Master's Thesis

Towards user-centered intelligent visualization of complex processes and critical pathways in safety critical systems

Evgenia Popova

Institute for Information Systems and Computer Media (IICM)
Graz University of Technology

Supervisor:
Assoc. Prof. Andreas Holzinger, PhD, MSc, MPh, BEng, CEng, DipEd, MBCS

Graz, May 2012

<This page is intentionally left blank>

Masterarbeit

(Diese Arbeit ist in englischer Sprache verfasst)

Über benutzerzentrierte, intelligente Visualisierung komplexer Prozesse und kritischer Pfade in sicherheitskritischen Systeme

Evgenia Popova

Institut für Informationssysteme und Computer Medien (IICM)
Technische Universität Graz

Betreuer:
Univ.-Doz. Ing. Mag. Mag. Dr. Andreas Holzinger

Graz, Mai 2012

<This page is intentionally left blank>

For my dad,

in honour of his 50th anniversary.

Для папы,

в честь пятидесятилетия.

Abstract

Controlling systems used at power plants offer a graphical visualization of the physical parts of the site but they rarely provide a sufficient visualization of the signal data. Depending on the size and complexity of the facility it can have 1 000, 10 000 or more measuring points. Each of them continuously sends data value updates to the controlling system (once in 20 ms or less often). This huge load of data can be analyzed only if appropriately visualized.

The company Technikgruppe Mess-, Steuer- und Regeltechnik GmbH has a tool called AutoDyn which allows processing and visualizing signal data efficiently. AutoDyn is a visualization tool supporting the decision-making process.

TGtool is an important interface between controlling system and AutoDyn. It is a configuration tool which is responsible for correct communication between a controlling system and decision-supporting AutoDyn. Users constantly interact with TGtool adding signals to AutoDyn, changing properties. TGtool has been developed using a system-oriented approach and therefore it is difficult to understand and apply for end users. It becomes very inconvenient as 36 clicks are required just for a routine configuration.

The configuration tool, which is hard to use, hard to understand and hard to learn is of extreme importance in its safety-critical area of application. Wrong configuration can lead to wrong signal data visualization, which provides a basis for wrong decision-making of plant personnel. An unintentional mistake, which would be of no consequence in everyday life, could have very dramatic consequences in the reality of power plants. Consequently, an error may not be underestimated.

The configuration tool is required for processing small modifications by the non-expert end user in order to adapt the decision-supporting AutoDyn to the dynamically changing controlling system.

The aim of this work is to apply user-centered approaches in order to redesign a configuration tool, so that a non-expert end user can make necessary configurations without excessive training.

This research follows the hypothesis that an user-centered cognitive map structure helps to deal with complexity without excessive training.

Keywords

Complexity, data visualization, process, user-centered, usability engineering, safety-critical systems, industrial design, heuristic evaluation, thinking aloud

ÖSTAT-Topics (maximum 4)

1140	40%	1138	30 %	1108	20 %	1150	10 %

ACM Classification

C.3, D.2.2, D.2.8, D.2.10, H.4, H.5, J.7

Zusammenfassung

Leitsysteme, die in Kraftwerken eingesetzt werden, bieten eine graphische Darstellung der physikalischen Komponenten. Allerdings liefern sie selten eine gute Visualisierung der Signaldaten. Abhängig von der Größe und der Komplexität der Anlage kann es sich um 1 000, 10 000 oder mehr Messstellen handeln. Sie schicken kontinuierlich (bis zu 20 ms Takt) Signalwerte an das Leitsystem. Diese enorme Menge an Daten kann nur durch eine optimale Visualisierung analysiert werden.

Die Firma Technikgruppe Mess-, Steuer- und Regeltechnik GmbH hat AutoDyn entwickelt, ein Program, das es erlaubt, Signaldaten effizient zu verarbeiten und zu visualisieren. AutoDyn ist ein Visualisierungsprogramm, das bei Anlagen- bzw. Sicherheits- relevanten Entscheidungen als Unterstützung verwendet wird.

TGtool ist eine sehr wichtige Schnittstelle zwischen AutoDyn und dem Leitsystem, es ist ein Konfigurationstool, das für eine sichere Verbindung zwischen Leitsystem und AutoDyn verantwortlich ist. TGtool wird regelmäßig verwendet um Signale hinzuzufügen oder Signaleigenschaften zu bearbeiten. Da das TGtool systemorientiert entwickelt wurde, ist es sehr schwer für die Endbenutzerin oder den Endbenutzer verständlich. Darüberhinaus ist es sehr ineffizient, da für eine Routinekonfiguration 36 Klicks erforderlich sind.

Das Konfigurationstool, das folglich schwierig zu verstehen, kompliziert zu erlernen und umständlich zu bedienen ist, wird in einer sicherheitskritischen Branche in einem sicherheitsrelevanten Bereich eingesetzt. Falsche Konfiguration führt zur falschen Signalvisualisierung, was wiederum zu falschen Entscheidungen durch das Anlagenpersonal führen kann. Ein kleiner Fehler, der im normalen Leben bedeutungslos wäre, kann in einer Anlage sehr dramatische Folgen haben. Deswegen dürfen Fehler nicht unterschätzt werden.

Das Konfigurationstool wird von non-expert Endbenutzerinnen und Endbenutzern für kleine Änderungen verwendet, um das entscheidungsunterstützende AutoDyn an das dynamische Leitsystem optimal anzupassen.

Das Ziel dieser Arbeit ist es, ein benutzerfreundliches und übersichtliches Konfigurationstool zu entwickeln, sodass eine non-expert Benutzerin oder ein non-expert Benutzer Konfigurationen ohne exzessives Training durchführen kann.

Diese Arbeit folgt der zentralen Hypothese, dass ein user-zentriertes kognitives Mapping hilft mit Komplexität umzugehen, ohne exzessives Training zu erfordern.

Schlüsselwörter

Komplexität, Datenvisualisierung, Prozess, benutzerorientiert, Usability Engineering, sicherheitskritische Systeme, industrielle Visualisierung, Heuristische Auswertung, Thinking aloud

ÖSTAT-Fachgebiete (maximum 4)

1140	40 %	1138	30 %	1108	20 %	1150	10 %

ACM Klassifikation

C.3, D.2.2, D.2.8, D.2.10, H.4, H.5, J.7

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, 24th May 2012 ………………………………………………

Evgenia, Popova

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, 24.Mai 2012 ………………………………………………

Evgenia, Popova

Danksagung / Acknowledgments

Я выражаю огромную благодарность моим родителям Попову Вадиму Геннадиевичу и Поповой Галине Витальевне за их поддержку и веру в меня.

Ich danke Bernd Zechner für die moralische Unterstützung und seine grenzenlose Geduld.

I offer my gratitude to my supervisor Univ.-Doz. Ing. Mag. Mag. Dr. Andreas Holzinger for his expertise, commitment and support.

Vielen Dank an Walter Martinjak für seine selbstlose unentbehrliche technische und moralische Unterstützung. Großen Dank auch an Marion Starzacher, die mich zum Masterstudium an der TU Graz motiviert und mich dabei unterstützt hat.

Abbreviations

AIT	Algorithmic Information Theory
CMC	Control IT Melody Controller
CSV	Comma-Separated Values
DB	Database
DNS	Domain Name System
DSS	Decision Support System
FHE	Focused Heuristic Evaluation
GUI	Graphical User Interface
HCI	Human–Computer Interaction
HE	Heuristic Evaluation
IS	Information System
JPEG	Joint Photographic Experts Group
MEDSI	Mobile Emergency Department Software Iconic
NPES	Navigation and Piloting Expert System
OS	Operating System
PC	Personal Computer
PLS	Prozessleitsystem
PVT	Passive Viewing Task
SCS	Safety-Critical System
SET	Shadow Expert Technique
TP	Test Person
TA	Thinking Aloud Test
UCD	User-Centered Development
UX	User Experience
VST	Visual Search Task
ZOV	Zentrales Objekt Verzeichnis

Table of Contents

1. Introduction and Motivation for Research

In this work it will be dealt with intelligent user-centered data visualization, especially applied in safety critical systems. Advantages of user-centered approaches will be demonstrated at the example of a configuration tool called TGtool developed by Technikgruppe Mess-, Steuer- und Regelungstechnik GmbH. In order to make clear what this tool is capable of and what it is responsible for, two other systems are described below.

Controlling system. Technikgruppe is working in the power plants branch. Each power plant has its own controlling system produced by Siemens, ABB or OPC Foundation. A controlling system (Figure 1) is an information system (IS) for the whole plant. It keeps track of all the measuring points, their settings and data. Each measuring point sends regularly its measurement (signal value) to the controlling system. All this data is saved; however, there is no sufficient way to visualize signal data on the time axis.

Decision-supporting AutoDyn. For the purpose of signal visualization Technikgruppe has developed a tool called AutoDyn (see Figure 2) which supports the decision-making process. AutoDyn allows visualizing of signal values over time in order to analyze correlations and other dependencies between signals. It serves to support the decision-making process, offering correct visualization of different situations. AutoDyn cannot make a mistake as a mistake may lead to a wrong reaction. Wrong reactions in power plant reality can cost lives.

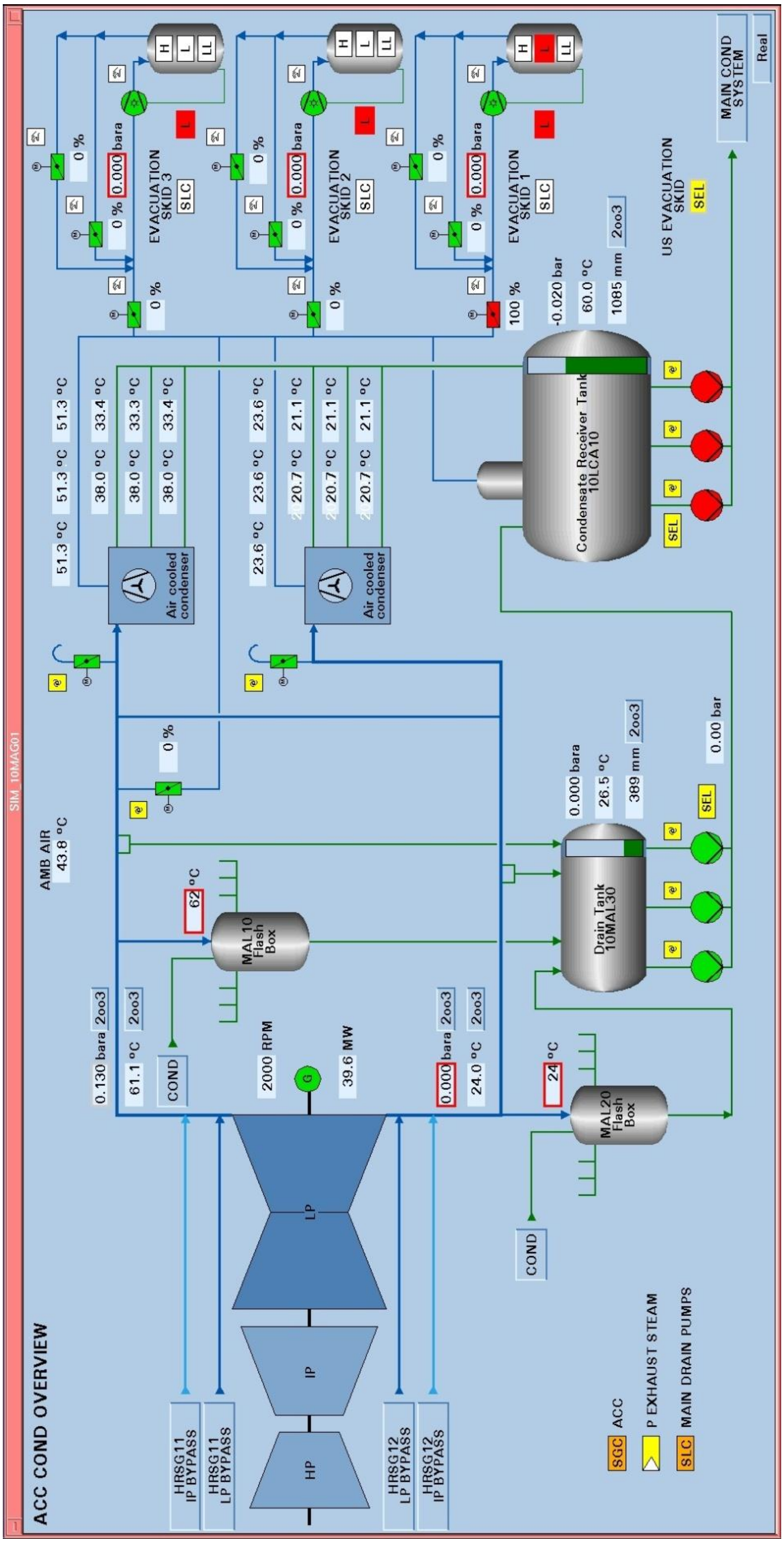


Figure 1 - Controlling system GUI

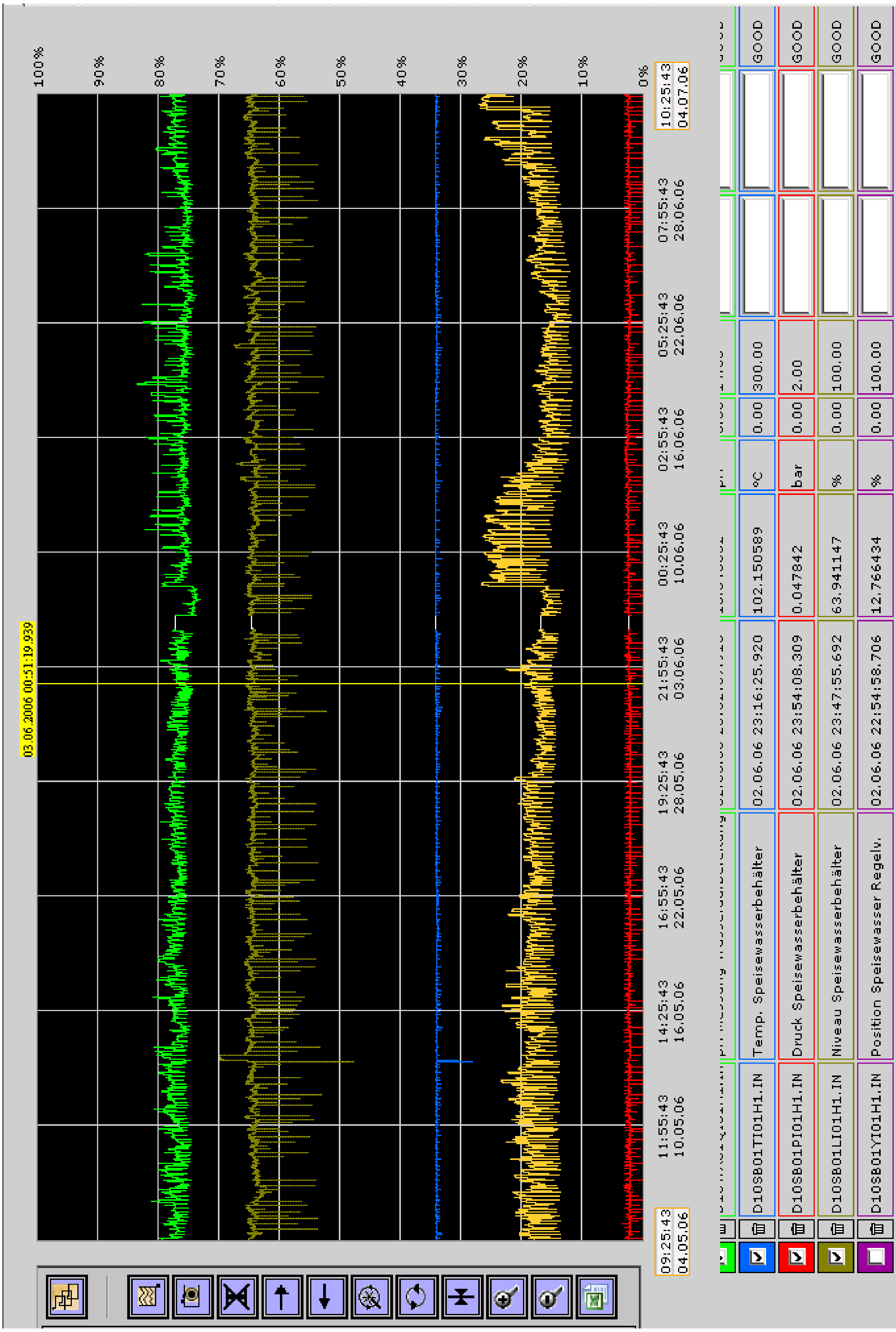


Figure 2 - AutoDyn

Configuration tool TGtool. A configuration tool (see Figure 3) is essential in order to ensure correct working of AutoDyn. This tool enables fast updates between two systems: the controlling system and AutoDyn.

Figure 3 - TGtool

Previously, Technikgruppe did not have any configuration tool, and all the settings were done directly in code and configuration files. Later it became obvious, that that approach was inefficient. A tool was required which would allow small configurations to be done quickly directly at the place. This shortage of a configuration tool became clear very suddenly and it was demanded urgently. TGtool was developed quite fast on the base of Webmin[1].

Webmin is a web-based interface for system administration for Unix operating system (OS). User accounts management, file sharing, Apache, DNS and more others can be administrated by using any modern web-browser and no manual edits of Unix configuration files are required.

As the configuration tool was not achieving high return-on-investment, almost no interest or attention had been paid to its general appearance. As time went by, more and

[1] http://www.webmin.com/

more functions were built into the tool, and most of them were developed under time pressure and urgency. Consequently, TGtool became extremely complex as it provided all vital functionality for good data exchange between controlling system and AutoDyn. Now this tool is very powerful, but unfortunately quite unusable by the non-expert end user.

The configuration tool is an extremely important link between controlling system and the decision supporting tool. Moreover, it is required for processing different modifications by the non-expert end user in order to update AutoDyn to the dynamically changing settings of the controlling system. Unfortunately, it has not been considered as an important safety-critical interface for a long time. Now its poor usability becomes a critical weak point in security issues and adds complexity to the workflow of commissioning technicians and plant operators.

This thesis follows the hypothesis that a user-centered cognitive map structure helps to deal with complexity without excessive training.

The aim of this work is to apply user-centered approaches in order to redesign the configuration tool, so that a non-expert end user can make necessary configurations without help.

1.1. Description of TGtool

TGtool is the currently used web-based configuration tool. Its main menu is shown on the Figure 4.

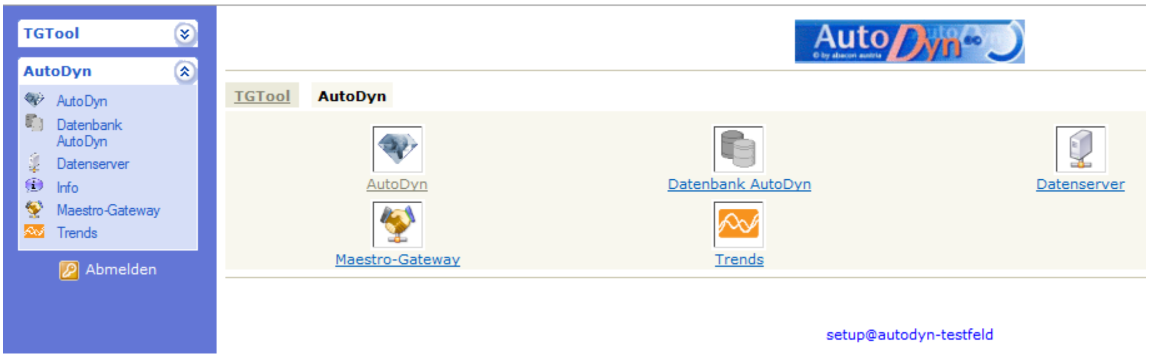


Figure 4 - TGtool main menu

Six hyperlinks are offered to proceed on the main page, the same navigation options are displayed on the navigation menu on the left:

- AutoDyn
- Datenbank AutoDyn
- Datenserver
- Info
- Maestro-Gateway
- Trends

From the first glance it becomes obvious that the GUI is system-centered. Proceeding with the first option "AutoDyn", a user will find himself at the following interface (Figure 5). For more information in the area of system- and user- centered design the papers of Norman & Draper (1986), Soloway et al (1994), Holzinger & Motschnig (2005) and Holzinger (2005) should be referred.

Figure 5 - TGtool main functions

Choosing the second option "Datenbank AutoDyn" following page will be called (Figure 6).

Figure 6 - TGtool database

It is easy to notice that three icons are just the same on both web pages, and their titles are the same. Moreover, they provide absolutely the same functionality. It may confuse a user, as he does not know sometimes, if this page was visited yet or not. In order to provide the whole structure of a current TGtool web application, a logical diagram is presented in Figure 7.

On the scheme the main access point is declared as “AutoDyn” and six categories are provided to select. Each provides from four up to seven functional web pages. The web pages filled with yellow are included into two categories and are unfortunately responsible for the users’ disorientation. These doubly-addressed web pages are sometimes differently labelled, but it is not reflected on the diagram for the sake of simplicity.

It is also easy to notice that some categories have a lot in common. For example, within the category “AutoDyn”, all six web pages are addressed twice. Moreover, in categories “Datenserver” and “Info” four out of seven web pages belong to some other category as well. Actually there are 23 unique functional web pages and almost half of them (9) belongs to two categories, thus there are 32 paths, which a user can follow. The web page filled with grey is out of date and should not be used anymore; however, it is still accessible for a user.

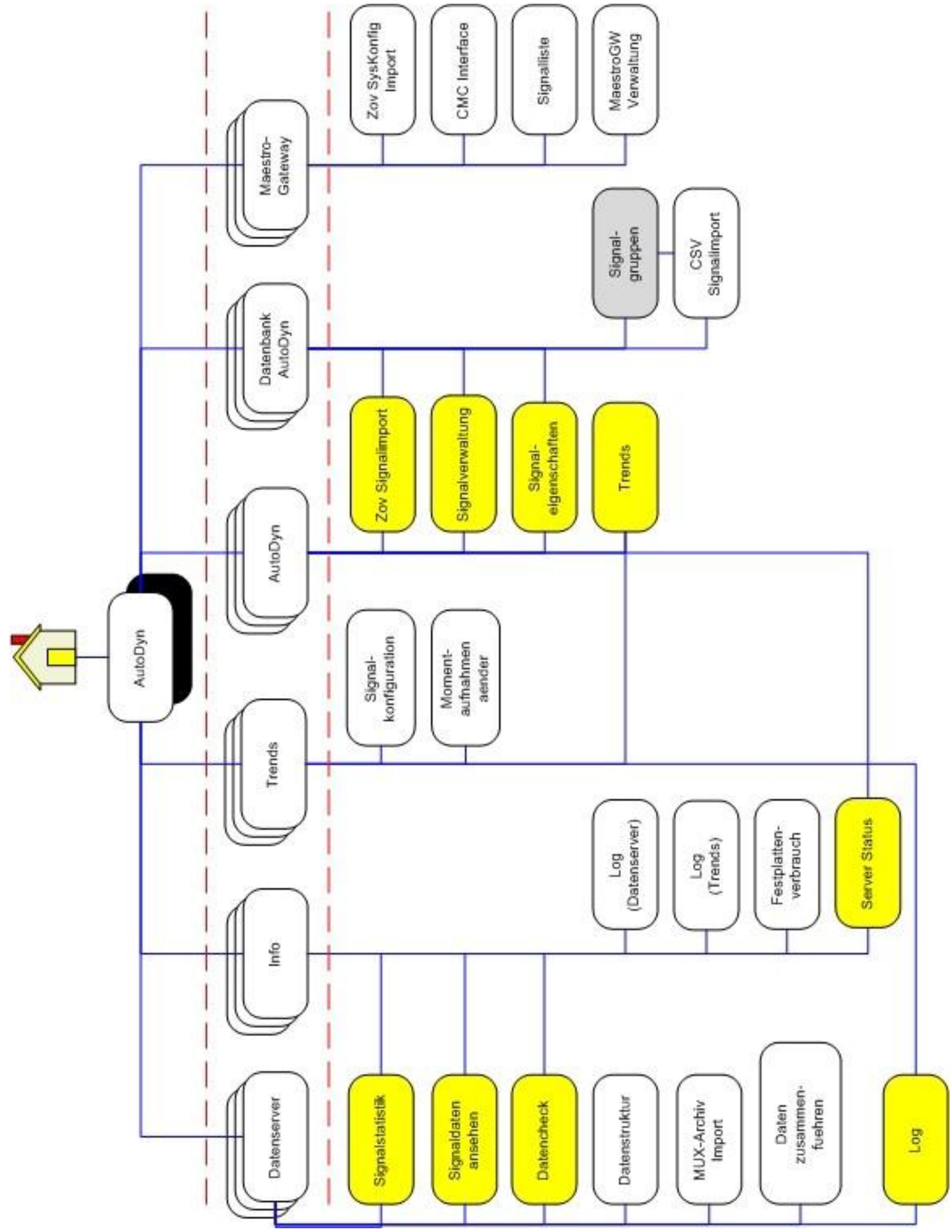


Figure 7 - TGtool structure diagram

The AutoDyn category offers five functions; they are ZOV signal import, signal management, signal properties, trends and server status. The categories are described in more detail below.

ZOV means “Zentrales Objekt Verzeichnis”. On this page (see Figure 8), signals from ZOV can be imported. There are several filter opportunities to refine import request.

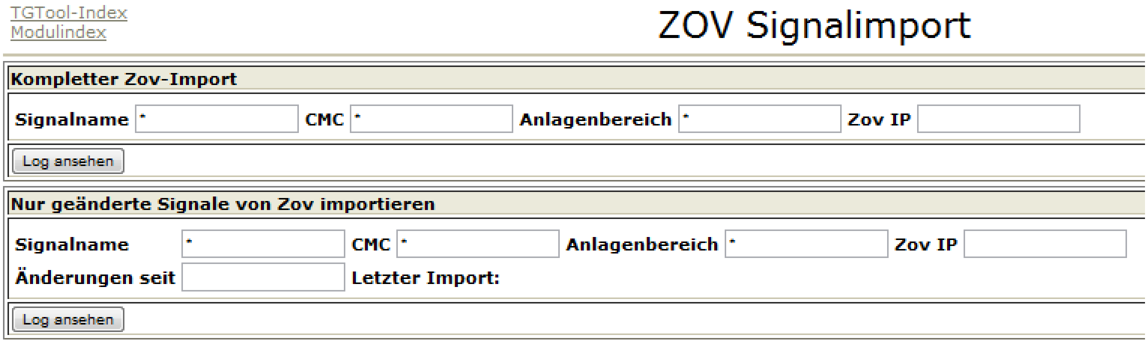


Figure 8 - TGtool ZOV signal import

Signal management (see Figure 9) allows users to see all the signals in the system, add new signals manually and change their reference group.

Figure 9 - TGtool signal management 1

If an end user types something into the filter line editor and clicks “Suchen”, a list of signals matching the filter request will appear (see Figure 10).

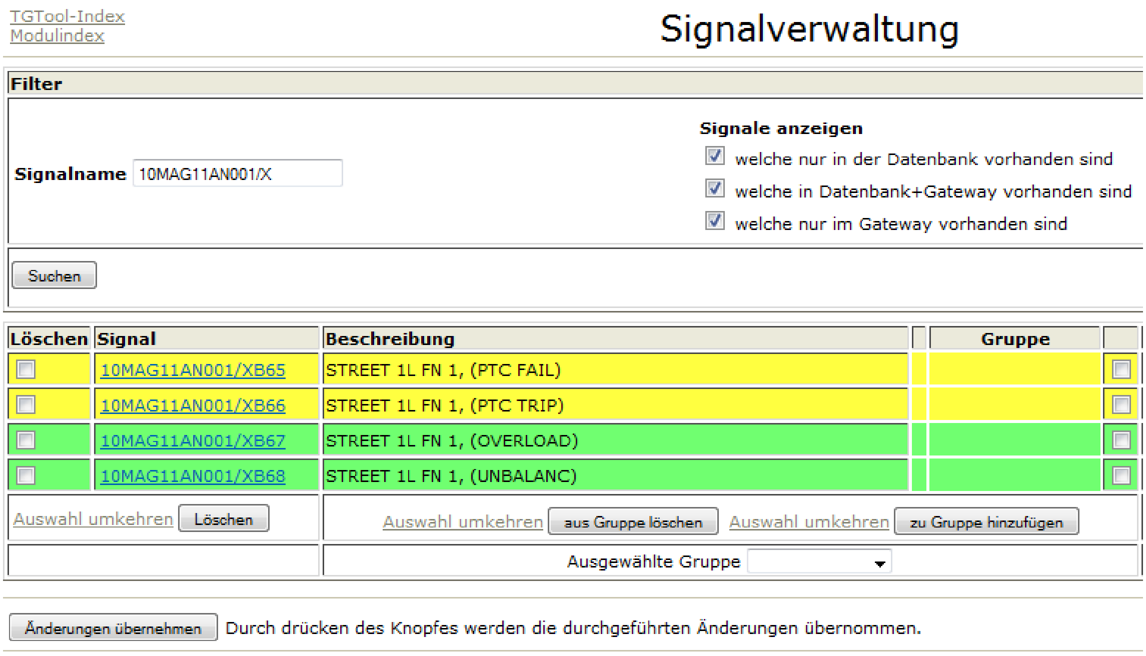


Figure 10 - TGtool signal management 2

Signals colored green are active, but do not belong to any signal group. Yellow signals exist in the database, but not at the controlling system.

The signal properties window (see Figure 11) displays metadata of the filtered signals. Here, the properties of the signals can be edited.

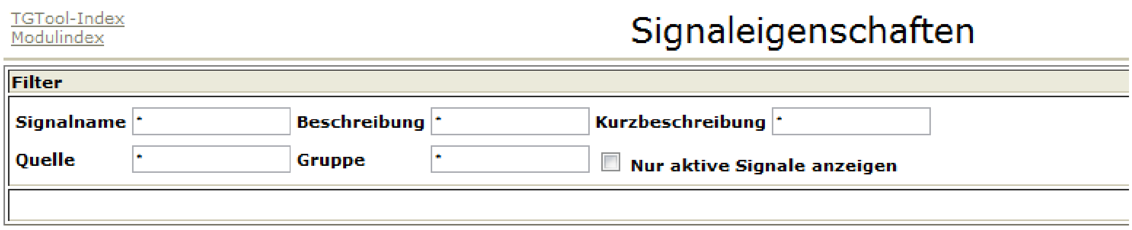
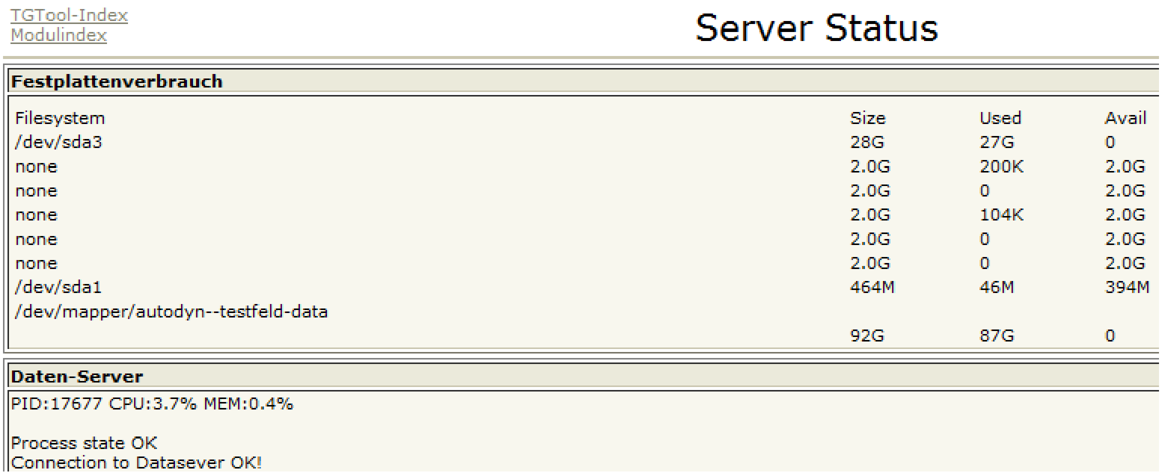


Figure 11 - TGtool siganl properties

The server status page as demonstrated on the Figure 12 allows checking all running processes at one glance.


TGTool-Index
Modulindex

Server Status

Festplattenverbrauch

Filesystem	Size	Used	Avail
/dev/sda3	28G	27G	0
none	2.0G	200K	2.0G
none	2.0G	0	2.0G
none	2.0G	104K	2.0G
none	2.0G	0	2.0G
none	2.0G	0	2.0G
/dev/sda1	464M	46M	394M
/dev/mapper/autodyn--testfeld-data	92G	87G	0

Daten-Server

PID:17677 CPU:3.7% MEM:0.4%

Process state OK
Connection to Datasever OK!

Figure 12 - TGtool server status

Almost all pages demonstrated above do not show data before filtering is applied. This is important, as a system may have thousands of signals and without applying a filter, a user may wait for a long time until page is loaded. However, every page is simply designed and does not offer much functionality. Mostly one page has a single function.

This was a short overview of the general look of the AutoDyn configuration tool.

2. Background and Related Work

2.1. Complexity

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 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 (Thomas & Richards, 2008).

Many dimensions are involved in Human-Computer Interaction (HCI). Each of them influences the overall complexity and thus usability of the graphical user interface (GUI).

A GUI is in fact an interface between a user and a system. Thus interacting with an application, users are interacting with a system behind it. System complexity, however, should be hidden from users, so that they can better 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 task (Ince et al, 2009). 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 graphical user interface complexity. The GUI is then perceived by users (see Figure 13) who create a mental model (cognitive map) of the system in their head. The complexity of this model is referred to as cognitive complexity.

In the paper “How to measure cognitive complexity in HCI” Matthias Rauterberg (1996) transferred the broad definition of cognitive complexity into the human-computer interaction context. According to him “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”.

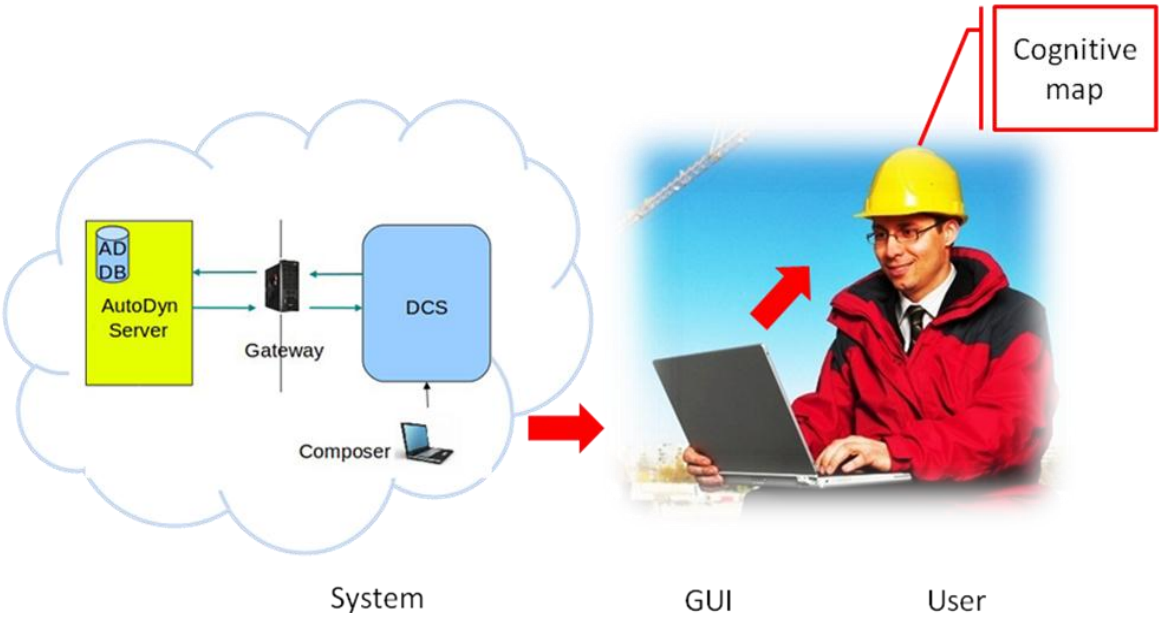


Figure 13 - Human computer interaction[2]

Although system and task complexity are constant, GUI designers can manipulate visual complexity, which affects users’ cognition.

2.1.1. Theory of visual complexity

Donderi (2003) in his review “Visual Complexity: A Review” summarizes the scientific research of the last century on visual complexity, and describes how the idea

---

[2] Picture is taken from http://coloradoenergynews.com/category/oil-and-gas/resources/

of visual complexity follows from its origin in gestalt psychology of the early 1900s to its present place in neural circuit theory, algorithmic information theory, and perceptual learning theory.

Two main approaches in understanding visual perception were developed in late nineteenth and early twentieth centuries. One approach emphasized analysis of the visual stimulus into sensory units, whose perceptual effects could be separately measured. According to this approach, the laws of perception were associationistic combination rules for the sensory units. While the analysts hoped to explain perception by first understanding its component parts, the synthesists produced visual experiences that the analysts could not explain. Gestalt psychology (Koffka, 1935; Kohler, 1947) was the result. This was the second approach, which was based on the evidence that the perceptual properties of many stimuli could not be gained by analyzing their separate parts. Instead, the laws of visual perception were rules that specified how the entire visual sensory input could be organized into a percept (Donderi, 2003). However, "good form" is the central idea that survived from gestalt psychology into the twenty first century, but there is little left of its original content, as expressed, for example by Koffka (1935) or Kohler (1947).

Many researches were devoted to the investigation of the perception of a single visual form. Anderson & Leonard (1958) found that the greater the degree of randomness in the figure set, the faster the figures could be recognized. Generalizing from this result, forms with more information (more "complex" forms) are easier to discriminate, while forms with less information ("simpler" forms) are harder to discriminate on the first reproduction trial, but take fewer trials to accurately reproduce. Mavrides & Brown (1969) varied the Fitts et al. (1956) technique and found out that constraints within a set which reduced the information (relative to the rest of the set) conveyed by each stimulus, made the forms harder to reproduce accurately on a first trial, and harder to discriminate from the rest of the set. Thus random images may be judged to be less complex than images that are structured in complex, non-random ways. Wolfram (2002) suggested, that the subjective complexity of an image will be at a maximum somewhere between simple order and complete randomness, and that complete order and complete randomness are not that perceptually different. In the extreme, an entirely black computer screen would be judged not much less complex than a screen generated by a random collection of red, green and blue pixels (Donderi, 2003).

Algorithmic information theory (AIT) tried to find a ‘code’ of symbols that could represent a visual form, and to equate the complexity of the visual form with the length of code required to represent the form. It appeared that there was a close relationship between the code length required to specify a visual form and the communication theory conditions established to generate that form. This approach led to the application of AIT to visual perception. AIT attempts to reconcile the competing claims of modern gestaltists and modern associationists. This theory is computational and does not invoke knowledge about the central nervous system. A practical analogue of the AIT definition of array complexity was found in data compression algorithms, and the possible use of a data compression algorithm to generate a measure of visual complexity (Donderi, 2003). A digitized image compression, for example, was an easy and reliable way to assess visual complexity. Some researches provided strong evidence that the file size measure predicts the subjective complexity rating of images (Donderi et al, 2005, Donderi, 2006). The other research of Stickel et al (2010) used digital image compression algorithm as a basis for generating a novel measure called the XAOS metric.

A convergence of ideas in AIT, perceptual learning theory and neural circuit theory has led to the following synthesis: according to perceptual learning theory, a simple visual form generates activity in higher-order levels of the visual system, while a complex form generates activity from these more central areas all the way out to the more peripheral sensory input areas of the visual system. In general: simple forms activate central visual areas; more complex forms activate central and peripheral visual areas. Simple forms have shorter AIT computational codes, which corresponds to their activation of fewer and less specific cortical sites; complex forms have longer AIT computational codes, corresponding to their activation of more and more specific cortical sites (Donderi, 2003).

2.1.2. Single display complexity

The information provided by a single display (window or a web page) has to be evaluated with respect to the context – all of the information needed to perform the task, which the display is designed to serve. The amount of information necessary to carry out the task puts a lower limit on the information complexity of the display. The GUI designer must consider the advantages and disadvantages of partitioning the required information across separate displays, which gains display simplicity but loses immediate accessibility, versus designing a single, always visible, but more complex display. Some recent evidence (Burns, 2000) suggests that putting more information on an already complex display can improve performance in a complex monitoring task (Donderi, 2003). Other investigation (Coskun & Grabowski, 2004) shows that expert users working with decision-supporting GUIs find complex displays, with immediate accessibility of all necessary information for decision-making, more usable, than visually simpler displays, where extra clicks have to be done in order to get decision-relevant information.

The organization and configuration of display information is as important as the total amount of information on the display. Many investigations focus on what kind of visual analogues should be used for numerical variables, how those variables should be combined, and how the display organization relates to the mental model of the task being controlled (Bennett et al, 1993, Carswell & Wickens, 1996; Donderi, 2003, Vicente et al., 1996).

Level of training will also interact with GUI complexity and task efficiency (Coskun & Grabowski, 2004). The complex GUIs require time to learn, an experienced user will be better at dealing with a given level of GUI complexity than an inexperienced user, and designers have to consider the tradeoffs between the information required to do the task, the kind of training required to use a GUI, and the complexity of the display (Donderi, 2003).

Complexity is a quantifiable property of a visual display. Its lower limit is specified by the information required to complete the tasks. The job of a GUI designer is to arrange and sequence the information to suit the task demands and to meet other criteria of efficient display design: for example, proximity compatibility (Wickens & Hollands, 2000, pps 97 – 100), integrality versus separability (Bennett et al, 1993), or information dimension selection (Carswell, 1992, Carswell & Wickens, 1996, Donderi, 2003).

2.1.3. Visual complexity and emotions

Arnheim (1966) defines complexity as “the multiplicity of the relationships among the parts” (p. 123). Harper et al. (2009) tried to define visual complexity as an implicit measure of cognitive load and found out that users expressed visual complexity in relation to the visual rendering of web pages. Thus, visual complexity measures the likelihood of cognitive load contained in the “browsing task.” Objects, which make pages more complex, are not necessarily highly visual components (e.g. pictures, photos), but those that signify the possibility of increase in cognitive load (interactive components).

The visual properties of complexity have been heavily researched in the studies of visual preference for natural and built environments and found to be the prominent dimension of environmental aesthetics evoking automatic human emotional response to the environment, such as pleasure and preference (Arnheim, 1966; Birkhoff, 1933; Kaplan & Kaplan, 1983; Nasar, 2000; Ngo & Byrne, 2001).

Bucy (2000) argues that emotional responses may determine which interfaces people choose to use as they seek pleasure or enjoyment beyond just task efficiency. This emphasizes the visual design features of the webpage interface, and stresses the need for interfaces that promote engagement, pleasure, and delight rather than just functionality or ease-of-use (Deng & Poole, 2010, Marcus, 2002, Wright et al., 2001).

It is generally agreed that human emotional response has at least two components: arousal and valence (Barrett 1998, Lang 1994, Reisenzein 1994, Russell 1989). The arousal is defined as the subjective experience of energy mobilization for psychological and motor activity (Russell and Barrett 1999). While arousal is a nondirectional component of emotional response, valence measures the direction of emotional response ranging from positive to negative (Russell, 1980). It is a subjective feeling of pleasantness or unpleasantness (Barrett, 1998, Deng & Poole, 2010). Arousal provides the basis for emotional response when a stimulus is detected, while the valence of the emotion depends on the person's interpretation of the felt arousal (Frijda, 1986).

Many theories have tried to relate arousal to emotional valence. Among them, reversal theory has received increasing attention (Apter, 2001, 2003) due to its emphasis on the dynamic aspects of human experience and behaviour. Reversal theory attempts to capture different motivational possibilities of human behaviour in a systematic manner (Apter, 1995). Two different metamotivational states are proposed—telic versus paratelic states—in which changes in felt arousal are interpreted and experienced in opposite ways (Apter 1982). Reversal theory holds that both low and high levels of arousal can be pleasant, depending upon which metamotivational state is operative (Deng & Poole, 2010).

The environmental psychology model proposed by Mehrabian & Russell (1974) (the M-R model) suggests that emotions function to mediate the effects of environmental stimuli on behaviour. It assumes that people's emotions determine what they do and how they do it, and that people respond with different sets of emotions to different environments which, in turn, induces individuals to approach or avoid these environments. Approach–avoidance behaviour of web user's toward the website not only reflects the user's perception of the quality of the website, but also strongly predicts desired user behaviours pertaining to the measurement of the success of a website, such as customer satisfaction, total number of website hits, user's return rate or future patronage, etc (Deng & Poole, 2010).

Studies of environmental aesthetics and preference have investigated the influences of complexity on arousal. Complexity has been consistently shown to be positively related to arousal (Berlyne 1971; Gilboa and Rafaeli 2003; Heath et al. 2000; Nasar 1987, 1997). A high level of complexity provides diverse and numerous information signals that require considerable attention and time to view and understand.

It thus serves as a source of stimulation and interest, which provoke more energy mobilization and higher levels of arousal in individuals. This finding is consistent with Kaplan and Kaplan's (1983) notion that the complexity present in the environment aids in the involvement process by invoking and maintaining the viewer's interest in the environment.

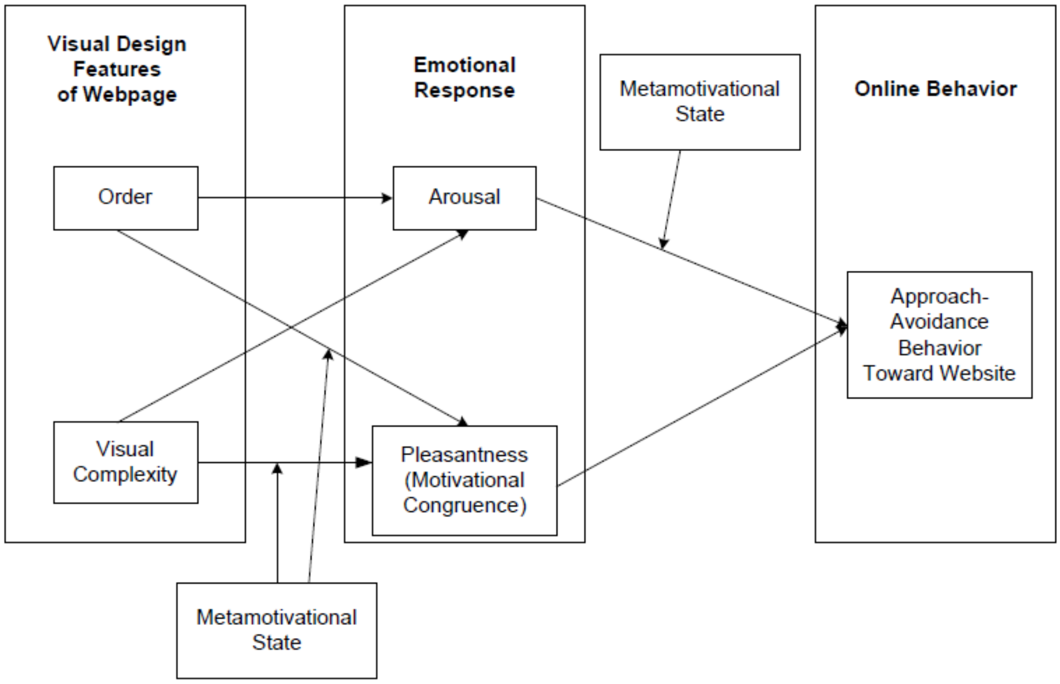


Figure 14 - Model of web page visual complexity and order, emotional responses, and approach-avoidance behaviour (Deng & Poole, 2010)

The study of Deng & Poole (2010) applied the hypotheses on environmental aesthetics and preference to the study of the emotional impacts of webpage visual complexity and order design features (see Figure 14). The results suggested that the salience/importance of webpage order and complexity to web users' pleasantness was largely dependent on the web users' metamotivational states. For web users in a telic state, who are usually motivated by a clearly defined goal and who emphasize the process of comprehending the website, webpage order seemed to elicit users' pleasantness and to motivate their approach tendency, due to its critical role in aiding sensemaking. However, webpage visual complexity is perceived as less important because it promotes involvement and interest rather than understanding. Conversely, when in a paratelic state, webpage visual complexity seems to be a more important design feature than webpage order in evoking pleasantness and promoting approach tendencies due to the important role of complexity in satisfying users' needs for stimulation and arousal.

2.2. Usability measurement

Monkey: "You are so long!"

Boa Snake: "Well I know that… but HOW long?"

Monkey: "Very long"

Boa Snake: "Very… "Very" is not a good answer!"

This dialog is taken from the Soviet Union cartoon "38 Parrots" (see Figure 15) where a monkey, an elephant and a parrot try to find out how long their friend Boa Snake is. This task seems impossible for them, as they do not know how to measure.

Figure 15 – "38 Parrots"

John Brooke (1996) stated that usability could only be defined with reference to particular context. Since the usability of a GUI is defined by the context in which it is used, measures of usability must necessarily be defined by that context too.

In the context of the cartoon "38 Parrots", the adequate measure of length became other animals.

Parrot: "You are 38 parrots, 5 monkeys or 2 elephants long."

Boa Snake: "Hah! In parrots I am so much longer!"

Heuristic evaluation (Afacan & Erbug, 2009, Karahoca et al., 2010), focused heuristic evaluation (Holzinger et al., 2009), cognitive walkthrough (Jaspers, 2008), thinking aloud (Stickel et al., 2009), shadow expert technique (Holzinger et al., 2009), questionnaires (Coskun & Grabowski, 2004, Holzinger et al., 2005, Liljegren, 2006) and other methods can be applied to evaluate usability. Each of them has its strengths and weaknesses, which must be considered by usability evaluators.

Jaspers (2008) investigated usability evaluation methods: the heuristic evaluation, the cognitive walkthrough and the think aloud. The first two of them belong to 'expert-based' methods, whereas the last one is 'user-based'. This means that in heuristic evaluation or cognitive walkthrough methods evaluation should be performed by usability experts. For the think aloud test, the end users are evaluators.

In her work, Jaspers compared all three methods and derived their advantages and disadvantages. The result of the heuristic evaluation appeared to heavily correlate with evaluators' experience. Generally this method is quite unstructured and allows finding a broad range of low priority ("cosmetic") and non-specific problems. The cognitive walkthrough is a more structured approach which allows identifying more severe problems. The think aloud is a good method to gain better understanding of the user's cognitive process and it allows revealing significant problems of a GUI.

The main idea of this research is that each usability evaluation method has its strengths and weaknesses; none of them is perfect. It is more effective to combine different techniques that complement one another as their collective application will tell the investigator more than the techniques applied in isolation (Jaspers, 2008).

Another interesting observation was made by Liljegren (2006).The researcher investigated application of usability evaluation methods in safety-critical context. Though usability evaluation methods judge different aspects of usability and provide different information about the product, it is important to apply methods that evaluate the desired aspect(s) of usability, as, for example, evaluating only the satisfaction aspect of a safety-critical system, or the efficiency aspect of a computer game would be inadequate. The author argued that the importance of each of five main aspects of usability according to Nielsen (1993) can be unequal, at least in the end user's opinion (Liljegren, 2006).

To clarify if this assumption may take place the researcher created a questionnaire and distributed it to medical personnel, which was constantly interacting with medical equipment. The questionnaire had, aside from questions for the statistics sake, five main questions referring to the importance of the five usability aspects. Respondents distributed 100 points over those five aspects. The aspects, which were more important, received more points.

The expectations were confirmed. Statistically tested results demonstrated that the most important aspect of usability in medical context is suggested to be "hard to make an error" and the least important "pleasure to use" or satisfaction factor. Other aspects (learnability, efficiency and memorability) showed equal results (Liljegren, 2006).

Karahoca et al (2010) provide a comparative analysis between two prototypes developed for Tablet PCs for an emergency department. The aim of the research is to pick out the tool, where the efficiency, learnability and fulfillment of the system immediately come into prominence (Amouh et al, 2005). Two prototypes had the same functionality but different interfaces. One of them had a text-based GUI, while the other one an icon-based GUI (the icons were selected by a sample of healthcare staff). By applying heuristics evaluation together with cognitive walkthrough, the usability test was completed by 32 evaluators – potential users of the software.

Results demonstrated that the icon-based GUI was more efficient, learnable and easy to use than the text-based one. The need of simplified forms with fewer elements and lower complexity became evident.

This research demonstrated that software applied in an emergency department should avoid complexity. The interface should be as self-explanatory as possible and support users to choose correct action for any result they want to achieve. This contribution could be used as a guideline for all safety-critical system interfaces (Karahoca et al, 2010).

2.3. Objective and subjective metrics and methods

The file size of a compressed image (Donderi & MacFadden, 2005), (Donderi, 2006), the XAOS metric introduced by Stickel et al (2010), the gestalt-like perceptual measure method proposed by Hsiao & Chou (2005), number of key presses (Buchanan, 2001) etc. provide an objective measure of complexity and usability of a GUI. These metrics and methods deal with properties which do not depend on user perception, whereas heart rate, skin conductance level, user satisfaction level, heuristics, thinking aloud etc. provide a subjective measure of complexity and usability, addressing user perception of a GUI. In this chapter, different researches on complexity and usability measurement are presented.

Visual complexity of a GUI is a crucial factor for usability, since it influences the cognitive load and forms expectations about the software and the system behind it. A novel method to calculate the visual complexity of a website is proposed in the paper of Stickel et al (2010). The novel metric, called XAOS metric, is based on the entropy of the 30 web pages' screenshots and on the number of actions and organizational elements of each website. This method has been evaluated against a well-known objective approach of using the file size of color JPEG images for determining visual complexity applied upon all 30 websites. The websites have also been evaluated subjectively with a web survey in order to produce an optimal comparison basis. The result of this investigation is shown in Figure 16. A strong correlation for both methods on subjective ratings of visual complexity and structure can be seen. This suggests that both methods are reliable for computation of visual complexity (Stickel et al, 2010).

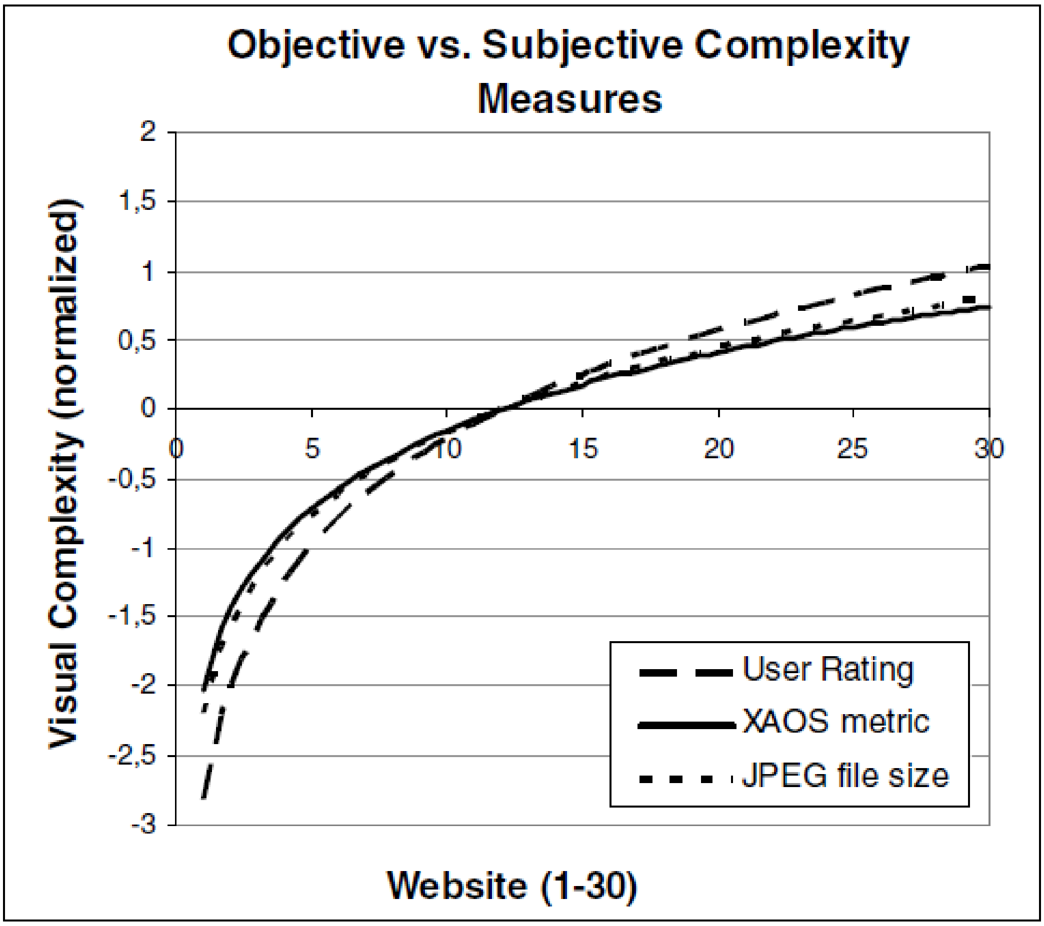


Figure 16 - The XAOS metric matches the JPEG file size method and user ratings (Stickel et al, 2010).

In recent, the department of computer information systems in the Yarmouk University in Jordan did extensive research on effective test automation of GUIs. They published a chain of papers (Alsmadi & Magel, 2007), (Alsmadi & Al-Kabi, 2009), (Alsmadi & Al-Kabi, 2011) describing the results of their research. In these papers they concentrate on the objective side of complexity, applying objective metrics and methods to GUI complexity test automation.

In earlier research, they came up with GUI-structural metrics which can be applied in automotive testing if presenting a GUI as a hierarchical tree. The total number of controls, tree depth and maximum horizontal width of the tree – these structural metrics do not represent user behavior, however, they allow evaluation of an objective complexity of a GUI structure. Using structural metrics it becomes possible to automate GUI evaluation.

In further papers they developed a GUI test automation tool, which creates an XML tree to represent a GUI structure. Scientists of Yarmouk University came up with several algorithms for test case generation. The GUI structure evaluation tool created unique test cases that provided good coverage of the different paths in tested applications.

They continue working on tool improvement and further testing of the test case generation algorithm's effectiveness (Alsmadi & Magel, 2007), (Alsmadi & Al-Kabi, 2009), (Alsmadi & Al-Kabi, 2011).

The automation of GUI testing attracts more and more interest of software companies, as it promises useful and usable GUI with little effort and at low cost. However, automation techniques do not cover the whole range of usability problems. Domain knowledge, context of application, end user specific parameters etc. are not involved into the automated testing process. Therefore it is important to complement objective evaluation methods with subjective techniques.

Stickel et al (2009) extended standard usability methods utilizing the valence arousal space. Emotions have an impact on cognitive processing during the interaction a user performs with a GUI, e.g. joy improves creative problem solving, whereas anxiety or stress creates barriers. It seems that attractive things work better (Norman, 2004). In order to check this hypothesis Stickel et al. applied the circumplex model of affect from Russell (1980) to UX dimensions (see Figure 17).

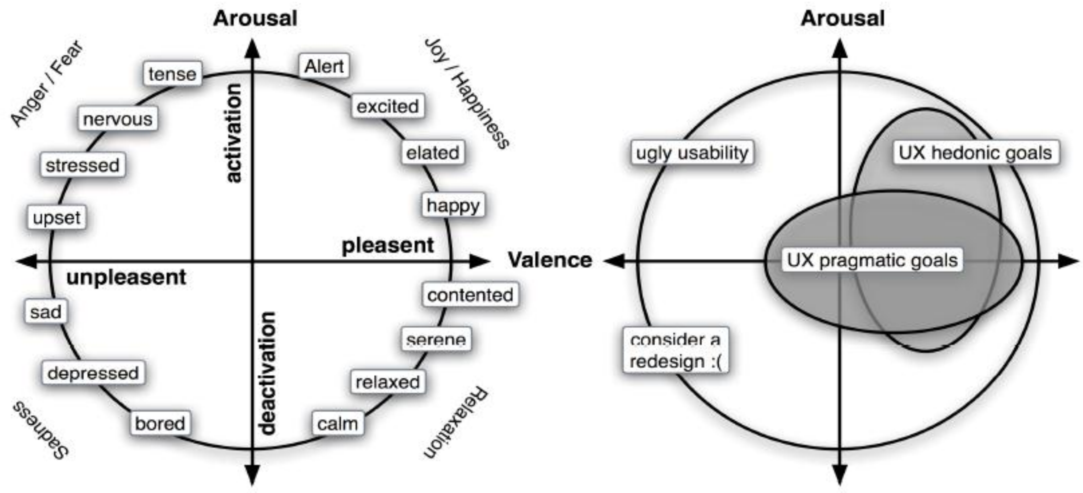


Figure 17 - The circumplex model of affect from Russell (on the left) applied to UX dimensions (on the right) (Stickel et al, 2009)

Users had to undergo a performance test and a short thinking-aloud test. In order to detect emotion changes during the tests, heart rate and skin conductance level have been measured. Analysis demonstrated a positive correlation between users' performance and their emotional state (Stickel et al, 2009).

In order to evaluate consistency of the learning management system of the Technical University of Graz Holzinger et al (2009) chose consistency-relevant heuristics and carried out a consistency focused heuristic evaluation (FHE). Next, researchers were divided into two groups. Each group created several tasks based on the FHE result. Test persons were invited to perform those tasks. Seven participants were assigned to perform tasks of the first group and another seven – tasks of the second group. The task processing part was recorded on video, both participants' activity on screen and their faces (for mimic analysis).

Further, a video of the average user was selected and demonstrated to the other group of experts. That second group watched the video and without help tried to guess the task participant was performing. Further, the mimic and emotional state of the participant was analyzed. The applied technique is called shadow expert technique (SET).

Holzinger et al (2009) expected that SET results would intersect the FHE results; however, SET provided deeper insight into the consistency problem, as it paid more attention to the user's progress in task processing and his or her emotional state.

Tuch et al. (2009) examined website complexity within the framework of aesthetic theory and psychophysiological research on cognition and emotion. They hypothesized that increasing the complexity of websites would have a destructive cognitive and emotional impact on users. 48 test persons took part in a passive viewing task (PVT). 36 website screenshots differing in their degree of complexity (defined by JPEG file size) were presented to participants in randomized order. Additionally, a standardized visual search task (VST) assessing reaction times, and a one-week-delayed recognition task on these websites were conducted and participants rated all websites for arousal and valence. Psychophysiological responses were assessed during the PVT and VST. Visual complexity was related to increased experienced arousal, more negative valence appraisal, decreased heart rate, and increased facial muscle tension. Visual complexity resulted in increased reaction times in the VST and decreased recognition rates. Reaction times in the VST were related to increases in heart rate and electrodermal activity. Tuch et al (2009) demonstrate that visual complexity of websites has multiple effects on human cognition and emotion, including experienced pleasure and arousal, facial expression, autonomic nervous system activation, task performance, and memory.

2.4. Usability Evaluation in Safety-Critical Systems

A graphical user interface 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).

They compared complexity of original and improved versions of Navigation and Piloting Expert System (NPES). This is an operational decision support system (DSS) which provides 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 same task scenarios in both NPES-1 and NPES-2 and afterwards fill out a questionnaire and give an interview.

The result showed that all 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 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 (Coskun & Grabowski, 2004).

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) were interested in application of heuristic evaluation methods in another sphere of our life - in building design. They chose a shopping mall, which had already been built and opened for customers. Their aim was to find out if the design of the mall was universal, if its customers could expend little physical effort and have security, safety and simplicity (Afacan & Erbug, 2009).

Observing a building as a safety-critical system, they aimed not just to evaluate the mall's design, but to find out if the problems and errors in design could have been noticed before and handled at the right time. According to this principle they divided all design problems into two categories: major and minor. First are those which should have been handled in the planning phase and it is not possible to correct them later, and the latter are those which can be changed without extraordinary costs.

The team of researchers chose heuristics evaluation and worked out several scenarios. Five different types of architectures were chosen for this research. They evaluated the drawings first from their offices, before they saw the building itself (session I) and found out as many problems as they could see. At session II evaluators moved to the building, where they completed some scenario tasks walking through the mall, thinking aloud all the time. After the task was completed evaluators made a last interview on construction drawings and conducted task scenarios (session III).

The most valuable result of this research is that out of fifty three (53) usability problems identified by all five evaluators, 28 were identified as major and ~64% of major problems have already been identified in session I – 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 (Afacan & Erbug, 2009).

Safety-critical systems are not always effectively testable in artificial conditions. The usability of driver information systems used in cars is highly important, especially a distraction component, as the road is unpredictable and demands the full attention for safe driving. Funk & Hamacher (2008) demonstrate the successful application of automatic usability evaluation instruments in car driver IS.

They introduce a two-step approach; where an observation component is embedded into the driver IS and collects usage data during habitual use. This component is a D'PUIS[3] framework, more information on it can be found in the work of Funk et al (2008). During the second step the collected data is put into a rule-based expert system REVISER[4], 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, 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.

---

[3] http://www.softreliability.org/dpuis

[4] http://www.hamacher.eu/reviser

This experiment demonstrated that the use of automatic usability evaluation methods can speed up usability testing and make this process more valuable (Funk & Hamacher, 2008)

2.5. Usability Engineering considered harmful

The paper of Greenberg & Buxton (2008) points out the limitations of usability evaluation. This is not the first, but one of many criticizing papers which argue that HCI is more than usability evaluation.

The authors present examples in which usability evaluation is harmful and demonstrate historical facts upon novel technologies which became widely used, however, if at that point of time usability evaluation had been performed, the technology would have failed. Limitations of usability evaluation include ignoring the innovations' culture of use and concentrating on getting the design right instead of getting the right design (Greenberg & Buxton, 2008).

Different interface ideas prototyping methods are described in a paper of Memmel et al (2010) "Agile methods and visual specification in software development: a chance to ensure universal access". They write about low-fidelity prototyping methods, like paper prototypes, sticky notes etc. These are cheap and fast visual techniques one can employ in a design process for sketching GUI ideas. There are also high-fidelity prototypes, which range from detailed drawings to fully interactive simulations (see Table 1) (Memmel et al, 2010).

Type	Advantages	Disadvantages
Low-Fidelity	less time & lower cost evaluate multiple design concepts communication device address screen layout issues	limited usefulness for usability tests navigational and flow limitations facilitator-driven poor specification
High-Fidelity	partial/complete functionality interactive user-driven clearly defines navigational scheme use for exploration and test marketing & sales tool	time-consuming to create inefficient for proof-of-concept designs blinds users to major representational flaws management may think it is real

Table 1 - Low- & high- fidelity prototyping (Rudd et al, 1996)

A sketch typically illustrates only one of many possible designs and variations under consideration. Early design demands many idea sketches, reflecting on this multitude of competing ideas, and choosing the one that appear the most promising (see Figure 18 b). The promising idea is then further varied and developed until it can serve as a testable prototype. That is, sketching is about ‘Getting the Right Design’ (Buxton, 2007, Tohidi et al, 2006). Only afterwards work on ‘Getting the Design Right’ of a particular idea can be done, through iterative testing and development. Thus sketching is akin to a heuristic that helps one move closer to the global maxima by circumventing the local hill climbing problem (Greenberg & Buxton, 2008).

Usability evaluation of sketches leads to local hill climbing, where much effort is expended in ‘Getting the Design Right’ (see Figure 18 a). Unfortunately, evaluation of early sketches is often at the expense of considering and developing other, probably better, ideas (Greenberg & Buxton, 2008).

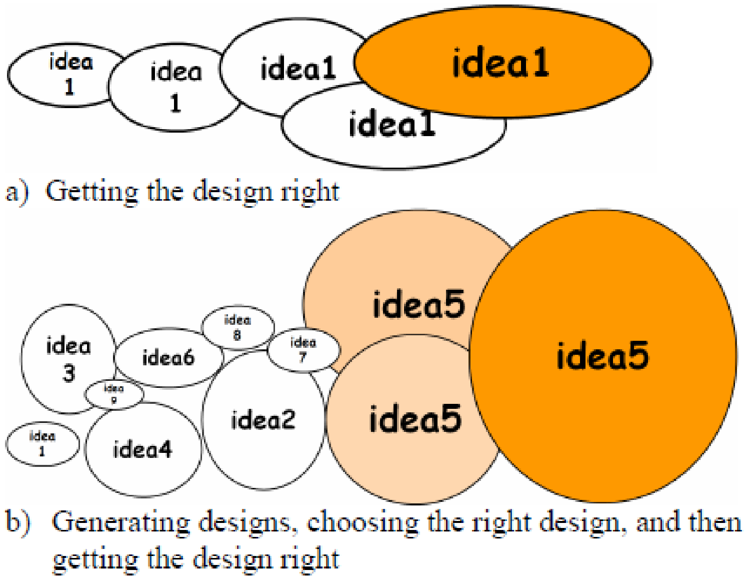


Figure 18 - Sketching first, iterative design and evaluation later (Greenberg & Buxton, 2008)

Interestingly, standard methodologies highly recommend, even require implementation of usability evaluation from the very beginning of the product development, however, Greenberg & Buxton (2008) consider this approach to be harmful on the sketching phase as it destroys ideas. “The net result is that we eliminate ideas too early, we consider far too few ideas at all, we converge on that which we can measure, which is almost always that which we are already familiar with. Our work degrades into a refinement of the known rather than innovation along new trajectories,” – Greenberg & Buxton (2008) claim.

The message this paper conveys is to think broader, not stay within limited methodologies, which focus on the problems instead of benefits.

3. Materials and Methods

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. This narrows the choice of possible test users and evaluators. For this project three persons can be involved as domain experts. Only they will be able to process go-live configurations (complex tasks). Applied usability engineering and evaluation methods should take this limitation in 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. Finally, several simple tasks[5] were created and a thinking aloud test was performed by the real end users of the tool. Thus, usability problems, which were not detected during interviews and development process, were found. Thinking aloud test results enforced the last optimizations of the tool.

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

- Given routine tasks, the amount of mouse clicks necessary for completion of each of the tasks in both configuration tools was calculated (Buchanan et al, 2001).
- Heuristic evaluation from Nielsen (1994) was completed for both the old and the new GUIs.

---

[5] Please, note that in this work only simple tasks for routine configurations will be evaluated, as the development effort for go-live configurations implementation has been underestimated at the research starting phase and will be implemented and evaluated in future work.

This is the general approach to the research task:

- 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

First, three investigation 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 gave evaluating interviews after each development iteration.

Detailed realization of the methodology is described in the next sections.

3.1. Configuration types

After three interviews with domain experts it was cleared up that the main functionality of the TGtool is continuously saving signal data and providing it for visualization. There are two types of essential configurations the tool should provide:

- Done once (setup and configuration at go-live phase)

- Continuously repeated (maintenance, adaptation to controlling system changes at running phase)

The configurations of the first type are very complex and usually done once by Technikgruppe experts and therefore should have a functional GUI considering a software developer as an end user. This GUI should not be available for the users who do not possess required knowledge.

The second type of configuration is a maintenance configuration and is done by the PLS (ProcessLeitSystem) technicians and possibly by other technical personnel from the client side. It is important to make the GUI for these configurations very intuitive and user-oriented.

3.1.1. Setup configuration

These configurations are usually done once; one of the interviewees (the end users of this tool) has proposed the idea of a step-by-step navigation through the configuration. It seems logical, as this type of configuration definitely has a step-by-step character.

3.1.1.1. Installation requirements

All other software, tools and the database must be installed and ready to use before the configuration tool can be installed.

3.1.1.2. Step-by-step actions

A graphical view of the setup configuration flow can be seen in Figure 19.

1. AutoDyn Server IP must be provided so that the configuration tool knows where the database (DB) is situated.
2. Database preferences (username and password) must be defined.
3. The database should have the access data (IP, name, etc.) of the drivers or it should be inserted.
4. If the Maestro driver has been installed, the option of ZOV data read should appear automatically. Otherwise the CMCs and/or other hardware should be inserted manually.
5. The destination to save signal data should be provided (CVS file, DB etc.).
6. The signal list and signal metadata must be gotten from hardware or from supporting information system (e.g. ZOV) or via CSV import.
7. Signals should be organized in groups.

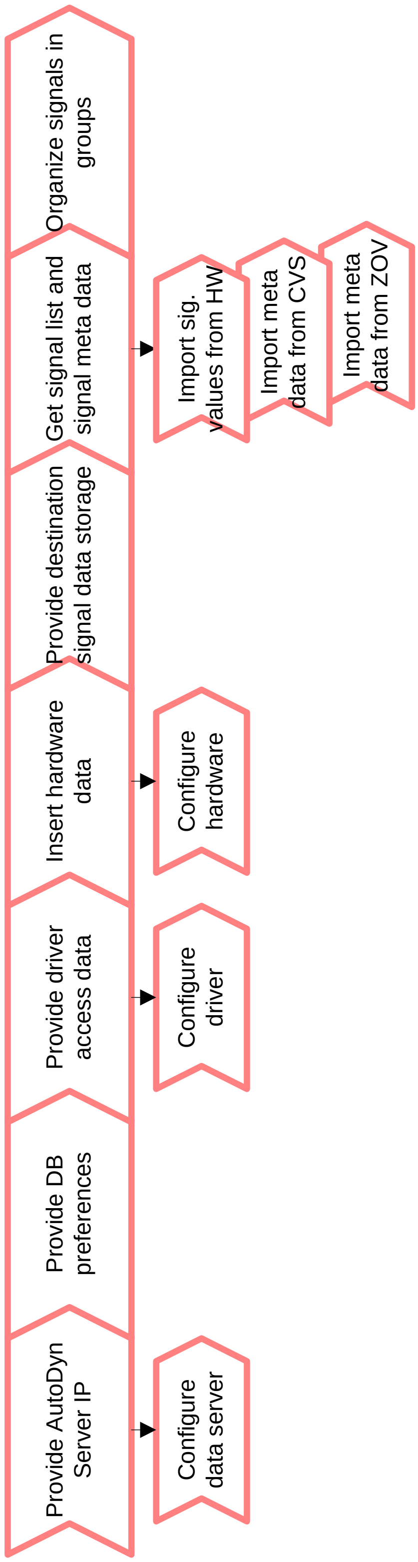


Figure 19 - Setup configuration flow

The following functionality belongs to core functionality, however, it is not essential for communication between controlling system and AutoDyn:

- Configuration of trend images and Snapshots
- Configuration of the report server

3.1.2. Maintenance configurations

Maintenance configurations are the most often used configurations. At the moment this configuration type is system-oriented and hard to use. Even the users, who have been working with this tool for years, are not able to add a signal to the AutoDyn without searching, thinking and trying out, whereas several clicks irrelevant to the task are done in order to “find a way around”. Concluding from the interviews, the following tasks belong to routine:

- Add/delete signals.
- Change signals’ metadata values.
- Import of signals’ metadata.
- Add/delete signal groups.
- Import from ZOV.
- Import directly from hardware (CMC) through a gateway.
- Import from controlling system through CSV.

3.2. Requirements

Requirements have been defined by the Technikgruppe and ought to be considered during development. They are as follows:

- The configuration tool has to be a stand-alone application for Windows.
- The application has to be written in Python.
- The graphical design has to be processed in Qt framework[6].
- The test users should be real end users of the application and employees of Technikgruppe.

[6] http://qt.nokia.com/

3.3. TGtool analysis

During the interview, the current end users of the TGtool ran into problems and had difficulties with obvious weak points of the tool all the time. The request to demonstrate the whole functionality was already too much for two of the three interviewees. However, many problems of the current GUI were revealed very fast. The disadvantages of the TGtool were formulated by the interviewees as follows:

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. Than the tool works. 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.

Table 2 - Pros and Contras of TGtool

3.4. Ideas of a user-centered GUI

Powered by the article “Usability evaluation considered harmful (some of the time)” by Greenberg & Buxton (2008), where the authors argue that application of usability evaluation methods should happen only after idea-generating phase, no evaluation techniques were applied in this phase.

Idea-generation for the future user-centered GUI has resulted in three proposals:

- Tab tool
- Icon-based menu
- Navigation – left, stacked widget – right

These ideas were implemented first as paper prototypes, and further low-fidelity prototypes in XRCed[7] and Qt4 have been offered for discussion (Memmel et al, 2007).

[7] http://xrced.sourceforge.net/

3.4.1. Idea 1 – Tab tool

The first GUI draft was developed in XRCed – simple resource editor for wxWidgets / wxPython GUI development which supports creating files in XRC format.

The first design was more suitable for go-live configuration, as it allowed working with different functionality, and changing tabs very quickly, but still leaded the users logically through the configuration process. It is a tab-based application.

Below you can see the low-fidelity prototype of this idea.

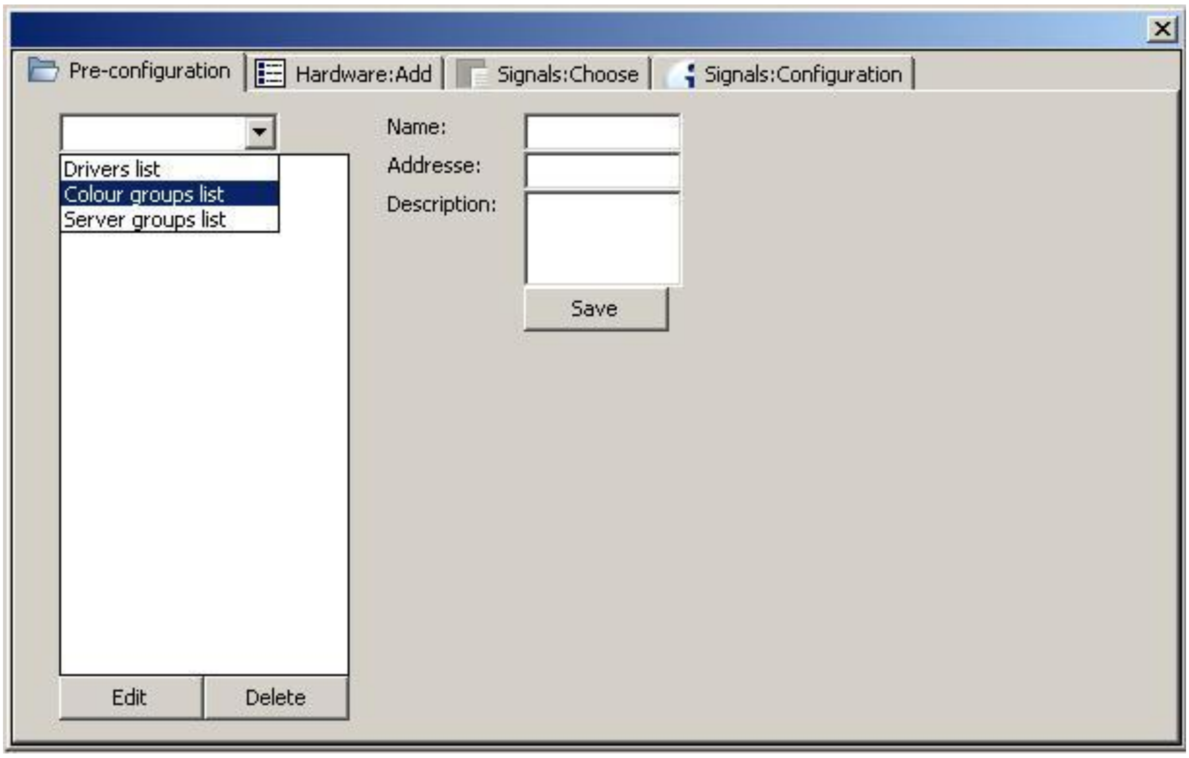


Figure 20 - Pre-configuration tab

Following the tab order, users can easily execute various pre-configurations and continue with hardware configuration on the second tab.

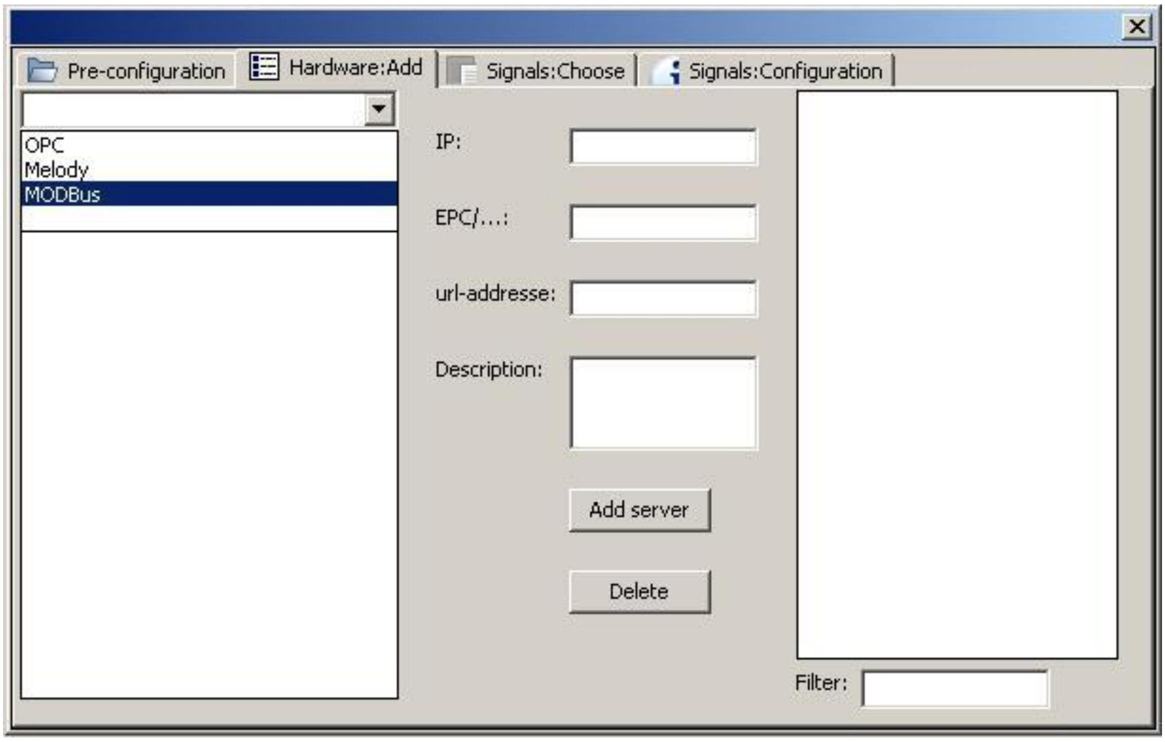


Figure 21 - Hardware management tab

Then users move to the third tab in order to choose and add preferable signals to the list from any configured hardware (server, CMC, workstation etc).

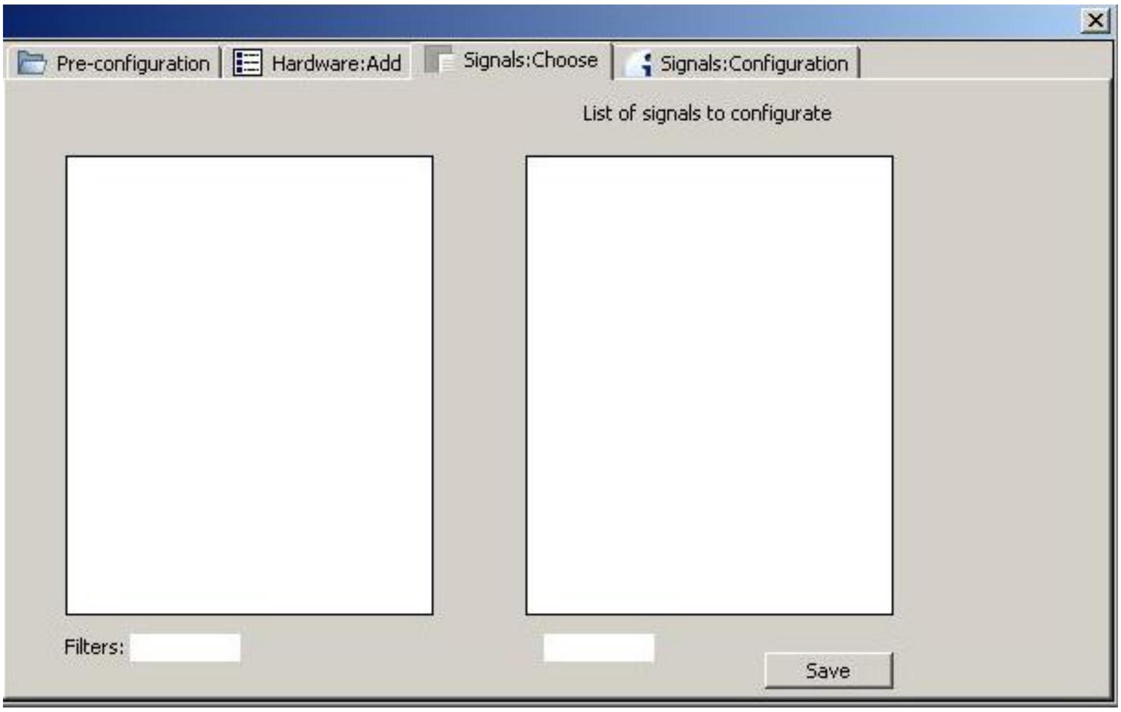


Figure 22 - Signal selection tab

The forth and the last tab allows configuring signals' properties as well as exporting and importing CSV lists of signals.

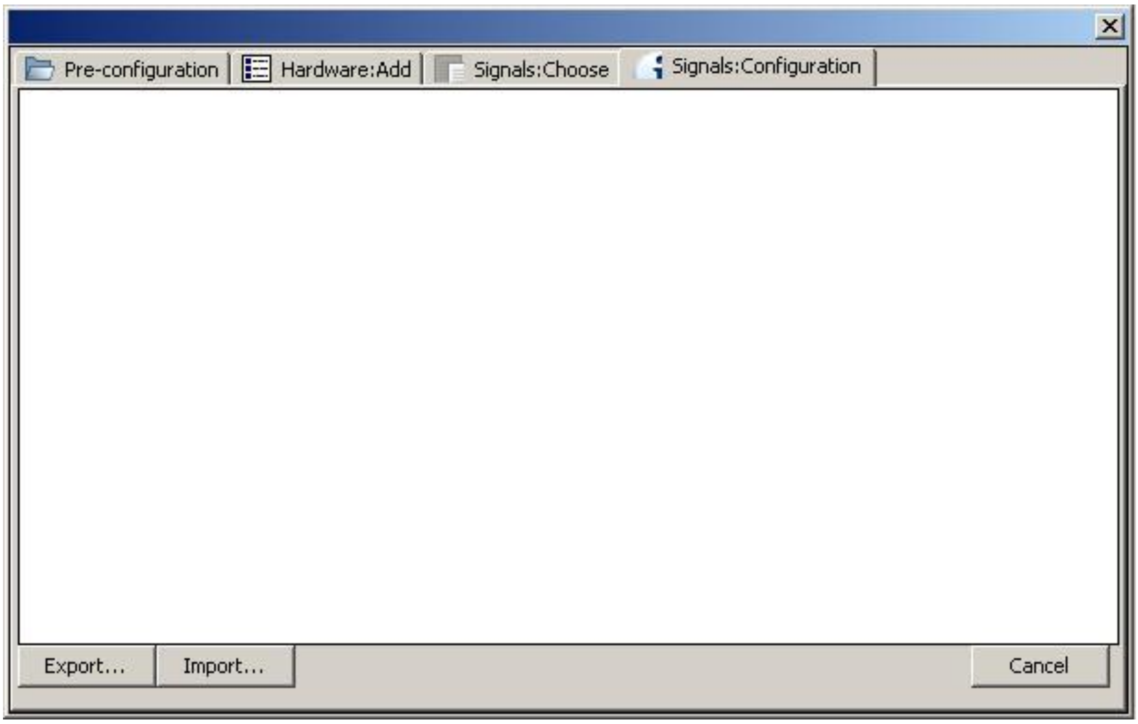


Figure 23 - Signal configuration tab

In this low-fidelity prototype only a small part of all required functionality is demonstrated. This approach has following pros and contras:

Advantages	Disadvantages
Tab concept has a guiding line.	In the high-fidelity prototype there are many more configurations.
It is easy to switch between tabs.	Too many tabs are required to cover all configurations.
Users always know how far they are.	Tool will become confusing with dozens of tabs.

Table 3 - Pros and Contras of a tab-tool

3.4.2. Idea 2 – Icon-based main menu page

The low-fidelity GUI prototype was developed in Qt4 Designer, which had an opportunity to compile C++ code into Python.

This design looked much like the TGtool, if it were to have a main window with the configuration tasks visually grouped together. However, there would not be categories in the main window but buttons which would open functional windows (see Figure 25). Moreover, those buttons would possess icons instead of text. The icons were thought to be created in tight cooperation with end users. The icon-based menu developed by Karahoca et al. (2010) for MEDSI (Mobile Emergency Department Software Iconic) (see Figure 24) would be an example to follow.

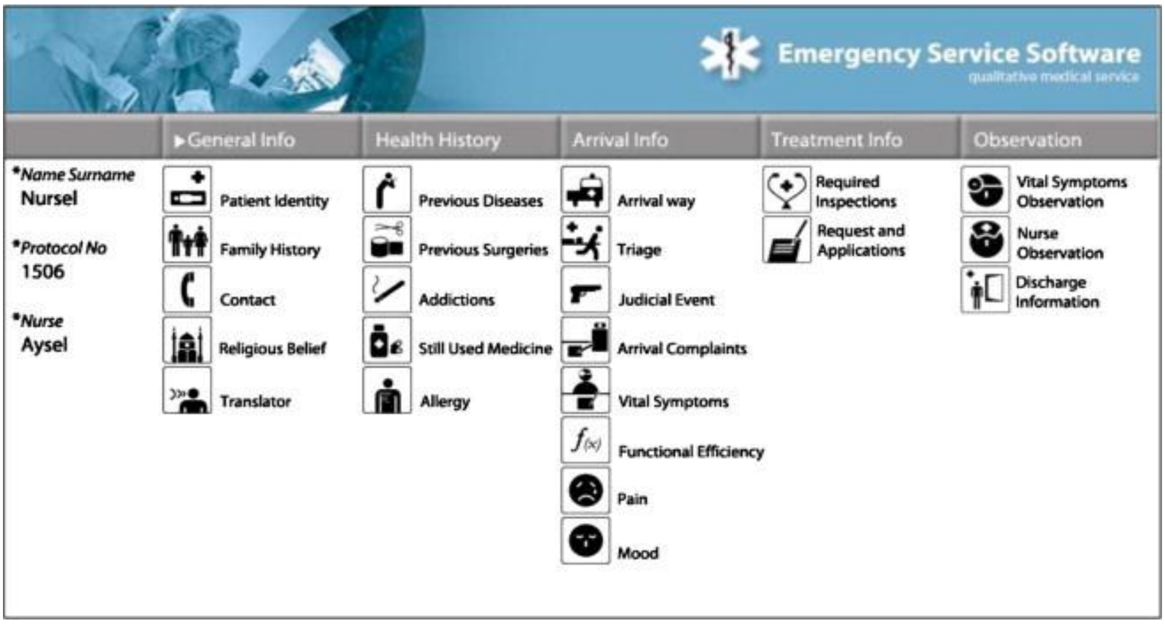


Figure 24 - Menu system for MEDSI (Karahoca et al., 2010)

An icon-based GUI was supposed to support all possible configurations, even the wizard-like ones. From its main window users could choose a configuration module (a concrete window with a specified task - see Figure 25) or wizards (which consists of the same configuration windows, which appear in a pre-specified sequence). This kind of wizard should make the configuration process easier and faster.

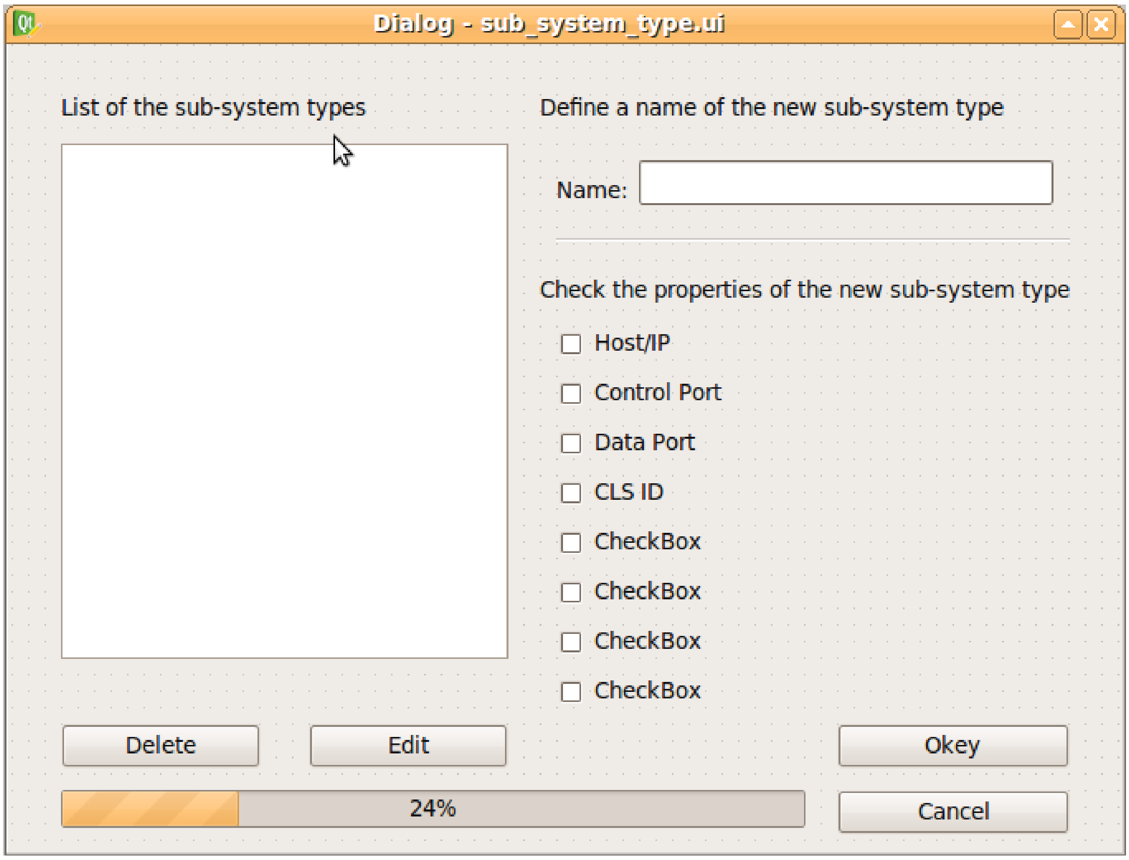


Figure 25 - Functional window of iconic tool

Above an example of a possible functional window is presented. This kind of window will appear after clicking an icon in the main menu.

Here are the main pros and contras for the GUI with an icon-based main menu.

Advantages	Disadvantages
Flexible to add or delete an icon.	It is uncomfortable to come back to main menu each time users need another configuration window.
The users are well acquainted with this kind of navigation.	The interface itself does not provide guiding elements.
Many icons can be added to the configuration menu.	The main menu becomes excessive when too much functionality is provided - too many icons.
The users have an overview of all the icons at one glance.	For too many functions it will be hard to create a good icon for each of them.

Table 4 - Pros and Contras of iconic main menu page

3.4.3. Idea 3 – Navigation – left, stacked widget – right

The low-fidelity prototype was developed in Qt4 Designer which had an opportunity to compile C++ code into Python.

The main point of this GUI was to combine quick navigation through the functional windows and still allow a big amount of configurations to be sufficiently added. This idea was supposed to become a compromise between the previous two ideas: use their advantages and eliminate their disadvantages.

The navigation tree on the left side should always be visible, so that users could switch between stacked widgets placed on the right side quickly. It is reminiscent of a typical web page view, thus it should look familiar to the end users. Moreover, a navigation tree allows collapsing unnecessary branches and expanding the needed ones, thus the space of the navigation menu could be used more efficiently and even a big amount of leaves would not confuse users.

It is also important to mention that this type of visualization leaves freedom for further development. A stacked widget element can have as many pages as necessary, thus a new page can be added anytime with minimal effort.

In the Figure 26 a low-fidelity interface can be seen with an empty navigation menu on the left and some configuration widget on the right.

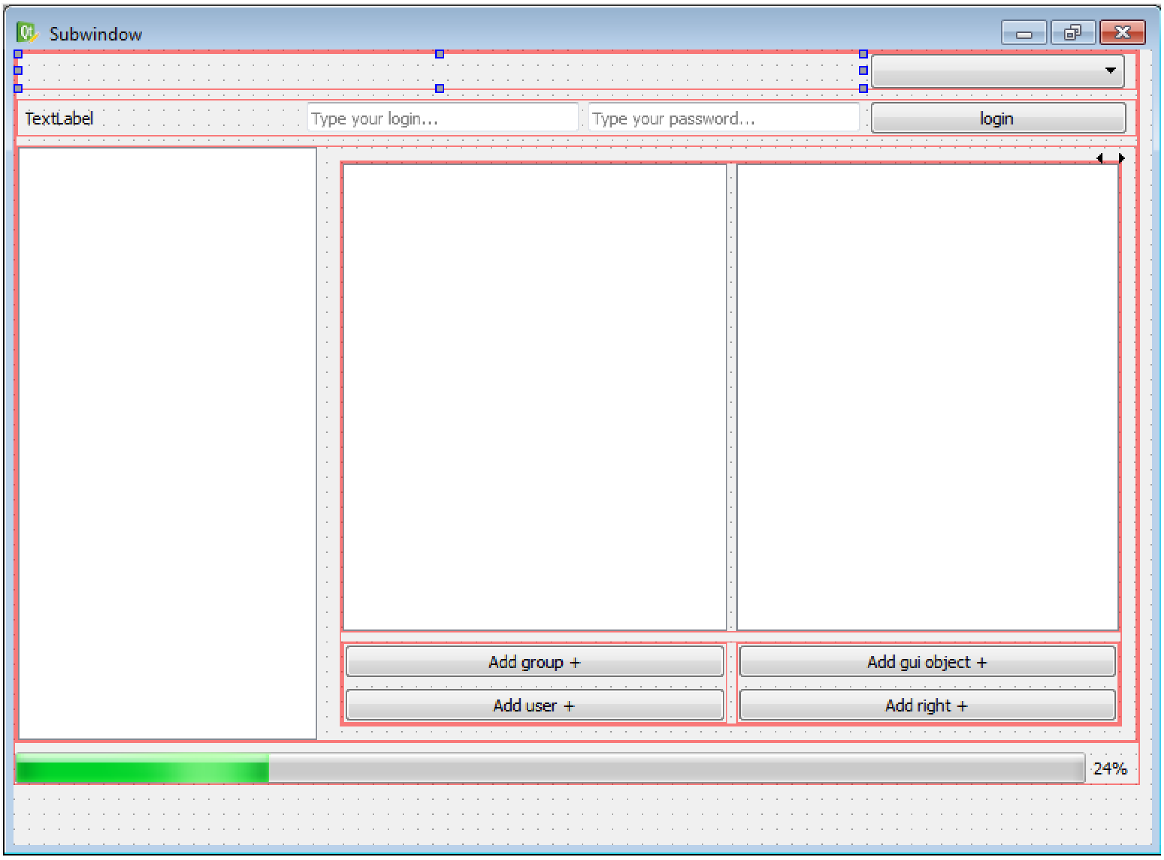


Figure 26 - Navigation left, stacked widget right

Here is the Table 5 with main pros and contras which shows the reached compromise.

Advantages	Disadvantages
Flexible in adding new functionality.	Difficult to implement.
Many leaves can be added to the navigation tree.	
The users are well acquainted with this kind of navigation.	
The view of the navigation tree can be managed.	
It is easy to switch between widgets.	
Navigation can also be organized as a step-by-step wizard.	

Table 5 - Pros and Contras of navigation plus stacked widget GUI

3.5. High-fidelity prototype development

After the second interview session the third option with a navigation tree was agreed to be the most suitable and user-oriented. As the development of the high-fidelity prototype was started, further interviews have been arranged in order to keep the development process under end user control.

A problem of adding and editing data appeared at the first development phase, where two opportunities have been proposed.

- Pop-up windows with add/edit functionality (see Figure 27)

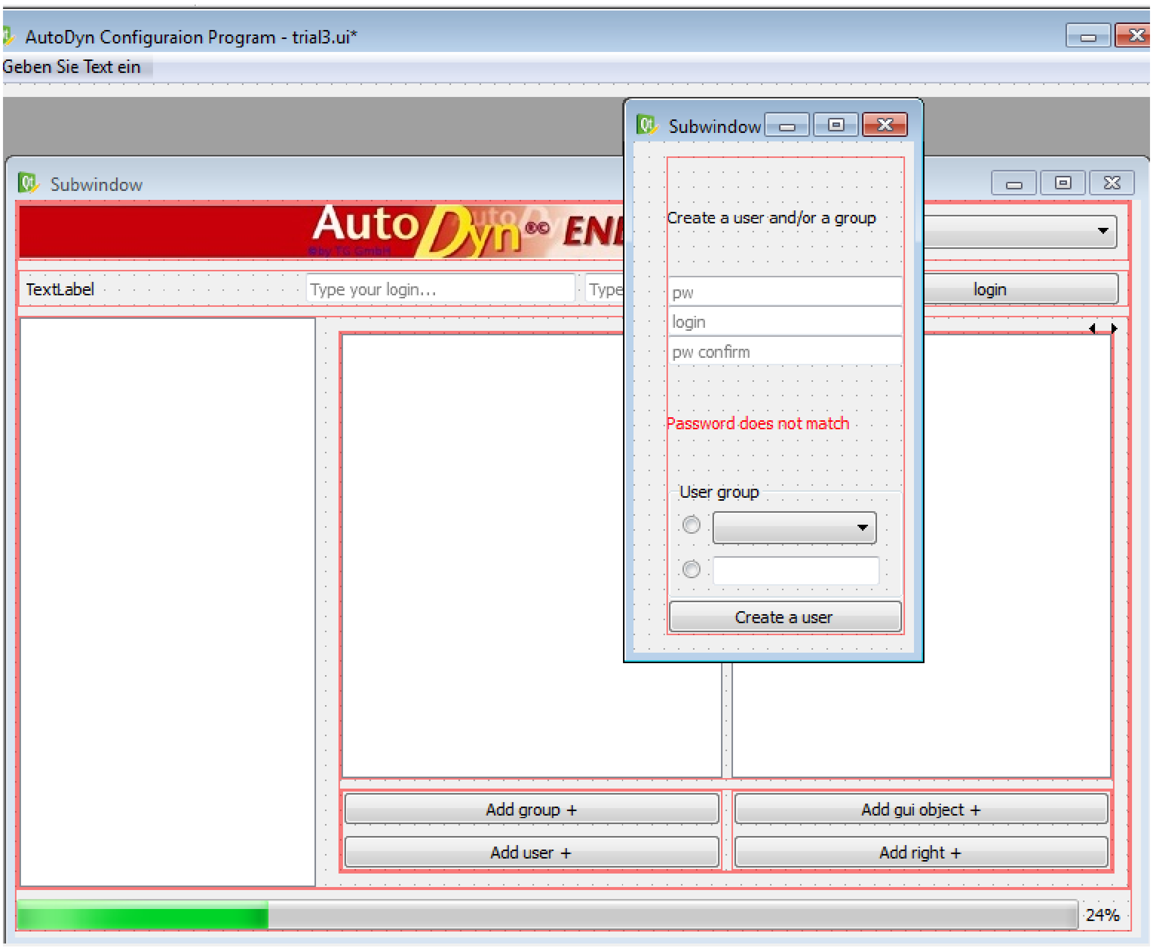


Figure 27 - Pop-up

The disadvantage was that for each data addition the users had to click for a pop-up. It could become annoying.

- Dock widgets on the right with add/edit/delete functionality (see Figure 28)

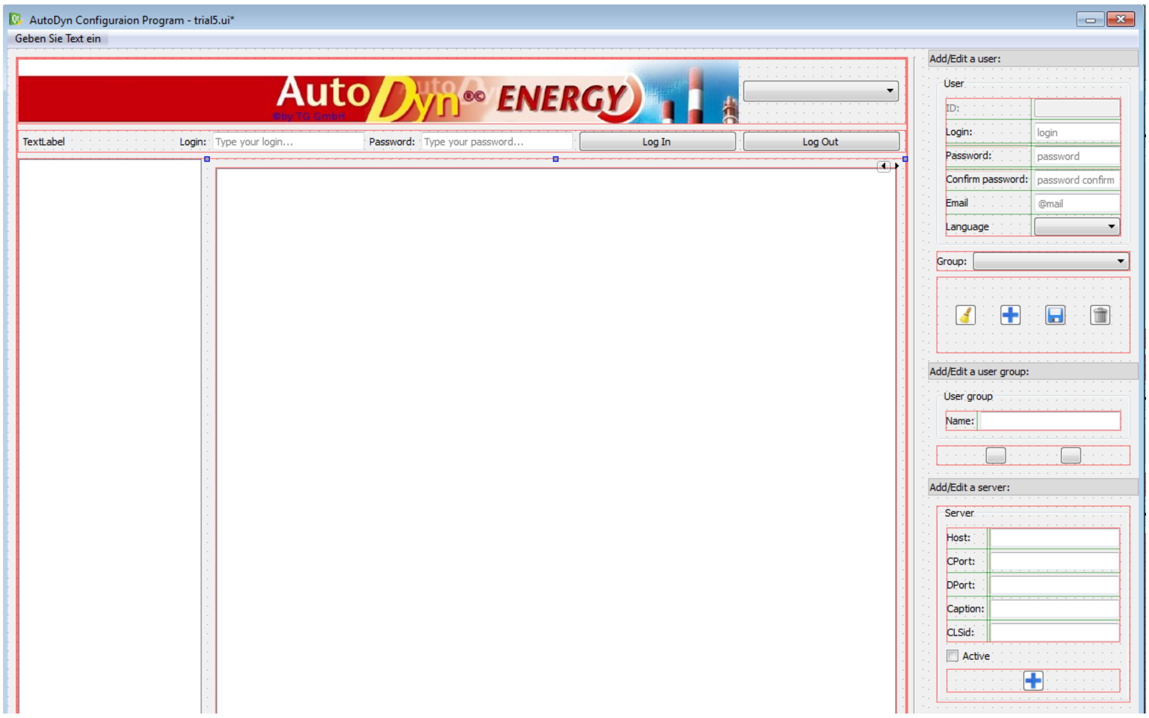


Figure 28 - Dock widgets on the right

This GUI is quite the same except that instead of a pop-up, the required filling forms appear on the right side of a working space. They also change in accordance to the chosen menu in the navigation tree. It is still not perfect as some menus need so many filling forms that there is too little space for them all. However, dock widgets can be closed and opened again if needed.

The second refinement option has been agreed to be the most comfortable. Interviewees also proposed to use pop-up messages for warnings or error communication (see Figure 29).

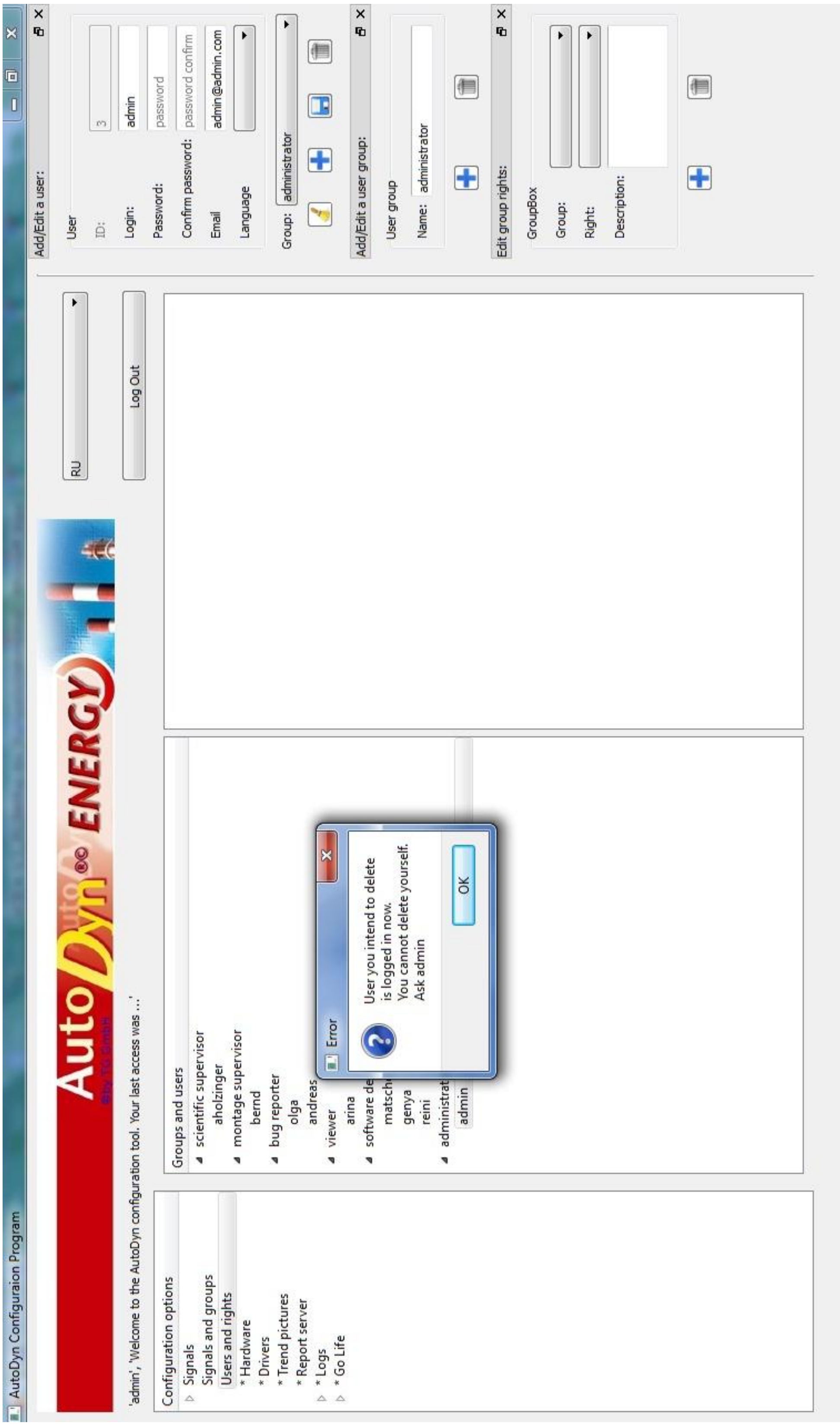


Figure 29 - Error communication

Further we came to the conclusion that if there were two types of tasks (simple and complex) and setup configurations were very sensitive, they had to be protected from the unqualified users. The safety-critical area of application dictates a necessity of user management. It is important that each user is able to access, view and edit only the data he or she is permitted to access, view and edit. The user rights management had to take care of this requirement. User rights management would also be responsible for adjusting visual complexity for each type of users. E.g. domain experts executing setup configurations on site require more functionality and more information provided by the GUI, whereas a technician on the client's side without knowledge of the Technikgruppe system will be satisfied to see the only function he or she needs for adding signals. This way unqualified user will not be confused by all the available functionality and thus the probability of human error decreases.

3.5.1. Database structure

In order to satisfy the flexibility requirement of Technikgruppe an optimized database scheme was developed (see Figure 30). Its flexible structure allows working with different controlling systems without reconstructing the database.

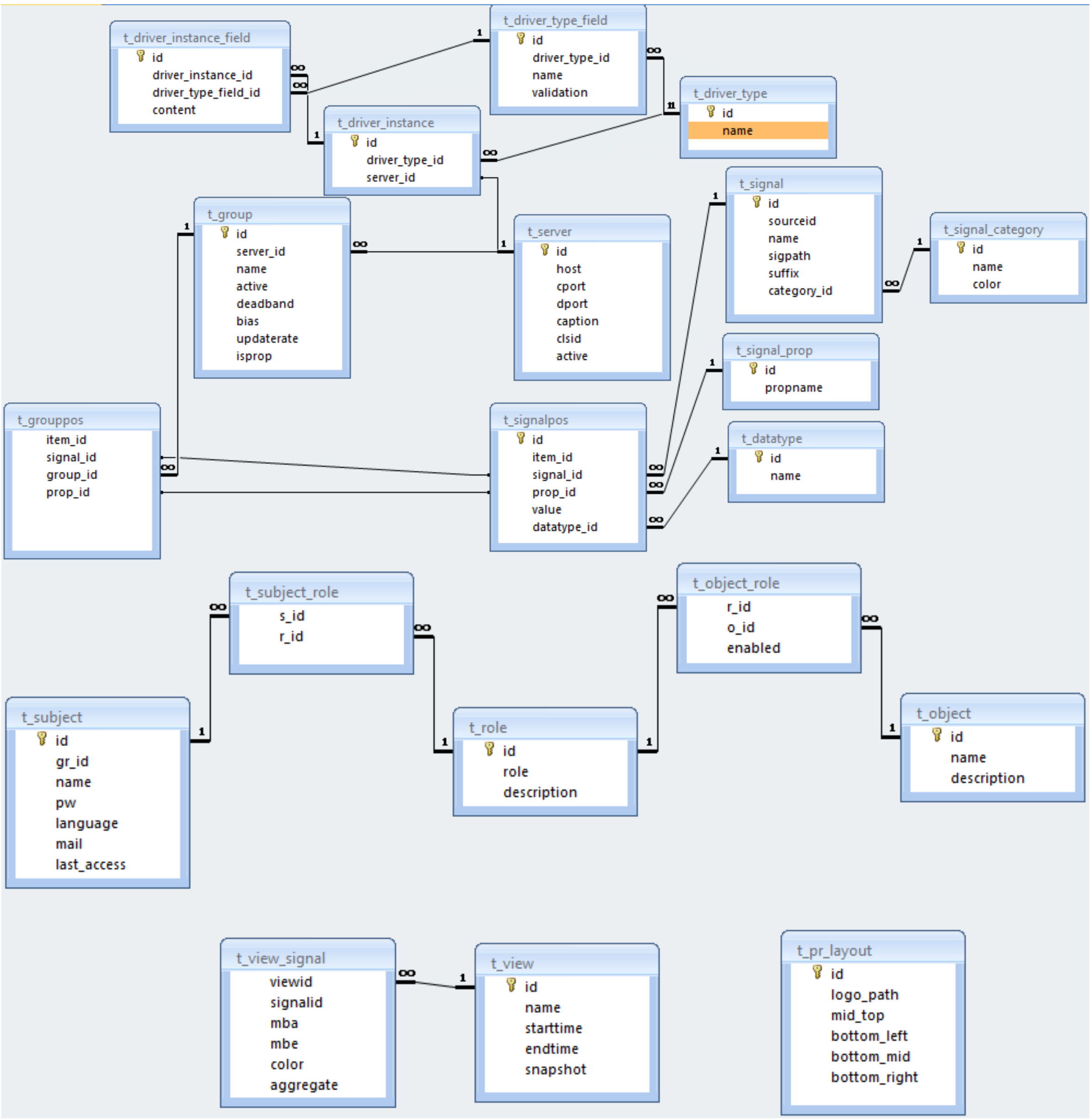


Figure 30 - Database structure

For example, in one controlling system all the signals have the same properties, but in another system all real and binary signals have the same properties among their respective groups, but the two groups are different. In a third system any signal may have individual properties. The developed database structure works with any of these controlling systems. However, this flexibility causes more complex database procedures, which require more computing and consequently, time resources. In any case, this system complexity is thoroughly hidden from the end user.

3.5.2. Code structure

The code structure below (Figure 31) demonstrates the general approach in development. The number in brackets means the number of code lines.

A green color stresses the main execution path. The program starts in **start.py**, where an instance of the **gui** class from **gui.py** is created, which in its turn creates instances of required models and tables for data visualization in graphical elements.

The **gui.py** only handles the graphical elements, whereas the graphical interface itself is inherited from the **gui_unchangable** class (emphasized with red). All the graphical elements are being imported in **gui_unchangable.py** from Qt Designer. This means that the application stays flexible. A developer can add some widgets in Qt Designer anytime, in this case the **gui_unchangable.py** will have to be generated again and the changes will be automatically displayed when the program is running.

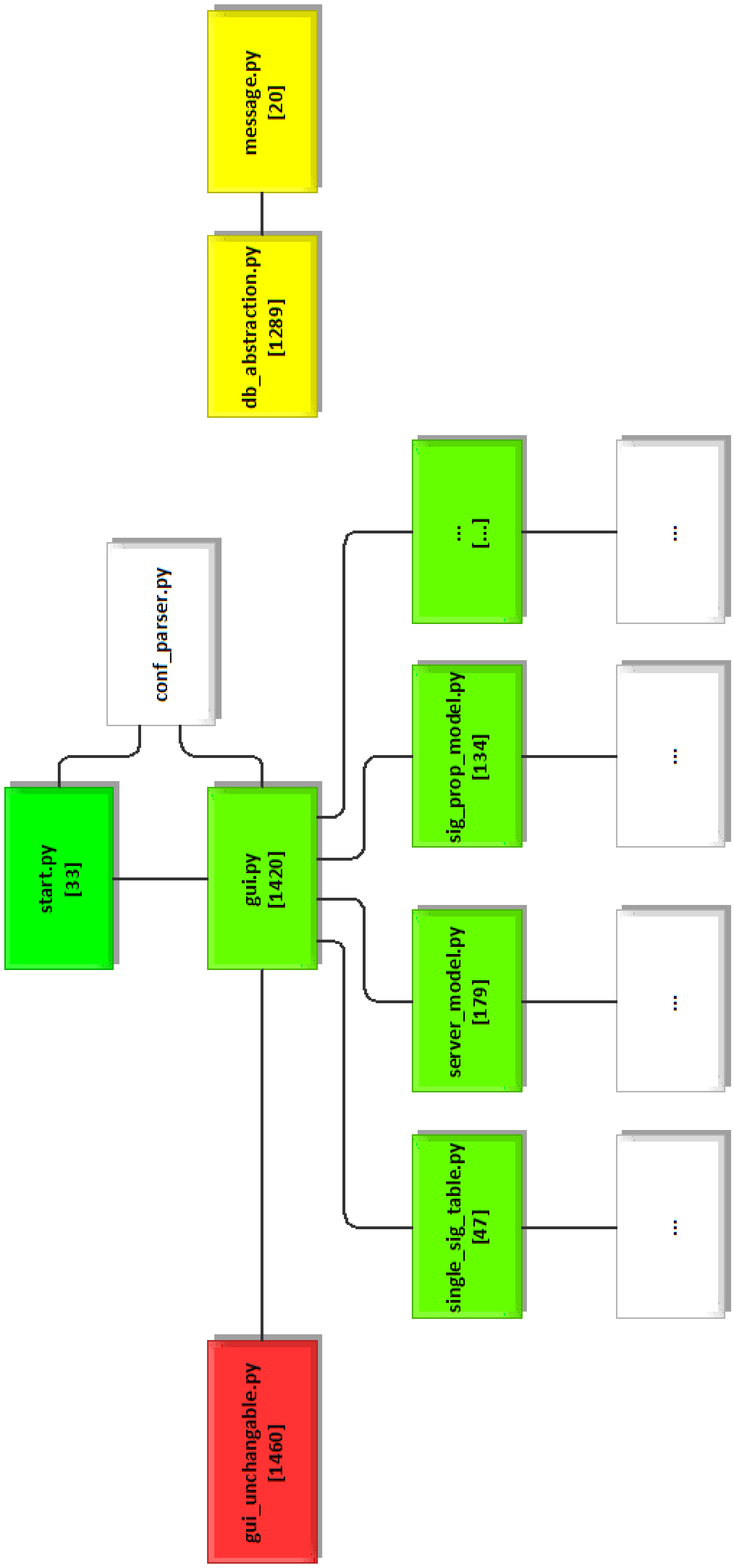


Figure 31 - Code structure

Technikgruppe is expected to use this application for a long period of time and even if the database will need some optimization later it will not cause an enormous financial investment as the code has its own database abstraction layer **db_abstraction.py**. This is the only class which interacts with the database and it is the only element which has to be updated in case of database optimization. This class is displayed in yellow. All the other classes refer to it if they need data from the database. Also **message.py** represents a service class which is required by all the other classes in order to throw a pop-up message when something goes wrong or a user makes an error.

Conf_parser.py contains the default configurations required for correct program function.

The code of over five thousand lines includes the GUI interface, the DB interface and takes care of all the human-computer interaction and internal computations. However, with increasing functionality this amount will grow.

3.6. Evaluation

This part describes the usability tests which were applied in order to test the prototype usability and to compare it with the usability of TGtool.

3.6.1. Thinking Aloud Test

The methodology of thinking aloud (TA) was developed on the basis of the HCI course materials[8] (Andrews, 2012).

3.6.1.1. Preparations

Six simple tasks were prepared for the TA (see Table 6) as well as a test environment (see Table 7). The videos were recorded with a Canon Exilim 12.1 mega pixels camera and Debut Video Capture Software[9] was used for registering user's activity on the screen.

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 6 - Tasks for TA

[8] http://courses.iicm.tugraz.at/hci/practicals/materials/en/taplan/

[9] http://debut-video-capture-software.softonic.de/

Environment	Details
Room	Standard working place (individual for each test user)
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

Table 7 – Hard- and software environment

Three users (one pilot and two test users) had to accomplish prepared tasks using the configuration tool prototype. Due to the Technikgruppe's requirement, only real end users of the tool could take part in the testing process, in particular the Technikgruppe employees, who are working as:

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

Therefore the typical profile of the average test user (see Figure 32) 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 in Windows.

This average user needs to process different routine configurations in order to update decision supporting AutoDyn quite often, 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.

Figure 32 - Thinking aloud test - test user

3.6.1.2. 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.

3.6.2. Mouse Click Measurement

Buchanan (2001) evaluated the usability of mobile applications analyzing the amount of key presses. In order to compare the existing configuration tool TGtool and the new prototype an objective metric – amount of mouse clicks – was chosen. For this purpose the six tasks from **Fehler! Verweisquelle konnte nicht gefunden werden.** were used, where each of them was performed in the most efficient way[10] in both applications and the amount of mouse clicks was recorded.

It is important to mention that the mouse clicks necessary for filling out the form fields have been discarded. Only navigating clicks, selecting clicks, filtering clicks and pushing the button clicks have been considered. The starting point of each task in both tools was the main access point – main menu.

3.6.3. Heuristic Evaluation

Afterwards heuristic evaluation of both the TGtool and the improved prototype was completed by two usability examiners using heuristics from Nielsen (1994) and the five-point severity scale from Pierotti[11].

The results are described in the next chapter.

---

[10] The most efficient way in this case is the path with the minimal amount of mouse clicks.

[11] http://www.stcsig.org/usability/topics/articles/he-activities.html

4. Results

4.1. 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 (see Figure 33):

- 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

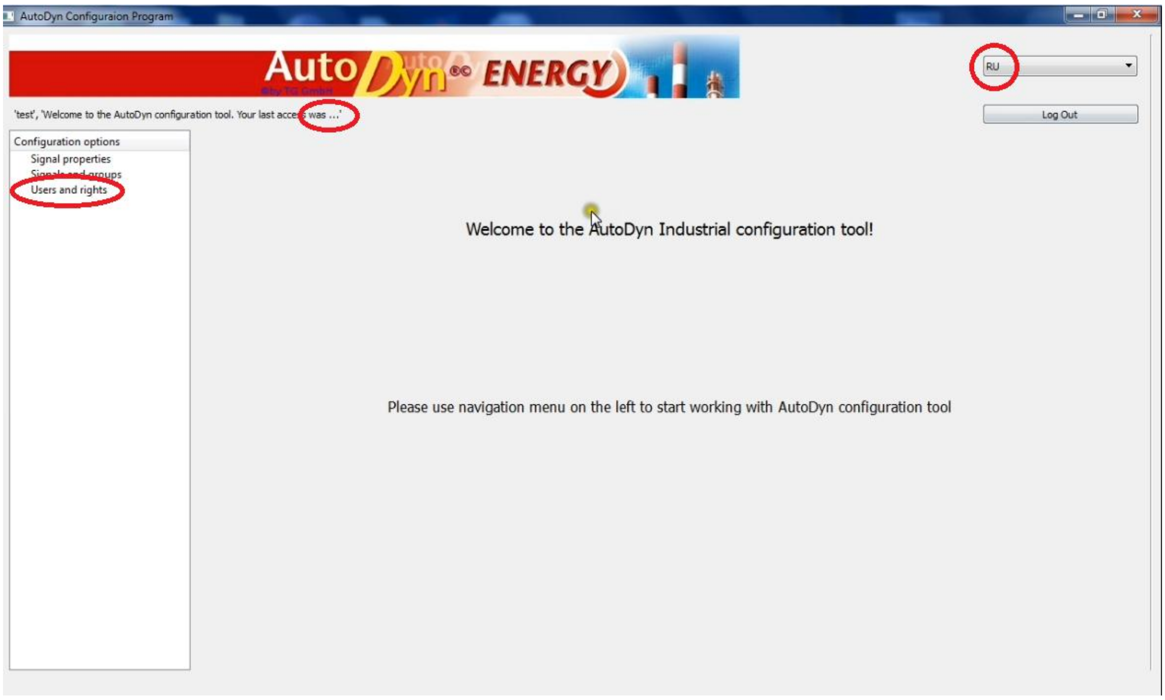


Figure 33 - Prototype - Minor problems in the main menu

Other usability "bugs" have been discovered during the test like buttons not working, absence of a sand clock 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 (see Figure 34). Several minor usability 'bugs" were detected as well.

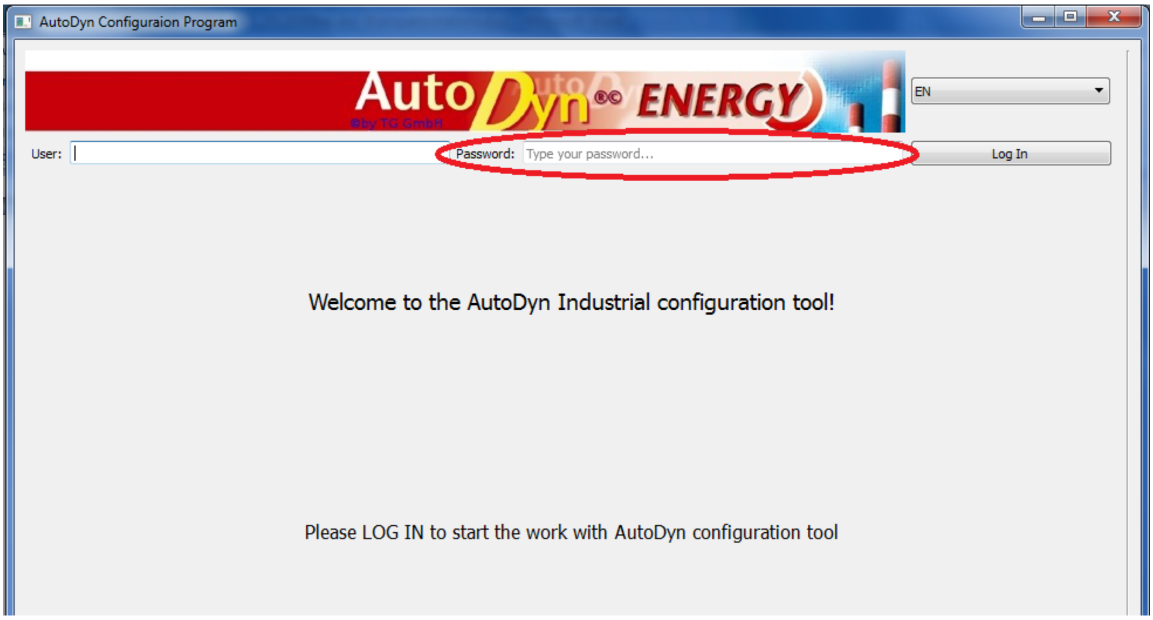


Figure 34 - Password field on the right

In general users have successfully accomplished all the tasks without any training or help, relying only on their cognition and domain knowledge. The diagram on Figure 35 shows time curves (minutes per task[12]) for each test person (TP). On the horizontal axis the task number and its complexity grade and on the vertical axis the time in minutes is shown.

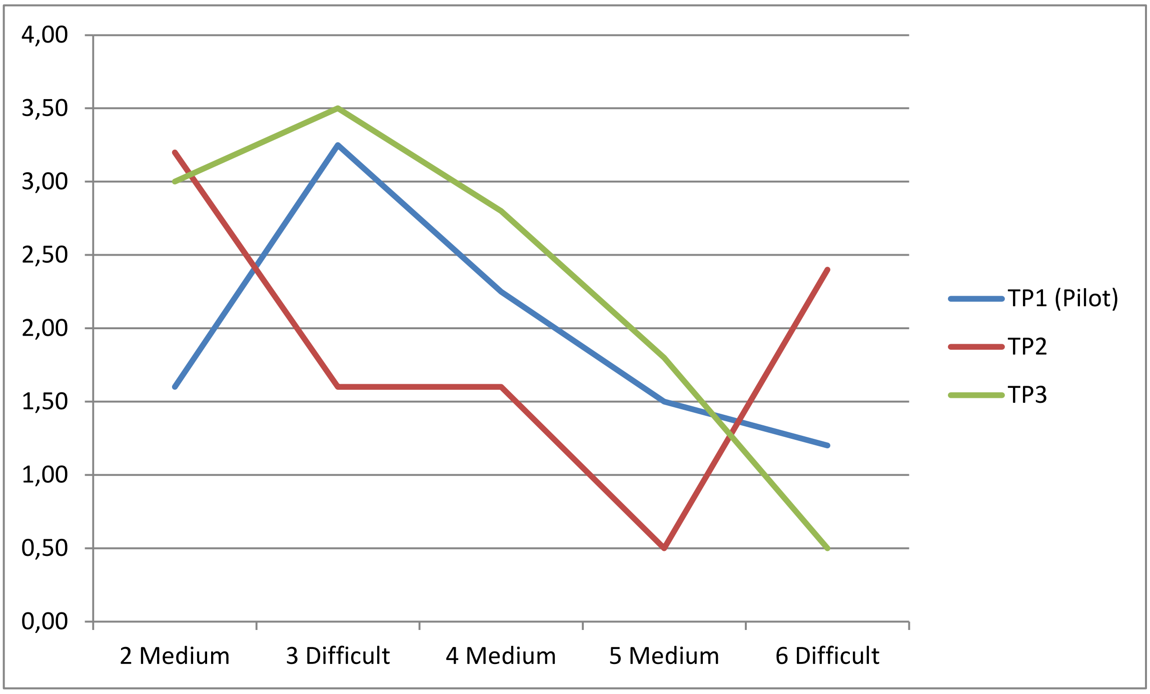


Figure 35 - Minutes per task

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

[12] 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.

individuality of users' cognition process and too few TPs. A higher amount of test users would provide more representative results.

In TA sessions after completion of all the tasks the users have had to fill out a feedback questionnaire. On the Figure 36 its results are presented.

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.

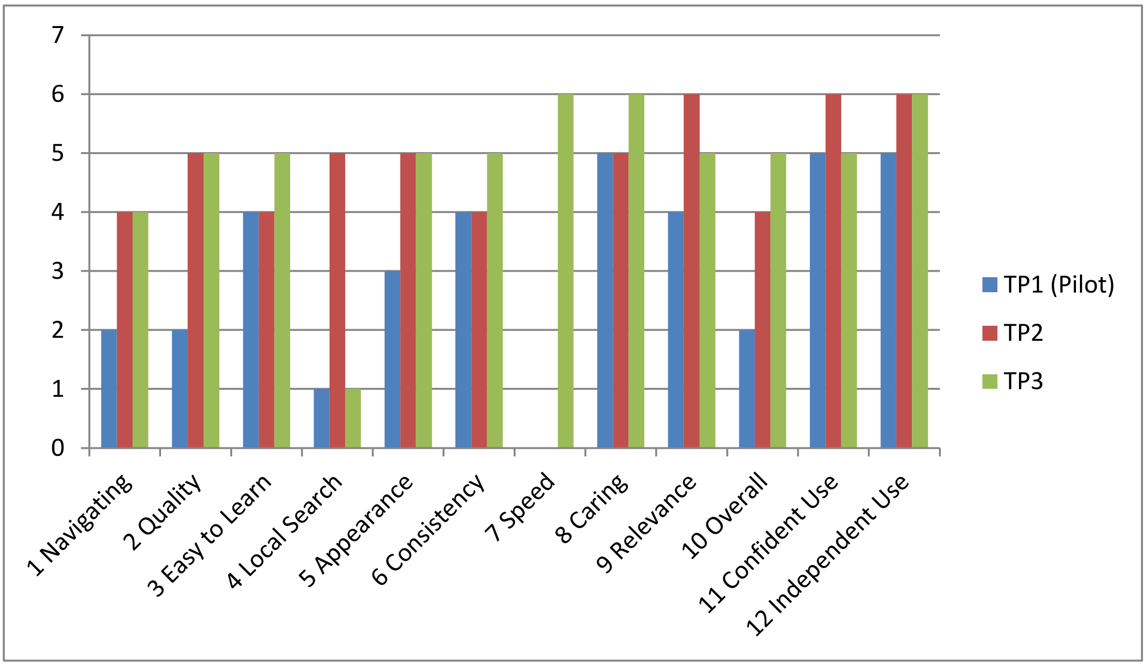


Figure 36 - Feedback questionnaire result

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

In order to eliminate the speed problem another hardware test environment (see Table 8) has been used for the last test.

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

Table 8 - Hardware environment for the last TA

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.

4.2. 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[13] have been used in this experiment as in the TA. In the Figure 37 it can be seen that at any task processing less clicks have to be done in prototype.

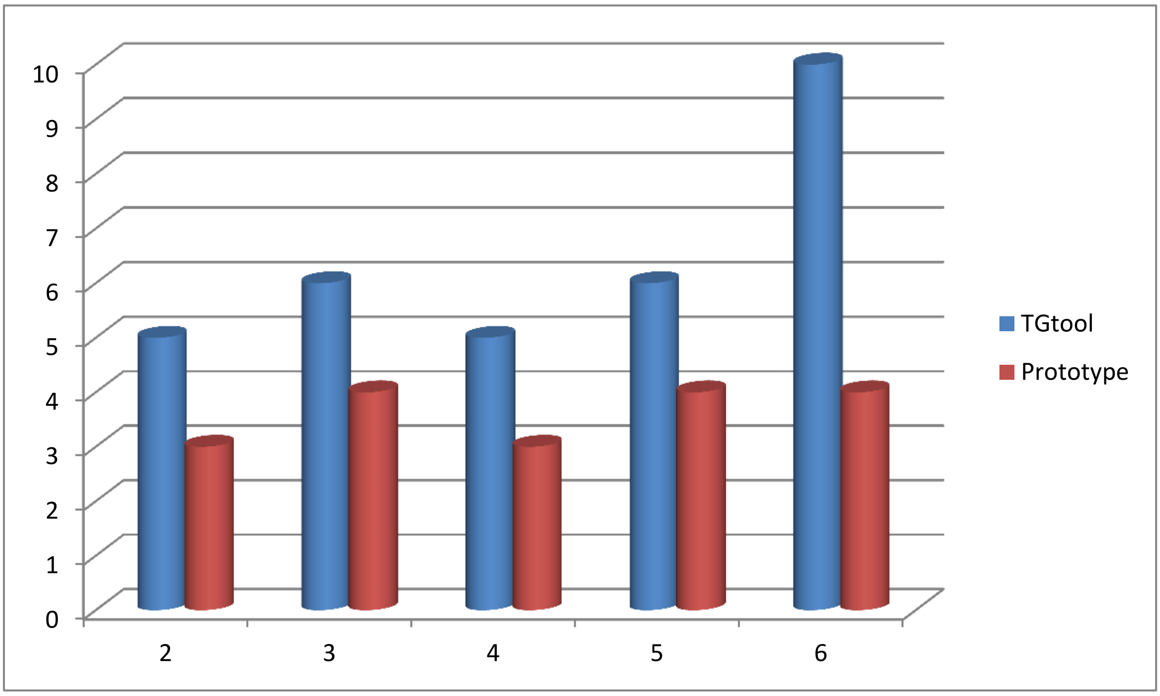


Figure 37 - Minimal amount of clicks per task

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

[13] The first looking-around task has not been considered as there is no a clear criteria of success.

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.

4.3. Heuristic evaluation

Heuristic evaluation has been performed by two heuristic experts with domain knowledge[14]. They have evaluated the TGtool and the prototype on the basis of ten heuristics from Nielsen (1994). The five-point rating scale from Pierotti, which is attached in the Appendix G, has been applied. In this scale 1 point refers to cosmetic problems and 5 to catastrophic ones. On the network diagram (Figure 38) the results[15] can be seen.

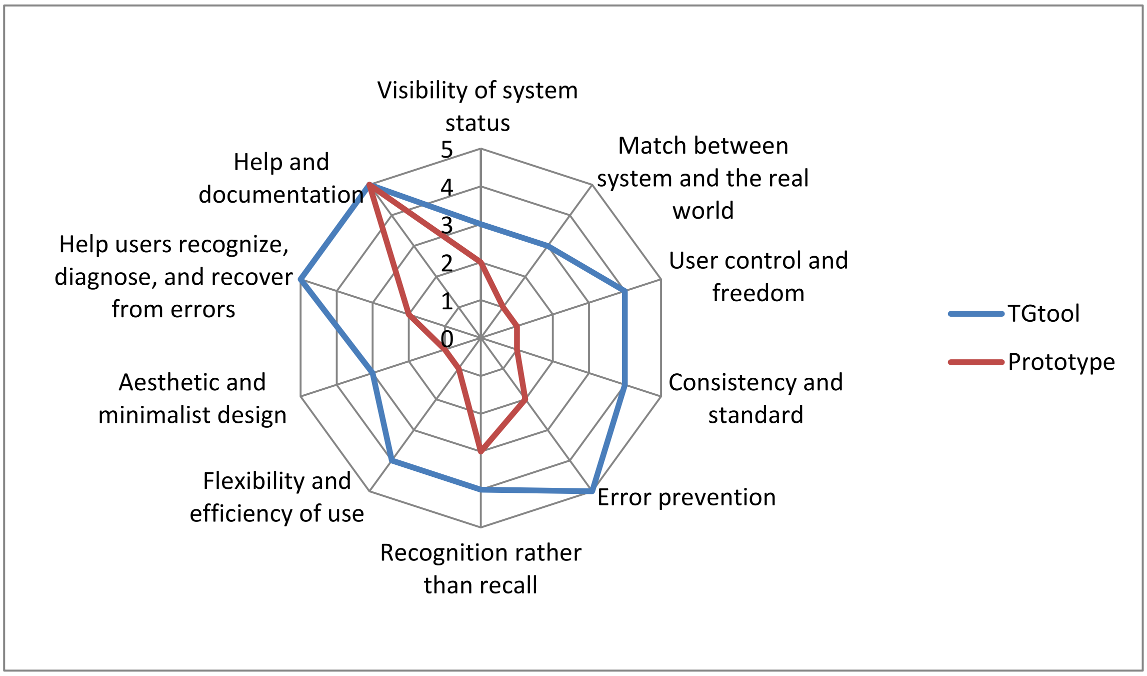


Figure 38 - Heuristic evaluation results

---

[14] The heuristic experts with domain knowledge were represented by the employees of Technikgruppe, who have graduated from the Graz University of Technology and during their study took part in HCI seminars.

[15] The average of the results of two independent heuristic evaluations has been used here.

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. Discussion

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 (1994) 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.

6. Conclusion

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 the thesis. Consequently, the hypothesis holds.

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, etc) 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.

7. Future Work

In this thesis the prototype of an industrial configuration tool with maintenance functionality was designed, developed and evaluated. In future work this prototype will be extended and the most complex functionality – setup configurations – will be added. This upgraded tool will be installed on the simulating system of Technikgruppe with real hardware elements sending signal data in real time. On the simulation platform the configurations will affect real system components and will be retraceable. In this context the testing of the system with TA will be more exciting as the users will have more responsibility and will have more complex tasks to perform. Also, the emotional state of the users will be analyzed on the basis of psychophysiological measurements like heart rate, skin conductance level, etc. The further research upon complexity and its combination with the stress factor is supposed to provide interesting insights.

8. List of Figures

9. List of Tables

10. References

AFACAN, Y. & ERBUG, C. (2009). An interdisciplinary heuristic evaluation method for universal building design. (ELSEVIER, Ed.) Applied Ergonomics (40), pp. 731-744.

ALSMADI, I. & AL-KABI, M. (2011) GUI Structural Metrics. The International Arab Journal of Information Technology, Vol. 8, No. 2, April 2011

ALSMADI, I. & AL-KABI, M. (2009) The Introduction of Several User Interface Structural Metrics to Make Test Automation More Effective. The Open Software Engineering Journal 3, 72-77

ALSMADI, I. & MAGEL, K. (2007) GUI path oriented test generation algorithms. In Proceeding of Human-Computer Interaction Conference. IASTED HCI, 2007

ALSMADI, I. & MAGEL, K. (2007) GUI path oriented test case generation. In Proceeding of the International Conference on Software Engineering Theory and Practice (SETP07), 2007.

ALSMADI, I. & MAGEL, K. (2007) GUI Structural Metrics and Testability Testing, IASTED SEA 2007.

ANDERSON, N.S. & LEONARD, J.A. (1958). The recognition, naming and reconstruction of visual figures as a function of contour redundancy. Journal of Experimental Psychology, 56(3), 262 – 270.

ANDREWS, K (2012) HCI Course: Thinking Aloud Plan [Online]. Available: http://courses.iicm.tugraz.at/hci/practicals/materials/en/taplan/ [Accessed 2012-05-08]

APTER, M. J. (1982) Experience of Motivation: The Theory of Psychological Reversals, New York: Academic Press.

APTER, M. J. (1995) "On Metamotivational Analysis," in Proceedings of the 7th International Conference on Reversal Theory, Melbourne, Australia, pp. 19-19.

APTER, M. J. (2001) "An Introduction to Reversal Theory," in Motivational Styles in Everyday Life: A Guide to Reversal Theory, M. J. Apter (eds.), Washington, DC: American Psychological Association, pp. 3-35.

APTER, M. J. (2003) "On a Certain Blindness in Modern Psychology," The Psychologist (16:9), pp. 474-475.

ARNHEIM, R. (1966) "Order and Complexity in Landscape Design," in Toward a Psychology of Art, R. Arnheim (ed.), Berkeley, CA: University of California Press, pp. 123-135.

BARRETT, L. F. (1998) "Discrete Emotions or Dimensions? The Role of Valence Focus and Arousal Focus," Cognition and Emotion (12), pp. 579-599.

BERLYNE, D. E. (1971) Aesthetics and Psychobiology, New York: Appleton-Century-Crofts.

BIRKHOFF, G. D. (1933) Aesthetic Measure, Cambridge, MA: Harvard University Press.

BENNETT, K.B., TOMS, M.L. & WOODS, D.D. (1993). Emergent features and graphical elements: Designing more effective configural displays. Human Factors, 35(1), 71 – 97.

BROOKE, J. (1996) SUS: a "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland. Usability Evaluation in Industry. London: Taylor and Francis.

BUCHANAN, G., JONES, M., THIMBLEBY, H., FARRANT, S. & PAZZANI, M. (2001) Improving mobile internet usability. Proceedings of the Web 2001 Conference, Hong Kong, ACM Press (2001)

BUCY, E. P. (2000) ''Emotional and Evaluative Consequences of Inappropriate Leader Displays,'' Communication Research (27:2), pp. 194-226.

BURNS, C.M. (2000) Putting it all together: Improving display integration in ecological displays. Human Factors, 42(2), 226 – 241.

BUXTON, B. (2007) Sketching User Experiences: Getting the Design Right and the Right Design. Morgan Kaufmann.

CARSWELL, C.M. (1992) Choosing specifiers: An evaluation of the basic tasks model of graphical perception. Human Factors, 34(5), 535 – 554.

CARSWELL, C.M. & WICKENS, C.D. (1996). Mixing and matching lower-level codes for object displays: evidence for two sources of proximity compatibility. Human Factors, 38 (1), 1 – 22.

COSKUN, E. & GRABOWSKI, M. (2004) Impacts of User Interface Complexity on User Acceptance in Safety-Critical Systems. Proceedings of the Tenth Americas Conference on Information Systems, New York, August 2004.

COOK, R.I., WOODS, D.D., HOWIE, M.B., HORROW, J.C. & GABA, D.M. (1992) Unintentional delivery of vasoactive drugs with an electromechanical infusion device. Journal of Cardiothoracic and Vascular Anesthesia 6 (2), 238-244.

Debut Video Capture Software [Online]. Available: http://debut-video-capture-software.softonic.de/ [Accessed 2012-05-08]

DENG, L. & POOLE, M. S. (2010) Affect in web interfaces: a study of the impacts of web page visual complexity and order. MIS Quarterly Vol. 34 No. 4 pp. 711-730/December 2010

DONDERI, D. (2006) An information theory analysis of visual complexity and dissimilarity. Perception 35, 823-835

DONDERI, D. & MACFADDEN, S. (2005) Compressed file length predicts search time and errors on visual displays. Displays 26, 71-78.

DONDERI, D. (2003) Visual Complexity: A Review. HUMAN FACTORS NORTH INC 2003.

D'PUIS [Online]. Available: http://www.softreliability.org/dpuis [Accessed 2012-05-09]

FITTS, P.M., WEINSTEIN, M., RAPPAPORT, M., ANDERSON, N. & LEONARD, J.A. (1956) Stimulus correlates of visual pattern perception: a probability approach. Journal of Experimental Psychology, 51, 1 – 11.

FLEISHER, R. B. & BULLINGER, T. A. (1998) Star Plots of analyte values over time. EP0877249. G01N33/48. ORTHO CLINICAL DIAGNOSTICS INC. Christopher Paul Mercer, Carpmaels & Ransford 43, Bloomsbury Square London WC1A2RA

FRIJDA, N. H. (1986) The Emotions, New York: Cambridge University Press.

FUNK, M. & HAMACHER, N. (2008). Concept of Automatic Usability Evaluation of Safety-Critical Interactive Systems in the Field. Sicherheitskritische Systeme , pp. 18-23.

FUNK, M., VAN DER PUTTEN, P. & CORPORAAL, H. (2008). Specification for User Modeling with Self-observing Systems. Proceedings of the First International Conference on Advances in Homputer-Human Interaction.

GILBOA S. & RAFAELI, A. (2003) "Store Environment, Emotions and Approach Behavior: Applying Environmental Aesthetics to Retailing," The International Review of Retail, Distribution and Consumer Research (13:2), pp. 195-211.

GOODMAN, C.S. & AHN, R. (1999) Methodological approaches of health technology assessment. International Journal of Medical Informatics 56, 97-105.

GREENBERG, S. & BUXTON, B. (2008). Usability Evaluation Considered Harmful (Some of the Time). CHI 2008 Proceedings. Florence, Italy**.**

HAMACHER, N., (2008) Reviser: Rapid Evaluation of Interactive Systems using Expert Knowledge [Online]. Available: http://www.hamacher.eu/reviser [Accessed 2012-05-09]

HARPER, S., MICHAILIDOU, E. & STEVENS, R. (2009) Toward a Definition of Visual Complexity as an implicit Measure of Cognitive Load. ACM Transactions on Applied Perception, Vol. 6, No. 2, Article 10.

HEATH, T., SMITH, S. G. & LIM, B. (2000) "Tall Buildings and the Urban Skyline: The Effect of Visual Complexity on Preferences," Environment & Behavior (32:4), pp. 541-556.

HOLZINGER, A. (2002) Multimedia Basics, Volume 3: Design. Developmental Fundamentals of multimedial Information Systems. New Delhi, Laxmi Publications.

HOLZINGER, A. (2005) Usability Engineering for Software Developers. Communications of the ACM, 48, 1, 71-74.

HOLZINGER, A. & MOTSCHNIK-PITRIK, R. (2005), Considering the Human in Multimedia: LearnerCentered Design (LCD) & Person-Centered e-Learning (PCeL). In: Mittermeir, R. T. (Ed.) Innovative Concepts for Teaching Informatics. Vienna, Carl Ueberreuter, 102-112.

HOLZINGER, A., STICKEL, C., FASSOLD, M. & EBNER, M. (2009) 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.) HCI and Usability for e-Inclusion. 5th Symposium of of the Austrian Computer Society, USAB 2009, Lecture Notes in Computer Science LNCS 5889. Heidelberg, Berlin, New York, Springer, 178-192.

HOLZINGER, A., KOSEC, P., SCHWANTZER, G., DEBEVC, M., HOFMANN-WELLENHOF, R. & FRÜHAUF, J. (2011) 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)

HSIAO, S.-W. & CHOU, J.-R. (2005) A Gestalt-like perceptual measure for home page design using a fuzzy entropy approach. International Journal of Human-Computer Studies 64 (2006) 137-156

INCE, I. F., SALMAN, Y.B. & YILDIRIM, M. E. (2009) A User Study: The Effects of Mobile Phone Prototypes and Task Complexities on Usability. ICIS 2009, November 24-26, 2009 Seoul, Korea

JASPERS, M.W.M (2009) A comparison of usability methods for testing interactive health technologies: Methodological aspects and empirical evidence. International Journal of Medical Informatics 78 (2009) 340-353

KAPLAN, S. & KAPLAN, R. (1983) Cognition and Environment: Functioning in an Uncertain World, Belmont, CA: Praeger Publishers.

KARAHOCA, A., BAYRAKTAR, E., TATOGLU, E. & KARAHOCA, D. (2010). Information system design for a hospital emergency department: A usability analysis of software prototypes. (ELSEVIER, Ed.) Journal of Biomedical Informatics (43), pp. 224-232.

KOFFKA, K. (1935). Principles of gestalt Psychology. New York: Harcourt, Brace & World.

KOHLER, W. (1947), Gestalt Psychology. New York: Liveright.

LANG, P. J. (1994) "The Varieties of Emotional Experience: A Meditation of James–Lange Theory," Psychological Review (101), pp. 211-221.

LILJEGREN, E. (2006). Usability in a medical technology context assessment of methods for usability evaluation of medical equipment. (ELSEVIER, Hrsg.) International Journal of Industrial Ergonomics (36), S. 345-352.

MARCUS, A. (2002) "The Cult of Cute: The Challenge of User Experience Design," Interactions (9:6), pp. 29-34.

MAVRIDES, C.M. & BROWN, D.R. (1969). Discrimination and reproduction of patterns: Feature measures and constraint redundancy as predictors. Perception and Psychophysics, 6(5), 276 – 280.

MEHRABIAN, A. & RUSSELL, J. A. (1974) An Approach to Environmental Psychology, Cambridge, MA: MIT Press.

MEMMEL, T., REITERER, H. & HOLZINGER, A. (2007) Agile Methods and Visual Specification in Software Development: A Chance to Ensure Universal Access. In C.Stephanidis (Ed.): Universal Access in HCI, Part I, HCII 2007, LNCS 4554, pp 453-462

NASAR, J. L. (1987) "Physical Correlates of Perceived Quality in Lakeshore Development," Leisure Studies (7), pp. 259-279.

NASAR, J. L. (1997) "New Developments in Aesthetics for Urban Design," in Advances in Environment, Behavior, and Design, E. H. Zube and G. T. Moore (ed.), New York: Plenum Press, pp. 149-193.

NASAR, J. L. (2000) "The Evaluative Image of Places," in Person-Environment Psychology, W. B. Walsh, K. H. Craik, and R. H. Price (eds.), Mahwah, NJ: Lawrence Erlbaum Associates, pp. 117-168.

NGO, D. C. L. & BYRNE, J. G. (2001) "Another Look at a Model for Evaluating Interface Aesthetics," International Journal of Applied Mathematics and Computer Science (11:2), pp. 515-535.

NIELSEN, J. (1993). Usability Engineering. San Diego: Academic Press.

NIELSEN, J. (1994) Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY

NORMAN, D.A. (2004) Emotional Design: Why we love or hate everyday things. Basic Books, New York, 2004

NORMAN, D.A., & DRAPER, S. W. (Eds.) (1986). User centered system design: New perspectives on human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates.

Picture taken from the website Colorado Energy News [Online]. Available: http://coloradoenergynews.com/category/oil-and-gas/resources/ [Accessed 2012-05-13]

PIEROTTI, D. Usability Techniques, Heuristic Evaluation Activities, Xerox Corporation [Online]. Available: http://www.stcsig.org/usability/topics/articles/he-activities.html [Accessed 2012-05-02]

Qt [Online]. Available: http://qt.nokia.com/ [Accessed 2012-05-06]

RAUTERBERG, M. (1996) How to Measure Cognitive Complexity in Human-Computer Interaction. Cybernetics and Systems '96. Vienna, Austrian Society for Cybernetic Studies, 815-820.

REISENZEIN, R. (1994) "Pleasure–Arousal Theory and the Intensity of Emotions," Journal of Personality and Social Psychology (67), pp. 525-539.

RUDD, J., STERN, K., & ISENSEE, S. (1996) Low vs. high fidelity prototyping debate. Interactions, 3, 1 (1996), 76-85.

RUSSELL, J.A. (1980) A circumplex model of affect. Journal of Personality and Social Psychology 39, 1161-1178

RUSSELL, J.A. (1989) "Measures of Emotion," in Emotion: Theory, Research, and Experience (Volum3 4), R. Plutchik and H. Kellerman (ed.), New York: Academic Press, pp. 83-111.

RUSSELL, J.A. & Barrett, L. F. (1999) "Core Affect, Prototypical Emotional Episodes, and Other Things Called Emotion: Dissecting the Elephant," Journal of Personality and Social Psychology (76), pp. 805-819.

SOLOWAY, E., GUZDIAL, M. & HAY, K. E. (1994) Learner-Centered Design: The challenge for HCI in the 21st century, Interactions, 1(2). 36-48

SOYUZMULTFILM, (1976) Children's cartoon "38 Parrots" Soviet Union 1976 [Online]. Available: http://www.youtube.com/watch?v=8gAjmPose1k [Accessed 2012-05-14]

STICKEL, C., EBNER, M. & HOLZINGER, A. (2010) The XAOS Metric - Understanding Visual Complexity as Measure of Usability. In G. Leitner, M. Hitz, and A. Holzinger (Eds.): USAB 2010, LNCS 6389, pp. 278-290

STICKEL, C., EBNER, M., STEINBACH-NORDMANN, S., SEARLE, G. & HOLZINGER, A. (2009) Emotion Detection: Application of the Valence Arousal Space for Rapid Biological Usability Testing to Enhance Universal Access. In C. Stephanidis (Ed.): Universal Access in HCI, Part I, HCII 2009, LNCS 5614, pp 615-624

THOMAS, J.C. & RICHARDS, J.T. (2007) In Sears, A., &, Jacko, J. A. (Eds.). (2007). The human-computer interaction handbook: Fundamentals, evolving technologies, and emerging applications (2nd ed.). Boca Raton, FL: CRC Press. (Edited book, Handbook), pp 493-498.

TOHIDI, M., BUXTON, W., BAECKER, R. & SELLEN, A. (2006) Getting the Right Design and the Design Right: Testing Many is Better than One. Proc ACM CHI, (2006), 1243-1252.

VICENTE, K.J., MORAY, N., LEE, J.D., RASMUSSEN, J., JONES, B.G., BROCK, R. & DJEMIL, T. (1996) Evaluation of a Rankine cycle display for nuclear power plant monitoring and diagnosis. Human Factors, 38 (3), 506 – 521.

Webmin [Online]. Available: http://www.webmin.com/ [Accessed 2012-05-05]

WICKENS, C.D. & HOLLANDS, J.G. (2000). Engineering psychology and human performance, 3rd ed. Upper Saddle River NJ: Prentice-Hall.

WOLFRAM, S. (2002). A new kind of science. Champaign, IL: Wolfram Research

WRIGHT, P., MCCARTHY, J. & Marsh, T. (2001) "From Usability to User Experience," British HCI Group Members Magazine (46), pp. 4-11.

XRCed [Online]. Available: http://xrced.sourceforge.net/ [Accessed 2012-05-09]

Appendix A – Thinking Aloud – Checklist (German)

1) Vorbereitung:
    i) Oberfläche für neuen Benutzer zurücksetzen.
    ii) Überprüfe, ob im Testraum alles bereit ist.
2) Begrüßung:
    i) Testperson begrüßen.
    ii) Orientierungsskript durchgehen.
    iii) Hintergrund Fragebogen durchgehen: Moderator stellt die Fragen und füllt Formular aus.
    iv) Testperson soll Einverständnis- und Geheimhaltungs-Erklärung durchlesen und unterschreiben.
3) Test Session:
    i) Zum Testbereich (Computer) hinübergehen.
    ii) Aufnehmen am Computer (Morae) starten.
    iii) Laut mitdenken (TA) trainieren und üben.
    iv) Testperson beginnt mit erster Aufgabe.
    v) Testperson beendet letzte Aufgabe.
4) Abschluß:
    i) Interview: Wie war's?
    ii) "Did anything strike you as particularly good?"
    iii) "Did anything strike you as particularly bad?"
    iv) "Did you feel yourself confident while working with the tool?"
    v) “Do you think you could implement routine configurations usig this tool independently?”
    vi) Individuelle Fragen stellen, die sich aus dem Test ergeben.
    vii) Feedback Fragebogen. Testperson füllt Formular aus.
    viii) Testperson danken, eventuelle Remuneration übergeben, hinaus begleiten.
5) Nachbereitung:
    i) Abschriften, Protkolle und Notizen organiseren.
    ii) Aufnahmen beschriften und sicherstellen.

Appendix B – Thinking Aloud – Background questionnaire (German)

1. Angaben zur Person

Geschlecht: [ ] männlich [ ] weiblich

Alter: ____________

Beruf: ____________

2. Sehvermögen

1. Verwenden Sie eine Sehhilfe bei der Arbeit am Computer?

[ ] Keine *[ ] Brille* *[ ] Kontaktlinsen* *[ ] Sonstige* __________

2. Sind Sie farbenblind?

[ ] Nein *[ ] Ja, und zwar*__________

3. Ausbildung

Abgeschlossene Ausbildung:

[ ] Lehre *[ ] Matura* *[ ] Studium* *[ ] Sonstige* __________

4. Umgang mit Computern

1. Wie lange benutzen Sie bereits Personal Computer?

_____ *Jahre*

2. Wie viele Stunden pro Woche verwenden Sie einen Computer?

_____ *Stunden*

3. Welche Betriebssystem verwenden Sie am meisten?

[ ] Windows *[ ] MacOS* *[ ] Linux/Unix* *[ ] Sonstige* ________

5. Domain-Spezifische Fragen

1. Wie oft brauchen Sie die Änderungen im TGtool einzutragen?

[ ] noch nie *[ ] selten* *[ ] gelegentlich* *[ ] regelmäßig*

2. Wie oft verwenden Sie das TGtool selbstständig?

[ ] noch nie *[ ] selten* *[ ] gelegentlich* *[ ] regelmäßig*

6. Erfahrung mit Usability Tests

Haben Sie schon an eine Usability Studie teilgenommen?

[ ] als Testperson *[ ] als Mitglied des Testteams*

Wenn ja, was war das für eine Studie? _______________

Appendix C – Thinking Aloud – Non-disclosure and consent form (German)

Vertraulichkeits- und Einverständniserklärung

Danke, dass Sie an unserer Studie teilnehmen. Bitte beachten Sie, dass Ihnen unter Umständen vertrauliche Informationen zuteil werden und dass Sie diese nicht weitergeben dürfen. Bild- und Tonaufnahmen werden von Ihrer Sitzung gemacht, um es anderen, die heute nicht anwesend sein können, zu ermöglichen, aus Ihrem Feedback Nutzen zu ziehen.

Bitte lesen Sie die untenstehende Einverständniserklärung und unterschrieben Sie an der dafür vorgesehenen Stelle. Vielen Dank.

Ich erkläre, keine Informationen aus der Studie weiterzugeben.

Ich weiß, dass Bild- und Tonaufnahmen von meiner Sitzung gemacht werden. Ich gebe die Erlaubnis, diese Aufnahmen für Lehrzwecke und im Rahmen wissenschaftlicher Forschung zu verwenden.

Testperson

Ort: ____________________________

Datum: ____________________________

Name: ____________________________

Geburtsdatum: ____________________________

Unterschrift: ____________________________

Appendix D – Thinking Aloud - Internal Task List

Task No.	Description	Prerequisites	Completion Criteria	Max. Time	Possible Solution Path
1	[First impressions] Please login and look around	Configuration tool running	User indicates to be finished looking around.	2 min.	Any
2	[Medium] Please create a signal with given parameters	Configuration tool running	Signal is successfully added	3 min.	In navigation menu, click "Signals and groups". Fill in the fields in “Add/Edit a signal” dock. Click “Add”
3	[Difficult] Please change signal properties to the given parameters	Configuration tool running Signal exists	Signal properties are successfully changed	3 min	In navigation menu, click "Signal properties". Filter the signal. Change its properties direct in the table.
4	[Medium] Please create a signal group with given parameters	Configuration tool running	Signal group is successfully added	3 min.	In navigation menu, click "Signals and groups". Fill in the fields in “Add/Edit a signal group” dock. Click “Add”
5	[Medium] Please add the signal to the group	The signal and the signal group are existing	The signal is added to the group	1 min.	In navigation menu, click "Signals and groups". In the Server-Group-Signal tree click on the signal group. In the list of signals without group click on the required signal. Click on the button “<<”
6	[Difficult] Please delete created signal and signal group	The signal and the signal group are existing	The signal and the group are deleted	2 min	In navigation menu, click on "Signals and groups". Choose the signal in the Server-Group-Signal tree and in “Add/Edit a signal” dock click “Delete”. In the Server-Group-Signal tree choose the signal group and “Delete” in “Add/Edit a signal group” dock

Appendix E – Thinking Aloud– Feedback Form (German)

Datum: _______________ Uhrzeit: _______________ Test Nr.: __________ User Nr.: __________

Feedback Formular

Bewerten Sie bitte anhand folgender Aspekte ihre Zufriedenheit mit dem Konfigurationstool. Markieren Sie dazu die passendste Nummer mit einem Kreis.

1.	Zum gewünschten Menü hinfinden.	Sehr leicht	3	2	1	0	1	2	3	Sehr schwer
2.	Qualität der graphische Darstellung der Funktionen.	Sehr gut	3	2	1	0	1	2	3	Sehr schlecht
3.	Man kommt ohne Hilfe sehr leicht zu recht.	Sehr leicht	3	2	1	0	1	2	3	Sehr schwer
4.	Signalfilter.	Sehr gut	3	2	1	0	1	2	3	Sehr schlecht
5.	Graphische Gestaltung des Konfigurationstools, inkl. Farben und Grafiken.	Sehr gut	3	2	1	0	1	2	3	Sehr schlecht
6.	Konsistenz des Konfigurationstools.	Sehr konsistent	3	2	1	0	1	2	3	Sehr inkonsistent
7.	Geschwindigkeit des Konfigurationstools.	Sehr schnell	3	2	1	0	1	2	3	Sehr langsam

8.	Dieses Tool kümmert sich über meine Zufriedenheit als Benutzer.	Sehr	3	2	1	0	1	2	3	Gar nicht
9.	Die Funktionalität dieses Tools ist relevant für meine Tätigkeit.	Sehr relevant	3	2	1	0	1	2	3	Gar nicht relevant
10.	Gesamteindruck des Tools.	Sehr gut	3	2	1	0	1	2	3	Sehr schlecht
11.	Würden Sie dieses Tool selbstständig verwenden?	Auf jeden Fall	3	2	1	0	1	2	3	Niemals
12.	Werden Sie dieses Tool öfter verwenden?	Auf jeden Fall	3	2	1	0	1	2	3	Niemals

Appendix F – Heuristic Evaluation - Detailed descripton of heuristics

Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Appendix G – Heuristic Evaluation - Severity five-point rating scale

1 Cosmetic,

will not affect the usability of the system, fix if possible.

2 Minor,

users can easily work around the problem, fixing this should be given low priority.

3 Medium,

users stumble over the problem, but quickly adapt to it, fixing this should be given medium priority

4 Major,

users have difficulty, but are able to find workarounds, fixing this should be mandatory before the system is launched. If the problem cannot be fixed before launch, ensure that the documentation clearly shows the user a workaround

5 Catastrophic,

users are unable to do their work, fixing this is mandatory