Zuerst ersch. in: Contemporary ergonomics 1994 : proceedings of the Ergonomics Society's 1994 Annual Conference, University of Warwick, 19-22 April 1994 : "Ergonomics for all" / S. A. Robertson. - London : Taylor & Francis, cop. 1994. - S.125-130. - ISBN 0-7484-0203-9

AN ENABLING SYSTEM FOR USER INTERFACE DESIGN

Harald Reiterer

German National Research Center for Computer Science (GMD), Human-Computer Interaction Research Division,
P.O. Box 1316 D-53731 St. Augustin, E-mail: reiterer@gmd.de
and
University of Vienna, Institute for Applied Computer Science and Information Systems, Liebiggasse 4/3-4, A-1010 Vienna

The paper presents an enabling system for user interface design, realised in the GMD project "User Interface Design Assistance (IDA)". The goal of the project is to assist interface designers during the design process of the user interface; enabling them to solve design problems in the area of *human factors* based user interface design (usability engineering). The enabling system is realised with the help of a *user interface design environment* (UIDE), which supports the learning on demand approach.

Introduction

The volume of information (e.g., standards, style guides, design guides) available to designers in the field of user interface design is increasing rapidly, especially in the area of designing graphical user interfaces (GUIs). In close relation with this circumstance is the fact, that today designers need more and more competence, knowledge, and experience to handle this great amount of information. For many designers this means that the execution of their jobs requires taking into account far more information than they can possible keep in mind or they can possibly apply. Computers seem a good chance to store this large amount of information and to support the designers applying them, because the computer is their primary tool doing their daily work. However, the problem is how to capture and encode information relevant to their tasks and how to present it to them in formats that support their mode of work. The result is a need for systems with domain competence based on domain knowledge (e.g., standards, style guides, design guides), which may be encountered, learned, practised, and extended during ongoing use - in other words, systems in which users *learn on demand* (Eisenberg and Fischer 1993). An innovative form of technological environments in the learning on demand area are *enabling systems*. They are characterised through a new form of co-operation between the user and the computer. This means the computer and the people using it are partners in the task-at-hand, bringing complementary strengths and weaknesses to the job (Fischer et al. 1991a).

Design Aid Tools for User Interface Design

To overcome the limitations of current user interface development tools - no methodology support for a human factor-based design of user interfaces, limited support of reusability of user interface software, no support for usability testing - was the starting point for the GMD project "User Interface Design Assistance" (IDA). The primary goal of this project is to develop a User Interface Design Environment (UIDE). The UIDE is based on a User Interface Management System (UIMS) that includes the following design aid tools and features (Reiterer 1993, 1993a):

- Library of reusable interface software (IDA CONSTRUCTION TOOL)
- Guidelines and advisers for methodology support (IDA ADVICE TOOL)
- Tool for evaluating user interfaces (IDA QUALITY ASSURANCE TOOL)
- Methodology- and tool-based integration of the interface development life cycle in an object-oriented application development life cycle

The IDA Construction Tool

This design aid tool offers domain-oriented building blocks (as models or templates) in a library, such as generic and domain-specific interface objects ("look") and dialogue scripts ("feel"). To construct the library an object-oriented mechanism is used. With the help of a UIMS (ISA Dialog Manager from ISA; XFaceMaker from NSL) the models of domain-oriented building blocks are constructed and saved as object-classes in the library. Using the library the designer generates an instance from each model. This instance will be integrated in the interface under design. Therefore designers construct interfaces by obtaining predefined building blocks from the library and placing them into the work area of the UIMS. Now the designer can modify the instance of the model, based on specific application requirements. So the IDA-construction tool supports design by modification.

Another important aspect is the maintenance of the models in the library. Existing models (e.g. domain specific user interfaces, dialogue boxes) can be modified by retrieving them from the library and manipulating them in the work area of the UIMS. Then they are saved again as models. If the designers chance the "look" and "feel" of a model, each "child" will also chance its "look" and "feel". So the maintenance activities could be dramatically reduced using an object-oriented library of models.

With the help of different information retrieval mechanisms the designer can search for relevant generic or domain specific interaction objects or dialogue scripts and use them as parts of the user interface. One information retrieval mechanism integrated in the IDA construction tool is a graphical browser, shown in Figure 2. The browser is based on a semantic tree, structuring the models . The designer has the possibility to browse through the library using the semantic tree structure finding a relevant interaction object or dialogue script. Another information retrieval mechanism is a query function (called „Textsuche ..." in the pull down menu „Bearbeiten", see Figure 1) based on keywords (e.g. typical attributes of interaction objects, semantic names of interaction objects and dialogue scripts, etc.) to search for a specific interaction object or dialogue script.

If the designer wants to see the look and feel of a model, he can press the push button "Anschauen" (which means look). A picture of the model and a list of all dialogue script names attached to this model are presented in a separate window. The designer can select a specific dialogue script name and with a double click he can see the whole dialogue script description. The description includes a lot of comments, declaring the functionality of the script. If the designer has found a model with the relevant look and feel, he can press the push button "Instantiieren" to make an instance of the model. This instance will appear in the work area of the UIMS and can now be used in the construction process of the user interface. If the designer needs further information when and how to use a specific model, he can press the push button "Beratung" (which means advice). This event triggers in a context-sensitive way a hypermedia document of the IDA advise tool.

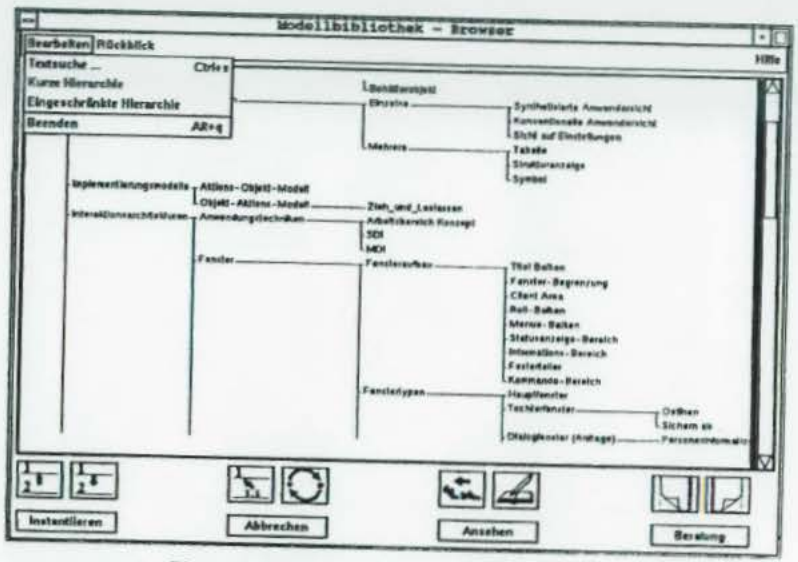

Figure 1. Browser of the IDA construction tool

The IDA-construction tool does not help designers perceive the shortcomings of an interface they are constructing. In that it includes only passive representatives; constructions of user interfaces in the work area of the UIMS do not talk back unless the designer has the skill and experience to form new appreciation's and understandings when constructing. Designers who are unaware of human factors knowledge about user interface design do not perceive a breakdown if one of the design guidelines is violated.

The IDA Quality Assurance Tool

This design aid tool identifies potential problems in the artifact being designed. It detects and critiques partial solutions constructed by the designer. These critiques are based on human factors knowledge and design principles for user interface design. It is a form of performance critics, whose prime objective is to help users create high-quality products in the least amount of time using as few resources as possible (Fischer et. al 1991a). Learning is not the primary concern of performance critic but can occur as a by-product of the interactions between users and critics.

With the help of an expert system (Kappa 3.0 from Intellicorp) the knowledge is implemented in a knowledge base as interface objects and condition-action rules, which are tested whenever the designer asked for a quality control. A very similar approach is found in Löwgren and Nordquist (1992).

The quality assurance procedure consists of six steps. The passive quality

assurance (critique) is explicitly invoked by the designer when he desires an evaluation (step 1). The result of the current design process is saved in a tool-dependent Dialogue Definition Language (DDL-) File. With the help of a parser the DDL-File is translated in a tool-independent DDL-File (step 2). The expert system of the IDA quality assurance tool uses this file as input and represents the result of the dialogue design with the help of an object tree (step 3). Using the rules of the knowledge base and controlled by an inference mechanism the expert system analysis the conformance of the user interface with the human factors knowledge (analytic critique) (step 4).This passive form of quality assurance usually evaluates the (partial) product of the design process, not the individual user actions that resulted in the product. The results of the analytic critiquing process are presented in a special window (step 5). This window contains a short description of all discovered ergonomic deficits (comments) and the identifier of the object which contains some ergonomic deficits. Now there are two possibilities (step 6): First, if the designer needs further explanation of the comments, he can activate the IDA advice tool or if a specific model is available, he can activate the IDA construction tool. Second, if the designer selects and double clicks the identifier of the object which contains some ergonomic deficits, he branches in the object editor of the UIMS and can reimplement the interaction object.

As an important feature, the designer can costumize and extend the IDA-quality assurance tool by modifying or adding interface objects, critique rules, and relationships. This end-user modifiability allows for evolution of the design environment as human factor knowledge and design requirements change.

The IDA Advice Tool

This design aid tool has two functions. First it explains the short critique messages of the IDA quality assurance more detail.

The second and primary function of the IDA advice tool is to support the designer during the design of the user interface. If the designer needs support in the area of human factors based interface design, he could get global or context-sensitive advise activating the advice tool. The global advice is based on a „guided tour: user interface design", explaining how graphical user interfaces should be designed, considering human factors. Activating the context-sensitive advice the designer gets some information when and how to use the selected interaction object in the UIMS and what should be the ergonomic "look" and "feel" of this interaction object (*advice level*). One goal is to determine an analogy between the examples of the IDA advice tool and the current task of the designer. The designers is aided in building an analogy by assuming that the presented example or information is relevant to his current task. If the designer wants deeper information, for example why the "look" and "feel" of a specific interaction object should be so, he gets it in a hypertextual form, e.g. by double clicking the relevant advise information (*explanation level*).

The IDA advice tool is based on hypermedia documents (see a typical example in Figure 2) and is developed with the help of hypermedia tools (MetaCard from MetaCard Coporation, IDS from ISA, Toolbook from Asymetrix). In Vanderdonckt (1993) a good survey of hypermedia systems on human-computer interaction principles and guidelines supporting the user interface designers is given. Unique characteristics of hypermedia documents are the multiplicity of connections between media fragments as opposed to the linear structure of traditional text and the availability of media other than text. The user modifiable connectivity of hypermedia documents provides quick access to explanatory, elaborative, and other related information. New media (e.g., graphic, animation, and sound) are more effective than text in conveying certain kinds of information such as two- and three-dimensional spatial relationships as well as processes,

behaviours, and evolution of systems. With the help of this different forms of representation, examples of good design are represented from multiple perspectives. In exploring examples from different perspectives, a minimalist strategy is followed. Designer using the IDA advice tool expand only the information they determine to be relevant to their task. The use of minimal explanation in the context of examples avoids overwhelming designers with irrelevant information. Whenever relevant design issues in the library of the IDA construction tool are available, the designer has the ability to retrieve them from the hypermedia documents and place them in the work area of the UIMS.

Figure 2. A typical IDA hypermedia document

Integration of interface development activities in an object-oriented life cycle

User interface design activities have to be embedded in the software development life-cycle. Today a lot of methods and tools for the application development are available (e.g., Structured Analysis, Entity-Relationship Model, Structured Analysis and Design Technique, Object-Oriented Methods). There are also some special methods for the problem structuring phase of the user interface development (e.g., State Transition Networks, Grammars, Rules and Constraints, Multiagent Techniques). Till now little work has been done to integrate methods of application development and user interface development. The research issue to be addressed here is the search for entirely new approaches to de-

sign representation, techniques, and methods. They should be based on a model that bridge the gap between the task-oriented behavioural world of the user and the constructional object- and tool kit-oriented world of interface software. The new object-oriented paradigm in the area of software engineering offers a good chance for bridging this gap . This assumption is based on the fact, that graphical user interfaces (GUIs) also use the object-oriented paradigm (e.g. X-Tool kit). The idea is to map, in a straight forward way, the application objects, attributes and methods (services) of the application development to the interaction objects, icons, windows, choices, and user interactions of the interface development (Balzert 1993). The aim should be to come to one general method for the whole development process (problem structuring, solution generation and solution evaluation, maintenance). Therefore the use of the UIDE will be integrated in an object-oriented development life-cycle.

Co-operations and Outlook

All project results are realised in close co-operation with companies (e.g., software developers, developers of UIMS). The aim of this co-operation is to develop the UIDE for specific application domains and to understand the day-to-day operational requirements and constraints placed on developers, as well as to comprehend the needs they have for methods and tools in the area of human factors based interface design. Workshops with members of the co-operating companies have been arranged, where prototypes of the design aid tools have been presented. These prototypes will now be the basis for domain specific design aid tools, which will be designed in 1994. It's also intended to evaluated the usefulness and usability of the domain specific design aid tools in the realistic context of application domains of the co-operating companies.

References

Balzert H. 1993: Der JANUS-Dialogexperte: Vom Fachkonzept zur Dialogstruktur. *Softwaretechnik-Trends*, 1993, Band 13, Heft 3, 62-72.

Eisenberg M. and Fischer G. 1993: Learning on Demand - Why Is It Necessary and Why Does It Make a Difference?, in: Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society, (Lawrence Erlbaum, Hilldale) 180-181.

Fischer G., Lemke A. and McCall R. 1991, Making Argumentation Serve Design. *Human-Computer Interaction*, 1991, Vol. 6, .393-419.

Fischer,G., Lemke, A., Mastaglio T. and Morch A. 1991a, The role of critiquing in cooperative problem solving. *ACM Transactions on Information Systems*, Vol. 9, No. 3, April 1991, 123-151.

Löwgren J. and Nordquist T. 1992, Knowledge-Based Evaluation as Design Support for Graphical User Interfaces. In *CHI'92 Porceddings*, pp.181-188.

Reiterer H. 1993, A Human Factors Based User Interface Design, in: Grechenig T., Tscheligi M. (eds.): *Human Computer Interaction, Vienna Conference, VCHCI'93, Fin de Siècle Vienna, Austria, September 1993, Proceedings*, Lecture Notes in Computer Science, (Springer, Wien), .291-302

Reiterer H. 1993a, The Development of Design Aid Tools for a Human Factor Based User Interface Design. In Conference Proceedings. 1993, IEEE International Conference on Systems, Man and Cybernetics. Volume 4, (IEEE, Piscataway) 361-366

Vanderdonckt J. 1993, Hypermedia on Human-Computer Interaction Principles and Guidelines: A Survey, *ACM SIGLINK Newsletter*, V2N4, December 1993, to be published.