The Development of Design Aid Tools for a Human Factor Based User Interface Design

Dr. Harald Reiterer
German National Research Center for Computer Science (GMD), Institute for Applied Information Technology, Human-Computer Interaction Research Division, PO. Box 1316D-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

Abstract **This paper reports first results of the GMD project "User Interface Design Assistance (IDA)". The objective of this project is to develop computer based design aid tools for a human factor based user interface design. The reasons for concentrating on human factors are their increasing importance in the new European Economic Area and the lack of knowledge of the designers in the area of human factors. The design aid tools are integrated in a user interface management system (UIMS) to support the designers during the development process. The aim is to impart human factors knowledge for innovative user interfaces (e.g. object-oriented graphical interfaces; 3-D user interfaces; hypermedia user interfaces) to the designers. This will be done with the help of design aid tools, like a composition tool, a tutoring tool and a quality assurance tool. They are presented in some detail. First results have shown, that it is principally possible to realise such design aid tools.**

1. HUMAN FACTORS BASED USER INTERFACE DESIGN

1.1. The Increasing Importance of Human Factors

One reason for the increasing importance of human factors based user interface design is the new European Economic Area (EEA, consisting of EC and EFTA). To establish common working conditions for visual display terminal (VDT) users, the European Community published a "directive concerning the minimum safety and health requirements for VDT workers" [13]. The national governments of the EC members have to transform this directive into national law. In this process the European standardisation activities of the CEN (Comité Européen de Normalisation) and the international standardisation activities of the ISO (International Organisation for Standardisation) concerning ergonomic requirements for VDTs will have great influence. Especially the ISO standard 9241 "Ergonomic requirements for office work with VDTs" [23] plays an important role [8]. In the future this standard will often be an integral part of software requirements specification. Therefore software designer will have to take the requirements and principles of this standard into consideration.

Does the software designer know how to apply standards like the ISO 9241 in the design process? Empirical results have shown that most of the software designers have no or only very limited knowledge about human factors [2, 3, 32]. Therefore most of them were not able to apply standards from the area of human factors in the design process. In [3] designers were asked what kind of support to they prefer to overcome their lack of human factors expert knowledge. A great amount of them said that they would prefer computer based design aids which should be integrated in their design tools (e.g. interface builder, UIMS). What they won't like is "paperware", e.g. manuals or technical reports with a great amount of written style guides or guidelines. So an important research goal in the area of interface design is to discover helpful, unobtrusive, structured, and organised ways to integrate the use of principles, guidelines, standards, style guides, and design rules into the design process without stifling creativity [19]. These research issues should include methods and tools for offering the designer assistance in understanding, searching, and applying design principles, guidelines, and standards. This leads to the question, what is the best presentation format for communicating design rules and how could we ensure that they will be observed?

1.2. User Interface Development Environments

To reach the goal designing human factor based user interfaces, the software designer needs excellent tools, especially for graphical and hypermedia user interfaces. These tools should allow the designer to concentrate on the design process and on the quality of the design results, e.g. usability. Tools as for example User Interface Management Systems (UIMS) help the designer to specify, design, prototype, implement, execute, evaluate, modify, and maintain user interfaces. UIMSs are to interface development what CASE tools are to development of application components [20]. The big difference from UIMS to other tools like user interface tool kits and interface builders is that they support the designer during the construction of the presentation layer (static component) as well as during the definition of the dialogue

scripts (dynamic component). Therefore they consist of a WYSIWYG-Editor for the presentation layout and a script editor for the dialogue flow. Normally they have the availability to simulate the results (user interface prototype without any application code), so that the designer can use a prototyping approach during the design of the user interface. The handling of UIMSs is easy in comparison with other user interface design tools (e.g. tool kits) or conventional programming language, so that also non programmers like graphic designers or human factors experts could use them during the user interface design. This fact opens the opportunity to carry out user interface design activities with a team of different specialists. Table I summarises the expected benefits from using an UIMS [41].

TABLE I
BENEFITS USING USER INTERFACE MANAGEMENT SYSTEMS

User-interface (Dialog) independence
- Separate interface design from internals.
- Enable multiple user-interface strategies.
- Establish role of user-interface architect.
- Enforce standards.

Methodology and notation
- Develop design procedures.
- Find ways to talk about design.
- Create project management.

Rapid prototyping
- Try out ideas very early.
- Test and revise repeatedly.
- Engage end users, managers, and other concerned people.

Software support
- Increase productivity.
- Offer constraint and consistency checks.
- Facilitate team approaches.
- Ease maintenance.

2. DESIGN AID TOOLS FOR A HUMAN FACTORS BASED USER INTERFACE DESIGN

The increasing importance of human factors based user interface design, the lack of human factors knowledge of the designers and the availability of new user interface design tools were starting points for a new GMD project called "User Interface Design Assistance (IDA)". The objective of this project is to develop design aid tools for a human factor based user interface design. These design aid tools are integrated in a UIMS to support the designers of innovative user interfaces (e.g. object-oriented graphical interfaces; 3-D user interfaces; hypermedia user interfaces) during the development process. The aim is to impart human factors knowledge for innovative user interfaces to the designers. Therefore the designers should have direct access to the human factors knowledge from their development tool. The presentation of the design knowledge is based on object-oriented, multimedia and knowledge-based techniques. There are multifarious reasons for developing design aid tools for the user interface designers using a UIMS:

- to overcome the lack of human factors knowledge of the designers ("training on the job"),
- to overcome the resistance of the designers reading and using written guide lines, standards, etc. ("new design aids"),
- to reduce the costs developing user interfaces using predefined models ("reusability of software"),
- to assure easy of use of user interfaces ("usability"),
- to guarantee the conformity of the user interface with standards and style guides ("quality assurance").

Fig. 1 shows three design aid tools that assist the designers during the user interface design using a UIMS ("new design aids"). These design aid tools are under development in the IDA project:

- A *composition tool* to construct innovative user interfaces; based on a library of interaction objects and dialogue scripts [25].
- An *advice tool* to present the human factors knowledge; based on hypermedia documents [7, 31], a library of interaction objects and an expert system.
- A *quality assurance tool* to evaluate the conformance of the user interface with the human factors knowledge; based on an expert system [14, 16, 17, 24, 28, 39].

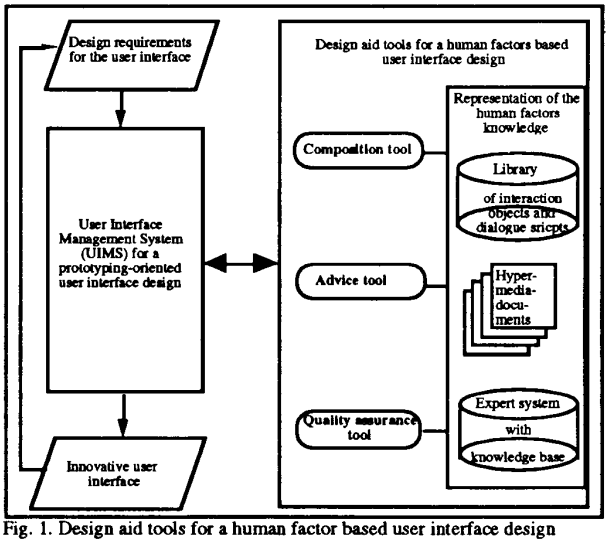

Fig. 1. Design aid tools for a human factor based user interface design

The GMD is starting cooperations with some companies to construct the design aid tools for specific application domains. For this purpose some workshops with members of companies, which are developing software applications and members of companies, which are developing UIMS were arranged. The workshops are completed with interviews and field studies. The scope of the interviews and field studies is

to determine, through observation and deduction, an operational model of what real interface developers presently do, as well as what they should be doing. The goal is to understand the day-to-day operational requirements and constraints placed on developers and the needs they have for methods and tools in the area of human factors based interface design. It's also intended to evaluated the usefulness and usability of the design aid tools in the realistic context of application domains of the co-operating companies.

2.1 . Human Factors Style Guide

An important pre-condition for developing computer based design aid tools is to formalise human factors knowledge to allow a computer based presentation. Especially in the area of object-oriented graphical user interfaces the GMD Human-Computer Interaction Research Division has a deep understanding of the necessary design principles and design guidelines, the relevant international standards, the available style guides and the construction and use of evaluation methods for usability testing [36, 37]. In the area of 3-D and hypermedia user interfaces the Human-Computer Interaction Research Division is doing much R&D activities. The results of these R&D activities and the available experiences in the field of Human-Computer Interaction are the basis for constructing new design principles and guidelines for innovative user interfaces. The human factors knowledge is summarised in a "Human Factors Style Guide", which is the foundation for the computer supported design aid tools. The content of this "Human Factors Style Guide" isn't restricted to commercial available user interface style guide requirements, e.g. OSF/MOTIF Style Guide [34], OPEN LOOK Style Guide [45], IBM Common User Access [21, 22], Windows Style Guide [30], SIEMENS/NIXDORF Style Guide [42], APPLE Style Guide [1], NeXT Style Guide [33]. It's based on a broader approach and also includes user interface design requirements from the relevant international standards [11, 23] and literature [5, 6, 12, 29, 35, 43]. The focus of the first version of our "Human Factors Style Guide" is on state of the art user interfaces (GUIs). Our research activities during the IDA project we will be the basis to include new design principles and requirements for innovative user interfaces (e.g. 3-D, hypermedia).

2.2 . Composition Tool

The *composition tool* supports the process of constructing the user interface and is based on a library. The content of the library is a collection of predefined generic and domain specific interaction objects and dialogue scripts. The generic interaction objects and dialogue scripts are application independent. Their "look" and "feel" is based on the principles of the "Human Factors Style Guide". The "look" and "feel" of the domain specific interaction objects and dialogue scripts is determined by the application domain, e.g. office system, telecooperation system, CSCW under consideration of the principles of the "Human Factors Style Guide". The idea behind is, that there is a real need for libraries of artefacts (interaction objects and scripts) that have been developed, tested, and refined to be shared within a given task domain, independent of software. This would lead to increased productivity by avoiding reinvention, by raising the level of abstraction in design work, and by providing necessary information to allow artefacts to be deployed successfully within interface design [19].

With the help of an information retrieval component (e.g. structural browser) the designers could search for a relevant generic or domain specific interaction object or dialogue script and use it as a part of the user interface. On the basis of predefined interaction objects and scripts the designers will be able to build the final interface ("reusability of software"). To construct the library an object-oriented mechanism is used. The interaction objects and scripts are designed as models or templates with the UIMS. Each model (object class) can transmit its "look" and "feel" to a specific interaction object (instance of this object class). If the designers chance the "look" and "feel" of a model, each "child" will also chance its "look" and "feel".

In the lead time of this project we have designed a small library with the UIMSs "ISA/Dialog Manager" and "XFaceMaker2" on SUN Workstations and on PCs. The library consists of a small set of interaction objects and scripts based on the Common User Access (CUA) of IBM. It was shown, that it is principally possible to design such a library as a basis for a composition tool.

2.3 . Advice and Explanation Tool

The *advice and explanation tool* presents the designers the human factors knowledge of the "Human Factors Style Guide" with the help of hypermedia documents and an expert system ("training on the job"). The knowledge is presented in a textual, graphical and animated form in the hypermedia documents. If the designers need support in the area of human factors design, they could get object-sensitive advise. After pressing a push button, they get advise when and how to use a specific interaction object and what should be the "look" and "feel" of this interaction object (*advice level* of the tutoring tool). If the designers wants deeper information, for example why the "look" and "feel" should be so, they get them in a hypertextual form, e.g. by double clicking the relevant advise information (*explanation level* of the tutoring tool). So there are two levels of tutoring available: the advise level with short information and the explanation level with deeper information.

If there exists a relevant model for an interaction object or a script in the library, the designers have the possibility to retrieve it from the library directly in the hypermedia document (link to the library).

A more active form of design support is based on the expert system. If the inference mechanism of the expert system de-

tects some design deficits, a commentary is generated automatically. It shows the designers the analysed deficit and presents them the relevant human factors knowledge in form of a hypermedia document or shows them a relevant interaction object of the library. With the help of these instructions the designers could improve the user interface.

In the lead time of this project we have designed some hypermedia documents with the hypermedia system "DIDOT" (from ISA GmbH, Stuttgart) on SUN Workstations. The content of the hypermedia documents is based on the Common User Access (CUA) of IBM. It was shown, that it is principally possible to design such a tutoring tool .

2.4 . *Quality Assurance Tool*

The *quality assurance tool* evaluates the conformance of the user interface with human factors knowledge. The knowledge is also based on the "Human Factors Style Guide" and is represented in a knowledge base of an expert system with the help of rules, frames and constraints [26, 40, 46]. The expert system uses the results of the user interface design process as an input and analysis - with the help of the knowledge base and an inference mechanism - the conformance of the user interface with the human factors knowledge ("quality assurance"). The results of the evaluation are comments that show existing deficits and give the designers some advise and explanations to improve the user interface. For this purpose the expert system activates the relevant topics in the tutoring tool or a relevant interaction object in the library. The use of the quality assurance tool is based on the assumption that the designers will use a prototyping approach [18]. After each design cycle the user interface prototype will be evaluated by the quality assurance tool. In the next design cycle all detected deficits can be removed.

In the lead time of this project we have evaluated some expert system shells for the purposes of this project [44]. We use a C/C++ based expert system shell (ProKappa from Intellicorp), because we have strong performance and portability requirements. We have built a small knowledge base - based on the Common User Access (CUA) - to demonstrate the possibility to analyse user interfaces with the help of an expert system. We use an analytic critic approach [15], which checks products with respect to predefined features and effects. In analytical approaches, critics do not need a complete understanding of the product. The quality assurance tool uses a set of rules to identify undesirable features among user interface units, but it does not identify all possible problems within user interface design. Its rule base allows it to criticise user interfaces without exactly knowing the requirements and preferences of the user interface designer.

2.5. *Tool Independence and Maintenance of the Design Aid Tools*

An important requirement for the project is to realise tool independent design aid tools. In principle it should be possible to integrate the design aid tools in each C-based UIMS. Therefore different UIMSs will be used as platforms for the integration of the design aid tools, e.g. the "XFaceMaker2" (from Concept asa, Frankfurt and NSL, Paris), the "ISA/Dialog Manager" (form ISA GmbH, Stuttgart). The control program IDA controls the communication and the data flow between the UIMS and all the design aid tools. So there is a clear interface between different UIMSs and the design aid tools. IDA is also responsible for the co-ordination of the different design aid tools. If for example the quality assurance tool detects a deficit IDA got a message to activate a relevant hypermedia document of the tutoring tool at the advise or explanation level. Therefore IDA acts also as an interface between the designer and the different design aid tools.

Another important feature from IDA is the support of the maintenance of the human factors knowledge in the different design aid tools. For this purpose a maintenance module (e.g. a structural browser based on a data dictionary and a meta data base) shows the designer where the knowledge is situated. If chances are necessary or new knowledge should be included the designer could see where the knowledge is located or should be included. In the data dictionary and the meta data base all necessary maintenance information is saved.

2.6 . *Design Aid Tools and the Software Life-cycle*

It's clear that the user interface design has to be embedded in the software development life-cycle. Today a lot of methods and tools for the application development (e.g. Structured Analysis, Entity-Relationship Model, Structured Analysis and Design Technique, Object-Oriented Method) are available. There are also some special methods for 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 design representation, techniques, and methods. They must be based on sound principles (e.g. abstraction, step-wise refinement) that have been successfully applied in the area of software engineering. They also should be based on a model that bridge the gap between the task-oriented behavioural world of the user and the constructional object- and toolkit-oriented world of interface software. We think that 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 modern graphical user interfaces (GUIs) also use the object-ori-

ented paradigm (e.g. X-Toolkit). 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. The aim should be to come to one general method for the whole development process. Therefore we are planning to integrate the use of the design aid tools in an object-oriented development life-cycle [4, 9, 10, 38].

3. SUMMARY

In the future a software development environment (e.g. CASE Tool) should contain both, a user interface development environment (UIDE) and an application development environment (ADE). Both environments will contain common tools including programming language compilers, linkers, loaders, debuggers, code analysers, configuration version control managers, and documentation tools. Each development environment will also contain special tools. This paper concentrates on tools and facilities that are unique to the UIDE. In an ideal UIDE the following tools and facilities should be included [27]:

- Tools for specifying user interfaces
- Libraries of reusable software
- Guidelines and advisers
- Tools for evaluating user interfaces

Today commercial UIDE normally includes only tools for specifying user interfaces and in some innovative UIDE you can also find libraries with limited use. In this paper we present some ideas how we can reach a real UIDE including sophisticated libraries of reusable software (composition tool), guidelines and advisers (advice and explanation tool) and tools for evaluating user interfaces (quality assurance tool). We called our tools design aid tools, because their primary focus is not to support the design process (like editors) but to give the designers some advice and evaluation during the design process. We are sure that with the help of such design aid tools the aim "to construct more usable innovative user interfaces" could be reached.

REFERENCES

[1] Apple, *Macintosh Human Interface Guidelines*. Menlo Park, California: Addison Wesley, 1992.

[2] G. Aschersleben and B. Zang-Scheucher, "Der Prozeß der Software-Gestaltung - Eine Bestandsaufnahme in Wissenschaft und Industrie," in *Software-Ergonomie '89*, S. Maaß and H. Oberquelle, Eds. Stuttgart: Teubner, 1989, pp.244-253.

[3] J. Beimel, J. Hüttner and H. Wandke, "Kenntnisse von Programmierern auf dem Gebiet der Software-Ergonomie: Stand und Möglichkeiten zur Verbesserung", unpublished paper of a lecture on the *Fachtagung der Sektion Arbeits-, Betriebs und Organisationspsychologie des Berufsverbandes Deutscher Psychologen "Arbeits- Betriebs- und Organisationspsychologie vor Ort"* 25.-27.5.1992, Bad Lauterbach.

[4] E. Booch, *Object-Orientied Design*. Redwood City, California: Benjam Cummings, 1991.

[5] C. M. L. Brown, *Human-Computer Interface Design Guidelines*. Norwood: Ablex Publishing, 1988.

[6] J. R. Brown and S. Cunningham, *Programming the User Interface, Principles and Examples*. New York: John Wiley & Sons, 1989.

[7] School of Information & Computing Sciences, *BRUIT-SAM: An Interface for User Interface Guidelines* (HyperCard stack), Gold Coast, QLD, 4229, Australia: Bond University, 1992.

[8] A. Cakir, "EG-Richlinie für Bildschirmarbeitsplätze," *Office Managment*, 1-2/1991, pp.46-53.

[9] P. Coad and E. Yourdan, *Object Oriented Analysis*. Englewood Cliffs: Prentice-Hall, 1991.

[10] P. Coad and E. Yourdan, *Object Oriented Design*. Englewood Cliffs: Prentice-Hall, 1991.

[11] DIN 66234 Teil 8, *Bildschirmarbeitsplätze, Grundsätze der Dialoggestaltung*, Februar 1988.

[12] J. S. Dumas, *Designing User Interfaces for Software*. London: Prentice-Hall, 1988.

[13] EEC, *European directive concerning "The minimum safety and health requirements for work with display sreen equipment"*, 90/270/EEC.

[14] G. Fischer, K. Nakakoji , J. Ostwald, G. Stahl and T. Sumner, "Embedding Computer-Based Critics in the Contexts of Design," in *INTERCHI '93 Proceedings*, April 1993, pp.157-164.

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

[16] J. Foley, W. Kim, S. Kovacevic and K. Murray, "UISW - An Intelligent User Interface Design Environment," in *Intelligent User Interfaces*, J. Sullivan and S. Tyler, Eds. New York: ACM Press, 1991, pp.339-384.

[17] P. Gorny and A. Viereck, "EXPOSE, Ein Software-Ergonomie-Expertensystem," in *Posterband zur Software-Ergonomie '91*, M. Rauterberg and E. Ulich, Eds. Zürich: IfAP-ETH Zürich, 1991, pp.152-161.

[18] H. Hartson and E. Smith, "Rapid prototyping in human-computer interface development," *Interacting with Computers*, vol. 3, no 1, 1991, pp.51-91.

[19] H. Hartson and D. Boehm-Davis, "User interface development processes and methodologies,"

Behaviour & Information Technology, vol. 12, no. 2, 1993, pp.98-114.

[20] D. Hix, "Generations of User-Interface Management Systems," *IEEE Software*, September 1990, pp.77-87.

[21] IBM, *Systems Application Architecture, Common User Access, Guide to User Interface Design*, 1991.

[22] IBM, *Systems Application Architecture, Common User Access, Advanced Interface Design Reference*, 1991.

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

[24] C. Jansen, A. Weisbecker and J. Ziegler, "Generation User Interfaces form Data Models and Dialogue Net Specifications," in *INTERCHI'93 Proceedings*, pp.418-423.

[25] J. Johnson, B. Nardi, C. Zarmer and J. Miller J., "ACE: Building Interactive Graphical Applications," *Communications of the ACM*, April 1993, vol. 36, no. 4, pp.41-55.

[26] P. Klahr and D. Waterman, *Expert Systems, Techniques, Tools and Applications*. Menlo Park, California: Addison-Wesley, 1986.

[27] J. Larson, *Interactive Software, Tools for Building Interactive User Interfaces*, Englewood Cliffs: Prentice Hall , 1992.

[28] J. Löwgren and T. Nordquist, "Knowledge-Based Evaluation as Design Support for Graphical User Interfaces," in *CHI'92 Porceddings*, pp.181-188.

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

[30] Microsoft, *The Windows Interface, An Application Design Guide*, Microsoft Press, 1992.

[31] MITRE, *Dynamic Rules for User Interface Design, DRUID 2.0* (HyperCard stack), Bedford: MITRE Corporation, 1991.

[32] R. Molich and J. Nielsen, "Improving a human-computer dialogue," *Communications of the ACM*, vol. 33, no. 3, 1990, pp.338-348.

[33] NeXT, *Interface Builder Styleguide Release 2.0*, 1991.

[34] Open Software Foundation, *OSF/MOTIF Style Guide, Revision 1.2*, London: Prentice-Hall, 1993.

[35] H. Reiterer, *Ergonomische Kriterien für die menschengerechte Gestaltung von Bürosystemen - Anwendung und Bewertung*, Dissertation, Universität Wien, 1990.

[36] H. Reiterer, "EVADIS II: A new Method to Evaluate User Interfaces," in *People and Computers VII*, A. Monk, D. Diaper and M. D. Harrison, Eds. Cambridge: University Press, 1992, pp.103-115.

[37] H. Reiterer and R. Oppermann, "Evaluation of User Interfaces, EVADIS II - A comprehensive Evaluation Approach," *Behaviour & Information Technology*, 1993, in press.

[38] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen, *Object-Oriented Modeling and Design*. Englewood Cliffs: Prentice-Hall, 1991.

[39] F. Rusell, P. Pettit and S. Elder, "INTUIT: A Computer Assisted Software Engineering Support For User-Centred Design," in *Mehtods and Tools in User-Centred Design for Information Technology*, M. Galer, S. Harker and J. Ziegler Eds. Amsterdam: North-Holland, 1992, pp.345-370.

[40] H. Schildt, *Artificial Intelligence using C*. McGraw Hill, 1987.

[41] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 2nd ed. Reading, Massachusetts: Addison-Wesley, 1992

[42] Siemens/Nixdorf, *Styleguide, Richtlinien zur Gestaltung von Benutzeroberflächen*. München: Siemens/Nixdorf, 1990.

[43] L. S. Smith and J. Moiser, *Guidelines for Designing User Interface Software*. Bedford: MITRE Corporation, 1986.

[44] A. C. Stylianou, G. R. Madey and R. D. Smith, "Selection Criteria for Expert System Shells: A Socio Technical Framework," *Communikations of the ACM*, vol. 35, no.10, Oct. 1992, pp.30-48.

[45] Sun Microsystems, *Open Look Style Guide*, 1989.

[46] E. Wenger, *Artificial Intelligence an Tutoring Systems*. Los Altos: Morgan Kaufmann, 1987.