

Andreas Baumert

Sprechmaschinen

Anmerkungen zum Stand der Kunst

Es ist wirklich kein Problem. Wenn ich den Schwierigkeitsgrad entsprechend einstelle (low), habe ich gute Chancen, heil von der Piste zu kommen. Schließlich kann ich den Fuß vom Gas nehmen, kann den Wagen langsam durch Kurven lenken und meinen Kontrahenten ausweichen. Ich werde das Rennen zwar nicht gewinnen, sammle aber genügend Punkte und werde leidlich abschneiden. Sicher, meine Fahrweise ist übervorsichtig; es bringt einen aber ganz ordentlich nach vorne, hat man nur wenige tödliche Unfälle durchleiden müssen. Mein Wagen ist ein IRGENDWAS-Modell, tankt Markstücke und steht in einem Automatensalon. Irgendeine Kombination elektronischer Bauelemente hat mir die Illusion eines Rennens auf den Bildschirm gezaubert; aber wie! Ich konsumiere nicht die Illusion, ich kann mit ihr interagieren, kann sie in gewissen Grenzen gestalten.

Kapitäne zur See und in der Luft werden auf Simulatoren trainiert; ein solcher Apparat ist beispielsweise SUSAN an der Hamburger Fachhochschule (1). SUSAN ist das Modell einer Schiffsbrücke, das System besteht aus einer hydraulisch gesteuerten 'Brücke', die mit allen nötigen Instrumenten, Radar, Echolot usw. ausgestattet ist. Durch die Fenster schaut der angehende Kapitän auf eine Projektionsfläche (248 horizontal, 16 vertikal). Elf Videoprojektoren erzeugen die Illusion eines fahrenden Schiffes, die Hydraulik simuliert die Schiffsbewegung, die Instrumente reagieren nicht anders als auf einem wirklichen Dampfer. Die Anlage wird von einem Zentralrechner (DEC) geleitet, der fünf Steuerrechner (KRUPP) zuständig für Instrumente, Hydraulik und Bildprojektion etc., antreibt.

Beide Apparate sind computergesteuerte Simulatoren und können dennoch nicht sinnvoll miteinander verglichen werden. Das Autorennen kann durch eine simple Parametermodifikation (GAS, BREMSE, HIGH/LOW) als quasi-individuelles Ereignis gesteuert werden. Die 'Schiffsbrücke' unterscheidet sich erheblich von

diesem Modell, sie verfügt, vorsichtig ausgedrückt, über eine Art von Wissen von der Welt, in der wir leben. Es wird keinen Rennwagen geben, der die katastrophischen Eigenschaften meines Automaten aufweist, es gibt aber die Hafeneinfahrten, die SUSAN simuliert; die Kais, die an einem echten Schiff vorbeigleiten, werden strukturell gleich in SUSAN projiziert, die Maschine reagiert auf Seegang und Wetterverhältnisse in einer wirklichen Schiffen vergleichbaren Weise.

Man könnte sich nun vorstellen, daß SUSANs Brücke eher einem Motorboot gleiche, ferner gestalten wir das Programm des Rennwagens so um, daß Katastrophen im Rennverlauf nicht mehr die Regel sind. Für Laien im Straßenverkehr und in der Schifffahrt wären beide Maschinen recht gleichwertig. Wer den Rotterdamer Hafen nicht kennt, weiß nichts über das in SUSAN enthaltene 'Wissen' über die Gegebenheiten dieses Hafens, er könnte ihn für eine der Rennpiste ähnliche Illusion halten.

1. Natürlichsprachliche Kommunikation und künstliches Programm

Wir werden nun eine Parallele zur maschinellen Verarbeitung natürlicher Sprache feststellen: Simple und theoretisch anspruchslose Verfahren sind mitunter publikumswirksamer als diejenigen Programme, die dem Stand der Kunst in Künstlicher Intelligenz (KI), Computerlinguistik und z.T. der kognitiven Psychologie entsprechen. Zusätzlich wird ein Urteil, was natürlichsprachliche Systeme heute zu leisten vermögen, durch Sensationsmeldungen in den Medien erschwert. Denkende Computer, maschinelle Übersetzung am Telefon und dergleichen mehr sind pure Science Fiction. Man verschätzt sich leicht – der Bluff und das entwickelte System können auf der gleichen Maschine laufen.

Nehmen wir an, Sie sind der einzige Benutzer eines Großrechners, auf Ihrer Maschine läuft das System TEXT (2) (Kathleen McKeown). Diesem Programm stellen Sie die recht einfache Frage

What is the difference between a destroyer and a bomb?

TEXT antwortet:

A DESTROYER IS A SURFACE SHIP WITH A DRAFT BETWEEN 15 AND 222. A SHIP IS A VEHICLE. A BOMB IS A FREE FALLING PROJECTILE THAT HAS A SURFACE TARGET LOCATION. A FREE FALLING PROJECTILE IS A LETHAL DESTRUCTIVE DEVICE. THE BOMB AND THE DESTROYER, THEREFORE, ARE VERY DIFFERENT KINDS OF ENTITIES.

Die Antwort ist sicher nicht falsch; was Sie vermutlich außer dem Thema beunruhigend finden, ist die Zeit, die Sie am Bildschirm warten müssen, bis TEXT Ihre Anfrage beantwortet hat: 5 Stunden und elf Minuten!

Sie laden nun ein anderes Programm, ELIZA (3), in den gleichen Rechner und bekommen die Antwort praktisch sofort nach dem Drücken der Sendetaste (ELIZA wurde von Joseph Weizenbaum in MADSLIP geschrieben, es liegt heute als Dokumentation und zum Spielen in praktisch allen LISP-Versionen vor).

Men are all alike.

IN WHAT WAY

They're always bugging us about something or other.

CAN YOU THINK OF A SPECIFIC EXAMPLE

Well, my boyfriend made me come here.

YOUR BOYFRIEND MADE YOU COME HERE

He says I'm depressed much of the time.

I AM SORRY TO HEAR YOU ARE DEPRESSED

It's true. I am unhappy.

DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY

I need some help, that much seems certain.

WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP

Perhaps I could learn to get along with my mother.

TELL ME MORE ABOUT YOUR FAMILY

My mother takes care of me.

WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU

My father.

YOUR FATHER

You are like my father in some ways.

WHAT RESEMBLANCE DO YOU SEE

You are not very aggressive but I think you don't want me to notice that.

WHAT MAKES YOU THINK I AM NOT VERY AGGRESSIVE

You don't argue with me.

WHY DO YOU THINK I DON'T ARGUE WITH YOU

usw.

Dieser Dialog wurde vor nunmehr zwanzig Jahren erstmals veröffentlicht. Es ist ein therapeutischer Dialog, wie er im Rahmen der 'Client centered therapy' (Rogers) favorisiert wird. ELIZA erzeugt Sätze, die sich in verblüffender Weise irgendwie zu den Äußerungen des menschlichen Gesprächspartners zu verhalten scheinen. Es erlangte eine traurige Berühmtheit, als Joseph Weizenbaum auf einige beängstigende Folgen seiner Arbeit hinwies:

Erstens meinten einige Psychiater, dieses Programm könnte weiterentwickelt und in der klinischen Praxis eingesetzt werden,

zweitens vergessen Benutzer schnell, daß sie mit einem Programm 'sprechen', einige vertrauen sich ihm tatsächlich an und schließlich wurde ELIZA anfangs zu Unrecht als ein Programm aufgefaßt, das menschliche Sprache versteht.

Wie wir sehen werden, hat ELIZA nicht die Spur eines wirklichen Sprachverständnisses. Es besteht aus zwei Teilen, dem eigentlichen Programm und einer Datenliste, dem Script. Die Datenliste ist einer bestimmten Domäne zugeordnet, in unserem Beispiel oben ist es eben das Script DOCTOR.

Der Eingabesatz wird nach Schlüsselwörtern, die hierarchisch geordnet sind, durchsucht. Jedem Schlüsselwort ist eine endliche Menge Zerlegungs- und Zusammensetzungsregeln zugeordnet (im folgenden 'D' (decomposition-rule) und 'R' (reassembly-rule), 'K' für keyword). Man

kann dieses Regelinventar in einer Listenstruktur darstellen:

(K

((D 1) (R 1 1) (R 1 2) ... (R 1 n))

((D 2) (R 2 1) (R 2 2) ... (R 2 n))

...

((D n) (R n 1) (R n 2) ... (R n n)))

Ein Beispiel soll das Vorgehen verdeutlichen:

You are very helpful.

„You“ ist ein Schlüsselwort und wird intern in „I“ umgeändert. ELIZA erkennt das Schlüsselwort und verändert den Satz in „I are very helpful“. Das Programm sucht nun die Liste durch, bis eine Zerlegungsregel gefunden ist, die mit „I are very helpful“ in Übereinstimmung zu bringen ist. Die erste Regel ist
(0 I REMIND YOU OF 0)

'0' ist Stellvertreter für eine beliebige Anzahl Wörter. Offensichtlich trifft diese Regel nicht zu, ELIZA geht zur nächsten Liste und findet
(0 I ARE 0).

Diese Regel unifiziert, das erste Element unseres Satzes ist eine leere Menge

von Wörtern (0), das letzte Element sind die Ausdrücke „very helpful“ (0) und die von der Regel verlangten Elemente „I“ und „are“ sind an der richtigen Stelle des Satzes. 1.: leer 2.: I 3.: are 4.: very helpful
(0 I ARE 0)

Die erste Zusammensetzungsregel, die auf (0 I ARE 0) folgt, ist (WHAT MAKES YOU THINK I AM 4),
(0 I ARE 0)
(WHAT MAKES YOU THINK I AM 4)
(DOES IT PLEASE YOU TO BELIEVE I AM 4)
(DO YOU SOMETIMES WISH YOU WERE 4)
(PERHAPS YOU WOULD LIKE TO BE 4))

Das vierte Element unseres Satzes wird nun eingefügt, und ELIZA antwortet:
WHAT MAKES YOU THINK I AM VERY HELPFUL

(WHAT MAKES YOU THINK I AM 4) wird schließlich mit einem Index versehen, so daß das Programm, wenn es wieder an der Stelle (0 I ARE 0) unifiziert, die nächste Zusammensetzungsregel anwendet; ist das Ende der Liste erreicht, wird auf das erste Listenelement zugegriffen.

ELIZA hat tatsächlich kein Wissen über die Welt, mit dieser Methode könnte folgender Dialog geführt werden:

*Ich habe mir eben die Pulsadern aufgeschnitten.
WAS BEDEUTET ES FÜR DICH DIE PULSADERN AUFGESCHNITTEN ZU HABEN*

Ich werde sterben.

ERZÄHL MIR MEHR ÜBER DEINE ZUKUNFT

Ein solches Programm hätte allerdings gewaltige Schwierigkeiten mit dem deutschen Flexionssystem, ELIZA verfügt schließlich auch über kein linguistisches Wissen. Regeln über Artikel, Nomina, das Passiv – alle aus linguistischen Theorien deduzierbaren Regeln – fehlen völlig. Dieses Verfahren wird in der Linguistik als pattern matching bezeichnet; geht der Abgleich fehl, antwortet das System mit *TELL ME MORE, PLEASE GO ON* usw. Die gespeicherten Daten sind linguistisch nicht näher spezifiziert; wenngleich auch die Dialogfä-

higkeit mitunter täuschend echt scheint, handelt es sich letztlich um einen Bluff.

In einem strengeren Sinn könnte man sagen, ELIZA simuliert nicht tatsächlich den Therapeuten, es ist bestenfalls die Simulation eines natürlichsprachlichen Systems, das die therapeutische Gesprächslenkung simuliert; man könnte es eine Metasimulation nennen, es als einen Apparat betrachten, dessen Output dem eines therapeutischen Simulationssystems manchmal gleicht oder ähnelt. Natürlich müßte ein echtes Simulationssystem auf den Satz „Ich werde sterben“ anders reagieren als auf „Ich werde heiraten“.

Diese verschiedenen Ebenen simulierender Programme sind in der KI-Forschung durchaus nicht unüblich, man weist aber gewöhnlich nicht explizit darauf hin. Ein anderes Programm, SHRDLU (1971, Terry Winograd) (4), ist vermutlich ebenso berühmt geworden wie ELIZA. SHRDLU bewegt Klötzchen, wie sie von Kindern zum Spielen benutzt werden. Fordert man es auf, einen Klotz, auf dem ein anderer liegt, zu entfernen, wird es zunächst den oberen beiseite legen und anschließend seine Aufgabe erfüllen. Es kann dabei in 'natürlichem' Englisch Rechenhaft über seine Handlungen abgeben. Doch sind diese Klötzchen nicht real, sie sind, ebenso wie der Roboterarm, nur im Programm repräsentierte Manipulationsgegenstände. SHRDLU simuliert nicht eigentlich einen Konnex zwischen Sprachverstehen, Handeln und Eingriff in die reale Welt bei Kleinkindern, es simuliert, indem es nicht wirklich handelt, eben diese Simulation, ist eine Simulation anderen Typs.

SHRDLU unterscheidet von ELIZA vor allem zweierlei. Erstens hat dieses Programm Weltwissen, es ist eben nur die etwas dürftige Klötzchen-Welt. Zweitens enthält es 'linguistisches Wissen', kann z.B. ein transitives Verb von einem intransitiven (die Repräsentation des Verbs vorausgesetzt) unterscheiden. Es war in der kurzen Geschichte der KI-Forschung vor allem deswegen von Bedeutung, weil Winograd in seiner Arbeit auf den Zusammenhang zwischen maschineller Sprachverar-

beitung und Weltwissen aufmerksam gemacht hatte.

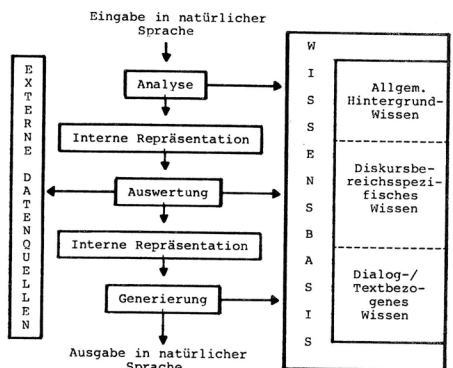
2. Sprachanalyse und Sprachgenerierung

Heutige KI-Programme sind wesentlich anspruchsvoller als z.B. ELIZA. Auch von Laborprojekten wird zumindest gefordert, daß sie über linguistisches Wissen, Weltwissen und eine reale Handlungskomponente verfügen.

Ein natürlichsprachliches System soll irgendetwas leisten. Dazu soll es entweder Sätze der Alltagssprache verstehen oder generieren; einige Programme schaffen beides.

Die Graphik (5) zeigt den Standardaufbau eines solchen Systems.

Mit ihm kann man nicht wirklich 'sprechen', der Dialog



wird über eine Tastatur, einen Bildschirm, u.U. einem Drucker und/oder einem Zeichengerät geführt. Vorläufig kann kein System gesprochene Sprache verstehen, obwohl es einige Versuche mit Sprachsynthesizern gibt, die uns hier aber nicht interessieren werden.

Schon der erste Schritt, die Analyse des Eingabesatzes, erfordert einen immensen Aufwand. In der Anfangsphase hatte man Formalismen entwickelt, die mit syntaktisch wohlgeformten Sätzen leidlich umgehen konnten.

Hat der Kunde sein Konto bei uns überzogen? (Satz 1)

JA.

Um wieviel? (Satz 2)

Ein System, das Ihren Deutschlehrer imitiert und Sie auffordert *SPRICH IN GANZEN SÄTZEN* wäre eine frustrierende Angelegenheit. Es muß also Ellipsen verstehen können, ferner benötigt es ein dialogbezogenes Wissen. Es ist eben nur für einen Menschen selbstverständlich, daß Satz 2 sich auf den Kunden aus Satz 1 bezieht. Das System muß Anaphern erkennen, es muß auch ein Wissen über den Diskursbereich haben (Satz 1: „bei uns“). Man darf auch erwarten, daß es in angemessenen Grenzen mit indirekten Sprechakten umgehen kann:

Kannst du mir sagen, wann die letzte Einzahlung auf sein Konto verbucht wurde?

JA.

Diese Antwort wäre im Sinne einer kooperativen Kommunikation völlig entgleist und entspricht 1986 sicher nicht mehr dem Stand der Kunst. Frühe Versuche der Syntaxanalyse mit wenigen semantischen Markierungen sind längst überholt. Heute wird bereits in der Phase der Satzanalyse auf eine Wissensbasis zurückgegriffen, wird ein Fokus gerichtet ('Jetzt sprechen wir über Schulzens Konto, speziell über den DISPO-Kredit') und werden auch Referenzbeziehungen festgelegt ('er' ist hier der Kredit, nicht Schulz) usw. Das Ergebnis der Analyse wird in einer internen Repräsentation, einem formalen Ausdruck, der nach den Spielregeln des Systems konstruiert ist, festgehalten.

Die Auswertung greift dann wieder auf eine Wissensbasis oder auf externe Datenquellen zu. Der eigentliche Arbeitsbereich natürlichsprachlicher Systeme liegt in der Datenbankabfrage, weniger auf dem Gebiet der Expertensysteme. Datenbanken sind zumeist Dateien, die mit einem eigenen Instrumentarium an Hilfsprogrammen und vor allem einer speziellen Sprache, einer Art Programmiersprache, ausgestattet sind. Diese Sprache ist für den Benutzer einer Datenbank ein äußerst lästiges Hindernis. Hier haben Programme, die den Zugriff

in einer 'natürlichen' Sprache gestatten, wohl vorerst den wichtigsten Anwendungsbereich.

Expertensysteme sind als Ergänzung menschlicher Fachleute konzipiert, in ihnen ist ein Wissensausschnitt in Form von Sätzen und WENN-DANN-Regeln der Prädikatenlogik erster Stufe modelliert. Diese Programme sind übrigens auch unter den KI-Enthusiasten nicht unumstritten. Natürlich können Sie damit die defekte Zündkerze in ihrem Wagen isolieren, vermutlich hätten Sie aber den Übeltäter ohne WENN-DANN-Regeln und mit weniger Zeitaufwand auch ausfindig gemacht. WENN Ihr Arzt das System für seine Diagnose benötigt, DANN sollten Sie besser den Arzt wechseln, vorerst jedenfalls.

Die Phase der Sprachgenerierung ist wohl ebenso heikel wie die Analyse. Welcher Abteilung wir den größeren Schwierigkeitsgrad attestieren wollen, ist hier unerheblich. Man hatte die Generierung lange Zeit unterschätzt, weil die Ergebnisse einer Datenbankabfrage ohnehin meist in Tabellenform, als Histogramme oder in irgendeiner Graphik vorgelegt werden. Zudem ist der Benutzer Fachmann und kann auch Antworten verstehen, die ungrammatisch sind. *KONTO (100102), SCHULZ = ZERO* ist dem Bankkaufmann durchaus verständlich, Schulz übrigens auch. Außerdem konnte die Maschinenprosa durchaus mit Lückentexten des ELIZA-Typus erzeugt werden.

Satzgenerierung, die nicht auf einem Bluff beruht, muß Ellipsen bilden können, soll selbstverständlich auch Anaphern produzieren und muß vor allem genau die Antwort erzeugen, die im Rahmen einer kommunikativen Kooperation zulässig ist. Das System soll nicht geschwätzig werden, soll nicht überbeantworten.

Wann geht der nächste Zug nach München?

1) *IN ZEHN SEKUNDEN*

2) *DER NÄCHSTE ZUG IST DER ICHER-RENCHIEMSEE. ER FÄHRT 17.13 UHR VON GLEIS 3 UND FÜHRTE EINEN SPEISEWAGEN, ZWEI GROSSRAUMWAGEN... usw.*

3) *17.13 UHR, GLEIS 3.*

4) *WENN SIE SICH BEEILEN, SCHAFEN SIE DEN 17.13, GLEIS 3. SIE KÖNNEN SICH ABER AUCH ETWAS ZEIT LASSEN UND DEN 17.25 GLEIS 7 NEHMEN.*

Wollen wir eine tatsächliche kooperative Antwort auswählen, sind die ersten beiden sicher keine geeigneten Kandidaten; auch die dritte Antwort ist mangelhaft, wenn beispielsweise die Zeit nicht ausreichen wird, um auf den Bahnsteig zu kommen. Antwort 4 könnte angemessen sein, sie verlangt dem System aber ein gehöriges Diskurs- und Weltwissen ab. Es muß 'wissen', daß Sie nicht an dem nächsten Zug interessiert sind, sondern an dem nächsten für Sie erreichbaren Zug. Es muß 'wissen', daß die menschlichen Extremitäten nur bis zu einer gewissen Geschwindigkeit die Fortbewegung unterstützen, daß hohe Geschwindigkeiten (rennen) mitunter als unangenehm empfunden werden, Menschen normalerweise einen leidlich bequemen Reiseantritt der Hektik vorziehen usw.

Diese kurze Reise in die Welt sprachverarbeitender Systeme könnte Anlaß zu einem Mißverständnis sein. Man könnte nämlich meinen, es gäbe solche Systeme bereits. Das ist nur bedingt richtig. Es gibt sicher einige Programme für die Datenbankabfrage, einige Expertensysteme, die schon heute auf dem Markt angeboten werden. Deren sprachliche Fähigkeiten sind aber stark eingeschränkt. Die interessanteren Systeme sind Laborprojekte, denen aber auch unterstellt werden darf, daß sie kaum mehr leisten als in wissenschaftlichen Publikationen, deren Gegenstand sie sind, von ihnen behauptet wird. Wie Katharina Morik, eine der westdeutschen KI-Forscherinnen, unlängst eingestand (6), ist der Satz „mein System bearbeitet das linguistische Phänomen xyz“ zu lesen als „es gibt wenigstens eine sprachliche Form, die xyz enthält und die mein System bearbeiten kann“. Dieses Verfahren ist durchaus nicht unredlich. Moderne linguistische Theorien arbeiten oft nicht anders; Theoreme werden an erfundenen Beispielsätzen diskutiert. Diese Beispiele sind, ob sie nun als Abweichung oder Bestätigung markiert sind,

sämtlich unter dem Gesichtspunkt theoriekonformer Affirmation kreiert. Der wesentliche Unterschied zwischen genuin linguistischen Theorien und denen der Künstlichen Intelligenz scheint in der Möglichkeit der Verifikation begründet. Wenn man sich schon auf die vermeintlich sicheren Geleise der formalen Theorien begibt, wird der Computer zu einem tückischen Verifikationsinstrument. Läßt man ein Programm laufen, das natürliche Sprache simulieren soll, liegt es nicht unbedingt am Programmierer, wenn die Ergebnisse düftig geraten: es liegt an der linguistischen Theorie selbst. Unbestreitbar könnten Methoden und Probleme der Künstlichen Intelligenz 'traditionelle' Wissenschaften befruchten – gäbe es nicht einen marktorientierten Erfolgszwang, der jedes erkenntnistheoretische Interesse schon kurzfristig zu dominieren droht.

3. Das Problem der Intelligenz

Kein Zweifel – es gibt Programme, die etwas sinnvolles tun, fordert man sie in natürlicher Sprache dazu auf, und einige von ihnen antworten gar in einer verständlichen Sprache. Darüber aber steht das Turing-Problem „Kann das Ding denken, tut es irgendetwas, wozu es nicht vorher programmiert worden ist, ist es intelligent?“

Einige KI-Forscher verweisen spöttisch auf die 'Fehlleistungen' längst kannibalisierter Disziplinen, der Linguistik, Psychologie und – wenn nötig: der Philosophie.

Man versucht letztere unter bestimmten Fragestellungen auszuschlachten, etwa der Frage 'gibt es eine viertelwegs unumstrittene Definition für Intelligenz', 'gibt die Wortfeldtheorie irgendetwas her, das meine Analyseprozeduren beschleunigen könnte' usw. Geisteswissenschaftliche Theoriebildung wird so leicht ihres diskursiven Charakters entkleidet und auf den – womöglich formalen – Aspekt operabler Regeldesklipion reduziert. Unter diesem Gesichtswinkel ist der geistes- und sozialwissenschaftliche Theorienstreit zutiefst ärgerlich. Es ist eben eine Frage des übernommenen Paradigmas, was intelligentes Verhalten ausmacht. Man könnte ohne nennenswerte Probleme ein Programm schreiben, das den IQ-Test nach Eysenckschem Muster glänzend besteht. Ist nun das Programm intelligent, obgleich es sonst nichts zu leisten vermag, oder ist nur der Test blöd?

Stellen wir eine andere Frage: Hat SUSAN Wissen? Die Antwort scheint zunächst klar. Es hat ebensowenig Wissen wie ein Konversationslexikon. Erst der Gebrauch eines Buches oder des Simulators läßt Wissen virulent werden. Verändern wir das System nun so, daß es nicht mehr physisch reagiert, sondern einen Dialog führt. *Was sollen wir machen, wir haben Windstärke 10, die Ladung ist verrutscht, und das Schiff droht zu kentern?*

GEBEN SIE NOTSIGNAL. STEIGEN SIE IN DIE RETTUNGSBOOTE (1) DUMUSST JETZT VOR ALLEM ETWAS GEGEN DIE PANIK UNTERNEHMEN, DIE VIELLEICHT BEI DEINEN LEU-

TEN AUFKOMMT... (2) TSCHÜSS. WAR NETT DICH KENNENGELERNT ZU HABEN (3)

Zur Bewertung maschineller Intelligenz postuliert Roger Schank, ein führender KI-Forscher, die Hierarchie des Verstehens (7). Sie reicht vom bloßen Erfassen bis zur völligen Einfühlung. Die Zwischenstufe, das kognitive Verständnis, wird in diesem Modell als höchstmöglicher Grad der Computerintelligenz angenommen. Die Ebene des Erfassens ist sicher unstrittig. In dem konstruierten Dialog oben setzt Antwort 1 voraus, daß die Situation erfaßt ist. Die weiteren Marken sind allerdings völlig willkürlich gesetzt. Sie dienen eher dazu, einen Verstehensbegriff zu selektieren, der unauflösbar an die Verknüpfung intelligenzproduzierender Hardware mit Prototypen gebunden ist. Völlige Einfühlung bedarf der Eigenerfahrung, einer Maschinen unzugänglichen Welt der Gefühle und Launen. Die Trennung völliger Einfühlung von kognitivem Verständnis ist letztlich ein Trick, der die eigentliche Frage umgeht. Er sagt nur 'Programme können intelligent sein, und Menschen sind anders (mehr?) intelligent'. Antwort 3 oben wäre kein Beispiels für diesen Typ Intelligenz; sollte es vielleicht ein herzerreißender Hilfeschrei sein?

Terry Winograd geht diese Frage mittlerweile völlig anders an. Intelligentes Verhalten setzt die Interaktion autonomer Systeme in einem strukturell prädisponierten Rahmen, der in der biologischen Evolution fundiert ist, voraus. „Es ist sehr unwahrscheinlich, daß ein von uns gebautes System eine evolutionäre Entwicklung (Lernen) durchmachen könnte, die es auch nur in die Nähe der Intelligenz eines Wurmes zu bringen vermöchte, geschweige denn einer Person.“ (8) Nach dieser Vorstellung fehlten dem oben verballhornten SUSAN selbst die elementaren Voraussetzungen: Es weiß nichts. In ihm ist aber das Wissen des Programmierenden gespeichert; es ist nicht autonom und vor allem kann es keine handlungsrelevanten Verpflichtungen eingehen. Von dieser Unfähigkeit zur kommunikativen Verpflichtung lebt eben gerade

die Idee des Roboters für Senioren. Unfreiwillig bestätigt Pamela McCorduck (9), die Propagandistin dieser albernsten Idee, Winograds Vorbehalte:

The geriatric robot:

„Er treibt sich nicht bei ihnen herum, weil er ihr Geld zu erben hofft – natürlich gibt er ihnen auch nichts ins Essen, um das Unausweichliche zu beschleunigen. Er treibt sich auch nicht bei ihnen herum, weil er woanders keine Arbeit findet. Er ist einfach da, weil er ihnen gehört. Er wird sie nicht nur baden und füttern, wird sie in die Sonne rollen, wenn sie etwas frische Luft schnappen und mal etwas anderes sehen möchten – neben diesen selbstverständlichen Angelegenheiten ist sein größter Vorzug, daß er *zuhören* kann. 'Erzähl mir noch einmal', wird er sagen, 'wie gräßlich/wunderbar deine Kinder zu dir sind. Erzähl mir nochmal die Geschichte von diesem irren Ding 1963, erzähl mir nochmal...'. Und genauso meint er es. Er wird ebensowenig müde, ihre Geschichten zu hören, wie sie des Erzählens müde werden. Ihre Lieblingsgeschichten kennt er, es sind auch seine. Glauben sie bloß nicht, ein menschlicher Fürsorger könnte das auch. Den werden sie langweilen, er wird habgierig und möchte auch etwas Abwechslung – Menschen sind eben so.“

Unsere Fragen nach 'Wissen' und 'Intelligenz' bleiben unbeantwortet; und der Linguist, der sie stellt, scheint ebenso anachronistisch wie seine von Programmen deklassierten Theoreme. Das Technetronic Age löst solche Probleme auf vertraut hemdsärmelige Weise:

„Es ist ziemlich sonderbar. Ich bin mir nicht sicher, daß ich sagen könnte, was ein 'intelligentes' Programm ausmacht; aber ich weiß, daß bei der Verfolgung dieses schlecht definierten Zieles gute Programmiertechniken und Benutzerschnittstellen entstanden sind.“

Das sagt nicht irgendein subalternen Programmierer, so denkt Guy Steele (10), der als erster Common LISP, den 'Standard' für eine entwickelte Programmiersprache der Künstlichen Intelligenz, festgeschrieben hat.

Ich sollte ein paar Markstücke einwechseln und das Rennen wiederaufnehmen...

- (1) Winkler, Nicole: Land in Sicht? Programmierte Täuschung. In: P.M. Computerheft 2, 1986, S. 14-18
- (2) McKeown, Kathleen R.: Discourse Strategies for Generating Natural-Language Text. In: Artificial Intelligence, Amsterdam 27 (1985), S. 1-41
- (3) vgl. Weizenbaum, Joseph: Eliza – A Computer Program for the Study of Natural Language Communication between Man and Machine. In: Communications of the ACM 9 (1966), S. 36-45
- (4) vgl. Winograd, Terry: Ein prozedurales Modell des Sprachverstehens. In: Eisenberg, Peter (Hrsg.) 'Semantik und künstliche Intelligenz. Beiträge zur automatischen Sprachbearbeitung II', Berlin, New York 1977, S. 142-179
- (5) vgl. Wahlster, Wolfgang: Natürlichsprachliche Systeme. Eine Einführung in die sprachorientierte KI-Forschung. In: Bibel, Wolfgang; Sieckmann, Jörg H. 'Künstliche Intelligenz' (Informatik Fachberichte 59), Berlin usw. 1982, S. 222
- (6) In ihrem Kurs auf der SiC-Sommerschule '86 in München
- (7) Schank, Roger C. mit Childers, Peter G.: Die Zukunft der Künstlichen Intelligenz. Chancen und Risiken. Köln 1986
- (8) Winograd, Terry; Flores, Fernando: Understanding Computers and Cognition. A new Foundation for Design. Norwood NJ 1986, S. 103
- (9) Feigenbaum, Edward A.; McCorduck, Pamela: The Fifth Generation. Artificial Intelligence and Japans Computer Challenge to the World. New York 1984, S. 100
- (10) LaGrow, Craig: Guy Steele, Common LISP, and the Connection Machine. In: Computer Language, San Francisco 3: 8 (1986), S. 81.