

Unterstützung bei der Gestaltung von Benutzungsschnittstellen durch die Bereitstellung von Software-Ergonomie-Wissen in einem Informations- und Beratungssystem

H. Wandke & J. Hüttner
Humboldt-Universität zu Berlin

Zusammenfassung

Die Entwicklung einer ergonomischen Benutzungsschnittstelle ist eine vielschichtige und komplexe Tätigkeit, die durch wissensorientierte Unterstützungsmittel erleichtert werden kann. Der Unterstützungsbedarf von Programmierern bei der Gestaltung benutzungsfreundlicher Software ist enorm. Hilfsmittel sind kaum vorhanden und nicht für die Praxis entwickelt. Mit dem Projekt *inra* (Teil vom Verbundprojekt WEDA, gefördert vom BMFT [Förderkennzeichen 01 HK 790-8] im AuT-Programm) wird ein Unterstützungsmittel für Software-Entwickler erstellt. Ein Hypertextsystem (für Windows) und ein Begleitbuch bieten vielfältige Informationen aus dem Bereich der Software-Ergonomie. Ein Schwerpunkt im Hypertextsystem ist die Bereitstellung von Wissen in Beispielen, mit diesen kann in einem speziellen Modus gearbeitet werden. Damit sollen Entwickler zu einem Perspektivenwechsel animiert werden, sie sollen in die Rolle eines Benutzers mit festgelegten Aufgaben schlüpfen. Das Unterstützungsmittel ist in der Software-Entwicklungspraxis getestet und anschließend verbessert worden. Dieser Prozeß ist noch nicht abgeschlossen. Das System kann sowohl in der Ausbildung (z.B. Universitäten, Weiterbildung) als auch zum selbstgesteuerten Lernen am Arbeitsplatz genutzt werden.

Einleitung

Software-Entwickler stehen vor neuen Aufgaben. Sie sollen - im Gegensatz zu den Ansprüchen noch vor wenigen Jahren - nicht nur Programme entwickeln, die korrekt und zuverlässig laufen, sondern die von späteren Benutzern einfach zu erlernen und leicht zu benutzen sind. Usability ist ein bestimmender Marktfaktor geworden, dessen Wert mit einer Übernahme der EU-Bildschirmrichtlinie (für die BRD wahrscheinlich 1995) noch steigen wird (vgl. Cakir, 1991).

Um diese neuen Anforderungen erfüllen zu können, müssen Software-Entwickler sehr viel über Benutzer und ihre Aufgaben, über Arbeitsorganisation und über Arbeitsgestaltung wissen. Eine Arbeitsgruppe der Gesellschaft für Informatik unter der Leitung von Susanne Maaß hat 1993 für ein Software-Ergonomie-Curriculum insgesamt 29 verschiedene Lernziele in neun Qualifikationsgebieten der Software-Ergonomie zusammengetragen, die im Rahmen der Informatikausbildung erreicht werden sollten (Maaß u.a. 1993). Diese Wissensinhalte sind vielen Entwicklern in der Praxis weitgehend unbekannt (Hüttner & Wandke, 1993).

Ergonomie-Wissen für die Benutzungsschnittstellengestaltung

Die erforderlichen Kenntnisse auf psychologischem und ergonomischem Gebiet werden auch zunehmend bedeutsamer, weil in der jüngsten Zeit die Gestaltung der Benutzungsschnittstelle immer weiter in den Mittelpunkt der Entwicklung von Softwaresystemen rückt. Während Smith und Mosier (1984) noch feststellten, daß ca. 30 - 35 % aller Software-Entwicklungskapazität in die Benutzungsschnittstelle gesteckt wird, ergab eine Umfrage bei Software-Herstellern in den USA von Myers und Rosson (1992), daß durchschnittlich 48 % des Programmcodes der Benutzungsschnittstelle gewidmet ist und daß der Arbeitszeitaufwand für die Gestaltung der Benutzungsschnittstelle sogar deutlich über 50 % der Gesamtentwicklungszeit liegt.

Diese und ähnliche Umfrageergebnisse basieren auf einem Schnittstellen-verständnis, wie es in der Praxis der Software-Entwicklung üblich ist. Legt man das IFIP-Benutzungsschnittstellenmodell (Dzida, 1984) zugrunde, so verstehen Praktiker häufig unter Benutzungsschnittstelle (der tatsächliche Sprachgebrauch im „Alltag“ ist *Benutzerschnittstelle*) nur die Ein-/Ausgabeschnittstelle und die Dialogschnittstelle, jedoch selten die Werkzeugschnittstelle und fast nie die Organisationschnittstelle. Berücksichtigt man, daß aus software-ergonomischer Sicht alle Hard- und Softwarekomponenten zur Benutzungsschnittstelle gehören, mit denen Benutzer nach der "klassischen" Definition von Moran (1981) in Kontakt kommen, so ist der Arbeitsaufwand, der in die Entwicklung der Benutzungsschnittstelle investiert wird, wahrscheinlich noch größer. Unterstützt wird diese eingeschränkte Problemsicht durch die Annahme, daß schon allein das Befolgen von Style Guide Regeln eine ergonomische Benutzungsschnittstelle entstehen läßt.

Es ist ein Ziel von *inra*, diese Problemsicht zu erweitern und auf Gestaltungsprobleme aufmerksam zu machen, die über das "look and feel" einer Benutzungsschnittstelle hinausgehen. Dies ist schwierig, da es in der Software-Ergonomie auf den unteren Ebenen einer Benutzungsschnittstelle (z.B. Informationsdarbietung, visuelle Gestaltung) sehr viel mehr Gestaltungswissen gibt, das außerdem sehr viel stärker formalisierbar ist, als auf den höheren Ebenen (Aufgabenebene, semantische Ebene). *inra* bietet deshalb auf drei verschiedenen Wegen Gestaltungswissen für eben diese höheren Ebenen der Benutzungsschnittstelle an:

1. Die Gestaltung der Funktionalität wird in die Beispiele des *inra* aufgenommen. Da die meisten Gestaltungsprobleme auf höheren Ebenen nicht durch bloßes Betrachten einer Oberfläche erkannt werden können, sind im *inra*-System Aufgaben integriert, die im Rahmen der Beispiele bearbeitet werden sollen.

2. Durch zahlreiche Hinweise in der Textkomponente des *inra* vom Typ "Dieses Problem können Sie nur lösen, wenn Sie die Aufgaben der Benutzer kennen, ...wenn Sie Benutzer fragen, ...wenn Sie Benutzertests durchführen, ...".
3. Durch die Aufnahme einer Zusammenstellung von empirischen Evaluationsmethoden der Software-Ergonomie in das *inra*-Begleitbuch. Hier wird erläutert, wie Benutzerbefragungen, -beobachtungen u.ä. durchgeführt werden, welche Vor- und Nachteile diese Methoden besitzen und welche Voraussetzungen erfüllt sein müssen, um sie anwenden zu können.

Unterstützung der Entwickler bei der Gestaltung von Benutzungsschnittstellen

Die Gestaltung von Benutzungsschnittstellen ist ein Prozeß der drei unterschiedliche Vorgehensweisen beinhalten kann, die mit verschiedenen Mitteln unterstützt werden können. Zum einen kann er als Technologie, d.h. als eine hochgradig formalisierte und durch Regeln bestimmte Vorgehensweise aufgefaßt werden, insbesondere dann, wenn Bauelemente, Verfahren oder Formeln benutzt werden können. Das ist beispielsweise beim Zusammenstellen einer grafischen Benutzungsoberfläche aus einer Menge von widgets der Fall. Hier können Baukastensysteme mit integrierten „Ergonomiekontrollen“ den Entwickler entlasten, indem sie beispielsweise Teile seiner Aufgaben automatisch realisieren und als Lösung anbieten. Da diese technologisch orientierte Gestaltung durch andere Unterstützungsmittel (vgl. z.B. Projekt JANUS; Balzert, 1993 oder Projekt IDA; Reiterer, 1992) bereits gut abgedeckt ist, wird im *inra* darauf nicht eingegangen.

Zugleich kann Gestaltung aber auch erfahrungsgeleitet erfolgen. Entwickler, die bereits viele Benutzungsschnittstellen gestaltet haben, zu diesen Rückmeldungen von Benutzern erhalten haben, können auf bestimmte Musterlösungen zurückgreifen und diese neuen Anforderungen anpassen. *inra* vermittelt zu großen Teilen solches Erfahrungswissen, das nicht formalisierbar ist und das - im Gegensatz zum technologischen Vorgehen - nicht direkt angewendet werden kann, sondern vom Entwickler aufgenommen und "umgesetzt", d.h. auf seine konkreten Gestaltungsprobleme übertragen werden muß. Dies wird unterstützt durch die Beispiele und Kommentare, die eine Verallgemeinerung des Erfahrungswissens ermöglichen sollen.

Neben technologischen und erfahrungsgeleiteten Aspekten schließt Gestaltung auch kreative Aspekte ein. Kreatives Gestalten kann nur sehr begrenzt unterstützt werden. *inra* bietet zwei notwendige, aber keineswegs hinreichende Rahmenbedingungen für kreatives Gestalten.

1. Es stellt (in Textform) psychologisches Hintergrundwissen bereit, das helfen kann, neue Gestaltungslösungen zu finden.

2. Es weist mit den Kommentaren für die Beispiele und mit den Texten auf die Vorläufigkeit und Offenheit der bisherigen Gestaltungslösungen - u.a. abhängig von der zur Verfügung stehenden Hard- und Software - hin. Die Gestaltungsbeispiele des *inra* sind keine perfekten, kopierbaren Gestaltungslösungen, sondern sollen anregen, wie man es anders und (noch) besser machen kann.

Informations- und Beratungssystem - *inra*

In einer Untersuchung wurden Entwickler mit einem Fragebogen nach ihren derzeit verfügbaren Hilfsmitteln und nach ihren Vorstellungen von einem praxistauglichen Unterstützungsmittel befragt (Beimel, Hüttner & Wandke, 1993). Sie entschieden sich in fast allen Aspekten der Benutzungsschnittstellengestaltung für eine Unterstützung durch ein klassisches Buch mit leicht verständlichen Texten *und* ein rechnerbasiertes Unterstützungsmittel. Die eher klassischen Unterstützungsmittel, wie Schulungen, Lehrbücher und externe Experten wurden weniger gefordert. Zusätzlich zu den technischen Vorteilen, die ein Computersystem als Unterstützungsmittel bietet, wird es auch aus motivationalen Gründen nahezu unmöglich sein, Personen, die ständig am Rechner arbeiten, zum Wissenserwerb ohne einen Computer zu motivieren.

Ein weiterer Gesichtspunkt ist die Freude von Software-Entwicklern am Explorieren. Das Erkunden und Ausprobieren einer Software ermöglicht einen individuellen Zugang zum jeweiligen Wissenskörper - eine neue Anwendungs-Software oder ein wissensorientiertes Unterstützungsmittel (vgl. den weitreichenden Vorschlag von Paul [1993, 1994] interaktive Systeme explorationsfreundlich zu gestalten).

Mit dem Informations- und Beratungssystem *inra* soll Software-Entwicklern ein modernes Unterstützungssystem zur Verfügung gestellt werden, das zwar nicht alle, aber doch wesentliche Anforderungen und Erwartungen der befragten Entwickler erfüllen kann. *inra* ist ein Unterstützungsmittel, das zusammen mit anderen (vgl. z.B. die Beiträge von Bachmann u.a., Heintzen u.a., Malinowski und Stolze, in diesem Band) für verschiedene Teilaufgaben und Phasen der Benutzungsschnittstellengestaltung herangezogen werden kann. Dieses System soll den Programmierern helfen, Kompetenz für einen Teil ihrer Arbeitsaufgabe zu erwerben. Spezifisches Fachwissen aus dem Forschungsgebiet Software-Ergonomie muß so aufbereitet werden, daß es für Personen außerhalb der kleinen scientific community verständlich, nachvollziehbar und damit erst nutzbar wird. Das Informations- und Beratungssystem besteht aus einem Hypertextsystem für IBM-kompatible PC's (entwickelt mit ToolBook) und einem Begleitbuch (Hüttner, Wandke & Rätz, 1994). Die folgenden Darstellungen beziehen sich vor allem auf das Hypertextsystem.

Das Gegenstandsgebiet von *inra* - die ergonomische Gestaltung von Benutzungsschnittstellen - verändert sich in hohem Tempo. Heute Gültiges kann schon in wenigen Jahren überholt sein, neue Techniken erfordern wahrscheinlich

andere ergonomische Lösungen. Ständig verbesserte Rahmenbedingungen in der Hard- und Softwaretechnik bieten die Voraussetzungen für die Entwicklung neuer Interaktionstechniken. „Das aktuell Gelernte kann also höchstens exemplarisch sein und sollte zu Innovationen befähigen und ermuntern.“ (Oberquelle, 1994, S. 24). Das Wissen, das durch *inra* vermittelt und bereitgestellt werden soll, ist deshalb weniger technologiebezogen, sondern stärker auf grundsätzliche Eigenschaften menschlicher Informationsaufnahme und -verarbeitung sowie auf die Steuerung von Handlungen ausgerichtet. Es wurde bei der Entwicklung des Systems auch nicht versucht, das Ergonomie-Wissen vollständig abzubilden.

Das Ziel von *inra* ist es also nicht, für jede Gestaltungsfrage eine konkrete Antwort zu bieten, sondern den Software-Entwickler in die Lage zu versetzen, selbst eine Antwort zu finden.

Bevor Entwickler diese Antworten finden können, müssen sie allerdings in der Lage sein, die richtigen Fragen stellen zu können. *inra* soll auch dazu einen Beitrag leisten. Durch die Arbeit mit dem System sollen Entwickler von Software auf Schwierigkeiten bei der Benutzungsschnittstellengestaltung aufmerksam gemacht werden. Dies wird vorrangig durch einen Perspektivenwechsel erreicht: die Entwickler sollen sich in die Rolle eines Benutzers mit festgelegten Aufgaben begeben und versuchen, mit einer Beispiel-Software diese Aufgaben zu lösen. Sie sollen anhand von komplexen Beispielen für Benutzungsschnittstellen erkennen können, an welchen Stellen und durch welche Gestaltungsdetails Benutzer Schwierigkeiten beim Umgang mit Systemen haben können. Es soll eine Sensibilisierung der Entwickler für die Belange der Benutzer erreicht werden. "The first step toward improving computer-human interfaces *involves changing the attitude of the software professional*. Software professionals need to be convinced that a high-quality user interface is in fact important." (Mayew, 1992, S.3, kursiv im Original). Zugleich sollen die Beispiele aber auch Anregungen bieten, wie man eine Benutzungsschnittstelle verbessern kann.

Inra dient sowohl der Wissensvermittlung (längerfristige Nutzung) als auch der Bereitstellung von Wissen (kurzfristige Nutzung). Primär ist *inra* dafür gedacht, daß Entwickler durch das Bearbeiten von Beispielen in einer Art "Selbsterfahrung", durch das explorative, von Kommentaren unterstützte Durchprüfen von Beispielen und durch das Lesen von Empfehlungen und Hinweisen Wissen erwerben, das sie später bei der Gestaltung von Benutzungsschnittstellen einsetzen können, ohne *inra* wiederholt zu benutzen. Die Empfehlungen und Hinweise werden in einer kurzen und knappen Form am Bildschirm dargeboten und sind ergänzend, in einer ausführlichen, auch stärker auf Begründungen und Hintergrundinformation eingehenden Form in dem Begleitbuch zusammengestellt.

Inra ist aber auch dazu gedacht, als externer Wissensspeicher benutzt zu werden. Software-Entwickler, die eine spezifische Information benötigen, können *inra*

heranziehen, um diese Information in den Texten, Beispielen und Kommentaren zu finden. Dazu gibt es in dem Hypertextsystem verschiedene Such- und Navigationsfunktionen, die es dem Entwickler einfach machen sollen, schnell die entsprechende *inra*-Komponente zu finden (siehe ein Beispiel in Abb. 1). Darüber hinaus bietet ein Glossar Informationshilfe durch die Erläuterung psychologischer und software-ergonomischer Begriffe. Dies ist allerdings eine eher wissensvermittelnde Komponente als eine, die Wissen zur direkten Nutzung bereitstellt. Ein kommentiertes Literaturverzeichnis liefert Hinweise auf weitere Informationsquellen. Diese Komponente ist geeignet, die Unabgegrenztheit und Offenheit von *inra* zu verdeutlichen, verweisen doch die Literaturangaben auf weitere, außerhalb von *inra* liegende externe Wissenspeicher.



Abb.1: Das *inra*-Stichwortverzeichnis ist ein Beispiel für einen schnellen Zugang zu verschiedenen Informationen zu einem Stichwort (hier „Gruppierung“).

Alle sechs *inra*-Komponenten (Textteil, Glossar, Einführung, Literaturverzeichnis, Beispielsammlung und Stichwortverzeichnis) sind nach einem einheitlichen Schema aufgebaut. Ein Überschriftsbereich enthält den Namen sowie ein Symbol für die jeweilige Komponente. Am unteren Rand jeder Komponente gibt eine Leiste mit Funktionsknöpfen einen Überblick über die anderen erreichbaren Komponenten und die gerade aktive Komponente - hier ist der Knopf eingedrückt und grau dargestellt. Alle Komponenten - ausgenommen das Stichwortverzeichnis - können wie Bücher

seitenweise durchgeblättert werden. Auf der ersten Seite wird eine Übersicht über den jeweiligen Komponenten-Inhalt gegeben. Die weiteren Seiten enthalten die Informationen in unterschiedlicher Form. Die Inhalte der einzelnen Komponenten sind in einer Hypertextstruktur über Führungsworte miteinander verbunden.

Komponente	Inhalte	Unterstützungsfunktionen
Einführung	Beschreibung der <i>inra</i> -Komponenten	Einführung bei Erstbenutzung / Erläutert den Umgang mit <i>inra</i>
Stichworte	Suchworte vorr. zu Gestaltungsprobl.	schnelle Navigation / Unterstützt den Zugriff auf das Wissen in den Komponenten
Textbuch	Software-Ergonomie-Wissen in kurzen Texten	Vermittlung von Grundwissen in Form von Empfehlungen u.ä. / Wissenserwerb für die langfristige Nutzung / Querverweise
Beispiele	Beispiele mit Erklärungen zum Ausprobieren	Sensibilisierung, Exploration, Entdeckung von Problemen, Fragen / Wissenserwerb durch selbstgesteuertes Lernen
Glossar	Begriffe aus der Psychologie / Arbeitswissensch.	Erklärung der Bedeutung der Begriffe / Unterstützt den Wissenserwerb / Schaffung einer deklarativen Wissensstruktur
Literatur	Quellenverweise und erklärende Hinweise	Weiterführende Informationen / „Weiterlernen“ / Erweiterung des Suchraums für Gestaltungsfragen

Tab. 1: Die Inhalte der Komponenten und deren Funktionen im *inra*-Hypertextsystem.

Die Entwicklung der Beispiele im Hypertextsystem

Die beiden wesentlichen Vorteile des entwickelten Hypertextsystems sind der flexible, individuelle Zugang zur Information und die Realitätsnähe der Beispiele auf dem Bildschirm. Die Beispiele wurden in einen inhaltlichen Rahmen gefaßt, um damit einzelne Wissensinhalte für eine fiktive, relativ komplexe Software-Anwendung umzusetzen. Nur dadurch ist es möglich, Wechselwirkungen zwischen Einzel-Gestaltungsentscheidungen zu illustrieren. Eine hinreichende Komplexität ist auch notwendig, wenn nicht nur Wissen zur Ein-/Ausgabeschnittstelle, sondern auch für die höheren Ebenen der Benutzungsschnittstelle vermittelt werden soll. Die fiktive Software-Anwendung darf allerdings kein Spezialwissen erfordern, zum Verständnis sollte Alltagswissen ausreichen. Es wurde deshalb eine stark vereinfachte und ausschnittshaft simulierte Tätigkeit eines Angestellten in einer Spedition als Anwendung gewählt. Dadurch, daß alle Einzelbeispiele in diesem Arbeitskontext angesiedelt sind, fügt sich jedes Einzelbeispiel in einen umfassenden Rahmen und die Wissensvermittlung geht über Einzelelemente hinaus. Um eine gute

Kontrastwirkung zu ermöglichen und damit den Lerneffekt zu erhöhen, wurde angestrebt von jedem komplexen Beispiel eine positive und eine negative Lösung zu entwickeln.

Die Entwicklung der Beispiele erfolgt in einem iterativen Prototyping-Prozeß, der hier für das erste Beispielpaar vorgestellt werden soll. In den ersten beiden Beispielen wurden insbesondere die Themen: Formulartechnik und Eingabemasken, Tabellengestaltung, und teilweise auch Aspekte der Dialogsteuerung und der Benutzerunterstützung durch entsprechende Suchfunktionen etc. aufgegriffen. In einem ersten Schritt entwickelte ein Mitarbeiter unserer Forschungsgruppe ein Beispiel mit einer festgelegten Programmfunktionalität zur Durchführung bestimmter Verwaltungsaufgaben in einem Speditionsbetrieb. Diese nach „bestem Wissen und Gewissen“ erarbeitete erste Version lieferte das erste „negative Beispiel“. Ausgehend von der ersten Version wurde in Teamsitzungen schrittweise eine zweite, ergonomisch verbesserte Version entwickelt, die dann als „positives Beispiel“ bewertet wurde. Dabei standen einige fiktive Arbeitsaufgaben, die mit dem Beispiel gelöst werden sollen, im Mittelpunkt der Umgestaltung. Für diese konkreten Aufgaben wurde versucht, die ergonomischen Kriterien umzusetzen.

Damit die Generalisierung des in den Beispielen angewandten Gestaltungswissens besser gelingt, sind Kommentare eingefügt worden. Die Kommentare geben in Kurzfassung noch einmal wieder, aus welchem Grunde eine bestimmte Gestaltungslösung zu empfehlen ist bzw. was - das negative Beispiel betreffend - an einer Gestaltungslösung genau zu kritisieren ist (siehe Abb. 2). Eine ihrer wichtigsten Funktionen ist es, die Entwickler bei der Abstraktion von der Beispielebene in die Ebene der Gestaltungsprinzipien zu unterstützen.

Die Beispiellösungen liegen in der Beispiel-Komponente als Abbildungen einer Oberfläche auf einer Bildschirmseite vor (siehe unterer Teil der Abb. 2) und sind in einem speziellen Modus benutzbar. Innerhalb einer spielerischen Situation kann der Benutzer der Beispielsammlung (also die Entwickler) unterschiedliche Aufgaben eines Dispatchers einer fiktiven Speditionsfirma interaktiv mit der Software bearbeiten. Die festgelegten, einfachen Arbeitsaufgaben sind so ausgearbeitet, daß bestimmte Funktionen des Beispielprogramms benutzt werden müssen bzw. in der „negativen Variante“ vermißt werden.

Einige Beispielaufgaben:

1. Auf Beschluß der Bezirksversammlung wurde in Unterhachingen die „Allee der Kosmonauten“ umbenannt. In dieser Straße wohnt ein Fahrer unseres Unternehmens. Die Straße heißt jetzt "Astronautenallee". Bitte aktualisieren sie die Straßenbezeichnung in der Datei.
2. Unser Fahrer Paul Klotzky hat letzte Woche geheiratet und den Namen seiner Gattin angenommen. Er heißt jetzt "Frühau". Bitte ändern Sie seine Personalangaben.



Abb. 2: Ausschnitt aus dem Beispielpart des Hypertextsystems. Die Kommentare im Feld oben rechts beziehen sich auf einzelne Aspekte die mit den Pfeilen gekennzeichnet sind.

Versetzt in die Lage des Benutzers, müssen die Entwickler mit der vorhandenen Software arbeiten, so bemerken sie gestalterische Mängel der Benutzungsschnittstelle, die sie ansonsten übersehen oder als trivial abtun würden. Viele Mängel werden erst deutlich, wenn die Aufgabenlösung zwar prinzipiell möglich - aber eben nicht angemessen erfolgen kann. In der Bearbeitung von alltäglichen, einfachen Aufgaben soll der Anwendungs-Entwickler im selbstgesteuerten Lernprozeß die Vorteile einer aufgaben- und benutzerorientierten Funktionalität und Oberflächengestaltung bei der konkreten Anwendung erleben.

Karte Suchen Statistik

Fahrer	Fahrzeug
Lindemann	001 20 t
Pächter	002 20 t
Klotzky	004 20 t
Schiffer	005 20 t
Tauscher	006 20 t
Sommerfeld	008 20 t
Reichenbach	010 20 t
Zumbusch	

02.01.1993 08:30

Auftrag	
Nummer	: 475
Strecke	: Nürnberg-Hannover
Ladung	: 20 t
Abfahrt	: 03.01. ab 8.00 Uhr
Ankunft	: 03.01. bis 15.00 Uhr
Fahrer	:
Fahrzeug	:

Hinweise zur Bearbeitung dieses Beispiels
 4. Reagieren Sie auf die Meldung von Fahrer Klotzky (etwa gegen 10.00 Uhr Systemzeit)!
 Vergeben Sie seinen zweiten Auftrag an den Fahrer van der Alm mit dem Fahrzeug 0010.

Abb. 3: Ausschnitt aus der lauffähigen Version des Beispiels „Dispatcher positiv“. Im unteren Teil der Abbildung befindet sich eine Bearbeitungs-Aufgabe.

Resonanz und weitere Arbeitsschritte

Einen breiten Personenkreis konnten wir als Aussteller auf der CeBit'94 in Hannover erreichen. Die Resonanz war sehr positiv. Vor allem die Art der Wissensdarstellung mit einem Hypertextsystem weckte Interesse, das System zu testen bzw. zu erwerben. Das erhebliche Wissensdefizit war unseren Gesprächspartnern bekannt. Der Entwicklungsprozeß von *inra* schloß im Sinne eines iterativen Prototyping Evaluationsschritte ein. Studenten und Software-Entwickler testeten Teile vom *inra*, die Ergebnisse wurden in die aktuelle Version eingearbeitet. Nach dem vorläufigen Abschluß der Entwicklungsarbeiten ist eine stärker formative Evaluation vorgesehen. Das Unterstützungsmittel *inra* soll auf seine Gebrauchstauglichkeit am Arbeitsplatz getestet werden. Dabei soll die Qualität und Alltagstauglichkeit der einzelnen Teile und des *inra*-Systems insgesamt geprüft werden. Begleitbuch und Prototyp sollen Entwicklern einige Wochen an ihrem Arbeitsplatz zur Verfügung stehen. Durch die Variation von Anwendungsbedingungen und durch die Kombination von *inra* mit Methoden der Usability-Forschung erwarten wir Aussagen zur Wirkungsweise, zur Akzeptanz und zur Effizienz. Nach ersten Test-Erfahrungen mit der Software in der IQ-Media Berlin lassen sich zwei Benutzertypen erkennen: der „neugierige Praktiker“

und der „zielgerichtete Analytiker“. Beide verfolgen unterschiedliche Ziele und benutzen den *inra* auf verschiedene, charakteristische Weise (Adrian, 1994).

Literatur

- [1] Adrian, V. (1994). Praxiseinsatz des Informations- und Beratungssystems *inra* in der IQ-Media GmbH. Vortrag auf dem WEDA-Abschlußworkshop am 7.10.94 in Stuttgart. (Unveröff.)
- [2] Asymetrix (1991). ToolBook - Benutzerhandbuch. Bellevue: Asymetrix Corporation.
- [3] Balzert, H. (1993). Der JANUS-Dialogexperte: Vom Fachkonzept zur Dialogstruktur. Vorabdruck zum Vortrag auf der Softwaretechnik-Tagung 1993 in Dortmund. (Unveröff.)
- [4] Beimel, J., Hüttner, J. & Wandke, H. (1993). Kenntnisse von Programmierern auf dem Gebiet der Software-Ergonomie: Stand und Möglichkeiten zur Verbesserung. In A. Gebert & U. Winterfeld (Eds.) Arbeits- Betriebs- und Organisationspsychologie vor Ort. Bonn: Deutscher Psychologen Verlag.
- [5] Cakir, A. (1991). Software-Ergonomie und Arbeitsorganisation - neue Regelungs-gegenstände im Arbeitsschutz in Europa 1992 - Was bringen die Europäischen Regelwerke für Bildschirmarbeitsplätze. Tagungsband, Berlin: Ergonomic Institut.
- [6] Dzida, W. (1984). Das IFIP-Modell für Benutzerschnittstellen. Office-Management, Sonderheft.
- [7] Hüttner, J. & Wandke, H. (1993) What do system designers know about software ergonomics and how to improve their knowledge? In H. Luczak, A. Cakir & G. Cakir (Eds.) Work With Display Units 92 (S. 304-308). Amsterdam u.a.: Elsevier
- [8] Hüttner, J., Wandke, H. & Rätz, A. (1994). Benutzerfreundliche Software - Psychologisches Wissen für die ergonomische Schnittstellengestaltung. (in. Vorb.) Berlin: Bernd-Michael Paschke Verlag.
- [9] Maaß,S., Ackermann,D., Dzida,W., Gorny,P., Oberquelle,H., Rödiger,K.-H., Rupietta,W. und Streitz, N. (1993): Software-Ergonomie-Ausbildung in Informatik-Studiengängen bundesdeutscher Universitäten. Empfehlungen des Fachausschusses 2.3 und des Fachbereichs 2 der GI. Informatik-Spektrum 16, 25-30.
- [10] Mayew,D.J. (1992). Principles and guidelines in software user interface design. Englewood Cliffs : Prentice Hall.
- [11] Moran, T.P. (1981): The command language grammar: a representation of the user interface of interactive computer systems. Int. J. Man-Machine Studies 15, S. 3 -50.
- [12] Myers, B. & Rosson, M. (1992). Survey of User Interface Programming. In CHI-Proceedings, S. 195-202.
- [13] Oberquelle, H. (1994) Software-Ergonomie lehren - Software-Ergonomie lernen. Ergonomie und Informatik 22, 24-26.
- [14] Paul, H. (1993). Das Explorative Modell als konzeptioneller Ansatz zur Gestaltung interaktiver Systeme. In K.-H. Rödiger (Hrsg.) Software-Ergonomie 93 - Von der Benutzungsoberfläche zur Arbeitsgestaltung. (S. 77- 86) Stuttgart: Teubner.
- [15] Paul, H. (1994). Exploratives Agieren - Ein Beitrag zur ergonomischen Gestaltung interaktiver Systeme. Ergonomie und Informatik, 22, 49-52.

- [16] Reiterer, H., Oppermann, R. und Bleimann, U. (1992) Integration von software-ergonomischem Gestaltungswissen in ein User Interface Management System - User Interface Design Assistance (IDA). internes Projektpapier (unveröff.).
- [17] Smith, L.S. & Mosier, J.N. (1986). Guidelines for designing user interface software. Bedford: The Mitre Corporation.

Hartmut Wandke & Jens Hüttner
Humboldt-Universität zu Berlin
Institut für Psychologie
Oranienburger Straße 18
10178 Berlin
e-mail: hwandke@psychologie.hu-berlin.de