

Die IDA-Entwicklungsumgebung

Einsatz von objekt-orientierten, multimedialen und wissensbasierten Unterstützungswerkzeugen zur ergonomischen Gestaltung von Benutzungsoberflächen

Harald Reiterer

Institut für Angewandte Informatik und Informationssysteme, Universität Wien, Liebiggasse 4/3-4, A-1010 Wien
Institut für Angewandte Informationstechnik, Gesellschaft für Mathematik und Datenverarbeitung, D-53731 Sankt Augustin

Zusammenfassung. Durch den Einsatz der IDA-Entwicklungsumgebung, bestehend aus einem Entwicklungswerkzeug für graphische Benutzungsoberflächen und verschiedenen Unterstützungswerkzeugen, werden Entwickler in Fragestellungen der software-ergonomischen Gestaltung unterstützt. Damit wird den Entwicklern die Möglichkeit zum bedarfsorientierten Erlernen und Anwenden von Ergonomiewissen geboten (learning on demand). Die verschiedenen Unterstützungswerkzeuge machen bei der Vermittlung von Wissen von objekt-orientierten, multimedialen und wissensbasierten Techniken Gebrauch. Es wird ein Vergleich mit verwandten Forschungsansätzen vorgenommen und eine Abgrenzung durchgeführt. Das Vorgehen bei der Entwicklung, der Einsatz, der funktionale Umfang sowie Evaluationsergebnisse bzgl. der IDA-Entwicklungsumgebung werden näher dargestellt. Abschließend wird ein Ausblick auf geplante Forschungsaktivitäten gegeben.

Schlüsselwörter: Software-Ergonomie, Benutzungsoberflächen, Mensch-Maschine Systeme, Entwicklungswerkzeuge und -techniken, Klassenbibliothek, Multimedia Systeme, wissensbasierte Evaluation

Abstract. The IDA user interface design environment consisting of a user interface development tool and different design aid tools supporting designers of graphical user interfaces during their work is presented. The design aid tools assist the designer applying human factors knowledge to develop ergonomic user interfaces. The use of the design aid tools in close relation with the development tool opens the designer the opportunity to learn on demand. The different design aid tools of the IDA user interface design environment have been developed with the help of object-oriented, multimedia, and knowledge based techniques. The development process, the typical use, the main characteristics, and results of an evaluation of the design aid tools are presented in some detail. Relations and differences with related research approaches are shown. Finally an outlook on planned research activities is given.

Key words: Human factors, user interfaces, man-machine systems, tools and techniques, object-oriented library, multimedia systems, knowledge-based evaluation

CR Subject Classification: H.5.2, D.2.2, H.1.2, H.5.1

1. Einleitung

Das Forschungsprojekt „User Interface Design Assistance (IDA)“ der Gesellschaft für Mathematik und Datenverarbeitung (GMD) verfolgt das Ziel, Entwicklern von graphischen Benutzungsoberflächen eine Entwicklungsumgebung zur Verfügung zu stellen, die sie bei der Umsetzung von software-ergonomischem Gestaltungswissen unterstützt [2, 25, 26]. Als wissenschaftliche Kooperationspartner konnten für das Projekt die Fachhochschule Darmstadt und die Universität Bonn gewonnen werden.

Anknüpfungspunkte für die Durchführung des Projektes waren einerseits die zunehmende Bedeutung von software-ergonomischen Forderungen aufgrund der von der EU-Kommission verabschiedeten „Richtlinie über die Mindestvorschriften bezüglich der Sicherheit und des Gesundheitsschutzes bei der Arbeit an Bildschirmgeräten (EG 90/270)“ und andererseits empirische Untersuchungen, welche die unzureichende Verbreitung software-ergonomischer Erkenntnisse in der Praxis festgestellt haben [4, 19]. Diese Untersuchungen zeigten, daß die Mehrheit der Software-Entwickler über keinerlei oder nur geringfügige software-ergonomische Kenntnisse verfügten. Zur Überwindung dieses Defizits wurde von den befragten Entwicklern eine rechnerbasierte Vermittlung software-ergonomischer Gestaltungsanforderungen während des Entwicklungsprozesses bevorzugt. Dazu benötigen Entwickler leistungsfähige Entwicklungswerkzeuge (z.B. User Interface Management Systems), die sie im Entwicklungsprozeß unterstützen und gleichzeitig bei der Einhaltung software-ergonomischer Anforderungen anleiten.

Durch die im Rahmen des hier vorgestellten Projektes realisierte Entwicklungsumgebung (Abb.1), bestehend aus einem Entwicklungswerkzeug und verschiedenen Unterstützungswerkzeuge, soll erreicht werden, daß Entwickler

- software-ergonomische Kenntnisse während des Implementierungsprozesses, gleichzeitig mit dem Einsatz des Entwicklungswerkzeuges, vermittelt bekommen (learning on demand),
- vorgefertigte, ergonomischen Anforderungen entsprechende Dialogbausteine einer Benutzungsoberfläche verwenden können (reusability),
- bei der Anwendung von ergonomischen Standards und Richtlinien angeleitet werden (usability) und
- die ergonomische Qualität der entworfenen Benutzungsoberflächen schon im Entwicklungsprozeß überprüfen lassen können (quality assurance).

Abbildung 1 zeigt die wesentlichen Komponenten der IDA-Entwicklungsumgebung. Als Input liegen Anforderungen aus dem Fachkonzept vor (siehe Abschn.5). Basierend auf diesen Anforderungen erfolgt, unter Verwendung eines Entwicklungswerkzeuges und unter Beachtung von software-ergonomischen Gestaltungsanforderungen, die Entwicklung der graphischen Benutzungsoberfläche. Bei deren Konstruktion kann der Entwickler auf vorgefertigte Dialogbausteine einer Klassenbibliothek zurückgreifen. Benötigt er zusätzlich Beratungsleistung zu ergonomischen Fragestellungen, kann er dazu online verfügbare Multimedia-Dokumente heranziehen. Zur Überprüfung der ergonomischen Qualität vorliegender Benutzungsoberfläche kann er die Wissensbasis eines Expertensystems nutzen, das eine automatische Qualitätssicherung vornimmt und ergonomische Defizite aufzeigt.

Da es nicht Ziel des Vorhabens war ein neues Entwicklungswerkzeug für graphische Benutzungsoberflächen

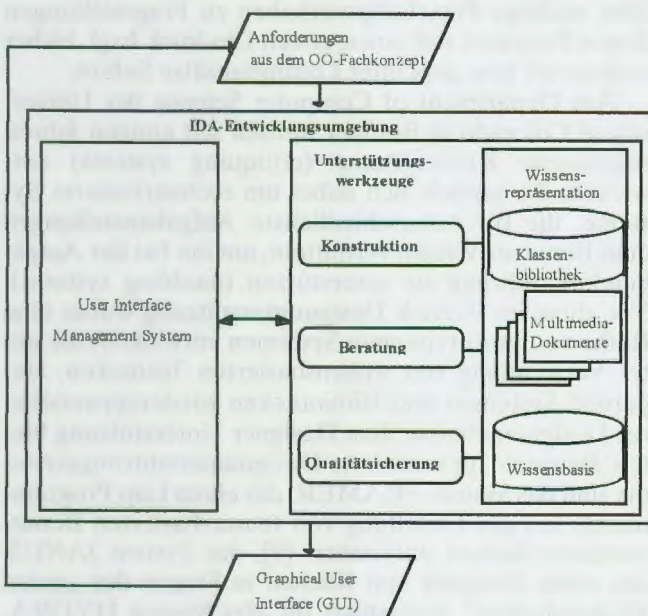


Abb.1. Die IDA-Entwicklungsumgebung im Überblick

zu entwickeln, sondern marktgängige mit zusätzlichen Unterstützungsleistungen auszustatten, wurden Hersteller derartiger Entwicklungswerkzeuge als industrielle Kooperationspartner gesucht. Anhand einer Kriterienliste [25] wurden Entwicklungswerkzeuge verschiedenster Hersteller evaluiert. Entscheidend für die endgültige Auswahl waren schlußendlich folgende Forderungen:

- objekt-orientierte Mechanismen zum Aufbau einer Bibliothek von Dialogbausteinen,
- Anbindung an kommerziell verfügbare Programmiersprachen (z. B. C bzw. C++),
- Schnittstellen zu externen Tools (z.B. Expertensystem, Multimedia Tools),
- Zugang zu den Entwicklern des Entwicklungswerkzeuges und deren Bereitschaft zur Anpassung des Entwicklungswerkzeuges,
- Zugang zu Anwendern des Entwicklungswerkzeuges.

Unter diesen spezifischen Anforderungen, die sich aus den Fragestellungen des Projektes ergaben, erwies sich das Entwicklungswerkzeug ISA Dialog Manager der Firma ISA GmbH (Stuttgart) als geeignet. Die Firma ISA konnte auch als Kooperationspartner gewonnen werden und beteiligte sich an der Bereitstellung notwendiger Schnittstellen und Leistungen.

Damit konkrete Anforderungen an die Entwicklungsumgebung erfaßt und sie im praktischen Einsatz evaluiert werden konnten, wurde eine Reihe von Industriekontakten (Software AG, Darmstadt; SAP AG, Walldorf/Baden; Hoechst AG, Frankfurt/Main) geknüpft. Diese führten zu Kooperationsprojekten, in deren Rahmen die IDA-Entwicklungsumgebung auf PCs (DOS/Windows) und Workstations (UNIX/Motif) prototypisch realisiert und evaluiert wurde.

Anhand eines typischen Einsatzbeispiels für die IDA-Entwicklungsumgebung wird im Abschn.2 ein Einblick in die angebotenen Unterstützungsleistungen sowie das Zusammenspiel der Unterstützungsleistungen dargestellt. Abschn.3 zeigt verwandte Forschungsansätze und grenzt das vorliegende Vorhaben von diesen ab. Dies erlaubt eine Einordnung des IDA-Vorhabens in die Forschungslandschaft und zeigt, in welchen Aspekten das Vorhaben über existierende Ansätze hinausgeht. Im Abschn.4 werden die einzelnen Unterstützungsleistungen der Entwicklungsumgebung ausführlich beschrieben sowie die zugrunde gelegten Konzepte diskutiert. Dabei werden die im Rahmen einer empirischen Evaluation gewonnenen Erkenntnisse dargestellt. Abschn.5 faßt die für die Akzeptanz der IDA-Entwicklungsumgebung entscheidenden Aspekte zusammen und gibt einen Ausblick auf weitere Forschungsaktivitäten.

2. Typischer Einsatz der IDA-Entwicklungsumgebung

Abbildung 2 zeigt, wie sich die IDA-Entwicklungsumgebung dem Entwickler präsentiert. Der Entwickler implementiert, unter Verwendung des Entwicklungswerkzeuges (unteres Fenster), eine graphische Benut-

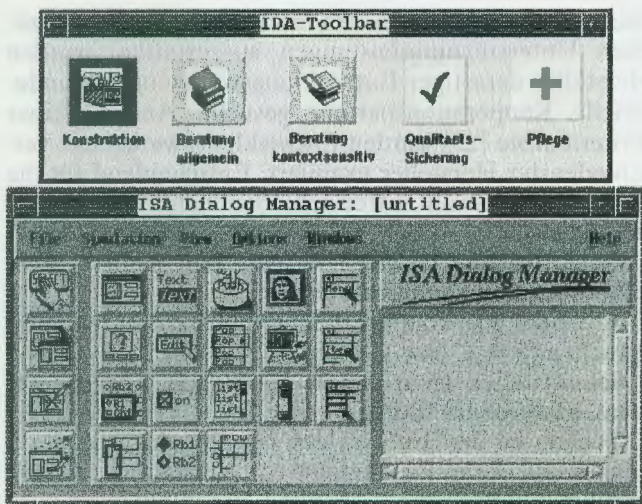


Abb. 2. Einsatz der IDA-Entwicklungsumgebung

zungsoberfläche einer Software-Applikation. Um den Implementierungsaufwand zu reduzieren, kann der Entwickler auf bereits vorgefertigte, nach ergonomischen Gesichtspunkten gestaltete Dialogbausteine zurückgreifen. Dazu nutzt er eine objekt-orientierte Klassenbibliothek. Zugang zu dieser Klassenbibliothek bietet ihm das Unterstützungswerkzeug „Konstruktion“, das über ein Icon in der „IDA-Toolbar“ (oberes Fenster) aktivierbar ist. Die Klassenbibliothek hat für den Entwickler eine Art „Baukastenfunktion“, indem sie ihm für die Erstellung komplexer graphischer Benutzungsoberflächen vorgefertigte Dialogbausteine zur Verfügung stellt. Durch leistungsfähige Navigations- und Präsentationsinstrumente (z.B. graphische Browser, Suchfunktionen, Animationen) wird der Entwickler beim Finden geeigneter Dialogbausteine unterstützt.

Ist dem Entwickler beispielsweise der Verwendungszweck eines ausgewählten Dialogbausteines nicht bekannt, kann er die online verfügbare multimediale „kontextsensitive Beratung“ mittels Icon in der „IDA-Toolbar“ aktivieren. Die Beratung setzt zur Wissenspräsentation Texte, Graphiken, Animationen, Sprache und interaktive Beispiele ein und bietet vielfältige Navigationstechniken zum Finden gewünschter Themeninhalte (z.B. Hotlinks, Browser, Query-Functions). Neben dem Verwendungszweck von vorgefertigten Dialogbausteinen kann sich der Entwickler auch in anschaulicher Weise ergonomisches Grundwissen zur Implementierung von graphischen Benutzungsoberflächen vermitteln lassen. Der Einstieg in die „allgemeine Beratung“ erfolgt durch Aktivieren des entsprechenden Icons in der „IDA-Toolbar“.

Möchte der Entwickler seine bisherigen Implementierungen einer automatischen software-ergonomischen Überprüfung unterziehen, kann er die wissensbasierte „Qualitätssicherung“ mittels Icon in der „IDA-Toolbar“ aktivieren. Sie wertet die erbrachten Implementierungen aus und weist auf erkannte Verstöße gegen software-ergonomische Anforderungen hin. Benötigt der Entwickler weitergehende Erklärungen zu fest-

gestellten Verstößen, kann er kontextspezifisch die Beratung oder Konstruktionsunterstützung aktivieren.

Die getrennte Präsentation der Komponenten der IDA-Entwicklungsumgebung – der Unterstützungswerkzeuge und des Entwicklungswerkzeuges – ist bedingt durch die prototypische Realisierung im Rahmen eines Forschungsvorhabens. Die enge Kooperation mit dem Hersteller des Entwicklungswerkzeuges wird es aber ermöglichen, die verschiedenen Leistungen der Unterstützungswerkzeuge in das Entwicklungswerkzeug zu integrieren. Anhand des Einsatzbeispiels sollte auch deutlich werden, daß die verschiedenen Unterstützungsleistungen der IDA-Entwicklungsumgebung in enger Wechselwirkung stehen (z.B. Konstruktionsunterstützung erzeugt Bedarf an Beratung; Beratung gibt Hinweise auf vorhandene Dialogbausteine in der Konstruktionsbibliothek; Qualitätssicherung animiert zum Einstieg in die Beratung). Es bestehen vielfältige Zugänge zu den einzelnen Unterstützungswerkzeugen, womit dem Entwickler eine seinem persönlichen Arbeitsstil entsprechende Nutzung eröffnet wird.

3. Stand der Forschung

Eine am Beginn des Vorhabens durchgeführte Auswertung einschlägiger Forschungsvorhaben hat gezeigt, daß die verfolgten Ziele und Fragestellungen bereits Gegenstand von Forschungsaktivitäten waren und sind. Gleichzeitig zeigte sich aber auch, daß viele der im Rahmen des IDA-Projektes verfolgten Ziele noch nicht realisiert worden sind und daher ein entsprechender Forschungsbedarf besteht. Dies gilt vor allem für die Realisierung des Zusammenspiels verschiedener Formen der Designunterstützung, die Art der Wissenspräsentation sowie für Art und Umfang der zu vermittelnden Wissensinhalte. Ein guter Überblick über neue Forschungsrichtungen im Bereich derartiger Entwicklungswerkzeuge findet sich in [20]. Der folgende kurze Überblick über wichtige Forschungsvorhaben zu Fragestellungen dieses Projektes soll einen ersten Eindruck bzgl. bisher realisierter bzw. geplanter Lösungsansätze liefern.

Am Department of Computer Science der University of Colorado in Boulder werden seit einigen Jahren sogenannte Kritiksyste (critiquing systems) entwickelt. Es handelt sich dabei um rechnerbasierte Systeme, die für unterschiedlichste Aufgabenstellungen dem Benutzer Wissen vermitteln, um ihn bei der Aufgabendurchführung zu unterstützen (enabling systems). Vor allem im Bereich Designunterstützung wurde eine Reihe von prototypischen Systemen entwickelt, die unter Verwendung von wissensbasierten Techniken, Hypertext-Systemen und Bibliotheken wiederverwendbarer Designergebnisse dem Designer Unterstützung bieten. Beispiele für realisierte Designunterstützungssysteme sind das System FRAMER, das einen Lisp Programmierer bei der Erstellung von fensterbasierten Benutzungsoberflächen unterstützt [9]; das System JANUS, das einen Designer von Küchen in Fragen des „guten Küchendesigns“ unterstützt [9], das System HYDRA, das eine Weiterentwicklung von JANUS ist und eine

Spezifikationskomponente zur Erfassung von Designanforderungen beinhaltet [10]. Die Ideen des IDA-Vorhabens wurden maßgeblich von den vorhin beschriebenen Arbeiten beeinflusst. Vor allem die Ideen, designunterstützende Systeme nicht ausschließlich auf wissensbasierte Kritik zu beschränken, sondern um Konzepte der Beratung und Wiederverwendung von Designergebnissen zu ergänzen, wurden aufgegriffen.

Am Department of Computer and Information Science der University of Linköping wurde ein Entwicklungswerkzeug mit einem Expertensystem verbunden, das mittels einer Wissensbasis die entworfenen graphischen Benutzungsoberflächen analysiert. Die Wissensbasis beinhaltet vorwiegend Gestaltungsforderungen aus dem Motif Style Guide [21] und allgemeinen Gestaltungsrichtlinien [26]. Ergebnis des Qualitätssicherungsprozesses sind Kommentare, die der Entwickler anschließend berücksichtigen kann. Die Qualitätssicherung bezieht sich nur auf die Präsentationsebene der Benutzungsoberfläche. Primäres Ziel dieses Projektes war es, ein passives Evaluationsinstrument in ein Entwicklungswerkzeug zu integrieren. In [16] wird über die gewählte Architektur, den Aufbau der Wissensbasis und den Einsatz bei einer Beispielanwendung berichtet. Die Arbeiten beeinflussten maßgeblich die Architektur der wissensbasierten Qualitätssicherung des IDA-Vorhabens. Der IDA-Ansatz geht aber über eine reine wissensbasierte Qualitätssicherung hinaus.

Am College of Computing des Georgia Institute of Technology in Atlanta wird seit Jahren an der automatischen Generierung von Benutzungsoberflächen gearbeitet. Basierend auf einem objekt-orientierten Datenmodell werden mit Hilfe einer Inference Engine automatisch Benutzungsoberflächen generiert und mittels eines Entwicklungswerkzeuges visualisiert. Die Inference Engine beinhaltet eine Reihe von Regeln, die vorwiegend auf dem Open Look Style Guide [28] basieren und die für die automatische und Style Guide-konforme Generierung verantwortlich sind. Auch hier bezieht sich die Unterstützung nur auf die Präsentationsebene der Benutzungsoberfläche. In [1] wird die Architektur beschrieben und es werden ausführliche Beispiele für abgeleitete Regeln gegeben. Das IDA-Vorhaben verfolgt keine automatische Generierung aus einem Datenmodell. Insofern gibt es eine klare Abgrenzung zu den Intentionen dieser Arbeiten. Berücksichtigt wurden aber die Anregungen bzgl. der Erweiterung bestehender Datenmodellierungsansätze in Richtung bessere Eignung für die Entwicklung von Benutzungsoberflächen.

Am Institut für Arbeitswissenschaft und Technologiemanagement der Universität Stuttgart und am Fraunhofer-Institut für Arbeitswirtschaft und Organisation in Stuttgart wurde das Projekt „Unterstützungswerkzeuge zur benutzergerechten Gestaltung der Mensch-Computer-Schnittstelle“ durchgeführt. Ziel war die Integration von Software-Engineering-Techniken und Techniken des Benutzungsoberflächendesigns. Ausgangspunkt war ein Entity-Relationship-Datenmodell, ergänzt um sogenannte Aufgabensichten (Daten die zur Bearbeitung einer Aufgabe notwendig sind)

und Dialognetzen (Navigationsstruktur des Dialogablaufes). Aus diesen Informationen wird mittels eines regelbasierten Systems automatisch die erste Version der Benutzungsoberfläche – unter Verwendung von software-ergonomischen Designregeln – generiert. Diese Oberfläche wird dann mittels eines Entwicklungswerkzeuges visualisiert und steht zur unumgänglichen weiteren Bearbeitung zur Verfügung. In [15] findet sich eine kurze Beschreibung der gewählten Architektur sowie der Vorgehensweise bei der Definition und automatischen Generierung der Benutzungsoberfläche. Auch hier gilt für das IDA-Vorhaben das zum vorigen Forschungsansatz gesagte. Die Einsatzerfahrungen dieses Vorhabens bzgl. Entwicklungswerkzeuge und Expertensysteme beeinflussten maßgeblich die Auswahlentscheidungen des IDA-Vorhabens.

Am Fachbereich Informatik der Universität Oldenburg wird das Projekt EXPOSE durchgeführt. Ziel ist die Unterstützung des Entwicklers von Benutzungsoberflächen mit ergonomischem Wissen. Dazu wurde eine Methode MUSE (4-Phasenkonzept zur Gestaltung der Benutzungsoberfläche) und ein wissensbasiertes Beratungswerkzeug entwickelt. Dieses wurde mit einem Entwicklungswerkzeug gekoppelt, das als Entwicklungswerkzeug dient. Das wissensbasierte Beratungswerkzeug stellt sowohl Bewertungs- als auch Beratungsleistungen zur Verfügung [11]. Eine wesentliche Stärke dieses Ansatzes liegt in der expliziten Berücksichtigung des Aufgabenbezuges. Dieses Vorhaben geht vom Anspruch der methodischen Unterstützung deutlich über das IDA-Vorhaben hinaus, da es alle Phasen des Software-Lebenszyklus im Sinne einer ergonomischen Beratung erfaßt. Das IDA-Vorhaben beschränkt sich primär auf die Unterstützung der Implementierungsphase der Benutzungsoberfläche.

Am Lehrstuhl für Software-Technik der Ruhr-Universität Bochum wird ein Projekt namens JANUS durchgeführt. Dabei soll aus einem objekt-orientierten Datenmodell mittels verschiedener wissensbasierter Generatoren (enthalten Wissen bzgl. Ergonomie, Benutzermodell, Konventionen, Selbstbild) automatisch eine Beschreibung einer Benutzungsoberfläche generiert werden. Diese wird dann mit Hilfe eines Entwicklungswerkzeuges visualisiert und steht anschließend dem Entwickler zur weiteren Bearbeitung zur Verfügung [3]. Dabei soll ein iteratives Design, unter Einbeziehung von Endbenutzern, unterstützt werden. Vom Ansatz unterscheidet sich dieses Vorhaben wieder grundlegend vom IDA-Projekt, da auch hier die automatische Generierung aus dem Datenmodell erfolgt. Anregungen für das IDA-Vorhaben ergeben sich aus den Erweiterungen des objekt-orientierten Ansatzes um Notationen für das Benutzungsschnittstellendesign.

Am Institut für Angewandte Psychologie der Humboldt-Universität Berlin wird im Rahmen des Projektes „Wissensgewinnung, -modellierung und -darstellung und ihre Anwendung bei rechnerbasierten Unterstützungswerkzeugen (WEDA)“ ein Interface-Ratgeber entwickelt. Dieser soll sowohl in einer Papierversion als Textbuch [12] als auch als rechnerbasiertes Unterstützungswerkzeug verfügbar sein. Zusätzlich wurden

im Rahmen des Projektes eine Reihe von empirischen Untersuchungen zu vorhandenen software-ergonomischen Wissensdefiziten der Software-Entwickler durchgeführt [4]. Die empirischen Erkenntnisse dieser Untersuchungen waren mit ein Anlaß für die Durchführung des IDA-Vorhabens. Das WEDA-Vorhaben beschränkt sich ausschließlich auf die multimediale Wissensvermittlung, die außerdem nicht in ein Entwicklungswerkzeug integriert wird. In diesen Punkten geht der IDA-Ansatz über die Möglichkeiten zur Wissensvermittlung hinaus.

Am Fachbereich Informatik der Technischen Hochschule Darmstadt wird im Rahmen des Vorhabens Diades-II ein objekt-orientiertes Entwurfswerkzeug für interaktive Benutzungsschnittstellen auf der Grundlage koagierender Agenten entwickelt [8]. Das Entwurfswerkzeug unterstützt einen inkrementellen strukturierten Entwurf, der durch ein Koagieren von Entwickler und Werkzeug realisiert wird. Das System arbeitet auf der Grundlage der Entwicklerangaben selbständig am Entwurf mittels intelligenter Agenten mit. Die Integration eines Einschätzungsagenten ermöglicht nicht nur die Bewertung der Entwurfsentscheidungen des Entwicklers, sondern auch, daß das Verhalten von Diades-II sich durch das Koagieren seiner Agenten selbst an Ergonomievorgaben orientiert. Diades-II ist als wissensbasiertes Mehr-Agenten-System geplant, von dem bisher primär Überlegungen zum Architekturkonzept verfügbar sind. Da bis jetzt nur Konzepte zu Diades-II vorliegen, konnte eine weitergehende Berücksichtigung im Rahmen des IDA-Vorhabens nicht erfolgen.

Wie die obige Aufstellung zeigt, gibt es zum Themenbereich Unterstützungswerkzeuge für Entwickler von Benutzungsoberflächen eine Reihe von Projekten mit ähnlichen Fragestellungen, aber recht unterschiedlichen Lösungsstrategien. Eine typische Lösungsstrategie der beschriebenen Vorhaben ist die automatische Generierung von Benutzungsoberflächen aus dem Daten- und Funktionenmodell, wie sie im Bereich des Software-Engineering erstellt werden. Im Rahmen des IDA-Projektes wird keine automatische Generierung der Benutzungsoberflächen durchgeführt, sondern der Entwickler ist aktiv und wird mittels einer Kombination von drei unterschiedlichen Unterstützungswerkzeugen bei seinen Entwicklungsaktivitäten unterstützt. Es wird somit ein komplementärer Ansatz verfolgt, da auch von seiten der Vertreter der Forschungsstrategie der automatischen Generierung nicht bestritten wird, daß die von ihnen erzielten Designergebnisse noch einer Nachbearbeitung durch den Entwickler unterzogen werden müssen. Eine weitere typische Lösungsstrategie der beschriebenen Vorhaben besteht in einer ausschließlichen wissensbasierten Unterstützung der Entwickler. Demgegenüber erfolgt im IDA-Projekt keine **ausschließliche Beschränkung** auf wissensbasierte Konzepte, sondern auch die Nutzung von Multimedia-Techniken (Beratung) und Klassenkonzepten der Objekt-Orientierung (Bibliothek von Dialogbausteinen). Insbesondere das Zusammenspiel dieser drei Techniken für Zwecke der Designunterstützung ist eine Besonderheit des IDA-Vorhabens und unterscheidet es von allen

bisher bekannten Forschungsansätzen im Bereich Designunterstützung für Entwickler von Benutzungsoberflächen.

Im IDA-Vorhaben wird somit ein möglicher Weg zur konsequenten Umsetzung von software-ergonomischen Gestaltungsforderungen, direkt im Implementierungsprozeß von graphischen Benutzungsoberflächen, aufgezeigt. Diese prospektive Gestaltungsorientierung ist ein entscheidender Vorteil dieses Ansatzes gegenüber der korrekativen Gestaltungsorientierung traditioneller software-ergonomischer Evaluationsverfahren, wie beispielsweise EVADIS [22, 24].

4. Die einzelnen Unterstützungswerkzeuge der IDA-Entwicklungsumgebung

Nachdem der Einsatzzweck der verschiedenen Unterstützungswerkzeuge der IDA-Entwicklungsumgebung im 2. Abschnitt, anhand typischer Einsatzsituationen, kurz erläutert worden ist und eine Abgrenzung zu verwandten Forschungsansätzen im Abschn.3 vorgenommen wurde, soll im folgenden jedes Unterstützungswerkzeug näher vorgestellt werden. Neben einer Beschreibung der zugrunde gelegten Konzepte und der damit verfolgten Ziele wird das funktionale Spektrum jedes Unterstützungswerkzeuges beschrieben. Zusätzlich werden Ergebnisse einer umfassenden Evaluation dargestellt. Diese Ergebnisse basieren auf einer zweitägigen Evaluation, die mit 10 Entwicklern der Software AG sowie 11 Studenten (mit Praxiserfahrung) der Fachhochschule Darmstadt erfolgte. Als Evaluationsmethoden wurden Benutzerbefragungen mit Fragebögen sowie Beobachtungen mittels Beobachtungsprotokollen und automatischen Systemaufzeichnungen (logfile recordings) angewandt. Die Probanden mußten eine typische Aufgabenstellung aus dem Bereich der Entwicklung von graphischen Benutzungsoberflächen, unter Nutzung der unterschiedlichen Werkzeuge der IDA-Entwicklungsumgebung, bearbeiten. Dabei wurden sie von einem Untersuchungsleiter beobachtet und gleichzeitig wurden wesentliche Interaktionen rechnerbasiert aufgezeichnet. Anschließend mußten die Probanden einen umfassenden Fragebogen beantworten, der Fragestellungen zu den wesentlichen Funktionen und Eigenschaften der Unterstützungswerkzeuge beinhaltete.

4.1 Konstruktionsunterstützung mittels einer objekt-orientierten Dialogbausteinbibliothek

Die Konstruktionsunterstützung basiert auf einer Klassenbibliothek von vordefinierten Dialogbausteinen. Was waren die Motive für die Verwendung von objekt-orientierten Techniken? Die Vererbungsmöglichkeiten der objekt-orientierten Techniken eröffnen eine wirksame Unterstützung des Entwicklers in Fragen der ergonomischen Gestaltung (usability) und der Wiederverwendbarkeit (reusability). Die Unterstützung bei der ergonomischen Gestaltung beruht darauf, daß die vordefinierten Dialogbausteine entsprechend den Gestal-

tungsforderungen der Normen [14], Style Guides [13, 18, 21] und herstellerunabhängigen Gestaltungsrichtlinien [17] vorgefertigt und dem Entwickler verfügbar gemacht werden. Die vorgefertigten Dialogbausteine enthalten nicht nur einfache und komplexe Dialogobjekte (z.B. Push Button, List Box, Tabelle, Dialogfenster), sondern auch typische Dialogabläufe, wie Metadialoge der Interaktion (z.B. Hilfe, Fehler, Auswahl- und Speicherdialoge). Dialogbausteine können auch komplette, anwendungsspezifische Fensterarchitekturen repräsentieren (z.B. Hauptfenster mit Tochterfenstern und Dialogfenstern), die auf einem der gängigen Architekturkonzepte basieren (Single Document Interface, Multiple Document Interface, Workplace Environment).

Der Aufbau des Bibliotheksinhaltes erfolgte mittels der objekt-orientierten Mechanismen des Entwicklungswerkzeuges. Grundlage für den Entwurf der vordefinierten Dialogbausteine sind die generischen Interaktionsobjekte des Entwicklungswerkzeuges. Sie beinhalten eine Vielzahl von Default-Attributen, die ihr standardmäßiges Aussehen festlegen. Mittels dieser generischen Interaktionsobjekte können einfache oder komplexe Dialogbausteine mit Hilfe des Entwicklungswerkzeuges definiert und in die Bibliothek als Objektklassen eingebracht werden. Dabei werden die Default-Attribute überschrieben, um ein gewünschtes Verhalten zu definieren (z.B. Normen- oder Style Guide-Konformität). Jeder Dialogbaustein kann seinerseits als Basis zur Definition von weiteren anwendungsspezifischen Dialogbausteinen dienen (Bausteinhierarchie). Der Vererbungsmechanismus beschränkt sich nicht auf den statischen Teil der Benutzungsoberfläche, d.h. auf die Interaktionsobjekte, sondern umfaßt auch den dynamischen Teil, d.h. den Dialogablauf. Dazu ist es er-

forderlich, daß mittels einer ereignisgesteuerten Dialogsprache (z.B. Regelsprache) das Dialogverhalten beschrieben werden kann und diese Dialogabläufe auch an Dialogbausteine gebunden werden können. Im Rahmen des Vorhabens wurde in enger Kooperation mit den Industriepartnern für ausgewählte Anwendungsdomänen eine Reihe von einfachen und komplexen Dialogbausteinen vordefiniert. Die Ausrichtung der Bibliotheksinhalte auf bestimmte Anwendungsdomänen ist eine unabdingbare Voraussetzung für ihre spätere Nützlichkeit. Ein wichtiger Aspekt dabei ist, das richtige Maß an anwendungsspezifischer Orientierung und notwendiger Allgemeinheit für die spätere Wiederverwendbarkeit zu finden. Dies ist nur möglich, wenn die Entwicklung der Dialogbausteine in iterativer Weise und in enger Kooperation mit den jeweiligen Anwendungsentwicklern erfolgt. Die Aufnahme eines Dialogbausteines in die Bibliothek ist eine sorgfältig zu prüfende Entscheidung, da mit dem Vererbungskonzept auch die Gefahr der Duplizität von falschen Designentscheidungen verbunden ist.

Eine weiteres wichtiges Kriterium für den Erfolg der Klassenbibliothek stellt die Zugänglichkeit (Information Retrieval) zu den vordefinierten Dialogbausteinen dar. Die Zugänglichkeit entscheidet über die Nützlichkeit für den Entwickler. Daher wurden eine Reihe von unterschiedlichen Navigations-, Retrieval- und Präsentationskonzepten realisiert, mit dem Ziel, ein schnelles Wiederfinden der Dialogbausteine zu ermöglichen. Abbildung 3 zeigt die dazu entworfene Benutzungsoberfläche der Konstruktionsunterstützung.

Im oberen Bereich der Benutzungsoberfläche des Hauptfensters befindet sich die Hauptmenüleiste, die Zugang zu allen Funktionen bietet. Die Evaluation erbrachte eine Fülle von Hinweisen bzgl. der Benennung

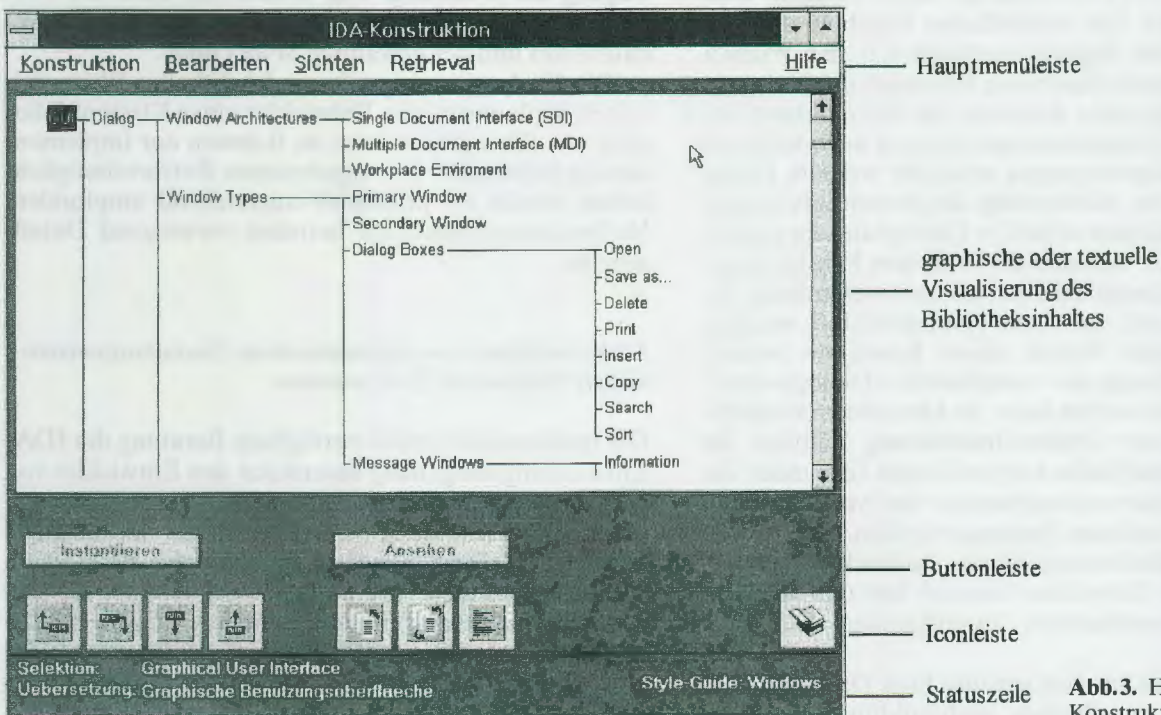


Abb. 3. Hauptfenster der Konstruktionsunterstützung

und Platzierung von Menüeinträgen. Sie wurden vollständig beim Redesign berücksichtigt.

Der Menüeintrag „Konstruktion“ ermöglicht das Beenden der Anwendung. Unter dem Menüeintrag „Bearbeiten“ können folgende Funktionen aktiviert werden:

- Klassen von Dialogbausteinen anlegen bzw. löschen
- Präsentation des „Look and Feel“ von Dialogbausteinen
- Zugriff auf Beratungsleistungen
- Erzeugen von Instanzen der Dialogklassen
- Einstellmöglichkeiten und Pflegeinformationen

Über die Menüeinträge „Klasse anlegen“ und „Klasse löschen“ können neue Klassen von Dialogbausteinen angelegt bzw. bestehende gelöscht werden. Bevor der Entwickler einen gefundenen Dialogbaustein verwenden möchte, wird er sich in der Regel dessen „Look and Feel“ näher ansehen wollen. Mittels des Menüeintrages „Ansehen“ kann er sich einen zuvor im Baum oder in der Liste selektierten Dialogbaustein in einer animierten Darstellung visualisieren lassen. Die Evaluation zeigte, daß eine animierte Darstellung der Dialogklassen vor allem für neue Nutzer der Konstruktionsunterstützung als hilfreich für den Auswahlprozeß angesehen wurde, während für regelmäßige Benutzer die Nützlichkeit als eher gering eingeschätzt wurde.

Werden dem Entwickler trotz animierter Darstellung wesentliche Konzepte des selektierten Dialogbausteines nicht klar, kann er über den Menüeintrag „Beratung“ kontextsensitiv in die Beratung einsteigen (siehe Abschn.4.2). Während der Evaluation wurde von dieser Möglichkeit intensiv Gebrauch gemacht und als ein wesentlicher Zugang zur Beratung eingestuft.

Kommt der Entwickler zur Erkenntnis, daß der gefundene Dialogbaustein seinen Wünschen entspricht, kann er mittels des Menüeintrages „Instantiieren“ eine Instanz erzeugen. Ein wesentliches Ergebnis der Kooperation mit den Industriepartnern war der Wunsch nach einem parametrisierbaren Instantiierungsvorgang. Das heißt, ausgewählte Attribute der Objektklasse des vordefinierten Dialogbausteines können noch während des Instantiierungsvorganges angepaßt werden. Daher erscheint nach der Aktivierung des Instantiierungsvorganges ein jeweils zum aktuellen Dialogbaustein gehöriges Dialogfenster. Anhand der in diesem Fenster angebotenen Einstellmöglichkeiten können wesentliche Attribute des „Look and Feel“ parametrisiert werden. Der entscheidende Vorteil dieses Konzeptes besteht darin, daß die Menge der vordefinierten Dialogbausteine klein gehalten werden kann. So kann einem wesentlichen Problem der Objekt-Orientierung, nämlich die mangelnde Überschaubarkeit und damit verbunden die mangelnde Wiederverwendbarkeit von vordefinierten Objektklassen, wirksam begegnet werden. Dies wurde durch die Evaluationsergebnisse eindrucksvoll unterstrichen, da alle Entwickler intensiv von der Möglichkeit der parametrisierten Instantiierung Gebrauch machten.

Das Wiederfinden von gewünschten Dialogbausteinen wird durch verschiedene Suchfunktionen, die über

den Hauptmenüeintrag „Retrieval“ angeboten werden, unterstützt. Es ist eine Suche nach „Name . . .“, „Einsatzzweck . . .“ und „verwendet in . . .“ möglich. Voraussetzung für spätere Retrievalmöglichkeiten ist aber, daß bei der Neuaufnahme eines Dialogbausteines in die Bibliothek entsprechende Namenskonventionen beachtet sowie Attribute über Einsatzzweck und Verwendung mitdefiniert werden. Die Evaluation zeigt, daß neben den oben genannten Suchfunktionen auch eine Kombinationssuche mit logischen Verknüpfungsmöglichkeiten gefordert wurde. Diese wurde im Zuge des Redesigns realisiert.

Im mittleren Bereich der Abb.3 wird die Strukturierung der Dialogbausteine in der Bibliothek visualisiert. Als graphische Darstellung wird ein Baum angezeigt (graphischer Browser). Dabei repräsentieren die einzelnen Blätter des Baumes die in der Bibliothek vorhandenen Dialogbausteine. Neben der graphischen ist auch eine textuelle Darstellung möglich. Diese repräsentiert die vorhandenen Dialogbausteine mittels einer alphabetisch sortierten Liste. Über den Hauptmenüeintrag „Sichten“ kann zwischen den verschiedenen Darstellungsarten gewählt werden. Zusätzlich besteht die Möglichkeit, durch die Definition von Teilbäumen gewünschte Ausschnitte aus dem Gesamtbaum zu extrahieren und damit die Übersichtlichkeit zu erhöhen. Die Evaluation zeigte die eindeutige Überlegenheit des graphischen Browser gegenüber der Listendarstellung als Instrument zur Visualisierung der Strukturierung der Inhalte.

Im unteren Bereich des Hauptfensters der Konstruktionsunterstützung befindet sich eine Buttonleiste, die einen schnellen Zugang zu den beiden wichtigsten Funktionen „Ansehen“ und „Instantiieren“ zuläßt. Die Iconleiste ermöglicht eine schnelle Navigation in der Struktur der Bibliothek sowie einen kontextsensitiven Zugang zur Beratung. Die Statuszeile dient zur Anzeige des deutschen Begriffs des selektierten Dialogbausteines und des gewählten Style Guide.

Die Evaluation zeigte in eindrucksvoller Weise den hohen Stellenwert, den Entwickler einer Klassenbibliothek von Dialogbausteinen im Rahmen der Implementierung beimessen. Die angebotenen Retrievalmöglichkeiten wurde als prinzipiell ausreichend empfunden, Verbesserungsvorschläge betrafen vorwiegend Detailsaspekte.

3.2 Vermittlung von ergonomischem Gestaltungswissen mittels Multimedia-Dokumenten

Die multimediale online verfügbare Beratung der IDA-Entwicklungsumgebung unterstützt den Entwickler von graphischen Benutzungsoberflächen in Fragen der ergonomischen Gestaltung von graphischen Benutzungsoberflächen. Dazu werden ergonomische Richtlinien, wie sie in Normen [14], Style Guides [13, 18, 21] und herstellerunabhängigen Publikationen [17] zu finden sind vermittelt. Dies eröffnet dem Entwickler die Möglichkeit, sich Wissen im Zuge des Entwurfsprozesses bedarfsorientiert anzueignen (learning on demand), bei

gleichzeitiger Unterstützung eines Wissenstransfers auf zukünftige ähnliche Aufgabenstellungen. Langfristig wird jedoch ein persistenter Wissenstransfer angestrebt, so daß sich der Entwickler immer seltener dieses Unterstützungswerkzeuges bedienen muß.

Was waren die Motive für die Wahl einer multimedialen Präsentation von Ergonomiewissen? Multimedia bedeutet in diesem Zusammenhang eine hypertextartige Struktur der Vernetzung von Beratungsinhalten, die mittels Text, Graphiken, Sprache, Animationen oder Videosequenzen präsentiert werden. Im Bereich der Entwicklung und Gestaltung von Benutzungsoberflächen spielen dynamische Aspekte eine wichtige Rolle. Beispielsweise läßt sich die Gestaltung von typischen Dialogabläufen, das Zusammenspiel der verschiedenen Fenstertypen der gängigen Fensterarchitekturen oder das „Look and Feel“ komplexerer Interaktionsobjekte in hervorragender Weise mittels multimedialer Techniken vermitteln. Rein statische Systeme, wie dies beispielsweise gängige Hilfesysteme oder Style Guides in Papierform darstellen, sind für diese Zwecke denkbar schlecht geeignet. Ein guter Überblick über verwandte Ansätze in [29] zeigt, daß multimediale Präsentationsformen im Bereich der Gestaltung von Benutzungsoberflächen zunehmend Verbreitung finden.

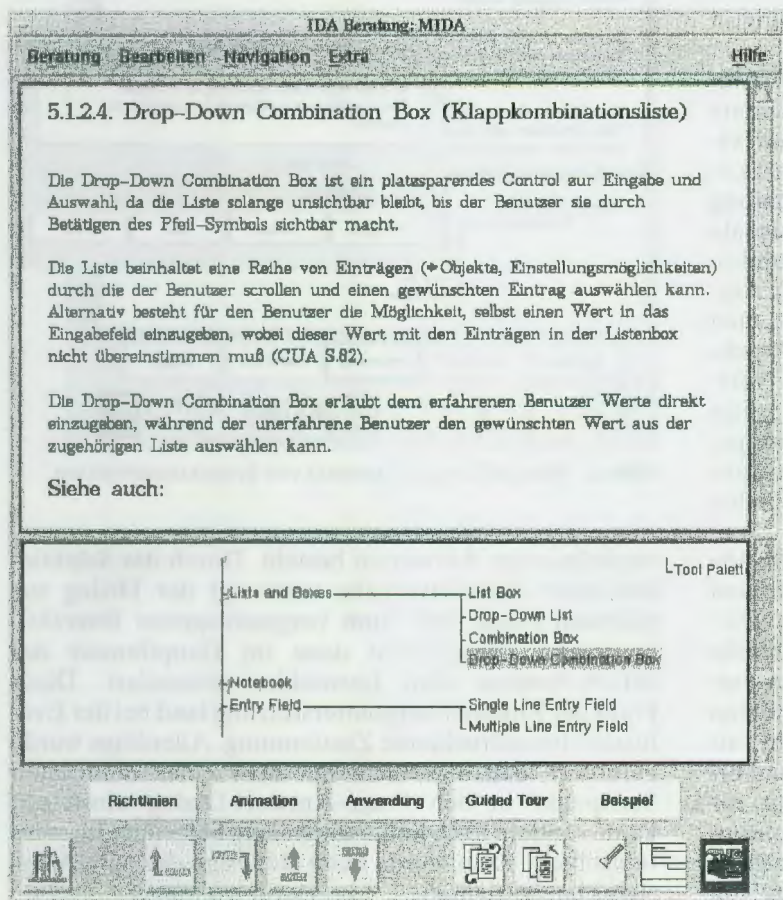
Die Beratung wurde auf den zwei Plattformen OSF Motif und Microsoft Windows, mit unterschiedlichen Entwicklungswerkzeugen und basierend auf unterschiedlichen Metaphern sowie Navigations- und Präsen-

tationskonzepten, implementiert. Ziel dieser Vorgehensweise war es herauszufinden, welche Metapher bessere Orientierungsunterstützung bietet und ob eine traditionelle Form der Navigation und der Funktionspräsentation oder eine innovative Form Akzeptanz finden. Mit Hilfe einer empirischen Evaluation wurden diese Fragestellungen überprüft.

3.2.1 MIDA – die IDA-Beratung unter Motif. Die Realisierung der IDA-Beratung unter Motif erfolgt unter der Bezeichnung MIDA (Multimedia Interface Design Documents for Advice). Als Entwicklungswerkzeuge kamen IDS (ISA), MetaCard (MetaCard) und XRecorder (Highland) zum Einsatz.

Dieses Beratungssystem basiert auf einer traditionellen Form der Funktions- und Wissenspräsentation. Unter Berücksichtigung der für Motif-Anwendungen üblichen Funktionspräsentation, werden alle Funktionen über ein permanent sichtbares Hauptmenü mit Pull-Down-Menüs angeboten. Zusätzlich wird über eine Button- und Iconleiste ein schneller Zugang zu wesentlichen Funktionen ermöglicht. Als Metapher zur Wissenspräsentation wurde die traditionelle Buch-Metapher gewählt.

Abbildung 4 stellt das zentrale Hauptfenster der MIDA-Beratung dar, von dem aus alle weiteren Tochterfenster aktiviert werden (Single Document Interface). Die Hauptmenüleiste beinhaltet alle verfügbaren Funktionen des Beratungssystems. Die Evaluation er-



— Hauptmenüleiste

— Beratungsinhalte

— graphisches
Inhaltsverzeichnis

— Buttonleiste

— Iconleiste

Abb. 4. Eine Beratungsseite im MIDA-Hauptfenster

brachte eine Fülle von Hinweisen bzgl. der Benennung und Platzierung von Menüeinträgen. Sie wurden vollständig beim Redesign berücksichtigt.

Der Hauptmenüeintrag „Beratung“ erlaubt das Beenden der Anwendung. Unter dem Hauptmenüeintrag „Bearbeiten“ sind folgende Funktionen aktivierbar:

- Suchen nach Inhalten
- Darstellung von Richtlinien
- Starten von Animationen
- Starten eines Abfragedialoges bzgl. des Einsatzes von Interaktionsobjekten
- Starten von Guided Tours
- Starten von interaktiven Beispielen

Im folgenden werden jene Funktionen, die zur Wissensvermittlung wesentlich sind, näher dargestellt. Über den Menüeintrag „Richtlinien“ kann zu jedem Beratungsthema, eine animierte Aufbereitung der ergonomischen Richtlinien aktiviert werden. Diese erscheint in einem eigenen Fenster. Abbildung 5 zeigt beispielhaft eine derartige Richtlinienpräsentation. Die Richtlinien werden mittels Mausklick in der interaktiven Graphik, die auf der linken Seite dargestellt wird, angezeigt. Beispielsweise werden durch Mausklick auf die Karteireiter des Notebooks alle Richtlinien bezüglich der Verwendung von Karteireitern dargestellt. Neben dieser interaktiven Präsentation können alle Richtlinien auch in einer Gesamtübersicht dargestellt werden. Diese Form der interaktiven Richtlinienpräsentation fand große Zustimmung bei der Evaluation. Allerdings wurde eine klarere visuelle Abgrenzung der Quellen der einzelnen Richtlinien gewünscht, sowie eine strikte Trennung in Muß- und Kann-Richtlinien.

Über den Menüeintrag „Animation“ werden Animationen gestartet. Diese vermitteln unter Einsatz von Text, Sprache, Bildschirmaufzeichnungen und Videos Dialogabläufe oder erklären komplexe Interaktionsobjekte, bei denen eine reine textuelle Erklärung nur schwer möglich ist. Die Ausgabe der Audiosignale kann entweder auf Lautsprechern oder auf Kopfhörern erfolgen. Die Evaluationsergebnisse bzgl. der Animationen waren sehr zwiespältig. Zum einen fand man sie hilfreich zur Vermittlung von dynamischen Aspekten, bezweifelte aber eine mittel- bis langfristige Nützlichkeit. Vielfach wurden die Animationen als „nette Spielereien“ bezeichnet, die für den Anfänger von gewissem Wert sein mögen, von erfahrenen Benutzern aber als zu umständlich und langwierig empfunden werden.

Über den Menüeintrag „Anwendung“ wird ein Abfragedialog zur Unterstützung bei der Auswahl und dem Einsatz von Interaktionsobjekten angeboten. Dieser kann als Frage-Antwort-Dialog oder als graphische Übersicht in Form eines Entscheidungsbaumes gestartet werden (abhängig von der individuell einstellbaren Konfiguration). Sowohl die graphische Übersicht, als auch der Frage-Antwort-Dialog sind interaktiv und ermöglichen ein angeleitetes Verzweigen zu einem, für den konkreten Einsatzfall sinnvolles Interaktionsobjekt. Abbildung 6 zeigt beispielhaft einen Frage-Antwort-Dialog, der aus einer Frage und einer Reihe von

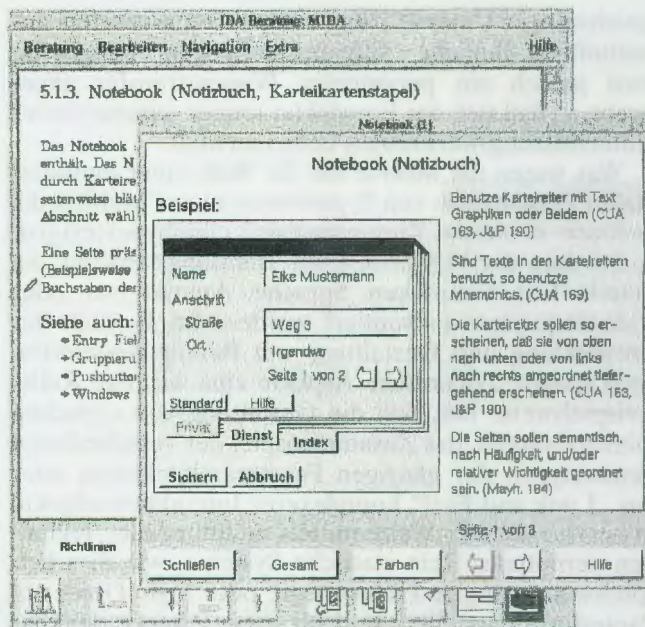


Abb.5. Interaktive Präsentation von Richtlinien

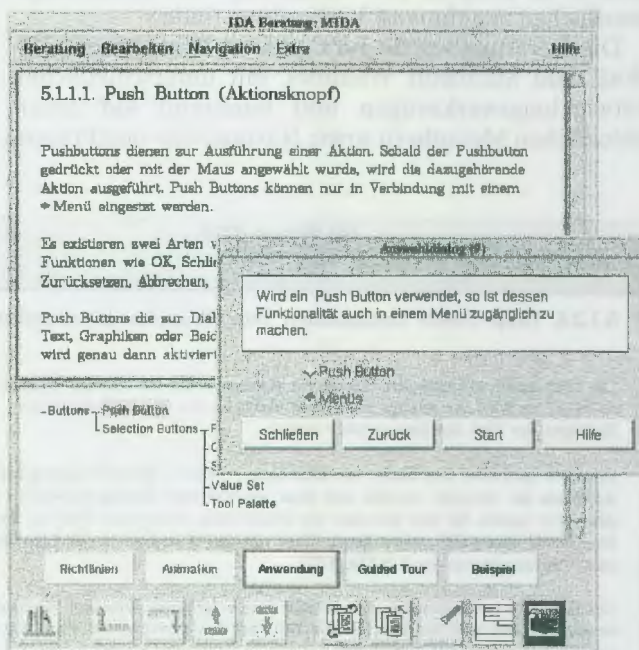


Abb.6. Abfragedialog zur Auswahl von Interaktionsobjekten

vordefinierten Antworten besteht. Durch das Selektieren einer Antwortvorgabe verzweigt der Dialog zur nächsten Frage, bzw. zum vorgeschlagenen Interaktionsobjekt. Dieses wird dann im Hauptfenster des MIDA-Systems dem Entwickler präsentiert. Diese Form der Entscheidungsunterstützung fand bei der Evaluation beeindruckende Zustimmung. Allerdings wurde gefordert, daß man nicht nur über einen definierten Startpunkt in den Frage-Antwort-Dialog einsteigen kann, sondern auch kontextabhängig bei jedem Interaktionsobjekt. Dies wurde beim Redesign des MIDA-Systems berücksichtigt.

Über den Menüeintrag „Guided Tour“ wird der globale Einstieg, in Form einer graphischen Übersicht, in die vorhandenen geführten Dialoge ermöglicht. Es gibt sechs eigenständige geführte Dialoge zu den verschiedenen Phasen der Entwicklung von graphischen Benutzungsoberflächen. Jeder geführte Dialog gibt Pfade durch das MIDA-System vor und ist eine Art Leseanleitung der Beratungsthemen unter dem Blickwinkel, wie eine ergonomisch gestaltete Benutzungsoberfläche entwickelt werden soll.

In Abb. 7 wird sowohl eine Seite des geführten Dialogs „Präsentation“, als auch das dazu korrespondierende MIDA-Hauptfenster dargestellt. Durch die parallele Nutzung des geführten Dialogs und der MIDA-Beratung können gleichzeitig globale und konkrete Wissensinhalte vermittelt werden. Bei der Evaluation äußerten sich vor allem Entwickler mit wenig Erfahrung im Bereich Entwicklung graphischer Benutzungsoberflächen sehr positiv zu dem Konzept der geführten Dialoge. Allerdings wurde eine Reihe von orientierungsunterstützenden Leistungen eingefordert (z. B. graphische Übersicht auf jeder Seite des geführten Dialogs mit visuellem Hinweis auf aktuelle Phase; Undo-Funktion; schrittweises Vor- und Zurückgehen; jederzeitiger Rücksprung zum globalen Einstieg). Diese Hinweise wurden aufgegriffen und im Redesign berücksichtigt.

Einen wesentlich stärker anwendungsorientierten Zugang zu den Inhalten der Beratung ermöglichen interaktive Beispiele, die über einen entsprechenden Menüeintrag aktivierbar sind. Es handelt sich dabei um ausgewählte Benutzungsoberflächen aus typischen Anwendungsdomänen des Entwicklers. Sie erfüllen zwei Funktionen. Zum einen vermitteln sie anschaulich mögliche Lösungsstrategien für typische Anwendungen, zum anderen erlauben sie einen kontextsensitiven Zugang zu

den Beratungsinhalten. Dieser erfolgt mittels Aktivierung eines Beratungsmodus (Taste „F1“), der zu einer Änderung des Mauszeigers (Fragezeichen mit Pfeil) führt. Wird jetzt ein Interaktionsobjekt im interaktiven Beispiel selektiert, so erfolgt eine kontextsensitive Aktivierung des dazugehörigen Beratungsthemas. Die Evaluation unterstrich die Bedeutung interaktiver Beispiele für die Vermittlung von Beratungsinhalten. Vor allem erfahrene Entwickler bevorzugten diesen Zugang zu den einzelnen Themen, da dies ihrer Meinung nach einen wesentlich zielgerichteteren Zugang ermöglicht.

Unterhalb der Menüleiste des MIDA-Hauptfensters werden immer die grundlegenden Beratungsinhalte (Definitionen, kurze Erläuterungen) mit Hilfe von Texten, Graphiken, Hotlinks zu verwandten Themengebieten, etc. präsentiert (Abb. 4). Der untere Bereich des MIDA-Hauptfensters enthält eine graphische Darstellung der Inhaltsstruktur (Abb. 4). Durch Mausklick auf Themen der Inhaltsstruktur kann man direkt zur entsprechenden Beratungsseite verzweigen. Das graphische Inhaltsverzeichnis erwies sich im Rahmen der Evaluation als eines der wichtigsten Navigationsinstrumente. Mit Hilfe der Buttonleiste unterhalb der graphischen Darstellung des Inhaltes können die wichtigsten Präsentations- und Navigationskomponenten des Beratungssystems alternativ zu den Menüeinträgen aktiviert werden (Abb. 4). Damit wird ein rascher Zugang zu diesen ermöglicht, was auch im Zuge der Evaluation als hilfreich angesehen wurde. Allerdings wurde in der ursprünglichen Version des MIDA-Systems keine klare Trennung zwischen lokalem und globalem Einstieg in die verschiedenen Präsentations- und Navigationskomponenten vorgenommen. Die Benutzer erwarteten aber über die Buttonleiste einen ausschließlichen lokalen Einstieg (z. B. zum entsprechenden Knoten in der Guided Tour; zur entsprechenden Frage im Frage-Antwort-Dialog). Diese Trennung wurde im Rahmen des Redesigns vorgenommen. Die unterhalb der Buttonleiste befindliche Iconleiste ermöglicht eine schnelle Navigation durch Beratungsinhalte sowie einen kontextsensitiven Zugriff auf die Konstruktionsunterstützung (Abb. 4).

Die Evaluationsergebnisse zeigten eine hohe Akzeptanz der gewählten Funktionspräsentation. Aufgrund der Kompatibilität zu anderen Motif-Anwendungen traten kaum Handhabungsprobleme auf. Auch die Anlehnung der Wissenspräsentation an der traditionellen Buch-Metapher ermöglichte den Benutzern eine rasche Orientierung im Beratungssystem. Die Evaluationsergebnisse erbrachten somit keine prinzipiellen Einwände gegen die gewählten Funktions- und Wissenspräsentationskonzepte, aber eine Fülle von Verbesserungsvorschlägen zu Detailspekten.

3.2.2 IDAT – die IDA-Beratung unter Windows. Die Realisierung der IDA-Beratung unter Microsoft Windows erfolgt unter der Bezeichnung IDAT (Interface Design Advice Tool). Als Multimedia-Entwicklungswerkzeug wurde ToolBook (Asymetrix) eingesetzt.

Abbildung 8 zeigt die wesentlichen Komponenten des IDAT-Beratungssystems. Abgesehen von einer zentralen Steuerungseinheit sind dies die eigentlichen Be-

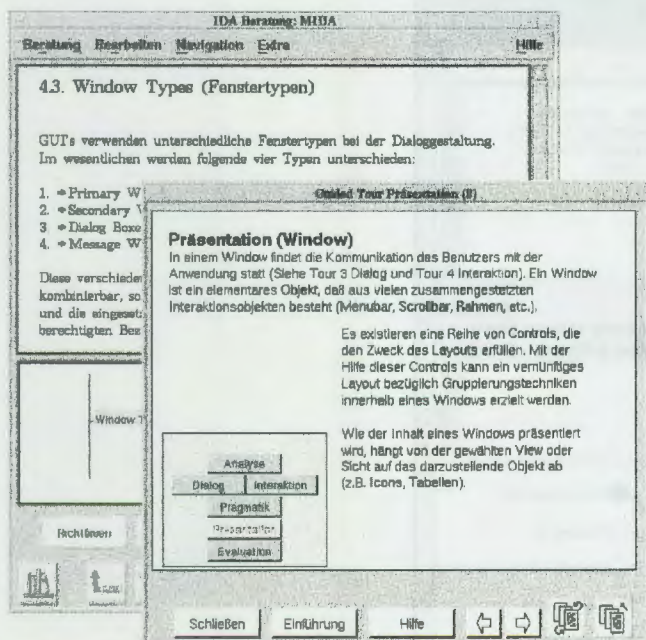


Abb. 7. Geführter Dialog Fenster und korrespondierendes MIDA-Hauptfenster

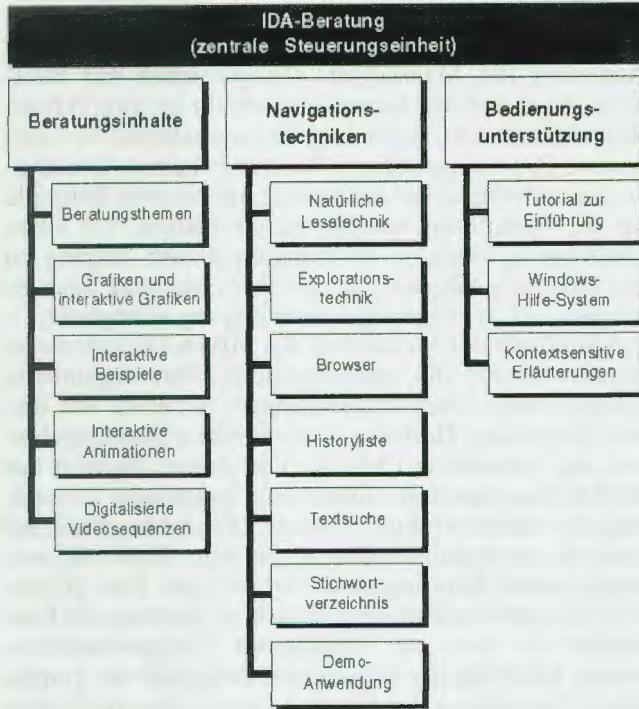


Abb.8. Komponenten des IDAT-Beratungssystems

ratungsinhalte, die Navigationstechniken und die Elemente zur Bedienungsunterstützung. Die Beratungsinhalte setzen sich aus verschiedenen Beratungsthemen mit textueller Erläuterung, Graphiken, Beispielen, Animationen und Videosequenzen zusammen. Die Vielfalt der Navigationstechniken ist als ein Angebot an den Entwickler zu verstehen, sich eine kleinere Anzahl von präferierten Techniken herauszusuchen. Alle Naviga-

tionstechniken sind jedoch in ein zentrales Navigationskonzept integriert, so daß die Anwendung jeder einzelnen Navigationstechnik nur geringe spezielle Kenntnisse voraussetzt. Die dritte Komponente dient zur Bedienungsunterstützung.

Abbildung 9 zeigt, wie sich die IDAT-Beratung dem Entwickler darstellt. Es gibt hier im Unterschied zur vorhin beschriebenen MIDA-Beratung kein zentrales Hauptfenster, von dem die Steuerung des Systems erfolgt. Der Fokus wird hier auf den zu vermittelnden Inhalt, in Form von Themenfenster, gelegt. Der Zugang zu den Interaktionsmöglichkeiten ist weitgehend verborgen. Ein wichtiges Ziel bei der Entwicklung des IDAT-Beratungssystems war Komplexitätsminimierung. Dies wurde durch die Präsentation der Funktionalität gemäß ihrer Bedeutung und Häufigkeit der Anwendung zu erreichen versucht.

- Wichtige Funktionen, wie z. B. Funktionen zum Navigieren im Themenraum, werden permanent in der Toolbar „IDA-Beratung“ an der Oberfläche repräsentiert und sind in einem Interaktionsschritt aktivierbar (Abb.9 oben links).
- Alle Funktionen werden in einem Pop-Up-Menü, das durch Drücken der rechten Maustaste angezeigt werden kann, angeboten (Abb.9); zusätzlich ist eine Aktivierung mittels Tastatur für alle Funktionen vorgesehen.
- Sensitive Bereiche (Hotlinks) im Text der Themenfenster sind optisch durch Unterstreichungen hervorgehoben und verbergen eine Funktionalität, die erst durch Änderung des Mauszeigers beim Bewegen in diesen Bereich angezeigt wird.

Durch die vorhin beschriebenen Maßnahmen der Funktionspräsentation sollte eine Konzentration auf die Beratungsinhalte erreicht werden (Primäraufgabe). Die

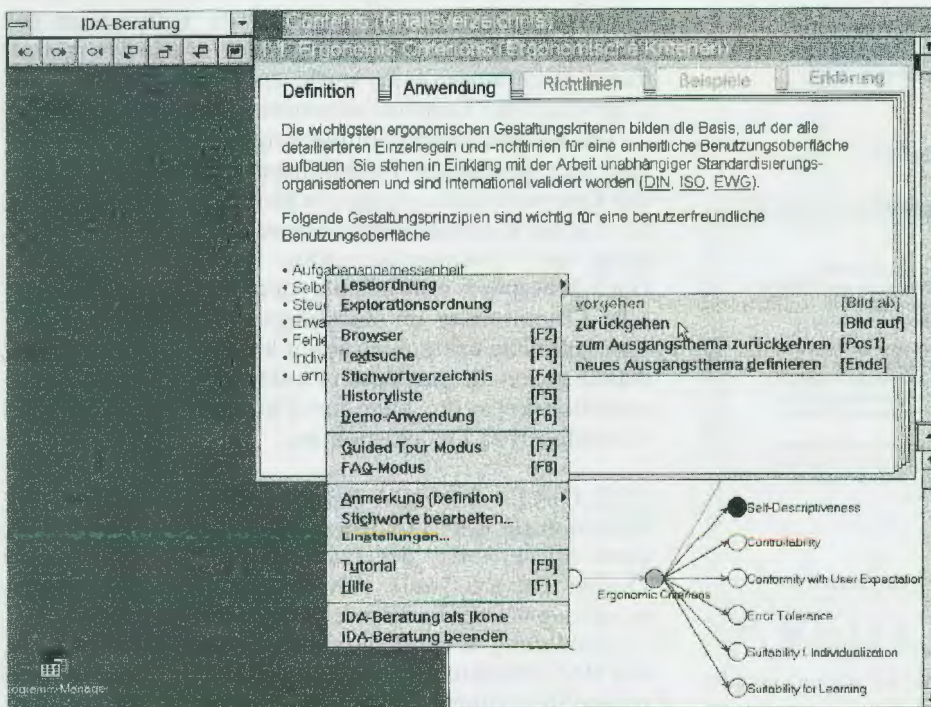


Abb.9. Präsentation der Funktionalität im IDAT-System

Handhabung (Sekundäraufgabe) sollte in den Hintergrund treten. Damit sollte eine Verstärkung des Wissenstransfers erzielt werden. Dies unterscheidet das IDAT-System vom vorhin beschriebenen MIDA-System, das einen traditionellen Ansatz der Funktionspräsentation verfolgte.

Die Evaluation zeigte aber, daß der IDAT-Ansatz der Funktionspräsentation auf erhebliche Akzeptanzprobleme stieß. Hauptproblem für die Benutzer war die nicht gegebene Konformität mit traditionellen Windows-Anwendungen. Es wurde zwar positiv vermerkt, daß eine sehr zielgerichtete Vermittlung der Inhalte erfolgte, aber die Konformität mit anderen Windows-Anwendungen wurde eindeutig höher gewertet. Dies zeigte auf eindrucksvolle Weise, daß MS Windows auf der PC-Plattform einen de facto Standard geschaffen hat, dessen Nichtberücksichtigung zu erheblichen Akzeptanzproblemen führen kann. Auch dann, wenn die verfolgten Funktionspräsentationskonzepte für die jeweilige Anwendung wesentlich aufgabenangemessener erscheinen. Abbildung 10 zeigt das Redesign der IDAT-Beratung, das aufgrund der Evaluationsergebnisse erstellt wurde. Die Funktionspräsentation basiert hier auf der üblichen Multiple Document Interface Fensterarchitektur von MS Windows-Anwendungen.

Ein weiteres Ziel war eine effiziente Orientierungsunterstützung. Um eine sinnvolle und akzeptable Beratungsleistung zu garantieren, muß insbesondere der lost-in-hyperspace-Effekt vermieden werden, der bei Systemen mit hypertextartigen Strukturen häufig auftritt. Das IDAT-Beratungssystem setzt folgende Techniken zur Orientierungsunterstützung ein (Abb. 9):

- einen graphischen Browser,
- die Overlapped-Window-Technik,
- die Karteikarten-Metapher.

Der graphische Browser zeigt die dem aktuellen Themenfenster unmittelbar vorangehenden bzw. nachfolgenden Themen an. Dadurch wird jedes Themenfenster in seinen inhaltlichen Kontext gestellt. Der Browser kann auch zum Navigieren durch das Beratungssystem verwendet werden. Die Evaluationsergebnisse zeigten eine hohe Akzeptanz des Browser. Er wurde als wesentliches Hilfsmittel zur Visualisierung der Explorationsmöglichkeiten angesehen und daher auch in das Redesign übernommen (Abb. 10).

Die Overlapped-Window-Technik bewirkt, daß aktivierte Themenfenster teilweise überlappend dargestellt werden. Durch eine leichte Versetzung der Themenfenster auf dem Bildschirm bleiben die Titelleisten jeder Beratungseinheit stets sichtbar, d. h. der aktuelle Explorationspfad wird visualisiert. Durch Selektion einer Titelleiste kann das entsprechende Themenfenster wieder in den Vordergrund gebracht werden. Die Evaluation zeigt, daß das starre, überlappende Positionieren der Themenfenster nicht akzeptiert wurde. Ein automatisches, überlappendes Positionieren der Themenfenster wurde zwar akzeptiert und gewünscht, aber die Fenster sollten frei beweglich und mit den üblichen Interaktionsmöglichkeiten (z. B. verkleinern, schließen) versehen sein. Dies wurde beim Redesign aufgrund der Realisierung nach dem Multiple-Document-Interface-Konzept berücksichtigt, da hier jedes Themenfenster als eigenes Tochterfenster realisiert wurde (Abb. 10).

Wie Abb. 9 zeigt, wird innerhalb der Themenfenster die Karteikarten-Metapher verwendet. Durch einen Mausklick auf den entsprechenden Reiter einer Karteikarte kann der zugehörige Abschnitt des Themas angezeigt werden. Ein Suchen durch Scrollen in einer durchgängigen Textdarstellung entfällt. Jedes Thema setzt sich aus höchstens fünf Abschnitten zusammen: Defini-

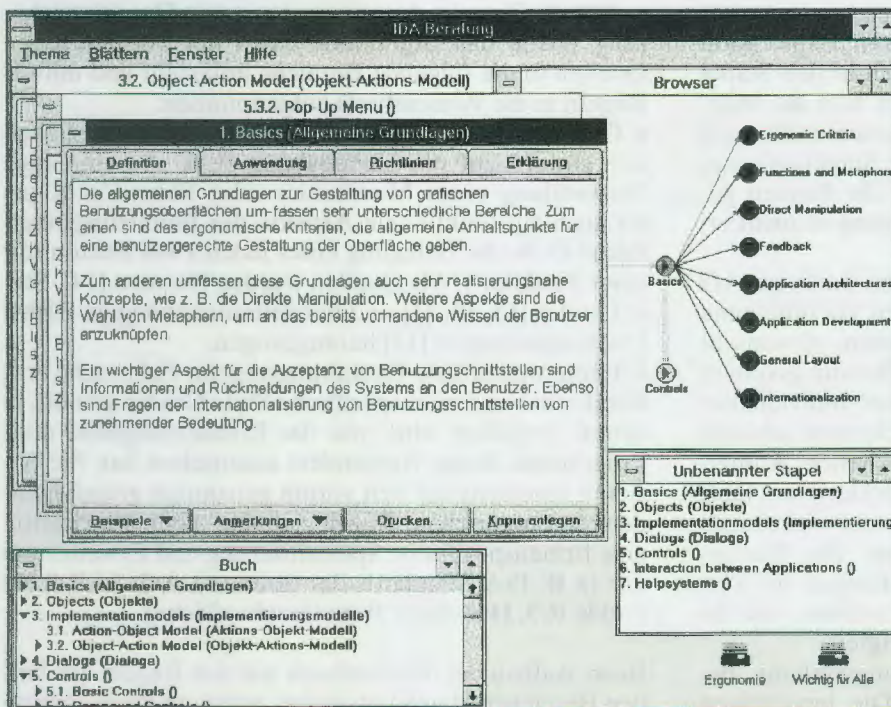


Abb. 10. Redesign des IDAT-Systems aufgrund von Evaluationsergebnissen

tion, Anwendung, Richtlinien, Beispiele und Erklärung. Diese Form der Inhaltspräsentation fand große Zustimmung im Rahmen der Evaluation und wurde daher auch in das Redesign übernommen (Abb.10).

Die Metapher innerhalb des IDAT-Beratungssystems basiert auf einer „Ringbuch-Metapher mit Herauslegetechnik“. Durch Verwendung dieser Metapher sollte dem Benutzer die Orientierung im Rahmen der Navigation erleichtert werden und ihm den Aufbau eines mentalen Modells ermöglichen. Die Metapher geht von der Annahme aus, daß alle Multimedia-Dokumente in Form eines Ringbuches vorliegen. Der Benutzer kann nun auf zwei Arten durch dieses Ringbuch navigieren:

- Blättern in der vordefinierten, linearen Leseordnung der Themen,
- Bewegen in einer benutzerdefinierten dynamischen Explorationsordnung durch die Themen.

Im zweiten Fall werden die gewählten Themenfenster aus dem Ringbuch „entnommen“ und auf einen gesonderten Stapel, mittels der Overlapped-Window-Technik, abgelegt. Dieser Stapel bleibt so lange erhalten, bis der Benutzer einen neuen definiert. Dann werden die bisher herausgelegten Themenfenster wieder automatisch in das Ringbuch einsortiert. Durch das Ablegen der entnommenen Themen in einen benutzerdefinierten Stapel sollte dem Benutzer ein jederzeitiges visuelles Nachvollziehen seiner Exploration möglich gemacht werden. Die Evaluation zeigte aber, daß die gewählte „Ringbuch-Metapher mit Herauslegetechnik“ zu komplex ist. Die Benutzer hatten Verständnis- und Orientierungsprobleme und damit wurde das Gegenteil des angestrebten Zieles erreicht. Als Ergebnis wurde die gewählte Metapher verworfen und eine „Buch-Metapher mit Kopiertechnik“ gewählt und im Redesign realisiert (Abb. 10). Hier bestehen die gleichen Navigationsmöglichkeiten wie vorhin beschrieben, nur werden hier bei der benutzerdefinierten Exploration die Seiten kopiert und dann automatisch in den Stapel einsortiert. Zusätzlich hat der Benutzer hier die Möglichkeit den Stapel unter einem gewünschten Namen abzuspeichern und kann somit mehrere Stapel anlegen. Wird ein Stapel gelöscht, werden nur die Kopien gelöscht und eine automatische Einsortierung ist nicht erforderlich.

Eine empirische Untersuchung bei der Software AG ergab, daß Möglichkeiten Anpassungen vorzunehmen und eigene Anmerkungen zu generieren, gewünscht wurden. Als Konsequenz ist im IDAT-Beratungssystem neben der Möglichkeit zur Definition individueller Stichworte auch eine besonders einfache und schnelle Technik zur Generierung von gesprochenen Anmerkungen vorgesehen worden. Der Entwickler kann **lautsprachliche Anmerkungen in das System einbringen** und wieder abrufen. Dadurch konnte die Hemmschwelle zur Anwendung von Anmerkungen im Vergleich zur textuellen Eingabe gesenkt werden, wie die Evaluationsergebnisse eindrucksvoll zeigten.

Die Evaluation des IDAT-Beratungssystems bewirkte ein grundlegendes Redesign. Die innovativen

Ideen des ursprünglichen Systems fanden nur eingeschränkte Akzeptanz. Sie konnten zwar teilweise in das Redesign übernommen werden, allerdings unter strikter Integration in vertraute Navigations- und Präsentationskonzepte.

3.3 Entwicklungsbegleitende Qualitätssicherung mittels wissensbasierter Techniken

Die Qualitätssicherung wertet bereits erbrachte Entwicklungsergebnisse mittels wissensbasierter Techniken und aufgrund einer expliziten Anforderung durch den Entwickler automatisch aus (software quality assurance).

Was waren die Motive für die Wahl eines wissensbasierten Ansatzes der Qualitätssicherung? Zum einen ist das ergonomische Wissen im wesentlichen vages Wissen. Die Überprüfung solch vagen Wissens mittels algorithmischer Programme ist in der Regel nur mit großem Aufwand zu realisieren. Darüber hinaus bieten wissensbasierte Systeme auch eine vereinfachte Wartbarkeit des Wissensbestandes, da in ihnen eine Trennung von Wissen und Wissensverarbeitung vorliegt. Zur Anpassung einer Regel an neue Standards muß nicht die Inferenzstrategie bekannt sein, sondern nur diejenige Regel bzw. Regelmenge, die für eine bestimmte ergonomische Forderung zuständig ist. Dies erfordert weniger Wissen über den Aufbau der Qualitätssicherung, als dies bei algorithmischen Systemen der Fall ist. Ein weiterer wichtiger Bestandteil eines wissensbasierten Systems ist die Erklärungskomponente, die Auskünfte über das Verhalten des Systems gibt. Auch dies ist in algorithmischen Systemen erheblich schwieriger zu realisieren. Der objekt-orientierte wissensbasierte Ansatz ermöglicht in einfacher Weise auf das Verhalten des Systems Einfluß zu nehmen.

Für die Zwecke der wissensbasierten Qualitätssicherung wurde das Ergonomiewissen der verschiedenen Quellen in die folgende Bereiche unterteilt und mittels Regeln in die Wissensbasis aufgenommen:

- Grundlegende ergonomische Regeln: Sie beziehen sich sowohl auf die Handhabung (z.B. die einmalige Verwendung eines Mnemonics oder Short-Cut-Keys) als auch auf funktionale Aspekte der Benutzungsoberfläche (z.B. die Belegung eines jeden Push Button mit einer Funktion). Als Quellen wurden Normen [14], Style Guides [13, 18, 21] und herstellerunabhängige Richtlinienansammlungen [17] herangezogen.
- Firmenspezifische Gestaltungsregeln: Zahlreiche Anwender haben firmenspezifische Style Guides erstellt, in denen festgelegt wird, wie das Erscheinungsbild aller Programme dieses Anwenders auszusehen hat. Sie basieren einerseits auf den vorhin genannten grundlegenden ergonomischen Regeln, stellen aber gleichzeitig eine firmenspezifische Spezialisierung und Erweiterung dar (z.B. ISA Standards der Software AG, SAP Style Guide R/3, Hoechster Benutzeroberfläche).

Beim Aufbau der Wissensbasis wurden Regeln aus beiden Bereichen abgeleitet, wobei aufgrund der Koopera-

tionsprojekte mit Industriepartnern das Schwergewicht auf firmenspezifischen Gestaltungsregeln lag.

Bei der software-technischen Realisierung der Qualitätssicherung wurde die Architektur eines objekt-orientierten wissensbasierten Systems gewählt. Dieses System wurde mit der Expertensystem-Shell Kappa (Intellicorp) realisiert. Die zwei wesentlichen Komponenten der Architektur sind die Objektbasis und die Regelbasis.

Die Objektbasis ist wiederum unterteilt in drei Klassenhierarchien. In der Oberflächenobjekt-Hierarchie sind Klassenstrukturen vorhanden, in denen mittels Instanzenbildung eine interne Repräsentation der zu überprüfenden Oberflächenentwicklung erstellt wird. Diese Strukturen beinhalten Wissen über den Aufbau einer graphischen Benutzungsoberfläche, sowie Wissen über den Inferenzmechanismus zur Überprüfung der Objekte. Beim Einlesen der zu überprüfenden Oberfläche werden für alle vom Entwickler erstellten Objekte entsprechende Instanzen erzeugt. Das Einlesen der Daten in die Qualitätssicherung erfolgt über eine C-Schnittstelle des Entwicklungswerkzeuges, über die direkt auf das aktuelle Design zugegriffen werden kann. Das aktuelle Design wird im Entwicklungswerkzeug in einer eigenen Beschreibungssprache (Dialog Definition Language) repräsentiert. Die Musterhierarchie ist die zweite Klassenhierarchie der Objektbasis. In ihr werden Muster von Interaktionsobjekten abgelegt, wie sie nach den ergonomischen Regeln aussehen müßten. Bei der Überprüfung von bestimmten Objekten kann dann ein Mustervergleich vorgenommen werden. Die dritte Klassenhierarchie der Objektbasis ist eine interne Hierarchie. In ihr werden Klassen und Instanzen gehalten, die intern zum Abarbeiten der Regeln dienen. So werden zum Beispiel für gefundene Abweichungen Instanzen einer Klasse gebildet, die Informationen zu diesen Abweichungen enthalten. Diese Informationen dienen zur weiteren internen Behandlung der Abweichungen, aber auch zur Erklärung für den Entwickler.

Die Regelbasis ist die zweite wesentliche Architekturkomponente der Qualitätssicherung und besteht unabhängig von der vorhin beschriebenen dreiteiligen Objektbasis. Sie enthält das Wissen über die Bedingungen zur Einhaltung bzw. Nichteinhaltung der ergonomischen Regeln. Dabei kann die Überprüfung einer graphischen Benutzungsoberfläche auf zwei Weisen geschehen. Zum einen kann die Richtigkeit eines Objektes, zum anderen das Vorliegen eines Fehlers bewiesen werden. In beiden Fällen kommen sowohl Vorwärts-, als auch Rückwärtsverkettung in kombinierter Form zum Einsatz.

Im folgenden wird der Ablauf der Qualitätssicherung beschrieben. Nach deren Aktivierung kann der Entwickler zuerst spezifizieren, nach welchem Style Guide die Überprüfung stattfinden soll, d. h. welche Regelbasis zugrunde gelegt werden soll und ob eine Gesamtausgabe bzw. eine schrittweise Ausgabe der erkannten ergonomischen Defizite erfolgen soll. Letztere orientiert sich an der Vorgehensweise, wie man sie von üblichen Rechtschreibprüfprogrammen kennt. Das heißt, nach

jedem erkannten Fehler wird dieser angezeigt und es werden Möglichkeiten zu dessen Behebung angeboten. Diese Form der Präsentation war ein Ergebnis der Evaluation und wurde von den Entwicklern bevorzugt. Nach den Spezifikationen durch den Entwickler werden die Daten der Benutzungsoberfläche abgespeichert und in die Oberflächenobjekt-Hierarchie eingelesen. Anschließend werden automatisch die Inferenzmechanismen aktiviert, die das Abarbeiten der Regeln über die Objekte der graphischen Benutzungsoberfläche steuern. In Abhängigkeit von der gewählten Form der Ausgabe erhält der Entwickler entweder nach der Regelbearbeitung ein Fenster, das Informationen zu allen gefundenen ergonomischen Abweichungen enthält oder sofort nach einem erkannten Regelverstoß ein Fenster, das Informationen zu dieser gefundenen ergonomischen Abweichung beinhaltet. In beiden Fällen werden dem Entwickler eine Reihe von Möglichkeiten zur Fehlerbehebung geboten. Abbildung 11 zeigt beispielsweise ein Fenster, wie es unmittelbar nach einem erkannten Regelverstoß erscheint. Hier werden Fehler im Menüeintrag „Hilfe“ des aktuellen Designs erkannt.

Zur weiteren Bearbeitung hat der Entwickler mehrere Aktionen zur Auswahl. Er kann durch Aktivierung der entsprechenden Icons sowohl in die Beratung als auch in die Konstruktion kontextsensitiv verzweigen, um Hinweise zur Fehlerkorrektur zu erhalten. Bei bestimmten Objekten kann unmittelbar in der Qualitätssicherung eine automatische Änderung vorgenommen werden. Diese Korrektur ist aber nicht bei allen Abweichungen möglich. Sie erfolgt über einen schreibenden Zugriff auf die Oberflächenbeschreibungssprache (über die C-Schnittstelle). Beispielsweise kann der Mnemonic eines Menüeintrages verglichen mit den Anforderungen des gewählten Style Guide als fehlerhaft erkannt worden sein. Die Qualitätssicherung schlägt nun diese Anforderung als Änderung vor.

Abweichungen, die auf vagem Wissen beruhen, entstehen aus einer bestimmten Interpretation von solch vagen Begriffen, wie beispielsweise „nah“ oder „bündig“. Auf die Interpretation dieser Begriffe sollte der Entwickler grundsätzlich Einfluß haben. Daher bietet die Qualitätssicherung in solchen Fällen die Möglichkeit, das Verhalten des Programmes anzupassen. Dies geschieht dadurch, daß bestimmte Regelparameter (z. B. Anzahl der Pixel die die Bündigkeit zweier Objekte bestimmen) zugänglich und änderbar gemacht werden. Diese Möglichkeit der Regelanpassung kann jedoch nur in dem Rahmen angeboten werden, in dem noch eine ergonomische Mindestqualität der Benutzungsoberfläche gewährleistet ist.

Bei der Evaluation der Qualitätssicherung erfolgte unter anderem auch ein Vergleich des wissensbasierten Ansatzes mit einer Expertenevaluation. Für diesen Zweck wurde eine Beispielanwendung von zwei Ergonomieexperten unabhängig voneinander evaluiert und die entdeckten Mängel in einem Prüfbogen erfaßt. Konkurrierend wurde diese Beispielanwendung mit der wissensbasierten Qualitätssicherung evaluiert. Anschließend erfolgt ein Vergleich der entdeckten ergonomischen Defizite. Es zeigte sich dabei die Überlegenheit

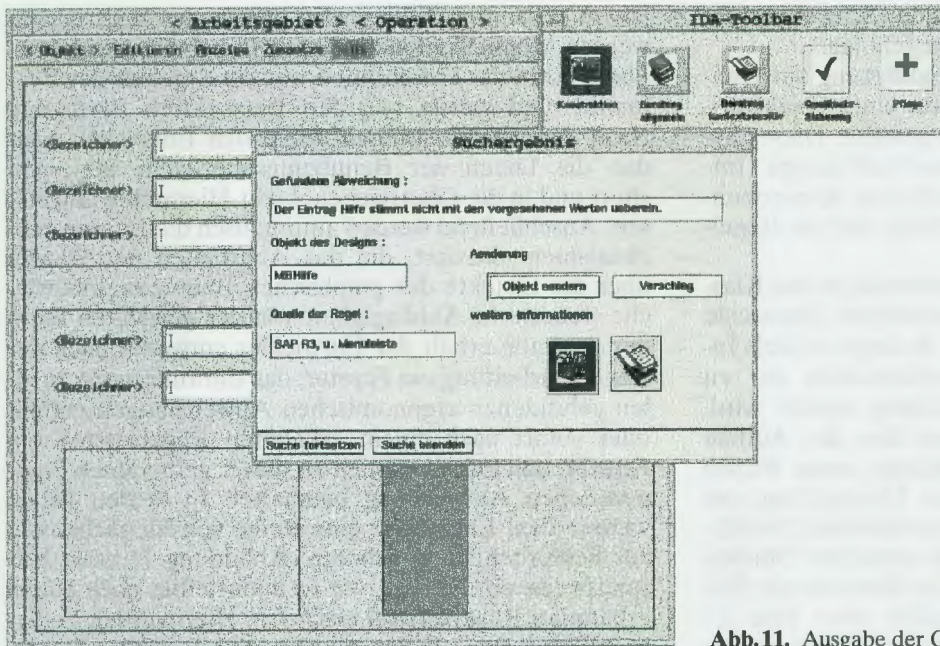


Abb. 11. Ausgabe der Qualitätssicherung

des wissensbasierten Ansatzes beim Entdecken von Defiziten der Kategorien: falsche Platzierung von Menüeinträgen und Interaktionsobjekten, falsche oder fehlende Mnemonics, mangelnde Bündigkeit von Interaktionsobjekten, fehlende Titel, fehlende Separatoren. Gerade das systematische Entdecken derartiger Defizite ist bereits bei einer mittleren Anwendung mühsam und zeitaufwendig für einen menschlichen Experten. Hier erweist sich eine automatisierte Qualitätssicherung als sehr hilfreich. Ihre Grenzen findet eine wissensbasierte Qualitätssicherung aber bei der Bewertung von semantischen Aspekten, wie beispielsweise aufgabenangemessene Platzierung von Interaktionsobjekten, benutzerangemessene Benennungen, sinnvoller Funktionalitätsausschnitt. Hierzu müßte beim Aufbau der Wissensbasis ein enormer Aufwand getrieben bzw. könnten derartige Informationen über einen Frage-Antwort-Dialog im Rahmen der automatischen Qualitätssicherung erfaßt werden. Dies würde aber den Handhabungsaufwand maßgeblich erhöhen. Somit kann eine automatisierte wissensbasierte Qualitätssicherung immer nur eine Ergänzung, aber nicht sinnvollerweise ein Ersatz für eine abschließende ergonomische Beurteilung sein. Hier haben nach wie vor herkömmliche Evaluationsverfahren, wie beispielsweise EVADIS, ihre Bedeutung.

3.4 Pflege der Unterstützungswerkzeuge

Ergonomiewissen unterliegt einem stetigen Wandel aufgrund neuer wissenschaftlicher Erkenntnisse und des technischen Fortschritts. Damit ist die Notwendigkeit offensichtlich, Möglichkeiten zur Pflege des in den Unterstützungswerkzeugen vorhandenen Ergonomiewissens bereitzustellen. Die Pflege vollzieht sich als zweistufiger Prozeß. Zum einen ist das in jedem einzelnen Unterstützungswerkzeug vorhandene Wissen zu pfle-

gen (lokale Pflege), zum anderen ist die Konsistenz zwischen den Wissensbeständen der einzelnen Unterstützungswerkzeuge zu gewährleisten (globale Pflege).

- **Lokale Pflege:** Die Art der Repräsentation des Ergonomiewissens in den einzelnen Unterstützungswerkzeugen ist durchaus unterschiedlich. Daraus ergibt sich, daß die Durchführung von Pflegeaktivitäten in den einzelnen Unterstützungswerkzeugen recht verschieden sein kann. Zur Unterstützung der lokalen Pflege verfügt jedes Unterstützungswerkzeug über einen Menüeintrag „Pflege“. Bei seiner Aktivierung erscheint ein Fenster mit Informationen, die für die Pflege von Bedeutung sind. Bei der Qualitätssicherung erfolgt ein angeleiteter Einstieg in den Programmiereditor der Expertensystem-Shell. In der Beratung gibt eine Tabelle eine Übersicht über alle Beratungsthemen einschließlich der Information, welche Beratungsformen (Anwendung, Animation, Richtlinien usw.) zu einem Thema vorhanden sind und welche Programme hierfür eingesetzt werden. Die Tabelle ermöglicht einen gezielten Einstieg in das jeweilige Programm für Zwecke der Pflege. Ebenfalls in Tabellenform werden die in der Konstruktionsunterstützung vorhandenen Dialogbausteine und ihre wichtigsten Eigenschaften angezeigt, wobei ebenfalls ein direkter Zugang in die Bibliothek vorgesehen ist.

- **Globale Pflege:** Nach jeder lokalen Änderung ist zu prüfen, ob sie Auswirkungen auf die Wissensbestände in den anderen Unterstützungswerkzeugen hat. Dazu wird ein weiteres Unterstützungswerkzeug angeboten, das über die „IDA-Toolbar“ mittels des Pflege-Icon aktiviert werden kann (Abb. 2). Die globale Pflege basiert auf einer Datenbank, welche die Beziehungen zwischen den Elementen der Wissensbestände der einzelnen Unterstützungswerkzeuge enthält. Sie gibt beispielsweise Auskunft darüber, welche mit einem Beratungsthema korrespondierenden Regeln in der Wissensbasis der Qualitätssicherung vorhanden sind. Das

geplante Realisierungskonzept basiert darauf, daß nach jeder lokalen Pflegeaktivität das jeweilige Unterstützungswerkzeug eine Nachricht mit Informationen über die Änderung an die globale Pflege sendet. Es entsteht so eine Liste von Änderungsnachrichten deren Bearbeitung einem Pflege-Administrator vorbehalten bleibt.

5. Resümee und Ausblick

Die bisherigen Projekt- und Evaluationsergebnisse zeigen, daß die in der IDA-Entwicklungsumgebung gewählten verschiedenen Formen der Designunterstützung einen vielversprechenden Ansatz zur Vermittlung und Umsetzung von software-ergonomischen Gestaltungsforderungen darstellen. Entscheidend für die Akzeptanz der IDA-Entwicklungsumgebung ist einerseits das Gelingen der Integration in das Methoden- und Werkzeugumfeld des Entwicklers und andererseits das unmittelbare Auffinden und Anwenden von Wissensinhalten. Der erste Sachverhalt berührt beispielsweise Fragen der Integration in objekt-orientierte Entwicklungsmethoden oder die Anbindung an vorhandene CASE Tools. Beim Entwurf von graphischen Benutzungsoberflächen wird zunehmend eine objekt-orientierte Sichtweise verfolgt. Es ist nun zu untersuchen, inwieweit der durchgängige Einsatz von objekt-orientierten Methoden und Werkzeugen (z.B. integriertes Modellieren von Anwendungs- und Oberflächenobjekten) möglich ist und inwieweit sich hierbei Vorteile ergeben. Eine wichtige Forschungsfrage in diesem Zusammenhang ist die „organische“ Überleitung der Anforderungen aus dem Fachkonzept der vorangegangenen Analyse- und Designphasen in die Implementierungsphase der Benutzungsoberfläche. Hier lassen die neuen objekt-orientierten Ansätze der Systementwicklung erfolgversprechende Anknüpfungspunkte erkennen (z.B. direkte Ableitung von Interaktionsobjekten aus Datenobjekten oder direkte Ableitung von Dialogabläufen aus Kontrollflüssen). Im Rahmen des Projektes sollen gängige objekt-orientierte Ansätze [5, 6, 7, 23] auf ihre diesbezügliche Eignung hin untersucht und gegebenenfalls um erforderliche Notationen erweitert werden. Die so gewonnenen Erkenntnisse könnten wiederum mit Hilfe der Unterstützungswerkzeuge – zusätzlich zum software-ergonomischen Wissen – dem Entwickler vermittelt werden. Damit wäre die notwendige Integration von Methoden und Techniken aus dem Bereich des Software-Engineering und der Software-Ergonomie erreicht.

Der zweite Sachverhalt berührt Fragen von geeigneten Retrieval Strategien und Mechanismen (z.B. Browser, Query-Functions, Guided Tours) sowie von geeigneten Präsentationsformen von Wissensinhalten (z.B. objekt-orientiert, multimedial, wissensbasiert). Durch die Anwendung einer Vielzahl von Retrieval Strategien und Wissenspräsentationsformen im beschriebenen Vorhaben wurde vorerst versucht, einen möglichst maximalen Rahmen aufzuspannen. Im nächsten Schritt wurden die verschiedenen Konzepte hinsichtlich ihrer Eignung und Akzeptanz mit Entwicklern evaluiert.

Dies führte in vielen Fällen zu einem Redesign der ursprünglichen Lösungsansätze und bildet gleichzeitig die Basis für die Weiterentwicklung der Projektergebnisse.

Literatur

1. Baar, D., Foley, J., Mullet, K.: *Coupling application design and user interface design*. CHI'92 Preceedings, pp.259–266. Reading: Addison-Wesley 1992
2. Bachmann, F., Porschen, M., Ramm, M., Reiterer, H., Schäfer, S., Simm, H.: *Unterstützungswerkzeuge für Entwickler von graphischen Benutzungsoberfläche*. In: Böcker, H.-D. (Hrsg.) *Software-Ergonomie '95, Mensch-Computer-Interaktion, Anwendungsbereiche lernen voneinander*, S.61–74. Stuttgart: Teubner 1995
3. Balzert, H.: *Der JANUS-Dialogexperte: Vom Fachkonzept zur Dialogstruktur*. *Softwaretechnik Trends* 13, 62–72 (1993)
4. Beimel, J., Hüttner, J., Wandke, H.: *Kenntnisse von Programmierern auf dem Gebiet der Software-Ergonomie: Stand und Möglichkeiten zur Verbesserung*. In: Gebert, A., Winterfeld, U. (Hrsg.) *Arbeits-, Betriebs- und Organisationspsychologie vor Ort*, S.72–82. Bonn: Deutscher Psychologen Verlag 1993
5. Booch, E.: *Object-oriented design*. Redwood City: Benjamin Cummings 1991
6. Coad, P., Yourdan, E.: *Object oriented analysis*. Englewood Cliffs: Prentice-Hall 1991
7. Coad, P., Yourdan, E.: *Object oriented design*. Englewood Cliffs: Prentice-Hall 1991
8. Dilli, I., Hoffmann, H.-J., Koschorek, D.: *Experience with an object-oriented design approach for a programming environment under a blackboard architecture*. In: Lasker, G.E. (ed.) *Advances in human systems and information technology. The International Institute for Advanced Studies in Systems Research and Cybernetics 1992*, pp.268–276
9. Fischer, G., Lemke, A., Mastaglio, T., Morch, A.: *The role of critiquing in cooperative problem solving*. *ACM Trans. Inf. Syst.* 9 (3), 123–151 (1991)
10. Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., Sumner, T.: *Embedding computer-based critics in the contexts of design*. *INTERCHI '93 Proceedings*, pp.157–164. Reading: Addison-Wesley 1993
11. Gorny P., Viereck A., Quin L., Daldrup U.: *Slow and principled prototyping of usage surfaces: a method for user interface engineering*. In: Züllighoven (ed.), *Proceedings RE'93 – Prototyping*, pp.125–133. Stuttgart: Teubner 1993
12. Hüttner, J., Wandke, H., Rätz, A.: *Benutzerfreundliche Software- psychologisches Wissen für die ergonomische Schnittstellengestaltung*. Berlin: Berd-Michael-Paschke Verlag (in Vorbereitung)
13. IBM: *Object-oriented interface design, IBM common user access guidelines*. Carmel: Que 1992
14. ISO 9241: *Ergonomic requirements for office work with visual display terminals*
15. Janssen, C., Weisbecker, A., Ziegler, J.: *Generation user interfaces form data models and dialogue net specifications*. *INTERCHI '93 Proceedings*, pp.418–423. Reading: Addison-Wesley 1993
16. Löwgren, J., Nordquist, T.: *Knowledge-based evaluation as design support for graphical user interfaces*. *CHI'92 Proceedings*, pp.181–188. Reading: Addison-Wesley 1992
17. Mayhew, D.: *Principles and guidelines in software user interface design*. Englewood Cliffs: Prentice Hall 1992
18. Microsoft: *The Windows Interface, an application design guide*. Microsoft Press, 1992
19. Molich, R., Nielsen, J.: *Improving a human-computer dialogue*. *Commun. ACM* 33 (3), 338–348 (1990)
20. Olsen D., Foley J., Hudson S., Miller J., Myers B.: *Research directions for user interface software tools*. *Behaviour Inf. Technol.* 12 (No. 2), 80–97 (1993)

21. Open Software Foundation: OSF/MOTIF style guide, revision 1.2. London: Prentice-Hall 1993
22. Oppermann, R., Murchner, B., Reiterer, H., Koch, M.: Software-ergonomische Evaluation, der Leitfaden EVADIS II. Berlin: Walter de Gruyter 1992
23. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorenzen, W.: Object-oriented modeling and design. Englewood Cliffs: Prentice-Hall 1991
24. Reiterer, H., Oppermann, R.: Evaluation of user interfaces, EVADIS II – a comprehensive evaluation approach. Behaviour Inf. Technol. 12 (No. 3), 137–148 (1993)
25. Reiterer H.: 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. München: Oldenbourg 1994
26. Reiterer, H.: A user interface design assistant approach. In: Brunstein, K., Raubold, E. (eds.) Applications and Impacts, Information Processing '94. Proceedings of the IFIP 13th World Computer Congress, Hamburg, Germany, 1994. IFIP Transactions A-52, Volume II, Amsterdam: North-Holland 1994, pp.180–187
27. Smith, L.S., Moiser J.: Guidelines for designing user interface software. Bedford: MITRE Corporation 1986
28. Sun Microsystems: Open Look Style Guide. Sun Microsystems 1989
29. Vanderdonck J.: Hypermedia on human-computer interaction principles and guidelines: a survey. ACM SIGLINK Newsletter, V2N4, December 1993



Harald Reiterer (1961). Von 1981 bis 1986 Studium der Betriebsinformatik an der Universität Wien. Von 1986 bis 1991 Doktoratstudium in der Studienrichtung Betriebsinformatik an der Universität Wien. Seit 1987 Anstellung als Universitätsassistent am Institut für Angewandte Informatik und Informationssysteme der Universität Wien. Von Oktober 1990 bis Februar 1995 Gastforscheraufenthalt in der Forschungsgruppe Mensch-Maschine-Kommunikation des Instituts für Angewandte Informationstechnik der Gesellschaft für Mathematik und Datenverarbeitung (GMD), Bonn. Im Mai 1995 Habilitation an der Universität Wien im Fachgebiet Mensch-Computer-Interaktion und anschließend Ernennung zum Universitätsdozenten. Seit 1988 Lehrtätigkeiten an den Universitäten Wien, Koblenz und Dortmund.

Forschungsschwerpunkte: Entwicklung von Vorgehensmodellen zur menschengerechten Gestaltung von Informationssystemen; Entwicklung von Methoden und Werkzeuge zur software-ergonomischen Gestaltung und Evaluation von Benutzungsoberflächen; Entwicklung von Konzepten zum aufgaben- und benutzerangemessenen Einsatz von Multimediatechniken in Benutzungsoberflächen.