Using Visual Analytics and Information Visualization to Investigate In-Car Communication Processes

Michael Sedlmair*
BMW Group Research and Technology

ABSTRACT

Modern cars provide a large spectrum of electronic functionality which is realized by a great number of interconnected electronic control units (ECUs). The constantly increasing complexity of these in-car communication networks challenges developers in terms of understandability. The goal of my PhD research is to find novel and suitable methods of visual analytics (VA) and information visualization (InfoVis) to support engineers in gaining insight into these complex networks. With a set of self-developed prototypes and their evaluation I want to show the feasibility of my approach.

1 PROBLEM STATEMENT AND RESEARCH QUESTION

During the last few years the functionality in automobiles has increased enormously. A huge amount of enhanced functions like ACC (adaptive cruise control), rear seat entertainment systems or automatic start/stop engines were integrated with the car. They enabled step-by-step safer, more enjoyable and efficient driving. Thereby the functions became more and more distributed and so several communication networks had to be integrated with the car. In-car communication networks currently use several bus systems with different technologies (CAN, MOST, FlexRay, Lin, Ethernet) to interconnect a great number of ECUs. However, this architecture means to developers of automobiles (the target group of my work) increasing complexity, high degree of specialization and an enormous amount of data to handle. Nowadays the description of the communication processes is mostly based on textual information. The trouble is that a lot of time and experience is needed to understand processes, correlations or problems and to derive adequate activities. The major research question of my PhD research is: Can the application of VA and InfoVis improve the work with in-car networks and how do smart solutions look like?

2 METHODOLOGY AND PLAN OF RESEARCH

The goal to support engineers in their work leads to an empirical based research approach with an equated proportion of prototypes and user studies. The plan of research is divided into three parts:

First, it is necessary to get a clear understanding of the current practice and problems and to drill down the research question to concrete use cases and specific problem fields. Workshops, focus groups, user observations, interviews, questionnaires, informal discussions as well as automotive specific literature research, data analysis and tool tests help to understand the engineers' technical background, their problems, demands and their challenges. Additionally, the results of these studies provide a huge source of insight into the diversity of possible application areas.

The second research part is about building and evaluating a set of different visualization prototypes to support automotive engineers. After identifying application areas and understanding their underlying tasks and data, for each area a user-centered, iterative development process is applied to create specific, problem-centric solutions. Initially, ideas are sketched via low-fidelity paper prototypes which are used for constant discussion and evaluation with end users. The ideas for my prototypes are mostly based on (a) fine-tuned adaptations of existing visualization techniques, (b) inspirations by findings from related areas and (c) novel, domain specific visualization methods with suitable interaction techniques. Subsequently, an application design is selected, finalized and implemented as a high-fidelity prototype with access to real data. Qualitative and quantitative studies provide information about effects and potentials of the proposed visual solutions.

Finally, in the third part I want to summarize the practical experience from the preceding work to provide generalizable solutions and comprehensive approaches. This includes (a) identifying correlations between specific solutions, (b) analyzing how specific solutions complement, and (c) outlining a process-comprehensive, visual based solution. At last, it might also be of interest to see if we can generally learn about how to improve the effective application of VA/InfoVis in an industrial environment.

3 PROGRESS TO DATA

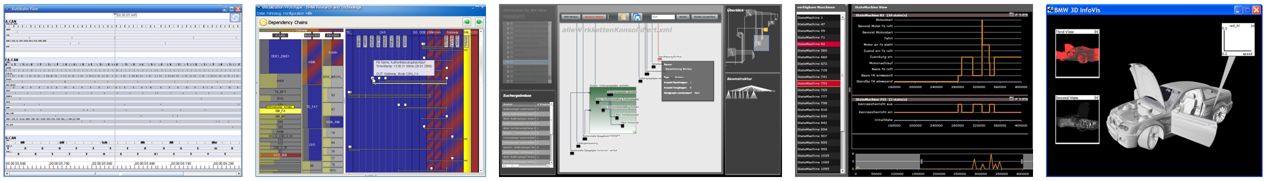
Current Practice and Problems: Hitherto, I conducted several empirical studies with a plurality of engineers from different departments at BMW to get a clear understanding of the target area.

During initial studies one important general characteristic became clear: Diversity! First, there are lots of different use cases and application areas in the field of in-car communication networks., e.g., functional planning and construction, developing soft- and hardware, testing, analyzing and diagnosing, or maintenance (Diversity of use cases and application areas). Second, in all this areas the data appears in different forms, structures and terms. During functional design, e.g., the engineers work with abstract features, clustered in systems and sub-systems. During soft- and hardware development the data is mapped on functions, functional blocks, signals and messages and to the hardware components – ECUs and bus systems. Analysis and diagnosis experts on the other hand work with trace files, recorded directly from the bus systems (Diversity of data appearance). In all areas different tools are used, specific tasks have to be performed, e.g., constructing a system, develop software or finding errors, and varying problems appear (Diversity of tools, tasks and problems).

After the initial studies, I started to concentrate on the area of analyzing and diagnosing in-car communication networks. From this point of view I look at several other application areas to learn about similarities, differences and approaches for generalizable solutions. However, to get a more precise understanding I will focus in the following on the area of analyzing in-car-communication processes.

The underlying data in this area are *trace files*. Within this trace files all messaging information of the in-car network is logged. In addition to the plain messages, the traces contain error entries which can be classified in automatically detected errors (an ECU detects an error) and manually detected errors (the driver detects an error, presses a specific button and annotates the initialized entry in the trace later on). By taking into account that there are up to 15.000 messages per second and that traces are recorded on weekend test runs, this leads to enormous trace files with up to 10GB raw data.

*e-mail:michael.sedlmair@bmw.de



(a) Autobahn View (b) Dependency Chains 1 (c) Dependency Chains 2 (d) Statemachines (e) 3D Model Vis.

Figure 1: Screenshots of current prototypes. For more detail please see referenced papers.

Traces are currently handled, interpreted and analyzed with specific analysis tools (e.g., Canalyzer[1]). With the help of these tools the engineers try to locate the source of errors to subsequently inform responsible developers to fix it. However, for several reasons locating error sources is not trivial, e.g.: (a) in most cases the ECU which detected an error is not the actual source of this error, (b) there are multiple reasons for misbehavior, e.g., message displacement, timing problems, hardware defects, and so on, (c) especially manually detected errors are enormously delayed within the trace. An analysis expert's typical procedure to locate an error then is to start with getting a general overview. Then s/he starts locating points of interest in the trace and try to get a clear understanding of the context in these situations. Afterwards s/he combine the obtained information to get insight into the complete situation and thus to precisely deduce the error source. However, this workflow is not properly support by current tools because: (a) tools are text based and represent the information mostly in lists and therefore fail to provide overview and correlation information, (b) fast navigation within the data is not well supported, (c) sophisticated data reduction and data mining techniques are not supported (d) the tools often fail to provide multiple different perspectives on the data, (e) neither distributed nor co-located collaboration is supported.

My top-level goals by applying VA/InfoVis are therefore: (a) gain working speed (e.g. finding the same amount of errors in less time), (b) more insight and better understanding, (c) discovery of novel aspects within the data, (d) providing different perspectives on the data and (e) a higher degree of successful communication and collaboration.

Visualization Prototypes: To learn how to overcome the drawbacks several prototypes were designed, built an evaluated:

The *AutobahnView* (cf. fig. 1-a) is a visualization prototype to support overview and navigation in bus traces. It is based on the metaphor of a crowded highway and represents messages horizontally ordered by time and vertically by their sending ECU. The AutobahnView uses zoom and pan interaction and provides a tight coupling to an additional raw data view. Qualitative user studies showed that the AutobahnView is useful for fast navigation, detection of message bursts and identification of repetitive patterns [3].

The usage of so-called *dependency chains* (i.e., directed graphs between functional blocks) allows analysis experts (a) to accurately track causal correlations in order to detect error sources and propagations and (b) to reduce the trace data by excluding causal redundant and irrelevant information. Two prototypes using dependency chains were built. The first prototype is a dual-view approach (cf. fig. 1-b) showing physical dependencies with a treemap-like approach and functional dependencies in a highly interactive message sequence chart visualization. Both views interactively highlight the transitive hull (i.e., all predecessors and successors) surrounding a selected functional block [2]. A second prototype is based on hierarchical clustered, semantically zoomable node-link diagrams and represents all dependencies within an in-car communication network (cf. fig. 1-c).

Another approach to reduce traces is the usage of *state machines*. State machines (e.g., a Door state machine with two states "open" and "closed") are defined upfront by analysis experts and can be applied to the trace. Afterwards a timeline based and zoomable visualization (cf. fig. 1-d) allows the user to have an overview, browse, search and examine the behavior of and the correlations between predefined machines.

Several experiments with *3D model visualizations* were made in order to bridge the gap between electronic and mechanical information. Fig. 1-e shows a screenshot of a configurable 3d model view that can freely be coupled and coordinated with other applications. The coordination can either be linking and brushing elements, e.g. hovering a message and highlighting the dedicated sending ECU in the 3D model view, or semantic linking, e.g., navigating over time and replay "real" behavior in the 3d model [4].

MostVis is not directly located in the area of analyzing bus traces but is a helpful visual based tool to browse and search function catalogs. Function catalogs contain detailed information and specification about all involved units of one bus system (ECUS, functional blocks, software functions, parameters, etc.) Quantitative user studies with a MOST-function catalog containing over 44.000 data items showed that MostVis is significantly faster for browsing-, searching- and overview-tasks than current tools [1].

4 Future Work

First, there is ongoing work on a couple of further prototypes, including experiments with interactive tables and prototypes for other application areas, which have to be accomplished. Second, more profound user studies are necessary to evaluate the prototypes' benefits in real world scenarios. Current studies were often restricted in time and transferability to daily practice. Reasons could be seen in the scarce availability and time restrictions of experts and in the detached character of the prototypes. Therefore, it is planed to closer integrate the solutions with currently used tools to allow the users an easy and time-independent access to the prototypes. In doing so, I want to study the concepts in a real world, day-to-day environment. Third, I want to step back and consider how to derive general guidelines and comprehensive understanding from the specific solutions. I want to outline how solutions could be integrated in a homogeneous environment and investigate what fundamental hardware and software setups are valuable (cf. section 2).

References

[1] 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 13th Int. Conference on IV (to appear)*, July 2009.

[2] M. Sedlmair, W. Hintermaier, K. Stocker, T. Bring, and A. Butz. A dual-view visualization of in-car communication processes. In *Proceedings of the 12th Int. Conference on IV*, July 2008.

[3] M. Sedlmair, B. Kunze, W. Hintermaier, and A. Butz. User-centered Development of a Visual Exploration System for In-Car Communication. In *Proceedings of 9th Int. Symp. on Smart Graphics*, May 2009.

[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 9th Int. Symp. on Smart Graphics*, May 2009.

[1] www.canalyzer.com