

Stefan Meretz (Februar 1997)

Informatisierung der Psychologie - Psychologisierung der Informatik

"Der Computer sieht nicht, ob er mit Ihnen allein oder mit sechs weiteren Personen spricht. Er weiß nicht, ob Sie Ihren Schlafanzug, eine Abendgarderobe oder gar nichts anhaben. Sie kehren ihm den Rücken zu, während er Ihnen etwas Interessantes zeigen will, oder befinden sich außerhalb seiner Hörweite, während er gerade mit Ihnen spricht. (...) (Es) wäre ... an der Zeit, darüber nachzudenken, wie man es dem Computer leichter macht, mit Menschen umzugehen." (Negroponte 1995, S. 158)

Warum wirkt das Zitat problematisch? Weil über den Computer in vermenschlichender Weise geredet wird? Weil der Computer hier als aktiv handelndes Wesen dargestellt ist? Warum wirkt dann jedoch folgende Aussage *nicht* problematisch: "Jetzt hat der blöde Toaster schon wieder die Toasts verbrannt"? Niemand würde hier dem Toaster eine aktive Handlung unterstellen, etwa, daß er die Absicht hatte, die Toasts zu verbrennen, weil ihm die Brotsorte nicht gefällt. Die meisten würden vermutlich versuchen, die Ursache für das Verbrennen der Toastbrotsscheiben herauszufinden, etwa nachsehen, ob die Heizstufe zu hoch eingestellt ist etc. Beim Toaster ist die *Ursache-Wirkungs-Relation* eindeutig. Der Toaster ist ein nach physikalischen Prinzipien funktionierendes Gerät, dessen Funktionsweise durch Parameter bestimmt wird wie z.B. den Zustand (ein/ausgeschaltet), die Heizstufe, den Verschmutzungsgrad etc.

Warum gibt es diese Eindeutigkeit beim Computer nicht, obgleich dieser doch nur ein physikalisches Gerät ist, dessen Schaltungen der Ursache-Wirkungs-Relation gehorchen? Ich will mich auf eine Antwort (aus vielen möglichen), die auf einen Schlüsselbegriff zusteuert, beschränken: Weil nicht klar ist, was *Bedeutung* ist.

Während für den Toaster noch halbwegs einleuchtet, daß es für diesen keine Bedeutungen gibt (etwa in Form von Vorlieben für Brotsorten), ist dies beim Computer nicht mehr so augenscheinlich. Wer jedoch beim Computer in Abhebung vom Toaster einen prinzipiellen Unterschied meint festzustellen, der muß auch einen *qualitativen* Unterschied des Geräts 'Computer' zum Gerät 'Toaster' herausheben können. Ich werde begründen, warum es diesen qualitativen Unterschied nicht gibt, und ich werde die prinzipielle Gleichheit von Toaster und Computer anhand des Begriffs der Bedeutung entwickeln. Dieser Bedeutungsbegriff wird dann auch nützlich sein bei der Hervorhebung wirklicher qualitativer Unterschiede, so zwischen Toaster und Tier und zwischen Tier und Mensch.

Bedeutungsbegriff

Bedeutung ist eine zentrale Kategorie der Kritischen Psychologie. In Tab. 1 ist der Begriff der Bedeutung im Kontext der jeweils mit ihr verbundenen Aktivitäten (bzw. Handlungen) in seiner Entwicklungslogik schematisch dargestellt - sozusagen die Schematisierung der Kritischen Psychologie durch einen Informatiker! Die Rolle der Wahrnehmung sowie des Bedarfs (bzw. der Bedürfnisse) berücksichtige ich hierbei nicht.

Prozeßebene	Voraussetzung	Rolle der Bedeutung	Resultat	Relation
physikalisch	Ursache	-	Wirkung	deterministisch
unspezifisch-biologisch	gegebene Umwelt	Aktivitätsrelevanz	Aktivität	deterministisch
spezifisch-menschlich	produzierte Welt und unbearbeitete Natur	Handlungsmöglichkeit	Handlung	fakultativ

Tab. 1: Schematische Darstellung der Rolle der Bedeutung für verschiedene Prozeßebenen der Entwicklung nach Auffassung der Kritischen Psychologie.

Die Bedeutungen sind die Vermittlungsinstanz zwischen der Umwelt und dem Tier bzw. der Welt und dem Menschen. Während dieses Verhältnis auf tierischem Niveau deterministischer Natur ist, die Bedeutungen also Aktivitätsdeterminanten sind, bilden die Bedeutungen auf menschlichem Niveau der Entwicklung nurmehr einen Raum von Handlungsmöglichkeiten.

Alle Prozesse beim Toaster und beim Computer sind physikalischer Natur, beide funktionieren nach dem Ursache-Wirkungs-Prinzip, für beide gibt es keine Bedeutung, beide sind folglich eindeutig einzuordnen. Das gilt für den Computer auch dann, wenn die Rolle der Bedeutung als Aktivitätsrelevanz *simuliert* wird. Eine solche Simulation ist möglich, weil auf tierischem Niveau der Bedeutungs-Aktivitäts-Zusammenhang wie der Ursache-Wirkungs-Zusammenhang auf bloß physikalischem Niveau deterministisch ist. Bedeutungs-Aktualisierung und Aktivität fallen beim Tier zusammen. Anders beim Menschen: Zwischen Bedeutungsrealisierung und Handlung liegt eine Erkenntnisdistanz, die es dem Menschen ermöglicht, sich zur Welt und sich selbst bewußt zu verhalten. Bedeutungen sind für Menschen niemals Handlungsdeterminanten, sondern immer nur Handlungsprämissen. Menschen können Wollen, Tiere nicht.

Ohne einen klaren Bedeutungsbegriff ist Verwirrung 'vorprogrammiert'. Ob über Computer vermenschlichend oder über Menschen in Computermetaphern geredet wird, ist keine moralische Frage, sondern eine der Erkenntnis. Informatisierung der Psychologie und Psychologisierung der Informatik sind Ausdruck theoretischer Unklarheit.

Warum ist die Bedeutungskategorie für Psychologie und Informatik wichtig?

Bedeutungen sind Ausdruck der von Menschen geschaffenen sachlich und sozial strukturierten Welt. Sie bilden die Vermittlungsebene zwischen den gesellschaftlichen Lebensbedingungen und dem individuellen Handeln. Die Bedeutungskategorie ersetzt den "Reiz"- bzw. "Input"-Begriff der traditionellen Psychologie. "Reiz" bzw. "Input" waren dort Ausdruck des unmittelbaren Einwirkens der Umwelt auf den Organismus. In dem groben Schema von Tab. 1 entspräche diese Sichtweise der unspezifisch-biologischen Prozeßebene. Die Bedeutungskategorie erst ermöglicht eine Fassung des Mensch-Welt-Verhältnisses als Möglichkeitsbeziehung, in welcher der Mensch Bedeutungen - also Handlungsmöglichkeiten - realisieren, oder es auch lassen kann. Vergleicht man traditionelle Psychologie (in behavioristischer und kognitiver Ausprägung) und Kritische Psychologie, so kann man die kategorialen Verhältnisse - wieder schematisch - wie in Abb. 1. darstellen.

Behaviorismus	Kognitivismus	Krit. Psychologie
Reiz	Eingabe (Input)	Bedeutungen
[Assoziation]	Verarbeitung	Prämissen
Reaktion	Ausgabe (Output)	Intentionen
		Handlungsvorsatz
		Handlung

Abb.1: Schematische Darstellung theoretischer Fassungen 'menschlichen Verhaltens' im Behaviorismus, im Kognitivismus und in der Kritischen Psychologie.

Behaviorismus und Kognitivismus sind Theoretisierungen des deterministischen Ursache-Wirkungs-Modells - im Kern des Modells einer Maschine. Sie verwenden jedoch unterschiedliche Begriffssysteme, aus denen sie ihre suggestive Kraft und methodisches Arsenal ziehen. Während sich der Behaviorismus an der Physiologie orientiert(e), verwendet der Kognitivismus den Computer als Denk- und Theoriemodell. Wer wissen möchte, was Kognitivismus ist, der schlage im "Duden Informatik" unter "EVA-Prinzip" (von Eingabe, Verarbeitung, Ausgabe) nach [1].

In der *Informatik* geht es um die Vergegenständlichung antizipierter Zwecksetzungen in Produktform, wenn man so will, um die Umwandlung symbolischer in gegenständliche Bedeutungen [2]. Was ist damit gemeint? In einem bestimmten Bereich soll durch Computereinsatz - Hard- und Software - ein vorgestellter Zweck erreicht werden. Als Beispiel mag eine Organisation dienen, die Mitgliederdaten zum Zwecke der Buchführung einerseits verwalten und andererseits analysieren möchte, um die Arbeit der Organisation auf die Zielgruppen besser zuschneiden zu können. Diese zwei Ziele sollen in Produktform "gegossen" werden. Die Transformation antizipierter symbolisch-weltlicher Bedeutungen in eine gegenständliche Form ist der Kern informatischer Tätigkeit. Informatik ist, so gesehen, eine *Transformationswissenschaft*: Sie wandelt vorgestellte Bedeutungen in gegenständliche Bedeutungen, in (mehr oder weniger) nutzbare Werkzeuge um. Hierin ist sie klassische Ingenieurwissenschaft, im Unterschied zu dieser ist jedoch in der Informatik die *konkrete* Form des zu realisierenden Produkts weitgehend irrelevant [3].

Für die Transformation der vorgestellten Zwecke in das fertige Produkt besitzt die Informatik Methoden. Methoden in ihrem Sinne sind Regelwerke (Algorithmen), die beschreiben, wie vorzugehen ist. Ein Meilenstein auf dem Weg von der unendlich reichhaltigen bedeutungsvollen Welt zum reduzierten, nur bestimmte Bedeutungen realisierenden Produkt ist die Anforderungsdefinition (als Teil des Pflichtenheftes). Eine Anforderungsdefinition beschreibt exakt, welche Eingaben welche Ausgaben produzieren sollen. Mit einer eindeutig beschriebenen E/A-Relation läßt sich das zugehörige V (der Verarbeitung) zum fertigen EVA-System konstruieren. Die Schwierigkeit besteht darin, die eindeutigen E/A-Relationen zu bestimmen. Hier nun spielt das Problem eines angemessenen Bedeutungsbegriffes eine zentrale Rolle.

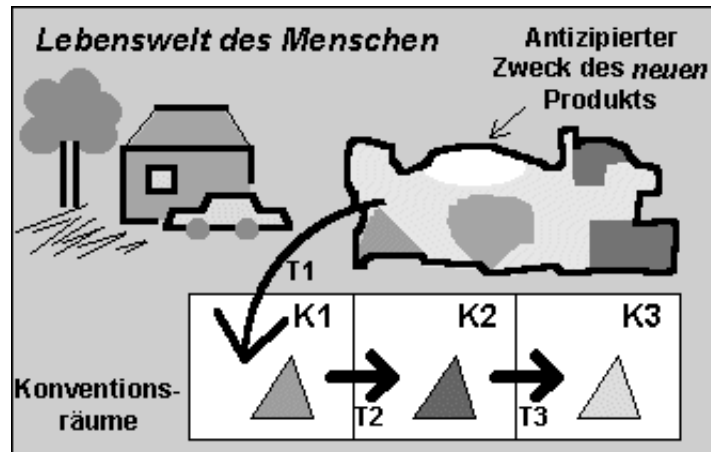


Abb. 2: (Neu-)Bestimmung relevanter Bedeutungen (Transformation T1) zur Erstellung einer Anforderungsdefinition im Konventionsraum K1 und ihre Transformation in weitere Konventionsräume K2, K3 etc. (Abbildung in formale Modelle, Programmierung etc.).

Abb. 2 soll die Aufgabe illustrieren. Mit dem neu zu schaffenden Werkzeug soll ein neuer Zweck erreicht werden. Die allgemeine Beschreibung des Zwecks, im vorangehenden Beispiel die Mitgliederverwaltung und Mitgliederdatenanalyse, muß transformiert werden (T1) in eine exakte Anforderungsdefinition (Dreieck in K1). Diese Definition wird nun über beliebige weitere Stufen (idealerweise) strukturgleich solange transformiert (T2, T3,...), bis am Ende das fertige Produkt steht (Dreieck in K3). Daß sich T1 *qualitativ* von T2 und T3 unterscheidet, ist in der Informatik bestenfalls als Ahnung vorhanden.

Das 'K' steht für Konventionsraum. Innerhalb eines Konventionsraumes gilt eine fest vereinbarte Zuordnung von Zeichen zu Bedeutungen. Technisch geschrieben gilt hier die Abbildung: *Syntax* \Leftrightarrow *Semantik* [4]. Die Konventionen sind wichtig, damit die grafisch oder textuell gefaßten Befehle der Programmiersprache zu dem führen, was vorher festgelegt wurde. Konventionsräume können beliebig komplex sein, sie können in mehrfacher Stufung weitere Sub-Konventionsräume enthalten etc. Jeder Konventionsraum ist jedoch bezüglich der Syntax-Semantik-Abbildung abgeschlossen, ggf. jedoch idealerweise wechselseitig durch regelgeleitete Transformationen ineinander überführbar (Für Programmiervertraute: Assemblieren bzw. Disassemblieren ist eine solche ideale wechselseitige Überführung).

Die Informatik nimmt nun erstens an, daß auch die Bedeutungen der Lebenswelt des Menschen einen 'Konventionsraum' bilden, in dem die Syntax-Semantik-Relationen vereinbart wurden. Sie führt die Bedeutungsentstehung auf eine explizite oder implizit-pragmatisch tradierte *Zuweisung* von Zeichen (z.B. Wörtern) zu Bedeutungen in der Gesellschaft zurück. In technischen Worten: Die Syntax-Semantik-Relationen wurden von der Gesellschaft definiert. Für die Informatik stellt sich mit dieser Vorstellung die Produktentwicklung wie in Abb. 3 veranschaulicht dar.

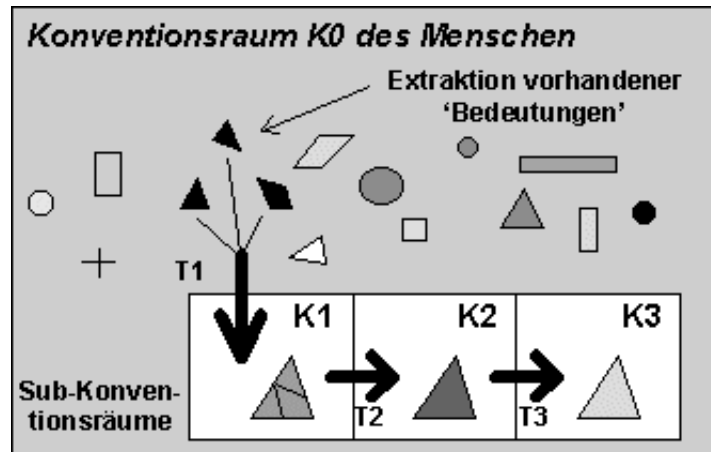


Abb. 3: Extraktion vorhandener Bedeutungen (Transformation T1) aus dem Konventionsraum K0 zur Erstellung einer Anforderungsdefinition im Konventionsraum K1 und ihre Transformation in weitere Konventionsräume K2, K3 etc. (Abbildung in formale Modelle, Programmierung etc.).

Die Informatik kann damit *Neues* im Kern nicht erklären. Transformationen sind immer nur Umsetzungen von Syntax-Semantik- bzw. E/A-Relationen eines Konventionsraum in einen anderen, sie unterscheiden sich folglich qualitativ nicht. Da Transformationen nie Neues generieren, sondern nur Vorhandenes umsetzen (umkodieren), kann Neues nur gedacht werden als Komposition aus vorhandenen Einzelteilen. Mit der Zuweisungshypothese läßt sich die Entstehung neuer Bedeutungen nicht erklären. Damit ist auch die Erklärung der Herkunft der vorhandenen Bedeutungen hinfällig. Im Kern setzt die Zuweisungsannahme nämlich das, was als Bedeutung vereinbart werden soll, bereits voraus. Beispiel Vereinbarung der Bedeutung eines Stuhls: Ein 'Stuhl' ist ein 'Ding-zum-drauf-Sitzen'. 'Stuhl' und 'Ding-zum-drauf-Sitzen' sind hier jedoch nur verschiedene textuelle Formen ein und derselben Bedeutung: Ein Stuhl ist ein Stuhl. Über Stühle kann nur geredet werden, wenn ihre Bedeutung bekannt ist. Die Bedeutung ist nicht durch eine Übereinkunft auf die Welt gekommen, sondern durch die Tatsache ihrer verallgemeinerten Herstellung (ausführlich in Lenz & Meretz, 1993, Kap. 3).

Die zweite Annahme der Informatik ist, daß menschliches Verhalten nach dem EVA-Prinzip funktioniert. Trifft ein Mensch auf den Reiz $x (=E)$, so wird dieser verarbeitet $(=V)$ und das Verhalten $y (=A)$ hervorgerufen. Die Frage ist jetzt: Wie bringe ich, der/die informatische Produktentwickler/in, den 'User' dazu, derart mit dem Produkt umzugehen, daß der antizipierte Zweck des Programms auch erfüllt wird? Wie muß ich, um das obige Beispiel zu verwenden, das Programm gestalten, damit der 'User' auch wirklich Mitgliederdaten verwaltet? Welche 'Reize' oder 'Inputs' muß ich bieten, damit der 'User' diese richtig 'verarbeitet', und dergestalt eine 'Ausgabe' erzeugt (d.h. aktiv wird), daß er das Programm zweckgerecht bedient?

Informatische Produktentwicklung und Rolle der Psychologie

Bei der informatischen Produktentwicklung müssen die *Funktions-* und *Benutzungsdimensionen* des neuen Produkts entwickelt werden. Beide Aufgaben können unter zwei verschiedenen Perspektiven angegangen werden, der *Werkzeug-* und der *Modellierungsperspektive*.

Die Realisierung der Funktionsaspekte des neuen Produkts ist eine klassische Ingenieurstätigkeit. Hier entscheidet sich, ob das Produkt seine antizipierte Zweckbestimmung erreicht, oder ob es eben seine Funktion nicht erfüllt. Während früher mit der Funktionslogik des Systems weitgehend auch die

Bedienlogik der Maschine bestimmte, fallen heute Funktions- und Benutzungsdimensionen auseinander. Seit dem Paradigmenwechsel in der Betriebssystemtechnik, ausgelöst durch die Firma *Apple*, hat sich die Sachlage geradezu verkehrt. Der 'User' befolgt nicht mehr nur die Anweisungen der Maschine, sondern er entscheidet nun selbst, was zu tun ist - zumindest der Möglichkeit nach. Eine durch den 'User' bestimmte Benutzungsweise steht jedoch häufig im Widerspruch zu vergegenständlichten Funktionslogiken von Maschinen und Bürokratien [5]. Die Entwicklung der Benutzungsdimensionen informatischer Produkte sind an eine Spezialdisziplin, die "Systemergonomie", überwiesen worden. Hier wird "Benutzung" als äußerliche Oberflächeneigenschaft der Produkte behandelt, die im übrigen nur wenig mit dem funktionalen Kern zu tun hat. So ist es nicht weiter verwunderlich, daß die Potenzen nutzerbestimmter Arbeitsplätze kaum ausgeschöpft sind.

Die Blickrichtung auf die neue Sache, die Perspektive, überlagert den gesamten Entwicklungsprozeß, leitet mithin auch die Entwicklungsrichtung der Dimension der Funktion und der Benutzung. In der Werkzeugperspektive steht die neu zu schaffende Bedeutung (das neue Produkt) im Zentrum der Aufmerksamkeit, in der die vorhandenen Syntax-Semantik-Relationen die Rolle von Konstituenten der neuen Bedeutung spielen. Trotz der verkürzten Bedeutungskonzeption ist sie weit realitätsgerechter als die Modellierungsperspektive. Hier geht es darum, vorhandene Sachverhalte computertechnisch nachzubilden, um vorhandene Bedeutungen zu simulieren. Das Produkt wird hier nicht als *Mittel* für den 'User' gesehen, sondern es dient als *Ersatz* für vermutete Eigenschaften des 'Users' (oder Arbeitsprozessen etc.). In der Praxis koexistieren beide Perspektiven oftmals unerkannt im Entwicklungsprozeß.

Dimension	Werkzeug-Perspektive	Modellierungs-Perspektive
Funktion	Anforderungserhebung Managementmethoden	selbsttätige Agenten mit quasi-menschlichen Fähigkeiten
Benutzung	Benutzermodelle ("Durchschnitts-User") Ergonomieforschung	Mensch als kognitives (EVA-) System

Tab. 2: Rolle der Psychologie für verschiedene Entwicklungsdimensionen und -perspektiven.

Die Psychologie spielt je nach Entwicklungsdimension und -perspektive sehr unterschiedliche Rollen (vgl. Tab. 2). Für die Entwicklung der funktionalen Dimension in der Perspektive der Werkzeugentwicklung bietet sie Theorie- und Methodenversatzstücke, das implizite und explizite Wissen der (zukünftigen) Nutzer/innen und Expert/inn/en zu erheben. Sie bietet Begriffe und Konzepte für das Projektmanagement etc. Dabei werden meist nicht direkt psychologische Theorien/Konzepte angewendet, sondern diese werden überarbeitet, den spezifischen Anforderungen angepaßt und 'methodisiert'. Nicht selten geht dabei die ursprüngliche Intention des Theorieansatzes verloren - was aber eher nützlich ist als schadet [6]. In der Perspektive der Modellierung verschärft sich der Einfluß des kognitionspsychologischen Mainstreams. Sobald es um die Modellierung von (vermuteten) Fähigkeiten des Menschen (oder von Tieren, hier wird nicht unterschieden) geht, kommen die computerkompatiblen Ansätze voll zu Tragen. Aus dem Computer als Werkzeug oder Medium wird ein selbsttätiger Agent, der, ausgeschmückt durch allerlei psychologische Termini, nun quasi-menschliche Fähigkeiten besitzt. Die Modellierungsperspektive, in der es noch einen gibt, der 'draußen' steht (und modelliert), kippt auf diese Weise um in eine Innenperspektive 'des Systems', in der es 'draußen' nicht mehr gibt. Die Mensch-Maschine-'Interaktion' ist so schlicht eine 'Kommunikation' verschiedener EVA-Systeme: Es gibt ja nur noch EVA-Systeme. Dem zugrunde

liegt, wie ich versucht habe zu zeigen, neben einer auf Maschinen reduzierten Vorstellung vom Menschen ein völlig unklares und untaugliches Bedeutungskonzept - so die Frage der Bedeutungen überhaupt ein Thema für die Kognitionstheorie ist [7].

Auch in der Entwicklung der Benutzungsdimension spielt der Kognitivismus eine beachtliche Rolle. Aufgrund der widerständigen Realität werden hier weit realitätsnähere Konzepte entwickelt. Sie stehen jedoch meist im Banne des WENN-DANN-Bedingtheitsmodells (Ursache-Wirkungs-Determinismus) der kognitiven Theorien. In der Regel wird das Werkzeug nur in seiner äußeren Form und seiner Wirkung auf den User untersucht, da Funktions- und Benutzungsdimension des Produkt als unabhängige Aspekte behandelt werden. Die Einbettung in einen Arbeitsprozeß, die Arbeitsorganisation und Verfügung über die Bedingungen des Arbeitens sowie allgemeine Arbeitsbedingungen bleiben meist ausgeblendet.

Fazit

Die theoretischen Verkürzungen in Psychologie und Informatik spiegeln sich in ihren jeweiligen Methoden wider. In der Psychologie führt ein *fehlender oder unangemessener Subjektbegriff* zu methodischen Konzepten des Außenstandpunkts, in den Menschen nur als unter Bedingungen stehend gesehen werden (können) - klassisch abgebildet in der sog. "experimentellen Standardanordnung". In der Informatik beginnt das Problem mit einem *fehlendem oder unangemessenem Bedeutungsbegriff* sozusagen bereits eine Ebene tiefer. Sie ist nicht in der Lage, Menschen und Maschinen begrifflich-analytisch voneinander zu unterscheiden. Es ist klar, daß damit auch der 'User' als Subjekt kategorial nicht erreicht werden kann, eine Eigenschaft, die durch die Konzepte der kognitiven Psychologie nur verfestigt wird. Folglich taucht der 'User' in den informatischen 'Toolsets' nicht mehr auf oder diffundiert als Bestandteil in das System hinein [8]. Die Verwendung psychologischer Termini ist nur schmückendes Beiwerk, um den konzeptionellen 'Subjektverlust' auf der Sprachebene zu kompensieren. Meistens wird jedoch selbst darauf verzichtet.

Literaturverzeichnis

Lenz, A. & Meretz, S. (1995), *Neuronale Netze und Subjektivität*, Braunschweig/Wiesbaden: Vieweg.

Negroponce, N. (1995), *Total Digital. Die Welt zwischen 0 und 1 oder: Die Zukunft der Kommunikation*, München: Bertelsmann.

Seidel, R. (1994), Maschinenperspektive und Subjektstandpunkt, *Forum Kritische Psychologie*, 34, S. 112-124.

Fußnoten:

↑ [1] Rainer Seidel betont daher zurecht, daß die Auffassung, "wonach der Kognitivismus den Computer als 'Metapher' benutze, untertreibend bzw. unpräzise (sei): Kognitivismus *ist* die Theorie der Berechenbarkeit des Menschlichen." (1994, S. 116).

↑ [2] Ich gehe von einem weit gefaßten Gegenstandsbegriff aus: Auch ein Text auf einer World-Wide-Web-Seite ist für mich ein "Gegenstand".

↑ [3] Die Tatsache, daß leider doch immer wieder die (mangelhafte) *konkrete* Technik "durchschlägt", mag für viele Informatiker/innen geradezu erleichternd wirken - kann man sich doch in aller Tiefe mit der Hard- und Software beschäftigen. Daß die konkrete Maschine für das Wesen der Informatik irrelevant ist, ist jedoch seit der Turing-Maschine, die ja nur auf dem Papier bestand, hinlänglich bekannt. So ist beim Verstehen von Texten das Buchstabenentziffern genauso

vernachlässigbar - es sei denn, es ist sehr dunkel. In der Hard- und Softwaretechnik sieht es danach noch ziemlich finster aus ...

↑ [4] Allgemeiner müßte die Abbildung lauten: *Form Bedeutung*. Da es die Informatik häufig mit Zeichensystemen (Texten o.ä.) zu tun hat, spricht sie selber in der Regel über das Thema als "Syntax-Semantik-Problem". Die Informatik ringt vor allem um die Abbildung *Syntax Semantik*, also um die programmiersprachliche Konstruktion von zweckadäquater Software (Hardware); die Umkehrrichtung, das Reengineering, spielt jedoch eine immer wichtigere Rolle. Ob Reengineering überhaupt möglich ist, sei an dieser Stelle außen vor gelassen.

↑ [5] Die Verwendung sogenannter Standard-Software als Alternative zur Eigenentwicklung ist ein Versuch, wenigstens den Problemkreis "der/die Nutzer/in, das unbekannte Wesen" ausblenden zu können. In Wirklichkeit handelt es sich jedoch nur um eine schlechte Kaschierung des Problems, das an anderer Stelle wieder auftaucht (als 'Konfigurationsproblem etc.).

↑ [6] Nützlich in dem Sinne, das umfangreiche Projekterfahrung und das übliche Maß Alltagspsychologie einem Projekt mehr bringen, als der Versuch der algorithmischen Anwendung traditionell-psychologisch fundierter Methoden.

↑ [7] Inhaltlich wird manchmal die Bedeutungsthematik in reduzierter Weise unter dem Thema 'Informationsgewinnung' diskutiert. Der Informationsbegriff wird (nicht nur im Mainstream) jedoch über die Zuweisungstheorie definiert, womit er seinem analytischen Potential wieder beraubt ist.

↑ [8] In extremer Form findet man dies in Entity-Relationship- und Funktions-Modellen, auf denen die meisten CASE-Tools (Computer Aided Software Engineering) basieren. Aber auch die objektorientierten Methoden schaffen es bestenfalls, den User als Homunculus in das System einzuschließen.