

Leitbilder für die Gestaltung interaktiver Systeme:

Computer sind keine Partner für Menschen

Christiane Floyd
Technische Universität Berlin
FB Informatik, Sekr. FR 5-6
Franklinstr. 28/29
D-1000 Berlin 10

Einleitung

Die Frage nach geeigneten Leitbildern für die Gestaltung interaktiver Systeme hängt sehr vom Blickwinkel ab, aus dem man das Thema sieht, also auch vom Standpunkt der jeweiligen Disziplin. Mein Gebiet ist die Softwaretechnik, Ihres die Linguistische Datenverarbeitung.

Aus meiner Sicht ist die Linguistische Datenverarbeitung ein Gesamtfeld aus mehreren Teilgebieten, deren Interaktion ich von außen schwer durchschauen kann. Zum einen erkenne ich einen starken Trend in Richtung Künstliche Intelligenz, bei der sogenannten natürlichen Sprachverarbeitung. Zum anderen sehe ich die Versuche computergestützter Sprachübersetzung, die davon befruchtet werden. Letztlich bildet die natürlichsprachliche Gestaltung von Schnittstellen den Bereich, der meinem Arbeitsgebiet am nächsten steht.

Ich beschäftige mich mit Softwareentwicklungsmethoden, und zwar mit solchen Methoden, die die Systemgestaltung als Ganzes betreffen. Darunter ist auch die Gestaltung von Mensch-Computer-Schnittstellen zu fassen, wobei von meiner Seite aus die Verwendung von natürlicher Sprache ein kleines Unterthema darstellt.

Was beide Disziplinen verbinden muß, ist das gemeinsame Anliegen, tragfähige Verständnismodelle über das Verhältnis von Menschen und Computern zu haben, um eine sinnvolle, technisch anspruchsvolle und gleichzeitig menschengerechte Systemgestaltung zu unterstützen. Dabei spielen Leitbilder (oder Metaphern) eine wichtige Rolle, weil sie eine Orientierung für die wissenschaftliche Arbeit, die Durchführung von Projekten und die Gestaltung von Systemen liefern.

Bei meinen nachfolgenden Überlegungen zu solchen Leitbildern werde ich, wenn ich auch in der Linguistischen Datenverarbeitung eine Laiin bin, das Thema Sprache in den Vordergrund stellen.

Ein weitverbreitetes Leitbild vom Verhältnis zwischen Mensch und Computer geht von der Möglichkeit aus, daß es einen tatsächlichen natürlichsprachlichen Dialog mit dem Computer als "Partner" geben kann. Ich werde dies im folgenden die **Partner-Metapher** nennen. Dieses Leitbild steht in jenen Wissen-

schaftsgebieten zur Diskussion, die sich an der Künstlichen Intelligenz orientieren. Meines Erachtens ist es irreführend und nicht realisierbar.

Ein tragfähiges Leitbild für die an der Künstlichen Intelligenz orientierte Linguistische Datenverarbeitung steht für mich dagegen nicht offenkundig im Raum. Die Metapher vom Computer als Partner ist in meinen Augen deshalb verfehlt, weil sie die jeweils verschiedene Einbettung von Sprache in ganzheitliche menschliche Kommunikationssituationen außer acht läßt.

Im folgenden werde ich zuerst die Partnermetapher aus meiner Sicht kritisch hinterfragen und mich dabei insbesondere auf Einsichten evolutionär ausgerichteter Erkenntnistheoretiker beziehen.

Dann werde ich Kriterien für die Brauchbarkeit von Metaphern aufstellen, die in verschiedenen Bereichen zur Geltung kommen. In der Softwareentwicklung beeinflussen Metaphern die Sicht des gesamten Prozesses, des Berufsbildes von Softwareentwicklern und des Verhältnisses zwischen Entwicklern und Benutzern. Für die Charakterisierung des Computers im Einsatz wird die Werkzeugmetapher verwendet, sie erscheint in der Linguistischen Datenverarbeitung künstlich. Dagegen führt die Medienmetapher zu fruchtbaren Analogien mit dem Buchdruck, die mir zukunftsweisend erscheinen.

Pragmatisch induzierte Begriffsverschiebungen und prinzipielle Grenzen der Partner-Metapher

Ist es nützlich, ist es brauchbar, *Kommunikation* zwischen Menschen und Computern zu postulieren?

Zunächst erscheint dies als eine Frage der Begriffswahl. Wie sehen wir den Begriff *Kommunikation*, wie verstehen wir ihn? Hierzu gibt es eine enorme Vielfalt unterschiedlicher Sichtweisen. Für mich betrifft der Begriff *Kommunikation* wesentlich Beziehungen zwischen Lebewesen. Von daher erscheint es mir nicht natürlich, den Begriff auf das Verhältnis von Menschen zu Computern anzuwenden. Die Begründung meiner Auffassung besteht darin, daß *Kommunikation*, so wie wir sie als Menschen kennen, aus der Ko-Evolution der Lebewesen entstanden ist.

Wir können uns nun auf eine neue Konvention einlassen, indem wir beschließen, *Kommunikation* künftig auf Menschen und Computer anzuwenden. Dabei müssen wir uns aber im klaren sein, welche Veränderungen sich ergeben, wenn wir diesen Begriff nicht mehr ausschließlich auf Lebewesen beziehen. Wir lassen uns dann auf eine systematische Verzerrung oder Verfremdung der Begriffe ein, die wir verwenden – Begriffe wie *Kommunikation*, aber auch wie *Sprache*, *Dialog*, *Beratung*, *Sprechakte* und so weiter, die bisher nur zwischen Menschen zur Anwendung kamen.

Von welcher Art sind die Verfremdungen, wenn solche Begriffe auf die Mensch-Computer-Interaktion übertragen werden? Sie bestehen im wesentlichen darin, Aspekte, die wir bei der Kommunikation zwischen Menschen erkennen oder ansatzweise beschreiben können, bei der Abbildung auf den Computer voneinander zu trennen, aus ihrem Kontext zu lösen, zu vergegenständlichen und in ihrer Wirkungsweise festzulegen.

Man kann dies zum Beispiel an der Arbeit von MAASS 84 zum Partnermodell nachvollziehen. In diesem sehr ernst zu nehmenden Beitrag zur Verbesserung der Mensch-Computer-Interaktion hat die Autorin ein

Partnermodell der Mensch-Rechner-Kommunikation entwickelt, indem sie Begriffe aus der Psychologie geliehen hat – Begriffe wie *Intention*, *Partnerbild*, *Selbstbild* und dergleichen. Alle diese Begriffe sind sinnhaft, wenn wir sie auf Menschen anwenden. In gewisser Weise kann man sie auch auf Computer anwenden, und das geschieht laufend in Disziplinen, die sich mit Rechnern befassen.

Dabei tritt der Effekt ein, daß das *Partnerbild* oder das *Selbstbild* oder die *Intention* in gewisser Weise vergegenständlicht und als eigenständige Komponenten dargestellt werden. Es wird unvermittelt eine solche Komponente mit einer entsprechenden Bezeichnung (z.B. *Intention*) als gegenständlich darstellbare kreiert. Ein zweiter Effekt besteht darin, daß wir gezwungen sind, in irgendeiner Weise diese Komponenten, also Nachbildungen von Aspekten des menschlichen Bewußtseins, oder wie immer man es nennen will, zu beschreiben und zu programmieren. Letztlich werden diese Aspekte aus dem ganzheitlichen Sinnzusammenhang gelöst. Wenn wir uns diese Effekte nicht bewußt machen, so bringen sie einschneidende Veränderungen in unserer Auffassung vom Menschen mit sich.

Die Beschreibung und Programmierung solcher Komponenten ist sehr anspruchsvoll. Es kann im Rahmen meiner Argumentation nicht darum gehen, mir anzumaßen, diese z.T. bewundernswerten Leistungen einem Urteil zu unterziehen. Ich will vielmehr die Aufmerksamkeit auf die Veränderungen lenken, die selbst bei der anspruchsvollsten Programmierung auftreten, wenn menschliche Begriffe auf den Computer übertragen werden.

Während wir über Begriffe wie *Intention*, *Partnerbild*, *Selbstbild* beim Menschen sinnvoll sprechen können, ohne uns festzulegen, wie sie im einzelnen zusammenpassen, können sie bei der Übertragung auf den Computer nicht mehr so behandelt werden. Sie müssen festgeschrieben werden, und zwar durch eine Modellierung, die sie als getrennte Komponenten betrachtet und sie formal, etwa durch Regeln, beschreibt. Es ist ebenfalls erforderlich, die Interaktion zwischen diesen einzelnen Komponenten zu programmieren, d.h. Schnittstellen zwischen diesen einzelnen Komponenten herzustellen. Das sind Schnittstellen, die Ableitungsschemata darstellen, die Ergebnisse und Inferenzen ermöglichen und dergleichen. Während wir beim Menschen einem fließenden und für uns ganzheitlich beobachtbaren Geflecht von komplizierten Interaktionen einzelner psychischer Bereiche gegenüberstehen, birgt die Übertragung solcher Begriffe auf den Computer immer die Notwendigkeit zu systematischer Reduzierung und Isolation einzelner Komponenten. Wir benötigen eine diskrete Beschreibung durch Fakten und Regeln und eine statische Festlegung ihrer Zusammenhänge. Nach meiner Auffassung handelt es sich hier um eine prinzipielle Veränderung, die nicht durch den Fortschritt der Computertechnologie behoben werden kann. Aus diesem Grund halte ich die Partnermetapher auch *auf Dauer* für irreführend.

Argumente dieser Art hat zuerst DREYFUS 85 in überzeugender Weise zusammengestellt. Im Anschluß an Dreyfus gibt es derzeit international und an vielen verschiedenen Orten ernsthafte Bemühungen um erkenntnistheoretische Grundlagen der Informatik, die auch die Klärung der verwendeten Grundbegriffe einschließen.

Ein sehr bekannter Ansatz ist das Buch *Understanding Computers and Cognition* von WINOGRAD, FLORES 87. Mein eigener Beitrag besteht darin, Softwareentwicklung als *Realitätskonstruktion* (FLOYD 89) zu verstehen. Es gibt einen engen Zusammenhang zwischen diesen beiden Ansätzen. Insbesondere orientiere ich mich an vergleichbaren philosophischen Quellen, wie sie auch von Winograd und Flores benutzt werden.

Die erkenntnistheoretische Kritik der Informatik stellt die rationalistische Tradition als ausschließliche Fundierung der wissenschaftlichen Arbeit in Frage und bezieht sich auf reichhaltigere Ansätze. Zu ihnen gehören die Hermeneutik, die Sprechakttheorie und der biologisch fundierte radikale Konstruktivismus. Hier ist vor allem auf Maturana zu verweisen, der Sprache als spezifische kognitive Leistung in den Gesamtbereich biologisch fundierter menschlicher Kognition einordnet (z.B. MATURANA, VARELA 87).

Aus allen diesen Sichtweisen muß man auch die Metapher der Wesensverwandtschaft von Menschen und Computern, die der Künstlichen Intelligenz zugrunde liegt, und den damit verbundenen Anspruch, Computer könnten reale Partner für Menschen sein, in Frage stellen. Die Computer-Metapher für den Menschen ist in gewisser Weise an ihre Grenzen gestoßen. Die Grenzen liegen dort, wo man nicht mehr das Ein-/Ausgabeverhalten von Computerprogrammen als Vergleichsbasis wählt, sondern das *Zusammenwirken von Menschen und Computern* als entscheidenden Betrachtungsschwerpunkt ansieht.

Es geht also darum, den Gesamtkomplex des Zusammenspiels unseres Denkens mit dem Computer zu thematisieren. Das heißt, unser Denken und Arbeiten wieder deutlicher in den Kontext unserer menschlichen Aufgaben und Anliegen eingebunden zu sehen. Diese Auffassung harmoniert sehr stark mit linguistischen Sichtweisen, die begründen, daß menschliche Kognition prinzipiell in Kontexte eingebunden ist. Die Kontexte betreffen unsere biologische Einbindung in die Umwelt und das übergeordnete Anliegen der Autopoesis, immer wieder die eigene Identität herzustellen. Ebenso sind kognitive Vorgänge des Menschen in den Kontext der sozialen Gemeinschaft eingebunden. Vor dem Hintergrund dieser Grundannahmen scheidet die Partner-Metapher prinzipiell.

Während die Leistungsmöglichkeiten eines Computers programmiert werden, entstehen die kognitiven Fähigkeiten des Menschen im Kontext der anliegende gebundenen Ko-Evolution und werden durch sie geprägt. Beim Menschen gibt es keine Analogie für das "Programmiert-Werden".

Diese grundsätzliche Grenze wird m.E. auch künftig so bleiben, selbst wenn man etwa von dem Ansatz der neuronalen Netze ausgeht. Bei allem Respekt, den man dieser Entwicklung entgegenbringen kann, halte ich es nicht für gerechtfertigt, im Konnektionismus eine Antwort auf alle kritischen Hinterfragungen zu sehen, mit denen die Künstliche Intelligenz bisher konfrontiert wurde.

Zur Wahl von geeigneten Metaphern

Angesichts dieser Einschätzung halte ich es für eine wichtige Aufgabe, für die Linguistische Datenverarbeitung tragfähige Metaphern oder Leitbilder zu entwickeln, an denen sich die Vertreter dieser Disziplin orientieren können und die nicht irreführend sind. Ich kann als Außenstehende keine konstruktiven Vorschläge für geeignete Leitbilder für die Linguistische Datenverarbeitung unterbreiten; ich kann jedoch versuchen, die eigene Suche dieser Disziplin zu unterstützen.

Zunächst stellt sich die Frage, ob der Diskussion um geeignete Metaphern oder Leitbilder überhaupt ein zentraler Stellenwert im Rahmen der Gesamtproblematik der Softwareentwicklung zukommt oder ob nicht ganz andere Faktoren, insbesondere der gesellschaftspolitische Kontext des Designs, ausschlaggebend sind.

Nach meiner Auffassung ist die Diskussion um Metaphern nicht müßig, weil Gemeinschaften nicht nur durch gesellschaftliche Vorgaben, Situationen und Zwänge bedingt werden, sondern sich durch die Art, wie sie denken, selbst gestalten. Und dieses Denken spiegelt sich in den verwendeten Metaphern implizit wider. Das kann zweierlei bedeuten: einerseits prägen die von uns verwendeten Metaphern unsere Fragestellungen und Schlußfolgerungen; andererseits können wir Metaphern bewußt als Leitbilder für unser Vorgehen wählen. Erst in den letzten Jahren haben wir erkannt, wie wichtig es ist, den Einsatzbereich für Metaphern abzustecken und diese Implikationen explizit zu machen.

Es erscheint daher wichtig, Kriterien für geeignete Metaphern zu finden. Ich halte Metaphern für tragfähig bzw. nützlich, wenn sie in sinnfälliger Weise aus bereits bestehenden menschlichen Kontexten Bilder in das computergestützte Milieu übertragen, so daß sie dort hilfreiche Assoziationen mit sich bringen. Die Desktop-Metapher halte ich z.B. in diesem Sinne für nützlich. Sie ist jedoch eine für die Linguistische Datenverarbeitung weniger geeignete Metapher, sondern paßt eher in den Kontext des computergestützten Büros.

Entscheidend ist, daß die Metapher sich auf die Intuition eines Benutzers in einem gewohnten Kontext stützt, aus dem er oder sie Analogien ableiten kann. Natürlich kann die sinnfällige Analogie zwischen einem vertrauten Vorbild und den Eigenarten des Computers nicht nur dadurch hergestellt werden, daß man die Metapher besonders geschickt wählt, sondern auch und vor allem durch geeignete Gestaltung des computergestützten Systems. Man kann beispielsweise durch gezielte Gestaltungsmaßnahmen dafür sorgen, daß sich der Rechner den Benutzern als Medium, als Werkzeug oder als Schreibtischoberfläche darstellt. Weil sich aber Systemdesigner implizit oder explizit an Leitbildern orientieren, ist es wichtig, die dabei benutzten Metaphern sorgfältig zu prüfen.

In dem uns interessierenden Bereich können wir zwei Hauptrichtungen der Verwendung von Metaphern unterscheiden:

- sie können unser **Verständnis der Softwareentwicklung** betreffen, wobei ggf. auch das Verhältnis von Entwicklern und Benutzern mit einbezogen ist;
- sie können das **Zusammenwirken von Menschen mit Computern** erklären, und zwar sowohl die Mensch-Computer-Interaktion im engeren Sinne, als auch die Einbettung des Computers in menschliche Sinnzusammenhänge, z.B. in Arbeitsprozesse.

Die beiden Kategorien sind nicht scharf trennbar, und in beiden geht es insgesamt um das Verhältnis von Mensch zu Computer.

In der Diskussion um angemessene Metaphern höre ich oft das wichtige Argument, daß man selbstverständlich um die Unterschiede zwischen Mensch und Computer wisse, daß man aber z.T. partnerorientierte Bezeichnungen aus begrifflicher Verlegenheit auf den Rechner anwende, ähnlich wie man auch bei der Entgegennahme von Leistungen Menschen als Partner bezeichne, zu denen man im eigentlichen Sinne kein partnerschaftliches Verhältnis habe. Darüber hinaus müsse man die geschichtlich nachvollziehbare Wandlungsfähigkeit unseres Sprachgebrauchs bedenken, die dazu führen könne, daß es bei einer Weiterentwicklung der Technik einmal völlig natürlich sein könne, von Computern als Partnern zu sprechen.

Es geht mir hier nicht darum zu kritisieren, daß man Bezeichnungen von einem Erfahrungsbereich auf den anderen überträgt und hierbei einen Wandel der Sprache in Kauf nimmt. Entscheidend ist vielmehr, ob wir uns der Unterschiedlichkeit der beiden Erfahrungsbereiche bewußt bleiben. Das kann einmal der Fall sein, weil wir ohnehin in beiden Bereichen kompetent sind. Wir wissen etwa, daß ein Eisenbahnbeamter im Dienstleistungsbereich einen anderen Typus von Partner repräsentiert als etwa ein Lebenspartner. Zum anderen kann man das Bewußtsein von der Unterschiedlichkeit zweier Bereiche durch eine geeignete Wortwahl schärfen. Wenn es also nicht dasselbe ist, ob ich von einem Computer ein Ticket ausgestellt bekomme oder von einem Bahnbeamten (auch unter der Bedingung automatischer Verarbeitung natürlicher Sprache), dann können wir dies auch durch eine geeignete Wortwahl ausdrücken. Es geht aber weniger um die Wahl der Wörter als um unsere Sensibilisierung für die Unterschiede; das ist viel wichtiger.

Unsere neugeprägten Wörter ergeben sich immer aus dem bereits existierenden Sprachschatz. Wenn wir z.B. an ein Wort wie *Fernsehen* denken, so sehen wir, wie man angesichts eines neuen Gerätes im Deutschen eine sehr prägnante, aus existierenden deutschen Wörtern zusammengesetzte Bezeichnung gefunden hat, eine Prägnanz, die übrigens durch die Wortschöpfung "Glotze" noch weit überboten wird. Ähnliche Wortschöpfungen für den Computer können sich vielleicht langsam herausbilden. Ich wünsche mir, daß sie nicht eine Gleichartigkeit zwischen Menschen und Rechnern postulieren, sondern sich an der Unterschiedlichkeit orientieren.

Nun erläutere ich zuerst an meinem eigenen Fachgebiet, wie fruchtbare Metaphern wirken, bevor ich mich Leitbildern für den Computereinsatz zuwende, die für die Linguistische Datenverarbeitung hilfreich sein könnten.

Metaphern für die Softwareentwicklung

Im Software-Engineering ist die **Produktions-Sicht** der Softwareentwicklung so etabliert, daß sie meist nicht als Metapher dargestellt wird. Wegen der tiefgreifenden Unterschiede zwischen Software und anderen Produkten halte ich diese Sicht für nicht mehr als eine Metapher und ihre Tragfähigkeit für begrenzt. In FLOYD 89 habe ich eine **Design-Sicht** der Softwareentwicklung ausgearbeitet, die das Zustandekommen von Einsichten in die Funktionalität und Nutzungsmöglichkeiten von Programmen in den Vordergrund stellt und insbesondere die Kommunikation zwischen Entwicklern und Benutzern berücksichtigt. Sie ist eng verwandt mit der **Theoriebildungs-Sicht** von Naur (NAUR 85), auf die ich mich im folgenden beziehe, weil Naur auch Implikationen für das Berufsbild von Softwareentwicklern aus dieser Metapher ableitet.

Naur vergleicht Softwareentwicklung mit Theoriebildung im Sinne von RYLE 63. Theorie bedeutet hier ein prozessual wachsendes Verständnis, das uns in die Lage versetzt, kompetent mit einer interessierenden Problemstellung umzugehen oder Fragen darüber zu beantworten. Man kann Theorien auch über Probleme der Alltagswelt bilden, wie etwa über die Einrichtung von Wohnungen etc. Davon ausgehend erarbeitet Naur eine nach meiner Auffassung sehr nützliche und ganzheitliche Metapher für die Softwareentwicklung, indem er argumentiert, daß der einzelne Programmentwickler oder die Gemeinschaft von Entwicklern und Benutzern in der Gesamtheit ihrer Tätigkeiten bei der Softwareentwicklung ein wachsendes Verständnis über den interessierenden Teilbereich der Welt und insbesondere auch über das, was ein Programm tun soll,

tun kann und wie es eingesetzt werden soll, bilden. Eine Theorie in diesem Sinne kann nicht vom Menschen losgelöst werden. Programme, Spezifikationen und Dokumentationen geben jeweils nur Anteile der Theorie wieder, während die Begründungszusammenhänge letztlich beim Menschen verbleiben.

Wenn man diese Sicht bis zu Ende denkt, dann kommt man zu einem anderen Berufsbild für Softwareingenieure. Gemeint ist ein Berufsbild, in dem Softwareingenieure nicht primär als Produzenten tätig sind, sondern eher als Konsulenten, ähnlich als wären sie Rechtsanwälte, die einen Klienten beraten. Wenn ich Klientin bei einem Rechtsanwalt bin, erwarte ich nicht in erster Linie, daß er ein Produkt abgeliefert. Er kann zwar beliebig viele Produkte abliefern, Briefe, Stellungnahmen oder ähnliches, aber primär erwarte ich von ihm, daß er mich in meiner maßgeblichen Angelegenheit so berät, daß ich sie erfolgreich bewältigen kann. Das ist die entscheidende Erwartung. In diesem Sinne müßte man vielleicht dazu übergehen zu sagen, daß die Beratung bei der Auswahl, Konzeption und Benutzung des Programmsystems ein Teil des Aufgabengebietes von Softwareentwicklern ist.

Eine ganz andere Metapher für die Softwareentwicklung liefern ABELSON, SUSSMANN 85 in der Einleitung zu ihrem Grundstudiumslehrbuch für Informatik am M.I.T.. Da sie die Programmierung mit mittelalterlichen Versuchen, Geister zu beschwören, vergleichen, will ich hier von einer **Zauberer-Metapher** sprechen. Bemerkenswert erscheint mir dabei vor allem, daß die Autoren sich auf diese Metapher überhaupt stützen und dann noch an einer Schlüsselstelle in einem für alle Studenten verbindlichen Text. Die Metapher selbst fasziniert mich. Zum einen weil sie die verschiedenen Dimensionen der Softwareentwicklung anschaulich verbindet; dabei entspricht der Programmtext der Beschwörungsformel, die Ausführung des Programms der Wirkung der Formel auf den "Geist", und der Einsatz des Programms der Tätigkeit des "Geistes" in seinem Umfeld. Zum anderen macht sie auch wichtige Aspekte des Verhältnisses zwischen Mensch und Computer deutlich: es gilt, den "Geist", dem viele überragende Fähigkeiten zuschreiben, zu beherrschen und für uns arbeiten zu lassen unter Einsatz von List und aller Kraft. Der Computer wird dabei als pseudogeistiges, aber nicht dem Menschen verwandtes Wesen charakterisiert.

Dennoch erscheint mir diese Zauberer-Metapher nicht für den uns interessierenden Bereich besonders treffend, sondern eher für den Bereich der Steuerung technischer Systeme.

Bei der Gestaltung interaktiver Systeme steht dagegen nicht der Computer als Akteur im Vordergrund, sondern die Menschen, die den Computer im Rahmen von Arbeits- und Kommunikationsprozessen verwenden. Dabei ist auch die Kommunikation von Benutzern und Entwicklern ein wichtiger Aspekt, der in einer treffenden Metapher berücksichtigt werden sollte.

Zu wünschen ist, daß sich eine Metapher finden läßt, die dieses Kommunikationsverhältnis zwischen Entwicklern und Benutzern aufgreift. Man könnte zunächst an das Telefon denken. Aber das Telefon ist natürlich ganz anders, weil wir am Telefon miteinander sprechen und weil derjenige, mit dem wir sprechen, zeitgleich verfügbar ist. Es muß eine Metapher sein, die sich auf indirekte Kommunikation bezieht, die über ein Medium gesteuert wird bzw. abläuft. Es muß ein kommunikativer Bezug zu demjenigen nachvollziehbar werden, der das Programm erstellt hat, ohne daß sich der Computer als Partner verselbständigt. Ein solches Bild erlaubt es auch, einen Bezug zur natürlichen Sprache herzustellen.

Ein einzelner wird die durch dieses Bild implizierten Aufgaben nicht bewältigen können. Man könnte aber versuchen, eine Beziehung herzustellen zwischen einem Softwareinstitut, das etwas entwickelt, und den Benutzern, indem man eine Art von Notttelefon einrichtet. Das könnte man sich überlegen, und es gibt auch schon Ansätze dazu. Man könnte das z.B. auch mit den technischen Möglichkeiten der digitalen Abspeicherung gesprochener Sprache unterstützen.

Nach der Auffassung, die die Softwareentwicklung als Theoriebildung charakterisiert und vom Softwareingenieur eine Problemlösung mit Hilfe des Computers erwartet, erscheint uns der Computer sowohl als Gegenstand als auch als Instrument der Theoriebildung. Im Zusammenhang mit der Kommunikation zwischen Entwicklern und Benutzern erscheint er als ein Medium, über das letztlich der Designer mit seinen Benutzern kommuniziert.

Es gibt den berechtigten Einwand, daß die **Medien-Metapher** für die Kommunikation zwischen Entwicklern und Benutzern unbefriedigend sei, weil man in Störfällen gar nicht daran denke, daß hier ein Entwickler etwas falsch gemacht haben könne. Das heißt, wenn ich am Computer sitze, ist für mich die Vorstellung, mit dem Entwickler zu kommunizieren, unnatürlich. Das betrifft sowohl den Normalfall als auch den Sonderfall, wenn mir das System nicht mehr *zuhanden* ist, wie man es in Anlehnung an Heidegger ausdrücken kann, was seit Winograd und Flores ein wenig in Mode ist. In dem Sonderfall also, in dem man im Sinne von Heidegger das System als *vorhandenes* überhaupt erst erfährt, ist diese Metapher besonders wenig befriedigend.

Die Frage ist, ob es nicht Maßnahmen geben kann, anhand derer deutlich wird, daß und wie unerwünschte Situationen bei der Benutzung mit den Gestaltungsentscheidungen der Systementwickler zusammenhängen. Dabei müßte man diesen Sachverhalt durch das Angebot einer von Menschen getragenen Infrastruktur, wie etwa von Beratungsdiensten, offenlegen. Bevor wir dies jedoch als den einzigen Weg ansehen, ist zu prüfen, in welchem Ausmaß wir die Argumente für ein solches Beratungsangebot in Designkriterien ummünzen können. Um dies zu leisten, müssen Entwickler die Benutzungsvorgänge antizipieren und an den Problemen der Benutzer partizipieren, wobei die Antizipation unabdingbar ist.

Man könnte nun so vorgehen, daß Entwickler mündlich ihre Intentionen, die sie bei bestimmten Entwurfsentscheidungen haben, erläutern, und daß diese Erläuterungen für Benutzer abfragbar in digitalisierter Form abgespeichert werden.

Natürlich gibt es einen Unterschied zwischen dem zeitgleich präsenten Entwickler einerseits, der unmittelbar oder technisch vermittelt auf meine Fragen antwortet, und der Sprachkonserve andererseits, die auf mögliche, antizipierte Fragen antwortet. Im zweiten Fall entsteht das Problem, daß man Situationen prädefinieren muß. Es ist wichtig, aufzuzeigen, welche technischen Möglichkeiten es gibt, um Modelle dieser Art durchzuspielen. Damit ist aber noch nicht das wesentliche Problem gelöst, welche Anfragemöglichkeiten für Benutzer gegenüber solchen Sprachkonserven vorhanden sein müssen und welche Rückfragen möglich sein sollten.

Auf der Suche nach sprachorientierten Metaphern für den Computer

Schon der Titel der Podiumsdiskussion der diesjährigen Jahrestagung – "Wie sag ich's dem Werkzeug?" – macht deutlich, daß die **Werkzeug-Metapher** für die Linguistische Datenverarbeitung unbefriedigend ist. Hier liegt etwas schief. Offenkundig ist es absurd, daß ich einem Hammer etwas sagen will. Es ist zu bezweifeln, ob das Wort "Werkzeug" noch eine sinnfällige Bezeichnung für etwas darstellen kann, mit dem wir sprachlich interagieren können; die Möglichkeit einer echten Natürlichsprachlichkeit bleibt hierbei noch dahingestellt.

Nutzen und Grenzen der Werkzeug-Metapher werden im Bereich Mensch-Computer-Interaktion ausführlich diskutiert. Speziell im Deutschen ergibt sich das zusätzliche Problem der vom englischen Sprachgebrauch abweichenden Bedeutung, nach der "Tool" bekanntlich etwas anderes heißt als "Werkzeug". Die Diskussion der Werkzeug-Metapher ist für die Linguistische Datenverarbeitung jedoch wenig aufschlußreich, da sie hier schon daran scheitert, daß sie so entsetzlich unnatürlich klingt, d.h. jeder Sinnfälligkeit entbehrt.

Im Gegensatz dazu liefert die **Medien-Metapher** fruchtbare Assoziationen.

Es ist zunächst überlegenswert, inwieweit für die Linguistische Datenverarbeitung eine **Buch-Metapher** sinnfällig sein kann, bei der der Computer als Lexikon, Handbuch, Nachschlagewerk oder ähnliches charakterisiert wird. Entscheidend ist, daß uns die Metapher aus unserem Alltagskontext heraus als natürlich nahelegt, daß beim Umgang mit Computern Sprache zur Verwendung kommt; und daß sie gleichzeitig nicht suggeriert, daß sich beim Computer Sprache im menschlichen Sinne frei entfalten kann.

Letztere Einschränkung beinhaltet die schon erwähnte Kritik an dem Anspruch, daß natürliche Sprache durch den Computer verarbeitbar sei. Dieser Anspruch ist nicht einlösbar, weil man die Sprache des Menschen nicht vom Menschen lösen und über semantische Bedeutungsaxiome ausspezifizieren kann. Vielmehr ist menschliche Sprache in menschliche Situationen eingebunden. Wann immer man die Verarbeitung natürlicher Sprache im Computer vorsieht, ist die Entfaltung dieser Situationsgebundenheit eingeschränkt.

Es gibt durchaus Gemeinsamkeiten zwischen dem Computer und Medien, die Sprache nicht nur übermitteln, sondern auch fixieren. Aus der Sicht der Sprachverarbeitung ist es instruktiv, einen Computer als Medium zu sehen und ihn zunächst mit einem Buch zu vergleichen. Allerdings kann es kein konventionelles Buch sein, und der Vergleich mit dem Lexikon ist wohl nicht ausreichend.

Die Buch-Metapher stellt uns auch Verhaltensmaximen zur Verfügung, die sich Menschen im Umgang mit gedruckten Medien über Jahrhunderte erarbeitet haben. Dazu gehört es zum Beispiel, die Verantwortlichkeit des Autors für den Inhalt explizit darzustellen und gerade auch in schwierigen Fällen – wie in der wissenschaftlichen Literatur – die Genese der Ideen im Text nachvollziehbar zu machen. Bei Büchern gibt es da auch sehr phantasievolle Beispiele. Man denke z.B. an das schon zitierte Buch MATURANA, VARELA 87. Die Autoren treten dort, insbesondere mit ihren unterschiedlichen Auffassungen, als Buchgestalten auf, die immer wieder die Ausführungen des Haupttextes kommentierend begleiten. Wenn man diese Idee auf Softwaresysteme überträgt, müßten Softwareentwickler sich selbst als Autoren in das System einbringen und ihre Rolle gegenüber den Systembenutzern explizit vertreten. Entwicklerteams könnten sich verhalten wie Autorenkollektive.

Gewonnen ist damit ein konkreter Vorschlag zu der Frage, in welcher Art und Weise man sich als Entwickler in einem System vergegenwärtigen kann. Der Entwickler, der sich an der Rolle des Autors

orientiert, könnte auch die Autonomie für sich in Anspruch nehmen, die einem Autor selbstverständlich zugebilligt wird.

Weiterhin lenkt die vorläufige Gleichsetzung von Druckmedien und Computern den Blick auf den historischen Prozeß, in dem Menschen seinerzeit mühsam gelernt haben, die rasant um sich greifende neue Technik des Buchdrucks sinnvoll zu interpretieren und sozial wie sprachlich zu integrieren. Es wird dann verständlich, daß auch die Auseinandersetzung mit dem neuen Medium Computer Zeit braucht und nicht auf Anhieb zu Vorstellungen führt, die sich auf Dauer bewähren.

Nun legen uns Autoren, die über die medialen Eigenschaften von Computern und Druckmedien nachgedacht haben (KAY, GOLDBERG 77; YANKELOVICH et al. 85) eher nahe, Computer im Vergleich zu Büchern als erweiterte, allgemeinere Medien aufzufassen, die interaktiv, dynamisch und fähig sind, die Funktionen von Büchern, Telefonen und audiovisuellen Medien miteinander zu verbinden, und zusätzlich noch programmiert werden können. Die Konsequenz daraus ist einfach: Wer aus begreiflichen Gründen die genannte Lexikon-Metapher unbefriedigend findet als Charakterisierung dessen, was natürlichsprachliche Systeme leisten sollten, ist aufgefordert, nicht nur die sehr differenzierte Buchkultur auf geeignete Leitbilder zu befragen – ihm oder ihr wird zusätzlich nahegelegt, die Kultur der interaktiven und audiovisuellen Medien auf übertragbare Gestaltungsvorbilder durchzugehen.

Danksagung

Ich danke Thomas Herrmann und Brigitte Endres-Niggemeyer für ihre weitreichende Hilfestellung bei der Ausarbeitung dieses Manuskriptes ausgehend von einer Tonbandaufzeichnung des Vortrags. Sie haben es insbesondere zuwege gebracht, die Diskussionsbeiträge in den Vortrag zu integrieren und damit das Papier wesentlich zu bereichern.

Literatur

Abelson, H.; Sussmann, G.J.; Sussmann, J.: Structure and Interpretation of Computer Programs. 2nd Ed., Cambridge, Mass.: MIT Press 1985.

Dreyfus, H.L.: Die Grenzen künstlicher Intelligenz – Was Computer nicht können. Königstein/Ts.: Athenäum 1985.

Floyd, Ch.: Softwareentwicklung als Realitätskonstruktion. Erscheint im Tagungsband der Fachtagung "Software-Entwicklung – Konzepte, Erfahrungen, Perspektiven" der Gesellschaft für Informatik in Marburg, 21.-23.06.1989.

Kay, A.; Goldberg, A.: Personal Dynamic Media. Computer 10 (1977) 3, 31-41.

Maaß, S.: Mensch-Rechner-Kommunikation. Herkunft und Chancen eines neuen Paradigmas. Univ. Hamburg, FB Informatik, Ber. 104, 1984.

Maturana, H.; Varela, F.: Der Baum der Erkenntnis. Bern: Scherz 1987.

Naur, P.: Programming as Theory Building. Microprocessing and Microprogramming 15 (1985) 253-261.

Ryle, G.: The Concept of Mind. Harmondsworth, England: Penguin 1963.

Winograd, T.; Flores, F.: Understanding Computers and Cognition. 2nd Ed. Reading, Mass.: Addison-Wesley 1987.

Yankelovich, N.; Meyrowitz, N.; van Dam, D.: Reading and Writing the Electronic Book. Computer 18 (1985) 10, 15-29.