Requirements and Applications for Visual Analysis of In-Car Communication Traces

Removed for Blind Review

ABSTRACT

We describe our experience developing and introducing visual analytics tools from within a large industrial work context. We focus on visual analytics for in-car communication networks in the domain of automotive engineering. From a three-year field study we provide insight into the types of data and challenges inherent in debugging the sensor networks of cars and discuss a set of 11 recommendations for visual analytics tool development. We focus not only on domain-specific recommendations but also integrate recommendations based on the large industrial context in which our domain experts worked. The recommendations are applied and discussed in context of two novel visual analytics applications including evidence that their usage led to tools which now effectively support engineers in understanding and debugging masses of data and in turn improve the safety of cars and passengers.

Author Keywords

Visual Analytics, In-Car Communication Analysis

ACM Classification Keywords

H.4.0 Information Systems Applications: [General]

INTRODUCTION

The field of visual analytics (VA) is continuously growing with research efforts expanding into many different domains. Visual analytics tools address the challenge of analyzing overwhelming amounts of data by combining methods from various disciplines, including information visualization (InfoVis), HCI and data analysis techniques from statistics, data mining, and others [33]. One very crucial aspect for the future of the field is to bridge the gap from research to application as the success of the field may eventually be measured by how many people will actively be using solutions in their everyday work [5]. In this paper, we present the results of a three-year research endeavor with the goal to develop, connect, and integrate visual analytics tools directly within an industrial work context. We worked closely with domain experts in a large automotive company to support data analysis challenges encountered during the debugging of sensor networks in modern automobiles. We provided analysts with several solutions for problems encountered in their everyday real-world work context. In order to ensure that our tools would meet the requirements of the industrial work context, we engaged in a long-term problem characterization phase [18], designed and tested various tool designs, and finally adopted and integrated some of them into the engineers' daily working practices. This process required careful consideration as our industry contacts had not been previously exposed to visual analytics research and solutions and mostly relied on purely textual representations which could not provide insight into more complex data relationships. In this paper, we describe the findings of our explorative field study, design requirements we derived from it, and introduce two of our novel visual analytics tools, Cardiogram and AutobahnVis, as exemplary approaches being adopted and integrated into tool sets currently in use by analysts.

Our work in this domain makes three primary contributions: We first discuss the results of our explorative field analysis. We closely cooperated with domain experts and used several analysis methods to gain a deep understanding of the problem domain. We summarize the results of this exploration as a set of 11 design recommendations and want to provide them as guidance for other visual analytics tool designers in similar contexts. This type of problem characterization in its own right has increasingly been called for in the areas of visual analytics and information visualization [4, 13, 18] in order to help researchers gain a better understanding of everyday data analysis practices across domains and to guide the design of visual analytics tools targeted towards real-life analysis needs. Our collection of characteristics is the first for this domain with a specific focus on visual analytics challenges. Second, we introduce two novel visual analytics tools, Cardiogram and AutobahnVis, that we built based on these design recommendations and discuss how we integrated them with real end users. Finally, we take a step back and review our experience from the domain-specific aspects of the project and discuss the aspects that led to our successful integration of visual analytics tools in a large industrial context. While the description of the characteristics of our application domain are meant to guide practitioners and developers in the same or similar domains, the description of our process may prove to be useful for visual analytics practitioners in many areas who wish to form closer connections to industrial partners. To our knowledge, our reflection on challenges encountered in developing visual analytics solutions in an industrial work context is the first in this domain.

Submitted for review to CHI 2011.

INDUSTRIAL BACKGROUND

Our work is situated in the domain of automotive engineering where we worked with engineers involved in the development of sensory networks in cars. The concentration of electronics and software in automobiles has increased enormously over the last years and has brought a variety of novel challenges such as intricate error diagnosis and recovery to automotive engineers [3, 7, 8, 23]. High-performance cars, for example, are equipped with several hundred sensors which can deliver information at speeds of more than 1,000 readings per second over a large interconnected sensory network. For example, in order to detect whether an airbag should be triggered in a car, accelerometers send information to a microprocessor at 10ms intervals and the evaluation of this data determines whether and how to inflate an airbag [1]. In order to transport all relevant information, up to 15,000 messages per second are distributed over a typical sensory network. To verify the correctness of these sensor networks, engineers log the messages via specific hardware that is installed in test cars. Logging one hour results in approx. 2 GB of data and roughly 50 million recorded messages. Thus, a lot of time and experience is needed to understand in-car communication processes and their complex correlations in order to detect sources of errors. These data related challenges make this work domain a prime candidate for dedicated visual analytics tools and—not least—providing effective tools for 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.

RELATED WORK

In the following, we review related work of previous pre-design field studies in information visualization and visual analytics, and also on visualization in our target domain, automotive engineering

End User Integration in Visual Analytics

Recently, information visualization researchers have explicitly called for a closer integration of end-users in InfoVis/VA tool development and for the dissemination of qualitative reports on data analysis practices in real-life work contexts to ground both design and/or subsequent evaluation [4, 13, 18]. Indeed, more and more InfoVis/VA projects focus on solutions for real-world, data-intensive application domains and integrate users into their development processes, such as a recent visual analytics tool for patent specialists [15] and a financial data analysis tool [32]. In our work, we used ethnographic field analysis of current practices to inform the design of our systems. While the need to base system design on an in-depth understanding of an domain has been understood in HCI (e. g., [Does sb know a good paper for this?]) for a long time, few examples of such studies exist in the information analysis and visualization area. Tory et al. [34], for instance, conducted a qualitative analysis in the building design field, Isenberg et al. [12] studied the work of traditional collaborative data analysis, as well as McLachlan et al. [17] and Henry and Fekete [9] who based their visualization system designs on a qualitative field analysis with system management professional and social science researchers respectively. None of these studies so far, however, have been conducted in a large industrial setting or in our application domain.

Visualization in the Automotive Domain

Visualization in the automotive domain is most commonly used in the context of computer-aided-design, virtual reality, and scientific visualization [31]. Within scientific visualization, many techniques have focused on the analysis of physically based (often simulated) data, such as the flow of particles for car body development [24]. Such techniques have also been integrated with information visualizations such as scatterplots and histograms for the analysis of, for example, a Diesel exhaust system [6]. While some of this work (e. g.,[16]) shows increasing interest in integrating information visualization as in our work, considerably less work has been dedicated to the support of *electronic* engineering for car development and testing. In our previous work, we presented several point-solutions for visualizing in car communication networks, including an approach for visualizing large catalogs for vehicles' electronic specifications [25], for gaining insight into dependency chains from message traces [26], and to help engineers in better understanding correlations between mechanical and electronic information via a 3d model visualization of a car [29]. We also explored solutions for time-based trace visualization [28] which formed the basis for AutobahnVis presented in this paper. These previous systems presented prototypical solutions for specific analysis challenges and helped us to gain more general insights into the problems faced by automotive engineers. In this paper, we build on this work and present extended insights, fully integrated solutions and in-depth evaluations of our tools.

METHODOLOGY

Our methodology was chosen based on two main objectives: (1) Learning about the field including current practices, problems and challenges, and (2) learning about how to design and integrate successful VA tools in our target domain. For this purpose, we primarily focused on working closely with experts in our target domain. Both the complexity of data and analysis tasks as well as the time-constraints of the industrial work context influenced the study methods we chose and how we adopted them to these specific requirements. In spirit, our approach is similar to MILCs (Multi-dimensional In-depth Long-term Case Studies) as advocated previously for information visualization for addressing the specific needs of this domain with its complex, exploratory and ill-defined tasks [30]. Similar to the traditional MILC approach, we used a *multi-dimensional* set of study methods, and carried out *in-depth* and *long-term* investigations over a period of three years. We carried out our investigation on the data analysis practices of a large number of 50 expert analysts using both, their own, currently used tools and our novel tools, and did not focus on studying any tool in particular. We, thus, conducted an "ambitious" MILC as proposed for future work in [30]. Approx. 90% of the analysts we worked with were male and had been working in the problem domain for 2–16 years.

In the following, we describe the various methods we used for studying current practices for introducing VA in our domain. We also briefly reflect our objectives and experience using these methods for InfoVis/VA in a large company setting in order to provide other researchers with guidance for such undertakings. Our study is among one of the first in-depth endeavors of studying the complex pattern of data analysis using

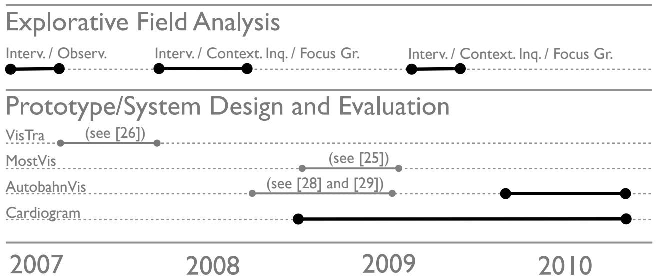

Figure 1: Overview over our development process. Black lines indicate work reported on in this paper. Grayed out lines indicate phases reported on earlier.

long-term MILCs. The time investment of three years was substantial but essential to our success later. Figure 1 provides a temporal overview over our studies as well as the different participatory tool design and evaluation phases where we closely cooperated with our target users.

Methods for Explorative Field Analysis

Interviews: Especially early on, we relied on semi-structured interviews to gain insights into technical aspects beyond what we could derive from technical documentation. Interviews provided us with a rough understanding of work practices, and showed engineers' estimations on challenges and problems as well as their ideas and approaches to overcome them. Overall, we conducted 30 1h-long interviews with both, analysts and tool designers. From our notes and recordings we derived main categories of tasks and analysis challenges.

User Observation and Contextual Inquiries: In order to collect more real-life data of our experts' work practices, we started observational studies with a fly-on-the-wall technique in which we followed but did not interfere with the daily practices of the engineers. However, due to task complexity and high expertise in our domain it was not possible for a non-domain-expert to derive meaningful results through observation alone. Therefore, we chose to use a variant of contextual inquiries [11] which we conducted with 14 analysts in up to five 1h-long sessions per participant. Due to IPR restrictions, we solely used note-taking for data logging purposes. We used 2–3 note takers to counterbalance the lack of recording equipment. We used these results to broaden our understanding of the diversity of working practices in our domain and to verify or falsify our prior findings.

Focus Groups: To evaluate, improve and focus our findings from interviews and user observations we conducted approx. 20 focus groups of 3–10 participants. We chose focus groups because we frequently encountered diverse and even opposed practices and statements during individual interviews. Our major goal in bringing experts together was, therefore, to form a common understanding—not just between the end users and us but also among the end users themselves. The groups consisted of varying groups of analysts, tool designers and also invited decision makers, an important group to address in large company settings as they usually decide whether a novel solution will be integrated and funded or not. As our goal was to reach adoption of our solutions, we decided to integrate decision makers early in the process and this proved to be essential to our success.

Participatory Design and Tool Evaluation

We augmented the methods used above with a tight collaborating with end users during the design and evaluation of our own VA tools. This helped us to gain additional insights into applying VA in our domain, specific challenges for VA tools and their integration into the end users' work environment. Our long-term process was, therefore, an iterative mix of field studies, tool design and tool evaluation—each of them informing another by providing us but also our end users with new insights. For designing and evaluating our tools we used the following methods:

Design Workshops and Personal Feedback: We designed all our tools in close cooperation with end users applying a participatory design approach [14]. During several exploratory design workshops we introduced engineers to visual analytics techniques, discussed ideas and fine-tuned possible solutions (paper mockups, system designs, features, etc.), and finally developed a basic concept. Subsequently, we started developing interactive tools and provided a group of carefully chosen test engineers with frequent iterative releases and encouraged them to provide us with feedback in personal meetings. We also conducted heuristic usability evaluations, basically with outside testers in order to save valuable time of our domain experts. This approach has been previously lead to success in the data analysis domain [17].

Think Aloud Protocols, Lab Studies and Field Studies: To evaluate the domain value of our tools we used both lab as well as field studies. We studied our early systems using think-aloud protocols and lab studies, investigating domain experts using our tools in artificial lab settings. While we received valuable feedback about the potentials and improvements of our tools, we did not gain insight into their usage under real-world conditions (cf. [20]). A major hindrance to gaining real-world insight on tool use was the close integration of our tools into current systems already in use at the company [27]. We report on how this hindrance was overcome in later sections.

Informal Collaboration

Besides our formal studies, we found it invaluable for our successful integration to engage in frequent informal conversations with engineers. These conversations were, for instance, meetings at the company's cafeteria, a joint lunch, or casual discussions at the workplace. We improved our understanding of the various facets of our target domain, iteratively refined our requirements for visual analysis tools, and established successful collaborations. A drawback of informal conversations, however, is the restricted opportunity to log data for scientific rigor. In order to allow for some direct data collection we always carried notepads in case spontaneous conversations would occur. Over the three year period we collected information from roughly 80 of these spontaneous encounters, with approx. 50 different engineers.

RESULTS OF EXPLORATIVE FIELD ANALYSIS

In the following, we summarize the understanding we have gained of our domain and its data analysis challenges by using this variety of study methods. In order to provide other researchers with a profile of our end users, we organized our results by data, task, current tools, and a more detailed description of practices and challenges—all of them relevant parameters for designing visual analytics tools in this domain. Our summarization is the first in the area of visual analytics for automotive engineering. It is meant both as such a description to enlighten our general understanding of real-life data analysis but also as a guide to practitioners in the field.

Data
Automotive engineers in this domain work with large test files called *traces*—temporally ordered lists of all messages logged during a test drive. Traces come in different formats (depending on the recording hardware) and are not necessarily complete (e. g., failure of recording hardware). Engineers use *journals* as specific, pre-filtered formats to reduce the trace data to a few specific message types such as error frames or fault memory entries, and manually add information such as markers, triggers or predefined events.

Main Task
If an error has been detected during a test drive, it is the task of the analysis engineer to locate the error source and to initiate further steps to solve the problem.

Current Tools
Several special-purpose analysis tools were in use by our engineers. The most important ones are an in-house tool called Carmen as well as Canalyzer [35]. Both of them were considered 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 combination of different digital modules that allow an individual configuration of a tailored measuring setup. 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 overviews. To analyze traces, our experts often used one or several of these tools in combination with general-purpose tools such as text editors.

Data Analysis: Practices, Problems and Challenges
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. 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.

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"* (quote from a contextual interview, translated from German). In close cooperation with engineers we identified three main origins of complexity:

Reproducibility: Reasons for low reproducibility of errors include external circumstances such as temperature, 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 systems before they are automatically detected and logged. 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 to reduce and abstract the recorded data. Without any abstraction it can become complex and laborious to analyze message 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 [...]"*.

DESIGN RECOMMENDATIONS
Based on the in-depth understanding of our industrial work context, we provide several design recommendations for visual analytics. These recommendations helped us in designing our own tools and are meant to guide other tool designers in this or similar domains. Some of these requirements are known from general information visualization advice (e. g., provide overviews) but we provide them for a complete description of necessary enhancements to current approaches.

New Perspectives on the Data
In our specific domain, complex errors posed the greatest analysis challenges to our engineers. The detection of complex errors requires dedicated data representations to show correlations between error sources.

OVERVIEW: Visual Overview Techniques
Most current overviews are restricted to the representation of journal data. To provide overviews of message 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.

NEWPERS: Perspectives Beyond Raw Data and Signal Plots
In our study, most data representation were based on lists and simple signal value plots. Beyond the capabilities of these tools, engineers need to analyze timing aspects and message propagation, detect outliers and see correlation between messages and between mechanical behavior and electronic data.

TIMLOG: Equal Representation of Time and Logic
This recommendation is an important sub-component of NEWPERS. 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 technique could not support this requirement.

MODULAR: Multiple, Modular, and Coordinated Solutions
Engineers require multiple different perspectives on the data to detect complex errors. Which perspectives were most relevant 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 [19]), but also the opportunity to work without coordination (e. g., for comparing behavior at different time stamps).

FASTACC: 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.

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. The following requirements reflect this challenge.

ABSTRACT: 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 OVERVIEW and NEWPERS).

AUTOMAT: 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 a lot of 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.

AVOREP: 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 to avoid redundant steps.

Engineer-centered Solutions
From previous research we know 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 [27]. This was also important in our domain.

EMBED: 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 may not be possible within realistic time and budget requirements. 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 and extend engineers' current work processes without any extra costs. This may have additional benefits in terms of adoption.

FAM: Take Familiarity into Account
Message sequence charts, state machine diagrams, as well as network representations are frequently used for system specification, communication between different groups of engineers and supported by frequently used tools such as Matlab Simulink for related tasks. To support communication with and between engineers it is advantageous to reuse these common representation techniques as they are well-known mental models and metaphors from the engineering domain. Similarly, supporting known interactions and work flows can help engineers to adopt a new tool.

COLLAB: Support Collaboration
Our target group is located in a large company setting where thousands of employees work on highly specified tasks. To form a greater understanding based on individual expertise and to master comprehensive challenges collaboration is indispensable. This is particularly true for complex error analysis. Tools in this application area should actively support collaboration around the data.

VISUAL ANALYSIS MODULES
Based on our field analysis and design recommendations we implemented several new interactive visualization modules. In this paper, we focus on two of these modules, Cardiogram and AutobahnVis, briefly discuss their design and explain the main choices we made in relation to our design recommendations. Both systems are additionally described in the accompanying video explaining their features in more depth.

We spent considerable effort to closely integrate our modules with a comprehensive in-house analysis software environment, Carmen, which supports valuable back-end features such as data storage, interpretation, and filtering (EMBED).

This tool is widely used by our target users and provides a well-known, accepted platform (FAM). Integrating our tools with Carmen helped (a) our engineers to avoid unnecessary data conversions and repetitive tasks (AVOREP) but also (b) us to study them analyzing data with our tools under realistic conditions (cf. [30, 27]).

The main analytics objective of the two modules was to provide a better understanding of temporal, logical, and behavioral aspects within the data (NEWPERS & TIMLOG). These aspects are crucial to detect the causes of or influences on complex errors and novel tools are urgently needed. Both applications are centered around a zoomable timeline (inspired by [21]), multiple coordinated views (MODULAR, inspired by [19]), and use consistent representation and interaction techniques. Our solutions were also inspired by work on previous trace visualization design studies (most importantly [22] and [10]). Similar to these solutions we chose to focus on the temporal aspect of trace data. Besides, the state machine analysis technique we use in Cardiogram is based on model-based automotive software testing [2].

Cardiogram Module

Our main analytics objective of Cardiogram was to reduce engineers' information load by automated data reduction and analysis (AUTOMAT) and to provide a better understanding of temporal, logical, and behavioral aspects within the data (NEWPERS & TIMLOG). For this purpose we set up two major software components: (1) A Data Preprocessing and Storage Component and (2) a Visualization Component.

Data Preprocessing and Storage: Based on our main goals, this unit contains three components:

State Machine Specification: We provided engineers with the ability to specify vehicle behavior into state machines. To do so, they define a set of logical vehicle states such as *"window front left open"*, transitions between states such as *"open window"*, and what events in a trace result in which transitions, e. g., *"message xy opens window"*. To test predefined conditions, each state can be additionally annotated as *error*, *warning*, *okay*.

Data Storage: Together with a description, all these 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 (COLLAB) and reduces redundant work (AVOREP).

Automatic Trace Preprocessing: The State Machine Evaluation Engine is the heart of Cardiogram and is directly integrated as a module in Carmen (EMBED). It supports loading specific state machines for analysis and imports traces by utilizing other modules in Carmen, such as a Trace Replay Module (AVOREP). Using all loaded state machines, the State Machine Evaluation Engine reduces the traces to a temporal ordered list of transitions in these state machines. For each analyzed state machine, an additional global tag, called *aggregated state machine result*, indicates the occurrences of *error* and *warning* states which can be used for later analysis. The engine thus reduces and preprocesses the data to be analyzed by an engineer (ABSTRACT) and makes novel overview techniques possible (OVERVIEW).

Visualization: The main purpose of Cardiogram's visualization (cf. Figure 2) is to support the exploration of errors, warnings, or other hints which may require further inspection, i. e., that have not been automatically detected. In such situations, 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, we allow engineers to gain a novel perspective on complex dependencies of in-car networks (NEWPERS) and correlate logical with timing aspects (TIMLOG).

Our basic decisions in designing the visualization component can be summarized as follows:

Prioritization: A *State Machine View* lists all state machines tested on a specific trace (OVERVIEW). As requested by our target group, this list is sorted according to priority based on aggregated state machine results (first those with at least one *error* state, then the ones with at least one *warning*, etc.).

Familiarity: Due to their familiarity (FAM), we used traffic light icons next to each state machine entry in the State Machine View that encoded the aggregated state machine outcomes *error*, *warning*, *okay* using the colors red, yellow, green, and no color coding for other outcomes (ABSTRACT). By selecting a state machine from the State Machine View a familiar state/time plot (FAM) is shown in the *Visualization View* showing all transitions of this state machine according to its transition table. 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 (FASTACC).

Time Reduction: 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 (ABSTRACT). 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 (FAM) in the overview bar. This supports fast readability of transitions that could be relevant for bugfixing.

Collaboration: Interactive annotation of the data is also supported. Each analyst can freely attach colored notes directly into a state/time plot. Symbols for notes 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 (COLLAB & AVOREP).

Convenience Features: According to engineers' requests during our user-centered design process, we integrated a variety of other interactive features, including: keyboard short-cuts for all features (FAM), unrestricted vertical scaling of state machine plots (OVERVIEW), minimization and closing of state machine plots, drag& drop positioning

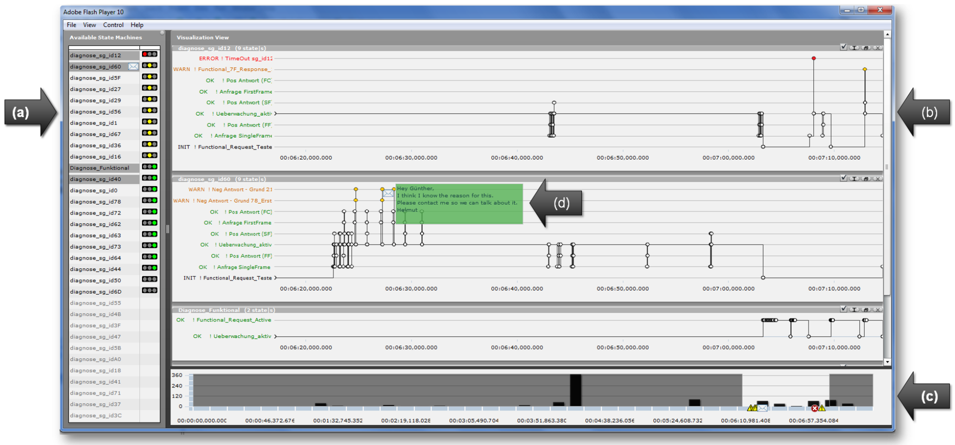

Figure 2: Screenshot of the Cardiogram Visualization. (a) State Machine View showing all tested state machines ordered by relevance for bugfixing; (b) Visualization View with several detailed state/time plots showing transitions via vertical and horizontal lines and additional glyphs at target states; (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.

of the state machine plots to allow side-by-side comparison (both NewPers), dynamic adding or subtractions of state machines to or from the overview bar (Overview), and the free configuration of nearly all system features and settings.

Evaluation: In order to ensure Cardiogram's successful adoption and integration we engaged in a two-phase evaluation separately for (1) the semi-automatic data preparation and (2) the visualization. To test our novel automation and abstraction approach using state machine data preparation, we first implemented a graphical state machine editor, the state machine database and the state machine engine and integrated all components as a single module in Carmen. Together with a textual representation of statistical results (# of errors/warnings) and transition lists for detailed inspection on demand (Fam), this approach was validated in a 12-month field study with 15 domain experts. During these 12 months, experts used the tool in their data analysis procedures and created state machines on their own. The length of the individual real-world usage sessions varied from several minutes up to three hours. During bi-weekly meetings we discussed their experiences with our tools and elicited feedback on their benefits and areas for improvement. The Cardiogram visualization was qualitatively evaluated with six domain experts each during a one-hour think-aloud session in which they used the visualization on their own data and/or on test data sets we provided. Additionally we received feedback from two test users who used Cardiogram for eight weeks during their daily work. The results of the study uncovered several main categories of benefits for our new approach which aided in adoption and acceptance of the tool:

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. 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. 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 traces, and to test or verify hypotheses on a broad testing basis.

Show Data Correlations and Overviews: All of our participants stated that the Cardiogram visualization was enormously helpful to understand and explore correlations between dependent state machines in cases an error or a warning appeared in the automatic data preparation. For example, we saw the Cardiogram visualization being used to compare timings, to verify correctness of temporal order, and to correlate the transitions of the state machines for several parallel procedures involved in shutting down a car. This in-depth analysis was not possible previously.

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 Abstract, Automat, Collab, Embed AvoRep, Overview, NewPers, and TimLog. At the time of writing, the tool has spread within the company and is now widely used by more than 30 engineers on a daily and weekly basis. We recently transferred our software to the Carmen tool developers who are now extending our solutions and directly embed the tool into Carmen's core components.

AutobahnVis Module

While abstraction techniques such as Cardiogram are valuable tools for data reduction and overview purposes, in many trace analysis processes it is still necessary to investigate the raw message information (FastAcc). For this purpose, we designed AutobahnVis in order to extend the current state-of-the-art technology for raw data analysis used by our engineers and to address shortcomings of the popular text-based representations. AutobahnVis focuses on timing aspects within traces' raw data and is based on our experience with a former prototype [28] which was not used by engineers in their daily analysis activities because of missing integration into current tools (EMBED) and insufficient compatibility and scalability to real-world traces. Like Cardiogram, we now designed AutobahnVis to be directly integrated as a module into Carmen (EMBED) and to closely interact with several of its other modules, most importantly the Replay Module inputting traces into the AutobahnVis module and existing filter modules to preprocess traces data in the familiar way (MODULAR & FAM). In doing so, engineers can use our AutobahnVis module directly for the data they currently work on without any extra costs (AVOREP).

Visualization: The basic idea behind AutobahnVis is to provide the user with a zoomable network visualization which represents raw messages (as diamonds) according to two main dimensions: time on the x-axis and the transporting bus system or sending ECU on the y-axis (TIMLOG). In addition to this basic concept and in line with our design recommendations, we added the following components and features:

- **Embedding with Traditional Views:** In our problem characterization phase, we learned that productive work with AutobahnVis would require a close connection to textual trace representations, message filters, and signal plots. For fast access (FASTACC) to this data, we designed these views and functions in resemblance to other tools used by engineers, directly integrated them with the AutobahnVis module, and coordinated all views based on common information visualization techniques ([19]). Figure 3 shows AutobahnVis' different views and their functions.
- **Convenience Features:** Similar to Cardiogram we integrated several additional usability features after close collaboration 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 to see systematic comparison of message timings (NEWPERS).

Evaluation: To evaluate AutobahnVis, we collected informal feedback during our user-centered design approach and frequently engaged in discussions with domain experts. Five lead automotive 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 think-aloud observational session with our tool. Based on these evaluations, we improved our tool and discuss its main benefits:

- **Novel Insights for Complex Error Detection:** AutobahnVis allowed for several instances of complex error detection during the analysts' daily work. Three of our lead analysts used AutobahnVis, for example, to track *messages dispersion* 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. 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. One lead analyst also 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. All these examples resulted in the detection of errors that would have been complex and extremely time-consuming to detect with current techniques.
- **Simplifying Current Practices:** Along with novel insights, our participants particularly pointed out the simplification of some analysis tasks compared to previous tools. Examples include the possibility for side-by-side comparison especially the close link between signal value changes in the SignalView and messages in the MessageView, and the usage of AutobahnVis to validate request/response message pairs and cyclic messaging for correctness and timing conditions. We, for instance, observed one analyst using the Multi-Color search and filters and reporting to be much faster than with traditional tools. Engineers appreciated the reduced work, and attention shifts required to perform the analysis.

Following our initial design recommendation, our studies showed that our design decisions did indeed support engineers with several of their analysis needs using AutobahnVis (most importantly, OVERVIEW, NEWPERS, MODULAR, FASTACC, EMBED and FAM). When we asked our participants about the adoption of AutobahnVis several months after our studies, all of them were still using AutobahnVis on an occasional basis in a similar frequency as they used it during our studies (once or twice a week). The basic reason for using it not more often is a current limitation of our module that restricts to showing messages solely separated by bus systems but not by ECUs. Adding this functionality is difficult as interpretation of this information is currently not supported by Carmen's connection technology for external modules (such as AutobahnVis). To make this possible, we recently—similar to Cardiogram—transferred the tool to the Carmen tool developers who are now working on overcoming this restriction and on making AutobhanVis available as a Carmen core component.

DISCUSSION

The evaluations of both Cardiogram and AutobahnVis revealed valuable information on the performance of both modules in real-world analysis situations. Overall, the recommendations from our exploratory field analysis together with a user-centered design process helped us to design tools that found their way into everyday routines of our analysis experts. We learned that the core ingredients to a successful deployment of our visual analytics techniques in this industrial setting were (a) the visualizations' simplicity, (b) strong user integration throughout the entire design

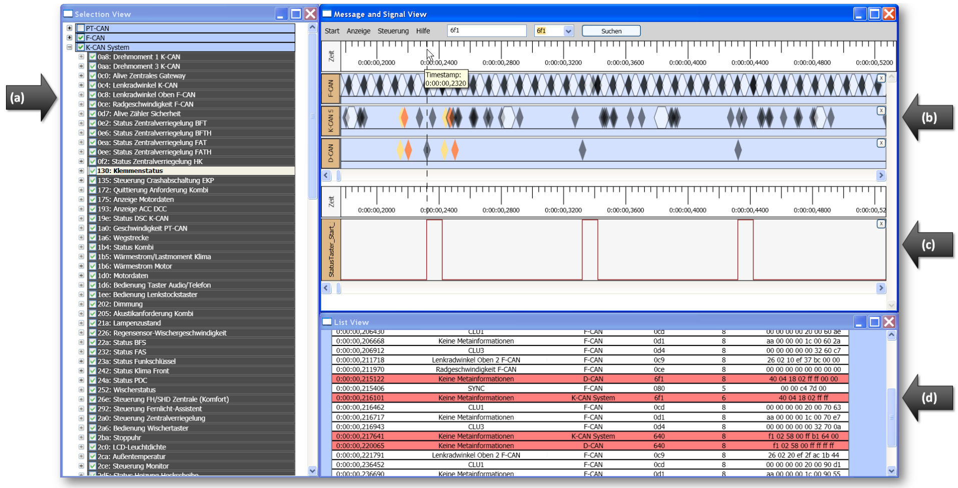

Figure 3: Screenshot of AutobahnVis. (a) Selection View: allows to select and filter messages, bus systems and signals; (b) Message View: shows messages sent over the bus systems ordered by time, a Multi-Color search shows four request/response pairs; (c) Signal View: shows signal value plots for selected signals; and (d) List View: provides a traditional text based representation of message details.

process, and (c) most importantly a tight integration into existing tools and workflows.

Simplicity: 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.

User Integration: Engineering is a complex area that requires expertise and background knowledge. As outsiders to the area we found it invaluable to counterbalance our little domain knowledge through an exploratory study of analysis practices in the domain and a user-centered design processes. Allowing end-users to closely participate in the design process helped us to implement tailored, well-directed, and valuable solutions. Besides, we especially see a large part of our successful integration in our in-depth and long-term field evaluation of current data analysis practices and needs, the usage of informal methods that helped us to form a close connection to our target users, and the early integration of stakeholders in our studies as they finally decided to integrate our tools as core components into daily practices.

Tool Integration: We also learned that in our domain 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. Especially in large companies, often a set of traditional tools already exist for a data analysis problem at hand. These tools are usually integrated into a well-defined process and re-implementing them for a research project is too expensive. However, neglecting these practices can pose additional costs for employees, for instance, due to additional data transformation efforts or tool-flipping costs. In our case, it turned out that such additional time costs were not acceptable for engineers and poorly integrated tools were disapproved of in daily work. Though integrating VA tools into daily practices was a labor-intensive process, it eventually helped us to study our tools under realistic conditions leading to additional research value.

With a focus on simple solutions, iterative design and testing, and tight integration we deviated from our earlier visualization prototypes which tended to be feature-rich, but were never properly connected to support real-world problem solving in real environments with real data. Consequently, evaluating earlier prototypes had always been restricted to expert estimation within more or less artificial conditions. Indeed, this provided valuable and helpful feedback, 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 modules after the evaluation phase was completed and also recommended it to their colleagues.

While our experience so far is solely based on working in one large company setting, our findings can serve as reference for others who are planning to closer cooperate with industrial partners in their VA research. We encourage other researchers to report their own experience in order to broaden our general understanding about such projects and to help VA"move from research to practice" [33].

CONCLUSION

In 2006, Broy [3] 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". This holds unchanged even four years later and is probably the strongest motivation behind the work presented in this paper. Analyzing in-car communication is already very difficult and will soon become entirely infeasible without the help of visualization and automation techniques. In this paper, we contribute general recommendations for visual analytics tools in this domain that we derived during a three year exploratory field evaluation. We further show how we applied the recommendations in two specific visual analytics modules to help automotive engineers analyze in-car communication traces. Both modules were integrated tightly into engineers daily work processes and evaluated with domain experts and their own data. Evaluations showed that our recommendations helped us to design highly valued tools that addressed practical data analysis needs of automotive engineers. Our work is one of the first which reports on a long-term field evaluation of both requirements in this domain as well as the work necessary to ensure continued adoption and future use.

REFERENCES

1. R. Bosch GmbH. *Kraftfahrzeugtechnisches Taschenbuch*. Vieweg&Sohn Verlagsgesellschaft, 26 edition, 2007.
2. E. Bringmann and A. Krämer. Model-based Testing of Automotive Systems. In *Proc. Conference on Software Testing, Verification, and Validation*, pages 485–493. IEEE, 2008.
3. M. Broy. Challenges in automotive software engineering. In *Proc. Software Engineering (ICSE)*, pages 33–42. ACM, 2006.
4. S. Carpendale. Evaluating information visualizations. In A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, editors, *Information Visualization: Human-Centered Issues and Perspectives*, pages 19–45. Springer LNCS, 2007.
5. K. Cook. From research to reality: Visual analytics technology transition. In *VAC Views*. PNNL, Nov. 2008.
6. 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 *Proc. of IEEE TCVG Symp. on Visualization (EuroVis)*, pages 91–96, 2004.
7. K. Grimm. Software technology in an automotive company: major challenges. In *Proc. Software Engineering (ICSE)*, pages 498–503. IEEE, 2003.
8. H. Heinecke. Automotive system design-challenges and potential. In *Proc. Design, Automation and Test (DATE)*, pages 656–657. IEEE, 2005.
9. N. Henry and J. Fekete. Matrixexplorer: a dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):677, 2006.
10. D. Holten, B. Cornelissen, and J. van Wijk. Trace Visualization Using Hierarchical Edge Bundles and Massive Sequence Views. In *Proc. of Visualizing Software for Understanding and Analysis (VISSOFT)*, pages 47–54, 2007.
11. 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.
12. P. Isenberg, A. Tang, and S. Carpendale. An exploratory study of visual information analysis. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1217–1226. ACM, 2008.
13. P. Isenberg, T. Zuk, C. Collins, and S. Carpendale. Grounded evaluation of information visualizations. In *Proc. BELIV*, pages 56–63, New York, USA, 2008. ACM.
14. F. Kensing and J. Simonsen. MUST: A method for participatory design. *Human-Computer Interaction*, 13(2):167–198, 1998.
15. S. Koch, H. Bosch, M. Giereth, and T. Ertl. Iterative Integration of Visual Insights during Patent Search and Analysis. In *Proc. VAST*, pages 203–210. IEEE, 2009.
16. 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 *Proc. CMV*, pages 3–15. IEEE, 2007.
17. P. McLachlan, T. Munzner, E. Koutsofios, and S. North. LiveRAC: interactive visual exploration of system management time-series data. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1483–1492. ACM, 2008.
18. T. Munzner. A nested process model for visualization design and validation. *Trans. on Visualization and Computer Graphics*, 15(6):921–928, 2009.
19. C. North and B. Shneiderman. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In *Proc. AVI*, pages 128–135. ACM, 2000.
20. C. Plaisant. The Challenge of Information Visualization Evaluation. In *Proc. Advanced Visual Interfaces (AVI)*, pages 109–116, New York, NY, USA, 2004. ACM Press.
21. C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. LifeLines: visualizing personal histories. In *Proc. CHI*, pages 221–227, 1996.
22. A. Pretorius and J. Van Wijk. Multiple views on system traces. In *Proc. PacificVis*, pages 95–102. IEEE, 2008.
23. A. Pretschner, M. Broy, I. Kruger, and T. Stauner. Software engineering for automotive systems: A roadmap. In *Future of Software Engineering*, pages 55–71. IEEE, 2007.
24. M. Schulz, F. Reck, W. Bartelheimer, and T. Ertl. Interactive visualization of fluid dynamics simulations in locally refined cartesian grids. In *Proc. VIS*, pages 413–416. IEEE, 1999.
25. M. Sedlmair, C. Bernhold, D. Herrscher, S. Boring, and A. Butz. Mostvis: An interactive visualization supporting automotive engineers in most catalog exploration. In *Proc. Information Visualisation (IV)*, pages 173–182. IEEE, 2009.
26. M. Sedlmair, W. Hintermaier, K. Stocker, T. Büring, and A. Butz. A dual-view visualization of in-car communication processes. In *Proc. Information Visualization (IV)*, pages 157–162. IEEE, 2008.
27. M. Sedlmair, P. Isenberg, D. Baur, and A. Butz. Evaluating Information Visualization in Large Companies: Challenges, Experiences and Recommendations. In *Proc. BELIV*. ACM, 2010.
28. M. Sedlmair, B. Kunze, W. Hintermaier, and A. Butz. User-centered Development of a Visual Exploration System for In-Car Communication. In *Proc. Symposium on Smart Graphics*, pages 105–116. Springer, 2009.
29. 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 *Proc. Symposium on Smart Graphics*, pages 27–39. Springer, 2009.
30. B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proc. BELIV*, pages 1–7. ACM, 2006.
31. J. Stevens. Visualization of Complex Automotive Data. *IEEE Computer Graphics and Applications*, pages 80–86, 2007.
32. T. Tekušová, G. Darmstadt, and T. Schreck. Visualizing Time-Dependent Data in Multivariate Hierarchic Plots-Design and Evaluation of an Economic Application. In *Proc. Information Visualization (IV)*, pages 143–150, 2008.
33. J. Thomas and K. Cook. Illuminating the path: The research and development agenda for visual analytics. *National Visualization and Analytics Center*, 2005.
34. M. Tory, S. Staub-French, B. Po, and F. Wu. Physical and Digital Artifact-Mediated Coordination in Building Design. *Computer Supported Cooperative Work (CSCW)*, 17(4):311–351, 2008.
35. Vector Informatik GmbH. http://www.vector.com/, 03/10.