Zuerst ersch. in : Work with display units 94 : selected papers of the Fourth International Scientific Conference on Work with Display Units, Milan, Italy, 2-5 October, 1994 / ed. by Antonio Grieco ... [et al.]. - Amsterdam [etc.] : Elsevier, 1995. - S. 247-251 . - ISBN 0-444-82145-7

DESIGN AID TOOLS FOR USER INTERFACE DESIGN

Harald Reiterer[a+b] and Stefan Schäfer[a]

[a] German National Research Centre for Computer Science (GMD), PO. Box 1316, D-53731 St. Augustin, Germany

[b]University of Vienna, Institute for Applied Computer Science and Information Systems, Liebiggasse 4/3-4, A-1010 Vienna, Austria

1. INTRODUCTION

To reach high ergonomic quality Graphical User Interfaces (GUIs), human factors knowledge has been developed and expressed as guidelines (for example [Mayhew,1992]), style guides (for example [IBM,1992]), and standards (for example [ISO,1994], [EN,1993]). The volume of these available sources is huge. Simultaneously, GUI designers need today more and more competence, knowledge, and experience to handle this great amount of information. This means for most of them that executing their jobs requires taking into account far more information than they can possible keep in mind or they can possibly apply.
Empirical results have shown that most of the software designers have no or only very limited knowledge about human factors [Molich and Nielsen,1990]. Therefore most of them were not able to apply human factors knowledge in the design process, even though they have information or get information about their existence. In this study, designers were asked what kind of support they prefer to overcome their lack of human factors expert knowledge. A great amount of them replied that they would prefer computer based design aids integrated in their development tools. However, the problem is how to capture and encode human factors knowledge relevant to designers' tasks and how to present it to them in formats that support their mode of work. This results into a need for GUI development tools with domain competence based on human factors knowledge that may be encountered, learned, practised, and extended during ongoing use - in other words, tools in which users *learn and use on demand*. An important research goal in GUI development tools is therefore to discover helpful, unobtrusive, structured, and organised ways to integrate the use of this human factors knowledge into the tools without stifling creativity [Hartson and Boehm-Davis,1993]. These research issues should include methods and tools for providing the designer assistance in understanding, searching, and applying this knowledge. This leads to following questions: what is the best presentation format for communicating human factors knowledge and how could we ensure that it will be observed?

2. DESIGN AID TOOLS FOR COMMUNICATING HUMAN FACTORS KNOWLEDGE

This last thought was the starting point for the GMD project IDA (User Interface Design Assistant) [Reiterer 1994]. The primary goal of this project is to explicitly incorporate domain

competence in user interface development tools to empower GUI designers. This means those development tools and designers are bringing complementary strengths and weaknesses to the job. Rather than communicating with tools, designers should perceive the use of the development tools as communication with an application domain [Fischer and Lemke,1988]. To shape the tools into a truly usable and useful medium, the tools should let the designers work directly on their problems and their tasks.

2.1 Basic design issues for design aid tools

To develop design aid tools, several issues have to be resolved. Lemke [Lemke,1989] defined basic design issues, which should be taken into account during the development of design environments. This section discusses these basic design issues for the area of user interface design.

- The *design domain*: The design domain GUI design is a sub domain of the whole software development process. As a field, GUI design includes GUI hardware and software, user and system modelling, cognitive and behavioural science, human factors, empirical studies, methodology, techniques, and tools. This is a wide area of knowledge and typical GUI designers do not necessarily have competence in all these areas. Design aid tools are a good chance to give them support required for their task, designing usable interfaces.
- The *elementary building blocks*: They are a collection of interaction objects like controls or widgets that are available for creating GUIs. The elementary building blocks for the design can either be taken directly from what is available in the tool (e.g. generic interaction objects of the UIMS) or a special higher-level building block can be created that is closer to the problem domain of the design environment (e.g. application domain specific interaction objects as templates).
- The *designers and their domain model*: It is important to identify the designers using the design environment. Knowing the types of designers will have important consequences for how the design environment will be shaped, the level of functionality provided, and how that functionality will be delivered to the designers. Typical properties of the GUI designers that should be considered are: known GUI programming languages, experience with previous GUI development tools, available learning time, etc.
- *Cognitive processes* in the use of design environments: Using a design environment remains a complex cognitive activity. This problem solving activity requires special needs for a design environment. It must support the designer's attention, comprehension and search activities. Attention is specially important if designers are unaware of some of the design issues. They do not search for something they are not aware. Therefore, the design environment must bring all important issues to the designer's attention. Comprehension means, that all information presented to the designer will be interpreted and integrated into his conceptual model of the domain.
- Supporting the *whole design process*: The design phases of problem structuring, solution generation, and solution evaluation should be supported by the design environment. The support of the whole GUI design process is a necessary precondition, but today seldom found in available development tools.

The consideration of all these basic design issues led to a multifaceted architecture for design aid tools assisting the GUI designers during the design, implementation, and evaluation process. They following have been developed in the IDA project and have been connected with a UIMS:

- An adviser (*Advice Giving Tool*) that supports design and implementation activities presenting human fac¬tor's knowledge for GUIs.

- A library of reusable ergonomic GUI software (*Construction Tool*) that supports implementation activities considering human factors.
- An evaluation tool (*Quality Assurance Tool)* evaluating the ergonomic quality of GUIs based on human factors knowledge.

2.2 Typical use of the design environment

The following design scenario gives an impression of the use of the IDA design aid tools. A designer is implementing a GUI for a specific application domain. The transient control panel, called IDA-Toolbar, enables the designer to launch design aid tools represented as icons in the toolbar, and is placed over the UIMS (Figure 1).

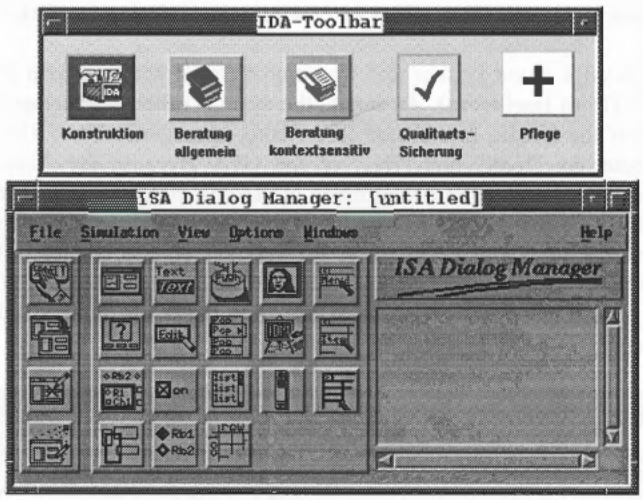

Figure 1. Presentation of the IDA design aid tools.

To reduce the implementation work load, the designer can use predefined ergonomic GUI templates. For this purpose, the *Construction Tool* is launched by clicking on the icon in the IDA-Toolbar. The construction tool then offers domain-oriented templates, contained in an object-oriented library, from which the designer specialises a required instance. This instance will be further integrated in the interface under design. Therefore the designer constructs the GUI by extracting predefined templates from the library and placing them into the working area of the UIMS. Now the designer can modify the instance of the template, based on specific application requirements. This allows a "design by modification" approach. If the designer needs advise the *Advice-giving Tool* is launched by clicking on the context-sensitive advice icon or the global advice icon in the IDA-Toolbar. This on-line advice giving system presents GUI design and human factors knowledge with the help of multimedia documents. The aim of the advice giving system is to determine a pattern matching between the examples of the adviser and the current task of the designer. The designer is aided in building an analogy by assuming that the presented example or information is relevant to his current task. If the designer now wants to evaluate the ergonomic quality of the GUI under design, the *Quality Assurance Tool* is launched by clicking on the icon in the IDA-Toolbar (see section 3). The quality assurance tool analysis the conformance of the GUI with the human factor's knowledge included in a knowledge base.

3. Quality Control of the User Interface Design Results

Several research projects in the domain of knowledge-based support for user interface designers have been carried out in the last years. Most of them tried to design the user interface automatically with the help of a rulebase in which styleguide rules were included. One result of these projects was the experience that this approach has its limits caused by the immense part of semantic knowledge used during the design process. Considering this experience we tried to find another way to support the designer during the design of a user interface. The main idea behind the *Quality Assurance Tool*, called QUID is to support the designer with a knowledge based system that could be putted into the statement "critiquing instead of solving". Therefore the main aim of the system QUID is to check a design for styleguide conformance and whether the main ergonomic ideas where taken into account during the design process.
QUID is an object-oriented knowledge-based system and was implemented with the help of the expert system-shell Kappa© (from Intellicorp). As input, the system takes the Dialogue Definition Language (DDL)-file of the UIMS, in our case ISA Dialog Manager©. The DDL-file contains all information about the "look" and "feel" of the GUI. Because the "feel" includes a great part of semantic information, the check made by QUID concerns only the "look". Using a c-interface, QUID asks for each object in the design and gets all information that is needed to check the ergonomic quality of an interface. All interface objects coming out of the c-interface are represented internally in QUID and the rules are chained over them. The results of QUID are shown in a window, that contains all information needed to understand the ergonomic deficiencies detected by QUID. Figure 2 shows an example window.

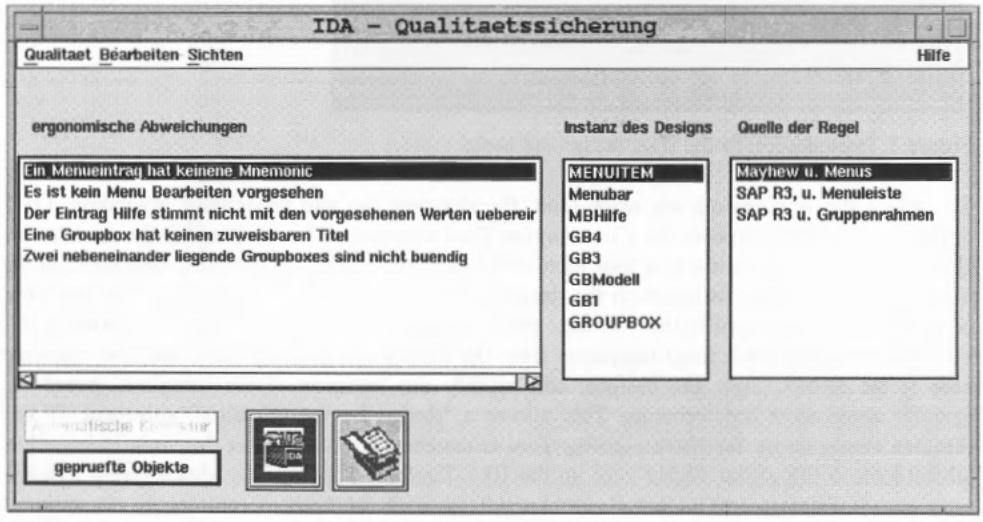

Figure 2. List presentation of all detected deficiencies.

In addition to this result presentation a second possibility was developed, presenting the results after every detection of an ergonomic deficit in the design. This presentation form is based on the common spell-checker metapher. Figure 3 shows an example window, which will be presented directly after the detection of each deficit.

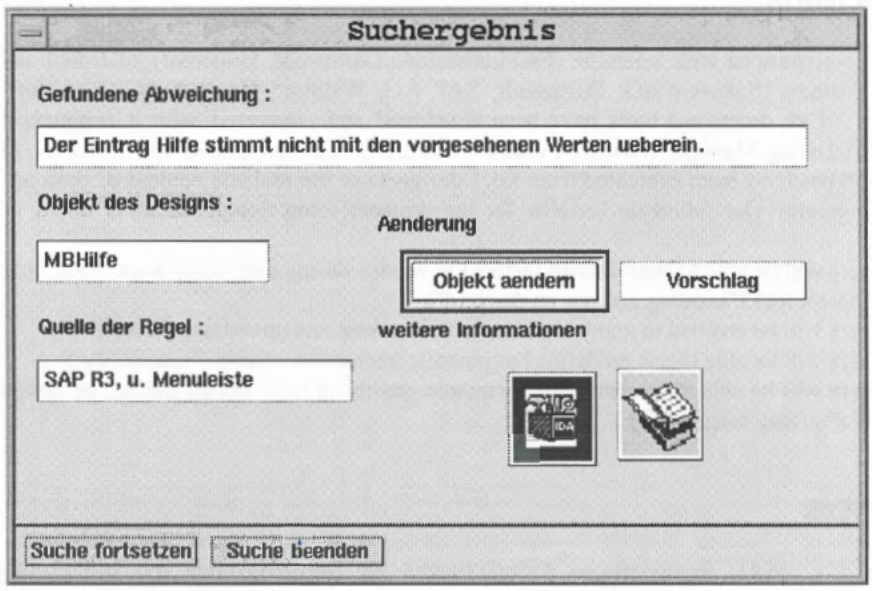

Figure 3. Spell checker oriented presentation of detected deficiencies.

The detected ergonomic deficits are documented with different information in both cases (Figure 2 and Figure 3). First of all there is a short comment given to the user, describing the kind of deficit that has been detected (labelled "ergonomische Abweichungen", "Gefundene Abweichung"). The second information identifies the faulty object, giving out the name of the object created by the designer (labelled "Objekt /Instanz des Designs"). Using this information, the designer can easily understand the output of QUID and can identify the problem and the object that has to be changed. The last information shown in the output-window concerns the source of the rule. Considering the output of QUID, the designer has several possibilities to continue his work:

1. If further explanation of the short comments are needed, the advice-giving tool can be activated (clicking on the icon , see Figure 2).
2. If a specific template is available, the construction tool can be activated (clicking on the icon , see Figure 2).
3. If the designer selects the identifier of an object, the object editor of the UIMS will be activated by double-clicking on the identifier or by pushing the push button labelled "Objekt aendern". Now the interaction object containing identified ergonomic deficiencies can be improved.
4. In some cases, detected deficiencies can be changed automatically. In this case the designer can activate a separate dialogue box (by pushing the push button labelled "Automatische Aenderungen" or "Vorschlag") which gives him the opportunity to correct the deficiency. The benefit for the designer is that he has not to branch into the object editor of the UIMS.

Putting it all together, the idea of QUID is a promising approach to provide knowledge-based support for a user-interface-designer. The combination of taking into account the limits of knowledge-based systems and maximising the abilities in analysing only the output of the UIMS and taking only those rules that concern the "look" of an interface, enables the system to give the optimal support that can be given by a knowledge-based system.

3. CONCLUSION

Based on co-operations with scientific (Fachhochschule Darmstadt; University of Bonn) and industrial partners (Software AG, Darmstadt; SAP AG, Walldorf; Hoechst AG, Frankfurt) prototypes of all design aid tools have been developed and connected with a commercial UIMS (ISA-Dialog Manager from ISA GmbH, Stuttgart). The usefulness and usability of these prototypes have been evaluated from GUI designers in the realistic context of their application domains. The following benefits for the designer using design aid tools could be shown:

- Designers will be able to learn human factors knowledge during their daily work using their development tool ("learning and use on demand").
- Designers will be enabled to apply ergonomic style guides and guidelines ("usability").
- Designers will be able to use predefined ergonomic interaction objects ("reusability").
- Designers will be able to evaluate the ergonomic quality of their design during the design process ("quality assurance").

REFERENCES

[EN,1993] EN 29241. Ergonomische Anforderungen für Bürotätigkeiten mit Bildschrimgeräten, 1993.

[Fischer and Lemke,1988] G. Fischer and A. Lemke. Construction Kits and Design Environments: Steps Toward Human Problem-Domain Communication, Human-Computer Interaction, Vol. 3, 1988, pp. 179-222.

[Fischer et al.,1991] G. Fischer, A. Lemke, T. Mastaglio and A. Morch. The role of critiquing in cooperative problem solving, ACM Transactions on Information Systems, Vol. 9, No. 3, 1991, pp. 123-151.

[Hartson and Boehm-Davis,1993] H. Hartson and D. Boehm-Davis. User interface development processes and methodologies, Behaviour & Information Technology, Vol. 12, No. 2, 1993, pp. 98-114.

[IBM,1992] IBM. Object-Oriented Interface Design, IBM Common User Access Guidelines. Que Corporation, Carmel, 1992.

[ISO,1994] ISO 9241. Ergonomic Requirements for Office Work with Visual Display Terminals, ISO, 1994.

[Lemke, 1989] C. Lemke. Design Environments for High-Functionality Computer Systems, Doctor Thesis, University of Colorado, 1989.

[Löwgren and Nordquist,1992] J. Löwgren and T. Nordquist. Knowledge-Based Evaluation as Design Support for Graphical User Interfaces, in Proc. of CHI'92, Addison-Wesley, Reading, 1992, pp. 181-188.

[Mayhew,1992] D. Mayhew. Principles and Guidelines in Software User Interface Design, Prentice Hall, Englewood Cliffs, 1992.

[Molich and Nielsen,1990] R. Molich and J. Nielsen. Improving a human-computer dialogue. Communications of the ACM, Vol. 33, No. 3, 1990, pp. 338-348.

[Reiterer, 1994] H. Reiterer. User Interface Evaluation and Design, Research Results of the Projects Evaluation of Dialogue Systems (EVADIS) and User Interface Design Assistance (IDA), GMD-Berichte Nr. 237, Oldenbourg, München, 1994