. Konzept zur 3D-Visualisierung von Fahrzeugbordnetzarchitekturen. Master's thesis, Technische Universität München, Germany, 2007.

[MRC91] J.D. Mackinlay, G.G. Robertson, and S.K. Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'91)*, pages 173–176, New York, NY, USA, 1991. ACM Press.

[Mun97] T. Munzner. H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'97)*, pages 2–10, Los Alamitos, CA, USA, 1997. IEEE Computer Society.

[Mun08] T. Munzner. Process and pitfalls in writing information visualization research papers. In *A. Kerren, J. Stasko, J.D. Fekete, C. North (eds.): Information Visualization: Human-Centered Issues and Perspectives*, volume 4950, pages 134–153. Springer-Verlag, Berlin, Germany, 2008. Springer LNCS Volume 4950.

[Mun09] T. Munzner. A nested process model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, 2009.

[MVSC05] Ji-Ye Mao, Karel Vredenburg, Paul W. Smith, and Tom Carey. The state of user-centered design practice. *Communications of the ACM*, 48(3):105–109, 2005.

[ND86] D.A. Norman and S.W. Draper. *User Centered System Design: New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1986.

[NJKJ05] S. Noel, M. Jacobs, P. Kalapa, and S. Jajodia. Multiple Coordinated Views for Network Attack Graphs. In *Proceedings of the International Workshop on Visualization for Cyber Security (VizSec'05)*, pages 99–106, 2005.

[NS97] C. North and B. Shneiderman. A taxonomy of multiple window coordinations. *Univ. Maryland Computer Science Dept. Technical Report# CS-TR-3854*, 1997.

[NS00a] C. North and B. Shneiderman. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In *Proceedings of the Conference on Advanced Visual Interfaces (AVI'00)*, pages 128–135, New York, NY, USA, 2000. ACM Press.

[NS00b] C. North and B. Shneiderman. Snap-together visualization: can users construct and operate coordinated visualizations? *International Journal of Human Computer Studies*, 53(5):715–740, 2000.

[NSGS07] G.M. Namata, B. Staats, L. Getoor, and B. Shneiderman. A dual-view approach to interactive network visualization. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM'07)*, pages 939–942, New York, NY, USA, 2007. ACM Press.

[NSP97] C. North, B. Shneiderman, and C. Plaisant. Visual Information Seeking in Digital Image Libraries: The Visible Human Explorer. In *G. Becker (editor): Information in Images*. Thomson Technology Labs, 1997.

[NSSLW05] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert. Trends in automotive communication systems. *IEEE Special Issue on Industrial Communications Systems*, 93(6):1204–1223, 2005.

[Pal09] W.B. Paley. Interface and mind - a “paper lecture" about a domain-specific design methodology based on contemporary mind science. *IT-Information Technology*, 51(3):131–141, 2009.

[PBKS07] A. Pretschner, M. Broy, I.H. Kruger, and T. Stauner. Software engineering for automotive systems: A roadmap. In *Proceedings of the ICSE Workshop on Future of Software Engineering (FOSE'07)*, pages 55–71, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[PG94] S.E. Poltrock and J. Grudin. Organizational obstacles to interface design and development: two participant- observer studies. *ACM Transactions on Computer-Human Interaction*, 1(1):52–80, 1994.

[PGB03] C. Plaisant, J. Grosjean, and B.B. Bederson. SpaceTree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In *B.B. Bederson, B. Shneiderman (eds.): The craft of information visualization: readings and reflections*, pages 287–297. Morgan Kaufmann, San Francisco, CA, USA, 2003.

[PHH69] D.S. Pugh, D.J. Hickson, and C.R. Hinings. An empirical taxonomy of structures of work organizations. *Administrative Science Quarterly*, 14(1):115–126, 1969.

[Pla04] C. Plaisant. The Challenge of Information Visualization Evaluation. In *Proceedings of the Conference on Advanced Visual Interfaces (AVI'04)*, pages 109–116, New York, NY, USA, 2004. ACM Press.

[PMR+96] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. LifeLines: visualizing personal histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'96)*, pages 221–227, New York, NY, USA, 1996. ACM Press.

[PMS07] M. Pastko, A. Mackie, and M. Simon. Virtual Environments for the Use of Data Visualization. Technical report, BMW Group Research and Technology, Munich, Germany, 2007. Internal technical report.

[PR02] B. Preim and F. Ritter. Techniken zur interaktiven Hervorhebung von Objekten in medizinischen 3D-Visualisierungen. *Simulation und Visualisierung 2002*, pages 187–200, 2002.

[PRS01] J. Preece, Y. Rogers, and H. Sharp. *Beyond Interaction Design: Beyond Human-Computer Interaction.* John Wiley & Sons, New York, NY, USA, 2001.

[PS08] A. Perer and B. Shneiderman. Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*, pages 265–274, New York, NY, USA, 2008. ACM Press.

[Pur97] Helen C. Purchase. Which aesthetic has the greatest effect on human understanding? In *Proceedings of the International Symposium on Graph Drawing (GD'97)*, pages 248–261, Berlin, Germany, 1997. Springer-Verlag.

[PvW05] A.J. Pretorius and J.J. van Wijk. Multidimensional visualization of transition systems. In *Proceedings of the International Conference on Information Visualisation (IV'05)*, pages 323–328, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[PVW08] A.J. Pretorius and J.J. Van Wijk. Multiple views on system traces. In *Proceedings of the IEEE Pacific Visualization Symposium*, pages 95–102, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[RE] D. Rose and T. Ertl. Interactive Visualization of Large Finite Element Models. In *Proceedings of Workshop on Vision, Modelling, and Visualization (VMV'03)*, pages 585–592. Akademische Verlagsgesellschaft Aka GmbH.

[Rhy03] T.M. Rhyne. Does the Difference between Information and Scientific Visualization Really Matter? *IEEE Computer Graphics and Applications*, 23(3):6–8, 2003.

[RLS+96] S.F. Roth, P. Lucas, J.A. Senn, C.C. Gomberg, M.B. Burks, P.J. Stroffolino, J.A. Kolojejchick, and C. Dunmire. Visage: a user interface environment for exploring information. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'96)*, pages 3–12, Los Alamitos, CA, USA, 1996. IEEE Computer Society.

[RSBO09] K. Ruhland, M. Sedlmair, S. Bioletti, and C. O'Sullivan. LibViz: Data Visualisation of the Old Library. *International Journal of Architectural Computing*, 7(1):177–192, 2009.

[RTM+03] T.M. Rhyne, M. Tory, T. Munzner, M. Ward, C. Johnson, and D.H. Laidlaw. Information and Scientific Visualization: Separate but Equal or Happy Together at Last. In *Proceedings of the IEEE Conference on Visualization (VIS'03)*, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[SBH+09] M. Sedlmair, C. Bernhold, D. Herrscher, S. Boring, and A. Butz. MostVis: An Interactive Visualization Supporting Automotive Engineers in MOST Catalog Exploration. In *Proceeding of the International Conference on Information Visualization (IV'09)*, pages 173–182, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[SC90] A.L. Strauss and J.M. Corbin. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage, Thousand Oaks, CA, USA, 1990.

[SCGM00] J. Stasko, R. Catrambone, M. Guzdial, and K. McDonald. An evaluation of space-filling information visualizations for depicting hierarchical structures. *International Journal of Human-Computers Studies*, 53(5):663–694, 2000.

[SCP95] R. Stoakley, M.J. Conway, and R. Pausch. Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'95)*, pages 265–272, New York, NY, USA, 1995. ACM Press.

[Sed07] M. Sedlmair. Methoden der Informationsvisualisierung für Kommunikationsprozesse im Fahrzeug. Master's thesis, Ludwig-Maximilians-Universität München, Germany, 2007.

[Sed08] M. Sedlmair. MSCar: Enhancing Message Sequence Charts with Interactivity for Analysing (Automotive) Communication Sequences. In *Proceedings of International Workshop on Layout of (Software) Engineering Diagrams (LED'08)*. Electronic Communications of the EASST Vol. 13, 2008. Article 7.

[Shn94] B. Shneiderman. Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11(6):70–77, 1994.

[Shn96] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of Symposium on Visual Languages (VL'96)*, pages 336–343, Los Alamitos, CA, USA, 1996. IEEE Computer Society.

[SHS+08] M. Sedlmair, W. Hintermaier, K. Stocker, T. Büring, and A. Butz. A dual-view visualization of in-car communication processes. In *Proceedings of International*

Conference on Information Visualization (IV'08), pages 157–162, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[SIBB10] M. Sedlmair, P. Isenberg, D. Baur, and A. Butz. Evaluating Information Visualization in Large Companies: Challenges, Experiences and Recommendations. In *Proceedings of CHI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV'10)*, pages 79–86, New York, NY, USA, 2010. ACM Press.

[SJOC01] H.S. Smallman, M.S. John, H.M. Oonk, and M.B. Cowen. Information availability in 2D and 3D displays. *IEEE Computer Graphics and Applications*, 21(5):51–57, 2001.

[SKHB09] M. Sedlmair, B. Kunze, W. Hintermaier, and A. Butz. User-centered Development of a Visual Exploration System for In-Car Communication. In *Proceedings of Symposium on Smart Graphics (SG'09)*, pages 105–116, Berlin, Germany, 2009. Springer-Verlag.

[SML$^+$09] M. Smuc, E. Mayr, T. Lammarsch, W. Aigner, S. Miksch, and J. Gärtner. To Score or Not to Score? Tripling Insights for Participatory Design. *IEEE Computer Graphics and Applications*, 29(3):29–38, 2009.

[SND05] P. Saraiya, C. North, and K. Duca. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):443–456, 2005.

[SNLD06] P. Saraiya, C. North, V. Lam, and K.A. Duca. An insight-based longitudinal study of visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1511–1522, 2006.

[Sof] SoftSolutions GmbH. Tracerunner. http://www.tracerunner.de/, 03/10.

[SP06] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proceedings of AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV'06)*, pages 1–7, New York, NY, USA, 2006. ACM Press.

[Spe07] R. Spence. *Information Visualization: Design for Interaction (2nd Edition)*. Prentice-Hall, Upper Saddle River, NJ, USA, 2007.

[SRBE99] M. Schulz, F. Reck, W. Bartelheimer, and T. Ertl. Interactive visualization of fluid dynamics simulations in locally refined cartesian grids. In *Proceedings of the IEEE Conference on Visualization (VIS'99)*, pages 413–416, Los Alamitos, CA, USA, 1999. IEEE Computer Society.

[SRE98] M. Schulz, T. Reuding, and T. Ertl. Analyzing engineering simulations in a virtual environment. *IEEE Computer Graphics and Applications*, pages 46–52, 1998.

[SRH$^+$09] M. Sedlmair, K. Ruhland, F. Hennecke, A. Butz, S. Bioletti, and C. O'Sullivan. Towards the Big Picture: Enriching 3D Models with Information Visualisation and Vice Versa. In *Proceedings of Symposium on Smart Graphics (SG'09)*, pages 27–39, Berlin, Germany, 2009. Springer-Verlag.

[SS06] H.J. Schulz and H. Schumann. Visualizing Graphs–A Generalized View. In *Proceedings of the International Conference on Information Visualization (IV'06)*, pages 166–173, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[Sta08] Statistisches Bundesamt Deutschland. STATMagazin, Unternehmen 2008/08, 2008. http://www.destatis.de/, last visited: 04/10.

[Ste07] J.A. Stevens. Visualization of Complex Automotive Data. *IEEE Computer Graphics and Applications*, 27(6):80–86, 2007.

[SWS07] C. Seeling, G. Watson, and K. Sun. GPU-Based Interactive, Stereoscopic Visualization of Automotive Crash Simulations. *IEEE Computer Graphics and Applications*, 27(6):6–11, 2007.

[SZ06] J. Schäuffele and T. Zurawka. *Automotive Software Engineering*. Vieweg Verlag, Braunschweig, Germany, 2006.

[SZG$^+$96] D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman. Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 3(2):188, 1996.

[TC05] J.J. Thomas and K.A. Cook. Illuminating the path: The research and development agenda for visual analytics. *National Visualization and Analytics Center*, 2005.

[TDS08] T. Tekušová, G. Darmstadt, and T. Schreck. Visualizing Time-Dependent Data in Multivariate Hierarchic Plots-Design and Evaluation of an Economic Application. In *Proceedings of International Conference on Information Visualization (IV'08)*, pages 143–150, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[The] The MathWorks. Matlab simulink. http://www.mathworks.com/, last visited: 05/10.

[TM04] Melanie Tory and Torsten Möller. Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):72–84, 2004.

[TM05] M. Tory and T. Möller. Evaluating visualizations: do expert reviews work? *IEEE Computer Graphics and Applications*, 25(5):8–11, 2005.

[Tre85] A. Treisman. Preattentive processing in vision. *Computer vision, graphics, and image processing*, 31(2):156–177, 1985.

[TSF08] M. Tory and S. Staub-French. Qualitative analysis of visualization: a building design field study. In *Proceedings of CHI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV'08)*, New York, NY, USA, 2008. ACM Press.

[Tuf83] E.R. Tufte. *The visual display of quantitative information.* Graphics Press, Cheshire, CT, USA, 1983.

[UL07] M. Utting and B. Legeard. *Practical model-based testing: a tools approach.* Morgan Kaufman, San Francisco, CA, USA, 2007.

[VCWP96] J. Viega, M.J. Conway, G. Williams, and R. Pausch. 3D magic lenses. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'96)*, pages 51–58, New York, NY, USA, 1996. ACM Press.

[Vec] Vector Informatik GmbH. Canalyzer. http://www.vector.com/, last visited: 05/10.

[VW06] J.J. Van Wijk. Bridging the gaps. *IEEE Computer Graphics and Applications*, 26(6):6–9, 2006.

[VWD04] F.B. Viégas, M. Wattenberg, and K. Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI'04)*, pages 575–582, New York, NY, USA, 2004. ACM Press.

[VWVS99] J.J. Van Wijk and E.R. Van Selow. Cluster and Calendar Based Visualization of Time Series Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'99)*, pages 166–173, Los Alamitos, CA, USA, 1999. IEEE Computer Society.

[WAPW06] C. Ware, R. Arsenault, M. Plumlee, and D. Wiley. Visualizing the underwater behavior of humpback whales. *IEEE Computer Graphics and Applications*, 26(4):14–18, 2006.

[War04] C. Ware. *Information visualization: perception for design.* Morgan Kaufmann, San Francisco, CA, USA, 2004.

[War08] C. Ware. *Visual thinking for design.* Morgan Kaufmann, San Francisco, CA, USA, 2008.

[Wat02] M. Wattenberg. Arc diagrams: Visualizing structure in strings. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, pages 110–116, Los Alamitos, CA, USA, 2002. IEEE Computer Society.

[Wea04] C. Weaver. Building highly-coordinated visualizations in Improvise. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'04)*, pages 159–166, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[WFR$^+$06] C. Weaver, D. Fyfe, A. Robinson, D. Holdsworth, D. Peuquet, and A.M. MacEachren. Visual analysis of historic hotel visitation patterns. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST'06)*, pages 35–42, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[WI90] J.Q. Walker II. A Node-Positioning Algorithm for General Trees. *Software: Practice and Experience*, 20(7):685–705, 1990.

[WKVD$^+$08] I. Wassink, O. Kulyk, E. Van Dijk, G. Van der Veer, and P.E. Van der Vet. Applying a User-Centered Approach to Interactive Visualisation Design. In *E. Zudilova-Seinstra, T. Adriaansen und R. van Liere (eds): Trends in Interactive Visualisation*. Springer-Verlag, London, England, 2008.

[WMM06] C.V. Wright, F. Monrose, and G.M. Masson. Using visual motifs to classify encrypted traffic. In *Proceedings of the International Workshop on Visualization for Cyber Security (VizSec'06)*, pages 41–50, 2006.

[WPCM02] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, 2002. Palgrave Macmillan.

[WR06] H. Wallentowitz and K. Reif. *Handbuch Kraftfahrzeugelektronik: Grundlagen, Komponenten, Systeme, Anwendungen*. Springer-Verlag, Berlin, Germany, 2006.

[WS92] C. Williamson and B. Shneiderman. The Dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 338–346, New York, NY, USA, 1992. ACM Press.

[Wür10] M. Würtenberger. Zukunftsweisende it fahrzeug architekturen durch integration leistungsfähiger vernetzungstechnologien, 2010. http://bicc-net.de/events/innovation-forum-embedded-systems/referenten/michael-wurtenberger/, last visited: 05/10.

[WW03] M. Weber and J. Weisbrod. Requirements engineering in automotive development: experiences and challenges. *IEEE Software*, 20(1):16–24, 2003.

[YWR02] J. Yang, M.O. Ward, and E.A. Rundensteiner. InterRing: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02))*, pages 77–84, Los Alamitos, CA, USA, 2002. IEEE Computer Society.

[ZKB02] J. Ziegler, C. Kunz, and V. Botsch. Matrix browser: visualizing and exploring large networked information spaces. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI'02)*, pages 602–603, New York, NY, USA, 2002. ACM Press.

[ZMC05] S. Zhao, M.J. McGuffin, and M.H. Chignell. Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'05)*, pages 57–64, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[ZS06] W. Zimmermann and R. Schmidgall. *Bussysteme in der Fahrzeugtechnik*. Vieweg Verlag, Braunschweig, Germany, 2006.

List of Figures

List of Tables