Ausarbeitung des Referats

Agenten vs. Direkte Manipulation

Sinn und Unsinn von Agentensystemen

zum Seminar:
„Intelligente Interface-Agenten“

Ingo Steuwer
11.01.2002

Inhalt

Einleitung

Diese Arbeit entstand im Rahmen des Seminars „Intelligente Interface-Agenten“ im Wintersemester 2001/2002 an der Universität Bielefeld.

Behandelt wurden verschiedene Ansätze die Bedienung technischer Systeme, mit Hauptaugenmerk auf dem PC, durch intelligente Benutzerschnittstellen zu verbessern.

Agentensysteme, die dabei besonders betrachtet wurden, versuchen sich durch ihre Benutzerführung hervorzuheben. Im Hinterkopf ist dabei das Bild eines guten Assistenten, der ihm gestellte Aufgaben selbständig und vorausdenkend ausführt. Im verlaufe des Seminars wurde versucht zu klären, ob und wie sich dieses Bild des „Agenten“ bereits in kleinen oder großen Ansätzen verwirklicht hat. Gefunden haben sich dabei in der Masse der „Assistenten“ einige gute Ansätze und wenige gute Implementationen - aber natürlich bei weitem noch nicht die vollständige Umsetzung des menschlichen Vorbildes.

Diese Ausarbeitung behandelt bei dieser Thematik den Vergleich mit einer anderen Entwicklung im Bereich der Benutzerinteraktion, der direkten Manipulation. Als „klassische“ Benutzerschnittstelle findet sie in allen gängigen grafischen Benutzeroberflächen Verwendung, es findet jedoch auch Forschung und Weiterentwicklung statt.

So geht es im ersten Teil mit Ben Shneiderman um einen Einblick in die aktuellen Entwicklungen in der direkten Manipulation. Der zweite Teil befasst sich anschließend mit einer direkten Konfrontation der beiden Ansätze in Form einer Podiumsdiskussion zwischen Ben Shneiderman und Pattie Maes als Vertreterin der Agentensysteme.

Teil I
Einführung Direkte Manipulation

Zur Person: Ben Shneiderman

Abbildung 1 Ben Shneiderman

Ben Shneidermann ist Professor im ”Department of Computer Science” und Kopf der ”Human Computer Interaction Laboratory” (University of Maryland) [Shneiderman WWW].

Mit seiner Arbeit ist er für viele Unternehmen beratend tätig gewesen, u.a. für Apple, IBM oder Intel. Darüber hinaus hat er an der Universität von Maryland in vielen Projekten Benutzeroberflächen mit direkter Manipulation entwickelt und getestet.

Als Autor zahlreicher Bücher und Schriften ist er bekannter Vertreter der direkten Manipulation. So widmet er sich auch in der vorliegenden Zusammenfassung seines Buches „Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces “ [Shneiderman 1997] diesem Thema, die im folgenden behandelt wird.

Begriffsdefinition ”Direkte Manipulation”

Ben Shneiderman definiert den Begriff der „Direkten Manipulation“ als eine Form der direkten Kontrolle über ein (komplexes) System am Beispiel des Computers.

Dazu fordert er eine direkte Repräsentation der für die Bedienung benötigten Objekte in grafischer Form, ebenso die der zugehörigen Aktionen und Reaktionen. Die Darstellung soll dabei gleichzeitig möglichst umfassend und möglichst direkt, aber auch möglichst einfach und intuitiv sein.

Das Ziel ist die Durchführung auch komplexer Abläufe durch direkte grafische Aktionen (mit der Maus). Durch die ebenso direkte und auch ständige, verzögerungslose Reaktion des Systems gibt diese Form der Schnittstelle dem Benutzer ein Gefühl des Vertrauens und der Beherrschbarkeit.

Direkte Manipulation läßt sich auch auf andere Gebiete übertragen wie z.B. Robotersteuerung oder Bedienung von anderen technischen Geräten (Stereoanlage, Handy etc.).

Vorteile Direkter Manipulation

Dank durchgängiger und einheitlicher Repräsentation aller Strukturen eines Systems gelingt es auch unerfahrenen Benutzern schnell sich einzuarbeiten. So sollte es bei einer guten grafischen Benutzeroberfläche genügen eine Einführung in einen Teil (bzw. ein Programm) zu geben, um dem Benutzer einen Überblick über die Bedienung des gesamten Systems zu verschaffen.

Durch diese einheitliche und offensichtliche Darstellung des Systems gelingt es dem Benutzer schnell und einfach Arbeitsabläufe durchzuführen oder Probleme zu erkennen und zu lösen. Das entstehende Gefühl der absoluten Kontrolle ist dabei Grundlage für ein effizientes Arbeiten, durch das sofortige Feedback kann das Ergebnis der Aktion schnell überprüft und Fehler ausgeschlossen werden.

Durch die einheitliche Darstellung ist es für den Anwender aber auch möglich sofort Zusammenhänge und/oder Lösungsansätze zu erkennen. Die systematische Visualisierung von Datenstrukturen z.B. durch farbliche Kennzeichnung (siehe Beispiel FilmFinder Abb.3/4 S.7) ermöglicht schnelle Erkennung z.B. von Datenhäufungen (cluster) oder chronologische Veränderungen.

Programmieren mit direkter Manipulation

Als wichtige Eigenschaft und Vorteil direkter Manipulation sieht Shneiderman auch die Programmierung nach diesen Prinzipien.

Die Umsetzung erfolgt dabei insbesondere durch exemplarische Aufzeichnung einzelner Arbeitsschritte direkt im grafischen Umfeld zur späteren Wiederholung. Systeme dieser Art sind (z.T. unausgereift) schon im Einsatz z.B. bei Macrorecordern oder bei der Programmierung von Robotern. Dabei lassen sich beliebige Verschachtelungen und komplexe Konstrukte abbilden wie sie auch bei herkömmlichen Programmiersprachen verwendet werden, z.B. durch Aufrufen eines anderen Macro innerhalb der Programmierung.

Um direkte Programmierung durchgängig realisieren zu können müssen Programmierer die "*five challanges of programming in the user interface*" einhalten, die Shneiderman aus [Potter 1997] zitiert:

- **ausreichende "Generalisierung"**
  Die Möglichkeiten der Programmierung müssen generell in allen Teilen der Benutzerschnittstelle vorhanden und implementiert sein. Ist sie in nur einem Teilbereich nicht anwendbar verliert der Benutzer das Vertrauen in die Möglichkeiten dieser Art der Programmierung.
- **Zugang zu den zugehörigen Datenstrukturen und Operatoren**
  Im Rahmen der Programmierung muss ein Zugriff auf tieferliegende Datenstrukturen möglich sein um die Ergebnisse nicht auf die bereits implementierten Repräsentationen zu beschränken.
- **Leichte Programmierung, sowohl durch Beispiele als auch durch Spezifikation**
  Die Programmierung darf nicht auf das rein intuitive, meist grafische beschränkt sein sondern muss auch anschließenden Verfeinerungen oder grundsätzlichen Änderungen/Entwürfen durch (Programmier-)sprachliche Definition erlauben.
- **Einbinden und Zuweisen von Argumenten muß möglichst einfach sein**
  Das Einbinden von Argumenten sollte z.B. durch Anklicken des gemeinten Objekts vereinfacht und direkter gestaltet werden. (Beispiel Programmierung einer Formel in einer Tabellenkalkulation: Das Argument (Zelle) kann durch die Maus ausgewählt werden)

- **geringes Risiko z.B. durch Undo-Funktionen**
  Alle Manipulationen an und durch das Programm sollten vollständig (systemweit) umkehrbar sein, nur dann kann der Anwender/Programmierer unbekannte Funktionen austesten und Vertrauen in das System gewinnen.

Problemabbildung Agenten/Benutzer-Modelle <-> Control-Panels

Shneiderman sieht in der Güte der Problemabbildung innerhalb eines Systems einen Garant für die schnellstmögliche Lösung desselben durch den Benutzer. Ist das Problem nicht einheitlich und übersichtlich dargestellt gelingt es dem unerfahrenen Benutzer nicht es vollständig zu erkennen bzw. zu lösen.

Im Umfeld der direkten Manipulation, bei Shneiderman insbesondere durch die Verwendung von „Control-Panels", ist diese Voraussetzung gegeben. Sowohl die Darstellung als auch die Lösung des Problems erfolgt direkt und nur grafisch, dadurch ist der Schritt von der Problemabbildung bzw. -erkennung zur Lösung denkbar klein.

Bei Agentensystemen, wobei sich Shneiderman besonders auf anthropomorphe Repräsentationen bezieht, ist diese Art der Problemlösung nicht möglich, im Idealfall wird hier ein Problem sprachlich definiert und anschließend vom Agenten gelöst. Da die Problemspezifikation nicht wie bei der direkten Manipulation im gleichen Umfeld wie ihre Lösung stattfindet ist dieser Prozess verzögernd und fehleranfällig. Er kritisiert dabei besonders die missverständliche Sprachsteuerung bei der Eingabe als auch die nur indirekt (nach Aufforderung) mögliche Kontrolle des Systems oder Abfrage von Zwischenergebnissen.

Es stellt sich an dieser Stelle jedoch die Frage nach der Lösung von Problemen die zum Zeitpunkt der Definition der Benutzerschnittstelle noch nicht bedacht waren. Die direkte Manipulation kehrt ihren Vorteil hier in einen Nachteil, da ohne direkte (also bei Planung des Systems bedachte) Darstellung des Problems keine Lösung erfolgen kann. Ein in seiner idealisierten anthropomorphen Form dargestellter Agent jedoch kann durch die Verwendung von Sprache alle für den Menschen auftretenden Probleme zumindest verstehen und versuchen zu lösen - Sprache ist wesentlich flexibler und einfacher zu erweitern.

Versagen von Agenten mit personalisiertem Interface

Aus der Argumentation Shneiderman's geht als Fazit hervor, dass das personalisierte Interface von Agentensystemen zum Scheitern verurteilt ist. Neben den Problemen der schwammigen und prinzipiell fehleranfälligen Sprachsteuerung (auch bei der Mensch zu Mensch - Kommunikation erweist sich Sprache oft als mehrdeutig) ist auch das Benutzerfeedback eines Agenten schlechter. Die direkte Manipulation bringt es mit sich, dass die Aktionen die das System ausführt sämtlich vom Benutzer gesteuert, beobachtet und nachvollzogen - letztendlich kontrolliert werden können. Der Agent hingegen führt nach eigenem Gutdünken Systemveränderungen durch - zwar „im Interesse" des Benutzers aber häufig ohne dessen Kenntnis. Ebenso wird der Benutzer zunächst vom Prozess der Problemlösung ausgeschlossen, nur durch Rückfrage ist ein Feedback und damit ein

Eingreifen möglich, erst die Meldung des Erfolgs bzw. Misserfolgs wird automatisch an den Benutzer weitergegeben.

Shneiderman bezieht sich bei diesen Vergleichen jedoch ausschließlich auf die Vorstellung eines Agenten in anthropomorpher Form, auf die meisten implementierten Agenten lässt sich diese Kritik nur eingeschränkt übertragen - die Benutzerschnittstelle ist den üblichen (Fenster-)Systemen nachempfunden. Die Problematik „was tut mein Agent eigentlich gerade“ ist jedoch leicht zu verallgemeinern (siehe Teil II).

Beispiele

Zahlreiche Beispiele und Links für Direkte Manipulation/Programmierung finden sich auf den Internetseiten der Universität von Maryland [HCIL WWW], insbesondere der Homepage von Ben Shneiderman [Shneiderman WWW]. Im Rahmen seines Textes geht er besonders auf die Verwendung von Control-Panels sowie Multiple-Choice -Abfragen ein, da diese besonders die Fehleingabe durch Benutzer verhindern und gleichzeitig einen Überblick über die Möglichkeiten des Systems verschaffen.

Im folgenden einige Programmbeispiele auf die Shneiderman eingeht:

HomeFinder [Homefinder WWW]

Entwickelt von Ben Shneiderman und Christopher Williamson dient der „Homefinder“ als Recherchesystem für freie Wohnungen. Innerhalb eines eingestellten Umkreises um den gewünschten Mittelpunkt werden alle Wohnungen angezeigt die den eingestellten Kriterien (Kosten, Alter, Anzahl Badezimmer / Schlafzimmer genügen.

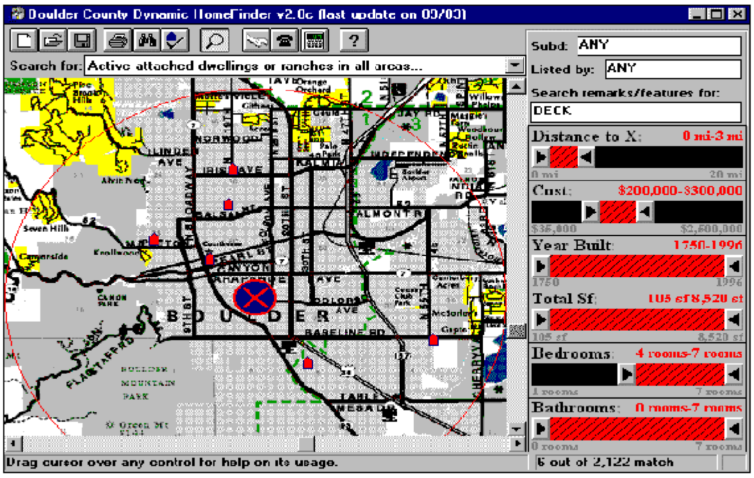

Abbildung 2 HomeFinder

Dabei kommen besondere „Slider“ zum Einsatz, die gleichzeitig die Einstellung von Minimum und Maximum erlauben.

FilmFinder

In Zusammenarbeit mit Christopher Ahlberg wurde der „FilmFinder“ ebenfalls als Recherchesystem entwickelt. Nach den Kriterien Genre, Titel, Beteiligte, Filmlänge, Popularität und Entstehungsjahr kann eine Auswahl an Filmen getroffen werden. Diese stellt sich als „Punktmenge“ dar, wobei die Lage des Punktes durch Entstehungsjahr und Popularität bestimmt wird, seine

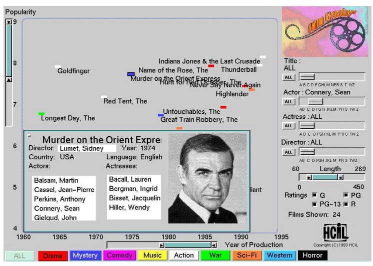

Abbildung 3 FilmFinder

Farbe durch das ihm zugeordnete Genre. Durch Auswahl eines dieser Punkte erhält man in einem neuen Fenster zusätzliche Informationen zum Film.

Shneiderman hebt an diesem Beispiel den zusätzlichen Nutzen einer guten Oberfläche hervor. So können in der Gesamtübersicht Zusammenhänge zwischen Entstehungsjahr, Popularität und Genre anhand der Farbanhäufungen erkannt werden.

Weitere Beispiele siehe: [HCIL Bsp. WWW]

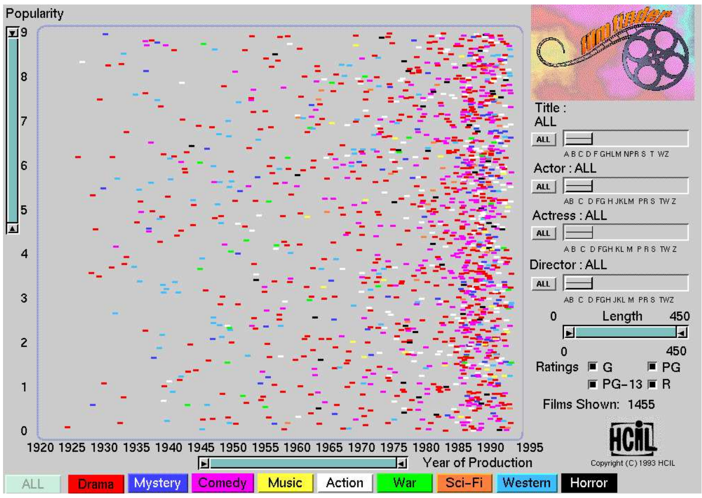

Abbildung 4 FilmFinder

Fazit Shneiderman‘s

Shneiderman ist überzeugter Vertreter der direkten Manipulation. Er sieht sie erst am Anfang ihrer Entwicklung und geht daher noch von starken Verbesserungen von Theorie und Implementation in der kommenden Zeit aus. Insbesondere hält er die menschliche Auffassungsgabe derzeit für noch nicht ausgereizt, so stellt er sich Oberflächen mit nicht 40 bis 60 Icons wie derzeit sondern vielmehr 4000 bis 6000 „glyphs“ zur animierten Visualisierung auf kommenden Displaygenerationen vor.

Kann man bei den implementierten Systemen dabei von sofortigem Feedback durch schnelle Reaktion auf Benutzereingaben ausgehen so sei die direkte Manipulation wesentlich besser beherrschbar als (anthropomorphe) Agentensysteme.

Offen bleibt vor allem die Frage der dazu aufgewendeten Zeit - was nützt es dem Benutzer wenn er die volle Kontrolle über alle systemrelevanten Dinge hat ohne die Zeit sie zu nutzen. Vielmehr wird er gerne einfache (folgenarme) Entscheidungen dem System überlassen wollen, aber dazu mehr in Teil II.

Teil II
Diskussion Direkte Manipulation vs. Agentensystem

Person Pattie Maes


Abbildung 5 Pattie Maes

Pattie Maes forscht und arbeitet theoretisch und praktisch an Softwareagenten. Als Associate Professor am MIT Media Laboratory [Maes WWW] ist sie Pionierin auf dem Gebiet der Sofware-Agenten. Sie ist Mitgründerin und führendes Mitglied der "Agent Society" und hält den "Sony Corporation Career Development Chair". Ebenfalls ist sie Autorin diverser Bücher über Softwareagenten und übernahm beratende Tätigkeiten z.B. für Apple oder Hughes Research.

Als Gründerin von "FireFly Network, Inc.", einer Softwarefirma zur Entwicklung kommerzieller Agentensysteme hat sie bereits kommerzielle Erfahrung mit der Vermarktung von Softwareagenten. „FireFly" (s.u.) ist leider inzwischen nicht mehr existent.

Im folgenden geht es um die Mitschrift einer Podiumsdiskussion zwischen Ben Shneiderman und Pattie Maes aus dem Januar 1997 anlässlich der "Intelligent User Interfaces Conference". [IUICONF WWW]

Ausgangspunkt der Diskussion

Der Gedankenaustausch fand als moderierte Podiumsdiskussion vor Publikum statt. Nachdem beide Seiten in ca. 15 min. Vortrag Ihren Standpunkt erläutert konnten hatten sie abwechselnd Gelegenheit auf die Argumentation des Gegenübers einzugehen. Anschließend fand sich für das Publikum Gelegenheit mit gezielten Fragen in die Diskussion einzuwirken.

Ziel der Veranstaltung war es sicherlich die Kontroverse bzw. den Unterschied zwischen den Ansätzen „Direkte Manipulation" und „Intelligente Agentensysteme" zu finden, was letztlich auch Gelungen ist.

Im Folgenden geht es um den inhaltlichen Ablauf der der Diskussion.

Standpunkt Ben Shneiderman

Wie im vorherigen Abschnitt bereits erläutert vertritt Ben Shneiderman mit seiner Argumentation die direkte Manipulation. In seinem kurzen Vortrag verweist er insbesondere auf das gute Benutzerfeedback bei Systemen direkter Manipulation durch direkte Reaktion auf Eingaben und der weitaus höheren visuellen Fähigkeiten des Menschen die von derzeitigen Systemen kaum ausgenutzt würden. Zusätzlich fordert er noch Anpassungsfähigkeit des Systems durch/an seinen Benutzer.

Mit Blick auf die Agentensysteme kritisiert er besonders die möglichen Fehlinterpretationen in der Zusammenarbeit Benutzer/Agent sowie die Verunsicherung des Benutzers durch fehlendes direktes Feedback des Systems. Dabei stellt er heraus dass durch eine gute grafische Darstellung eines Sachverhaltes viele Fragen und Zusammenhänge für den Menschen schon durch kurzes hinsehen geklärt werden können während Agenten dazu aufwendig programmiert werden müssten.

Beispiele

Neben einigen der Beispiele aus dem ersten Teil des Vortrages erläutert Ben Shneiderman noch das „Visible Human“ [VHuman WWW] Programm. Auf Grundlage eines gescannten menschlichen Körpers ermöglicht es das Programm durch einige Slider einzelne Schnitte des Körpers zu betrachten. Zusätzlich sind auch 3D-Ansichten beliebiger Körperausschnitte möglich.

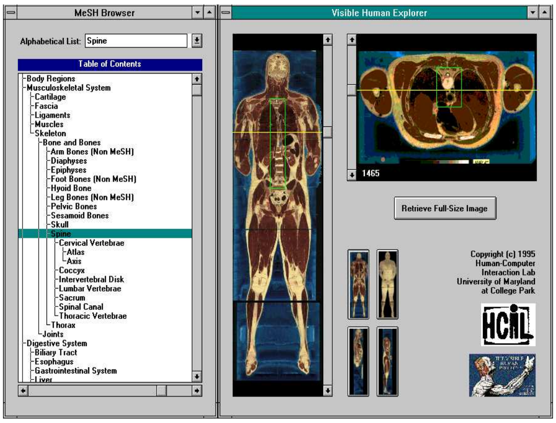

Abbildung 6 Visible Human

Standpunkt Pattie Maes

Pattie Maes übernimmt den Gegenpart und spricht sich für Ihre Forschung und Entwicklung von Agentensystemen zur Unterstützung des Menschen aus.

Im Gegensatz zu den Systemen der direkten Manipulation sieht Maes besonders Vorteile durch das selbständige Arbeiten von Agentensystemen. Da Agenten durch Beobachten des Benutzers die gängigen Verhaltensweisen und Bedürfnisse kennt kann er ihm helfen seine, täglichen Aufgaben zu erledigen. Durch „vorarbeiten“ oder Hintergrundprozesse kann eine Aufgabe längst in Arbeit sein wenn der Nutzer sie stellt. Dadurch arbeitet der Softwareagent immer wenn der Computer eingeschaltet ist - eben unabhängig vom Benutzer.

In dieser Zeit ist er insbesondere in der Lage aus der für den Menschen unüberschaubaren Informationsflut z.B. aus dem Internet diejenigen Informationen herauszufinden die den Benutzer wirklich interessieren oder zu seiner täglichen Arbeit beitragen.

Weiter kann ein Agent (auch heute schon) insbesondere kleine lästige Aufgaben übernehmen, ein Beispiel dafür sind Kauf- oder Verkaufsagenten. Diese können nach Vorgaben des Benutzers untereinander Verkäufe z.B. von Gebrauchtartikel übernehmen oder an Versteigerungen teilnehmen.

Zum Abschluss geht Pattie Maes noch auf die Benutzerschnittstelle dieser Agenten ein. Dabei sieht sie Agenten nicht als Alternative zu Direkter Manipulation - ohne eine gute Benutzerschnittstelle kann kein Agent mit seinem Benutzer interagieren. Nur wenn diese einfach verständlich ist und dem Benutzer ein Vertrauen dem Agenten gegenüber gibt kann ein System erfolgreich sein. Daher geht Pattie Maes im Zusammenhang mit Agenten nicht zwangsläufig von einer personifizierten Benutzerschnittstelle aus wie sie Ben Shneiderman‘s wesentlichen Kritikpunkten zugrunde liegt.

Beispiele

Yenta [Yenta WWW]

Mit Yenta stellt Pattie Maes ein System vor daß es sich zur Aufgabe gemacht hat Menschen gleicher Interessen über das Internet zusammenzuführen.

Die Implementierung, vorgesehen auf UNIX-Derivaten, ist ein ständig auf dem PC laufender Hintergrundprozess. Dieser nimmt Kontakt mit anderen Benutzern von Yenta auf und vergleicht die Interessensgebiete beider Nutzer auf Gemeinsamkeiten. Anhand dieser Vergleiche bildet es Interessens-"cluster" beteiligter Personen. Die Kommunikation zwischen Yenta und seinem Benutzer erfolgt dabei auf Basis eines html-Frontends.

Technisch interessant dabei ist insbesondere die Eigenschaft Yentas nicht auf einen zentralen Server zuzugreifen sondern selbständig im Internet nach anderen Nutzern von Yenta zu suchen.

Remembrance Agent [RAgent WWW]

Der „Erinnerungs-Agent", in seiner ursprünglichen Form ein Plugin für den emacs, versucht aus dem derzeit bearbeiteten Text (E-Mail) zugehörige, in der Vergangenheit bearbeiteten Texte zu finden. Ziel ist es dabei z.B. beim Schreiben einer E-Mail die letzten E-Mails des Empfängers oder andere E-Mails mit ähnlichem Betreff/Inhalt aufzulisten.

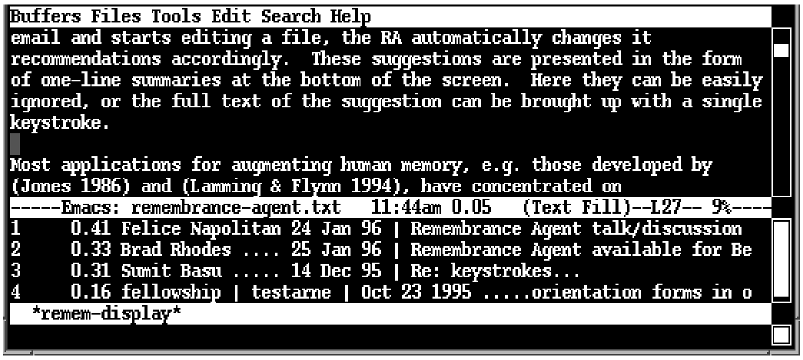

Abbildung 7 Remembrance Agent

Im emacs bettet er sich dazu am unteren Fensterrand ein und bietet ohne Zutun des Benutzers aktuelle Referenzen zur derzeitigen Arbeit. Weiterentwicklung sind u.a. in der Forschung zu Wearable Computer zu finden.

Letizia [Letizia WWW]

Bereits bekannt aus anderen Vorträgen ist Letizia, ein Browserplugin von Henry Lieberman. Es zeigt sich durch ein weiteres Browserfenster (Netscape), indem es die interessantesten Links aus der aktuell dargestellten Internetseite sowie ähnliche Seiten aufführt. Basis dafür ist die Beobachtung des Surfers aufgrund dessen eine Wortliste über seine Interessen geführt wird um neue Seiten bewerten zu können.

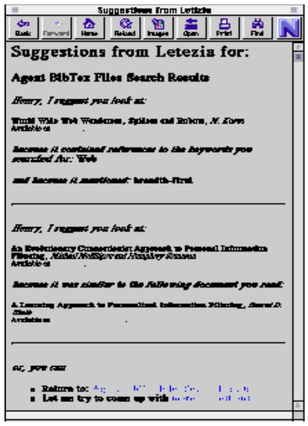

Abbildung 8 Letizia

FireFly

Ehemals auf www.firefly.com zu finden diente dieses System zum Zeitpunkt des Vortrages der Musik- bzw. Medienrecherche. Aufgrund der Nutzerangaben u.a. über dessen Lieblingsmusik versuchte das System Vorschläge für Interessante Musikstücke und Fundorte im Internet zu machen. Firefly war ein gewerblicher Ableger der Forschung von Pattie Maes der nicht mehr existiert.

Diskussion

Die Diskussion und Fragestunde ist in diesem Falle weniger eine Kontroverse sondern mehr eine Annäherung. Beide Parteien sind sich insbesondere über die hohe Priorität der Benutzerschnittstelle einig. Kein Programm, Agenten eingeschlossen, komme ohne ein gutes Interface aus: „User understanding is central, and user control is vital for peopl e to be successfull" - nur ein vom Benutzer verstandenes Programm ist ein funktioniere ndes Programm.

Dabei sind hinter dieser Oberfläche intelligente und selbständige Algorithmen erwünscht, nicht nur bei Agenten. Ben Shneiderman sieht solche Techniken auch in seinen Programmen bereits implementiert.

Wesentliche Einigung war auch über die Sprachsteuerung vorhanden, beide waren der Meinung daß diese zu mehr Verwirrungen und Interpretationsfehlern führe als wirklichen Nutzen bringe. Die klassischen grafischen Schnittstellen ziehen beide klar vor. Begründet wird dies auch dadurch, daß das Sprachzentrum des Menschen auch am logischen Denken und Planen beteiligt sei. Während der Nutzer der grafischen Oberfläche während der Bedienung des Systems im Kopf noch frei für andere Einflüsse ist muß sich der Sprachlich kommunizierende ganz auf diese Aufgabe konzentrieren.

Ausnahme sind natürlich Systeme die keine andere Eingabe ermöglichen wie z.B. Handy's im PKW.

In einigen Ansätzen sehen beide jedoch Unterschiede. Ben Shneiderman greift dabei auf seine bereits erläuterten Argumente zurück und sieht Systeme mit direkter Manipulation deutlich im Vorteil gegenüber den Agenten wenn es um schnelle Problemlösungen geht. So seien vielfach Problemlösungen für die Agenten eingesetzt werden schon durch eine geschickte Programmoberfläche direkt erkennbar. Viele Agentensysteme wie z.B. „Firefly" sind gute Programme, leben jedoch von ihrer direkten Oberfläche die Agentenstrukturen gar nicht erkennen läßt.

Pattie Maes erweitert das Spektrum der Agenten über das der direkten Manipulation, sie sieht es in wesentlich mehr Anwendungen wie z.B. bei Schnittstellen zu Robotern oder anderen (nicht-Computer-) Systemen. Daher möchte sie die beiden Themen von unterschiedlichen Blickwinkeln betrachtet sehen. Shneiderman's Systeme seien Expertensysteme die auf einen Spreziallfall optimiert wurden, sowohl Ziel als auch (geschulter) Benutzerkreis sind klar abgesteckt. Agenten hingegen versuchen auch unerfahrenen Benutzern oder ohne umfangreiche Einarbeitung Ergebnisse zu liefern, sie sind daher auf seltene oder unwichtige Aufgaben anzuwenden. Zusätzlich geht sie davon aus daß Probleme aussreichender Komplexität exisitieren die nicht mehr für die direkte Manipulation visualisierbar sind.

Vergleich der Ansätze

Die beiden Argumentationen bzw. Ansätze erscheinen nicht so unterschiedlich wie sie der Titel des Textes vermuten lässt, vielmehr sind sich die beiden Kontrahenten in den wesentlichen Punkten einig. Übereinstimmend wird festgehalten, daß Programme, egal ob agentenbasierend oder nicht, mit der Güte ihrer Benutzerschnittstelle stehen oder fallen.

Pattie Maes versucht dabei vermehrt die Benutzerschnittstelle durch den Einsatz unterstützender Agenten zu vereinfachen (im Sinne von einfacher verständlich), Ben Shneiderman durch komplexere und umfassendere Darstellungen Informationen schneller greifbar und Zusammenhänge besser erkennbar zu gestalten.

Letzendlich verfolgen beide ein gemeinsames Ziel: Schneller und besser mit dem Computer zu arbeiten. Dabei erreichen sie jedoch grundsätzlich unterschiedliche Zielgruppen.
Ben Shneiderman versucht auf der Benutzeroberfläche immer mehr Informationen unterzubringen, Pattie Maes durch mitdenkende und lernende Programme nur die wirklich wichtigen.

persönliches Fazit

Beim Studieren der Texte und Internetseiten gewinnt man den Eindruck, daß beide Seiten voneinander lernen könnten.
Wie Ben Shneiderman in der Diskussion feststellt, hat sich die Position Pattie Maes seiner stark angenähert, besonders indem sie nicht mehr an früheren Zielen wie z.B. einem Avatar als Schnittstelle zwischen Mensch und Computer festhält.

Auf der anderen Seite zielt Ben Shneiderman in seinen Programmen auf ein „selbständiges Mitdenken“, wie es auch Grundsatz eines Agentensystemes ist - z.B. bei der Beschreibung eines Macrosystems aus dem Jahre 1992: ”*...users can create macros by simply doing their tasks and letting the system construct the proper generalization automatically*” [Shneiderman 1997].

Dabei könnten die Kontrahenden nicht kontrastreicher sein. Ben Shneiderman als ”alter Hase”, dessen Forschung nach wie vor auf den ersten grafischen Benutzeroberflächen und deren Weiterentwicklungen basiert, steht mit Pattie Maes einer ”Visionärin” gegenüber, die bereits vor Jahren begann an Avatar-basierten Agentensystemen zu arbeiten - die bis heute nicht in ihrer gewünschten Form realisiert werden konnten.

Das Ergebnis der Diskussionsrunde lässt jedoch erwarten und erhoffen, daß sich die beiden Forschungsrichtungen gegenseitig unter die Arme greifen - ein guter Agent mit schlechter Benutzerinteraktion ist genauso wenig zukunftsweisend wie eine brillante Informationsdarstellung die den Benutzer durch ihre Detailverliebtheit behindert.

Literaturverzeichnis

Shneiderman WWW: Ben Shneiderman, Ben Shneiderman, http://www.cs.umd.edu/~ben/

Shneiderman 1997: Ben Shneiderman, Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces, Proceedings ofIUI97, 1997 International Conference on Intelligent User Interfaces (pp. 33-39), Orlando (FL), 1997

Potter 1997: Robert V. Potter, Eric J. Gollin, Steven P. Reiss, Thinkpad: A graphics system for programming languages, ,

HCIL WWW: University of Maryland, Human-Computer Interaction Lab, http://www.cs.umd.edu/hcil/

Homefinder WWW: Christopher Williamson, Homefinder SIGCHI 1996 Short Paper, http://www.dq.com/homefind/dhsigchi.htm

HCIL Bsp. WWW: of Maryland, Dynamic queries, starfield displays, and the path to Spotfire, http://www.cs.umd.edu/hcil/spotfire/index.shtml

Maes WWW: Pattie Maes, Pattie Maes' Home Page, http://pattie.www.media.mit.edu/people/pattie/

IUICONF WWW: IUI committee, Intelligent User Interfaces Conference,, http://www.iuiconf.org/

VHuman WWW: Ben Shneiderman, Visible Human, ftp://www.cs.umd.edu/pub/hcil/screenshots/Visible-Human/

Yenta WWW: Lenny Foner /Pattie Maes, A brief Introduction into Yenta, http://foner.www.media.mit.edu/people/foner/yenta-brief.html

RAgent WWW: Bradley Rhodes, Remembrance Agents, http://rhodes.www.media.mit.edu/people/rhodes/RA/

Letizia WWW: Henry Lieberman, Letizia: An Agent That Assists Web Browsing, http://lieber.www.media.mit.edu/people/lieber/Lieberary/Letizia/Letizia-AAAI/Letizia.html

Shneiderman 1997: Ben Shneiderman, Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces, , 1997