

MostVis: An Interactive Visualization Supporting Automotive Engineers in MOST Catalog Exploration

Michael Sedlmair, Christian Bernhold, Daniel Herrscher, Sebastian Boring, Andreas Butz
BMW Research and Technology, University of Munich
{michael.sedlmair, christian.bernhold, daniel.herrscher}@bmw.de,
{sebastian.boring, andreas.butz}@ifi.lmu.de

Abstract

The MOST bus is a current bus technology for connecting multimedia components in cars, such as radios, navigation systems, or media players. The bus functionality is described in a large hierarchically structured catalog of some 4'000 entries. Browsing this catalog has become infeasible on paper as well as with currently used textual database interfaces. An observation of current work practices has revealed many problems and inefficiencies. We describe the (iteratively developed) design of MostVis, a visual tool for exploring MOST function catalogs, as well as an evaluation of our implemented prototype. Our design carefully adapts existing visualization techniques and combines them in a multiple coordinated view (MCV) approach to satisfy the specific needs of our target group. With this paper, we hope to provide a living example of how existing general-purpose techniques can be successfully trimmed and tailored for a very specific audience.

Keywords—MOST, Visualization, In-Car Communication

Video

An additional video describing the prototype implemented during this work is available at: www.medien.ifi.lmu.de/team/michael.sedlmair/

1 Introduction

Large hierarchies appear in many different forms and environments, in file systems, in social structures and, not surprisingly, also in datasets of in-car communication networks. Visualizing such hierarchical datasets is an important topic in the domain of information visualization. Two main challenges are the handling of very large data sets and the design of adequate interactive exploration techniques. A number of general-purpose concepts and technologies have been developed and integrated into tools. However, for many real world scenarios, exclusively applying general approaches is not sufficient. Combined approaches and customized, fine-tuned solutions are required to exactly focus on a special target group's needs or to enable

the integration of specific requirements.

Automotive engineers of in-car communication networks are one of these “special target groups”. Many specialized concepts and small scale solutions have emerged over the years and helped to ease routine procedures. However, the constant integration of novel, distributed functionality within the automobile has caused an enormous increase in the complexity of in-car-communication networks. By observing daily work practices, we found that the information each engineer had to deal with, has multiplied in the last years and that understanding the underlying data has been made difficult. Although the conventional text-based tools are still helpful, they are approaching their limits. Novel solutions have to be developed for enriching or complementing these conventional tools.

The MOST (Media Oriented Systems Transport) bus is a prominent example of this increased complexity. MOST is one of several in-car communication technologies. The static description of all MOST functions is compiled in so-called *MOST function catalogs* which either can be accessed by a textual database interface, a pdf document or an xml version. The catalogs are used as information sources by many engineers during the development process. However, those engineers are facing more and more of a challenge, because the current content of MOST function catalogs has increased enormously over the last few years. A printed version of an up-to-date catalog adds up to a stack of 4'000 pages. Currently used tools are mostly based on text and reach their limits for a usable and understandable representation of these datasets. Their use requires much time and a high degree of expert knowledge. We propose a more visual approach for browsing MOST function catalogs. Our goals are to better support browsing and searching tasks for non- and medium-expert users (in the following grouped to “MOST-Non-Experts”), but at the same time provide expert-users (“MOST-Experts”) with additional support for advanced exploration. Both groups frequently work with MOST function catalogs and therefore have to be addressed.

2 Technical Background

MOST is a state-of-the-art networking technology to connect infotainment devices in luxury class automobiles. The MOST Cooperation¹ standardized both the physical layer of the technology and the upper layers, including the general concepts of function addressing, coding of arguments etc. The definition of concrete functions, however, is usually done by each car manufacturer on its own. Because the various infotainment devices in a car, such as the navigation system, audio amplifier, hands-free telephone etc. come from different suppliers, the definition of MOST functions for these devices is an essential part of the requirements specification given to respective suppliers.

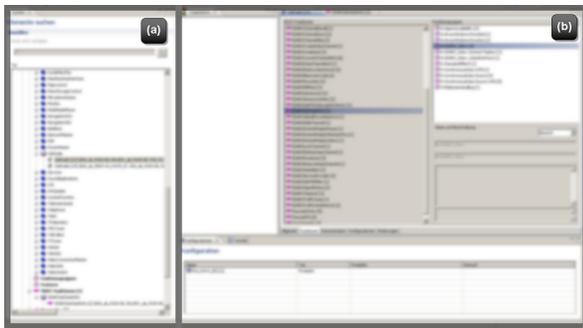


Figure 1: Database interface with (a) an overview and navigation frame and (b) a frame for showing all detailed information within different sub-frames²

The specification process for MOST functions today is based on a database solution with a custom, text based front-end (cf. fig. 1). Several engineers work with the in-house tool, adding new functions and descriptions, and altering existing ones. Many others use the database for informational purposes, e.g., to browse for a specific function description. After a “feature freeze” for a specific vehicle, PDF and XML dumps are generated from the database. These documents are part of the requirement specification for a device and given to the respective device suppliers. In addition to that, the PDF dump (cf. fig. 2) is used throughout the company as quick reference for device functions, especially by MOST-Non-Experts without access to the live database.

This process is now being challenged by the increasing complexity of distributed infotainment functions. The current MOST function catalog consists of more than 3’500 functions, each with several parameters and additional documentation. Especially for MOST-Non-Expert users, it is

very hard to get an overview of the MOST functions that belong to one device, or to find a specific customer function.

MOST functions are organized as follows: Each function has several so-called “op-types”, which indicate the semantic of the operation such as “request” or “response”. Each op-type can have its own set of parameters. Functions that belong to a certain functionality are grouped together in “functional blocks” (FBs), similar to a class in the object-oriented world. A device, so-called “electronic control unit” (ECU), implements several FBs. Finally, several devices work together in one car. For further information about MOST we recommend [10].

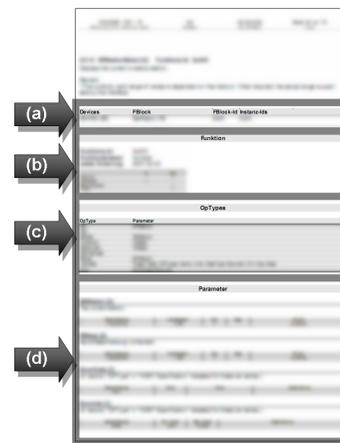


Figure 2: An example page of the PDF catalog with information about (a) an specific FB including (b) its functions, (c) otypes and (d) parameters; Altogether, there are over 4’000 pages

3 Related work

MostVis is based on interactive multiple coordinated views (MCV) and represents the underlying hierarchy via a node-link-representation.

MCV has become a very helpful technique in information visualization application design. It has been found in many scenarios, that MCV improve user performance, e.g., [8, 16, 17]. However, it is important to design MCV applications very carefully to reduce unnecessary context switching and efficiently use display space [1, 6, 7].

The problem of hierarchical information representation has been investigated for a long time and several tree visualization solutions have emerged over the years. Generally, tree visualizations can be classified into node-link

¹<http://www.mostcooperation.com/>

²Parts have been blurred in order to protect IPR of BMW, this also applies for fig. 2, fig. 3, fig. 4, fig. 5-b and tab. 1 (where original names are replaced by *Element-xxx*)

representations and space-filling approaches [20]. In both areas a variety of layouts has been investigated. Prominent space-filling approaches are Treemaps [21] with its several design variations [2, 5] and radial ring layouts [23, 24]. Likewise, in the area of node-link representations, lots of different 2d [13, 14] and 3d [15] layouts have been developed and evaluated. A combination of both approaches can be found in [25]. Throughout the process of finding novel layouts, a lot of effort was spent to make the tree visualizations scalable to large data sets, e.g., [9, 15]. User studies about how and when to use which layout have been investigated in [3, 23]. Because of the automatically increasing display space for each layer of the hierarchy, radial layouts turned out to be advantageous for the representation of large and broad hierarchies.

Browsing large hierarchies is a common task in many different application areas. Nearly every computer user organizes her/his files, music, pictures, etc. in hierarchical systems and accesses the data via folder structures. While widespread approaches such as the windows explorer are mostly text based, there are also several tree visualization based tools for browsing large hierarchical data. The PDQ Tree-browser [12], 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 [4, 11, 18, 19]. While basic tree visualization techniques are frequently reused, all these application areas, however, also required different application designs, interaction concepts and fine-tuned solutions.

4 User-centered Design

Analyzing current working practices and tools in use (cf. section 2) showed that many engineers have to work with the MOST function catalog more or less frequently, regardless whether they are MOST-Experts or not. Especially MOST-Non-Experts, however, can hardly derive advantages from the 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 in interactive browsing, advanced searching and gaining insight into relational or overview aspects. Not surprisingly, due to their long-time experience, MOST-Experts did not have major problems in working with the currently available tools. Interviews, however, revealed that they demand 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.

In order to fill this gap with an adequate system we started to conduct a user centered design approach. We designed a series of lightweight, highly interactive visualization concepts and presented our ideas in the form of paper prototypes to five domain experts. Discussing the ideas with the experts helped us to evaluate the presented novel concepts. Based on the outcomes of these studies we chose the most promising concept and elaborated it according to the experts input. In a second iteration we presented the elaborated concept to experts and subsequently finalized it based on their feedback. The final concept was implemented as a java prototype using the *prefuse*³ framework.

5 MostVis

The resulting application MostVis is based on a MCV approach and visualizes MOST function catalogs of a specified vehicle series which can be selected upfront. The catalog is loaded in the form of a hierarchically structured xml exported from the MOST database. The xml contains several different hierarchies which can be swapped in MostVis during runtime. Though it is very important for the engineers to provide different perspectives on the data, we will focus in the remainder of the paper - for a better understanding - on the basic MOST hierarchy which is structured as follows: Car series (root node), ECUs (1st layer), FBs (2nd layer), functions (3rd layer), parameters and op-types (4th layer, leafs).

5.1 Visualization and Basic Features

The *main view* (cf. fig. 3-a and 4-a) of MostVis is positioned in the center of the application and represents the MOST function catalog data by means of a node-link tree visualization. Each element of the MOST function catalog is allocated to a node, and the edges depict the hierarchical correlations. Besides a textual label, each node incorporates an additional icon representing the element's type. The icons are inherited from the currently used database interface to retain knowledge and to support established recognition patterns. The general navigation is realized via zooming (scroll wheel) and panning (left click on the background). Additional shortcuts enable fast zooming (double left click on the background) and automatic alignment (double right click on the background).

The main view provides two different modes: *local view* and *expanded view*. The local view (cf. fig. 3-a) is based on a easy-to-navigate, horizontal, left-to-right tree layout. Initially 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* (single click) and *expanding* (double click) nodes. By selecting, the associated node

³<http://prefuse.org/>

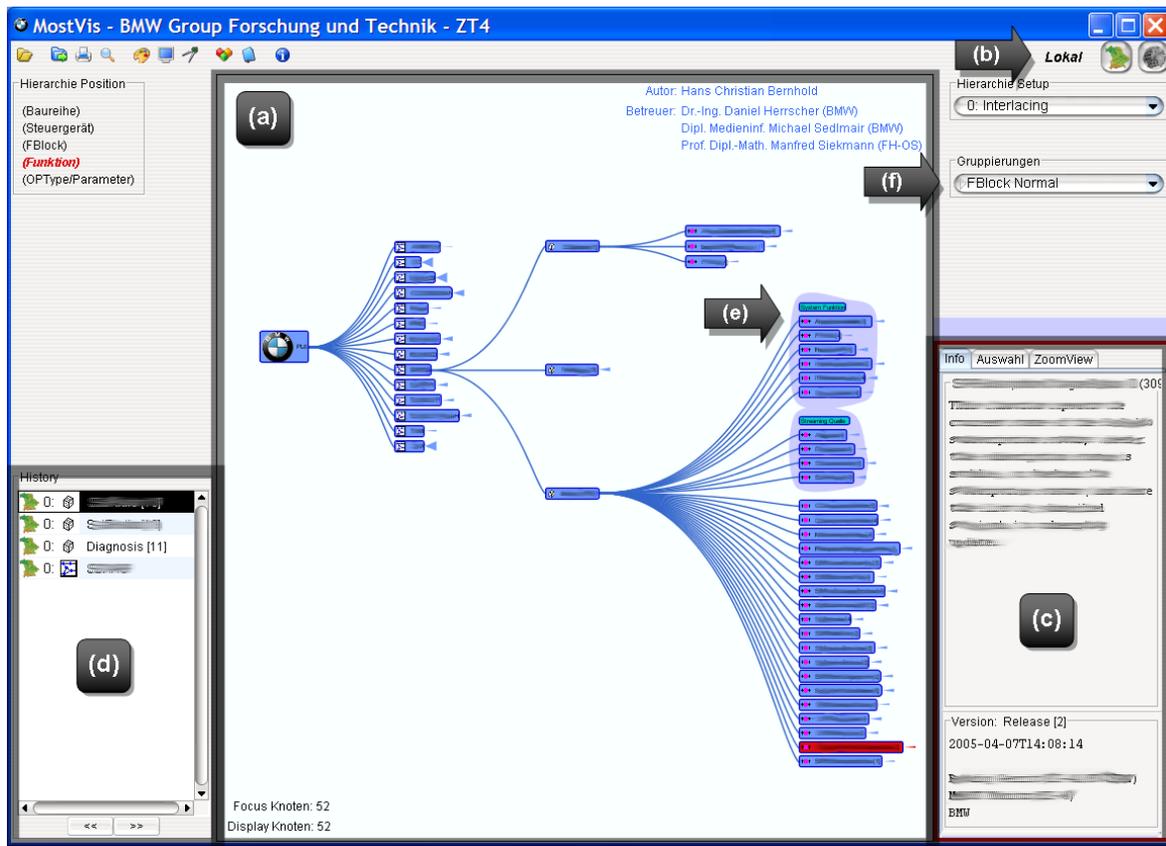


Figure 3: MostVis with (a) local view: left-to-right presentation of the MOST function catalog, (b) mode selection: shows current mode and allows the user to change modes, (c) detail view: shows additional information to the selected node, (d) history with undo and redo, (e) a set of grouped elements, and (f) grouping dialog

is highlighted and additional detailed information is shown in an extra *detail view* (cf. fig. 3-b). Expanding a node also selects it and, additionally, extends its children with a short animation. We integrated a mechanism which automatically reduces “old” branches when a “new” node is expanded. However, if the user intends to show branches in parallel, (s)he also can accomplish this by holding the ctrl-key. In doing so, (s)he can autonomously explore the hierarchy without being overwhelmed by the entire information or distracted by unnecessary interactions. Therefore, this approach well matches Shneiderman’s information seeking mantra “Overview first, Zoom and Filter, then Details on Demand” [22]. The user starts with a rough overview showing only the first level of the hierarchy. Interactively unfolding and folding subtrees enables navigation into regions of interest and details can be found for each node by looking into the detail view. Additional *angle visualizations* help the user to get a better orientation in the local view. Each visible folded node gets a shortcut 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. If the node is extended the triangle is replaced by the “real” edges.

Switching to the expanded view (cf. fig. 4-a) enables the user to get insight into overview aspects of the underlying MOST function catalog or to examine elements in their global context. The expanded view is based on a hyperbolic 2D tree layout [13] which is adequate for an overview representation of the broad and flat MOST hierarchies. Indeed, this view on the one hand enables displaying the entire catalog. However, due to the enormous amount of data it is also obvious that this is challenging in processing the information - both for the computer and the human perceptual system. So on the other hand we integrated a *filter* where the user can dynamically set the demanded number of displayed layers starting from the root node (cf. fig. 4-c). This turned out to be very useful for the engineers because their demands showing overview information is mostly constrained to lower layer information,

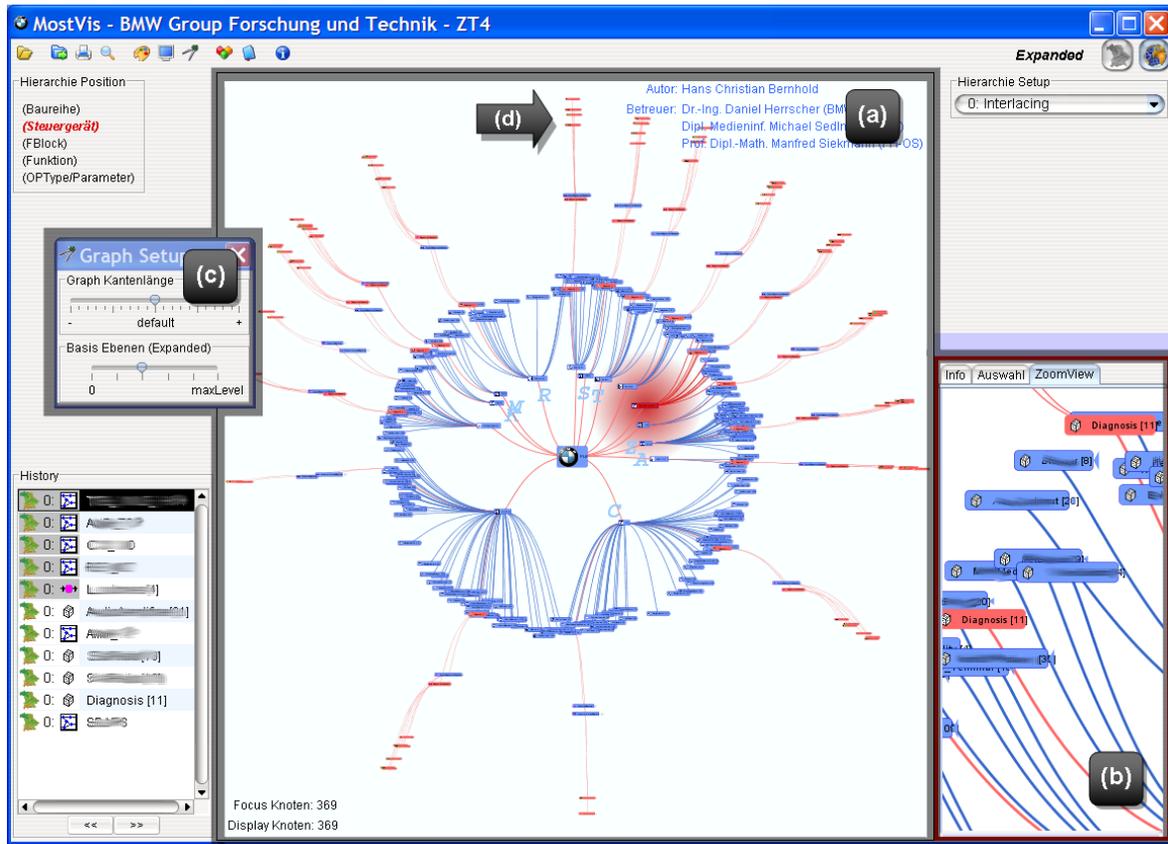


Figure 4: MostVis with (a) expanded view: hyperbolic presentation of the MOST function catalog, (b) zoom view: provides an adaptive magnification, (c) base level and edge length dialog, currently two levels are completely shown, and (d) highlighted search results in red

such as showing all ECUs with their associated FBs. An entire overview on the contrary is rarely required. Due to the fact that the filtered overviews still are very big, which complicates the readability of labels, we integrated a so-called *zoom view*. The zoom view is either available via an integrated tab in the lower right corner (cf. fig. 4-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.

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. Therefore, 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. This helps the user to maintain a clear orientation.

5.2 Additional Features

In addition to the basic visualization we integrated a set of useful functionalities to support the user in interactively exploring the data.

Probably the most important feature is the *search*. Via a search dialog (cf. fig. 5-a) the user can conduct varying kinds of search queries. For instance, (s)he can either search the entire catalog or constrain the query to subtrees, or (s)he can chose a specific data type to search for. The results of the search query are visualized in diverse, coordinated forms. First, an additional *list view* (cf. fig. 5-b) pops up after conducting the search query. It textually lists all found elements together with their path within the hierarchy and provides an established method of representation. Second, the results are highlighted in the tree visualization by coloring the dedicated nodes as well as their paths. Depending on which main view is currently opened, either the search results are shown sequentially or all at once. In the local view only the first result is shown by expanding the related branch. This avoids that the view becomes confus-



Figure 5: Additional search views

ing in the case of many search results. The user now can interactively select items of interest within the list which automatically causes the selected search result to appear. In addition, the path highlighting allows him/her to manually navigate alongside the paths to each search result. To graphically represent all search results at once the user has to go to the expanded view. The expanded view extends all branches to search results and highlights their nodes and paths (cf. fig. 4-d).

Another feature is the *grouping* functionality. The user is able to specify grouping patterns, either automatically during database exports or manually using meta-knowledge. The grouping patterns in turn are stored in an xml file and can be imported into our tool. The user now can group child elements according to the chosen grouping pattern (cf. fig. 3-d). The grouping feature was highly demanded by the MOST-Experts. It depicts one currently unsolved complexity problem of long, ungrouped lists of elements. An example will clarify the problem: The number of functions in a FB can be more than 100. Current tools list all functions in alphabetical order which hinders the user in quickly finding necessary information. On the other hand there already exist meta-information based grouping patterns which are merely used by the engineers in an informal manner. For instance, some functions are management functions that belong to each FB, others are common for all digital audio sources and sinks, and there are of course a lot of application-specific functions. Our grouping function therefore uses this additional meta-information to enable dynamic and storable clustering of elements.

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 left lower corner (cf. fig. 3-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 “short-cut 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 as well as a label and icon of the selected element are provided. This helps the user to get a better orientation within the history.

Further additional features are *exporting* and *printing*. The export was demanded by the engineers to support re-integration of information in other tools or to get fast access to representation material. Therefore, we support excel exports (e.g., for search results) and image exports of each tree view. Especially for these features it is helpful that the layouts are *customizable*. Edge length, colors, etc. can be changed by the user.

6 Expert User Study

We conducted an expert user study with our target-group engineers to evaluate MostVis in terms of performance, understandability as well as likability aspects. In this section we first describe our study design and the users’ tasks. We then show both qualitative and quantitative results of our study followed by a discussion.

6.1 Study Design and Tasks

We conducted the study with 14 participants (aged 20 to 39 years, all male) that are all automotive engineers. Their level of expert knowledge regarding MOST varied (average of 2.5 years) which reflects the current situation in the automotive industry. Before the study began, each of the participants had to fill a (1) *pre-questionnaire* to gather information about the subject’s current working practices and level of MOST expertise. Subsequently, the participants received an (2) *introduction of MostVis* and its features which usually lasted 15 to 20 minutes. Additionally, each user accomplished nine (3) *example tasks* in order to verify that he entirely understood both the application and the interaction concepts.

After the practice period, the (4) *main part of the user study* followed. The participants were asked to perform eleven tasks with increasing complexity. The tasks were derived from the analysis phase (cf. section 4) and dealt with exploring, browsing and searching within the MOST

Table 1: Tasks (paraphrased from German), gray marked rows were excluded from the results

Nr.	Task	Classification
1.	Find all functions of the FB <i>FB-aaa</i> !	lightweight
2.	Sort the following three ECUs by means of their complexity (equal to number of FBs): <i>ECU-bbb</i> , <i>ECU-ccc</i> , <i>ECU-ddd</i> !	lightweight
3.	Name 3 functions of the FB <i>FB-eee</i> that contain <i>booleans</i> !	advanced
4.	Show the description of the <i>FB-fff</i> that is located in the <i>ECU-ggg</i> <i>FB-hhh</i> !	lightweight
5.	How many <i>enums</i> are located in the <i>ECU-iii</i> ?	advanced
6.	Return to task 4!	lightweight
7.	Name 3 functions from the FB <i>FB-mmm</i> which are assigned to its! <i>sinks</i> and 3 functions which are <i>sources</i>	advanced
8.	What is the element <i>Element-nnn</i> and where can you find it?	lightweight
9.	Is there an available function in the <i>FB-ooo</i> that gives you information about the currently selected radio station?	lightweight
10.	Give an overview over all elements contained in <i>ECU-jjj</i> (FBs, functions, OpTypes, parameters)!	advanced
11.	Name 3 ECUs that are member of the group <i>Group-kkk</i> as well as 3 of the <i>Group-lll</i>	advanced

function catalog which are important tasks in employees' every-day work practices. Additionally, all tasks were pre-evaluated in a pilot study with a MOST-Expert as well as a MOST-Non-Expert user. We classified the tasks in two categories, (a) *lightweight tasks* (Searching and browsing elements/information) and (b) *advanced tasks* (aggregated search queries, grouping, overview; in general, tasks which are hardly realizable with current tools but were demanded by the engineers, cf. section 4). Table 1 shows a list of all tasks.

The participants had to pass through the eleven tasks twice, on the one hand with MostVis and on the other hand with their preferred MOST tool(s) (pdf and/or database interface; asked in the pre-questionnaire; xml was also offered, however never chosen). The order of the tools was alternated between participants to minimize fatigue effects. The order of the tasks was constant. With both tools, we measured the task completion time in seconds for each task, beginning after the participants were instructed with their task (i.e., reading the description) 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 is not feasible, he could also proceed to the next task. 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. We chose the independent variables as *Task* and *Tool* (*preferred MOST tool*, *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 which was measured by the amount of time the participants used the MOST functional catalog (asked in 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 tak-

ing notes (due to IPR security requirements we were not allowed to record audio or video-tape). Additionally, after the main user tests, each participant was asked to fill out a (5) *post-questionnaire* with qualitative questions regarding the acceptance and likability of MostVis.

6.2 Hypotheses

Based on our understanding of the problem field, currently used tools and the tasks, we had four hypotheses: First, we expected MostVis to outperform current tools with MOST-Non-Experts independent of the task (H1). Second, for MOST-Experts we estimated MostVis to be better for advanced tasks (H2), because they initially were stated by the engineers to be hardly realizable with current tools. Third, we hypothesize that MostVis will have a similar performance for all of our tasks while current tools will strongly depend on the task, lightweight or advanced (H3). H3 is therefore deducible from H1 and H2. And fourth, we expected that the ratings of MOST-Non-Experts will be considerably better for MostVis compared to current tools (H4).

6.3 Results

Repeated measures analysis of variance showed that there was 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. 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). Those that were able to execute the task also had high task completion times (averages of 161.4 seconds for MOST-Experts and 134.3 seconds for MOST-Non-Experts respectively). Using MostVis, however, the average task completion times were much lower (14.6 seconds for MOST-Experts and 27.4 seconds for MOST-Non-Experts respectively). This shows that these *Tasks* have to be treated as outliers and are not applicable for comparison leading to their removal from further analysis.

6.3.1 Selection Time

For the remaining 9 *Tasks*, we analyzed the measured task completion times. We performed a $9 \times 2 \times 2$ (*Task* \times *Tool* \times *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* \times *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. 6). Therefore, our hypotheses H1, H2 and H3 are supported.

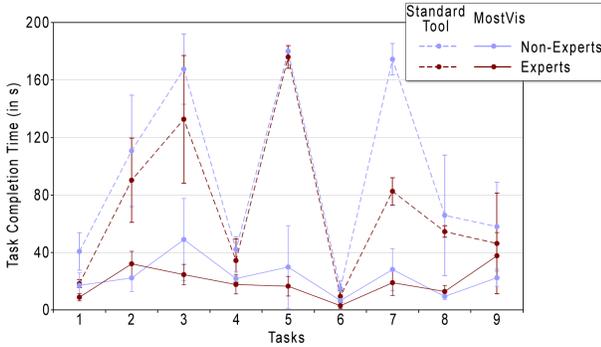


Figure 6: Average selection times for MOST-Expert users versus MOST-Non-Experts depending on the used tool.

Besides achieving significantly better results with MostVis, the saved time especially for advanced tasks is of great importance. As shown in table 2 the average decreased 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.

Table 2: Improvement of selection times using MostVis

	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

6.3.2 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 of the different views, participants ranked them 1.33 on the mentioned scale. This generally 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 extreme large data sets that partly made the expanded view confusing for the users. These findings encourage us to perform future investigations regarding potentials and room for slight improvements with the expanded view. 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, here are some exemplary excerpts (paraphrased from German): (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 overview is not available in current tools” (1 subject). In addition to the general positive feedback and acceptance (a) the subjective ratings indicted improved productivity (b), presence of learnability (c) and overview support (d). These statements furthermore support the usage of our MostVis tool.

6.4 Discussion

As already shown in the results, MostVis performed overall better than currently used tools. This can be understood by providing better browsing and searching functionality needed in large data sets 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 (advanced tasks). 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 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, in terms of MOST-Non-Experts (H4), these results were not expected from MOST-Experts as we assumed that they are highly familiar with current tools. However, the results show that the visibility of coherences embedded in MostVis can explain the positive ratings from the MOST-Experts. Even though none of the participants was missing a key feature in our system, we believe that new tasks could be conducted with MostVis.

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 referring to richness of functionality (database application) and to the intention of documentations (pdf), 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.

Summary and Future Work

Applying information visualization to a complex domain, in which the only access to data was textual so far, is a promising endeavor. The thoughtful adaptation and modification of existing visualization techniques, done in a user-centered, iterative, design process, has led to the MostVis prototype described in this paper. It supports the highly specific target group of automotive engineers of in-car communication networks at different levels of expertise. We have evaluated our prototype in a user study with the target group, using real data, and received positive results and feedback.

In our future work, we intend to enhance MostVis to a wide range applicable version which can then be used by a larger number of engineers on a day-to-day basis. This version will have to be directly coupled to the MOST database, provide even richer detail information and work in different setups. Furthermore, in our study, we received a number of useful requests for additional functionality (e.g., a direct integration of grouping into MostVis).

Acknowledgements

This work was funded by BMW Forschung & Technik GmbH, as well as the state of Bavaria. We would like to thank all of our study subjects for sharing their valuable time and insights with us.

References

- [1] M.Q.W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces*, pages 110–119. ACM Press New York, NY, USA, 2000.
- [2] M. Balzer and O. Deussen. Voronoi Treemaps. In *Information Visualization, 2005. INFO VIS 05. Proceedings of the 2005 IEEE Symposium on*, pages 7–7, 2005.
- [3] T. Barlow and P. Neville. A Comparison of 2-D Visualizations of Hierarchies. In *Proceedings of the IEEE Symposium on Information Visualization*, volume 2001, pages 131–138, 2001.
- [4] B.B. Bederson. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 71–80. ACM New York, NY, USA, 2001.
- [5] T. Bladh, D.A. Carr, and J. Scholl. Extending Tree-Maps to Three Dimensions: A Comparative Study. *LECTURE NOTES IN COMPUTER SCIENCE.*, pages 50–59, 2004.
- [6] S.K. Card, J.D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.
- [7] 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 cmv*, 2003.
- [8] RM Edsall, AM MacEachren, and L. Pickle. Case study: design and assessment of an enhanced geographic information system for exploration of multivariate health statistics. In *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on*, pages 159–162, 2001.
- [9] J.D. Fekete and C. Plaisant. Interactive information visualization of a million items. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 117–124, 2002.
- [10] A. Grzemba. *Most: The Automotive Multimedia Network*. Franzis Verlag, Poing, Germany, 2008.
- [11] P. Isenberg and S. Carpendale. Interactive Tree Comparison for Co-located Collaborative Information Visualization. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, pages 1232–1239, 2007.

- [12] 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.
- [13] 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.
- [14] 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*, pages 1414–1426, 2006.
- [15] T. Munzner et al. H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. In *Proceedings of the 1997 IEEE Symposium on Information Visualization*, pages 2–10, 1997.
- [16] G.M. Namata, B. Staats, L. Getoor, and B. Shneiderman. A dual-view approach to interactive network visualization. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 939–942. ACM New York, NY, USA, 2007.
- [17] C. North and B. Shneiderman. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In *Proceedings of the working conference on Advanced visual interfaces*, pages 128–135. ACM New York, NY, USA, 2000.
- [18] R.D.M. Page. Tree View: An application to display phylogenetic trees on personal computers, 1996.
- [19] C. Plaisant, J. Grosjean, and B.B. Bederson. Space-Tree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation, 2002.
- [20] H.J. Schulz and H. Schumann. Visualizing Graphs-A Generalized View. In *Proceedings of the conference on Information Visualization*, pages 166–173. IEEE Computer Society Washington, DC, USA, 2006.
- [21] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics (TOG)*, 11(1):92–99, 1992.
- [22] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343, 1996.
- [23] 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.
- [24] J. Yang, M.O. Ward, and E.A. Rundensteiner. Inter-ring: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis' 02)*, page 77. IEEE Computer Society Washington, DC, USA, 2002.
- [25] S. Zhao, M.J. McGuffin, and M.H. Chignell. Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams. In *Proceedings of IEEE Symposium on Information Visualization*, pages 57–64, 2005.