

Computertechnik – falsche Versprechen und enttäuschte Erwartungen

Warum gescheiterte Organisationsprojekte den Entwicklungspfad pflastern

Peter Brödner

1 Einführung: Maschinen für Kopfarbeit

In diesem Beitrag wird der Frage nachgegangen, warum Projekte der Nutzung von Computertechnik in Prozessen kooperativer Wissensarbeit so häufig scheitern. Als Ursache dafür, dass die hohen Erwartungen an den Technikeinsatz so oft enttäuscht werden, ist vornehmlich mangelndes Verständnis der tatsächlichen Funktionsweise von Computersystemen und ihrer Einsatzbedingungen zu vermuten. Es zeigt sich u.a. in den vielen unzutreffenden und auch wechselnden Benennungen dieser Technik, in den Schwierigkeiten, die bei Entwurf und Konstruktion von Systemfunktionen auftreten, sowie in den oft vernachlässigten Bemühungen um Aneignung derselben für den praktischen Gebrauch.

In der Tat wurde mit dieser neuen Technik unvergleichlich Neuland beschritten: Statt in Arbeitsprozessen wie bislang Kräfte, Stoffe und Effekte der Natur maschinell zu nutzen, werden nun berechenbare Vorgänge zeichenbasierter Kooperation logisch analysiert und formal modelliert. Statt thermo- bzw. elektrodynamischer und mechanischer Wirkungen als Funktionen zur Krafterzeugung und -übertragung werden algorithmische Funktionen zur Verarbeitung, Speicherung und Übertragung von Daten aktiviert. Statt wie zuvor körperliche Arbeit wird nun kognitive Arbeit maschinisiert. Entsprechend hat die Computertechnik vom Beginn ihrer tatsächlichen Verwirklichung an, nach einer langen Vorgeschichte logisch-konzeptioneller Entwicklung, Menschen fasziniert und ihre Phantasie beflügelt.

Es wundert daher nicht, dass die neue Technik große Erwartungen weckte und sich rasch Mythen um sie rankten: McCulloch & Pitts stellten bereits 1943 ein logisches Kalkül vor, um das Verhalten von Nervensystemen zu beschreiben, und 1949 erschien davon inspiriert das Buch »*Giant Brains or Machines that Think*« von E.C. Berkeley. 1950 publizierte Turing seinen ebenso berühmten wie fragwürdigen Test (Turing 1950), mit dem er die Frage zu klären hoffte, ob eine Maschine »denken« kann – entgegen seinen im Denkmodell der »Turingmaschine« präzisierten Erkenntnissen über Algorithmen und berechenbare Funktionen (Turing 1936). Und im gleichen Jahr erschien das Time Magazine mit der Titelgeschichte »*Can Man Build a Superman?*«. Seither jagt eine Welle großspurig angekündigter »Künstlicher Intelligenz (KI)« (McCarthy et al. 1955) die nächste: In den 1960er Jahren das logische »Theorem Proving« und »General Problem Solving« (Newell/Shaw/Simon 1959), in den 1980er Jahren die sog. »symbolische KI« auf Basis von »Physical Symbol Systems« (Newell 1980) vor allem in Gestalt wissensbasierter und Expertensysteme im Umfeld von Computer-Integrated Manufacturing (CIM), die letztlich an den Grenzen der Explizierbarkeit von Erfahrungswissen scheiterten (vgl. die kritische Übersicht in Brödner 1997). Schließlich wird derzeit versucht, die Grenzen mit hochgejubelter subsymbolischer bzw. »konnektionistischer KI« auf Basis komplex strukturierter künstlicher neuronaler Netze (KNN) zu umgehen, zu denen auch die »Large Language Models (LLM)« gehören.

Zwar wurden diese an die Computertechnik geknüpften großen Erwartungen schon früh durch philosophisch fundierte Kritiker gründlich in Frage gestellt (vgl. etwa Dreyfus 1979, Searle 1980), jedoch sind die Protagonisten von ihren »KI«-Ansätzen so eingenommen, dass sie von der Kritik unbeeindruckt einfach weitermachen und erst nachlassen, wenn die anfängliche Euphorie infolge des Ausbleibens praktischen Nutzens gebrochen ist und öffentliches Interesse erlahmt (diese Wellen-Dynamik lässt sich durch den Analyseansatz der »promising technology« gut erklären; Hirsch-Kreinsen 2023). Das den »KI«-Ansätzen zugrunde liegende, in den Kognitionswissenschaften allgemein dominante Weltbild der »computational theory of mind« (Rescorla 2024) wurzelt tief in der rationalistischen Tradition und gipfelt letztlich in der technokratischen Dystopie des Transhumanismus. In diesem Weltbild wird menschlicher Verstand schlicht mit Logik, werden Wahrnehmen, Denken und Vorstellen mit Berechnen gleichgesetzt: »computation and cognition« (Pylyshyn 1984). Menschen rechnen zwar gelegentlich wie Maschinen, etwa bei schrittweiser Durchführung eines Berechnungsverfahrens oder eines logischen Beweises; ihr intuitives und kreatives Denken, auch ihr erfahrungsbasiertes Urteilen und Handeln reichen aber weit über Berechnen hinaus (Weizenbaum 1977, Fetzer 2001).

Dieser Perspektive wird hier ein Weltbild gegenübergestellt, das in der noch weiter zurückreichenden humanistischen Tradition wurzelt. Ihm zufolge wird die kulturelle Evolution als Fortsetzung der natürlichen mit anderen Mitteln aufgefasst. Sie wird durch uns Menschen – selber ein Produkt natürlicher Evolution – durch unsere gegenständliche bewusste Tätigkeit in der Natur vorangebracht. So vollzieht sich die kulturelle Evolution, angetrieben durch jeweils dominante Interessen, auf Basis kollektiv geteilter Intentionalität in Praktiken der Herstellung und des Gebrauchs von Werkzeugen als Mitteln zweckmäßiger Nutzung von Naturstoffen und -kräften (Technik: vermittelt Wirkungen) sowie von Zeichen als Mitteln sozialer Interaktion, Kommunikation und Reflexion (Sprache: vermittelt Bedeutungen). Der Zeichengebrauch der Sprache konstituiert die soziale Welt der Bedeutungen (anstelle physischer Wirkungen) und ermöglicht kooperative Praktiken, in denen Bedeutungen von Zeichen veränderlich (re-)produziert, institutionelle Tatsachen deklarativ geschaffen (Searle 2012) und Eigenschaften von Dingen oder Vorgängen erschlossen werden (»Prädikate«; Taylor 2017). Für Zeichenprozesse der Kommunikation und Kooperation bietet Computertechnik heute maschinell operierende Werkzeug- und Medienfunktionen, die jedoch begrenzt bleiben.

Vor dem Hintergrund dieser Perspektiven ist ein nüchterner Blick auf inzwischen rund acht Dekaden realer Entwicklung der Computertechnik geboten. Ab den 1970er Jahren hat sie (von »KI«-Systemen abgesehen) schnell wachsende Investitionen und rasche Verbreitung erfahren, anfangs in schon zuvor formal strukturierten Prozessen von Wirtschaft und Verwaltung, ab Mitte der 1980er Jahre dann auch im persönlichen Gebrauch. Anwendungsfelder wurden zahlreicher und auch überwiegend höchst erfolgreich erschlossen: bei der Steuerung von Maschinen, Geräten und physischen Prozessen, als CAD-, EDM- und FEM-Systeme in Entwicklung und Konstruktion oder als ERP-Systeme in Produktion und Logistik, bei Tabellenkalkulation, Dokumentedition und bei medialen Diensten auf Basis des Internets.

Währenddessen hat die Rechenleistung der Hardware mit einer Verdoppelung ca. alle zwei Jahre exponentiell zugenommen. In der Softwaretechnik ist die Entwicklung dagegen

seit den 1990er Jahren unter der Last selbst verschuldeter Komplexität und massiver Rebound-Effekte weitgehend erlahmt und erschöpft sich heute vorwiegend in der Wartung und Erneuerung bestehender Systeme (Wirth 1995). Daran ändert auch der Hype um ChatGPT nichts, denn er gaukelt Fortschritt nur vor, beruht er doch auf lange bekannten mathematischen Verfahren, ausgeführt auf nun leistungsstarker Hardware. Durch Computereinsatz oftmals erwartete und wegen Jobverlusten auch befürchtete Produktivitätssteigerungen bleiben weitgehend aus. Gesamtwirtschaftlich sind sie nicht erkennbar und einzelwirtschaftlich werden sie nur in wenigen besonderen Fällen verwirklicht. Ferner gibt es im Vergleich zu Feldern herkömmlichen Technikeinsatzes bei der Einrichtung computerunterstützter Wissensarbeit in Organisationen von Beginn an eine auffallend hohe Zahl gescheiterter Projekte (vgl. Brödner 2023: Kap. 8).

Diese persistente Diskrepanz zwischen erhofften und tatsächlich realisierten Ergebnissen verweist nochmals auf gewichtige Defizite im Verständnis der Funktionsweise von Computern, der Grundlagen maschineller Ausführung logisch-arithmetischer Operationen und deren praktischer Verwendung in Prozessen kooperativer Wissensarbeit. Eingedenk verbreiteter unzureichender Technikmündigkeit werden daher im folgenden zunächst schrittweise grundlegende Einsichten in Aufbau und Funktionsweise von Computern entwickelt, um die Kluft zwischen Anspruch und Wirklichkeit zu erklären und im Ergebnis eine besser gelingende Gestaltung computerunterstützter Wissensarbeit in Organisationen anzuleiten. Am Ende ermöglichen diese Einsichten auch eine realistische Einschätzung jüngster »KI«-Entwicklungen.

2 Grundsätzliches: Computertechnik und Zeichenprozesse

2.1 Ein hinreichend entwickelter Zeichenbegriff als Fundament

Sollen Computersysteme in Zeichenprozessen kooperativer Arbeit sinnvoll eingesetzt und Nutzen stiftend verwendet, sodann auch deren Wirkungen angemessen beurteilt werden, ist es unerlässlich, deren Funktionsweise wenigstens in ihren Grundzügen zu verstehen. Dies erscheint umso notwendiger, als in vielen Diskursen über Computertechnik mit unzutreffenden bis irreführenden Begriffen und Aussagen operiert wird.

Als Grundlage der Analyse und Schlüssel zum Verständnis der Vorgänge maschinellen Rechnens sowie der Interaktion zwischen Menschen und Computern erweist sich der triadische Zeichenbegriff, der von dem amerikanischen pragmatistischen Logiker C. S. Peirce – zeitgleich und unabhängig auch von dem deutschen Logiker G. Frege – in den 1870er Jahren entwickelt wurde (Peirce 1983, Frege 1879). Beide haben im Zusammenhang damit als erste eine formale Prädikatenlogik erster Stufe vorgestellt, die für die theoretischen Grundlagen der Computertechnik grundlegende Bedeutung hat. Sie standen dabei auf den Schultern des britischen Mathematikers George Boole, der mit seiner »Analysis of Logic« (1847) die seit der Antike entwickelte Aussagenlogik durchgängig formalisierte und damit den Grundstein für die später axiomatisierte Boolesche Algebra legte, auf der die Funktionsweise binärer Schaltsysteme als heutige Computer-Hardware beruht. Wesentlich dabei ist, dass sich mittels der dafür geltenden Axiome und daraus abgeleiteten Regeln zum einen mit nur drei logischen Grundoperationen – der Konjunktion (UND-Verknüpfung), der Disjunktion (ODER-Verknüpfung) und der Negation (NICHT) – jeder beliebig zusammengesetzte aussagenlogische Ausdruck

erzeugen lässt und zum anderen auch die arithmetische Operation der Addition (und mit ihr alle höheren arithmetischen Operationen) auf getakteten binären Schaltsystemen physisch vollzogen werden kann (genauere Einzelheiten z.B. in Bauer/Goos 1971 Teil 1, Kap. 4; Brödner 2023: 157-166).

Ein *Zeichen* ist den Einsichten von Peirce und Frege zufolge kein ›Ding‹ (wie oft unterstellt), sondern eine in Prozessen sozialer Interaktion (re-)produzierte dreistellige Relation zwischen den folgenden Entitäten (vgl. Abb. 1; leider verwenden beide dafür unterschiedliche Benennungen, weswegen hier aus praktischen Gründen auf Peirce (1983) zurückgegriffen wird; vgl. auch Eco 1991, Falkenberg et al. 1998, Nake 1997, 2001):

- (1) dem Repräsentamen *R*, dem physischen Zeichenträger oder -körper, ein als Zeichen gedeuteter, wohlgeformter Gegenstand oder Vorgang (z.B. Worte eines Alphabets),
- (2) dem Objekt *O*, dem damit bezeichneten (ggf. nur gedachten oder auch abwesenden) Gegenstand oder Vorgang,
- (3) dem Interpretanten *I*, der sozial konstruierten Bedeutung, die ein Interpret einem Paar (*R*, *O*), der Bezeichnung, situativ zuschreibt (Begriff).

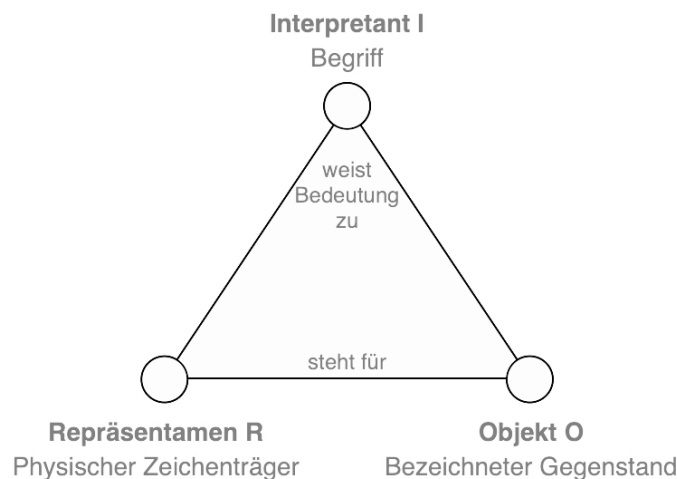


Abb. 1: Zeichen als dreistellige Relation (eigene Darstellung)

Ein Zeichen ist mithin die dreistellige Relation $((R - O) - I)$: »Ein Zeichen ist etwas, das für jemanden in einer bestimmten Hinsicht oder Fähigkeit für etwas steht« (Peirce 1960: CP 2.228). Der Begriff ist rekursiv: Ein Interpretant kann als gedachtes Objekt selbst wieder durch Zeichen benannt und gedeutet werden, was zu unendlichen Zeichenprozessen (»Semiosen«) führt.

In der Semiotik üblich ist dabei die Unterscheidung von drei Ebenen der Analyse von Zeichenprozessen (Eco 1991, Morris 1975; vgl. auch Ogden/Richards 1989):

- *Syntaktik* beschreibt die auf *R* reduzierten Aspekte von Zeichenprozessen: womit wird bezeichnet? (→ Daten in Gestalt bedeutungsloser Alphabetzeichen, durch Signale physisch repräsentierbar);
- *Semantik* beschreibt deren auf die Relation (*R* – *O*), die Bezeichnung, reduzierten Aspekte: was wird mit *R* bezeichnet? (→ Information i.S. von ›steht für‹);
- *Pragmatik* beschreibt die sozio-kulturellen Aspekte von Zeichenprozessen: wie wird Bezeichnetes in einem Handlungskontext gedeutet? (→ explizit begriffliches Wissen).

Der Clou dieses avancierten Zeichenbegriffs ist, dass damit zum einen auch über nicht-gegenwärtige sowie über durchweg nur gedachte Objekte wie etwa in der Mathematik gesprochen werden kann und zum anderen eine logische Verbindung zwischen der physischen Welt der Wirkungen und der sozialen Welt der Bedeutungen, somit auch der bewussten tätigen Auseinandersetzung des Menschen mit der Natur, hergestellt wird. Zudem werden damit den postmodernen Dekonstruktion geschuldeten »Narrativen«, der vermeintlichen Beliebigkeit von Aussagen und dem resultierenden Realitätsverlust, strikte Gültigkeitsgrenzen gesetzt, etwa bei Aussagen über mathematische Objekte oder Naturgegenstände. Erstere müssen geltenden Axiomen und formal abgeleiteten Sätzen genügen, letztere im Einklang mit den für sie geltenden Naturgesetzen stehen, ansonsten sind sie schlicht falsch (für diesbezügliche Prüfungen liefern die jeweiligen Wissenschaften anerkannte Methoden der Erkenntnissicherung).

Mit dem Zeichengebrauch hat sich der menschliche Verstand neben den tatsächlichen Naturvorgängen und seiner kooperativen Tätigkeit in ihnen eine zweite Ebene der Realität geschaffen, die Metaebene ihrer Beschreibung – eine Art ›Verdoppelung‹ der Welt in Zeichen. Das hat weitreichende Folgen: Sprache kann selbst zum Gegenstand von Sprache, damit jede natürliche Sprache auch Metasprache ihrer selbst werden, mit der Folge logischer Irrtümer durch Selbstreferenz (vgl. die Antinomien von Epimenides oder Russell) sowie der Möglichkeit der Täuschung und der Selbsttäuschung. So hat Umberto Eco in seinem Lehrbuch »Semiotik« (1991) eine gleichfalls allgemein gültige, freilich paradox erscheinende Zeichen-Definition gegeben: »Zeichen ist ... alles, was man zum Lügen verwenden kann« (26). Reichlich Anschauungsmaterial für diese ebenso irritierende wie zutreffende Definition bieten etwa Lebenslegenden von Geheimagenten, Verwechslungskomödien à la Molière oder auch die Illusionen über »KI« (s.u.).

Bedauerlicherweise leidet die Computertechnik – eine dem angelsächsischen ›Computer‹ bzw. ›Computing Science‹ entlehnte Bezeichnung – von Beginn an unter abwegigen Benennungen. So wird infolge des nahezu gleichzeitigen Entstehens von Kybernetik und Computertechnik während des Zweiten Weltkriegs bis heute von ›Informationstechnik‹, von ›Informationsverarbeitung‹ in und ›Informatisierung‹ mit Computern, sogar von ›Informatik‹ als Disziplinbezeichnung gesprochen. ›Information‹ steht aber für mindestens drei ganz unterschiedliche und unvereinbare Begriffe: 1. dem für die signaltechnische Übertragung von Nachrichten rein syntaktisch auf einem Alphabet definierten Informationsmaß (explizit ohne jeden Bezug zu Bedeutung; Shannon 1948), 2. dem sozialwissenschaftlichen Begriff von Information als »any difference that makes a difference« (Bateson 1980: 250) und 3. dem umgangssprachlichen Begriff von Information als Inhalt einer Auskunft, Mitteilung oder Belehrung. In der Computertechnik werden zumindest die ersten beiden Informationsbegriffe zugleich gebraucht, sodass niemand recht weiß, wovon jeweils die Rede ist (vgl. Brödner 2014). Als wissenschaftlich-analytischer Begriff ist ›Information‹ daher unbrauchbar, Quelle unklarer Vorstellungen und enttäuschter Erwartungen als Gründe des Scheiterns.

Gänzlich unangemessen ist auch die neuerdings in Mode gekommene Bezeichnung ›Digitalisierung‹ für den Einsatz von Computertechnik in sozialen Praktiken. Eine Digitalisierung gesellschaftlicher Prozesse fand tatsächlich um ca. 3000 v.u.Z. mit der Erfindung diskreter Zahl- und Schriftzeichen durch eine Funktionselite von Buchhaltern

im Auftrag der damaligen Herrscher statt, nahezu simultan in allen großen Flusskulturen, zeitgleich auch mit der ersten Urbanisierung. Das war aus der Not geboren, ein großes, rasch anwachsendes gesellschaftliches Mehrprodukt verwalten zu müssen, was zuvor übliche mündliche Überlieferungen und Gedächtnisleistungen deutlich überforderte. Die Bezeichnung basiert auf ›digitus‹, dem Finger als erstem Zähl- und Rechenmittel. Damit konnten, wie Schriftzeugnisse belegen, stoffliche Vorgänge der Bewirtschaftung und Verwaltung von Vorräten oder Bautätigkeiten genau beschrieben, symbolisch nachvollzogen oder geplant werden (Vermessung von Feldern, Flächen- und Volumenberechnung (»Geometrie«), Bestandsführung, Besteuerung, Geld und Kredit etc.; Ifrah 1989). Mittels der Zeichen wurden damals im Kern die gleichen Aufgaben der Berechnung bewältigt wie durch heutige ERP-Systeme, mit dem entscheidenden Unterschied freilich, dass letztere sie programmgesteuert automatisch ausführen. Dafür wäre ›algorithmische Revolution‹ eine sachgerechte Bezeichnung, da sie das eigentlich Neue betont. Die Begriffe ›Algorithmus‹ und ›berechenbare Funktion‹ sind durch das Denkmodell der Turingmaschine exakt definiert, damit dargestellte Berechnungsverfahren unterliegen aber unüberwindlichen logischen Grenzen (Hermes 1971).

Der in der Computertechnik so häufige, in diesem Ausmaß in anderen Technikfeldern nicht übliche Benennungs- und Begriffswirrwarr zeugt weithin von mangelhafter Mündigkeit im Verständnis von Computersystemen, der Vorgänge in ihrem Innern und der Prozesse ihres Entwurfs und Gebrauchs durch Interaktion mit ihnen. Das wird im Weiteren als wichtige Ursache des Scheiterns auf Basis des triadischen Zeichenbegriffs näher ausgeleuchtet.

2.2 Einblicke in Kommunikation und Mensch-Computer-Interaktion

Infolge der Fixierung auf ›Information‹ liegen einem verbreiteten Verständnis von Prozessen menschlicher Kommunikation bis heute Vorstellungen zugrunde, die Kommunikationsprozesse als Austausch von ›Informationen‹ betrachten. Das wird aber den tatsächlichen Vorgängen nicht gerecht. Mittels des in Abb. 1 dargestellten triadischen Zeichenbegriffs lassen sich die Vorgänge stattdessen wie folgt analysieren: Kooperativ handelnde soziale Akteure stehen in einem gemeinsamen Handlungskontext geteilter Aufmerksamkeit, in dem sie mit Zeichen aufgrund der ihnen zugeschriebenen Bedeutungen umgehen, die sich ihrerseits aus deren (veränderlicher) Interpretation im Zuge koordinierter sozialer Interaktionen ergeben. Indem A per intendierter Expression in einem Medium als Zeichen deutbare Gestalten formt, ist B imstande, sich durch deren empathische Wahrnehmung und Interpretation zu informieren und umgekehrt. Bildung und Deutung der Zeichen sind jeweils schöpferische, zueinander inverse Tätigkeiten der Kommunikation. Intention der Zeichenbildung (Gemeintes) und Ergebnis der Deutung (Verstandenes) stimmen oft zunächst nicht überein – Kommunikation kann misslingen –, durch mehrfaches Durchlaufen der Expressions- und Interpretationsschleifen im Kontext geteilter Aufmerksamkeit können sie sich aber annähern. Kommunikation ist mithin Abstimmung intentionaler Zustände zwischen A und B und nicht die Übertragung von ›Information‹ (Falkenberg et al. 1998, Tomasello 2009, Maturana & Varela 1987).

Mit Blick auf Entwurf und Gebrauch von Computersystemen sowie den interaktiven Umgang mit ihnen ermöglicht der triadische Zeichenbegriff – im Unterschied zur unscharfen ›Information‹ – ebenfalls einen deutlich sachgerechteren analytischen Zugang.

Damit lässt sich über die in beiden Welten verfügbaren Zeichenträger R eine logische Verbindung zwischen der Computertechnik in der physischen Welt der Wirkungen und kooperativer Praxis in der sozialen Welt der Bedeutungen herstellen (vgl. Abb. 2).

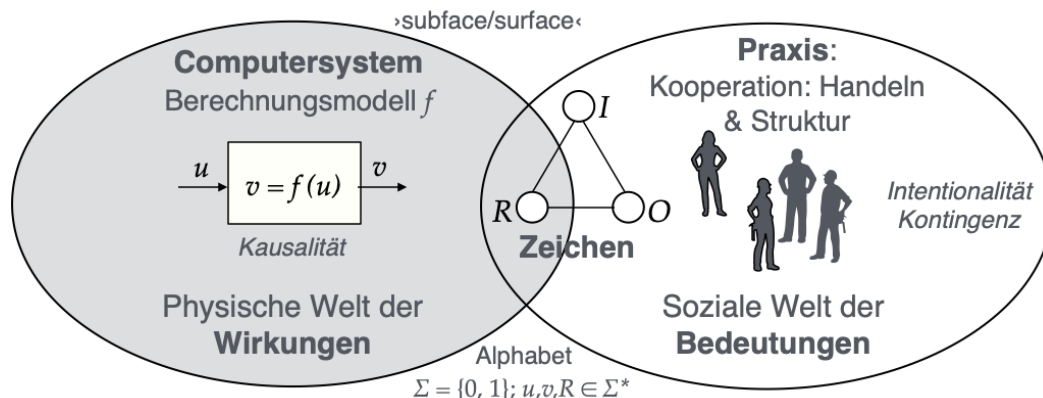


Abb. 2: Logische Verbindung von Physik und Semantik mittels Zeichen (eigene Darstellung)

Die wichtigste Frage, die sich dabei stellt, ist, wie das Berechnungsmodell f analytisch aus einer gegebenen Praxis für eine künftig gewünschte Praxis passend bestimmt werden kann. Um die Funktionen des Computersystems zweckmäßig für den praktischen Gebrauch entwerfen bzw. ein existierendes Standardsystem (z.B. ein ERP-System) entsprechend konfigurieren zu können, müssen zunächst künftig erwünschte, auf Nutzung seiner Funktionen zielende Praktiken kooperativer Arbeit analysiert und in mehreren Stufen der Abstraktion und Reduktion modelliert werden. Die in dieser Perspektive sachgerecht vorzunehmende Analyse und Explikation der den sozialen Praktiken zugrunde liegenden Zeichenprozesse liefert explizites begriffliches Wissen über deren Gegenstände, Strukturen und Abläufe. Dabei unterliegt die Modellbildung – als Kern des Software Engineering – stets Interessen und Machteinflüssen beteiligter Akteure und durchläuft folgende Schritte der Abstraktion und Formalisierung (vgl. Andelfinger 1997):

- *Semiotisierung*: Die sprachlich präzise Beschreibung einer gewünschten neuen sozialen Praxis liefert ein perspektivisch reduziertes Abbild wirklicher Arbeitsprozesse als Ergebnis gemeinsamer Reflexion und Kommunikation der Akteure (Sprachanalyse, »Ontologie«). Es ist ein sprachliches *Anwendungsmodell*.
- *Formalisierung*: Durch Abstraktion von situations- und kontextgebundenen Bedeutungen werden Prozesse auf funktionale Standardoperationen und -vorgänge und dafür relevante Beziehungen und Bedingungen reduziert. Das Ergebnis ist eine *funktionale Spezifikation* bzw. ein *formales Modell* etwa in Form von UML (»unified modelling language«).
- *Algorithmisierung*: Überführung von Objekten und Operationen des formalen Modells in selbsttätig maschinell ausführbare Prozeduren in Form aufgabenspezifischer Daten und berechenbarer Funktionen. Ergebnis sind Algorithmen und Datenstrukturen (Wirth 1979) in Form von Programmen als *Berechnungsmodell*.

Damit liefern teils deskriptive, teils präskriptive Modelle gewünschter kooperativer Praxis, die in aufwendiger Kommunikation mittels Text und Grafik auf der Grundlage von Selbst- und Fremdbeobachtung der beteiligten Akteure vereinbart wurden, die Basis zur

Konstruktion von Software. In perspektivischer Reduktion und Abstraktion werden so bestimmte Vorgänge gewünschter kooperativer Praxis formal modelliert und in Programme als formale Darstellung von Berechnungsverfahren überführt: $v = f(u)$ mit $u, v, R \in \Sigma^*$ (Menge aller syntaktisch zulässigen Wörter des Alphabets $\{0,1\}$, vgl. Abb. 2). Die Programme werden per Compiler automatisch in ausführbaren Maschinencode übersetzt und maschinell ausgeführt. Zur Laufzeit existiert die Maschine ›Computer‹ nur noch als Abfolge bedeutungsloser Signalzustände binärer Schaltsysteme, als physisch ablaufender Prozess. Für Dateneingaben werden mittels des Berechnungsmodells f Ergebnisse erzeugt und auf der Oberfläche des Systems angezeigt; diese Ergebnisse lassen sich dann durch Aneignung der Funktionen des Berechnungsmodells im Kontext des Gebrauchs wieder interpretieren. Dadurch wird soziale Praxis insgesamt neu strukturiert. Mit anderen Worten: »Informatik betreiben heißt: operationale Form explizieren und als autooperationale Form verfügbar machen.« (Floyd 1992).

Computertechnik gilt irrtümlich oft als universell einsetzbar. Universell nutzbar sind aber nur die auf Basis Boolescher Algebra operierenden binären Schaltsysteme. Um sie für je bestimmte Aufgaben funktional einzurichten, bedürfen sie der Steuerung durch aufwendig zu bestimmende Programme. In der Interaktion mit Computersystemen geben Nutzer zur Lösung ihrer Aufgaben Zeichen für Daten und damit operierende Funktionen ein, die in ihrem jeweiligen Handlungskontext Sinn ergeben. Im System werden diese im Außen sinnvoll interpretierbaren Zeichen auf bedeutungslose Signale als deren physischer Verkörperung reduziert und durch Anweisungen zweckmäßig konstruierter Programme maschinell verarbeitet (vgl. Abb. 3).

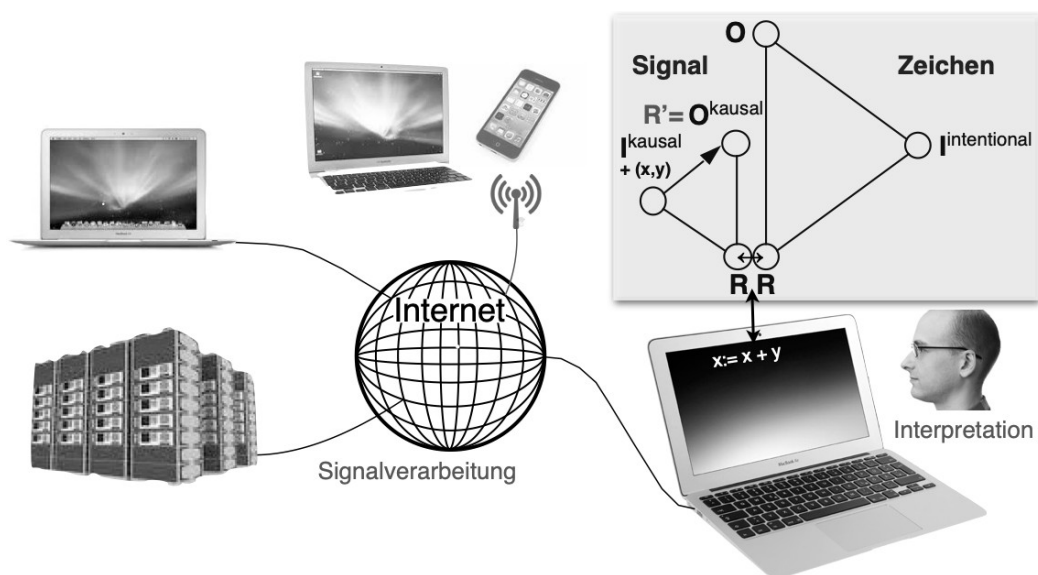


Abb. 3: Interaktive Nutzung von Computerfunktionen (eigene Darstellung)

Das durch f kausal determinierte Resultat R' der Signalverarbeitung ist bei Erscheinen auf der Systemoberfläche wieder als praktisch bedeutsames Zeichen interpretierbar. Beide Zeichenprozesse inner- und außerhalb des Systems sind über das gemeinsame Repräsentamen R fest gekoppelt. Dem außen kontextabhängig durch Nutzer bestimmten intentionalen Interpretanten entspricht intern der algorithmisch determinierte kausale Interpretant. Die intern programmgesteuerte Signalverarbeitung bewirkt, dass Objekt O und Resultat R' zusammenfallen. Eben diese funktionale Kausalität ermöglicht im Außen

deren sinnvolle Interpretation. So ist die Interaktion bestimmt durch Koppelung sinnfreier, aber kausal determinierter Signalverarbeitung im Innern mit sinngebender Interpretation der an der Oberfläche als Zeichen gedeuteten Signale im Außen. Der soziale Raum der Zeichenprozesse wird dabei nicht verlassen (Nake 2001).

Software in Form von Programmen und Daten als deren Funktionskern hat mithin zwei Seiten: Zum einen wird sie in der sozialen Welt der Bedeutungen nach dort bestimmten Zwecken und Anforderungen konzipiert und gestaltet. Zum anderen vermag sie in der physischen Welt der Wirkungen Signal-Zustandsänderungen in Schaltsystemen zu steuern. In dieser Doppelnatur wurzelt die Illusion, Computer seien zu kognitiven Leistungen fähig, wobei aber Funktionen des Berechnungsmodells mit Fähigkeiten zu dessen Erzeugung verwechselt werden.

Programme und Daten gehören als (oft nur schwer) lesbarer Text zur sozialen Welt der Bedeutungen. Sie beschreiben formal den gesteuerten Ablauf einer zweckgemäß rechnenden Maschine, wie sie funktioniert und wozu. Als solche müssen sie sowohl sozialen, von Macht und Interessen abhängigen Anforderungen und Zwecken genügen als auch eine formal korrekte Beschreibung berechenbarer Funktionen (Algorithmen und Datenstrukturen, Berechnungsmodell f) darstellen. Nach automatischer Compilierung in maschinell ausführbaren Code geht infolge von Abstraktion und Formalisierung jeder Bezug zur sozialen Welt von Bezeichnungen und Bedeutungen verloren. Der Code steuert bedeutungslose Signal-Zustandsänderungen binärer Schaltsysteme und bildet das Herzstück der Maschine ›Computer‹ in der physischen Welt der Wirkungen (ohne zu ›wissen‹ wozu). Das so berechnete Ergebnis ist erst aufgrund der Kenntnis der Funktionen des Berechnungsmodells in der sozialen Welt wieder interpretierbar.

Damit wird die Frage aufgeworfen, auf welche Weise so konzipierte und implementierte Computersysteme derart in soziale Praktiken einer Organisation integriert werden können, dass durch Aneignung und Gebrauch ihrer Funktionen die dem Organisationszweck entsprechende Kooperation möglichst wirksam unterstützt wird. Entsprechend ist die Güte der so entworfenen Computerartefakte sowohl an ihrer funktionalen Korrektheit als auch ihrer praktischen Gebrauchstauglichkeit zu bemessen.

3 Soziale Praxis und Aneignung von Artefakten

3.1 Einbettung von Computersystemen in soziale Praxis

Der genannten Frage wird hier aus einer praxistheoretischen Perspektive nachgegangen (Reckwitz 2003). Die nachstehenden Ausführungen stützen sich aus pragmatischen Gründen im Wesentlichen auf die Strukturationstheorie von Giddens (1988), da sie sich in einschlägigen empirischen Forschungsprojekten bewährt hat (z.B. Orlikowski 2000) und sich anschlussfähig an technisch-funktionale wie auch betriebswirtschaftliche Überlegungen zeigt (Ortmann & Sydow 1999, Rohde et al. 2017).

In Prozessen sozialer Interaktion, wie sie etwa in Organisationen auftreten, bilden sich im geteilten Handlungskontext kontinuierlicher kooperativer Tätigkeit von selbst internalisierte Gewohnheiten und Routinen als eine regelmäßige Praxis aus, auch im Gebrauch der als Ressourcen angeeigneten Artefakte. Diese Genese von Regeln bzw. Routinen ist unausweichlich, aber auch unverzichtbar, weil ohne sie der Sinn

vergegenständlichter Ressourcen für koordiniertes Handeln intersubjektiv nicht zu vermitteln wäre. So entstehen die Regeln im Vollzug der wechselseitig aufeinander bezogenen Handlungen, worin sie sich variabel reproduzieren und zugleich die gemeinsame Praxis strukturieren. Damit bilden die ausgeübten Routinen mit den zum praktischen Gebrauch angeeigneten Ressourcen – zu denen auch Organisationsformen und Computerartefakte zählen – ihrerseits erkannte oder nicht erkannte Bedingungen für weiteres Handeln, das zu beabsichtigten oder auch nicht beabsichtigten Ergebnissen und damit zu veränderten Routinen führt. So bringt praktisches Handeln Regeln und Ressourcen als dessen formativen Kontext hervor; sie sind Produkt und Bedingung des Handelns zugleich (»Dualität sozialer Struktur«; Giddens 1988). Die Beteiligten konstruieren so die Wirklichkeit ihrer geteilten sozialen Praxis, aber sie tun es nicht aus freien Stücken, sondern als Gefangene der Umstände, die sie durch ihr gemeinsames Handeln selbst geschaffen haben (vgl. Abb. 4).

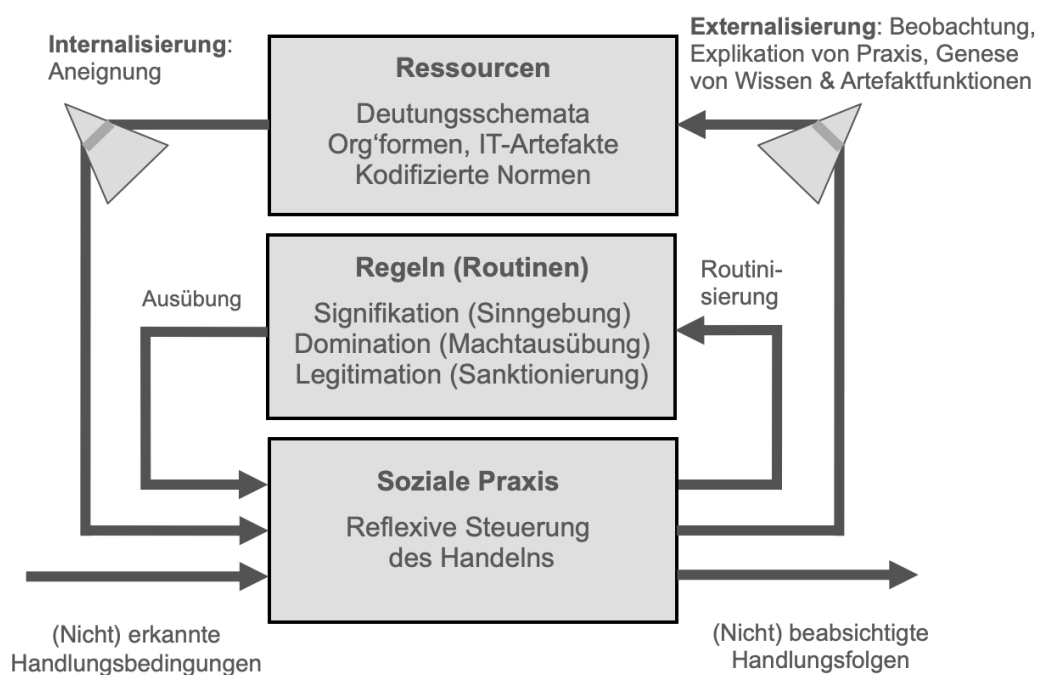


Abb. 4: Rekursive Konstitution von Handeln und Struktur (eigene Darstellung)

Im Verlauf derart eingespielter Praxis kann es in Ausübung praktischer Handlungskompetenz zu Momenten der Irritation oder Überraschung kommen, in denen Dinge oder Vorgänge »anstößig« werden und besondere Aufmerksamkeit erregen, etwa weil etablierte Routinen versagen. Normalerweise als selbstverständlich geltende Dinge oder Abläufe verlieren ihre »Objektivität« durch Verlust gewohnter Bedeutung und dadurch verunsicherte Akteure. Solche Handlungshindernisse führen zu Reflexions- und Suchprozessen, um die Fähigkeit zu angemessenem Handeln wiederzuerlangen. Das erfordert reflexiv erweiterte Begriffsbildung, die intuitive und abstrahierende Bildung gedanklich-begrifflicher Hypothesen, um zu expliziten, die frühere Praxis revidierenden Einsichten zu gelangen, die eine Erklärung und Überwindung der Irritation ermöglichen (Brödner 2023, Kap. 2). Dabei müssen die Hypothesen anschlussfähig sein und das damit versuchsweise neu ausgerichtete, experimentelle Handeln hat sich im Lichte der Organisationsziele zu bewähren, bevor es durch Verinnerlichung wieder »in Fleisch und

Blut« übergehen kann. Auf diese Weise (re-)strukturieren sozial konstituierte Regeln und angeeignete Ressourcen künftiges kooperatives Handeln in Gestalt neuer Prozessabläufe und Computerfunktionen (Giddens 1988, Ortmann 1995, Ortmann & Sydow 1999).

Dieser Kreislauf reflexiv gesteuerter Praxis durch Aneignung explizit revidierter Organisationsformen und Computerfunktionen zu wirksamem Gebrauch stellt die Entwicklung passender Organisationsformen und zweckmäßiger Computerfunktionen vor mindestens drei grundlegende Herausforderungen.

Erstens ist dieser Prozess der Restrukturierung selbstbezüglich in dem Sinne, dass durch die Tätigkeiten des Modellierens, Formalisierens und dann auch Aneignens veränderter Organisationsformen und Computerfunktionen zum praktischen Gebrauch die sozialen Praktiken als deren Gegenstand selbst in Bewegung geraten und sich ändern. Hierin ist die eigentliche Ursache für den häufig zu beobachtenden ›ärgerlichen‹ Umstand zu sehen, dass sich funktionale Anforderungen an Software während der Entwicklung ständig, teils substanziell ändern. Diesem unvermeidlichen Umstand muss das Projektmanagement bei Reorganisation und Softwareentwicklung Rechnung tragen.

Zweitens leiden die Hauptakteure der Umsetzung solcher Projekte, namentlich Sachverständige der Computertechnik und Organisationsentwicklung einerseits und die eigentliche Arbeitstätigkeit ausübende Personen andererseits, unter wechselseitiger Ignoranz: Erstere verstehen wenig vom betrachteten Arbeitsprozess und letztere wissen nicht, wie sich einzelne Aspekte oder Vorgänge dieses Prozesses in Berechnungsverfahren abbilden und nutzen lassen. Das setzt zunächst einen dichten Kommunikations- und gemeinsamen Lernprozess voraus, um die praktische Erfahrung aus der wirklichen Arbeit, bereits expliziertes Prozesswissen und das Wissen über funktionale Möglichkeiten und Grenzen von Computersystemen produktiv miteinander zu verbinden.

Drittens wird in aller Regel der Aufwand an Zeit und Kosten für die individuelle und kollektive Aneignung der Computerfunktionen und der neuen Arbeitspraxis beträchtlich unterschätzt. Oftmals ist er, jedenfalls bei komplexen Entwicklungen, deutlich größer als der Aufwand für den funktionalen Entwicklungsprozess von Software und Organisation selbst. Diese Aneignungsprozesse gilt es daher frühzeitig bewusst zu gestalten und anzugehen.

3.2 Konsequenzen für eine sozialverträgliche Systemgestaltung

Um Risiken des Scheiterns zu verringern, müssen Projekte der Organisationsentwicklung, die mit dem Einsatz von Computersystemen einhergehen, diesen Einsichten zufolge von Beginn an und durchweg *partizipativ* und *evolutionär* angelegt sein: partizipativ, um den kommunikativen Anforderungen zur Überwindung der wechselseitigen Ignoranz gerecht zu werden und um die Gebrauchstauglichkeit entstehender Artefakte zu gewährleisten. Evolutionär, um die ›ärgerliche‹, aber unvermeidliche Selbstbezüglichkeit von Organisationsentwicklung mit Software produktiv zu bewältigen. Grundvoraussetzung dafür ist, sich zu Beginn eines Projekts über die strategischen Ziele und verfügbaren Ressourcen zu verständigen, die dann in der Projektdurchführung als Richtschnur und Erfolgsmaßstab dienen (denn wo kein Ziel ist, ist jeder Weg richtig).

Dementsprechend erfordert eine im Sinne dieser Strategie erfolgreiche Projektdurchführung zu Beginn einen gemeinsamen Lernprozess, in dem die Akteure ihre unterschiedlichen Perspektiven, Fähigkeiten und Interessen zusammenführen, ein

gemeinsames Verständnis der Arbeitsprozesse und eine geteilte Sprache über Computerfunktionen entwickeln. Sie erlaubt es, über sich ändernde Arbeitsprozesse und darin zu nutzende Computerartefakte geteiltes explizites Wissen zu artikulieren. Kritische Faktoren sind dabei (Brödner 2009):

- die Etablierung eines konstruktiven Verhältnisses zwischen allen an der Entwicklung, Anwendung und konkreten Nutzung der Computersysteme beteiligten Akteuren, das sich auf Vertrauen, Offenheit und dichte Kommunikation gründet;
- ein reflexives, evolutionäres Vorgehen in Gestalt überschaubarer Entwicklungs- und Lernschleifen, das die Systemfunktionalität und deren Aneignung durch die Akteure schrittweise erzeugt und so wiederholt Gelegenheit zur Bewertung und Einflussnahme bietet, mit begrenzten Anforderungen und Entwicklungsaufgaben innerhalb jeder einzelnen Entwicklungsschleife, sodass die Risiken in Grenzen gehalten werden (selbst bei Rücknahme des letzten Schritts existiert die zuvor erreichte bessere Praxis);
- ein kompetentes, mit hinreichenden Machtressourcen versehenes Projektmanagement, das für Einhaltung des evolutionären Vorgehens in Entwicklungszyklen, striktes Projektcontrolling mit formativer Evaluation jeweils erreichter Resultate (»change management«) sowie der verbleibenden Risiken (»risk management«) Sorge trägt und dabei Projekterfahrungen laufend reflektiert und expliziert (»lessons learned«);
- eine diesem Vorgehen entsprechende Untermauerung durch softwaretechnische Methoden mit besonderem Augenmerk auf einer modularisierten, änderungsfreundlichen Systemarchitektur, auf systematischer Anforderungsermittlung und auf Systemintegration und -verifikation.

Diesen Einsichten folgt im Wesentlichen auch das in der Siegener Wirtschaftsinformatik langfristig verfolgte Forschungsprogramm des »Grounded Design« (vgl. Rohde et al. 2017). Wie dort gesammelte reiche Erkenntnisse und Erfahrungen aus unterschiedlichsten Projekten zeigen, vermag das Negieren dieser Einsichten und Prinzipien der partizipativ-evolutionären Vorgehensweise das häufige Scheitern von Projekten der Organisationsentwicklung weithin gut zu erklären.

Ein vergleichsweise schlichtes und daher leicht zu durchschauendes Projekt aus jüngster Zeit mag die Gründe des Scheiterns beispielhaft illustrieren: Für den Münsteraner »Kita-Navigator« wurde 2022 eine Software mit einem Algorithmus installiert, der jedem Kind ein Platzangebot zuwies. Als bald fiel auf, dass Geschwistern nicht mehr dieselbe Kita zugewiesen wurde, obwohl dies zuvor üblich gewesen war. Infolge von Beschwerden – auch darüber, dass die angebotenen Betreuungsstunden nicht den angefragten entsprachen – stoppte das Jugendamt das System. Es zeigte sich nicht nur von dem massiven Protest, sondern auch über den gravierenden vermeintlichen »Systemfehler« überrascht. Der dabei verwendete Gale-Shapley-Matching-Algorithmus (Gale/Shapley 1962) ist aber bekanntermaßen nicht in der Lage, Randbedingungen wie die genannten zu berücksichtigen (vgl. Lulamae 2023). Ursache des Desasters sind mithin nicht vermeintliche »Systemfehler«, sondern Denkfehler der Verantwortlichen, die in ihrem Glauben an »magische« Fähigkeiten des Computers und ihrer ignorant-dilettantischen Vorgehensweise die dargelegten Grundsätze des Projektmanagements missachteten: eine sorgfältige Analyse und sprachliche wie formale Modellierung praxisgerechter Prozesse

der Platzvergabe sowie ein partizipativ und evolutionär angelegtes Vorgehen bei der funktionalen Systemspezifikation.

Insgesamt ist für die Gestaltung soziotechnischer Systeme (Ulich 2011) die Entwicklungsperspektive entscheidend, insbesondere beim Einsatz von Computersystemen in sozialen Praktiken kooperativer Wissensarbeit in Organisationen. Die praxistheoretische Perspektive zielt darauf, sich an Eigenheiten menschlichen Handelns, Lernens und der Entfaltung von Arbeitsvermögen zu orientieren, um gute Arbeit, Produktivität und Innovationsfähigkeit zu ermöglichen. Statt mit Computersystemen menschliches Arbeitsvermögen nachahmen und weitgehend ersetzen zu wollen – wie vermeintlich »lernfähig« oder »intelligent« diese auch seien –, kommt es gerade darauf an, die besonderen Fähigkeiten lebendiger Arbeit zu Intuition, Kreativität und Urteilsvermögen mit der hohen Leistungsfähigkeit rechnender Maschinen produktiv zu verbinden: »Intelligence Augmentation (IA)« statt »Artificial Intelligence (AI)« (Brödner 2023: Kap. 3; vgl. auch Winograd 1996, neuerdings Acemoglu et al. 2023). So sucht die praxistheoretische Gestaltungsperspektive der »Intelligenzverstärkung« nach Wegen, menschliches Arbeitsvermögen zu entfalten und durch reflexive Gestaltung anspruchsvoller Arbeitsaufgaben in Verbindung mit sachgerecht und gebrauchstauglich gestalteten Computerfunktionen und Interaktionsformen zu ertüchtigen.

4 »KI« – das Phantom der Computertechnik

Dass die Computertechnik von Beginn an mit Vorstellungen der »computational theory of mind« verknüpft und mit wellenförmig wiederkehrenden Phantasien über »künstliche Intelligenz«, neuerdings sogar mit »Artificial General Intelligence (AGI)«, verbunden wird, wurzelt zum einen in der unzureichend verstandenen Funktionsweise des neuartigen Maschinentyps Computer, zum anderen in überaus vagen Bestimmungen dessen, was »KI« überhaupt ist. Einschlägige Definitionsversuche erschöpfen sich entweder in zirkulären Scheindefinitionen, die »KI« als Fähigkeit zur Bewältigung von Aufgaben bestimmen, die »menschliche Intelligenz« erfordern, oder sie schreiben Computersystemen anthropomorphisierend »rationale Handlungsfähigkeit« zu (Brödner 2023: 189 f; Russell & Norvig 2009). Derart irreführende Definitionen verwechseln Fähigkeiten zum Herstellen technischer Artefakte mit deren Eigenschaften als Hergestelltem, die kognitiven Fähigkeiten des analytischen Verstehens sozialer Praktiken und des intelligenten Entwerfens der Artefakte mit den Funktionen operativer Programme als deren Ergebnis – ein schwerer Kategorienfehler und Quelle der Mystifizierung von Computersystemen. Diese sind aber stets Resultat der Intelligenz der Personen, die sie entwickeln: Sie analysieren praktische, oft »verwickelte Probleme« (Rittel & Webber 1973) und lösen sie methodisch durch menschengemachte Berechnungsverfahren, die Computer dann lediglich operativ ausführen.

Eine genauere Analyse der neueren »konnektionistischen KI« ergibt, dass diese auf einem simplen Trick beruht: Statt sich den aufwendigen Mühen der Modellierung und Formalisierung, der Bestimmung von f (vgl. Abb. 2), zu unterziehen, werden diese Mühen mittels sog. »maschinellen Lernens (ML)« umgangen, indem durch theorieloses, auf Erfahrung gegründetes Probieren (sog. »Bricolage«) eine komplex strukturierte Funktion f mit vorerst noch unbestimmten Parametern einfach vorgegeben wird, die bestmöglich an

gegebene anwendungsspezifische Daten angepasst wird (mathematisch eine Funktions-Approximation mittels Minimierung einer Verlustfunktion durch Gradientenverfahren und Rückpropagierung, sog. »Backpropagation«; vgl. Russell & Norvig 2009).

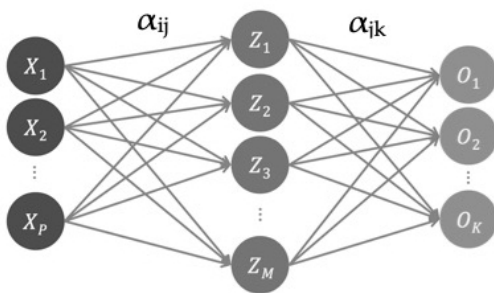
Damit wird logisch eine Kausalität postuliert, die bei vielen Anwendungen gar nicht existiert. Methodisch sinnvoll ist diese Vorgehensweise, mit der Ergebnisse aufgrund vorausgehender Beobachtungen vorhergesagt werden, aber nur, falls entweder tatsächlich Kausalität besteht, oder, bei stochastischen Beziehungen, in einem stabilen, sich kaum verändernden Umfeld; beides ist aber in Anbetracht der Kontingenz sozialer Praktiken meist nicht gegeben. Dafür seit langem bekannte Verfahren sind etwa die lineare Regression (geht zurück auf Gauß ~ 1820), die logistische Regression für Funktionen mit nur 2 Werten (geht zurück auf Verhulst ~ 1830/40) oder verschiedene Varianten von Entscheidungsbaum-Verfahren (1960/80er Jahre; Quinlan 1989).

Derzeit gilt sog. »Deep Learning« (Wick 2017) als neue »KI«-Wunderwaffe: Sie nutzt mehrschichtige künstliche neuronale Netze (KNN) als Funktionsträger für f , oftmals in Gestalt einer Kombination aufgabenspezifischer Module (sog. »Mixture of Experts«). Ihre Bedeutung verdanken sie der Fähigkeit zu *universeller Approximation*: Mittels eines mehrschichtigen KNN und sigmoider Aktivierungsfunktion kann jede stetige Funktion beliebig genau approximiert werden (Hornik 1991). Die sehr aufwendige Anpassung der vielen Funktionsparameter erfordert jedoch extrem hohe Rechenleistung spezieller Hardware für parallele Datenverarbeitung und Tensoroperationen in hochdimensionalen Vektorräumen, die erst seit kurzem verfügbar ist. Diese Vorgehensweise wirft allerdings neue schwerwiegende Probleme auf:

- Oftmals werden mangelhafte Daten methodisch unkontrollierter Herkunft und unbekannter Qualität verwendet, von der aber der Ergebnisnutzen entscheidend abhängt (»garbage in – garbage out«).
- Ohne tatsächlich vorliegende Kausalität werden nur angenäherte oder wahrscheinliche, daher prinzipiell unsichere und nur schwer zu beurteilende Ergebnisse erzeugt.
- Das daher undurchschaubare »Black-Box«-Verhalten ist Haupthindernis im Gebrauch, weil dessen Ergebnisse für praktisches Handeln nicht sicher einschätzbar sind.

Mit üblichem »Lernen« hat die Anpassung von KNN-Parametern nichts gemein, verführt aber zu der irrigen Vorstellung »lernender Systeme« (tatsächlich sind es nur *adaptive* Systeme). Ihr Verhalten lässt sich von Schicht zu Schicht mittels Methoden linearer Algebra, durch Operationen mit Vektoren und Matrizen, vollständig beschreiben, wobei die noch zu bestimmenden Parameter α_{ij} als Gewichte die Stärke der Verbindungen zwischen den Knoten benachbarter Schichten darstellen (vgl. Abb. 5). Sie sind Elemente der schichtspezifischen Gewichtsmatrix α^T (T steht für »transponiert«, d.h. Zeilen und Spalten werden verfahrensbedingt vertauscht), mit der jeweils die Komponenten des Ergebnisvektors Z als Summe der gewichteten Komponenten des Eingangsvektors X nach Maßgabe einer Aktivierungsfunktion σ (üblicherweise eine ableitbare logistische bzw. Softmax-Funktion) berechnet werden. Im »Training« werden die Gewichte zwecks Funktionsanpassung aus der Forderung nach Minimierung einer Verlustfunktion durch »Backpropagation« schrittweise neu berechnet (dabei können freilich numerische Probleme auftreten, deren Bewältigung viel mathematische Erfahrung erfordert). Da

heutige große KNN in Vektorräumen mit bis zu 10^5 Dimensionen modelliert werden, ist der Rechenaufwand gigantisch.



Die KNN-Struktur (Anzahl der Schichten und Knoten) ist ohne unterliegende Kausalität willkürlich gewählt. Sie wird je nach Aufgabe aufgrund von Erfahrung durch theorieloses Probieren festgelegt (»Bricolage«). Vorerst noch unbestimmte Parameter α_{ij} und α_{ik} werden als Verbindungsgewichte durch bestmögliche Anpassung an gegebene Kontextdaten berechnet (sog. »Training«).

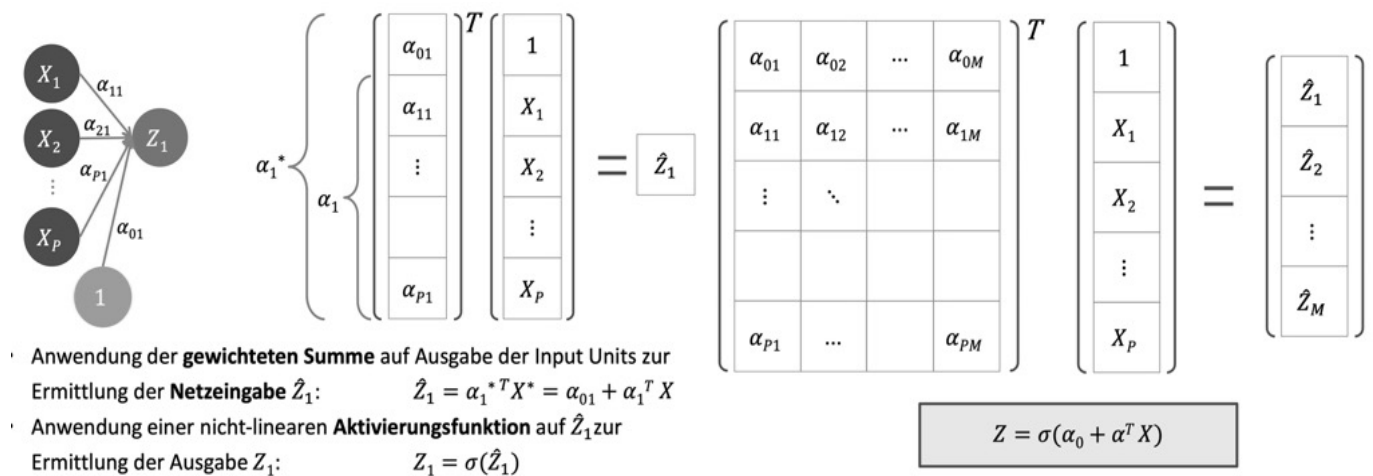


Abb. 5: Mathematisches Modell eines KNN mittels linearer Algebra (Darstellung: Uni Köln)

Ähnlich beruht auch die Leistung der derzeit großes Aufsehen erregenden »Generative Pretrained Transformer (GPT)« als Kern großer Sprachmodelle (»LLM«) auf modular angeordneten, für maschinelle Sprachverarbeitung passend strukturierten KNN, die mittels sehr großer Textsammlungen ($> 10^2$ TB) »trainiert« werden. Dazu müssen Wörter und Satzzeichen des verwendeten Vokabulars zunächst durch Vektoren als mathematisch handhabbare Objekte (sog. »Tokens«) repräsentiert werden, wobei für zwischen ihnen bestehende Kontextbeziehungen bedingte Wahrscheinlichkeiten mittels »vortrainierter« Gewichtsmatrizen berechnet, in weiteren Netzmodulen noch optimiert werden (z.B. durch sog. »Attention«-Module, die Bezüge auch zwischen entfernten Wörtern berücksichtigen). Dabei verlieren die Wörter aber ihre Bedeutung (vgl. den triadischen Zeichenbegriff, Abb. 1). Die so berechneten bedingten Wahrscheinlichkeiten für Beziehungen zwischen Wörtern eines Kontexts suggerieren Sinn, wo tatsächlich nur syntaktische Korrelationen bestehen. Sie bestimmen lediglich, welche Wörter in neu zu generierenden Textsequenzen höchstwahrscheinlich auf ein gegebenes Wort folgen und wie Sprachausdrücke (sog. »Prompts«) mit auf diese Weise generierten, aber infolge der Abstraktion ihrer Bedeutung beraubten Wortfolgen scheinbar »vernünftig« fortgesetzt werden (Vaswani et al. 2017; zu solchen Berechnungsverfahren vgl. exemplarisch Brödner 2023: 7 f).

Syntaktisch korrekte Wortfolgen bilden noch keinen sinnvollen Text. So können mittels GPT verblüffend ›menschlich‹ erscheinende ›Texte‹ wie auch Fehlleistungen zugleich erzeugt werden, ohne dass Nutzer jedoch imstande wären, beides auf Anhieb zu erkennen, wie dies Experimente immer wieder demonstrieren (z.B. Chomsky et al. 2023, Floyd 2023). Mithin haben paradoxerweise »KI«-Enthusiasten und -Skeptiker zugleich recht: Weil die erzeugten Wortfolgen allein auf berechneten abstrakten Wahrscheinlichkeitsbeziehungen zwischen Wörtern entsprechend allgemeinem Sprachgebrauch beruhen, können sich gleichermaßen sprachlich gelungen erscheinende wie auch sachlich falsche Resultate ergeben. Obgleich viele ›Texte‹ syntaktisch und idiomatisch korrekt erscheinen, täuschen sie Bedeutung nur vor. Da sie keinen persönlichen Urheber haben, besitzen sie nur ›verliehene‹ Intentionalität ohne sinnhaften situativen Bezug zur Welt. Zu Recht werden sie daher als »stochastische Papageien« (Bender et al. 2021) bezeichnet, die in großen Textsammlungen enthaltene Wendungen und Floskeln ungeachtet ihrer Bedeutung nur ›nachplappern‹ und vorherrschende Sprachgewohnheiten samt der darin verwobenen Vorurteile und Verzerrungen imitieren – ohne zu ›wissen‹, wovon die Rede ist. Eingedenk dessen ist eine Kennzeichnungspflicht für computergenerierte Texte und Bilder dringend geboten.

Dabei stellt sich verschärft die Kernfrage nach der Gebrauchstauglichkeit, d.h. bei welchen Aufgaben und in welchen Situationen der breitgefächerten Praxis von Wissensarbeit solche Systeme überhaupt sinnvoll genutzt werden können – zumal die den Sprachmodellen zugrunde liegende GPT-Technik darüber hinaus unter schweren, grundsätzlich nicht behebbaren Mängeln leidet, die dem »Training« ihrer Funktionen mittels stochastischer Methoden auf Basis begrenzter und oft situativ unspezifischer und schwer einzuschätzender Textdaten geschuldet sind:

Erstens produzieren sie im Ergebnis ›Pseudotexte‹ aus nur wahrscheinlichen, daher stets unsicheren Wortfolgen, die mitunter schwer erkennbare Falschaussagen enthalten, was deren sinnvolle Interpretation im praktischen Gebrauch erheblich behindert.

Zweitens liefern sie aufgrund begrenzter, oft auch kontaminierter »Trainings«-Daten häufig verzerrte, gar ›erfundene‹ Ergebnisse (sog. »Halluzinationen«). Insbesondere werden häufige Beziehungen zwischen Repräsentamen für Wörter und Satzzeichen (»Tokens«) systematisch über- und seltene Beziehungen systematisch unterschätzt mit der Folge, dass von Sprachmodellen selbst erzeugte und zu deren »Training« rekursiv verwendete Ergebnisse zu Degeneration, letztlich zu deren »Kollaps« führen (Shumailov et al. 2024).

Drittens versagen sie – ebenfalls mangels Verständnisses der verarbeiteten ›Texte‹ – bereits bei einfachen Aufgaben logischen Schließens (und beharren bei Nachfragen zur Begründung gleichwohl darauf, Recht zu haben), so etwa bei folgender einfacher Testaufgabe:

»Alice hat N Brüder und auch M Schwestern. Wieviel Schwestern hat ein Bruder von Alice?«
Bei einem Test sind daran sämtliche untersuchten LLM überwiegend gescheitert (mit Ausnahme von ChatGPT 4 in wenigen Fällen; Nezhurina et al. 2024).

Eingedenk dieser Einblicke in die Funktionsweise »konnektionistischer KI« und ihrer Grenzen kann festgehalten werden, dass die vordergründig verblüffenden Ergebnisse von GPT einerseits auf lange bekannten mathematischen Methoden, andererseits auf der erst seit kurzem verfügbaren, enorm gesteigerten Rechenleistung spezialisierter GPU-

Hardware beruhen («Graphics Processing Units»). Tatsächlich werden KNN (darunter GPT) durch Algorithmen gesteuert wie andere Computersysteme auch, allerdings mit der Besonderheit, dass die zugrunde gelegten generischen Berechnungsverfahren nicht aus sozialer Praxis analytisch bestimmt, sondern willkürlich vorgegeben und mittels Funktions-Approximation für die jeweilige Aufgabe »trainiert« werden, was sie zu adaptiven Systemen ohne eigene »Intelligenz« macht. Algorithmen sind als kausal determinierte berechenbare Funktionen das genaue Gegenteil von natürlicher Intelligenz, wenn man darunter mit Jean Piaget dasjenige versteht, »was man einsetzt, wenn man nicht weiß, was man tun soll«, also vor einem Problem steht, zu dessen Lösung Intuition, Kreativität und Urteilsvermögen unter Unsicherheit gefragt sind. Die ihnen in üblichen Erzählungen zugeschriebene »Intelligenz« oder »Lernfähigkeit« – häufig ist von »autonomen« (tatsächlich: automatischen) oder von »lernenden Systemen«, gar von »lernenden Algorithmen« die Rede (ein Oxymoron sondergleichen) – beruht allein auf irreführenden, anthropomorphisierenden Metaphern, die menschliche Fähigkeiten auf kausale maschinelle Funktionen nur projizieren, letztlich ein Etikettenschwindel.

5 Schlussfolgerungen

Diese Einblicke können helfen, die Risiken zu verringern, dass Projekte rechnergestützter Organisationsentwicklung scheitern. Mit Blick darauf lassen sich am Ende aus ihnen eine Reihe von Konsequenzen ziehen. Vor allem gilt es, zunächst die grundlegende Funktionsweise von Computersystemen nachzuvollziehen und zu verstehen, wie die darin ablaufenden Algorithmen zustande kommen. Dabei muss stets der fundamentalen Differenz zwischen bei Menschen vorgängigem praktischem Können als Inbegriff von Erfahrung, Intuition und Abstraktionsfähigkeit einerseits und dem stets nur partiell daraus explizierbaren begrifflich-propositionalen Wissen als Grundlage von Modellierung und Formalisierung andererseits Beachtung geschenkt werden. Daran sind schon die vielfältigen Anstrengungen symbolischer »KI« gescheitert. Diese Differenz tritt selbst in der weitgehend formalisierten Mathematik in Gestalt der Ebene des operativen Kalküls und der diese reflektierenden metamathematischen Ebene der Axiome und Beweisregeln in Erscheinung. Zur mathematischen Fähigkeit gehört eben, dass sie durch Nachdenken über von ihr formalisierte operative Kalküle, kraft kreativen »abduktiven Schließens« (Peirce), zu außer Reichweite der Kalküle liegenden Einsichten gelangen kann, die gleichwohl formal operationalisierbar sind. Eben dies ist ein Ausweis natürlicher Intelligenz (vgl. z.B. Brödner 2023: 23 f).

Trotz oft wahnhafter Vorstellungen über »KI« sind und bleiben Computerartefakte technische Mittel der Automatisierung formalisierbarer Vorgänge in Zeichenprozessen sozialer Interaktion. Als solche sind sie eingebettet in umfassendere Prozesse lebendiger Wissensarbeit und müssen folglich wie andere Arbeitsmittel auch zweckmäßig und gebrauchstauglich gestaltet werden. Diese Entwicklungsperspektive erfordert, dass beteiligte Akteure erstens die Zusammenhänge zwischen den Funktionen der Berechnungsverfahren und ihrer Nutzung im praktischen Gebrauch verstehen, dass sie diese zweitens in einer Weise entwerfen, implementieren und zum Gebrauch aneignen, in der sich das menschliche Arbeitsvermögen zu entfalten vermag (statt es zu ersetzen) und dass sie dabei drittens partizipativ und evolutionär vorgehen, um den Fallstricken

wechselseitiger Ignoranz und Selbstbezüglichkeit zu entgehen. Dabei erfordert die Aneignung der Artefaktfunktionen besondere Aufmerksamkeit.

Darüber hinaus ergeben sich infolge unzureichender Verlässlichkeit der Ergebnisse von KNN – wegen ihrer auf Approximation bzw. Stochastik beruhenden Funktionsweise – Probleme im instrumentellen Gebrauch dieser Systeme. Die daraus erwachsende Unsicherheit erweist sich als beträchtlicher Stressfaktor, den es zu vermeiden gilt, zumal wenn Nutzer unter Qualitäts- oder Leistungsdruck stehen (Brödner 2023, Kap. 7). Solange es noch keine ihr Verhalten selbst erklärende »KI«-Systeme gibt (was aber absehbar kaum zu erwarten ist), bleibt das Zustandekommen der Ergebnisse für Nutzer verborgen und schwer einzuschätzen. Dennoch droht ihnen häufig, zur Verantwortung gezogen zu werden. In solchen Situationen dürfen diese Systeme daher nicht eingesetzt werden (ansatzweise werden sie durch den AI Act der EU auch eingeschränkt).

Trotz dieser kritischen Beurteilung »konnektionistischer KI« gibt es eng begrenzte Aufgaben und Situationen, in denen adaptive Systeme durchaus sinnvoll eingesetzt werden können. Das setzt aber ein gründliches Umdenken voraus: Statt diese Systeme wie meist üblich in der Illusion imitierbarer menschlicher Intelligenz unreflektiert und explorativ zur Bewältigung schwer zu analysierender, verwickelter Probleme einzusetzen, kann es unter stabilen Verhältnissen sinnvoll sein, adaptive Systeme theoretisch angeleitet, methodisch reflektiert und bei akzeptabler Störanfälligkeit, zur Lösung wohlbestimmter Aufgaben zu verwenden (methodisch vergleichbar einer Regressionsanalyse). Beispiele dafür sind etwa die Klassifikation von Bildern in der Qualitätssicherung bzw. von Lautgestalten in der Sprachverarbeitung oder die Übersetzung einfacher Alltagssprachlicher Texte.

Der hohe Anspruch der Protagonisten »konnektionistischer KI«, mit ihren adaptiven Systemen natürliche Intelligenz generell auf breiter Front imitieren, gar ersetzen zu können, führt derzeit (wie schon bei symbolischer »KI« in den 1980er Jahren) in eine Art »KI«-Wetlauf ohne Rücksicht auf die inhärenten Unsicherheiten ihrer Funktionsweise – die sich unter anderem mit Blick auf automatisierte soziale Kontrolle oder militärische Operationen als fatal erweisen können. Dabei kann auch die neuere »KI«-Entwicklung aus dargelegten Gründen als bedingt gescheitert gelten. Letzten Endes bestimmen die methodischen Aspekte der Formalisierung und Stochastik wie auch die harten Fakten der Logik und Arithmetik die Grenzen des Machbaren in der Computertechnik (Hermes 1971, Hoffmann 2018) und nicht der schöne Schein des in »Narrativen« sozial konstruierten Zaubers der »KI«.

Literatur

Acemoglu, Daron/ Autor, David/Johnson, Simon (2023): Can We Have Pro-Worker AI? Choosing a path of machines in service of minds. CEPR Policy Insight No. 123. Cambridge (MA): MIT CEPR.

https://cepr.org/system/files/publication-files/191183-policy_insight_123_can_we_have_pro_worker_ai_choosing_a_path_of_machines_in_service_of_minds.pdf

- Andelfinger, Urs (1997): Diskursive Anforderungsanalyse. Ein Beitrag zum Reduktionsproblem bei Systementwicklungen in der Informatik. Frankfurt a. M.: Peter Lang
- Bateson, Gregory (1980): Mind and Nature. A Necessary Unity. Toronto: Bantam Books
- Bauer, Friedrich L./Goos, Gerhard (1971): Informatik. Eine einführende Übersicht. Teile 1 & 2. Berlin, Heidelberg, New York: Springer
- Bender, Emily L./Geburu, Timnit/McMillan-Major, Angelina/Shmitchell, Shmargaret (2021): On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? Conference on Fairness, Accountability, and Transparency (FAccT '21), Virtual Event, March 3-10, 2021, New York (NY): ACM.
<https://dl.acm.org/doi/pdf/10.1145/3442188.3445922>
- Berkeley, Edmund Callis (1849): Giant Brains or Machines that Think. New York: Wiley
- Boole, George (1847): The mathematical analysis of logic, being an essay towards a calculus of deductive reasoning. Cambridge: Macmillan, Barclay & Macmillan.
<https://www.gutenberg.org/files/36884/36884-pdf.pdf>
- Brödner, Peter (1997): Der überlistete Odysseus. Über das zerrüttete Verhältnis von Menschen und Maschinen. Berlin: edition sigma
- Brödner, Peter (2009): The Misery of Digital Organisations and the Semiotic Nature of IT. In: AI & Society 23, S. 331-351
- Brödner, Peter (2014): Durch „Information“ desinformiert? Zur Kritik des Paradigmas der Informationsverarbeitung. In: Arbeits- und Industriesoziologische Studien 7 (1), S. 42-59
- Brödner, Peter (2023): »Künstliche Intelligenz«: Dichtung und Wahrheit. Einblicke in die Technik des Berechnens und in Mythen um Intelligenz. International Reports on Socio-Informatics 20 (1). Bonn: IISI,
https://www.iisi.de/wp-content/uploads/2023/09/IRSI_V20I1.pdf
- Chomsky, Noam/Roberts, Ian/Watumull, Jerry (2023): The False Promise of ChatGPT. In: New York Times, 8. März 2023.
<https://www.nytimes.com/2023/03/08/opinion/noam-chomsky-chatgpt-ai.html>
- Dreyfus, Hubert L. (1979): What Computers Can't Do. The Limits of Artificial Intelligence. New York: Harper & Row
- Eco, Umberto (1991): Semiotik. Entwurf einer Theorie der Zeichen. München: Wilhelm Fink
- Falkenberg, Eckhard D./Hesse, Wolfgang/Lindgreen, Paul/Nilsson, Björn E./Oei, J. L. Han/Roland, Colette/Stamper, Ronald K./Van Assche, Frans J. M., Verrijn-Stuart, Alexander A./Voss, Klaus (1998): A Framework of Information Systems Concepts. The FRISCO Report. IFIP. www.mathematik.uni-marburg.de/~hesse/papers/fri-full.pdf
- Fetzer, James H. (2001): Computers and Cognition: Why Minds are not Machines. Dordrecht: Springer Science & Business Media
- Floyd, Christiane (1992): Software Development as Reality Construction. In: Floyd, Christiane/Züllighoven, Heinz/Budde, Reinhard/Keil-Slawik, Reinhard (Hrsg.): Software Development and Reality Construction. Berlin, Heidelberg, New York: Springer, S. 86-100

- Floyd, Christiane (2023): From Joseph Weizenbaum to ChatGPT. Critical Encounters with Dazzling AI Technology. In: Weizenbaum Journal of the Digital Society 3 (3), 1-29
- Frege, Gottlob (1879): Begriffsschrift. Eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle: Louis Nebert
- Gale, David / Shapley, Lloyd (1962): College Admissions and the Stability of Marriage. In: American Mathematical Monthly 69 (1), S. 9-14.
<https://kostochk.web.illinois.edu/math412-10/Gale-Sh.pdf>
- Giddens, Anthony (1988): Die Konstitution der Gesellschaft. Grundzüge einer Theorie der Strukturierung. Frankfurt a. M.: Campus
- Hermes, Hans (1971): Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit. Berlin, Heidelberg, New York: Springer
- Hirsch-Kreinsen, Hartmut (2023): Das Versprechen der Künstlichen Intelligenz. Gesellschaftliche Dynamik einer Schlüsseltechnologie. Frankfurt a. M., New York: Campus
- Hoffmann, Dirk W. (2018): Grenzen der Mathematik. 3. Auflage. Berlin, Heidelberg: Springer Spektrum
- Hornik, Kurt (1991): Approximation Capabilities of Multilayer Feedforward Networks. In: Neural Networks 4 (2), S. 251-257
- Ifrah, Georges (1989): Universalgeschichte der Zahlen. Frankfurt a. M., New York: Campus
- Lulamae, Josephine (2023): Automatisierte Kitaplatz-Vergabe: Wenn sich die Wege von Geschwistern trennen. Algorithm Watch, 28. März 2023.
<https://algorithmwatch.org/de/automatisierte-kitaplatz-vergabe/>
- Maturana, Humberto R. / Varela, Francisco J. (1987): Der Baum der Erkenntnis. Die biologischen Wurzeln des Erkennens. Bern: Scherz
- McCarthy, John / Minsky, Marvin L. / Rochester, Nathaniel / Shannon, Claude E. (1955): A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. <https://archive.computerhistory.org/resources/access/text/2023/06/102720392-05-01-acc.pdf>
- McCulloch, Warren S. / Pitts, Walter A. (1943): A Logical Calculus of the Ideas Imminent in Nervous Activity. In: Bulletin of Mathematical Biophysics 5, S. 115-133
- Morris, Charles William (1975): Grundlagen der Zeichentheorie. München: Hanser
- Nake, Frieder (1997): Der semiotische Charakter der informatischen Gegenstände. In: Bayer, Udo (Hrsg.): Festschrift zu Ehren von Elisabeth Walter. Baden-Baden: agis, S. 24-35
- Nake, Frieder (2001): Das algorithmische Zeichen. In: Bauknecht, Kurt / Brauer, Wilfried / Mück, Thomas (Hrsg.): Informatik 2001. Tagungsband der GI/OCG Jahrestagung, S. 736-742
- Newell, Allen / Shaw, J. Cliff / Simon, Herbert A. (1959): Report on a general problem-solving program. Proceedings of the International Conference on Information Processing.
http://bitsavers.informatik.uni-stuttgart.de/pdf/rand/ipi/P-1584_Report_On_A_General_Problem-Solving_Program_Feb59.pdf
- Newell, Allen (1980): Physical Symbol Systems. In: Cognitive Science 4, S. 135-183

Nezhurina, Marianna / Cipolina-Kun, Lucia / Cherti, Mehd / Jitsev, Jenia (2024): Alice in Wonderland: Simple Tasks Showing Complete Reasoning Breakdown in State-Of-the-Art Large Language Models. [arXiv:2406.02061v2 \[cs.LG\]](https://arxiv.org/abs/2406.02061v2)

Ogden, Charles Kay / Richards, Ivor Armstrong (1989): The Meaning of Meaning. Reprint of 1923 edition. San Diego, New York, London: HB

Orlikowski, Wanda J. (2000): Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations. In: Organization Science 11 (4), S. 404-428

Ortmann, Günther (1995): Formen der Produktion. Organisation und Rekursivität. Opladen: Westdeutscher Verlag

Ortmann, Günther / Sydow, Jörg (1999): Grenzmanagement in Unternehmensnetzwerken: Theoretische Zugänge. In: Die Betriebswirtschaft 59 (2), S. 205-220

Peirce, Charles S. (1960): Collected Papers (CP), Bd. 2: Elements of Logic. Cambridge (MA): Harvard University Press.
https://archive.org/details/collectedpaperso0000unse_v0n0/page/n3/mode/2up

Peirce, Charles S. (1983): Phänomen und Logik der Zeichen. Frankfurt a. M.: Suhrkamp

Pylyshyn, Zenon (1984): Computation and Cognition. Towards a Foundation of Cognitive Science. Cambridge, MA: MIT Press

Quinlan, John Ross (1986): Induction of decision trees. In: Machine Learning 1 (1), S. 81-106

Reckwitz, Andreas (2003): Grundelemente einer Theorie sozialer Praktiken. In: Zeitschrift für Soziologie 32 (4), S. 228-301

Rescorla, Michael (2024): The Computational Theory of Mind. In: Stanford Encyclopedia of Philosophy, <https://plato.stanford.edu/entries/computational-mind/>

Rittel, Horst W. J. / Webber, Melvin M. (1973): Dilemmas in a General Theory of Planning. In: Policy Sciences 4, S. 155-169

Rohde, Markus / Brödner, Peter / Stevens, Gunnar / Betz, Matthias / Wulf, Volker (2017): Grounded Design – a Praxeological IS Research Perspective. In: Journal of Information Technology 32 (2), S. 163-179

Russell, Stuart / Norvig, Peter (2009): Artificial Intelligence: A Modern Approach. 3. Auflage. Essex (GB): Pearson

Searle, John Rogers (2012): Wie wir die soziale Welt machen. Die Struktur der menschlichen Zivilisation. Berlin: Suhrkamp

Searle, John Rogers (1980): Minds, Brains and Programs. In: The Behavioral and Brain Sciences 3, S. 417-457

Shannon, Claude E. (1948): A Mathematical Theory of Communication. In: The Bell System Technical Journal 27, S. 379-423 und 623-656

Shumailov, Ilia / Shumaylov, Zakhar / Zhao, Yiren / Gal, Yarin / Papernot, Nicolas / Anderson, Ross (2023): The Curse of Recursion: Training on Generated Data Makes Models Forget. [arXiv:2305.17493v2 \[cs.LG\]](https://arxiv.org/abs/2305.17493v2)

Taylor, Charles (2017): Das sprachbegabte Tier. Grundzüge des menschlichen Sprachvermögens. Berlin: Suhrkamp

- Tomasello, Michael (2009): Die Ursprünge der menschlichen Kommunikation. Frankfurt a. M.: Suhrkamp
- Turing, Alan M. (1936): On Computable Numbers, with an Application to the Entscheidungsproblem. In: Proceedings of the London Mathematical Society. Band 42, S. 230-265. <https://doi.org/10.1112/plms/s2-42.1.230>
- Turing, Alan M. (1950): Computing Machinery and Intelligence. In: Mind 59, Nr. 236, S. 433-460, <https://www.cs.mcgill.ca/~dprecup/courses/AI/Materials/turing1950.pdf>
- Ulich, Eberhard (2011): Arbeitspsychologie. 7. Auflage. Stuttgart: Schäffer Poeschel und Zürich: vdf
- Vaswani, Ashish/Shazeer, Noam/Parmar, Niki/Uszkoreit, Jakob/Jones, Llion/Gomez, Aidan N./Kaiser, Lukasz/Polosukhin, Illia (2017): Attention Is All You Need. [arXiv:1706.03762v5](https://arxiv.org/abs/1706.03762v5) [cs.CL]
- Weizenbaum, Joseph (1976): Computer Power and Human Reason. From Judgment to Calculation. New York: Freeman & Co.
- Winograd, Terry (Hrsg.) (1996): Bringing Design to Software. Reading (MA): Addison-Wesley
- Wick, Christoph (2017): Deep Learning. In: Informatik Spektrum 40 (1), S. 103–107
- Wirth, Niklaus (1979): Algorithmen und Datenstrukturen. 2. Auflage. Stuttgart: Teubner. <https://archive.org/details/algorithmenundda0000wirth/page/n3/mode/2up>
- Wirth, Niklaus (1995): A Plea for Lean Software. In: Computer 28 (2), S. 64–68. <https://cr.yp.to/bib/1995/wirth.pdf>