On Complexity Reduction of User Interfaces for Safety-Critical Systems

Andreas Holzinger[1], Evgenia Popova[1], Bernhard Peischl[2], and Martina Ziefle[3]

[1] Medical University Graz, A-8036 Graz, Austria
Institute for Medical Informatics, Statistics & Documentation,
Research Unit Human-Computer Interaction
{a.holzinger,e.popova}@hci4all.at
[2] Softnet Austria, A-8010 Graz, Austria
bernhard.peischl@soft-net.at
[3] RWTH Aachen University, D-52062, Germany
Communication Science, Human Technology Centre (Humtec)
ziefle@humtec.rwth-aachen.de

Abstract. Control and communication systems used at power plants or incineration facilities offer various graphical visualizations of the physical parts of the site; however, they rarely provide sufficient visualization of the signal data. There is the problem, that such facilities contain 10,000 or more data acquisition points; each of them continuously sending data updates to the control system (once in 20 ms or less). This huge load of data can be analyzed by a human expert only if appropriately visualized. Such a visualization tool is AutoDyn, developed by the company Technikgruppe, which allows processing and visualizing complex data and supports decision making. In order to configure this tool, a user interface is necessary, called TGtool. It was originally developed by following a system-centered approach, consequently it is difficult to use. Wrong configuration can lead to incorrect signal data visualization, which may lead to wrong decisions of the power plant personnel. An unintentional mistake could have dramatic consequences. The challenge was to re-design this tool, applying a user-centered approach. In this paper we describe the re-design of the configuration tool, following the hypothesis that a user-centered cognitive map structure helps to deal with the complexity without excessive training. The results of the evaluation support this hypothesis.

Keywords: Complexity reduction, data visualization, process visualization, usability engineering, safety-critical systems, industrial design.

1 Introduction and Motivation for Research

Simplicity and complexity seem to be opposites. However, there is an interesting asymmetry in their opposition. It is as though absolute simplicity is the center point of an n-dimensional sphere, where absolute complexity is the surface. In other words, there are many ways of moving away from simplicity toward complexity. In the

G. Quirchmayr et al. (Eds.): CD-ARES 2012, LNCS 7465, pp. 108–122, 2012.

iterative process of design and behavioral observation, it is relatively difficult to move inward toward greater simplicity, while it is relatively easy to move along the surface of a sphere, resolving some types of complexity while simultaneously introducing others, and therefore increasing neither overall simplicity nor ease of use [1].

Similarly, many dimensions are involved in Human-Computer Interaction (HCI). Each of them influences the overall complexity, consequently the usability of the graphical user interface (GUI).

System complexity should be hidden from end-users, so that they can concentrate on their tasks. Task complexity can also vary, e.g. mobile phone prototypes sending a message belongs to simple task and saving a schedule to complex tasks [2]. A GUI is supposed to support a user through the tasks; however, a useful GUI is not necessarily usable. A GUI is representing the system in front of a user, thus visual complexity should be adjusted.

System complexity, task complexity and visual complexity, contribute to the overall graphical user interface complexity. The elements of the GUI are perceived by the users, who immediately create a mental model (cognitive map) of the system; such mental models help users to interpret perceptions from the environment, in order to work out appropriate actions. Simplified, we can say that external visualizations are internalized as mental models [3]. The complexity of such a model is referred to as cognitive complexity.

In this paper we are dealing with intelligent user-centred data visualization, in a safety-relevant area, including decision support within a control system of a power plant. Advantages of user-centred approaches will be demonstrated at the example of a configuration tool called TGtool developed by the company Technikgruppe Mess-, Steuer- und Regeltechnik GmbH. In order to describe what this tool is capable of and what it is responsible for, two other systems are described briefly:

Control System: Technikgruppe is working in the power plant branch. Each power plant has its own control system produced by various corporations including Siemens, ABB or OPC Foundation. A control system can be seen as an information system (IS) for the whole facility. The system keeps track of all the measuring points, their settings and the data. Each measuring point (sensor) sends regularly its measurement (signal value) to the control system. All this data is saved; however, there is no sufficient way to visualize signal data in a longitudinal way on the time axis.

Decision-Support System AutoDyn: For the purpose of signal visualization Technikgruppe has developed a tool called AutoDyn (see Figure 1) which supports the decision-making process. AutoDyn allows visualization of signal values over time in order to analyse correlations and other dependencies between signals. It supports the expert end user in their decision-making processes, offering correct visualization of different situations. Inappropriate or incorrect visualisation may have severe consequences as the visualisation has a major impact of the decision-making process in operating the whole power plant.

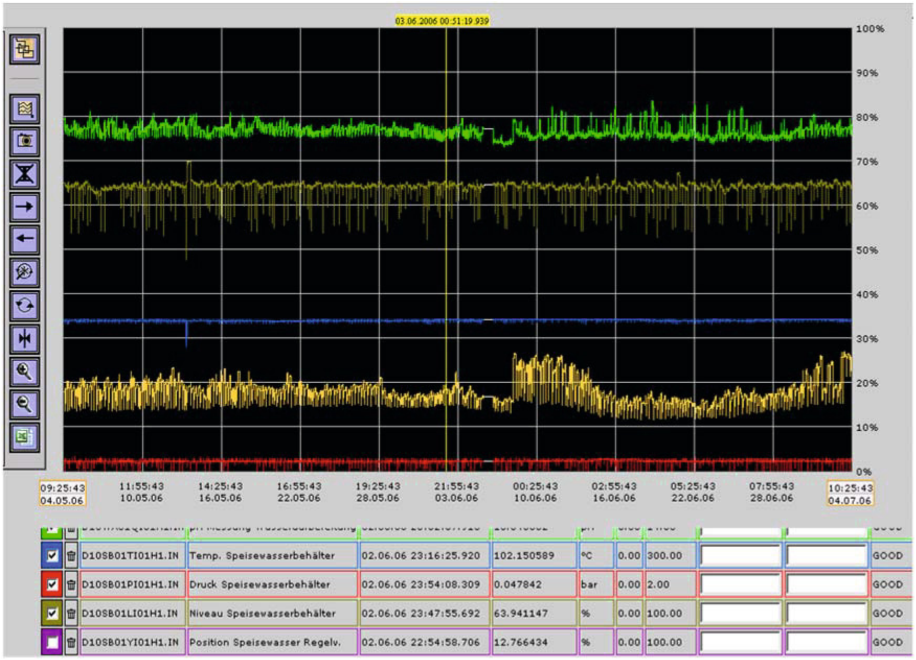

Fig. 1. The user interface of the decision support system, called AutoDyn

2 Related Work

In the paper *"How to measure cognitive complexity in HCI"* Rauterberg (1996) transferred the broad definition of cognitive complexity into the context of HCI. Accordingly, "the complexity of the user's mental model of the dialog system is given by the number of known dialog contexts ("constructs") on one hand and by the number of known dialog operations ("relationships") on the other hand" [4].

A GUI with a defined level of complexity will be perceived differently by expert and novice users. There is interesting research on this topic by Coskun & Grabowski (2004) [5]: They compared complexity of original and improved versions of Navigation and Piloting Expert Systems (NPES). These are operational decision support systems (DSS) which provide intelligent decision support to Chevron oil tanker ship's masters, mates and pilots navigating the restricted waters of San Francisco Bay. Three users took part in the evaluation process. Two of them were experts in navigation and piloting with both theoretical and practical knowledge. The third participant was a senior student with theoretical knowledge but less practical experience. Users had to complete the same task scenarios in both NPES-1 and NPES-2 and afterwards to fill out a questionnaire and to take part in an interview.

The result showed that the users preferred the original version of NPES-1, which used a raster image digital chart, even though it provided less detailed information and was visually more complex. NPES-2 used a fully vectorized electronic chart, reduced visual complexity and offered more functionality. However, in order to get information in NPES-2 a user had to click several times and this was a disturbing factor for the participants. They preferred to have all available data short and concise at one glance without clicking around. The officers also felt annoyed using NPES-2, as its intelligent implementation analyzed the data and offered a ready solution (users only had to push the "Accept" button) without requiring users to think. Expert users would like NPES to play the role of helper and not a director telling them, what to do. The senior student was more loyal to NPES-2, as the lack of practical experience made him use programs suggestions more often.

This study demonstrated that user interface complexity can be perceived differently by users of different experience and knowledge levels. While novice or less experienced users need to be lead through the tasks, get recommendations and ready made decisions to accept, experts prefer to have the program under their control and be free to analyze and decide what to do themselves.

Afacan & Erbug (2009) [6] demonstrated how heuristic evaluation as a usability evaluation method (see [7] for an overview) can contribute to building design practice to conform to universal design principles [8]. They took seven universal design principles as a set of heuristics and applied an iterative sequence of heuristic evaluation in a shopping mall, aiming to achieve an efficient evaluation process. The evaluation was composed of three consecutive sessions: Five evaluators from different professions were interviewed regarding the construction drawings in terms of universal design principles; each evaluator was asked to perform predefined task scenarios; finally, the evaluators were asked to reanalyse the construction drawings. The results showed that heuristic evaluation can integrate universal usability into building design practice.

The most valuable result of the work of Afacan & Erbug is, that out of 53 usability problems identified by all five evaluators, 28 were identified as major problems and had already been identified during pre-interview on construction drawings.

This research has demonstrated that usability pre-evaluation of safety-critical systems allows detection and a significant decrease in the amount of serious usability problems, which might generate further major and minor problems.

Safety-critical systems are not always effectively testable in artificial conditions [9]. An everyday example is the usability of driver information systems used in cars [10]: a new, previously unknown road ahead is unpredictable and demands full attention for safe driving. Funk & Hamacher (2008) [11] demonstrated the successful application of automatic usability evaluation instruments in such car driver IS: They introduced a two-step approach; where an observation component is embedded into the driver IS and collects data during habitual use. This component is a D'PUIS framework (www.softreliability.org/dpuis): The collected data is put into a rule-based expert system REVISER (www.hamacher.eu/reviser), which reasons over the aggregated data and evaluates it according to integrated guidelines.

During automated testing of the driver information systems AUDI MMI and BMW iDrive (refer also to the work of [12]). REVISER identified numerous faults in both of them. The tool has also provided proposals and hints for improvement. Empirical evaluation confirmed almost all automatically identified faults. This experiment demonstrated that the use of automatic usability evaluation methods can speed up usability testing and make this process more valuable.

3 Methods and Materials

The branch of this research is quite specific and so is the configuration tool. It is fairly hard to use, evaluate or test without special background knowledge and training. This narrows the choice of possible test users and evaluators. Consequently, for this project only three persons could be involved as domain experts. Only they are able to carry out complex task, for example the process of go-live configurations.

Applied usability engineering and evaluation methods should take this limitation into account. For routine configurations (simple tasks) theoretically there is no limitation on possible attendances. As a policy, Technikgruppe allows participation by its own employees only.

In order to retrieve as much valuable information as possible from the three domain experts, the interviewing was chosen as a main inspection method, which was applied for the first phase of analysis and design as well as during iterative refinement of the new GUI. The general methodology for the research looks as follows:

- Define core and complementary functionality
- Define requirements
- Define problems of the TGtool
- Produce interface ideas and create their low-fidelity prototypes
- Pick the best idea and iteratively refine it
- Implement the refined idea into a high-fidelity prototype
- Process thinking aloud tests and improve the prototype
- Evaluate the high-fidelity prototype performance compared to TGtool performance

The first three points were cleared during interview sessions with the domain experts. Afterwards, based on the results of the interviews several GUI ideas were produced, low-fidelity prototypes were sketched for promising ideas. The prototypes were discussed and evaluated during the second interview session and the best one was chosen and iteratively developed into the high-fidelity prototype. The development process had an iterative character, whereas domain experts provided their feedback, after completion of each development iteration.

Finally, several simple tasks (in our context "simple" means routine configurations) were created and a thinking aloud test was performed by the end users of the tool. Thus, usability problems, which were not detected during the interviews and the development process, were found. The thinking aloud test results enabled a final optimization of the tool.

Finally, in order to evaluate the result most adequately both an objective and a subjective method were applied.

Given routine tasks, the amount of mouse clicks necessary for completion of each of the tasks in both configuration tools was calculated.

Heuristic evaluation was completed for both the old and the new GUIs, following some guidelines from the medical domain – which is a similar safety-critical application area [13], [14], [15], [16].

Thinking Aloud Test

Out of the broad spectrum of Usability Engineering Methods one particular method is very beneficial: Thinking-aloud (aka TA). This „think out loud“ originally derived from problem solving research [17] and allows to get insight into mental processes of people [18], [19].

For our TA tests we prepared six tasks (see Table 1) as well as a test environment. The videos were recorded with a Canon Exilim 12.1 mega pixels camera and Debut Video Capture Software (http://debut-video-capture-software.softonic.de) was used for registering user’s activity on the screen.

Table 1. Tasks for TA

Task #	Task description
Task 1	Look around.
Task 2	Create a new signal.
Task 3	Edit signal’s properties.
Task 4	Create a new signal group.
Task 5	Add this signal to the group.
Task 6	Delete both the group and the signal.

Table 2. Hard- and software environment

Environment	Details
Room	Standard work place (individual for each test)
Hardware	Intel(R) Core(TM) Duo CPU T2450, 2 GB RAM
OS	Windows 7 Professional.
Monitor Colours	32 bit
Monitor Resolution	1280 x 800
Monitor Size	13" TFT

In order to achieve more objectivity in TA results “fresh” test users were required, who had not seen the prototype before. Also due to the Technikgruppe’s policy, only real end users of the tool could take part in the testing process, in particular the Technikgruppe employees, who are working as:

- PLS (in German: ProcessLeitSystem) technicians,
- commissioning (in German: Inbetriebnahme) technicians,
- measurement (in German: Mess- und Regel-) technicians,
- plant operators (in German: Anlagenbediener).

These requirements decreased the amount of possible test users to three persons. Although for any usability test it is important to have a higher amount of test users, we were limited to these three persons. In order to get more out of this limitation, we decided to apply an *iterative TA,* so that after each TA session all critical issues were first analyzed and implemented, and only then the next TA session took place. This approach allowed us to concentrate on the finding of more usability problems instead of discovering the same problems again and again in each TA session.

The average profile of a test user according to a background questionnaire was a 39 years old male with education in electrical engineering, working as a commissioning technician with 26 years of PC experience, working around 40 hours per week with PC, predominantly on the Windows operating system.

This average user needs to process different routine configurations in order to update decision supporting AutoDyn quite often in his professional life, however, in most cases he delegates this to the developers of the tool as he does not feel confident working with it and is afraid to make a mistake.

In the context of our research these three test users (one pilot and two test users) had to accomplish prepared tasks using the configuration tool prototype.

Process Description

At the beginning of the test the user went through the orientation script, filled out a background questionnaire and signed the non-disclosure and consent form. Afterwards he trained to think aloud painting a house in Paint (Windows application). On completion the test person opened the prototype and performed six prepared tasks. Finally the user was interviewed and in the end he filled out the feedback questionnaire.

The test was performed iteratively. It means that the feedback of each user had been analyzed, GUI "debugging" was done and then the next user tested the prototype in a thinking aloud session. This approach (eliminating small and easy-to-fix problems after each TA session) was chosen in order to reduce the amount of already known minor problems and concentrate the users' attention on discovering more severe ones.

After the TA completion all the minor and major findings were analyzed and implemented during final optimization of the configuration tool.

4 Experimental Results and Lessons Learned

During the interviews, the participants ran into problems and had difficulties with obvious tasks of the tool. The request to demonstrate the whole functionality was too complexfor two out of three. Many problems of the current GUI were revealed very fast. The disadvantages of the TGtool were formulated by the interviewees as follows:

Table 3. Pros and Contras of TGtool

No.	Disadvantages	Details
1	No distinction between two configuration types	GUI does not distinct between two types of configuration
2	Inconsistent titles	Different terms used for the same object, some titles do not correspond with their functions
3	Inconsistent links	Different categories offer the same web pages, different pages have the same functionality, some pages are broken
4	Minimalistic functionality	Often at one page users can perform only one action
5	Status of the system is not visible	Users do not see if an operation has been completed and whether it has been completed successfully
6	No progress status on time-consuming operations	The text “This operation may take about several minutes” is usually provided
7	Tool does nothing when it should	Users have to press some 'secret' button, obvious for the developer, but not for the users. Documentation does not point it out
8	Unnatural to add elements from right to left	Users are used to adding elements into the list from left to right
9	Absence of warning messages	Users would not be aware of a mistake until it affects the system
10	Absence of error messages	If users get one, they would not be able to diagnose it
11	No user management	No information on who has done changes. No opportunity to manage users, passwords and rights
12	No language support	Only German is supported
13	Poor documentation	The available documentation is not adequate

Outcome of the Thinking Aloud Test

In the first session with the pilot user (TP1) the highest amount of minor problems has been discovered. Already during the first look-around task five minor problems have been detected. For instance:

- no last access data was displayed,
- current language has been set to Russian, while the interface was in English
- the maintenance user has been allowed to create users and associate access rights

Other usability "bugs" have been discovered during the test, e.g. buttons not working, absence of a sandglass, when processing needs over one second, no explicit description of a filter field etc.

All these issues have been fixed before the next session. Interestingly, the second test user found more consistency problems in the titles and dock widget's usage. As well as the pilot user he also complained about the tool's speed and discovered more operations where the sand clock was missing. The absence of a sand clock seemed to be an important issue as some tasks took over five seconds for processing and the user was confused since no feedback was given.

In the third session some more usability problems were discovered, like the password field at the login page was to the right of the user field and not under it, as is commonly done. Several minor usability 'bugs" were detected as well.

In general users have successfully accomplished all the tasks without any training or help, relying only on their cognition and domain knowledge. The diagram below shows time curves, minutes per task for each test person TP; the task number 1 has not been taken in account as a looking around task cannot have a concrete definition of success. Thus some users took a long time (about 3 min) to explore the tool, others clicked over two menu options rapidly (30 sec) and said they are done;on the horizontal axis the task number and its complexity grade and on the vertical axis the time in minutes is shown.

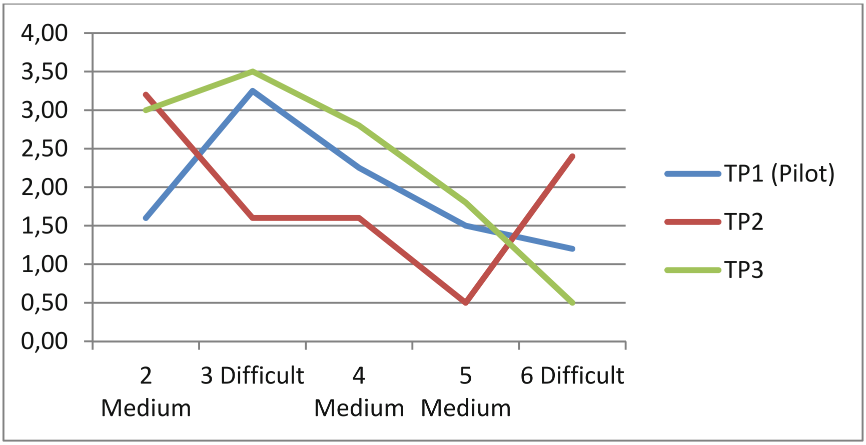

Fig. 2. Minutes per task

Though there is some deviation due to the users' individual cognition, the general trend is clear – after the completion of the first two tasks the time required for the tasks with the same complexity level (medium and difficult) decreases. The fact that the third TP required more time than first and second, can be hardly evaluated due to the individuality of users' cognition process and too few TPs. A higher amount of test users would provide more representative results.

In TA sessions after the completion of all the tasks the users have had to fill out a feedback questionnaire. Figure 3 outlines the results.

The graph demonstrates the result of the iterative approach, whereas the TP1 has graded the prototype with the lowest points (average grade 3.1). The TP2's average grade is notably higher (4.5). And the last user TP3 demonstrates the highest satisfaction with the average of 4.8, however, only a small growth of 0.3 point can be observed as most problems have already been fixed after the first session.

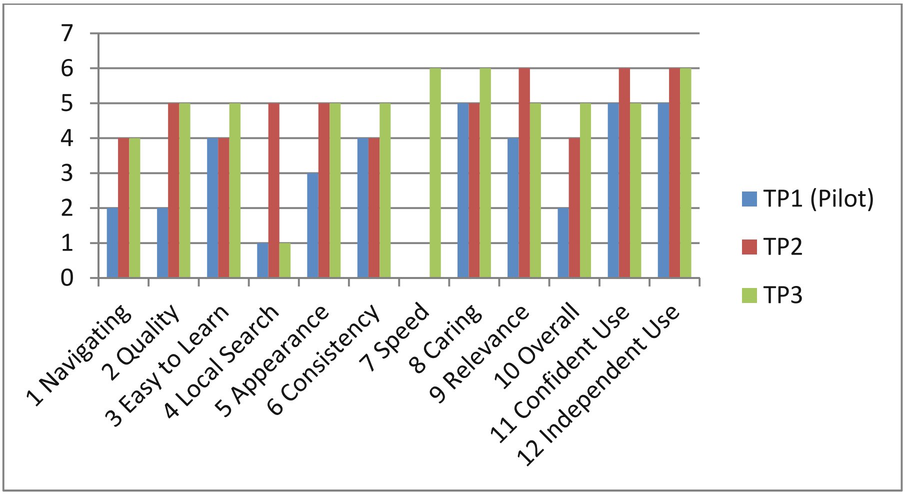

Fig. 3. Feedback questionnaire result

During the TA sessions the TP1 discovered a usability problem and TP3 a computational problem at a signal filter, whereas the TP2 did not. This explains their evaluation of the fourth feedback aspect "Local Search".

In order to eliminate the speed problem another hardware test environment with 4 GB RAM (see Table 4) was used for the last test. A monitor size of a test computer became bigger, which influenced slightly the test results, as the TP3 required more time for each task due to smaller items.

Table 4. Hardware environment for the last TA

Environment	Details
Hardware	Intel(R) Core(TM) Duo CPU T9600, 4 GB RAM
Monitor Colours	64 bit
Monitor Resolution	1900 x 1200
Monitor Size	15,6" TFT

This explains why TP1 and TP2 have graded the seventh feedback aspect "Speed" with 0 points while TP3 has given almost the maximum - 6 points.

In general, the results of the feedback questionnaire follow a positive trend, as in evaluation of 9 aspects out of 12 every following TP gave it equal or more points.

Result of the Mouse Click Measurement

In order to see if users really require less clicks in order to successfully complete their task a mouse click measurement has been performed. The same tasks have been used in this experiment as in the TA. In Figure 4 it can be seen that at any task processing less clicks have to be done in prototype.

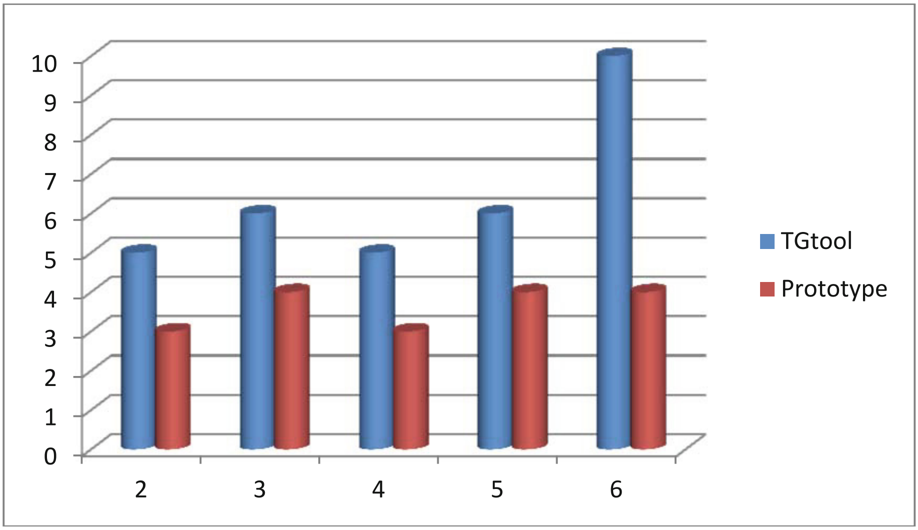

Fig. 4. Minimal amount of clicks per task

It is remarkable that in the tasks 2 to 5 the prototype needs exactly 2 clicks less than the TGtool.

The reason is found in the nested menus of TGtool. As it has been demonstrated in the introduction of this work, the TGtool offers 6 menus to choose at the access point. Selecting one menu the user gets to the next page where again from 4 to 7 menus can be chosen. This is an inefficient clicking process where the user does not perform the actual task but navigates to the page, where the task can be performed. And this makes, according to this research, two more clicks than is actually necessary.

The final task where the user deletes both the signal group and the signal shows a difference of 6 clicks. As TGtool provides an individual page for signal configuration and another individual page for signal group configuration, each of them has to be navigated to in order to complete the task. Thus the amount of "navigation" clicks doubles and only 6 out of 10 clicks are efficient.

Outcome of the Heuristic Evaluation

Heuristic evaluation has been performed by two heuristic experts with domain knowledge. They have evaluated the TGtool and the prototype on the basis of ten heuristics from Nielsen (1993) [20]. The five-point rating scale from Pierotti [21], has been applied. In this scale, 1 point refers to cosmetic problems, and 5 to catastrophic ones. In the net-diagram (Figure 5) the results can be seen.

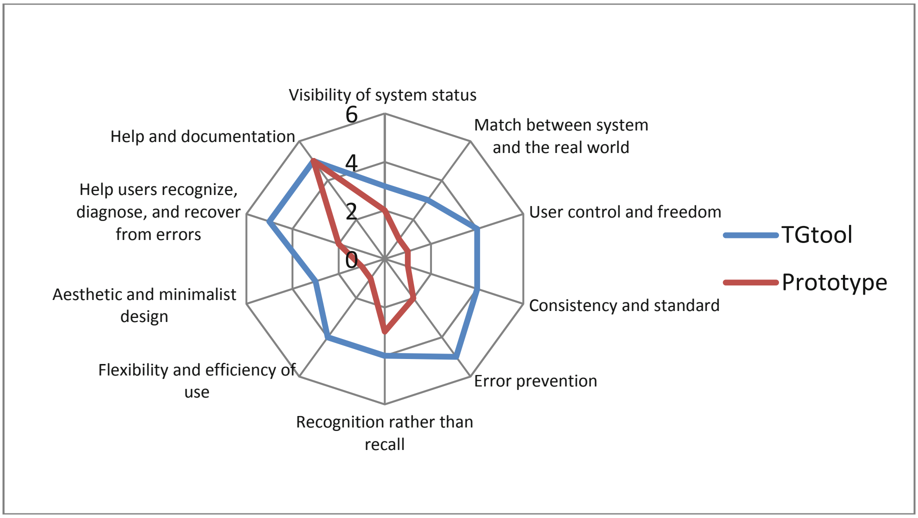

Fig. 5. Heuristic evaluation results

The user-centered prototype is not yet perfect; however, its heuristic evaluation has demonstrated a strong improvement in comparison to TGtool. It can be seen clearly on the network diagram, that there is no aspect where the prototype would be evaluated worse than the TGtool.

5 Conclusion and Future Research

In this work a usability engineering approach was applied to the development of configuration tool prototype, in order to reduce complexity, or more specifically, to adapt visualization of a complex system to the end user’s cognitive map. To achieve this, end users were actively integrated into the whole development process from the very beginning to the very end. A big amount of usability problems was detected during the TA session, even though all the decisions upon GUI design were made by the end users during the interview sessions.

The research came up with interesting findings as TA was performed. In the TA test “fresh” end users participated, who did not see a prototype before and did not take part in interview sessions during development. These test persons detected 20 usability problems. However, all of the problems were classified as minor problems which were easy to fix.

This finding confirms that the iterative user-centered GUI design and development minimizes the amount of major problems, because it iteratively adapts the GUI to the end users' cognitive map. All of the TA tasks were successfully completed by all of the users with the average trend of time required for a task decreasing. Therefore, the users succeeded to learn the tool on their own within a short time and all of them stated that they felt confident using the tool. Furthermore, they all shared the opinion that the prototype was hard to use only at the beginning but, as soon as the users got the hang of it, it became easy.

Finally, objective and subjective inspections of the TGtool and the prototype confirmed the effectiveness of usability engineering and clearly pointed out the faults of the prototype to be improved, like the absence of help and documentation and weak recognition support. Mouse click measurement demonstrated the efficiency and minimalistic design of the prototype. The HE fully confirmed that result and TA participants were satisfied with a clear and consistent layout design.

Although TA was performed during the last phase of development and HE afterwards, their results correlate to a great extent. Almost all the heuristics from Nielsen (1993) were commented by the test users in a positive or a negative way. The heuristics like *visibility of system status*, *match between system and the real world*, *consistency* and *recognition* were heavily criticized in the TA sessions and thus have been improved in the last development phase before HE. The *error prevention* aspect as well as the *aesthetic and minimalist design* and *help users recognize, diagnose, and recover from errors* were evaluated positively. The other heuristics for *user control*, *flexibility* and *help and documentation* were not mentioned by the TPs. This basically implies that despite a significant intersection, HE uses a wider range of aspects which allows evaluating a GUI more thoroughly and objectively, whereas TA concentrates on the issues which have caught the users' attention, which is subjective. In this work the knowledge of visual complexity and its influence on the users' perception and cognition has been applied in development of safety-critical software. The aim was to reduce complexity of the configuration tool, more specifically, to design complexity of a tool in accordance to the users' cognitive map. For this purpose iterative usability engineering method was used, so that decisions on GUI design were made by the end users of the GUI. This approach ensured that structure and complexity of the prototype would match users' cognitive map structure. As the main development process was completed, thinking aloud test was applied for the refinement of the configuration tool prototype. Usability evaluation of both TGtool and the developed prototype was performed by means of mouse click measurement test and heuristics evaluation.

Neither heuristics evaluation test nor mouse click measurement could refute the hypothesis that an end user-centered cognitive map structure helps to deal with complexity without excessive training. On the contrary, the result of heuristics evaluation demonstrated clearly that there was no usability aspect where the user-centered prototype would be inferior to system-oriented TGtool. The mouse click measurement determined the prototype as more efficient than TGtool. Moreover, the thinking aloud test showed that the users could successfully deal with the complexity of the prototype without excessive training. This evidence supports the hypothesis of this works.

There are some limitations of this research.

The TA and the mouse click measurement were performed on the basis of the same six tasks. Thus the space of these tasks restricts the validity of their test results.

The participants of TA were the Technikgruppe employees, who had at least some minimal knowledge of the system behind the software. Thus application of new configuration tool might require more training for the technicians from customers' side.

The other limitation is the absence of the stress factor. In real conditions of a safety-critical system users (commissioning technicians, plant operators) are stressed while working with the configuration tool, because they are aware of consequences which can be caused by wrong action. This stress factor influences users' emotional state, thus their perception and cognition.

Acknowledgements. We are grateful for the industrial cooperation with the Technikgruppe, especially to Matthias Lukic. The work presented herein has been partially carried out within the competence network Softnet Austria II (www.soft-net.at, COMET K-Projekt) and funded by the Austrian Federal Ministry of Economy, Family and Youth (bmwfj), the province of Styria, the Steirische Wirtschaftsförderungsgesellschaft mbH (SFG), and the City of Vienna, in support of the Center for Innovation &Technology (ZIT).

References

1. Thomas, J.C., Richards, J.T.: Achieving psychological simplicity: Measures and methods to reduce cognitive complexity. In: Sears, A., Jacko, J.A. (eds.) The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, 2nd edn., pp. 489–508 (2008)
2. Ziefle, M.: The influence of user expertise and phone complexity on performance, ease of use and learnability of different mobile phones. Behaviour & Information Technology 21(5), 303–311 (2002)
3. Liu, Z., Stasko, J.T.: Mental Models, Visual Reasoning and Interaction in Information Visualization: A Top-down Perspective. IEEE Transactions on Visualization and Computer Graphics 16(6), 999–1008 (2010)
4. Rauterberg, M.: How to measure cognitive complexity in human-computer interaction. Cybernetics and Systems Research, 815–820 (1996)
5. Coskun, E., Grabowski, M.: Impacts of User Interface Complexity on User Acceptance in Safety-Critical Systems. In: Proceedings of the 10th American Conference on Information Systems, AMCIS 2004, pp. 3433–3443 (2004)
6. Afacan, Y., Erbug, C.: An interdisciplinary heuristic evaluation method for universal building design. Applied Ergonomics 40(4), 731–744 (2009)
7. Holzinger, A.: Usability engineering methods for software developers. Communications of the ACM 48(1), 71–74 (2005)
8. Shneiderman, B.: Universal usability. Communications of the ACM 43(5), 84–91 (2000)
9. Bastide, R., Navarre, D., Palanque, P.: A tool-supported design framework for safety critical interactive systems. Interacting with Computers 15(3), 309–328 (2003)

10. Heimgärtner, R., Holzinger, A.: Towards Cross-Cultural Adaptive Driver Navigation Systems. In: Holzinger, A., Weidmann, K.-H. (eds.) Empowering Software Quality: How Can Usability Engineering Reach These Goals?, pp. 53–67. Austrian Computer Society (2005)
11. Funk, M., Hamacher, N.: Concept of Automatic Usability Evaluation of Safety-Critical Interactive Systems in the Field (Konzept der automatischen Bewertung der Gebrauchstauglichkeit sicherheitskritischer Systeme). i-com 7(1), 18–23 (2008)
12. Holzinger, A., Waclick, O., Kappe, F., Lenhart, S., Orasche, G., Peischl, B.: Rapid Prototyping on the example of Software Development in the automotive industry: The Importance of their Provision for Software Projects at the Correct Time ICE-B 2011, pp. 57–61. SciTePress, INSTICC, Setubal (2011)
13. Holzinger, A.: Application of Rapid Prototyping to the User Interface Development for a Virtual Medical Campus. IEEE Software 21(1), 92–99 (2004)
14. Holzinger, A., Errath, M.: Mobile computer Web-application design in medicine: some research based guidelines. Universal Access in the Information Society International Journal 6(1), 31–41 (2007)
15. Holzinger, A., Stickel, C., Fassold, M., Ebner, M.: Seeing the System through the End Users' Eyes: Shadow Expert Technique for Evaluating the Consistency of a Learning Management System. In: Holzinger, A., Miesenberger, K. (eds.) USAB 2009. LNCS, vol. 5889, pp. 178–192. Springer, Heidelberg (2009)
16. Holzinger, A., Kosec, P., Schwantzer, G., Debevc, M., Hofmann-Wellenhof, R., Frühauf, J.: Design and Development of a Mobile Computer Application to Reengineer Workflows in the Hospital and the Methodology to evaluate its Effectiveness. Journal of Biomedical Informatics 44(6), 968–977 (2011)
17. Duncker, K.: On problem-solving. In: Dashiell, J.F. (ed.) Psychological Monographs of the American Psychologoical Association, APA, vol. 58, pp. 1–114 (1945)
18. Nisbett, R.E., Wilson, T.D.: Telling More Than We Can Know: Verbal Reports on Mental Processes. Psychological Review 84(3), 231–259 (1977)
19. Holzinger, A., Leitner, H.: Lessons from Real-Life Usability Engineering in Hospital: From Software Usability to Total Workplace Usability. In: Holzinger, A., Weidmann, K.-H. (eds.) Empowering Software Quality: How Can Usability Engineering Reach these Goals?, pp. 153–160. Austrian Computer Society (2005)
20. Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Francisco (1993)
21. Pierotti, D.: Usability Techniques: Heuristic Evaluation Activities, `http://www.stcsig.org/usability/topics/articles/he-activities.html` (last access June 01, 2012)