

# DIE GESTALTUNG DER MENSCH-COMPUTER INTERAKTION

Harald Reiterer

Universität Konstanz

Fachbereich Informatik und Informationswissenschaft

## Einleitung

Der Autor ergreift mit diesem Beitrag zur Festschrift, anlässlich des fünfzigsten Geburtstages von Prof. Dr. A Min Tjoa, die Gelegenheit, eigene Beiträge zur Forschungsdisziplin Mensch-Computer Interaktion (MCI) Revue passieren zu lassen, die in vielfältiger Weise durch Prof. Tjoa maßgeblich beeinflusst worden sind, wie in den weiteren Ausführungen noch deutlich werden wird. Die eigenen Forschungsarbeiten der letzten 15 Jahre werden dabei ausgewählten Entwicklungen der Forschungsdisziplin MCI zugeordnet und in chronologischer Reihenfolge beschrieben. Damit ergibt sich gleichzeitig ein - zugegeben sehr lückenhafter - historischer Abriss von wesentlichen Entwicklungen im Bereich der Gestaltung der MCI in den letzten 15 Jahren.

Im ersten Abschnitt dieses Beitrages wird auf die große Bedeutung der Entwicklung der *graphischen Benutzeroberflächen* für das umfassende Vordringen des Computers in alle Lebensbereiche eingegangen. Als direkte Folge daraus ergab sich, dass sich die Gestaltungsaufgaben des Informatikers im Bereich der Softwareentwicklung maßgeblich veränderten. Einige dieser Veränderungen werden in den folgenden Abschnitten vorgestellt. Zuerst wird gezeigt, dass aus der *Softwaregestaltung* eine wesentlich umfassendere *Arbeitsgestaltung* geworden ist. Fragen, welchen Beitrag die MCI zur *Qualitätssicherung von Softwareprodukten* leisten konnte, bestimmte vor allem die Forschung in den 80er und frühen 90er Jahren. Diese produktorientierte Betrachtungsweise wurde dann in den 90er Jahren durch eine verstärkt prozessorientierte Betrachtungsweise erweitert, in der man untersuchte, wie der Softwareentwicklungsprozess werkzeuggestützt und methodisch ergänzt werden muss, damit die resultierenden Softwareprodukte auch software-ergonomischen Qua-

litätsstandards entsprechen. Damit wurde der Disziplin Software Engineering die Disziplin *Usability Engineering* zur Seite gestellt. Im vorletzten Abschnitt dieses Beitrages werden einige aktuelle Entwicklungen im Bereich MCI aufgegriffen und vor allem Aspekte der *Visualisierung großer Datenbestände* behandelt. Persönliche *Schlussbemerkungen* beenden diesen Beitrag.

## **Der Siegeszug der graphischen Benutzeroberfläche**

Die ersten Schritte des Autors in das spannende Forschungsfeld MCI erfolgten im Jahre 1988 unter der Ägide von Prof. Tjoa am damaligen Institut für Statistik und Informatik der Universität Wien. Im Rahmen eines Projektes mit der Gewerkschaft der Privatangestellten (Projektleiter: Dr. Paul Kolm, Gewerkschaft der Privatangestellten) wurde ein Handbuch zur menschengerechten Gestaltung von EDV-Systemen im Bürobereich entwickelt [18]. Ziel dieses Vorhabens war es, ein Vorgehensmodell sowie Methoden und Gestaltungsrichtlinien zur ergonomischen Gestaltung von Software im Bürobereich zu erarbeiten, die vor allem Softwareentwickler als Handreichung dienen sollten. Als Vorbild dieses Vorhabens dienten Handbücher zur hardware-ergonomischen Gestaltung von Bildschirm- und Büroarbeitsplätzen, die in den 80er Jahren sehr populär waren. Die graphische Benutzeroberfläche, oder neudeutsch GUI für Graphical User Interface, war aufgrund der Verfügbarkeit von grafikfähigen Bildschirmen sowie Dank des Erfolges des Apple Macintosh und des Fensterverwaltungssystems namens Windows der Firma Microsoft gerade dabei, die bisherigen Formen der MCI zu revolutionieren. Bis dahin beherrschten alphanumerische Benutzeroberflächen – meist mit einfacher Menüführung bzw. mit kommandosprachlicher Dialogführung – die Szene, die auf sogenannten unintelligenten Terminals (sprich Endgeräten, die über keine Möglichkeit der Graphikdarstellung verfügten) betrieben wurden. Die GUIs in Kombination mit Zeigeinstrumenten (z.B. der Maus) eröffneten völlig neuen Benutzergruppen den Zugang zum Computer. Gepaart mit der raschen Verbreitung von sogenannten Endbenutzerwerkzeugen (z.B. Textverarbeitungs-, Graphik- oder Tabellenkalkulationsprogrammen) konnten eine Vielzahl von Arbeitsplätzen mit Computern ausgestattet werden, die bisher von den Möglichkeiten des Computers nicht unmittelbar profitieren konnten. Neben den Endbenutzerwerkzeugen spielte in diesem Prozess auch der Siegeszug von leistungsfähigen, meist relationalen Datenbanksystemen – gepaart mit mächtigen Transaktionsprogrammen zur Durchführung von typischen operationalen Geschäftsprozessen – eine wichtige Rolle. Ein Forschungsfeld, das Prof. Tjoa ja schon seit seinen wissenschaftlichen Anfängen maßgeblich mitbeeinflusst hat. Durch die Kombination von datenbankbasierten Transaktionssystemen mit graphischen Benutzeroberflächen auf den gerade sich ebenfalls rasch verbreitenden PCs als Endgerät, eröffneten sich weite Anwendungsfelder, die beinahe alle

Endgerät, eröffneten sich weite Anwendungsfelder, die beinahe alle Geschäftsprozesse eines Unternehmens oder einer Behörde erfassten. Die wesentlichen Vorteile der graphischen Benutzeroberfläche basierten auf einer Reihe von Faktoren. Die mentale Belastung des Benutzers wurde verringert, da er nicht durch verschachtelte Menüstrukturen navigieren bzw. keine kryptischen Kommandosprachen erlernen musste. Alle für die Interaktion mit dem Computer erforderlichen Informationen werden mittels Fenster in Kombination mit graphischen Menüs (Hauptmenübalken mit Pull-down Menüs) und Icons angezeigt. Damit reduzierte sich die mentale Belastung des Benutzers auf das (Wieder-)Erkennen der Bedeutung von Menüeinträgen bzw. Icons. Durch die Möglichkeit der direkten Manipulation von Objekten am Bildschirm mit einem Zeigeeinstrument wie es beispielsweise die Maus darstellte, konnten grundlegende Kommandos auf intuitiv erfassbare, direkt-manipulative Art und Weise ausgeführt werden (z.B. das Verschieben eines Files von einem Folder in einen anderen Folder).

Dieses massive Vordringen des Computers – wie gezeigt unter anderem auch Dank des Erfolges der GUIs - bewirkte auch eine Reihe von Konsequenzen, mit denen der typische Informatiker bzw. Wirtschaftsinformatiker bis dahin nicht konfrontiert worden war und die neue Lösungswege erforderten. Dies soll anhand der folgenden Sachverhalte exemplarisch verdeutlicht werden:

- Softwaregestaltung wurde auch zur Arbeitsgestaltung.
- Die Softwarequalitätssicherung erforderte neue Methoden zur software-ergonomischen Qualitätssicherung.
- Die werkzeugmäßige Unterstützung des Interaktionsdesigns bei der Softwareentwicklung und die Ergänzung des Software Engineering um Methoden des Usability Engineering.

## **Softwaregestaltung wird (auch) zur Arbeitsgestaltung**

Das umfassende Vordringen des Computers in viele Anwendungsbereiche eines Unternehmens oder einer Behörde bewirkte, dass mit dem Einsatz des Computers auch die bisherigen Aufgabenverteilungen und oft auch die bisherige Organisationsgestaltung neu überdacht werden mussten, um die Potentiale dieses neuen Werkzeuges voll ausschöpfen zu können. Dies war bei den bis in die frühen 80er Jahre dominierenden zentralen Rechenzentren in den Unternehmen nicht der Fall, da die Auswirkungen des Einsatzes von Computern für die meisten Arbeitsplätze nur indirekte Folgen hatten (meist in Form von dicken, schwer verständlichen Computerausdrucken, die in regelmäßigen Abständen auf den Schreibtisch landeten). Jetzt konnten aber aufgrund den einfach zu bedienenden, auf leistungsfähigen Datenbanken basierenden Anwendungsprogrammen ganze Geschäftsprozesse

in Software abgebildet und unmittelbar am Arbeitsplatz bereitgestellt werden. Dies war in der Regel keine simple 1:1 Abbildung der bisherigen Geschäftsprozesse, sondern der Computer eröffnete ganz neue Möglichkeiten der Aufgaben- und Organisationsgestaltung. Typische Schlagworte dieser Zeit waren die „Rundumsachbearbeitung“, „One-Face-to-the-Customer“ oder „Business Process Re-Engineering“. Dadurch entstanden Arbeitsplätze mit völlig neuen Aufgabenzuschnitten und der jeweilige Stelleninhaber bekam leistungsfähige Programmpakete auf seinem Computer zur Verfügung gestellt, die ihm die Erledigung, in sich geschlossener Geschäftsvorfälle, ermöglichten. Vielen Unkenrufen zum Trotz, eröffnete der Einsatz des Computers damit eine bisher nicht da gewesene, anspruchsvollere und damit menschengerechtere Aufgabengestaltung an vielen Arbeitsplätzen im Büro- und Verwaltungsbereich. Diese neuen Aufgaben zeichneten sich beispielsweise durch Eigenschaften wie Ganzheitlichkeit, Persönlichkeits- und Lernförderlichkeit aus. Dies bedeutete aber auch für den Softwareentwickler, dass er im Zuge der Entwicklung und Einführung von Anwendungsprogrammen in Gestaltungsbereiche vorstieß, die bisher nicht die seinen waren. Gerade die MCI hat diese Problematik früh erkannt und als Forschungsdisziplin versucht, Methoden, Werkzeuge und auch Gestaltungsprinzipien zu entwickeln, die den Informatiker in die Lage versetzten auf diese neuen Möglichkeiten der Softwaregestaltung adäquat zu reagieren. Dies führte dazu, dass die klassischen Methoden des Software Engineering durch Methoden und Techniken des sogenannten Usability Engineering ergänzt wurden, beispielsweise um Methoden der Aufgabenanalyse oder des Work Re-Engineering. Bei diesen Methoden steht nicht die adäquate Gestaltung der Daten oder der Funktionen im primären Blickfeld, sondern hier wird versucht, unter Berücksichtigung der Organisations- und Aufgabenerfordernisse sowie der Charakteristika der zukünftigen Benutzer, gebrauchstaugliche Software zu entwickeln unter *gleichzeitiger* Ausschöpfung der Gestaltungspotentiale, hinsichtlich einer ergonomischen Aufgaben- und Organisationsgestaltung. Gebrauchstaugliche und aufgabenangemessene Software zeichnet sich dadurch aus, dass sie sowohl effektiv (sprich für den jeweiligen Aufgabenbereich die erforderlichen Funktionen bietet) als auch effizient (sprich unter Minimierung der notwendigen Dialogschritte) genutzt werden kann. Zusätzlich sollte der Benutzer auch ein subjektives Gefühl der Zufriedenheit (Joy of Use) bei der Benutzung des Arbeitsmittels Computer empfinden.

In dem bereits genannten Projekt mit der Gewerkschaft der Privatangestellten wurde Ende der 80er Jahre von der Forschungsgruppe von Prof. Tjoa ein entsprechendes Handbuch entwickelt, das den Anwendungsentwicklern von Software eine Vielzahl von methodischen Hinweisen und auch zu beachtende Gestaltungsrichtlinien an die Hand gab [18]. Die Ergebnisse des Projektes wurden Anfang der 90er Jahre auch in einer Monographie mit dem Titel „Software-Ergonomie“ veröffentlicht

[4]. Es war zum damaligen Zeitpunkt eines der wenigen deutschsprachigen Bücher, das sich dieser Thematik angenommen hatte. Das dort vorgestellte Vorgehensmodell (z.B. iterativ, prototypingorientiert und mit Einbeziehung der Benutzer in die verschiedenen Phasen der Softwareentwicklung), die empfohlenen Methoden (z.B. Evaluationsverfahren zur Bewertung der ergonomischen Qualität) und die grundlegenden Gestaltungsprinzipien haben auch nach über 10 Jahren noch ihre Gültigkeit!

## **Software-ergonomische Qualitätssicherung**

Methoden der Qualitätssicherung, basierend auf unterschiedlichen Metriken, haben eine lange Tradition im Bereich der Software-Entwicklung (z.B. Inspektion, Review, Walkthrough). Da aber durch den Erfolg der GUIs die Programmierung des Teiles eines Anwendungsprogramms, der für die Interaktion zwischen dem Benutzer und der Anwendung zuständig ist, immer wichtiger wurde, zeigte sich, dass die klassischen Qualitätssicherungsverfahren sowohl vom Standpunkt der Metriken als auch hinsichtlich der verfügbaren Testmethoden Defizite aufwiesen. Dies ist bei der bis dahin bestehenden geringen Bedeutung der MCI für den Erfolg eines Programms auch nicht weiter verwunderlich. So standen bis dahin vor allem Fragen der Performance, der Portabilität, der Korrektheit eines Programms im Vordergrund. Das Kriterium der Benutzbarkeit bzw. der Gebrauchstauglichkeit (Usability) gewann jetzt aber maßgeblich an Bedeutung. Die Forschungsgruppe von Prof. Tjoa, welcher der Autor zum damaligen Zeitpunkt angehörte, initiierte daher 1988 ein Forschungsprojekt mit der damaligen Forschungsgruppe Mensch-Maschine-Kommunikation der GMD in St. Augustin/Bonn<sup>1</sup>, das die Entwicklung eines Testverfahrens zur Bewertung der software-ergonomischen Qualität von Software zum Ziel hatte. Das Projekt wurde von Seiten der GMD von Prof. Dr. Reinhard Oppermann geleitet. Das Verfahren trug den Namen EVADIS (Evaluation von Dialogsystemen) und war eines der ersten software-ergonomischen Testverfahren. Es wurde 1992 in einer überarbeiteten Version in Buchform veröffentlicht [9]. Anhand von ergonomischen Dialogprinzipien (z.B. Erwartungskonformität, Selbstbeschreibungsfähigkeit, Fehlerrobustheit) und von Prüfitems konnte eine Beurteilung der software-ergonomischen Qualität eines Programms erfolgen. Es handelt sich dabei um ein heuristisches Evaluationsverfahren, das von einem Experten (z.B. Usability Engineer) durchgeführt wird. Das Verfahren wurde vor allem sehr erfolgreich in der Lehre für Studierende der Informatik bzw. Wirtschaftsinformatik eingesetzt, da es die Studierenden in praktischer Art und Weise mit den wesentlichen ergonomischen Dialogprinzipien und den dabei zu

---

<sup>1</sup> Heute Fraunhofer Institut für Angewandte Informationstechnik (FIT)

beachtenden Gestaltungsregeln vertraut machte. Dies führte auch dazu, dass das Evaluationsverfahren EVADIS von der GI im Rahmen des von ihr vorgelegten Software-Ergonomie Curriculums als Evaluationsverfahren für die Lehre empfohlen wurde.

## Die werkzeuqmäßige Unterstützung des Interaktionsdesigns

Die intensive Beschäftigung des Autors mit Fragestellung der software-ergonomischen Qualitätssicherung hatte gezeigt, wie wichtig die Unterstützung der Anwendungsentwickler im eigentlichen Entwicklungsprozess mit software-ergonomischem Gestaltungswissen ist. Nicht erst wenn der Entwicklungsprozess abgeschlossen ist, sollte die Unterstützung der Entwickler einsetzen (z.B. durch Evaluationsverfahren), sondern bereits im Entwicklungsprozess selbst. In einem nachfolgenden Forschungsprojekt IDA (Interface Design Assistance) entwickelte der Autor als Gastforscher in der damaligen Forschungsgruppe Mensch-Maschine Interaktion der GMD in St. Augustin/Bonn<sup>2</sup> ein User Interface Management System (UIMS), das über eine Reihe von Funktionen verfügte, die den Programmierer bei der Realisierung von ergonomischen GUIs unterstützen [15]. Abbildung 1 zeigt das als Basis genutzte UIMS namens Dialog Manager der ISA GmbH (Stuttgart) und die zusätzlich mittels einer „IDA-Toolbar“ angebotenen Funktionen.

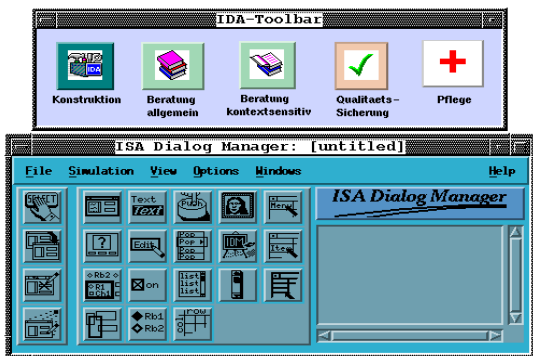


Abbildung 1: IDA – Interface Design Assistance (Stand 1995)

Dazu gehörte eine objekt-orientierte Bibliothek von wiederverwendbaren Dialogbausteinen (Konstruktion genannt), ein hypermediales Beratungssystem und eine wissensbasierte Qualitätssicherung. Die Bibliothek beinhaltete eine Reihe von anwendungsspezifischen Dialogbausteinen, die nicht nur das statische Look eines GUIs umfassten (z.B. ein Dialogfenster zum Anlegen eines

<sup>2</sup> Heute Fraunhofer Institut für Angewandte Informationstechnik (FIT)

neuen Kunden), sondern auch das dynamische Feel (z.B. das Öffnen weiterer Dialog- oder Mitteilungsfenster, die mit dem ursprünglichen Dialogfenster in einem semantischen Zusammenhang stehen). Die Dialogbausteine wurden nach software-ergonomischen Prinzipien für bestimmte Anwendungsdomänen entwickelt und ermöglichten dem Anwendungsentwickler auf wiederverwendbare Dialogkomponenten zurückzugreifen. Das hypermediale Beratungssystem war in der Lage, Anfragen des Entwicklers zu beantworten (z.B. Wann setze ich sinnvollerweise ein Notebook ein?) und lieferte in anschaulicher Form Erklärungen anhand von interaktiven Beispielen (mit Text, Ton, Animation und Interaktion). Siehe dazu Abbildung 2.

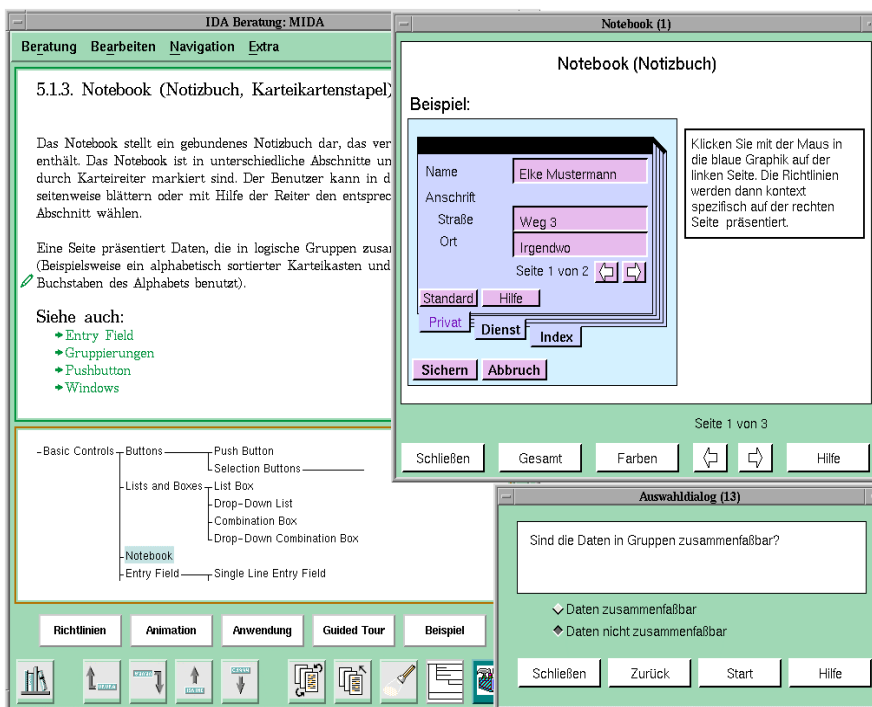


Abbildung 2: IDA – hypermediales Beratungssystem (Stand 1995)

Die wissensbasierte Qualitätssicherung stellte eine Art software-ergonomischen Debugger dar. Er konnte software-ergonomische Regelverstöße erkennen, sowie automatisch korrigieren und lieferte die notwendigen Erklärungen unter Einsatz des hypermedialen Beratungssystems. Siehe dazu Abbildung 3. Die Qualitätssicherung basierte auf einem Expertensystem, das ergonomische Gestaltungsregeln in seiner Regelbasis beinhaltete und mittels seiner Inferenzmaschine diese Regeln auf das jeweilige GUI anwenden konnte.

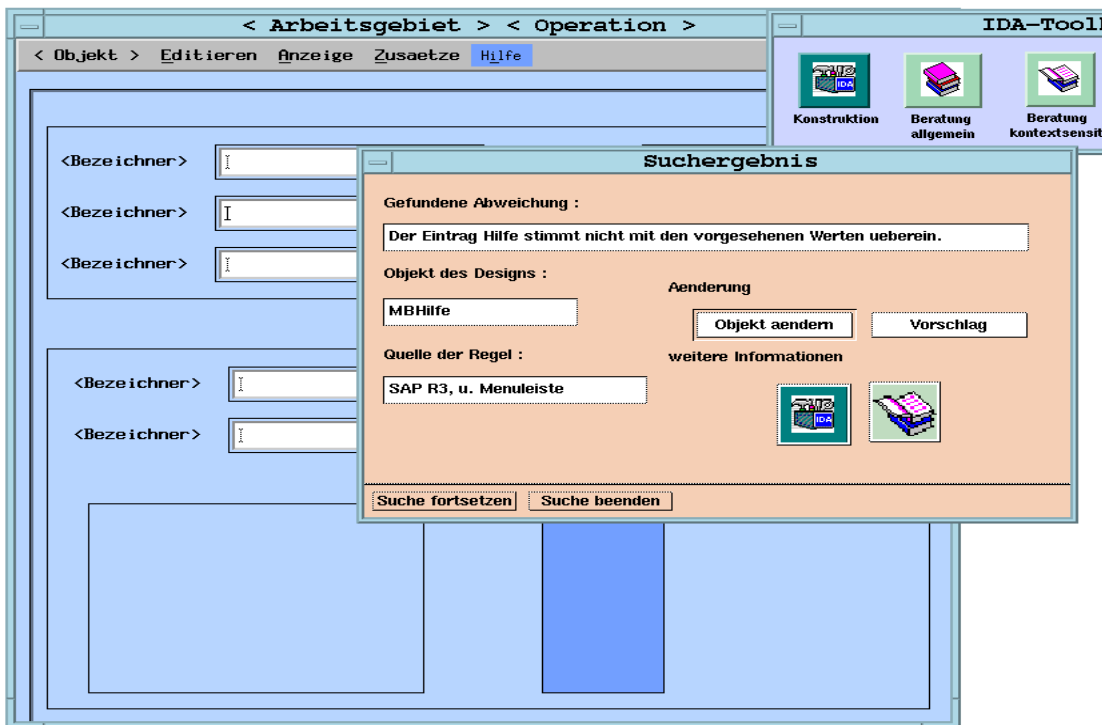


Abbildung 3: IDA – Qualitätssicherung (Stand 1995)

Mitte der 90er Jahre war IDA eines der wenigen Forschungssysteme im deutschsprachigen Raum, das auf nationalen und internationalen Konferenzen gezeigt werden konnte (siehe [1], [11], [12], [13], [14]). Der Siegeszug der Programmiersprache Java, die zunehmende Verbesserung der Ausbildung der Anwendungsentwickler im Bereich der Entwicklung von GUIs sowie der Trend zu einheitlichen Programmierumgebungen (keine Trennung mehr in UIMS zur Entwicklung des GUIs und in Programmierwerkzeug zur Entwicklung des Anwendungscodes) haben dazu geführt, dass UIMS heute nur mehr eine geringe Bedeutung haben. Wiederverwendbare Bibliotheken von Dialogbausteinen und online verfügbare Style Guides (z.B. Java Look and Feel Design Guidelines<sup>3</sup>) sind heute nicht mehr wegzudenken.

Neben der werkzeuggestützten Unterstützung des Anwendungsentwicklers bei der Programmierung des GUIs wurden in den folgenden Jahren im Bereich MCI wesentlich umfassendere, methodische Unterstützungsmaßnahmen entwickelt. Diese beschränkten sich nicht mehr primär auf die Qualitätssicherung durch die Entwicklung ausgereifter Evaluationsverfahren oder von Werkzeugen zur Unterstützung des Implementierungsprozesses, nein, der gesamte Softwareentwicklungsprozess

<sup>3</sup> <http://java.sun.com/products/jlf/ed1/guidelines.html>



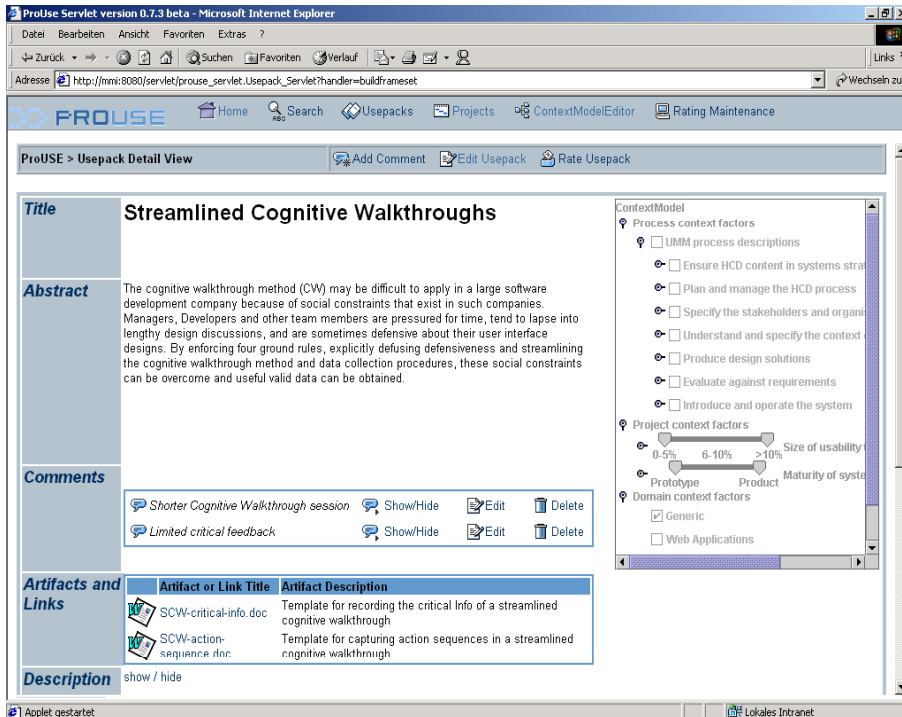
wurde zum Gegenstand der Forschung. Unter dem Schlagwort Usability Engineering wurden eigene Vorgehensmodelle entwickelt, die sich durch Eigenschaften wie iterativ, prototyping-orientiert und evaluationszentriert auszeichneten. Ergänzend wurden eine Vielzahl von Methoden zur Erfassung der Anforderungen (vergleichbar dem Requirement Engineering) entwickelt, die für die erfolgreiche Umsetzung der MCI maßgeblich sind. Als Stichworte seien hier genannt: Essential Use Cases, Hierarchische Aufgabenanalyse, Aufgabenszenarien, Fragebögen zur Erfassung von Benutzereigenschaften, Contextual Design, Participatory Design<sup>4</sup>. Auch spezielle Methoden zur Unterstützung der eigentlichen Entwurfsarbeiten wurden entwickelt. Auch hier mögen die folgenden Stichworte zur Verdeutlichung genügen: Designszenarien, Prototyping mittels Mock Ups oder Storyboards, HCI Patterns, Gestaltungsprinzipien und Normen.<sup>5</sup>

Aktuelle Arbeiten des Autors in diesem Forschungsbereich versuchen die Softwareentwickler während des gesamten Entwicklungsprozesses mit dem notwendigen Methodenwissen (im Sinne von base practices) aber auch Erfahrungswissen aus vorangegangenen Projekten (sogenannten best practices) zu unterstützen. Dazu wird im Rahmen eines Projektes mit der Forschungsabteilung der Daimler Chrysler AG in Ulm eine Forschungssystem namens PROUSE (Process centred Usability Engineering Environment) entwickelt [7], [8]. Dieses stellt eine Art Wissensmanagementsystem für Usability Engineers dar, in dem neben dem notwendigen Methodenwissen auch sogenannte Erfahrungspakte zur Verfügung gestellt werden. Dazu wird Erfahrungswissen semi-formal beschrieben und in einem Repository verwaltet. Durch geeignete Mechanismen, wie beispielsweise Filterwerkzeuge, die unter Berücksichtigung der jeweiligen Rahmenbedingungen eines Projektes nur bestimmte Methoden bzw. Erfahrungspakte vorschlagen oder einem ausgefeilten statistischen Bewertungsverfahren, das auf den Beurteilungen der Benutzer der Wissensinhalte basiert, werden dem Benutzer des Systems maßgeschneiderte Wissensbausteine zur Verfügung gestellt. Diese beinhalten auch eine Reihe von Artefakten (z.B. Fragebögen, HCI Patterns, Templates, Codebeispiele), die vom Usability Engineer unmittelbar bei seiner Arbeit genutzt werden können. Abbildung 4 zeigt ein derartiges Erfahrungspakt, wie es sich dem Benutzer von PROUSE präsentiert.

---

<sup>4</sup> Einen aktuellen und umfassenden Überblick über die Methoden des Usability Engineering gibt das sehr lesenswerte Buch zu diesem Thema von D. Mayhew [6].

<sup>5</sup> Einen guten Überblick zum aktuellen Stand des Wissens im Forschungsbereich HCI findet man in [10].



**Abbildung 4: ProUSE - Process centred Usability Engineering Environment (Stand 2002)**

Das Usability Engineering stellt heute eine breite Palette an Methoden und auch Werkzeugen zur Verfügung, die den traditionellen Software Engineering Prozess umfassend ergänzen. Unter dem Schlagwort UMM (Usability Maturity Modell) werden heute bereits methodische Vorgehensweisen entwickelt (in Anlehnung an CMM), die es einer Entwicklungsorganisation ermöglichen, den Reifegrad hinsichtlich Methodenkompetenz im Bereich Usability Engineering zu steigern.<sup>6</sup>

## **Herausforderungen bei der zukünftigen Gestaltung der Mensch-Computer Interaktion**

In den 90er Jahren hat die MCI im Lichte der Entwicklung neuer Endgeräte (Stichworte: Information Appliances, Ubiquitous Computing, Wall-sized Displays, Digital Paper) dramatische Veränderungen erfahren. In Kombination mit dem Siegeszug des Internetdienstes WWW wurden immer mehr Lebensbereiche vom Computereinsatz erfasst. Dominierten in den 80er und frühen 90er Jahren noch die Gestaltungsfragen des Computereinsatzes im Büro- und Verwaltungsbereich sowie im

<sup>6</sup> Siehe dazu beispielsweise den DATech-Prüfbaustein: Qualität des Usability-Engineering Prozesses (Version 1.2) unter [www.datech.de](http://www.datech.de).

Produktionsbereich (z.B. CIM Computer Integrated Manufacturing) die Forschungsanstrengungen, so gewannen in den letzten Jahren ganz neue Forschungsfragen an Bedeutung:

Unter dem Schlagwort *CSCW* (Computer Supported Cooperative Work) werden Anstrengungen zusammengefasst, welche die optimale Gestaltung der MCI bei vernetzten Arbeitsplätzen, bei denen mehrer Personen gemeinsam Aufgabenstellungen lösen, zum Gegenstand haben.

Unter dem Schlagwort *Ubiquitous Computing* werden Anstrengungen zusammengefasst, welche die optimale Gestaltung der MCI beim Einsatz von sogenannten Information Appliances (z.B. PDAs Personal Digital Assistant, Handys der 3. Generation, Tablets, Telematikanwendungen in Fahrzeugen) im Blick haben.

Unter dem Schlagwort *Web Usability* wird die optimale Gestaltung von Web Sites für Web Anwendungen (Schlagworte: E-Commerce, Online-Communities) zusammengefasst, wobei hier die besondere Herausforderung in der Einbeziehung von Benutzergruppen besteht, die bisher keinerlei Erfahrungen im Umgang mit dem Medium Computer hatten. Diese sollen in die Lage versetzt werden, teilweise komplexe, transaktionsbasierte Anwendungen selbständig zu bedienen (z.B. Internetbanking, Auto Konfigurator, Buchen bzw. Einkaufen im Web) und damit quasi Teile von Geschäftsprozessen selbst zu erledigen (und damit nicht mehr der Berater oder Sachbearbeiter im jeweiligen Unternehmen).

Unter dem Schlagwort *multimodale User Interfaces* wird die Entwicklung von Interaktionsformen zwischen dem Mensch und dem Computer zusammengefasst, die sich durch das Einbeziehen von Sprache, Gestik, Haptik und Mimik in die Interaktion auszeichnen. Der Einsatz multimodaler Interaktionsformen wie beispielsweise bei der Steuerung von Navigations- und Verkehrsinformationssystemen im Auto, von Kiosksystemen, von PDAs oder von Edutainment Anwendungen wird es in Zukunft unabdingbar machen, dass beim Interaktionsdesign verstärkt auch auf Erkenntnisse aus den Bereichen Design, Film, Theater und Psychologie zurückgegriffen werden muss. Eine nur auf die Informatik reduzierte Umsetzung derartiger Anwendungen wird vor allem dem Anspruch des Joy of Use nicht gerecht werden können.

Unter dem Schlagwort *User Experience* werden daher Anstrengungen unternommen, die klassischen Usability Goals um Aspekte wie „to be fun, enjoyable, pleasurable, aesthetically pleasing“ zu erweitern. Die allgegenwärtige Verfügbarkeit von interaktiven Anwendungen und Endgeräten sowie der umfassende Einsatz von Web-basierten Anwendungen hat dazu geführt, dass immer größere Bevölkerungsgruppen mit Computertechnologie in Kontakt kommen, die bisher nicht zur klassischen Benutzergruppe gehörten (z.B. Kinder und Teenager, Senioren, Computerlaien). Für diese Personengruppe erfolgt die Nutzung heute verfügbarer interaktiver Produkte wie

beispielsweise des Computers, des Handys, des PDAs, des Tablet PCs, eines Navigations- oder Kiosksystem, des digitalen Papiers, des digitalen Schmuckes oder eines E-Jackets weniger aus beruflichen Gründen, sondern zum Zwecke des Einkaufs, der Information, der Kommunikation, der Unterhaltung oder als Ausdruck eines bestimmten Lifestyles. Gerade für diese Zielgruppe ist der Aspekt des Joy of Use zumindest so wichtig, wie eine effektive oder effiziente Nutzung. Zur Erreichung dieses Zieles wurde vom Autor gemeinsam mit anderen Organisationen (ZGDV Darmstadt, Kunsthochschule für Medien Köln, FH Darmstadt, FH Erfurt, ion2s Darmstadt, GESIS Berlin) ein Forum mit dem Arbeitstitel „Informatik als Gestaltungswissenschaft“ gegründet (siehe [www.zgdv.de/IAGW-FORUM](http://www.zgdv.de/IAGW-FORUM)), welches Vertreter der Informatik (hier vor allem die Teildisziplin Mensch-Computer Interaktion) und Vertreter verschiedener Designdisziplinen (hier vor allem Kommunikations-, Graphik-, Industriedesign) zusammenbringt. Ein Ergebnis dieses Forums sollte die Entwicklung eines, von allen beteiligten Disziplinen getragenen, „Methodenkompendiums“ sein, das beim Interaktionsdesign sowohl klassische Usability als auch User Experience Goals gleichwertig berücksichtigt.

Unter dem Schlagwort *Information Visualization* werden heute Forschungsbemühungen gebündelt, welche die Visualisierung großer, oft abstrakter Datenbestände (z.B. Inhalte von Data Warehouses, Digital Libraries, des WWW) zur Unterstützung von wissensintensiven Recherche- und Entscheidungsaufgaben zum Ziel haben (z.B. Data Mining, Information Retrieval).

Im weiteren soll der Fokus auf die Herausforderungen bei der Gestaltung der MCI im Zusammenhang mit der *Visualisierung* großer Datenbestände gelegt werden. Dies ist darin begründet, dass der Autor gemeinsam mit Prof. Tjoa bereits Mitte der 90er Jahre ein Forschungsprojekt namens WWW-EIS-DWH (Web-basiertes Executive Information System basierend auf einem Data Warehouse) durchführte, dessen Ziel in der Visualisierung von entscheidungsrelevanten Daten eines Data Warehouses bestand [2]. Dazu wurde eine web-basierte Benutzeroberfläche für ein Executive Information System (EIS) entwickelt. Die besondere Herausforderung bestand dabei darin, eine möglichst intuitive Benutzeroberfläche für die Zielgruppe Top-Manager, mit geringen bis nicht vorhandenen Computerkenntnissen, unter Verwendung der noch in den Kinderschuhen steckenden Webtechnologien (z.B. HTML, Java, Javascript), zu realisieren, wobei große Datenmengen in intuitiver Art und Weise präsentiert werden mussten. Abbildung 5 zeigt beispielhaft einen Bildschirmabzug des EIS.

Alle Web-Anwendungen, so auch das Web-basierte EIS, wurden von Benutzern in umfassenden empirischen Benutzertests unweigerlich am bereits gut etablierten Standard graphischer Benutzeroberflächen, traditioneller Anwendungen (in diesem Fall an Tabellenkalkulations- und Graphikpro-

grammen) gemessen. Es zeigte sich sehr deutlich, dass die damals verfügbaren Möglichkeiten im Web-Umfeld zur Gestaltung der MCI und zur Visualisierung großer Datenmengen unzureichend waren (z.B. Antwortzeitverhalten, fehlende Interaktionselemente, beschränkte Möglichkeiten der Layoutgestaltung). Erst die Weiterentwicklung der Programmiersprache Java in Richtung einer vollwertigen Programmiersprache im Sinne der Gestaltung der MCI (Stichwort: Entwicklung der Java Foundation Classes im Projekt Swing mit dem Ziel, das Spektrum an GUI Interaktionselementen zu erweitern) und die immer neuen Features von HTML erlauben es heute, vergleichbare Interaktionsformen in web-basierten Anwendungen zu realisieren.

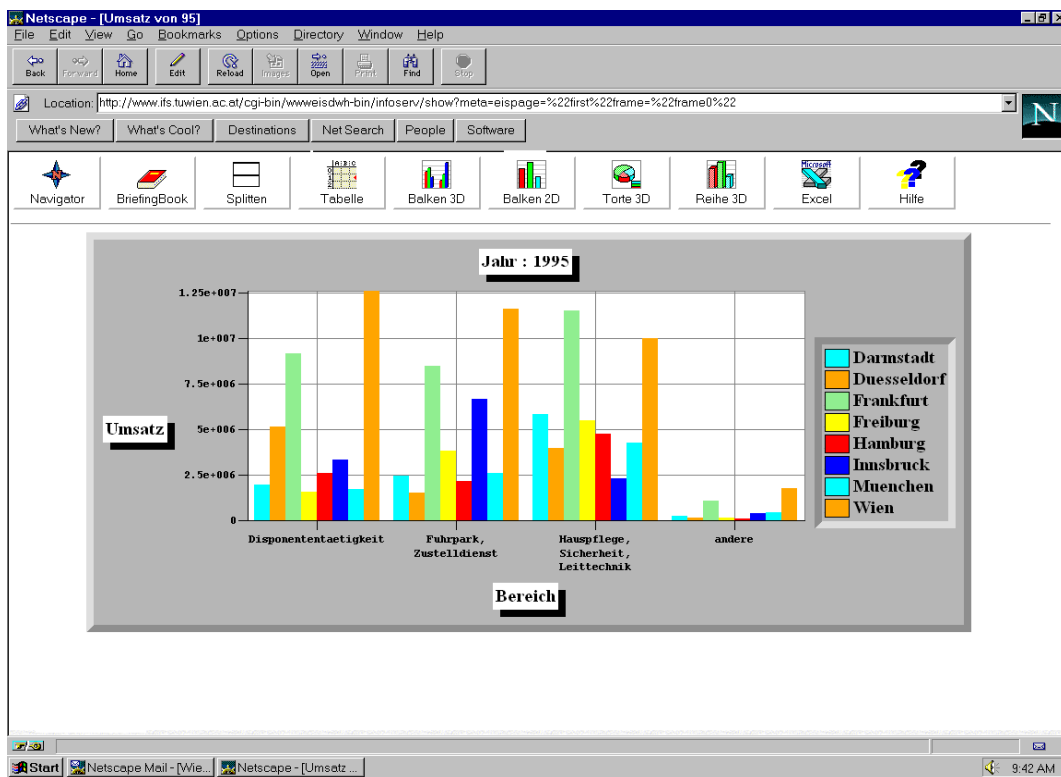


Abbildung 5: WWW-EIS-DWH – webbasiertes Executive Information System (Stand 1995)

Basierend auf den Erfahrungen des oben genannten Forschungsprojektes entwickelte der Autor im Rahmen von nachfolgenden Forschungsprojekten web-basierte visuelle Suchsysteme zur Unterstützung von Rechercheaufgaben (Visual Information Retrieval Systems). Im Forschungsprojekt INSYDER<sup>7</sup> (Information Systeme de Recherche) wurde eine Suchmaschine für das Web entwickelt (siehe [5], [16], [17]), die neben der traditionellen Listendarstellung der Suchergebnisse über weitere

<sup>7</sup> EU ESPRIT Project No. 29232

Visualisierungsformen zur Darstellung der Suchergebnisse verfügte (z.B. ResultTable, Scatterplot, Barcharts, Tilebars). Siehe dazu Abbildung 6.

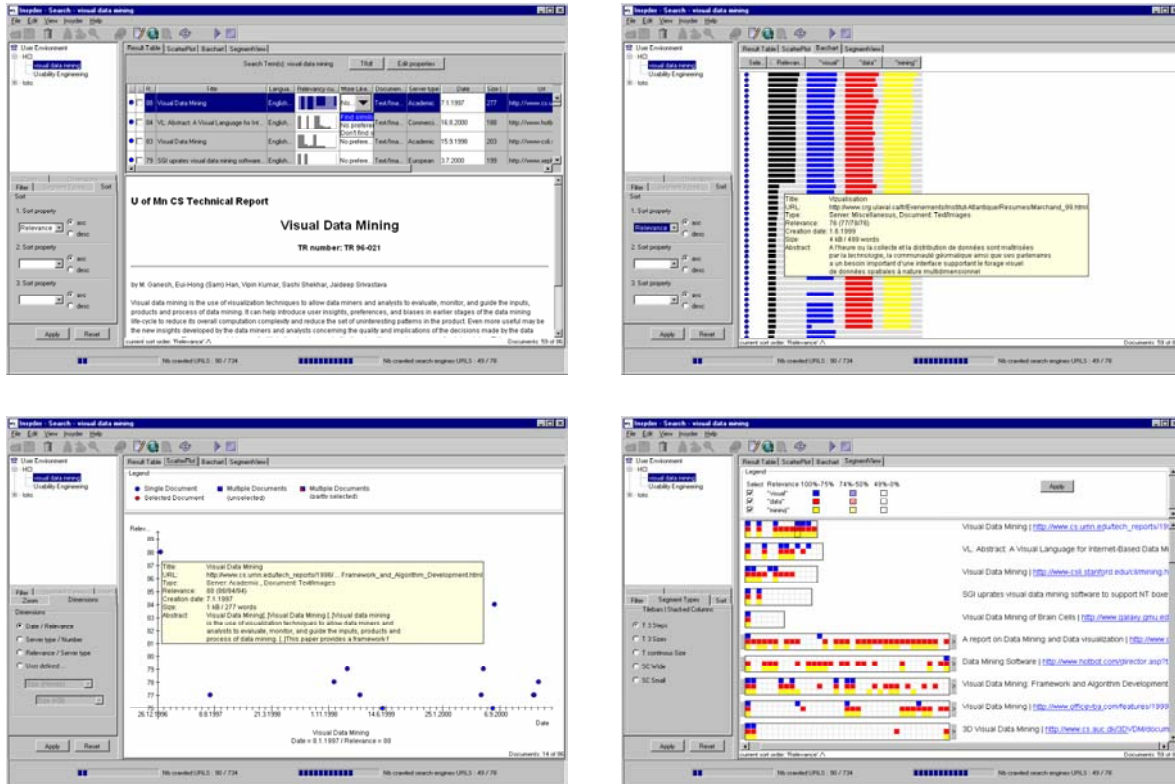


Abbildung 6: INSYDER - Visuelles Suchsystem für das Web (Stand: 2000)

Ziel des Forschungsfeldes Information Visualization ist es, abstrakte Daten mittels interaktiver Artefakte so zu visualisieren, dass der Betrachter neue Einsichten gewinnen und daraus Schlussfolgerungen ableiten kann. Durch die Interaktivität der Visualisierung, ein wesentliches Spezifikum des Mediums Computer, ist der Betrachter in der Lage, die visuelle Darstellung zu verändern. So kann er beispielsweise die Belegung der Achsen eines Scatterplots mit ihm interessierenden Betrachtungsdimensionen selbst wählen, oder er kann unter Einsatz von Zoomtechniken den ihn interessierenden Ausschnitt selbst bestimmen. Neben dem Finden von geeigneten visuellen Metaphern für abstrakte Daten – in der Regel werden Metadaten eines bestimmten Betrachtungsgegenstandes visualisiert, beispielsweise bestimmte Attribute eines Dokuments im Web (z.B. Titel, Erstellungsdatum, Größe, Sprache, Relevanz) – besteht die Herausforderung in der Gestaltung der möglichen Interaktionsformen. Hierzu wurden beispielsweise eine Reihe von sogenannten Focus + Context Techniken (z.B. Fish Eye View, Table Lense) sowie visuelle Filter (z.B. Magic Lense) entwickelt.

Die Erfahrungen des Forschungsprojektes INSYDER flossen in ein nachfolgendes Forschungsprojekt namens INVISIP<sup>8</sup> (Information Visualization for Site Planning) ein. Auch hier wird ein visuelles Suchsystem entwickelt, das Planer, beim Finden von entscheidungsrelevanten Daten für die Standortplanung, unterstützen soll. Die im Rahmen des Forschungsprojektes INSYDER entwickelten Visualisierungen wurden einer umfassenden empirischen Evaluation mit 40 Benutzern unterzogen. Die Erkenntnisse dieser Evaluation, mit Schwerpunkt auf der Beurteilung der Gebrauchstauglichkeit dieser Visualisierungen, führten zur Entwicklung neuer Visualisierungen namens SuperTable + Scatterplot [3]. Abbildung 7 zeigt die SuperTable in der Variante als Level Table und den dazugehörigen Scatterplot.

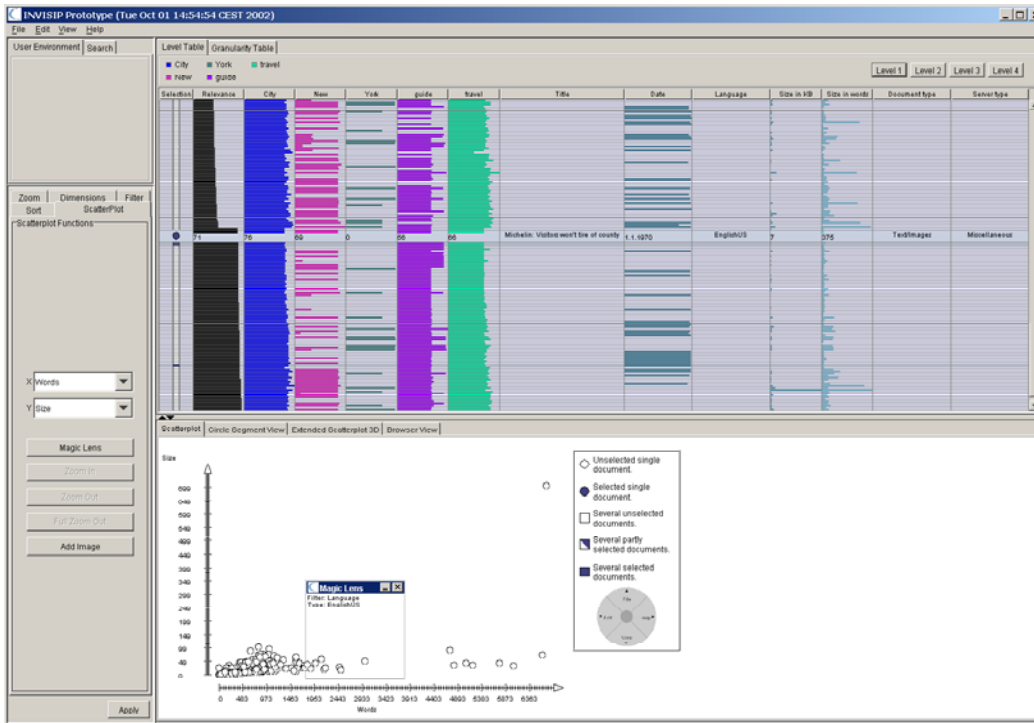


Abbildung 7: INVISIP – Visuelles Suchsystem für Geodaten im Web (Stand: 2002)

Der Scatterplot verfügt über eine Reihe von Focus + Context Techniken (z.B. Fish Eye View) sowie visuellen Filtern (z.B. Magic Lense). Die SuperTable verfügt ebenfalls über Focus + Context Techniken (z.B. Table Lense) und integriert unterschiedliche visuelle Darstellungen (z.B. Barcharts, Relevance Curve, Tilebars) in Abhängigkeit von der jeweiligen Granularität der Informationsdarstellung. Der Betrachter kann zwischen vier verschiedenen Granularitätsstufen der Informationsdarstellung wählen.

<sup>8</sup> EU Projekt IST-2000-29640 INVISIP



Abbildung 8: INVISIP – Visuelles Suchsystem für Geodaten im Web (Stand: 2002)

Der SuperTable in der Variante des Granularity Tabel ist in Abbildung 8 zu sehen und verfügt über Drill-Down bzw. Roll-Up Techniken, die es dem Betrachter beispielsweise ermöglichen, ihm interessierende Zeilen (sprich einen Metadatensatz zur Beschreibung beispielsweise eines Web Dokumentes oder eines Geodatenbestandes) zu selektieren und dann mittels Drill-Down (z.B. durch Verschieben des Sliders für die Granularität) eine immer feinere Granularitätsstufe der Informationsdarstellung zu wählen (durch Verschieben des Sliders in die andere Richtung erfolgt ein entsprechendes Roll-Up, mit dem die Granularität der Informationsdarstellung verringert wird). Die feinste Granularitätsstufe zeigt beispielweise das vollständige Web Dokument mit hervorgehobenen Suchtermen (Keyword Highlighting). Diese „Focus of Interest“ genannten Operationen kann der Betrachter sowohl auf ausgewählte Datensätze, als auch auf den gesamten visualisierten Datenbestand ausführen. Das Ziel all dieser Visualisierungsbemühungen besteht darin, den Betrachter darin zu unterstützen, aus der oft sehr großen Treffermenge einer Suchanfrage, die für ihn wirklich relevanten Daten herauszufiltern. Ein Sachverhalt, der im Zeitalter der Informationsüberflutung aufgrund der Vielzahl der mittlerweile verfügbaren digitalisierten Datenbestände immer mehr an Bedeutung gewinnt.



Vom Standpunkt der Forschung im Bereich der MCI bestehen die besonderen Herausforderungen darin, durch die Entwicklung bisher nicht verfügbarer interaktiver, visueller Artefakte und neuer Interaktionstechniken ganz neue Formen der MCI zu ermöglichen. Hier wird das Feld des traditionellen GUI Designs verlassen, das mittlerweile auch forschungsmäßig gut abgesichert ist und nach völlig neuen Wegen der visuellen Interaktion mit dem Medium Computer gesucht.

## **Persönliche Schlussbetrachtung**

Der Autor dieses Beitrages hat im Jahre 1988 seine Entdeckungsreise in die damals noch überschaubare Forschungslandschaft MCI angetreten und die obigen Ausführungen haben einige Stationen dieser Reise Revue passieren lassen. Der Autor hatte 1988 das Glück in Prof. Tjoa einen „Fremdenführer“ zu finden, der ihn bei dieser Reise auf den richtigen Weg brachte. Gerade zu dieser Zeit war es für einen „gestandenen“ Informatiker mehr als ungewöhnlich, dass er sich derartigen – oft als zu „soft“ abqualifizierten - Forschungsfragen widmete und ihnen aufgeschlossen gegenüberstand. Diese Aufgeschlossenheit und die fürsorgliche Begleitung in den ersten Jahren meiner wissenschaftlichen Laufbahn haben dazu geführt, dass meine Reise, die naturgemäß auch über viele Klippen und Abgründe führte – wie sich dies für eine ordentliche Entdeckungsreise gehört – zu einem außergewöhnlichen Erlebnis wurde! Dafür sei dem Jubilar herzlich gedankt.

## **Literaturverzeichnis**

- [1] Bachmann F., Porschen M., Ramm M., Reiterer H., Schäfer S., Simm H.: Unterstützungswerkzeuge für Entwickler von graphischen Benutzungsoberflächen, in: Böcker H.-D. (Hrsg.): Software-Ergonomie '95, Mensch-Computer-Interaktion, Anwendungsbereiche lernen voneinander, Teubner, Stuttgart, 1995, S.61-74
- [2] Kirkgöze R., Kurz A., Reiterer H., Tjoa A M.: The Relevance of Meta Modeling and Data Warehouses for Executive Information Systems, in: Tjoa A M.(ed.): Information and Communication Technologies in Tourism 1997, Wien: Springer Computer Science, 1997, pp.20-28
- [3] Klein P., Müller F., Reiterer H., Eibl M.: Visual Information Retrieval with the SuperTable + Scatterplot, in Proceedings of the 6th International Conference on Information Visualisation (IV 02), IEEE Computer Society, 2002, S.70-75
- [4] Koch Manfred, Reiterer Harald, Tjoa A Min: Software-Ergonomie, Gestaltung von EDV-Systemen - Kriterien, Methoden und Werkzeuge, Springer, Wien 1991
- [5] Mann T., Reiterer H.: A Combined Visualization Approach for WWW-Search Results in: Proceedings of the IEEE Visualization 1999 (Vis '99), San Francisco
- [6] Mayhew D.: The Usability Engineering Lifecycle, Morgan Kaufmann, 1999
- [7] Metzker E., Reiterer H.: Evidence-Based Usability Engineering, in Kolski C., Vanderdonck J. (eds.), Computer-Aided Design of User Interfaces III (CADUI 2002), Kluwer Academics, 2002, pp. 323-336
- [8] Metzker E., Reiterer H.: Use and Reuse of HCI Knowledge in the Software Development Lifecycle - Existing Approaches and What Developers Think, in: J. Hammond, T. Gross, J. Wesson (eds.), Usability: Gaining a Competitive Edge, Kluwer Academic Publishers, 2002, pp.39-55

- [9] Oppermann Reinhard, Murchner Bernd, Reiterer Harald, Koch Manfred: Software-ergonomische Evaluation - Der Leitfaden EVADIS II, Walter de Gruyter, Berlin, 1992
- [10] Preece J., Rogers Y., Sharp H.: Interaction Design – beyond human-computer interaction, John Wiley & Sons, 2002
- [11] Reiterer H.: A User Interface Design Assistant Approach, in: Brunnstein K., Raubold E. (eds.): Applications and Impacts, Information Processing '94, Proceedings of the IFIP 13th World Computer Congress 1994, IFIP Transactions A-52, Volume II, North-Holland, Amsterdam, 1994, pp.180-187
- [12] Reiterer H.: Design Assistance for User Interface Designers, in: Conference Proceedings. 1994, IEEE International Conference on Systems, Man and Cybernetics. Volume 1, IEEE, 1994, pp.19-24
- [13] Reiterer H.: User Interface Design Assistance, CHI'94 Conference, Special Interest Group (SIG) on Tools for Working with Guidelines, Boston, 1994, in: ACM SIGCHI Bulletin Vol. 27, No. 2, April 1995, pp.46-47
- [14] Reiterer Harald: IDA - A Design Environment for Ergonomic User Interfaces, in: Nordby K., Helmersen P., Gilmore D., Arnesen S. (Eds.): Human-Computer Interaction, Interact '95, IFIP Conference, Chapman & Hall, London, 1995, pp.305-310
- [15] Reiterer H.: User Interface Design Assistant, in: Hoschka P. (ed.), Computers as Assistants. A New Generation of Support Systems, Lawrence Erlbaum, Mahwah, 1996, pp.156-168
- [16] Reiterer H., Mußler G., Mann T., Handschuh S.: INSYDER – An Information Assistant for Business Intelligence, Proceedings of the 23 Annual International ACM SIGIR 2000 Conference on Research and Development in Information Retrieval, ACM press, 2000, pp.112-119
- [17] Reiterer H., Mußler G., Mann T.: A visual Information seeking system for Web search, in: Oberquelle H., Oppermann R., Krause J. (Hg.), Mensch & Computer 2001, Teubner, Stuttgart, 2001, S.297-306
- [18] Tjoa A Min, Kolm Paul, Koch Manfred, Reiterer Harald, Gärtner Johannes: EDV im Büro, Handbuch zur menschengerechten Gestaltung, Oldenbourg, Wien, 1990

Anschrift des Autors:

Prof. Dr. Harald Reiterer  
Universität Konstanz  
Fachbereich Informatik und Informationswissenschaft  
78457 Konstanz  
Web: [www.inf.uni-konstanz.de](http://www.inf.uni-konstanz.de)  
Email: [harald.reiterer@uni-konstanz.de](mailto:harald.reiterer@uni-konstanz.de)