

# IDA - A DESIGN ENVIRONMENT FOR ERGONOMIC USER INTERFACES

Harald Reiterer

University of Vienna, Institute for Applied Computer Science and Information Systems,  
Liebiggasse 4/3-4, A-1010 Vienna, Austria, Email: reiterer@ifs.univie.ac.at  
and  
German National Research Centre for Computer Science (GMD),  
Human-Computer Interaction Research Division (HCI), D-53731 St. Augustin, Germany

**KEY WORDS:** Advice-giving systems, Guideline, Heuristic Evaluation, Standards, Style guides, Tools for Working with Guidelines, Usability Engineering, User Interface Design.

**ABSTRACT.** Surveys have shown that user interface development tools with domain competence based on human factors knowledge (e.g. guidelines, style guides, standards) are needed. This knowledge should be encountered, learned, practised, and expanded during ongoing use so that designers can learn and use it on demand. This paper addresses an important research approach of user interface development tools, namely to discover helpful, unobtrusive, structured, and organised ways to integrate the use of human factors knowledge into development tools. The aim is to support designers during the design process with powerful design aid tools leading to high ergonomic quality user interfaces.

## 1. INTRODUCTION

The empirical survey of [Myers and Rosson, 1992] showed that

- an average of 48 % of the code in software applications with Graphical User Interfaces (GUIs) is devoted to the user interface (UI) portion;
- 74% of the UI of new applications was implemented using a tool kit, an interface builder or a User Interface Management System (UIMS);
- 6% were implemented using no tools.

These results reveal that most software development projects spend significant time and resources on designing and programming UI and that most projects are using UI development tools. Nevertheless the use of such tools does not guarantee GUIs of high ergonomic quality.

To reach high ergonomic quality GUIs, human factors knowledge has been developed and expressed as guidelines (e.g. [Mayhew,1992]), style guides (e.g. [IBM,1992], [OSF 1992], [Microsoft,1992], and standards (e.g. [ISO,1994], [EN,1993]), complete design guides (e.g. [Vanderdonck,1994]). The volume of these available sources is huge and UI designers need more and more competence, knowledge, and experience to handle them.

Most software designers have no or only very limited knowledge about human factors [Molich and

Nielsen,1990]. To overcome this lack of knowledge 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 UI development tools 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* [Eisenberg and Fischer,1993]. An important research goal is therefore to discover helpful, unobtrusive, structured, and organised ways to integrate the use of human factors knowledge into tools without stifling creativity [Hartson and Boehm-Davis,1993]. This leads to the questions: what is the best presentation format for communicating human factors knowledge and how could we ensure that it will be observed?

## 2. A DESIGN ENVIRONMENT 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 human factors knowledge in UI development tools to empower UI designers.

Before developing a user interface design environment (UIDE), several basic design issues should be taken into account [Lemke, 1989]:

- The *design domain*: The first decision is, what the problem domain should be, and what space of design problems should be supported. The design domain UI design is a sub domain of the whole software development process. Therefore UI design includes UI 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 UI designers do not necessarily have competence in all these areas. The UI design domain of the IDA project is concentrated on UI development tools for direct manipulative GUIs for office applications.
- 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 environment 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). An important aspect in this context is the reuse of higher-level building blocks. The design environment should provide mechanisms for reusability of elementary building blocks. In the IDA project, generic (e.g. interaction objects for GUIs) and higher-level building blocks (e.g. templates), based on the windowing systems Motif and Windows, have been used.
- The *designers and their domain model*:: 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 UI designers that should be considered are: known GUI programming languages, experience with previous UI development tools, available learning time, etc. The domain model in the UI design domain is based on the conceptual model of GUIs. It is based on different paradigms, perspectives, and metaphors, like the desktop metaphor and the object-oriented paradigm.
- *Cognitive processes* in the use of design environments: Using a design environment remains

a complex cognitive activity. This requires special needs for a design environment. It must support the designer's attention, comprehension and search activities. Attention is especially important if designers are unaware of some of the design issues. Therefore, the design environment must bring all important issues to the designer's attention. Comprehension is constrained by the designer's knowledge about the problem domain. To overcome this constraint the design environment must use adequate textual and graphical representations as well as problem domain-specific abstractions, and must give instructive context information. In the IDA project different retrieval, presentation, and navigation techniques have been implemented to support the cognitive processes of the UI designers (see sections 2.2 to 2.4).

- 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 UI design process is a necessary precondition, but today seldom found in available development tools. In the first phase of the IDA project the main focus was on UI implementation and evaluation activities, but it is intended to extend the focus to UI design activities (see section 3).

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

- An adviser (*Advice Giving Tool*) that supports design and implementation activities presenting human factor'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.

This multifaceted architecture derives its power from the integration of its components. Used individually, the components are unable to achieve their full potential (e.g., the designer must interact with different systems; there will be no consistent presentation of human factors knowledge; restricted possibilities to activate the different design aid tools in a context-sensitive way from each other). Used in combina-

tion, however, each component augments the value of the others in a synergistic manner.

### 2.1 Typical use of the IDA design environment

A UI designer using the IDA UIDE is implementing a GUI for a specific application domain. A transient control panel, called IDA-Toolbar (fig. 1), enables the designer to launch the IDA design aid tools represented as icons in the toolbar.

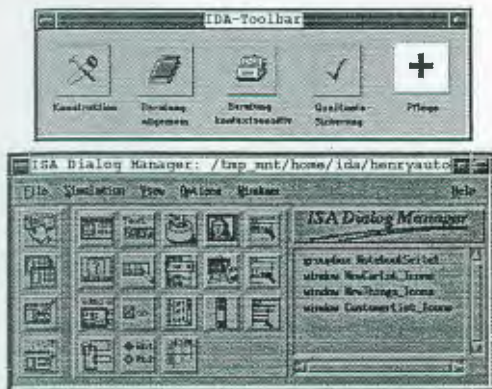


Fig. 1. Presentation of the IDA design aid tools.

To reduce the implementation work load, the designer uses predefined ergonomic GUI templates. For this purpose, he launched the *Construction Tool* by clicking on the icon in the IDA-Toolbar (see section 2.2). The construction tool offers domain-oriented templates, contained in an object-oriented library, from which the designer specialises a required instance. Therefore the designer constructs the GUI by extracting predefined templates from the library. 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 advice the *Advice-giving Tool* is launched by clicking on the context-sensitive advice icon or the global advice icon in the IDA-Toolbar (see section 2.3). This on-line advice giving system presents GUI design and human factors knowledge with the help of multimedia documents. The aim is to determine a pattern matching between the examples of the adviser and the current task of the designer.

If the designer now wants to evaluate the ergonomic quality of the GUI under design, he launched the *Quality Assurance Tool* by clicking on the icon in the IDA-Toolbar (see section 2.4). The quality assurance tool analysis the conformance of the GUI with the human factors knowledge included in a

knowledge base. The quality assurance tool offers the designer several opportunities to correct all detected ergonomic deficiencies.

### 2.2 Support during the Construction of the User Interface

The IDA construction tool offers domain-oriented templates in a library, such as generic and domain-specific UI objects (look) and dialogue scripts (feel). The templates are constructed with the help of an object-oriented mechanism of the UIMS - under consideration of the human factors knowledge - and saved as object-classes in the library. The predefined templates could represent a complete domain-specific window architecture (e.g. primary window with different secondary windows and dialogue boxes) based on common window architectures, like single document interface (SDI), multiple document interface (MDI) or workplace environment.

Using the library the designer generates an instance from each template. During the instantiation process the designer has the opportunity to customize the final look and feel of the instance of the template (e.g. a notebook for data entry of customer information in a car sales application). The designer can change a number of important attributes that define the look and feel (e.g. the sections of the notebook, the number of pages of each section, the label of each tab). The main advantage of this parametric instantiation process is that the amount of predefined templates can be minimized. This seems a successful way to reduce one of the main problems of the object-oriented approach: the poor reuse of predefined objects.

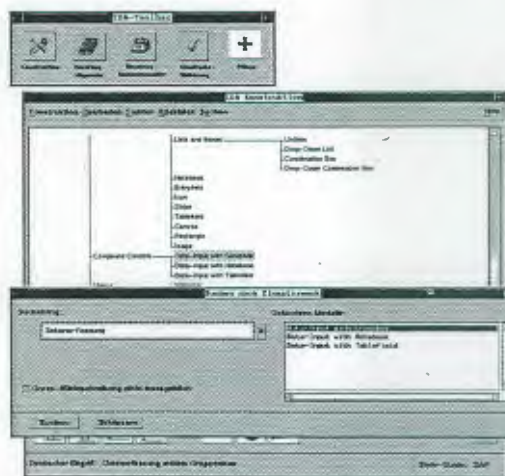


Fig. 2. IDA construction tool - graphical browser, query function.

An important requirement for the usefulness of the library is the available *information retrieval mechanism*. Without powerful information retrieval features the designer will not be able to find any relevant template for his specific design situation:

1. One information retrieval mechanism is a *graphical browser* based on a semantic tree, structuring the templates (fig. 2). This structure differentiates the templates from a designer's point of view and is based on tables of content typically found in style guides or guidelines (e.g., a domain-specific template based on the window architecture SDI is integrated in the semantic tree as a leaf of the following sub tree: dialogues - window architectures - single document interface - domain specific template SDI). The designer is able to browse through the library by using the semantic tree structure searching for a relevant template.
2. The *query function* is another information retrieval mechanism based on key words (e.g., typical uses of the templates, typical look and feel attributes of the templates, names of the templates) to search for a specific template. Fig. 2 shows as example the dialogue box of the query "Typical use of the templates". One of the predefined key words can be selected in a drop-down combination box (e.g. data entry) or a self-defined one can be entered. The result of the search is reported in a list box allowing a selection of all found templates (e.g. all predefined templates supporting data entry activities). An important precondition of the query function is, that the designer defines all necessary keywords when he develops a new template.
3. If the designer wants to explore important characteristics of a template before instantiating it, a *simulation mode* can be initiated. A visualisation of the template (look and feel) appears, that can be used in the same way as the instance of the template. The simulation mode supports the decision of the designer to use a template or not.

### 2.3 Advice during the Design Process of the User Interface

The IDA advice-giving tool assists the designer during UI design. The aim is to determine an analogy between the examples of the adviser (e.g. good and bad examples of ergonomic UI) and the current task of the designer.

The access to the advice-giving tool could be context-sensitive or global. When the designer activates the advice-giving tool in a *context-sensitive way* with

the help of the icon in the IDA-Toolbar (see fig. 3), he gets the related human factors knowledge of the selected interaction object in the working area of the UIMS (e.g. notebook). The knowledge will be presented in the main window of the advice-giving system. The system includes different navigation controls (e.g. menu items, icons, push buttons, hot spots) allowing multiple presentations of the human factors knowledge (see fig. 3).

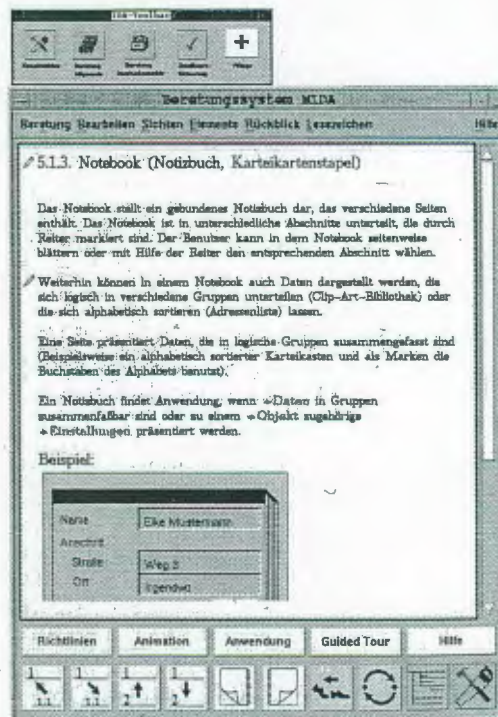


Fig. 3. IDA advice-giving tool - main window.

When the designer activates the advice-giving tool in a *global way* in the IDA Toolbar, the start-up window of the advice-giving tool appears. It gives the designer a short introduction in the tool and then he can use the different navigation and presentation facilities to get more information.

If the designer wants an interactive *presentation of design guidelines* related to the selected interaction object (e.g. notebook), he has the possibility to open a follow-up window. It offers in textual and graphical form an explanation of all important guidelines, which have to be considered using or designing the interaction object notebook.

For some interaction objects an animated *presentation* of its use and behaviour is available. Each animation is developed as a small movie (based on

screen or video recordings) that could be played, stopped, rewinded, etc. using a device control panel. If the designer needs support for the design decision when and how to use an interaction object, he can activate a question and answer dialogue. Depending on the answers the designer gets recommendations about the use of available interaction objects and templates. In the main window of the advice-giving system the recommended interaction object or template is shown.

A different way to present human factors knowledge in a global way is the *guided tour* "user interface design". The purpose of this guided tour is to explain how GUIs should be designed considering human factors. Considering an object-oriented GUI development life cycle consisting of six steps, all necessary design activities are described. If the designer goes through all six development steps, all sections of the global advice will be visited. This form of presentation allows a guided use of the advice-giving tool and is useful for novice designer in the area of GUI design.

Whenever a relevant template in the library of the IDA construction tool is available, the designer can retrieve it from the advice-giving tool.

An important aspect in multimedia systems is the *navigation support* to prevent users from getting "lost in hyperspace". Therefore different powerful navigation mechanisms have been embodied in the advice-giving tool (e.g., graphical browser, index and text query function).

#### 2.4 Quality Control of the User Interface Design Results

The IDA quality assurance tool identifies potential problems in the artefact being designed. It criticises partial solutions constructed by the designer. These critiques are based on human factor's knowledge. 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., 1991]. 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 the knowledge is implemented in a knowledge base as objects and condition-action rules, which are tested whenever the designer asks for a quality control. A very similar approach is found in [Löwgren and Nordquist, 1992].

The passive quality assurance is explicitly invoked by the designer when he desires an evaluation. The result of the current design process is saved in a tool-dependent Dialogue Definition Language (DDL-)

File of the UIMS. The expert system of the IDA quality assurance tool uses this file as input (C-interface) and represents the result of the dialogue design with the help of an object base. Using the rules of the knowledge base and controlled by an inference mechanism the expert system analyses the conformance of the UI with the human factors knowledge base (analytic critique).



Fig. 4. IDA quality assurance tool.

The results of the analytic critiquing process are presented in a special window reproduced in fig. 4. The window of the explanation component contains a short description of all discovered ergonomic deficiencies (heading "ergonomische Abweichungen"), the source of the short description (heading "Quelle der Regel") and the identifier of the evaluated object (heading "Instanz des Designs") which contains some ergonomic deficiencies. Now the designer has different possibilities to continue his work.

1. If further explanation of the comments are needed, the advice-giving tool can be activated.
2. If a specific template is available, the IDA construction tool can be activated.
3. If the designer selects the identifier of an object, the object editor of the UIMS can be activated and the interaction object containing identified ergonomic deficiencies can be improved.
4. In some cases, detected deficiencies can be changed automatically. The designer can activate a separate dialogue box that gives him the opportunity to correct the deficiency.

#### 3. CONCLUSION AND OUTLOOK

Based on co-operations with scientific (Fachhochschule Darmstadt; University of Bonn) and industrial partners (Software AG, Darmstadt; SAP AG, Wall-dorf; Hoechst AG, Frankfurt) prototypes of all design aid tools have been developed and connected with a commercial UIMS (ISA-Dialog Manager

with a commercial UIMS (ISA-Dialog Manager from ISA GmbH, Stuttgart). The usefulness and usability of these prototypes have been evaluated from UI designers. First interpretations of the evaluation results show the following benefits:

- Designers will be able to learn human factors knowledge 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 during the design process ("quality assurance").

It is planned to build a specific design aid tool supporting the design decision process. With the help of this tool the rationale for the various UI design decisions can be made explicit and recorded for later reference [Moran and Carroll, 1994]. Having access to an audit trail through the design rationale is important during iterative development. Since changes to the UI will often have to be made, it is helpful to know the reasons underlying the original design. Design rationales can be captured in a hypertext structure or with the help of video records of design meetings and selected user tests. It is planned to extend the IDA advice tool with special features for capturing design rationales based on a special repository including hypertext documents and videos.

#### ACKNOWLEDGEMENTS

Thanks to IDA project members Frank Bachmann, Doris Kamnitz-Kraft, Michael Porschen, Manfred Ramm, Stefan Schäfer, and Helmut Simm for their contributions to the project results. Special thanks to Jean Vanderdonck for helpful comments on earlier drafts of the paper.

#### REFERENCES

- EN 29241 (1993) Ergonomische Anforderungen fuer Buerotaetigkeiten mit Bildschrimgeraeten, 1993.
- Eisenberg, M. and Fischer, G. (1993) Learning on Demand - Why Is It Necessary and Why Does It Make a Difference?, in Proc. of the 15th Annual Conf. of the Cognitive Science Society, Lawrence Erlbaum, Hilldale, 1993, pp. 180-181.
- Fischer, G., Lemke, A., Mastaglio T. and Morch, A. (1991) The role of critiquing in cooperative problem solving, ACM Transactions on Information Systems, Vol. 9, No. 3, 1991, pp. 123-151.
- Hartson, H. and Boehm-Davis, D. (1993) User interface development processes and methodologies, Behaviour & Information Technology, Vol. 12, No. 2, 1993, pp. 98-114.
- IBM (1992) Object-Oriented Interface Design, IBM Common User Access Guidelines. Que Corporation, Carmel, 1992.
- ISO 9241 (1994) Ergonomic Requirements for Office Work with Visual Display Terminals, ISO, 1994.
- Lemke, C. (1989) Design Environments for High-Functionality Computer Systems, University of Colorado, 1989.
- Loewgren, J. and Nordquist, T. (1992) Knowledge-Based Evaluation as Design Support for Graphical User Interfaces, in Proc. of CHI'92, Addison-Wesley, Reading, 1992, pp. 181-188.
- Mayhew, D. (1992) Principles and Guidelines in Software User Interface Design, Prentice Hall, Englewood Cliffs, 1992.
- Microsoft (1992) The Windows Interface, An Application Design Guide. Microsoft Press, 1992.
- Molich, R. and Nielsen, J. (1990) Improving a human-computer dialogue. Communications of the ACM, Vol. 33, No. 3, 1990, pp. 338-348.
- Moran, T. and Carroll, J. (1994) Design Rationale. Hillsdale: Lawrence Erlbaum, 1994.
- Myers, B. and Rosson, M. (1992) Survey on user interface programming, in Proc. CHI'92, Addison-Wesley, Reading, 1992, pp. 195-202.
- OSF (1993) Open Software Foundation. OSF/MOTIF Style Guide, Revision 1.2. Prentice-Hall, London, 1993.
- Reiterer, H. (1994) A User Interface Design Assistant Approach, in Applications and Impacts, Information Processing'94, Proc. of 13rd IFIP World Computer Congress, Brunstein K. and Raubold E. (Eds.), IFIP Transactions A-52, Volume II, North Holland, Amsterdam, 1994, pp.180-187
- Vanderdonck, J. (1994) Guide ergonomique de la présentation des applications hautement interactives, Presses Universitaires de Namur, Namur, 1994.