Das kognitive Schlachtfeld

Reinhard Keil-Slawik, College Park/USA

Cogito Ergo Bumm
TAZ

Erst wenn der Mensch gelernt hat zu rechnen, kann er auch eine Maschine bauen, die für ihn rechnet. Was Software leistet, hängt im wesentlichen davon ab, was die Entwickler verstehen und konstruktiv umsetzen können. Dies gilt auch für die Künstliche Intelligenz. Sowohl die Besonderheiten der Softwareherstellung als auch die Erfahrungen beim Einsatz großer Softwaresysteme speziell im militärischen Bereich zeigen deutlich, daß die Zuverlässigkeit des Gesamtsystems durch den Menschen und nicht durch die Maschine sichergestellt wird. Maschinen können nicht denken – sie sind bestenfalls berechnend. Aber ein wesentliches Moment von Kriegen ist, daß sie unberechenbar sind.

Der Begriff *Künstliche Intelligenz* verkörpert eine Zauberformel, die nicht nur helfen soll, den Menschen zu ersetzen, sondern auch, ihn effektiver zu vernichten. Krieg, Computer und Kognition bilden ein enges Beziehungsgeflecht mit einer unheilvollen neuen Qualität: *Krieg als kognitives Ereignis;* eine gefährliche Reduktion, die durch das *elektronische Schlachtfeld* technisch machbar scheint.

Kognition leitet sich vom lateinischen *cognitio,* Bekanntschaft, ab und heißt Erkenntnis. Da Erkenntnisvorgänge als Informationsverarbeitungsprozesse charakterisiert werden können, glaubt man, mit dem Computer ein Instrument in der

Dr. **Reinhard Keil-Slawik**, geb. 1953, Professor am Computer Science Department der University of Maryland. Berufsausbildung als Elektromechaniker, Studium: Elektrotechnik an der GHS Siegen/Abt. Gummersbach und nach Abschluß Informatik an der TU Berlin. Von 1979 bis 1984 wiss. Assistent in der Forschungsgruppe Softwaretechnik TU Berlin. 1981 Promotion. 1985–1990 Hochschulassistent am Institut für Angewandte Informatik der TU Berlin. Mitbegründer des Wissenschaftsladens Berlin (WILAB) e.V. und des Forum Informatiker für Frieden und gesellschaftliche Verantwortung (FIFF) e.V. Veröffentlichungen u.a. zu den Themen Computer in Alternativprojekten, menschliche und künstliche Intelligenz, Informatik und Militär.

Dr. Reinhard Keil-Slawik, Computer Science Department, University of Maryland, College Park, MD 20742, USA

Hand zu haben, das es gestattet, solche Prozesse abzubilden und zu simulieren. Nun kann Intelligenz wissenschaftlich fundiert behandelt werden. Der Ausdruck Künstliche Intelligenz verkörpert das dabei zugrunde gelegte Paradigma: der Computer als Metapher für menschliche Informationsverarbeitung.

Mit dem Aufschwung des Computers nach dem Zweiten Weltkrieg beginnt also auch für die kognitive Psychologie, die sich mit der Erforschung menschlichen Erkennens und Denkens befaßt, ein neuer Boom; und für beide setzt die militärtechnologische Entwicklung in Forschung und Anwendung wesentliche Rahmenbedingungen. Man kann schon von einem symbiotischen „Dreiecksverhältnis" sprechen.[1] Doch diese „Beziehung zum wechselseitigen Nutzen" hat eine Situation entstehen lassen, in der ein fast unbegrenzter Glaube an die Leistung von Computersystemen einhergeht mit einem militärischen Wunschdenken, das von der Steuerbarkeit und Kontrollierbarkeit eines weltweiten Atomkrieges ausgeht.

Die Auseinandersetzung um die Möglichkeiten und Grenzen der künstlichen Intelligenz ist durch die Gegenüberstellung von Mensch und Maschine geprägt. Mit Argumenten, die häufig mit der Phrase *„im Prinzip..."* beginnen, wird für und wider die potentiell mögliche Leistungsfähigkeit der Maschine gestritten. Ein solcher Streit bietet zwar viele nützliche Hinweise und interessante Überlegungen[2], doch ist es letztlich unumgänglich, solche *Im-Prinzip*-Argumente durch *tatsächliche* zu ersetzen.

Denn das Studium der Informationsverarbeitung hat, wie Ulric Neisser feststellt, „sich noch keinem Menschenbild verpflichtet, das jenseits der Grenzen des Laboratoriums gültig wäre."[3] Und obwohl insbesondere auch das amerikanische Militär bisher enorme Summen zur Erforschung der Künstlichen Intelligenz bereitgestellt hat, ist diese Aussage immer noch gültig.

Ich will deshalb aufzeigen, was es tatsächlich bedeutet, komplexe technische Systeme zu entwickeln und einzusetzen, welche Möglichkeiten sie eröffnen und welche Grenzen sie setzen. Der Kernpunkt meiner Argumentation ist dabei, daß man nicht Mensch und Maschine vergleichen kann, sondern in einer gegebenen Situation immer nur die Möglichkeiten und Fähigkeiten der Konstrukteure einer Maschine mit denen der Benutzer.

Es sind die Prozesse der Herstellung und des Einsatzes von Artefakten, die man studieren muß, wenn man die Leistung beurteilen will, die ihnen potentiell innewohnt. Artefakte sind Produkte menschlicher Schöpferkraft; man darf deshalb ihre Charakteristika nicht mit den Eigenschaften und Merkmalen der sie hervorbringenden Prozesse verwechseln.

Artefakte bedeuten für sich alleingenommen nichts. Nur durch ihre Einbettung in das menschliche Handeln erhalten sie ihren Bezug zur Welt und damit ihren Sinn. Ob allerdings der Einsatz von künstlicher Intelligenz im militärischen Bereich unter den heute gegebenen Bedingungen sinnvoll ist, wage ich zu bezweifeln; ich halte ihn für gefährlich und unverantwortlich.

Software – ein besonderer Stoff

Grundlage für die Entwicklung von Computern mit Künstlicher Intelligenz (kurz: KI-Systeme) ist die Möglichkeit, logische Zusammenhänge, Verarbeitungsschritte und Daten (Fakten) in Form von Programmen (Software) festzulegen. Ich werde deshalb zunächst einige Anmerkungen zu den Gestaltungsmöglichkeiten und -gren-

zen von Software im Vergleich zu anderen ingenieurmäßig hergestellten Produkten machen. Dazu habe ich vier Punkte zusammengestellt, die die Unterschiede verdeutlichen sollen. Das Zusammenwirken dieser Faktoren ist im besonderen Maß ein Charakteristikum von Software und ihrer Herstellung.

Betriebszustände: Ingenieurmäßig erstellte Produkte können entweder als diskrete, analoge oder als hybride (eine Mischung aus den beiden ersten Typen) Systeme klassifiziert werden. Für analoge Systeme, die den Hauptbestandteil in den Ingenieurwissenschaften bilden, gibt es heute ausgereifte Hilfsmittel, wie zum Beispiel die Mathematik kontinuierlicher Funktionen, mit deren Hilfe es dem Ingenieur möglich ist, sicherzustellen, daß keine unvorhergesehenen Zustände auftreten, solange die Systemkomponenten innerhalb ihres normalen Betriebsbereiches arbeiten.

Mit der Entwicklung von Softwaresystemen jedoch, so argumentiert David Parnas, haben wir zum ersten Mal diskrete Systeme geschaffen, die eine ungeheure Vielzahl von Betriebszuständen aufweisen. Zwar werde hier versucht, die mathematische Logik entsprechend einzusetzen, doch habe sich in der Praxis bisher nicht gezeigt, daß ihr eine ähnlich hohe Bedeutung zukommt wie der traditionellen Ingenieurmathematik.

Hinzu kommt, daß Software in der Regel nur eine geringe repetitive Struktur (das heißt vielfache Verwendung ein und desselben Bausteins) aufweist. Dies führt zur relativen Unzuverlässigkeit von Softwaresystemen und, so Parnas weiter: „Es ist ein grundlegender Unterschied, der auch nicht durch technologische Verbesserungen aufgehoben werden wird."[4]

Funktionalität: Die Funktionen legen das Ein-/Ausgabeverhalten eines Softwaresystems fest. Generell kann man sagen, daß Funktionen beschreiben, wie ein technisches System auf Einwirkungen des Menschen oder Signale und Impulse anderer technischer Systeme reagiert. Die Gesamtheit der Funktionen gibt also an, welche Einwirkungen bzw. Eingaben insgesamt zulässig sind; die Funktionalität ist somit das wesentliche Merkmal im Hinblick auf die zweckbestimmte Verwendung.

Traditionell bestehen ingenieurmäßige Probleme darin, neue technische Lösungen für bereits bekannte Funktionen zu entwickeln. Beispielsweise hat sich die Funktionalität eines Autos in vielen Jahrzehnten kaum geändert, wohl aber die technische Realisierung. Bei der Softwareentwicklung jedoch müssen die funktionellen Anforderungen überhaupt erst ermittelt werden.

Dabei spielen zwei Faktoren eine besondere Rolle: der Mangel an allgemein anerkannten Standards und die Individualität der Akteure. Datenverarbeitungsverfahren sind in der Regel situations- und organisationsspezifisch zugeschnitten; es gibt hier kaum Normen oder verbindliche Verfahrensvorschriften. Je weniger aber das Problemfeld (technisch) vorstrukturiert oder standardisiert ist, desto stärker kommen individuelle Vorlieben, unterschiedliche Interessen und persönliche Motive der an einer Systementwicklung beteiligten Personen zum Tragen. Dies gilt gerade auch für komplexe technische Probleme, bei denen eine Vielzahl unterschiedlicher Lösungsmöglichkeiten gegeben ist. Die Funktionalität des Softwaresystems ist weniger das Ergebnis einer technischen Analyse als vielmehr das Resultat von Verhandlungen und Vereinbarungen.

Rückbezüglichkeit: Ausgangspunkt für die Softwareherstellung sind bestehende Regelungen, Verfahren und Arbeitsabläufe. Mit der Einführung eines Software-

systems ändern sich diese aber grundlegend und damit auch das Verhalten der Menschen. Dadurch werden viele Annahmen, die der Entwicklung zugrunde gelegt worden sind, hinfällig. Die Folgen sind, daß zum einen das der Entwicklung zugrundegelegte Modell nicht mehr im vollen Maße gültig ist und zum anderen, daß durch die veränderte Situation neue Anforderungen entstehen, die erst jetzt in ihrer Tragweite und mit all ihren Konsequenzen erkannt werden können.

Nimmt man hinzu, daß sich zum Beispiel aufgrund neuer technischer Möglichkeiten oder veränderter Markterfordernisse das Einsatzumfeld ohnehin mit der Zeit ändert, so wird unmittelbar einsichtig, was Meir M. Lehman auch aus seinen empirischen Untersuchungen ableitet: „... die Notwendigkeit zu fortwährenden Veränderungen ist ein Wesensmerkmal der Computerbenutzung."[5]

Entscheidend ist hier die Vielfalt der Nutzungsmöglichkeiten in Verbindung mit der Komplexität des Einsatzumfeldes. Von besonderer Bedeutung ist dabei, daß Qualitätsmaßstäbe nicht wie in anderen Ingenieurbereichen durch technisch/physikalische Anforderungen wie zum Beispiel die Materialbeschaffenheit bestimmt sind, sondern überwiegend von der Bewertung des Menschen im Hinblick auf Zweckmäßigkeit und Angemessenheit abhängen. Die ausgeprägte Rückbezüglichkeit entsteht u.a. deshalb, weil im Rahmen der Systementwicklung und der Einführung die Menschen jeweils versuchen, ihre Interessensphäre zu bewahren. Will man beispielsweise die Anzahl der Programmzeilen als Maßstab für die Leistungsfähigkeit der Programmierer nehmen, so wird man feststellen, daß sich die Zahl der Zeilen eines Programms vervielfacht, sobald diese Absicht bekannt wird. Bezüglich des Einsatzes ist diese Rückbezüglichkeit vor allem auch bei militärischen Anwendungen gegeben. Sobald zum Beispiel ein Gegner die einem System zugrundeliegende Einsatzstrategie erfährt, wird er sein Verhalten umgehend ändern. Dies erfordert die erneute Anpassung des Systems und so fort.

Theoriebildung: Beim Entwurf von Software kommt es wesentlich darauf an, alle Sonderfälle und Ausnahmebedingungen vollständig zu erfassen und ordnungsgemäß zu verarbeiten. Entscheidend ist dabei, inwieweit die Systemingenieure das zu lösende Problem mit all seinen Randbedingungen durchschauen. Da dies nicht so ohne weiteres und auf Anhieb möglich ist, sondern entsprechende Lernprozesse erfordert, müssen Programme wie auch die ihnen zugrundeliegenden Dokumente mehrfach überarbeitet oder gar verworfen werden. Jede Version eines Programmes oder Dokumentes spiegelt jedoch immer nur den erreichten Zustand wider, denn die Gründe und Argumente, die zu solchen Revisionen führen, können nicht in ihrer Gesamtheit dokumentiert werden.

Programmieren kann nach Peter Naur daher nicht in erster Linie als Produktion von Programmen und zugehörigen Texten betrachtet werden, sondern als ein Prozeß, bei dem die Programmierer im Laufe der Systementwicklung eine Theorie darüber entwickeln, wie die vorhandenen Probleme durch die Programmausführung gelöst werden können. Diese Theorie ist nur in den Köpfen der Entwickler vorhanden: „Das Wiederherstellen der Theorie lediglich aufgrund der Dokumentation ist gänzlich unmöglich."[6] In traditionellen Ingenieurbereichen sind aber die theoretischen Grundlagen bereits vor der Produktentwicklung erarbeitet.

Was für die Programmierer bzw. Systementwickler gilt, stellt in veränderter Form auch ein Problem für die Benutzer

dar: Sie müssen sich im Rahmen der Benutzung die Systemrationalität – also die Theorie der Entwickler – erschließen. Da sie aber weder am gesamten Entwicklungsprozeß teilnehmen können noch die Entwickler während der Benutzung in der Regel verfügbar sind, ist es notwendig, Ausdrucksmöglichkeiten bereitzustellen, die die Systemrationalität transparent machen.

Etwa die Hälfte des Codes praktisch eingesetzter Software besteht daher aus Benutzungsschnittstellen, Fehlermeldungen, Ausnahmebehandlungen und Kommentierungen, Aspekten also, die der Verständigung dienen und nicht der algorithmischen Verarbeitung durch den Rechner. Diese Aspekte sind weder aus formalen Qualitätseigenschaften ableitbar noch sind sie formal „überprüfbar".

Software-Entwicklung ist also in erster Linie ein sozialer und kognitiver Prozeß, bei dem die Qualität des erzeugten Produktes im wesentlichen vom Problemverständnis der beteiligten Menschen abhängt. Dies gilt sowohl für die Herstellung als auch für die Benutzung.

Grundsätzlich kann man Software auch als eine Ansammlung von Plänen auffassen, denn die Entwickler legen fest, was zur Laufzeit des Programms wie, in welcher Reihenfolge und unter welchen Umgebungsbedingungen ausgeführt werden soll. Mit dieser Sichtweise lassen sich wiederum viele Anknüpfungspunkte an die traditionelle Ingenieurwissenschaft angeben, denn Planungsprozesse offenbaren auch dort dieselben Probleme, die man aus der Softwareentwicklung kennt: Kosten werden nicht eingehalten, Termine werden überschritten, das Ergebnis entspricht oft nicht den Vorstellungen der Auftraggeber und der Betroffenen und nicht alle Ereignisse, die die Ausführung der Pläne und ihr Ineinandergreifen beeinflussen, können vorhergesehen werden. Auch heute noch gilt in der Softwaretechnik der Grundsatz, daß sich die Qualität von Software erst im Einsatz erweist.

Die praktischen Konsequenzen dieses Grundsatzes will ich im nächsten Abschnitt am Beispiel der militärtechnologischen Entwicklung[7] behandeln. Der historische Abriß zeigt, daß mit der zunehmenden Komplexität der eingesetzten Systeme die Probleme wachsen, denn je komplexer die technischen Systeme werden, desto stärker ist unsere Sicherheit von den Menschen abhängig, die sie einsetzen und benutzen. Je mehr der Mensch aber aus den Entscheidungsprozessen verdrängt wird, desto größer wird das Risiko eines Atomkrieges.

Das Pearl-Harbour-Syndrom

Aus Gründen der Übersichtlichkeit werde ich den geschichtlichen Abriß in fünf Phasen unterteilen, die sich jeweils an der Integration verschiedener Funktionsbereiche bzw. unterschiedlicher Teilsysteme orientieren. Dies unterstreicht einerseits die Entwicklungsrichtung und ermöglicht mir andererseits, die besonderen Probleme der einzelnen Integrationsstufen aufzuzeigen. Man muß sich aber vor Augen halten, daß mit dieser Strukturierung nicht einzelne, explizit definierte und in sich abgeschlossene Entwicklungsphasen verbunden sind; viele Entwicklungen verlaufen zum Teil parallel oder zeitlich stark überlappend ab.

Seit dem Überfall der Japaner auf den Hafen von Pearl Harbour, bei dem die amerikanische Pazifikflotte fast vollständig vernichtet wurde, ist die Angst, daß sich ein solcher Vorfall auch auf amerikanischem Territorium wiederholen könnte, zu einem treibenden Element der militärtechnologischen Entwicklung in den USA

geworden. Pearl Harbour steht für die Folgen, die durch ungenügende Information und mangelnde Vorwarnung entstehen können.

Vom amerikanischen Militär werden daher nach dem Zweiten Weltkrieg enorme finanzielle Mittel für die Entwicklung von computergestützten Frühwarnsystemen bereitgestellt. Der Aufbau verschiedener Radarketten und Frühwarnsysteme sowie die Einrichtung der nordamerikanischen Luftverteidigungszentrale NORAD im Jahre 1957 schließen die *Aufbauphase* ab. Die Einrichtung von NORAD trägt der Tatsache Rechnung, daß mit den elektronischen Frühwarnsystemen ein kontinuierlicher Datenstrom entsteht, der auch in Friedenszeiten ohne Unterbrechung verarbeitet werden muß.

In den sechziger Jahren wird die Entwicklung des nuklearen Warnsystems wesentlich durch den von Präsident Kennedy bewirkten Übergang von der Doktrin der „massiven Vergeltung" zur Doktrin der „flexiblen Antwort" geprägt. Das bedeutet auf der einen Seite, daß der Präsident nun direkt in den Prozeß der Auslösung der verschiedenen Alarmbereitschaftsstufen einbezogen wird und auf der anderen Seite, daß Aufklärung und Einsatzplanung stärker integriert werden.

Als Konsequenz wird im Juni 1962 im Pentagon das nationale militärische Befehlssystem NMCS etabliert, durch das der Präsident direkt Frühwarninformationen vom strategischen Luftkommando SAC und von NORAD erhält und über das er Befehle an die Nuklearstreitkräfte erteilen kann. Militärtechnologisch ist dies die *zweite Phase,* die sich als Integration von Frühwarnung und Einsatzplanung charakterisieren läßt.

Als Prototyp einer solchen Integration kann das Anfang der sechziger Jahre mit einem Aufwand von insgesamt über 20 Mrd. Dollar aufgebaute halbautomatische bodengestützte Luftverteidigungssystem SAGE betrachtet werden. Das Ziel ist, die Hauptquartiere (combat centers) der Luftverteidigung (Air Defense Division) zu automatisieren. Es ist das ambitionierteste Projekt der Militärs in den sechziger Jahren und es ist ein Fehlschlag. Nach unzähligen Revisionen scheitert die Entwicklung an der technischen Komplexität der korrekten Erfassung feindlicher und befreundeter Flugzeuge, insbesondere wenn deren Flugbahnen sich kreuzen.[8]

An SAGE wird noch ein weiteres Problem komplexer Frühwarnsysteme deutlich: Wenn die Bedienungsmannschaften sich an die zur Betreibung des Systems festgelegten Regeln und Vorschriften gehalten hätten, wäre ein Zusammenbruch des Systems schon bei kleinen Störungen im Radarbild unvermeidlich gewesen. „Sehr wenige komplexe Systeme würden jemals laufen", folgert Paul Bracken, „falls die Benutzungsregeln buchstabengetreu befolgt würden."[9] Irgendwie haben es die Operateure jedenfalls geschafft, das System funktionsfähig zu halten. Die dazu nötigen mündlichen Absprachen sind in offiziellen Berichten nie erwähnt worden. Ja mehr noch, die Bediener sind noch nicht einmal in der Lage gewesen, den Ingenieuren zu erklären, welche Bedienungsprozeduren sie in bestimmten Situationen anwenden würden.

Parallel zur Entwicklung der ersten Frühwarn- und Entscheidungssysteme beginnt der Aufbau von Rechnernetzen, mit deren Hilfe ein schneller und sicherer Datenaustausch zwischen den verschiedenen computergestützten Systemen gewährleistet werden soll. Das Militär fordert ein Verteidigungssystem, das selbst im Falle größter Zerstörungen – analog zum Fall Pearl Harbour – weiter funktionieren würde. Bei der Entwicklung des ersten

Rechnernetzes, des ARPANET, geht es also nicht darum, „den Stand der Kunst bei Kommunikationsnetzen für Computer zu verbessern, sondern die militärische Überlebensfähigkeit sicherzustellen."[10]

Doch trotz einer zunehmend verbesserten Informationstechnologie bleibt der „Erfolg" aus. Überrascht wird die amerikanische Regierung beispielsweise von der sowjetischen Intervention in der Tschechoslowakei, der Tet-Offensive in Vietnam, dem Yom-Kippur-Krieg und von der Übernahme der Falklandinseln durch Argentinien. Dies führt zwar Anfang der siebziger Jahre zu einer Untersuchung durch den amerikanischen Kongreß, doch kann bis heute keine definitive Ursache für dieses Versagen angegeben werden.[11]

Ungeachtet dessen beginnt die *dritte Phase* mit dem Aufbau des weltweiten militärischen Befehls- und Kommunikationssystems Wimex (WWMCCS, Worldwide Military Command and Control System). Die Installation von Wimex geht einher mit einer generellen Verbesserung der technischen Infrastruktur sowie einer organisatorischen Umstrukturierung. Damit ist die nächste Stufe erreicht, die Integration der Frühwarn- und Befehlssysteme mit den Kommunikationssystemen.

Zum ersten Mal gibt es ein vollständig integriertes Frühwarn- und Entscheidungssystem mit einem Hauptquartier, das auf vier Befehlszentren verteilt ist: NORAD, SAC, das NMCC im Pentagon und das Ausweichzentrum in Raven Rock. NORAD spielt dabei eine besondere Rolle, denn nur hier werden die Daten sämtlicher Systeme empfangen und analysiert und die Auswertungen an die anderen Befehlszentralen übermittelt.

Das NORAD-System besteht aus 84 Großcomputern, deren Programme über 10000000 Zeilen umfassen. Das System ist teilweise veraltet und für die Aufgabe ungeeignet. Computer müssen ausgetauscht werden und die Software muß fortwährend angepaßt, verbessert und erweitert werden. Zudem führen die Positionierung und die Konfiguration von Satelliten und Sensorsystemen jeweils zu Änderungen im Gesamtverhalten des Systems. Mit jeder Änderung werden umfangreiche Tests erforderlich.

Um möglichen Fehlalarmen vorzubeugen, sind daher allein in den Jahren 1979 und 1980 mehr als 3700 routinemäßige Raketenerkennungskonferenzen einberufen worden. Trotz dieser Kontrollen hat es über 147 Fehlalarme gegeben, wobei viermal die zweite von drei möglichen Warnstufen erreicht worden ist, das heißt sowohl die mit Atombomben bestückten Langstreckenbomber der Luftwaffe als auch die Interkontinentalraketen waren startklar gemacht worden und warteten lediglich darauf, den atomaren Gegenschlag auszuführen.[12]

Die Senatoren Gary Hart und Barry Goldwater, die im Auftrag des amerikanischen Kongresses die Fehlalarme bei NORAD untersucht haben, kommen daher in ihrem Bericht zu dem Schluß, daß wir uns auf die gemeinschaftlich getroffene Bewertung der Menschen, die mit dem System arbeiten, verlassen müßten, um Fehlalarme korrekt zu erkennen und damit umgehen zu können.[13] Wie bei SAGE gilt auch für NORAD, daß es nicht die Zuverlässigkeit der Technik ist, die Sicherheit schafft, sondern der flexible und verantwortungsbewußte Umgang des Menschen mit dieser Technik.

Die weitgehende Verschmelzung von Frühwarn- und Aufklärungsdaten sowie der kurze Zeitraum, in dem im Konfliktfall entsprechende Maßnahmen ergriffen werden müssen, erfordert aber, daß auch die Einsatzplanung automatisiert wird. Diese basiert für die amerikanischen Nuklear-

streitkräfte auf einem umfangreichen Plan mit über 40000 strategischen Zielen weltweit (SIOP, Single Integrated Operational Plan); der entsprechende Einsatzplan der NATO (NOP, Nuclear Operations Plan) enthält zwischen 18000 und 25000 Zielen.[14]

Die Verwaltung dieser Ziele ist nur mit Computern durchführbar. Sie umfaßt ihre kontinuierlich notwendige Aktualisierung und Erweiterung sowie die Zusammenfassung nach Prioritätsklassen gemäß ihrer militärischen Bedeutung. Nur so ist es möglich, sie nach der Strategie der flexiblen Antwort geeignet auszuwählen und zu Einheiten zusammenzufassen, um sie dann mit unterschiedlichen Waffensystemen angreifen und zerstören zu können.

Das beinhaltet zum einen eine Palette unterschiedlicher Optionen und zwar vom gezielten atomaren „Warnschuß" über einen regional begrenzten Atomkrieg bis hin zu einem massiven Angriff auf zahlreiche Ziele gleichzeitig und zum anderen die Möglichkeit, das sowjetische Befehlssystem auszuschalten.[15]

So eine detaillierte Einsatzplanung gründet notwendigerweise auf Annahmen, deren Richtigkeit nicht überprüfbar ist. Kriegsspiele (war games), die auf einem Computer ablaufen, dienen dazu, SIOP regelmäßig auszuwerten und aufgrund der Interpretationen sowjetischer Kriegspläne zu aktualisieren. Diese Aktualisierung ist also rein hypothetisch: „Das System ist schon vor langer Zeit zu komplex geworden, um noch von Menschen gehandhabt werden zu können."[16]

Die mit der Automatisierung scheinbar gegebene und politisch gewollte Flexibilität verleitet zu der Annahme, daß ein begrenzter Atomkrieg möglich sei. Über Forschungs- und Entwicklungsprogramme soll daher eine nukleare Infrastruktur aufgebaut werden, die es gestattet, „bewaffnete Auseinandersetzungen mit den Staaten des Warschauer Vertrages in jeder Form und in jedem Ausmaß zu führen und siegreich zu beenden."[17] In einem speziell aufgelegten Forschungsprogramm zur Steigerung der Produktivität und Qualität von Software heißt es u.a.: „Um die Führung in dieser Technologie und implizit die militärische Überlegenheit zu halten, müssen die USA für eine Belebung ihrer Grundlagenforschung sorgen..."[15]

Erklärtes Ziel des amerikanischen Verteidigungsministeriums ist es, die militärische Überlegenheit durch den Einsatz neuer Technologien sicherzustellen. Dazu reichen die vorhandenen Systeme nicht aus. Frühwarnsysteme registrieren beispielsweise den Feuerstrahl von Interkontinentalraketen oder auch die Schraubengeräusche von Unterseebooten. Sie liefern also Daten erst, wenn ein Angriff bereits erfolgt. Um aber flexibel und vor allem vor dem Gegner reagieren zu können, müssen auch die Aufklärungsinformationen der Geheimdienste ausgewertet und bei der Einsatzplanung berücksichtigt werden.

Die damit einhergehende Integration sämtlicher bisher angesprochenen Komponenten und Teilsysteme wird durch die Formel C^3I (Command, Control, Communication and Intelligence)[19] charakterisiert; sie verkörpert die *vierte Phase.*

Unabdingbare Voraussetzung für alle militärischen Operationen ist das ordnungsgemäße Funktionieren der C^3I- Systeme, weil alle für das Führen einer Schlacht wichtigen Funktionen jetzt in einem einzigen computerisierten System integriert sind. Sie werden deshalb auch als Schlachtenführungssysteme (battle management systems) bezeichnet.

Wesentlich aber ist die Integration dieser Systeme in eine militärstrategische Gesamtplanung. In der seit August 1982 vorliegenden Militärdoktrin „Airland Battle

2000" wird daher das Konzept des „integrierten und erweiterten Schlachtfeldes" eingeführt und begründet. Vor allem unter der Annahme, daß das Schlachtfeld chemisch, biologisch und atomar verseucht sein wird, wird die besondere Bedeutung, die der Computertechnik zukommt, gewürdigt: „Der Einfluß, den die Technologie auf das Gefechtsfeld haben wird, ist gekennzeichnet durch größere Mobilität, höhere Feuerkraft, umfangreiche Manövrierkräfte, die unabhängig operieren können, sowie weitreichende Kampfpläne, die zwar kompliziert sein werden, jedoch mit Hilfe Künstlicher Intelligenz ohne weiteres synchronisiert und aufeinander abgestimmt werden können."[20]

Hier kündigt sich bereits an, was ein Jahr später zum offiziellen Forschungsprogramm wird. Das Militär soll in der Lage sein, noch schneller und flexibler als bisher zu reagieren. Aufklärungsinformationen müssen demzufolge noch detaillierter sein, und damit auch die Einsatzpläne. Dieser Grad an Flexibilität macht ein zentrales, integriertes Schlachtenführungssystem unerläßlich.[21]

Der Versuch, diese Anforderungen umzusetzen, leitet die *fünfte Phase* ein. Sie führt zur nächsten und vermutlich letzten Integrationsstufe, die darin besteht, ein einziges welt(raum-)umspannendes computergestütztes Schlachtenführungssystem aufzubauen. Am 23. März 1983 verkündet Präsident Reagan seine Vision einer strategischen Verteidigungsinitiative (SDI, Strategic Defense Initiative), die als „Krieg der Sterne" in der Öffentlichkeit bekannt wird. Kernstück von SDI ist ein computerisiertes zentrales Schlachtenführungssystem. Die Planungen für dieses System sehen vor, daß es sowohl die Abwehr feindlicher Atomraketen mit Laserwaffen steuert, als auch die gesamte für einen atomaren Gegenschlag erforderliche Einsatzplanung. Das beinhaltet u. a. auch die Integration sämtlicher Frühwarn- und Aufklärungsfunktionen.[22]

Die technische Realisierung scheint jedoch aufgrund der enormen Anforderungen nur unter Zuhilfenahme bisher nicht bekannter „revolutionärer" Techniken möglich. KI-Systeme, insbesondere Expertensysteme, sollen diese revolutionäre Technologie verkörpern.[23] Allerdings ist der Entwicklungsaufwand ungeheuer hoch: „Es beinhaltet nicht mehr als ein volles, rationales Verständnis des menschlichen Verhaltens, das nachgebildet werden soll."[24]

Der Einsatz von KI-Systemen stellt also eine konsequente Weiterentwicklung der mit Pearl Harbour eingeleiteten Entwicklung dar. Zwei Punkte, die vor allem für die Bewertung der weiteren Entwicklung wichtig sind, möchte ich noch einmal hervorheben. Zum einen geht es bei der Anwendung von KI-Systemen im militärischen Bereich nicht um voneinander unabhängige Einzellösungen, sondern um ihre Integration in ein militärtechnologisches und militärpolitisches Gesamtkonzept. Zum anderen soll der Mensch, der bisher der Garant für Zuverlässigkeit und Sicherheit war, soweit wie möglich aus allen Entscheidungsprozessen verdrängt werden.

Künstliche Intelligenz – Neuer Wein in alten Schläuchen

Um die technologischen Grundlagen zu entwickeln, leitet die DARPA (Defense Advanced Research Projects Agency) eine Institution, deren Aufgabe es ist, die Forschungsaktivitäten der drei Teilstreitkräfte Heer, Luftwaffe und Marine zu koordinieren, am 28. Oktober 1983 dem amerikanischen Kongreß ein Forschungsprogramm

mit dem Titel „Strategic Computing" zu. Dieses Vorhaben stellt im wesentlichen heraus, daß man nur mit Hilfe einer neuen Generation von „Maschinenintelligenz" in der Lage sein werde, die bestehenden Probleme im militärischen Bereich und insbesondere bei SDI zu lösen. In der Einleitung zu diesem Forschungsprogramm heißt es:

> „Im Gegensatz zu bisherigen Computern wird die neue Generation menschenähnliche, 'intelligente' Fähigkeiten zum Planen und Denken aufweisen. Die Computer werden ebenfalls Fähigkeiten besitzen, die durch visuelle Wahrnehmung und Sprache direkte natürliche Interaktionen mit ihren Benutzern und ihrer Umgebung ermöglichen.
>
> Bei Benutzung dieser Technologien werden Maschinen komplexe Aufgaben mit nur geringer menschlicher Intervention oder sogar vollständig autonom ausführen."[25]

Um die Tauglichkeit von KI-Systemen für militärische Zwecke demonstrieren und die dabei auftretenden Probleme studieren zu können, werden in dem Forschungsprogramm drei Anwendungen beschrieben, die in den nächsten zehn Jahren entwickelt werden sollen. Dabei handelt es sich um ein autonom operierendes Landfahrzeug für das Heer, einen automatischen Co-Piloten für die Luftwaffe und ein Schlachtenführungssystem für die Marine.

Das autonome Fahrzeug soll Aufklärungsfahrten unternehmen können, die weit ins gegnerische Hinterland führen. Um einen Einsatz erfolgreich beenden zu können, muß es in der Lage sein, unvorhergesehenen Hindernissen auszuweichen und eine neue Route zu planen. Dieses Fahrzeug könnte vor allem in atomar, biologisch oder chemisch verseuchten Gebieten eingesetzt werden.

Der automatische Co-Pilot soll den Piloten bei der Navigation, der Steuerung und dem Einsatz der Bordwaffen unterstützen. Wird das Flugzeug von einer Rakete angegriffen, soll das System die Steuerung übernehmen, da die Piloten bei den für die Ausweichmanöver notwendigen Beschleunigungen das Bewußtsein verlieren. Der Rechner soll vom Piloten während des Fluges programmiert werden.

Das Schlachtenführungssystem soll in der Lage sein, einen detaillierten Überblick über das Schlachtfeld zu geben, Hypothesen über die Absichten des Gegners aufzustellen und unter Unsicherheitsbedingungen sowie bei mehreren sich widersprechenden Zielen Entscheidungen zu treffen. Die Interaktion mit dem System soll in natürlicher Sprache erfolgen.

Die wissenschaftliche Beratungskommission des amerikanischen Verteidigungsministeriums geht davon aus, daß die Rechengeschwindigkeit (Zahl der Operationen pro Zeiteinheit) der zur Zeit in den Computerlaboratorien entwickelten Systeme etwa um das 1000000000fache gesteigert werden müßte, um zu militärisch einsatzfähigen Systemen zu gelangen. Außerdem müßte die Antwortzeit von einer Stunde bei heutigen Systemen auf eine Sekunde verkürzt werden.

Derart überzogene Anforderungen lassen sich nur aus dem Wunsch nach einer konsequenten Fortsetzung der bisherigen Entwicklung ableiten; sie können nicht mit dem aktuellen Stand von Forschung und Technik begründet werden. Das macht auch verständlich, warum die heute immer noch ungelösten Probleme ignoriert werden. Künstlicher Intelligenz scheint eine Art magischer Zauberformel innezuwohnen, die alles Wünschenswerte möglich scheinen läßt. Dies muß wohl so sein, weil man sonst gezwungen wäre, grundsätzlich umzudenken.

Doch schon bei den ingenieurmäßigen Voraussetzungen zeigen sich erschrek-

kende Defizite: Die technischen Mittel sind für Problemstellungen dieser Größenordnung völlig unzureichend. Aber auch „neuer" Wein kann in „alten", porösen Schläuchen nicht reifen.

Dem Ingenieur ist nichts zu schwer – oder doch?

Gewiß ist es möglich, mit KI-Systemen Anwendungsbereiche zu erschließen, die mit traditionellen Verfahren nicht oder nur unzureichend erfaßt und modelliert werden können. Das ändert jedoch nichts an der Tatsache, daß auch zur Entwicklung von KI-Systemen sowohl ein entsprechendes ingenieurmäßiges Rüstzeug erforderlich ist als auch, daß der jeweilige Anwendungsbereich einer ingenieurmäßigen Betrachtungsweise zugänglich sein muß. Insofern hat künstliche Intelligenz auch nichts Magisches an sich.

Heute sind allgemein zwei verschiedene Definitionen von künstlicher Intelligenz im Gebrauch:

> „KI-1: Die Verwendung von Computern zur Lösung von Problemen, die bisher nur durch Anwendung der menschlichen Intelligenz zu lösen waren;"
> und:
> „KI-2: Die Verwendung einer bestimmten Reihe von Programmiertechniken, die als heuristische bzw. regelbezogene Programmierung bezeichnet werden. Bei diesem Ansatz werden menschliche Experten einer Untersuchung unterzogen, um festzustellen, welche heuristischen Methoden bzw. Faustregeln sie bei der Problemlösung verwenden."[26]

Bei der ersten Definition wird Künstliche Intelligenz als Sammlung von Problemen charakterisiert, während die zweite Definition sich auf die Benutzung bestimmter Techniken bezieht. KI-Systeme können durchaus beiden Definitionen entsprechen. Nach David L. Parnas werden selbst bei den besten Programmen im Sinne von KI-1 bei der Programmierung herkömmliche ingenieurwissenschaftliche Methoden angewandt. Was sie leisten, hängt folglich von den Fähigkeiten der Entwickler ab. Das heißt aber, daß auch hier die im zweiten Abschnitt aufgeführten Probleme der Softwareherstellung zum Tragen kommen.

Die Probleme sind sogar noch größer, weil die wenigen in der Softwaretechnik entwickelten Techniken und Verfahren bisher nicht auf die Entwicklung von KI-Systemen übertragen worden sind. Beispielsweise hat man aufgrund der vergleichsweise strengen Anforderungen im militärischen Bereich für die Herstellung traditioneller Softwaresysteme eine Fülle von Standards und Prüfkriterien festgelegt, die bei der Entwicklung und der Systeminstallation eingehalten werden müssen.[27] Sie sind jedoch schon für die traditionelle Entwicklung komplexer Softwaresysteme nicht ausreichend, wie dies die geschilderten Probleme zeigen.

KI-Systeme sind bisher fast ausschließlich unter experimentellen Laboratoriumsbedingungen entwickelt worden, wo selbst diese Maßstäbe und Normen nicht gefordert sind. Vielmehr legen die Entwickler einen größeren Wert auf die bei der Programmentwicklung auftauchenden neuen Probleme (open-ended research) als auf die technische Zuverlässigkeit.[28] Insbesondere auch für die sogenannten Expertensysteme besteht das Problem der Zuverlässigkeit: Die Regeln, die man durch Befragung menschlicher Experten (Wissensakquisition) erhält, sind meist inkonsequent, ungenau und unvollständig. Die Programme werden durch praktisches Herumprobieren entwickelt, wobei immer dann eine neue Regel eingebaut wird,

wenn sich ein Fall ergibt, der mit den bisherigen Regeln nicht behandelt werden kann. Nach Parnas sind solche Programme noch weniger vertrauenswürdig als herkömmliche.

Ohnehin ist es schwer, solche Programme zu verstehen; die Abhängigkeiten und Zusammenhänge zwischen der Vielzahl von Regeln sind kaum zu durchschauen. Das liegt nicht – wie häufig behauptet – an der Verwendung formaler Sprachen mit einem eingeschränkten Ausdrucksrepertoire, sondern daran, daß es für den Menschen schwierig ist, die Vielzahl der Abhängigkeiten und Sonderfälle geordnet zu erfassen. Wie P. K. Davis betont, ist dieses Problem nicht durch die Verwendung von Regeln, die in natürlicher Sprache codiert werden, aus der Welt zu schaffen. Aufgrund seiner Erfahrungen bei der Benutzung von KI-Techniken zur Entwicklung von Schlachtensimulationsprogrammen (war games) kommt er zu dem Ergebnis, daß die solchen Regeln unterstellte Transparenz irreführend sei, weil es schwierig ist, komplette Regelblöcke zu verstehen und zu überprüfen. Dies liegt zum Teil daran, daß man nicht feststellen kann, ob sie vollständig sind.[29]

Zwar wird viel darüber geredet und geschrieben, was man im Prinzip mit Expertensystemen alles machen könnte, stellt Frank Puppe fest: „Aber wir sind sehr weit entfernt davon, hinreichend generelle und vor allem erprobte Techniken hierfür zu haben, die es rechtfertigen, ihre Anwendung als 'Engineering' zu bezeichnen."[30] Wo aber nicht einmal die ingenieurmäßigen Voraussetzungen vorhanden sind, können keine vertrauenswürdigen Systeme zustande kommen.

Das „Expertensystem-entwicklungsexpertensystem"

Der Versuch, die aufgeführten Defizite selbst wiederum durch die Entwicklung und den Einsatz von KI-Systemen zu beheben, kann als „ballistisches" Verhalten der Problemlösung bezeichnet werden. Dietrich Dörner hat diesen Ausdruck geprägt, um ein Problemlöseverhalten zu kennzeichnen, das nicht nach den wirklichen Ursachen und Gründen für die jeweils anstehenden Probleme forscht, sondern unabhängig von den tatsächlichen Ereignissen an einer einmal aufgestellten Behauptung festhält, so wie die Flugbahn eines Geschosses, das einmal abgefeuert ist, im nachhinein nicht mehr korrigierbar ist. „Eine Hypothese wird aufgestellt, und damit ist die Realität bekannt. Eine Überprüfung erübrigt sich."[31] Die Hypothese lautet in diesem Fall: Im Prinzip ist die Leistung von KI-Systemen unbegrenzt. Und da dies so ist, können nicht nur die bisher ungelösten Probleme beseitigt, sondern zugleich auch die zu bewältigenden Anforderungen um mehrere Größenordnungen über dem heute technisch Machbaren angesetzt werden.

Beispielsweise geht die wissenschaftliche Beratungskommission des amerikanischen Verteidigungsministeriums davon aus, daß der Aufwand zur Aufstellung der zehn- oder gar hunderttausend Regeln, die ein Expertensystem verarbeiten soll, die Gesamtkosten der Hard- und Softwareentwicklung weit übersteigen könnte.[32]

Um dies zu vermeiden, lautet daher die „ballistische" Lösung, den Wissensakquisitionsprozeß zu automatisieren, indem Expertensysteme zur Entwicklung von Expertensystemen entwickelt werden. So schlägt die Beratungskommission vor, Expertensysteme zu konzipieren, die imstande sind, „Texte zu lesen und zu ver-

stehen", um so ohne einen menschlichen Experten Wissensstrukturen erzeugen und revidieren zu können. „Die Systeme sollten auch in der Lage sein, direkt mit einem menschlichen Experten zu interagieren, um ohne die Intervention eines Informatikers die Expertise zu erhalten."[33]

Abgesehen davon, daß eine solche technische Lösung lediglich eine Verlagerung des Problems darstellt, weil es auch solche Expertensysteme noch nicht gibt, besteht das Problem darin, daß sowohl der Experte als auch der Konstrukteur nicht mehr in der Lage wären, festzustellen, inwieweit tatsächlich alle Fälle und Ausnahmen vollständig und adäquat berücksichtigt worden sind. Weitaus schwieriger noch würde sich das Problem der Anpassung und Erweiterung solcher Systeme gestalten.

Man weiß in der Softwaretechnik, daß es für die Ingenieure in der Regel nicht möglich ist, größere Programmteile, die von einer Maschine generiert worden sind, zu verstehen und zu überprüfen. Die vielen Möglichkeiten, den Programmtext informell anzureichern, zum Beispiel durch Kommentare oder eine geeignete Namensgebung für Prozeduren, Regeln und Daten, gehen bei einer automatischen Erzeugung verloren. Die Konsequenz lautet hier, bei Änderungen die manuell erstellten Teile vom Menschen ändern zu lassen und, soweit es die anderen Teile betrifft, diese grundsätzlich neu zu generieren.

Soll also ein System vollständig oder zum größten Teil automatisch erzeugt werden, dann müssen auch alle Änderungen und Erweiterungen automatisch erzeugt werden. Das bedeutet aber, daß die Experten, die die Problemstellung formulieren, das Ein-/Ausgabeverhalten des Erzeugersystems exakt kennen müssen. Schließlich müssen sie die Eingabe für dieses System präzise und so formulieren, daß es daraus ein System generiert, das sich genau ihren Intentionen gemäß verhält. Das würde aber wiederum gerade die Informatikerkompetenz erfordern, die durch das Erzeugerexpertensystem ersetzt werden sollte.

Abgesehen davon wären die Experten, mit welcher Kompetenz auch immer, mit dieser Aufgabe überfordert. Bisher gibt es kein einziges praktisch eingesetztes Softwaresystem, das nur unter Bezugnahme auf ein geschriebenes Dokument entwikkelt worden wäre; dies ist praktisch nicht machbar.

Zu Beginn einer Systementwicklung wird ein Modell des Anwendungsbereiches erstellt. In jedem Modell sind aber nur diejenigen Aspekte der Wirklichkeit erfaßt, die für den jeweiligen Zweck als relevant erachtet werden. Was aber jeweils für ein bestimmtes Modell relevant ist, kann man nicht durch Modellieren herausfinden, sondern nur durch Erproben in der Wirklichkeit. Nur indem man handelt, kann man feststellen, wie gut die konzeptionellen Vorstellungen sind: „Selbst wenn man jeden Gedanken in Form eines Modells faßt, bedeutet Handeln, sich von dem Modell zu verabschieden und an der ganzen, reichhaltigen Welt teilzuhaben."[34]

Es bleibt also nur die Möglichkeit, die Systeme zu erproben, indem man sie einsetzt. Aufgrund der Besonderheiten im militärischen Bereich ist dies aber nicht möglich; es gibt keine Testmöglichkeiten unter realen Bedingungen – ein Atomkrieg wird nur einmal stattfinden.

Zurück zur Natur

Variabilität und Flexibilität sind in der Natur grundlegende Voraussetzungen für die erfolgreiche Anpassung von Lebewe-

sen und Umwelt. Daher scheint es naheliegend, auch für die Interaktion des Menschen mit seiner technischen Umwelt eine möglichst natürliche Form zu finden – eben die natürliche Sprache. Neben der Vielfalt der sprachlichen Ausdrucksmöglichkeiten und ihrer universellen Verwendbarkeit weist sie den Vorteil auf, daß sie im Gegensatz zu computerspezifischen Kommandosprachen nicht zusätzlich erlernt werden muß. Außerdem kann bei natürlichsprachlichen Schnittstellen jeder Benutzer seinem eigenen individuellen Stil frönen.

Mit Hilfe von Expertensystemen, die natürliche Sprache verstehen und verarbeiten können sollen, erhofft man sich deshalb, die Komplexität der Benutzung reduzieren zu können.

Doch der Begriff „Verarbeitung natürlicher Sprache" ist bei Softwaresystemen irreführend, denn Computer erfüllen keine der für den Erwerb der natürlichen Sprache notwendigen biologischen und sozialen Voraussetzungen. Sie sind die Produkte von Entwicklern und Ingenieuren und können daher immer nur die Kreativität und Erkenntnisfähigkeit ihrer Schöpfer widerspiegeln.

Wer an einem System mit sogenannter natürlichsprachlicher Schnittstelle arbeitet, merkt nach Terry Winograd und Fernando Flores, daß er zwar die Strukturen der natürlichen Sprache benutzt, das System aber diese Sprache nicht verarbeitet. Die Antworten des Systems spiegeln „eine spezielle Darstellung wider, die von einer Person oder einer Gruppe von Personen erzeugt wurde und die eine Blindheit verkörpert, der sich selbst die Entwickler nicht vollständig bewußt sein können."[35]

Da Computer also weder über Lebenserfahrung verfügen noch über gesunden Menschenverstand, kann die Bedeutung von Sätzen immer nur vor dem Hintergrund der im Rechner vom Menschen angelegten Strukturen und Objekte erschlossen werden. Von daher wird es immer Sätze geben, die für den Menschen einen Sinn ergeben, bei denen der Computer aber eine Fehlermeldung generiert, die zum Beispiel besagt, daß die gestellte Frage, bezogen auf den Inhalt der Datenbank, keinen Sinn ergibt. Was aber sinnvoll ist und was nicht, legen die Entwickler bei der Konzeption des Systems fest.

Unabhängig davon muß man berücksichtigen, daß für die Benutzung eines Softwaresystems weniger die Form der Interaktion entscheidend ist als vielmehr der Inhalt und der Umfang der mit dem System ausgetauschten Daten. So ist zum Beispiel Wimex nach Ansicht von James Fallows von zweifelhaftem Wert: Nach einer NATO-Studie müßten die Kommandeure des zentralen Kommandobunkers für Europa rund um die Uhr 790 Wörter pro Minute lesen, wollten sie mit dem Informationsstrom des Systems Schritt halten.[36] Darüber hinaus zeigte ein Systemtest am 6. November 1980, daß die Kommandeure während der angenommenen Krisenzeit über 12 Stunden keine wesentlichen Informationen über den Bereitschaftszustand ihrer Truppen erhielten.[37] Das heißt, das System liefert zu viele Daten, die zudem für die anstehenden Entscheidungen nicht relevant sind.

Je komplexer die Problemstellung ist, desto präziser und eindeutiger muß die Interaktion mit dem Computer sein. Nur so kann man sicherstellen, daß keine Mißverständnisse zwischen den Entwicklern und den Benutzern eines Systems auftreten und tatsächlich genau die Anweisungen ausgeführt werden, die jeweils beabsichtigt sind. Was ein Kommando bedeutet, welche Konsequenzen seine Ausführung hat usw., muß explizit festgelegt werden.

Aus diesem Grund bezeichnet der Linguist Geoffrey K. Pullum auch die in SCI beschriebenen Anwendungen als typische Fälle, in denen natürlichsprachliche Systeme gerade nicht eingesetzt werden sollten, denn:

> „Die Computern innewohnenden und unüberwindbaren Beschränkungen machen sie nur unzuverlässiger, wenn auf ihnen KI-Programme laufen, und dies um so mehr, wenn sie natürlichsprachliche Ein-/Ausgabemöglichkeiten besitzen. Ihre Reaktionen auf unerwartete, mehrdeutige Äußerungen, vor denen man sich bei linguistisch ungeschulten Benutzern nicht schützen kann, werden immer unvorhersehbar sein."[38]

Aber auch linguistisch geschulte Offiziere wären für Europa und speziell die Bundesrepublik im wahrsten Sinne des Wortes vernichtend, wenn sie ihre erlernten Fähigkeiten anwenden sollten. Denn was sprachlich passiert, muß nicht unbedingt mit den tatsächlichen Konsequenzen übereinstimmen. Beispielsweise müßten sie gemäß der bereits 1983 im NATO-Dokument MC 14/3 festgelegten Strategie der flexiblen Antwort im Krisenfall zuerst versuchen, den Gegner auf der Ebene zu bekämpfen, die er selbst gewählt hat (direct defense). Mißlingt dies, könnten sie zur Strategie der begrenzten Eskalation übergehen (deliberate escalation), die, im Falle ihres Scheiterns, als dritte Stufe den umfassenden Einsatz von Atomwaffen (general nuclear response) nach sich ziehen würde. „Aber die Dichte von atomaren Einheiten in der Bundesrepublik bedeutet, daß selbst begrenzte Angriffe massiv sein müßten."[39]

Es ist nicht die natürliche Sprache, die für sich allein Bedeutung hat; es ist die Verknüpfung mit unserem Handeln in der jeweiligen Situation. Das Wort erlangt Bedeutung, indem es zur Tat wird. Man kann zwar von einem *begrenzten Atomkrieg* reden, ob es aber möglich ist, einen solchen Krieg zu begrenzen, ist mehr als zweifelhaft. Die Verwendung natürlicher Sprache für KI-Systeme unter den gegebenen militärstrategischen Voraussetzungen kann diese „Blindheit" der Designer grundsätzlich nicht überwinden.

Dienst nach Vorschrift

Die Stärke des Menschen besteht darin, jeweils situationsbedingt flexibel und angemessen reagieren zu können und dabei auch – falls dies erforderlich sein sollte – gezielte Regelverletzungen zu begehen, wenn damit der vorgegebene Zweck erreicht werden kann; es kann sogar die Revision der ursprünglich gesetzten Ziele beinhalten. Seine Schwäche aber besteht darin, daß er komplexe Zusammenhänge und komplizierte Prozesse mit all ihren Wechselwirkungen und Nebeneffekten nicht auf Anhieb durchschauen und vorherbestimmen kann.

Bei allen Produkten des menschlichen Geistes gilt, daß man zu keinem Zeitpunkt absolute Gewißheit darüber haben kann, ob das entsprechende Produkt in all seinen Eigenschaften auch dem entspricht, was man von ihm erwartet, unabhängig davon, ob es sich dabei um mathematische Beweise[40], Bauwerke[41], Maschinen[42], Software[43] oder auch andere ingenieurmäßig hergestellte Produkte[44] handelt. Jeder Versuch, Entwicklungsfehler, Irrtümer und Mißverständnisse von vornherein grundsätzlich auszuschließen, ist zum Scheitern verurteilt. Gemäß Christopher Alexander ist die Gestaltung von Artefakten immer ein Prozeß der fortwährenden Reduktion von Fehlern.[45]

Eine gewisse Stabilität und damit Vertrauenswürdigkeit in ingenieurmäßig erstellte Produkte ist nur dann erreichbar,

wenn der Entwicklungszyklus von Entwurf, Einsatz und Auswertung unter einigermaßen stabilen Umgebungsbedingungen mehrfach durchlaufen werden kann. Denn je weniger die Entwickler mit der Problemstellung vertraut sind und je komplexer das System ist, desto weniger können sie im voraus die Zuverlässigkeit des Systems sicherstellen. Da diese Rahmenbedingungen aber im militärischen Bereich nicht gegeben sind, kann sich die Sicherheit des Einsatzes von Softwaresystemen nur aus der situationsbedingten Bewertung des Systemverhaltens durch den Menschen begründen.

Sicherheit ist dabei weiter gefaßt als Zuverlässigkeit. Gewöhnlich bezeichnet man ein System als zuverlässig, wenn es seine Funktionen über einen gewissen Zeitraum und unter festgelegten Umgebungsbedingungen so ausführt wie beabsichtigt. Sicherheit dagegen bezieht sich darauf, daß bestimmte Bedingungen, die zu Katastrophen oder Unfällen führen können, nicht eintreten, gleichgültig ob die beabsichtigte Funktion korrekt ausgeführt wird oder nicht.[46]

Wie SAGE und NORAD belegen, ist es nicht die Zuverlässigkeit der Technik, die eine Katastrophe bisher verhindert hat, sondern der flexible und verantwortungsbewußte Umgang des Menschen mit dieser Technik. Es gibt keine Garantie, daß Fehlalarme nicht passieren. Das gilt auch für den Einsatz von KI-Systemen.

Unsere relative Sicherheit begründet sich aus zwei Faktoren. Zum einen sind die Kommandeure in Zeiten relativer Entspannung in ihrer Erwartungshaltung auf einen Fehlalarm vorbereitet; die Zuverlässigkeit der Systeme hat also einen anderen Stellenwert als in Krisen- und Konfliktzeiten. Zum anderen scheuen sie davor zurück, die letzte Konsequenz zu ziehen. So ist es zum Beispiel schwierig, den Ernstfall zu „proben", wenn dabei der Einsatz von Atomwaffen vorgesehen ist. Unter der Annahme, daß es sich um einen wirklichen Krieg handele, weigerten sich die Offiziere häufig, den Einsatz von Atomwaffen auszulösen. Dies sei einer der Gründe, so Carl Builder von der Universität Santa Barbara, weshalb man Kriegssimulationen bräuchte.

Wie die Erfahrungen mit SAGE und NORAD außerdem belegen, sind geübte Anwender in der Lage, Fehler der Systementwickler so zu behandeln, daß trotzdem ein verwertbares Ergebnis zustande kommt. Dies setzt aber einen entsprechenden Handlungs- und Entscheidungsspielraum voraus. Trotzdem sind die Akteure häufig nicht in der Lage, für ihr Verhalten entsprechende Regeln anzugeben. Das heißt, sie wissen mehr als sie explizit beschreiben können. Daraus folgt, daß
1. Expertise nicht vollständig in Regeln erfaßt und beschrieben werden kann,
2. komplexe Systeme nur funktionieren, wenn die Benutzer mehr als nur das Vorgeschriebene tun und
3. verantwortliches Handeln meist Regelverstöße beinhaltet.
Verantwortung, Pflicht, Motivation und Intuition sind Aspekte, die weder durch Regeln noch durch Vorschriften eingefangen werden können.[47]

Diese Grundsätze haben auch für Organisationen ihre Gültigkeit: durch *Dienst nach Vorschrift* können sie an den Rand des Zusammenbruchs gebracht werden.

Softwaresysteme im allgemeinen wie auch KI-Systeme im besonderen machen aber grundsätzlich nur *Dienst nach Vorschrift.*

Das kognitive Perpetuum mobile

Die Erwartungen an eine der menschlichen Intelligenz ebenbürtige oder gar über-

legene künstliche Intelligenz sind reine Wunschvorstellungen. Der Traum, daß der Mensch etwas hervorbringt, was nicht nur in Teilbereichen, sondern im Ganzen besser ist als er selbst, ist nichts anderes als die Suche nach einem *kognitiven Perpetuum mobile.*[48]

Maschinen und Softwaresysteme verkörpern immer nur den Kenntnisstand ihrer menschlichen Schöpfer. Es gibt keine Maschinenevolution; Maschinen leben nicht, sie irren sich nicht und lernen nicht aus Fehlern. Eine Maschine wird von Menschen für einen bestimmten Zweck entworfen, gebaut und benutzt. Ihr Wert besteht darin, daß sie sich auf eine wohldefinierte und voraussagbare Art und Weise gemäß des gesetzten Zweckes verhält.

Treten Fehler auf, setzen sich Entwickler und Konstrukteure hin und analysieren den Fehler, um ihn zu beheben. Das Ergebnis kann eine verbesserte Konstruktion oder eine neue Maschine sein. Sie können aber auch nach ganz anderen Wegen und Möglichkeiten suchen, um das gewünschte Ziel zu erreichen.

Aufgrund der bisherigen Erfahrungen im militärischen Bereich kann der projektierte Einsatz von KI-Systemen nur als *„ballistische"* Problemlösungsstrategie charakterisiert werden. Seit Pearl Harbour lautet die Hypothese, daß wir nur durch den Einsatz zunehmend komplexer werdender Softwaresysteme unsere militärische Sicherheit erhöhen können. KI-Systeme können aber nur dann unsere Sicherheit erhöhen, wenn wir bezüglich unserer Entscheidungsmöglichkeiten insgesamt weniger abhängig von der Technik werden. Dies gilt insbesondere unter der Voraussetzung, daß ein Krieg im Schatten der Atombombe geführt werden wird.

Die Verantwortung trägt der Mensch. Er kann aber nur verantwortlich handeln, wenn er auch Wahlmöglichkeiten hat, wenn er entscheiden kann. Hier kommt das *Foerstersche Theorem* zur Anwendung, das besagt, daß wir nur die prinzipiell unentscheidbaren Fragen entscheiden können, denn die entscheidbaren Fragen, die sich mit zwingender Logik ergeben, sind ja durch die vorgegebenen „Spielregeln" bereits entschieden. „Bei prinzipiell unentscheidbaren Fragen haben wir jeden Zwang – sogar den der Logik – abgeschüttelt, und haben mit der gewonnenen Freiheit auch die Verantwortung der Entscheidung übernommen."[49]

Mit der Übernahme von Entscheidungen vertauschen wir die Rolle des „unbeteiligten Beschreibers" mit der des „mitfühlenden Beteiligten", wobei das Fühlen hier wörtlich zu nehmen ist. Ängste, Wünsche, Hoffnungen und Ideale sind ein wesentlicher Bestandteil unserer Intelligenz, genauso wie Schmerz und Leid; ohne sie wäre sie sinnlos.

Krieg ist kein kognitives Ereignis; er ist die Zerstörung von Leben. Wenn wir das mitfühlen, können wir Verantwortung übernehmen. Anstatt die Technik zu gestalten, um einen Atomkrieg führbar zu machen, müssen wir nach Möglichkeiten suchen, Krieg zu vermeiden. Dabei müssen wir uns selbst wie auch unseren potentiellen Gegnern mehr vertrauen als den von uns geschaffenen Artefakten.

Soziale Systeme sind vertrauensbasierte Systeme, wissen können wir etwas immer nur im nachhinein. Wir können unsere Verantwortung nicht als vorgefertigte Entscheidungen in Maschinen verlegen – auch nicht in angeblich intelligente, denn KI-Systeme sind unverantwortlich; sie sind bestenfalls berechnend.

[1]Vgl. den geschichtlichen Abriß in *Edwards, P.N.:* Artificial Intelligence and High Technology War. The Perspective of the Formal Machine. Silicon Valley Research Group, Working Paper No. 6, Santa Cruz 1986.

- [2]Vgl. u.a. den Literaturüberblick in *Keil-Slawik, R.:* Von der Mechanisierung des Kopfes zur Ökologie des Geistes. Ein Literaturüberblick zu anthropologischen Aspekten menschlicher und künstlicher Intelligenz. Erscheint in: M. Stöhr/H. Wendt (Hrsg.): Menschliche und künstliche Intelligenz. Frankfurt 1990. - [3]*Neisser, U.:* Kognition und Wirklichkeit. Prinzipien und Implikationen der kognitiven Psychologie. Stuttgart 1979. S. 16f. - [4]*Parnas, D.L.:* Software Wars. Kursbuch Nr. 83. Berlin 1986. S. 53. - [5]*Lehmann, M.M.:* Programs, Life Cycles and Laws of Software Evolution. Proceedings of the IEEE; Vol. 86; September 1980. S. 1061. - [6]*Naur, P.:* Programming as Theory Building. Microprocessing and Microprogramming 15. 1985. S. 258. - [7]Hier betrachte ich nur eine Entwicklungslinie. Weitere Beispiele werden behandelt in *Keil, R./Reisin, F.-M.:* Informatik in militärischen Diensten - Informatiker in sozialer Verantwortung. WSI-Mitteilungen; 37. Jahrgang, Nr. 5, Mai 1984; sowie *Keil-Slawik, R.:* Von der Feuertafel zum Kampfroboter - Die Entwicklungsgeschichte des Computers. In: J. Bickenbach/R. Keil-Slawik M. Löwe/R. Wilhelm (Hrsg.): Militarisierte Informatik. Schriftenreihe Wissenschaft und Frieden, Nr. 4, FIFF Berlin. Marburg - Berlin - Münster 1985; und *Keil-Slawik, R.:* Militärtechnologische Interessen und Computerentwicklung. LogIn Informatik und Computereinsatz in der Schule; 5. Jahrg., Heft 4, 1985. - [8]Vgl. *Fallows, J.:* National Defense. New York 1981. S. 59. - [9]*Bracken, P.:* The Command and Control of Nuclear Forces. New Haven - London 1983. S. 12. - [10]*Emerson, S.:* Some Perspectives on Networks - Past, Present, and Future. Introduction to an article of P. Baran; The Journal of Community Communications; Vol. 4, No. 1 (1982). S. 30. - [11]Siehe *Bracken, P.* (Anm. 9), S. 34 u. 51. - [12]Ausführliche Darstellung in *Blasius, K.H. Siekmann J.H.:* Computergestützte Frühwarn- und Entscheidungssysteme. Informatik-Spektrum; Band 10, Heft 1 (1987); und Borning, A.: Computer System Reliability and Nuclear War. Communications of the ACM; Vol. 30, No. 2 (1987). - [13]*Hart, G./Goldwater, B:* Recent False Alerts from the Nation's Missile Attack Warning System. U.S. Government Printing Office, Washington D.C., October 1980. S. 13. - [14]Nach *Arkin, W.M./Fieldhouse, R.W.:* Nuclear Battlefields. Global Links in the Arms Race. Cambridge (Mass.) 1985. S. 89, 93 u. 96. - [15]*Bracken, P.* (Anm. 9), S. 88f. - [16]Nach *Arkin, W.M./Fieldhouse, R.W.* (Anm. 14), S. 99. - [17]*Reisin, F.-M./Wilhelm, R.:* Präzisionsgelenkte Munition - Ein Bruch mit allen Konventionen. In: Bickenbach/Keil-Slawik/Löwe/Wilhelm (Anm. 7), S. 113; sowie *Pringle, P./Arkin, W.:* S.I.O.P. The Secret U.S. Plan for Nuclear War. New York - London 1983. S. 87f.; sowie *Bracken, P.* (Anm. 9), S. 88f. - [18]DoD: Software Development for Adaptable, Reliable Systems (STARS) Program Strategy. Department of Defense; ACM SIGSOFT Software Engineering Notes; Vol. 8, No. 2; April 1983. S. 67. - [19]Im Deutschen auch als Aufklärungs- und Führungssysteme bezeichnet. - [20] Zitiert nach *Crumley, D.V.:* Konzepte für den Einsatz von Robotern mit Künstlicher Intelligenz durch das Heer im 21. Jahrhundert. In: Bickenbach/Keil-Slawik/Löwe/Wilhelm (Anm. 7), S. 134. - [21]*Bracken, P.* (Anm. 9), S. 241. - [22]*Fletcher, J.C./McMillan, B.:* Report of the Study on Eliminating the Threat Posed by Nuclear Ballistic Missiles. Vol. V; Battle Management, Communications, and Data Processing; Contract MDA 90384 C 0031, Task T-3-191; February 1984. - [23]Siehe „KI und die nationale Verteidigung". In *Feigenbaum, E.A./McCorduck, P.:* Die Fünfte Computer-Generation. Basel - Boston - Stuttgart 1984. S. 286-293. - [24]Secretary of Defense: Report of the Defense Science Board Task Force on Military Applications of New-Generation Computing Technologies. Office of the Under Secretary of Defense for Research and Engineering, Washington D.C., December 1984. S. 10. - [25]DARPA: The DARPA Program. A Proposed Strategic Plan for the Development and Application of Next-Generation Technology for the Military. In: E.A. Torrero (Ed.): Next Generation Computer. Spectrum Series. New York 1985. S. 148f. - [26]*Parnas, D.L.* (Anm. 4), S. 63f. - [27]*Borning, A.* (Anm. 12), S. 119. - [28]*Jacky, J.:* The „Star Wars" Defense Won't Compute. The Atlantic; Vol. 225, No. 6; June 1985. S. 25. - [29]*Davis, P.K.:* Applying Artificial Intelligence Techniques to Strategic-Level Gaming and Simulation. In: E.T.I. Oren/B.P. Ziegler (Eds.): Modelling and Simulation Methodology in the Artificial Intelligence Era. Amsterdam - New York - Oxford 1986. - [30]*Struß, P.:* Gibt es Expertensysteme? Computer Magazin; Jahrg. 15, Nr. 5 (1986). S. 53 - [31]*Dörner, D.:* Die Logik des Mißlingens. Reinbek 1989. S. 40; vgl. auch S. 267. - [32]Secretary of Defense (Anm. 24), S. 10. - [33]Ebenda, S. XIII. - [34]*Smith, B. C.:* Limits of Correctness in Computers. Stanford University, Center for the Study of Language and Information; Report No. CLSI-85-36; Palo Alto (Ca.), October 1985. S. 8. - [35]*Winograd, T./Flores, F.:* Understanding Computers and Cognition. A new Foundation for Design. Norwood (N. J.) 1986. S. 124. - [36]*Fallows, J.:* National Defense. New York 1981. S. 52. - [37]*Pringle, P./Arkin, W.* (Anm. 17), S. 150 f. - [38]*Pullum, G.K.:* Natural Language Interfaces and Strategic Computing. AI & Society; Vol. 1, No. 1 (1987). - [39]*Pringle, P./Arkin, W.* (Anm. 17), S. 97. - [40]*Lakatos, I.:* Proofs and Refutations. The Logic of Mathematical Discovery. Cambridge - London - New York - New Rochelle - Melbourne - Sydney 1984; vgl. auch *DeMillo, R./Lipton, R./Perlis, A.:* Social Processes and Proofs of Theorems and Programs. Communications of the ACM; Vol. 22,

No. 5 (1979). – [41]Vgl. *Alexander, C.:* Notes on the Synthesis of Form. Cambridge (Mass.) – London 1964. – [42]Vgl. *Browning, R. L.:* The Loss Rate Concept in Safety Engineering. New York 1980. – [43]*Lehmann, M. M.* (Anm. 5) u. *Winograd, T./Flores, F.* (Anm. 35). – [44]Vgl. *Petroski, H.:* To Engineer is Human. The Role of Failure in Successful Design. London 1985. – [45]*Alexander, C.* (Anm. 41), S. 15–27 u. S. 102. – [46]*Leveson, N. G.:* Software Safety: Why, What and How. Computing Surveys; Vol. 18, No. 2; June 1986. S. 135. – [47]Siehe dazu auch *Dreyfus, H. L./Dreyfus, S. E.:* Künstliche Intelligenz. Von den Grenzen der Denkmaschine und dem Wert der Intuition. Reinbek 1987. – [48]Vgl. dazu die Ausführungen in *Keil-Slawik, R.:* Das kognitive Perpetuum mobile. Die Rolle von Computern mit künstlicher Intelligenz in der militärtechnologischen Entwicklung. In: G. Bechmann/W. Rammert (Hrsg.): Technik und Gesellschaft. Jahrbuch 5. Frankfurt 1989. – [49]*Von Foerster, H.:* Wahrnehmung. In: J. Baudrillard/H. Böhringer/V. Flusser/H. von Foerster/F. Kittler/P. Weibel: Philosophien der neuen Technologie. Hrsg. von Ars Electronica Berlin 1989. S. 30.