Ausarbeitung des Referats

Anthropomorphisierung

zum Seminar
"Intelligente Interface-Agenten"

Michael Köllmann

1. April 2002

Inhaltsverzeichnis

1 Einleitung

In dieser Ausarbeitung werden nur kleine Einblicke in das Gebiet der anthropomorphen Agenten geworfen. Zunächst möchte ich aber die Begriffe "anthropomorph" und "Agent" klären. Agent wird hier im Sinne von Software-Agent gebraucht und ein Software-Agent ist ein längerfristig arbeitendes Programm, dessen Arbeit als eigenständiges Erledigen von Aufträgen oder Verfolgen von Zielen in Interaktion mit einer Umwelt beschrieben werden kann. Es handelt sich hierbei aber nicht um die allgemein anerkannte Definition von Software-Agenten, da es hier verschiedenste und sich stark unterscheidende Definitionen gibt. Der Begriff "anthropomorph" ist da schon genauer und eindeutiger definiert. Bei Ray Eberts findet sich folgende Definition des Begriffs:
"Das Wort anthropomorphisch kann definiert werden als die Anwendung von menschlichen Qualitäten auf nichtmenschliche Gebilde."
Die Definition aus dem dtv-Brockhaus-Lexikon ist da noch etwas genauer. Demnach handelt es sich bei Anthropomorphismus um:
"die Übertragung menschl. Eigenschaften und Verhaltensweisen auf Außermenschliches (Götter, Gestirne, Tiere, leblose Wesen); charakteristisch für naiv-anschaul. Erleben und noch wenig differenziertes Denken von Kindern und Naturvölkern."
Kurz zusammengefasst könnte man jetzt sagen, bei "Anthropomorphen Interface Agenten" handelt es sich um selbstständig arbeitende Software, die sich menschenähnlich repräsentiert. Im nächsten Kapitel werden die Meinungen von fünf Wissenschaftlern zur Nützlichkeit von anthropomorphen Agenten aufgezählt. Dann wird ein Experiment zur Wirkung von anthropomorphen Interfaces auf Benutzer vorgestellt. Etwas ausführlicher wird danach das "Persona Project" von Microsoft behandelt, wo ein anthropomorpher Agent entwickelt wird.

2 Meinungen zu anthropomorphen Agenten

2.1 Brennan, Walker, Takeuchi

Brennan: Brennan hat wie auch Walker und Takeuchi Versuche durchgeführt, um die Reaktionen von Nutzern auf anthropomorphe Interfaces zu erforschen. Dabei stellte er fest, dass Menschen emotional auf Agenten mit Gesichtern reagieren. Sie interpretieren menschliche Charakterzüge in das Verhalten der Agenten hinein und deuten auch die Gesichtsausdrücke. Dieser Prozeß verlangt mehr Anstrengung vom Benutzer als das bei einem nicht anthropomorphen Interface der Fall wäre. [Koda, Maes, 1996]

Walker: Walker hat mit einem System gearbeitet, welches mit einem Gesicht ausgestattet war. Das Gesicht war in der Lage verschiedene Gesichtsausdrücke anzunehmen. In Experimenten hat Walker erforscht, wie sich Leistung und die Produktivität eines Nutzers ändern, wenn er mit einem solchen System arbeitet. Ein Benutzer der mit einem solchen System arbeitet, engagiert sich nach Walkers

Arbeit stärker. Die Nachteile können allerdings sein, dass eine solche Repräsentation mehr Aufmerksamkeit verlangt und es daher im Regelfall mühsamer ist mit ihr zu arbeiten. Je deutlicher und extremer die Gesichtsausdrücke sind, um so mehr engagiert sich der Benutzer, es nehmen aber auch die eben genannten Nachteile zu, so dass dies nicht immer sinnvoll erscheint. [Koda, Maes, 1996]

Takeuchi: Takeuchi entwickelte ein Spiel, das er mit einem Gesicht ausstattete um zu testen, wie die Benutzer auf das anthropomorphe Interface reagieren und welchen Unterschied es macht, ob ein anthropomorphes Interface vorhanden ist oder nicht. Er berichtet, dass die Nutzer sich anders verhalten, wenn sie im Spiel einen Gegner haben, der durch ein Gesicht repräsentiert wird. Ein Ergebnis seiner Arbeit war, dass sich die Nutzer mehr anstrengten, um die Reaktionen des Gesichts zu interpretieren. Er schreibt, dass diese Art die Aufmerksamkeit des Benutzers auf die Anwendung zu lenken nicht immer als Nachteil gesehen werden muß. [Koda, Maes, 1996]

2.2 Shneiderman

Shneiderman schreibt, dass auf personifizierende Interfaces weitestgehend verzichtet werden sollte. Er zählt mehrere Nachteile auf, die den Umgang mit solchen Systemen erschweren. Zum einen steigt die Anstrengung, die der Nutzer bei der Bedienung aufbringen muß, stark an, wie es auch Brennan, Walker und Takeuchi geschrieben haben. Zusätzlich erwähnt er, dass ein solches Interface größere Anforderungen an den Rechner stellt. Als drittes Argument nennt er die Möglichkeit, dass ein anthropomorpher Agent leicht von einem Benutzer überschätzt werden könnte. Wenn ein Agent aussieht wie ein Mensch und selber spricht wie ein Mensch, dann könnte der Benutzer meinen, der Agent müßte auch gesprochene Sprache verstehen. Zumindest erwartet man von einem durch einen Menschen repräsentierten Agenten ein gewisses Maß an Intelligenz. Als letzten Nachteil schreibt Shneiderman noch, dass ein solches Interface die direkte Kontrolle des Nutzers auf das System reduziert.
Shneiderman schreibt aber auch, dass es Fälle gibt, in denen es nicht von Nachteil ist, wenn man sich stärker auf das Programm konzentrieren muß. Bei Unterhaltungssoftware ist es zum Beispiel nicht schlimm, wenn man sich mehr engagiert und bei Lernprogrammen ist es sogar gewünscht, damit bessere Resultate erzielt werden. [Koda, Maes, 1996]

2.3 Maes

Über anthropomorphe Agenten schreibt Maes, dass man bei ihrer Entwicklung darauf achten sollte, dass ihr Aussehen und ihre Fähigkeiten zueinander passen. Der Benutzer sollte an der Repräsentation des Agenten erkennen können, welche Fähigkeiten er hat und welche er nicht hat. Es wäre auch von Vorteil, wenn man durch das Aussehen des Agenten auf seine Funktionsweise schließen könnte. [Koda, Maes, 1996]
Da zum Beispiel Peedy (siehe Kap.3.1.) nicht in der Lage ist, alle gesprochenen

Sätze zu verstehen, sondern nur bestimmte Sätze wiedererkennt, war es sinnvoll ihn nicht als Mensch sondern als Papagei zu repräsentieren.

3 Experiment: Pokern mit anthropomorphen Agenten

Tomoko Koda und Pattie Maes haben ein Experiment durchgeführt, um die Frage zu beantworten, wie Softwareagenten repräsentiert werden sollten, um bestimmte Wirkungen zu erzielen. Um den Effekt, den verschiedene Gesichter und Gesichtsausdrücke haben, untersuchen zu können, wurden die Reaktionen von Benutzern verglichen, die mit Agenten arbeiteten, die durch unterschiedliche Gesichter repräsentiert wurden. Zur Auswahl standen ein unsichtbarer Gegner, jeweils eine Frau und ein Mann als Karikatur und als Foto, eine Zeichnung von einem Hund und ein Smiley. Jede Figur hatte zehn verschiedene Gesichtsausdrücke, die der Benutzer beim Pokern interpretieren konnte. Koda und Maes entwickelten dazu ein Spiel, in dem der Benutzer und vier Agenten gegeneinander pokern. Ein paar Ergebnisse habe ich hier zusammengefaßt:

- Benutzer versuchen die Gesichter und die Gesichtsausdrücke zu interpretieren und werden dadurch davon abgehalten, sich zu konzentrieren.
- Wenn der Gegner durch ein Gesicht repräsentiert wurde, wurde er durchgehend als intelligenter, sympathischer und motivierender eingestuft als ein Agent, der nicht sichtbar war.
- Das Geschlecht hatte keinen Einfluß auf das Ergebnis.
- Je realistischer das Interface gestaltet ist um so sympathischer und intelligenter wirkt es.
- Ein durch ein Tier repräsentierter Gegner wirkt sympathischer als ein menschliches Gesicht.

[Koda, 1996]

4 Das Persona Project

Im nächsten Punkt wird das “Persona Project” von “Microsoft Research” vorgestellt, in dem soziale Interfaces erforscht werden, die durch einen anthropomorphen Charakter repräsentiert sind und in natürlicher Sprache mit dem Benutzer kommunizieren.

4.1 Peedy

Als Beispiel ist hier der Prototyp Peedy aufgeführt, der sich in simplen Dialogen, die aus ganz bestimmten Sätzen und Wörtern bestehen, mit dem Benutzer unterhalten kann. Peedy tritt als dreidimensionaler animierter Papagei auf, der auch durch seine Bewegungen auf Geschehnisse reagiert. Peedy verwaltet eine CD-Sammlung, kann Playlists zusammenstellen und einen CD-Wechsler bedienen.

4.1.1 Einführung

Vor dem Beginn des Persona Projekts gab es bereits eine Reihe von Bedingungen, die für ein einsetzbares Interface erfüllt sein müßten.

- Ein Assistent sollte ein interaktives Geben und Nehmen unterstützen. Assistenten reagieren nicht nur, wenn sie eine direkte Frage gestellt bekommen, sie stellen auch selber Fragen, wenn sie zusätzliches Wissen benötigen, um eine Anweisung zu verstehen, ihre Pläne zu beschreiben oder um Probleme zu erläutern. Agenten sollten mit dem Benutzer Aufgabenbeschreibungen verhandeln, um die vorhandenen Ressourcen bestmöglich auszunutzen und über Resultate Auskunft geben, sobald sie verfügbar werden.

- Der Assistent sollte die Kosten der Interaktion und der Verzögerung erkennen. Es ist nicht angebracht, bei jeder getroffenen Entscheidung zu einer Aufgabe die Bestätigung des Benutzers einzuholen. Aktuelle Systeme fragen häufig nach, weil sie einen sehr schlechten Überblick über die Konsequenzen ihrer Tätigkeiten haben. Eine assistierende Schnittstelle muß die Stichhaltigkeit ihrer Entscheidungen und die möglichen Kosten eines Fehlers gegeneinander aufwiegen, um den Benutzer nicht mit Details zu stören, die nicht wichtig sind. Der Aufwand, den Benutzer zu unterbrechen, der sich auf etwas konzentriert (oder die Kosten des Wartens, wenn der Benutzer nicht anwesend ist) muß, besonders wenn der Assistent für wiederkehrende Aufgaben eingesetzt wird, in Betracht gezogen werden.

- Der Assistent sollte Unterbrechungen effektiv handhaben. Wenn es notwendig ist, mit dem Benutzer zu interagieren, muß der Assistent dies unter der Annahme machen, dass die Aufmerksamkeit des Benutzers noch auf eine andere Aufgabe gelenkt ist. Das System sollte es erkennen, wenn der Benutzer schreibt oder telefoniert und sollte, abhängig von der Dringlichkeit der Unterbrechung, warten bis er eine Pause macht. Selbst wenn der Nutzer anscheinend untätig ist, könnte er in Gedanken sein, also sollte eine unkritische Unterbrechung in jedem möglichen Fall vermieden werden.

- Der Agent muß die sozialen und emotionalen Aspekte der Interaktion beachten. Ein menschlicher Assistent erkennt schnell, dass passendes Verhalten von der Aufgabe, der Zeit und von der Stimmung des Chefs abhängt. Um ein angenehmer Arbeitspartner zu werden, muß ein Computerassistent sein Verhalten abhängig von solchen Variablen verändern. Mit sozialen Benutzerschnittstellen hat man die Möglichkeit, lebensnahe Schnittstellen zu bilden, die das Arbeiten für den Benutzer angenehmer gestalten, aber sie müssen schnell erkennen, wenn Fragen nicht erwünscht sind.

Das Persona Project wurde 1992 gegründet, da man der Meinung war, dass ein Interface das auf direkter Manipulation basiert, nicht sinnvoll sei, wenn man den Computer als Assistenten des Nutzers betrachtet. Außerdem war man bei Microsoft der Meinung, dass der Fortschritt in der Spracherkennung und dem Verständnis natürlicher Sprache ausreichend wäre, so dass die Aussicht auf eine nützliche und funktionierende Schnittstelle ein realistisches Ziel geworden ist.

4.1.2 Überblick und Ziele

Die Entscheidung einen ersten Assistenten zu entwickeln wurde durch zwei Ziele motiviert:

- Ein solches System könnte als Testgelände für die einzelnen Technologien dienen. Die Anforderungen eines Unterhaltungsassistenten würden jede einzelne Komponente in den wichtigen Bereichen fordern und die Forschung für diese Bestandteile vorantreiben. Außerdem müssen noch viele Probleme gelöst werden, bevor Unterhaltungsassistenten erfolgreich kommerziell eingesetzt werden können, und ein Prototypsystem kann eine produktive Methode sein, das Kombinieren der komplizierten Technologien in einer zusammenhängende Architektur zu erforschen.
- Zweitens ist die Interaktion mit einem komplexen Computerassistenten völlig anders als die Arbeit mit den einzelnen Teiltechnologien. Ein anthropomorphes Interface wird beim Benutzer automatisch andere emotionale Reaktionen hervorrufen als dies traditionelle Schnittstellen tun. Zusätzlich ist durch den Gebrauch von natürlicher Sprache wahrscheinlich, dass beim Benutzer sehr hohe Erwartungen an die Kompetenz des Computers entstehen, die kontrolliert bzw. gesenkt werden müssen, um den Benutzer nicht zu enttäuschen. Es wurde erwartet, dass ein Prototyp eine nützliche Testumgebung wäre, um das Verhalten von Benutzern gegenüber anthropomorphen Interfaces zu erforschen.

Der Prototyp Peedy kann nun in drei Teile aufgeteilt werden:

1. Sprachverarbeitung.
2. Dialogmanagement.
3. Video und Audio Ausgabe.

In jedem dieser drei Punkte wurde das Projekt mit mehreren Langzeitzielen gestartet. Dann wurden realistische Minimalziele gewählt, um einen ersten Prototyp zu erstellen. Die drei Langzeitziele und ihr letzter bekannter Status werden hier aufgezählt.

Sprachverarbeitung: Es sollte möglich sein, sich in normaler gesprochener (englischer) Sprache mit dem System zu unterhalten. Das System sollte alle Anweisungen, die in seinem Aufgabengebiet liegen verstehen können. Der vorgestellte Prototyp ist in der Lage, ungefähr 150 Äußerungen zu verstehen, die im Zusammenhang mit der CD-Applikation wahrscheinlich sind.

Dialogmanagement: Das Dialogmanagement ist wahrscheinlich der Bereich, in dem die meisten Weiterentwicklungen denkbar sind. Da es als Peedys Gehirn dient, entscheidet es, wie er auf Impulse reagiert. Hier könnten Peedys Ziele, Pläne und sogar Gefühle bearbeitet werden. Um schnell an ein Ergebnis zu kommen, wurden Peedys Gedankenwege auf vordefinierte Bahnen beschränkt. Sinnvoll wäre es, verschiedene Sequenzen ineinander verschachteln zu können. Es könnte z.B. eine Korrektur innerhalb einer Auswahl stattfinden. Dies ist im vorgestellten System allerdings noch nicht möglich.

Video und Audio Ausgabe: Das Ziel des Animationssystems ist es, einen optisch und klanglich realistischen Charakter zu schaffen, der sich auch passend zu den jeweiligen Situationen verhält. Sinnvoll wäre es, beliebige Reaktionen in Echtzeit generieren zu können. Mit Techniken, die auch in Filmen eingesetzt werden, könnte man eine noch angenehmere Umgebung schaffen. Im aktuellen System müssen allerdings alle möglichen Animationen und Kameraführungen von Hand modelliert werden. Auch die möglichen Antworten, die Peedy gibt, müssen vorher aufgenommen werden.

4.1.3 Implementation

Dieser Abschnitt beschreibt die Möglichkeiten und die Implementationen der einzelnen Teilsysteme im Detail. Obwohl der Prototyp noch sehr primitiv ist, kann er bereits relativ realistisch wirkende Unterhaltungen führen. Die Implementationen hatten die Ziele, diese Illusion möglichst simpel zu erzeugen und bis zu den vorher genannten Fernzielen erweiterbar zu sein.

Sprachverarbeitung

Wie bereits beschrieben, ist es ein Fernziel der Sprachverarbeitung, es dem Benutzer zu ermöglichen, sich mit dem Agenten in natürlicher Sprache zu unterhalten. Deshalb wurde ein Interface gewählt, das auf natürlicher Sprache basiert, obwohl der Assistent selbst nur wenige Wörter und Sätze verarbeiten kann. Es ist die Flexibilität (und Vertrautheit) der gesprochenen Sprache, durch die sie so eine attraktive Schnittstelle wird: Die Benutzer entscheiden, was sie

zum Assistenten sagen möchten und drücken es so aus, wie sie es am natürlichsten finden. So lange die Bedeutung der Anweisung innerhalb des (begrenzten) Bereiches ist, den der Assistent versteht, sollte das System passend reagieren. Der Versuch, nur eine Teilmenge der englischen Sprache als Kommandosprache einzuführen, kann auf Benutzer, deren natürlichsprachliche Anweisungen nicht verstanden werden, frustrierend wirken.
Peedy ist auf einem natürlichsprachlichen System aufgebaut, das auch aus komplizierten Anweisungen semantische Strukturen konstruiert. Diese werden dann direkt in eine aufgabenbasierte Struktur verwandelt. Das Ziel ist, die Flexibilität und die Ausdrucksmöglichkeiten der natürlichen Sprache innerhalb eines begrenzten Aufgabegebietes zur Verfügung zu stellen und den Implementationsaufwand dabei gering zu halten. In dieser Hinsicht ist das System pragmatischen natürlichen Kommandosprachen am ähnlichsten, aber es wurde beschlossen, es auf natürlicher Sprache basieren zu lassen, um die Fähigkeiten des Systems erweitern zu können, wenn die Spracherkennung sich weiterentwickelt. Der Rest dieses Kapitels beschreibt die Verarbeitung der gesprochenen Sprache im aktuellen Prototyp und konzentriert sich besonders auf die Schnittstelle zwischen dem System für die Verarbeitung natürlicher Sprache und dem semantischen Modul.

Whisper Spracherkennung

Die gesprochenen Wörter werden bei Peedy von Whisper erkannt. Whisper ist ein Spracherkennungsprogramm, das sprecherunabhängig ist und in Echtzeit die Wörter in ganzen Sätzen erkennt. Alle möglichen Anweisungen, die Peedy kennt, sind in einer kontextfreien Grammatik beschrieben.

Auszug aus der Grammatik:
STATEMENT play something by ARTIST TIMEREF
TIMEREF after that
TIMEREF next
ARTIST madonna
ARTIST joe jackson
ARTIST claude debussy
ARTIST andrew lloyd webber
ARTIST synchro system
ARTIST pearl jam
ARTIST joe cocker
ARTIST bonnie raitt

Erzeugt z.B.:
Play something by madonna after that

Während der Benutzer eine Anweisung spricht, muß er einen Knopf drücken, um dem System mitzuteilen, wann die Anweisung beginnt und wann sie endet. Der Satz kann dann flüssig gesprochen werden. Die Erkennungssoftware verwendet ein Modell, das auf einer großen Zahl männlicher Sprecher basiert. Für weibliche Sprecher muß ein separates Sprachmodell benutzt werden. Das System muß also nicht erst für einen neuen Sprecher trainiert werden. Wegen des geringen Umfangs der Grammatik muß der Sprecher aber wissen, welche Sätze verstanden werden. Whisper vergleicht das akustische Signal mit den erlaubten Sätzen der Grammatik und wählt den mit dem geringsten Fehler aus. Wenn der Fehler klein genug war, reicht Whisper den Text als String an das nächste Modul weiter.

Wortersetzung

Wenn ein Benutzer mit Peedy spricht, wird er häufig Titel und Namen nennen, die der Parser leicht durcheinanderbringen könnte. Whisper ist nicht in der Lage anhand der Betonung herauszufinden, bei welchen Wörtern es sich um den Titel handelt. Wenn Peedy die Anweisung "Play before you accuse me by Clapton" erhält, vergleicht die Wortersetzung die Anweisung mit den Einträgen in der Datenbank. Zuerst werden alle exakten Treffer ersetzt:

play track1 by clapton
track1 = "before you accuse me"

Clapton wird nicht ersetzt, da der Eintrag in der Datenbank "Eric Clapton" ist. Diese Anweisung wird von Parser zurückgewiesen, da der Sinn des Satzes nicht erkannt werden kann. Als nächsten Versuch sendet die Wortersetzung den Satz ohne Wörter zu ersetzen:

play before you accuse me by clapton

Auch diese Version wird vom Parser zurückgewiesen. Als dritten Versuch sendet die Wortersetzung auch teilweise Übereinstimmungen mit der Datenbank. Für "clapton" wird "Eric Clapton" eingesetzt und der folgende Satz wird an den Parser geschickt:

play track1 by artist1
track1 = "before you accuse me"
artist1 = "Eric Clapton"

Die Anweisung wird vom Parser akzeptiert und die Wortersetzung versucht daher keine weiteren Versionen. Dieser Ansatz, die Titel zu ersetzen, funktioniert zwar sehr zuverlässig, wenn es um das Verstehen gültiger Anweisungen geht, aber wenn nach einem Titel oder Interpreten gefragt wird, der in der Datenbank nicht vorhanden ist, kann das System den Satz nicht verstehen.

Parser

Wenn die Namen ersetzt wurden, wird der Satz an den Parser weitergereicht. Der Parser erstellt dann einen semantischen Graphen. Aus der Anfrage "I'd like to hear something composed by Mozart" wird dann der Satz "I (the speaker) would like that I hear something, where Mozart composed that something."

Verschiedene Sätze produzieren identische logische Formen:

I'd like to hear something that was composed by Mozart.
I would like to hear something that Mozart composed.
I'd like to hear something Mozart composed.

Der Parser bearbeitet jeden Input in drei Schritten:

- Erstellen einer syntaktischen Skizze.
- Filtern von syntaktischen Mehrdeutigkeiten durch semantische Informationen aus Datenbanken.
- Konstruktion eines semantischen Graphen.

Der Ergebnisgraph beschreibt den semantischen Aufbau des Satzes. Nach einigen weiteren Umformungen und Nachbearbeitungsschritten wird der resultierende Graph mit vorhandenen Schablonen verglichen, die die Äußerungen darstellen, die die Anwendung versteht. Dann werden die Befehle an der CD-Wechsler, die Grafikanwendung usw. gegeben, die für Peedys Reaktion nötig sind.
[Ball, et al., 1997]

5 Fazit

Bei der aktuellen Hardware und Software haben anthropomorphe Agenten die drei von Shneiderman beschriebenen Nachteile, dass es anstrengender ist mit ihnen zu arbeiten, dass sie grössere Anforderungen an den Rechner stellen und das der Benutzer den Agenten sehr leicht überschätzt. Wie man auch bei Peedy noch sieht, ist es in den meisten Fällen einfacher, einen CD-Wechsler selbst zu bedienen als ihn von Peedy bedienen zu lassen. Man kann aber damit rechnen, dass die Hardware mit der Zeit wesentlich leistungsfähiger wird, so dass man mit entsprechender Software in der Lage sein könnte, einen benutzbaren und sinnvollen anthropomorphen Agenten zu erschaffen. Um solche Agenten dann entwickeln zu können, ist es sicherlich notwendig, vorher in diesem Bereich zu forschen, indem man einfachere Testsysteme wie Peedy aufbaut.

6 Quellen

[Ball, et al., 1997] Ball, G. Ling, D. Kurlander, D. Miller, J. Pugh, D. Skelly, T. Stankosky, A. Thiel, A. Van Dantzich, M. Wax, T.
Lifelike Computer Characters: the Persona Project at Microsoft Research
Microsoft Research, 1997, 07.02.2002
http://research.microsoft.com/research/ui/persona/chapter/persona.htm

[Koda, Maes, 1996] Koda, T. Maes, P.
Agents with Faces: The Effects of Personification of Agents
MIT, Media Laboratory, 1996, 07.02.2002
http://tomoko.www.media.mit.edu/people/tomoko/docs/HCI_final.doc.ps

[Koda, 1996] Koda, T.
Agents with Faces: A Study on the Effects of Personification of Software Agents
MIT, Media Laboratory, September 1996, 07.02.2002
http://tomoko.www.media.mit.edu/people/tomoko/thesis.html