

# Wissenschaftliche Grundlagen der angewandten Informatik

## in memoriam Tadeusz Kasprzak

Christian Dahme

### 1. Vorbemerkungen

Will man die wissenschaftlichen Grundlagen der Informatik beziehungsweise der Softwareentwicklung ergründen, so sind mindestens drei Richtungen oder Sichtweisen zu unterscheiden:

- A) Aus der Sicht des **Anwenders**  
was mit dem Computer unterstützt beziehungsweise durch den Computer (wünschenswerter Weise) realisiert werden sollte
- B) Aus der Sicht des **Informatikers** (des Soft- oder Hardwareentwicklers),  
was aus dieser Sicht mit einem Computer möglich ist oder wie Soft- beziehungsweise Hardware zu konstruieren ist, um eine vorgegebene Spezifikation zu erfüllen.  
(Diese Sicht entspricht der so genannten „Kerninformatik“.)
- C) Aus der Sicht des „**Interface-Designers**“  
(Im klassischen Ingenieurbereich würde man diese Richtung der industriellen Formgestaltung zuordnen.)

Aus Sicht der Softwareentwicklung könnte man A) den frühen Phasen zuordnen, B) der Softwarekonstruktion und C) der Einbettung der Software in den Anwendungszusammenhang.

Dabei geht es in C) darum, ausgehend von A) und B) einen Computer als Werkzeug so zu gestalten, dass dieser eine gute virtuelle Nachbildung der realen Situation des Anwenders darstellt. Hier spielen Begriffe wie „usability“, brauchbare Software, Benutzerfreundlichkeit, Fehlerfreundlichkeit u.dgl. eine Rolle.

Im Folgenden werde ich mich überwiegend auf die Anwendersicht und damit auf die frühen Phasen der Softwareentwicklung beschränken. Statt Anwendersicht könnte man auch angewandte Informatik sagen.

Der Anspruch besteht hier darin, einen Computer zu bekommen (erfinden zu lassen – im Sinne von B) aber auch unter Einbeziehung von C)), der den Wünschen eines Anwenders (Auftraggebers) entspricht. Aus dieser Sicht kann man den Computer als ein Werkzeug zur Unterstützung realer oder möglicher (menschlicher) Tätigkeit verstehen.

In diesem Zusammenhang sind unter anderem folgende Fragen von Interesse:

Wie kommt man zu einem solchen Computer?

Was sind die konzeptionellen Grundlagen?

Was könnte sich potentiell durch einem Computer unterstützen lassen?

Was wäre wünschenswert?

Wo sind die Grenzen?

### 2. Wurzeln der angewandten Informatik

Im Folgenden wollen wir uns möglichen konzeptionellen Grundlagen zuwenden. Wissenschaftshistorisch lassen sich drei Wurzeln identifizieren:

- a) Der Computer zur Unterstützung bis hin zur Automatisierung des Rechnens

Speziell:

Der Computer als Mittel zur Unterstützung von wissenschaftlichen und kommerziellen Berechnungen auf der Basis mathematisierter Modelle  
Letzteres werde ich als **gegenstandsorientierten Modellansatz** bezeichnen.

- b) Automatisierung von (Teilen von) Tätigkeiten durch Übertragung an einen Computer  
Diese Wurzel bezeichne ich als **tätigkeitstheoretisch orientierten Ansatz**.
- c) Psychische Phänomene werden als „Informationsverarbeitung“ rekonstruiert  
Dieses führt zum **kognitivistischen Ansatz**.

## 2.1. Der gegenstandsorientierte Modellansatz

Wissenschaftshistorisch ist dieser Ansatz meines Erachtens die älteste Wurzel. Hier geht es insbesondere darum, das Rechnen zu erleichtern.

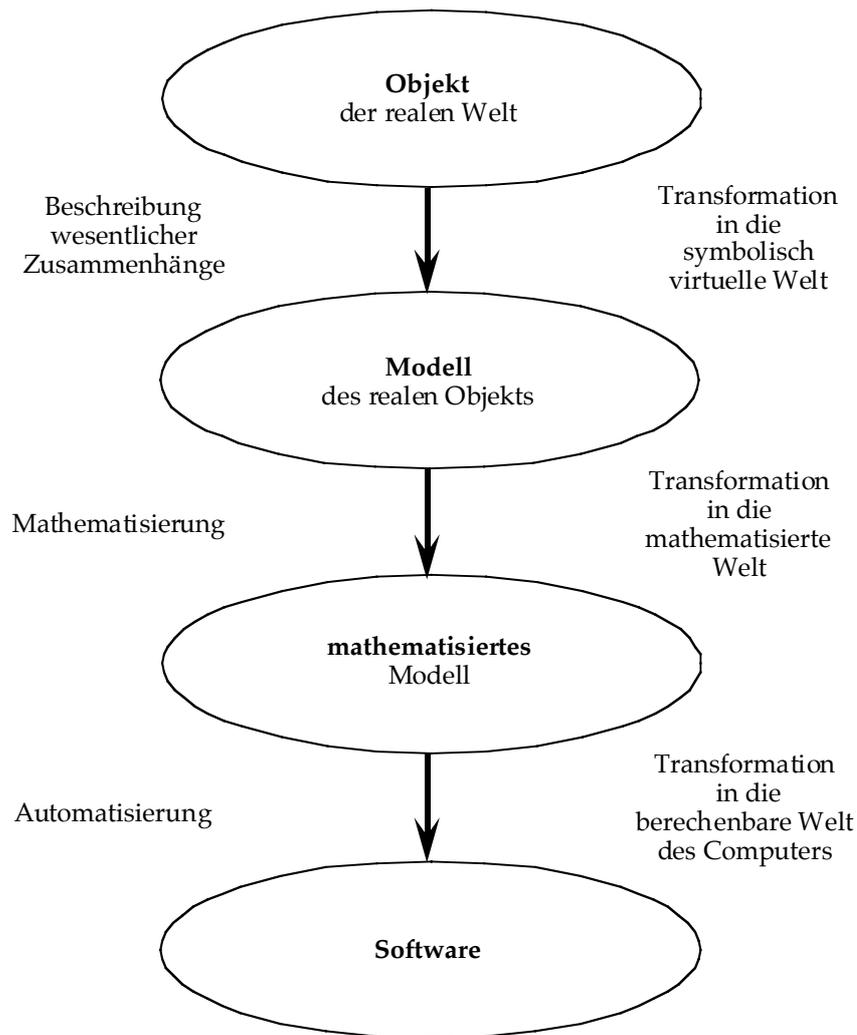
Ein wesentliches Anliegen aber auch eine treibende Kraft für die Entwicklung von "Rechenwerkzeugen" später Rechenautomaten, bestand darin, aufwendige Berechnungen zu unterstützen und Rechenfehler zu vermeiden.

Mit dem Aufblühen der Naturwissenschaften im 17. Jahrhundert entstand verstärkt das Bedürfnis, die uns umgebende beobachtbare Welt (Wirklichkeit) zu erklären und gegebenenfalls zu prognostizieren. Gleichzeitig wuchs damit die Bedeutung mathematisierter Modelle.

So konstruierte 1623 Wilhelm Schickard eine Rechenmaschine, um Johannes Kepler das Berechnen seiner Modelle zu erleichtern. Diese Richtung führte zur Unterstützung bis hin zur Automatisierung der Berechnung mathematisierter Modelle der modernen Physik und Naturwissenschaften und diente damit auch als Grundlage für die Entwicklung von Raketen, Atombomben, Raumfahrt, Computertomographie u. dgl. im Sinne von angewandten Naturwissenschaften.

Andererseits gab es schon sehr lange das Bedürfnis kommerzielle Berechnungen zu unterstützen. So entwickelte Blaise Pascal 1642 seine Rechenmaschine „Pascaline“, um die Verwaltungsarbeit seines Vaters zu erleichtern. Charles Thomas - Chef zweier Versicherungsgesellschaften in Paris – kam auf die Idee, eine Rechenmaschinenproduktion aufzubauen. Grundlage hierfür war das von ihm 1820 fertig gestellte "Arithmometer", von dem ungefähr 1500 in seiner Werkstatt produziert wurden. Diese Richtung führte ebenfalls zur Unterstützung bis hin zur Automatisierung der Berechnung (normativer) ökonomischer Modelle (einschließlich Operations Research und Ökonometrie) wie z.B. Weltmodellen des Club of Rom oder Modelle im Rahmen der Projekte des International Institute for Applied Systems Analysis - IIASA . Solche Modelle sind heute eine wesentliche Basis der modernen Ökonomie.

Aus naturwissenschaftlicher Sicht lässt sich dieser Ansatz durch folgende Transformation beschreiben:



In diesem Kontext kann **Software** als Automatisierung der Berechnung von mathematisierten Modellen verstanden werden. Grundlage für die Software ist hier ein Modell des Gegenstandes<sup>1</sup> – daher gegenstandsorientierter Modellansatz -, und damit die Disziplin des Gegenstandsbereiches, zu dem der Gegenstand gehört. Folglich ist aus dieser Sicht Softwareentwicklung ein interdisziplinärer Prozess.

## 2.2. Der tätigkeitstheoretisch orientierte Ansatz<sup>2</sup>

In diesem Ansatz steht folgende Frage im Mittelpunkt: Welcher Anteil einer Tätigkeit lässt sich potentiell automatisieren und hat damit die Chance, von einem Automaten übernommen zu werden?

Wissenschaftshistorisch geht dieser Ansatz insbesondere auf die Automatisierung des Webens<sup>3</sup> (1728 Lochplatte von Falcon, 1808 Jacquardscher Webstuhl) zurück, mit der gleichzeitig das für die Informatik wichtige Prinzip der "Programmsteuerung" erfunden wurde.

Ausgehend von diesem Prinzip entwickelte

- Charles Babbage (um 1830) mit der Analytical Engine schon das Grundprinzip des Computers,

<sup>1</sup> siehe hierzu: Dahme 1997, insbesondere das 1. Kapitel, aber auch Dahme 2001

<sup>2</sup> siehe: Dahme, Ch.; Raeithel, A. 1997, S. 5-12

<sup>3</sup> Automatisieren des Anhebens der Kettfäden zur Musterbildung

- Herman Hollerith ein elektromechanisches Lochkartensystem für die Auswertung der Volkszählung in den USA von 1890 - im Sinne von Datenerfassung, -bereitstellung und -auswertung insbesondere großer Datenmengen<sup>4</sup> - das später unter anderem zur Entstehung von Datenbanken und Informationssystemen führte.

Bei all diesen Erfindungen ging es darum, Teile einer Tätigkeit einem Automaten zu übertragen unter Nutzung von Software.

In diesem Kontext kann Software wie folgt interpretiert werden:

### **Software**

- als Automatisierung von operationalisierbaren Anteilen (innerer) menschlicher Tätigkeit,
- als Resultat der Transformation von Anteilen realer oder möglicher menschlicher Tätigkeit in eine maschinelle Form,
- als Vergegenständlichung von menschlichen Fähigkeiten.

Grundlage für die Softwareentwicklung ist hier die Tätigkeitstheorie.

Aus dieser Sicht könnte man die Tätigkeitstheorie als eine "Grundlagendisziplin" für die Softwareentwicklung ansehen.

In diesem Zusammenhang ergeben sich zwei Fragen:

1. Welcher Anteil einer Tätigkeit hat die Chance in Software übertragen zu werden?
2. Wie kommt man von diesem Anteil zu einem Automaten, der diesen realisiert?

Die erste Frage wurde in „Ein tätigkeitstheoretischer Ansatz zur Entwicklung von brauchbarer Software“ (Dahme, Ch.; Raeithel, A. 1997, S. 6) weitgehend beantwortet.

Folgende Transformation ist dafür notwendig:

1. Der Anteil, der die Potenz hat, sich in Software transformieren zu lassen, bezieht sich auf Anteile einer inneren, orientierenden Tätigkeit oder lässt sich in solche transformieren.
2. Diese Anteile liegen als Operationen vor oder lassen sich von Handlungen in Operationen überführen (Operationalisierbarkeit).
3. Es liegt Wissen vor, mit dem sich diese Operationen (vollständig) reproduzieren lassen – reproduzierbares Wissen.
4. Dieses Wissen ist (vollständig) mitteilbar.
5. Es kann in öffentliches Wissen überführt werden.

Die Frage, wie man zu einem Automaten kommt, ist dann aus meiner Sicht ein Ingenieursproblem, bei dem es darum geht, einen Automaten zu erfinden (hier eine Software zu entwickeln), die diesen Teil der Tätigkeit realisiert. Das ist überwiegend ein kreativer Prozess.

### **2.3. Der kognitivistische Ansatz<sup>5</sup>**

Wissenschaftshistorisch ist diese Richtung am jüngsten. Sie entstand um 1956 am MIT, ist eng mit der Kybernetik verbunden und wird in der Psychologie auch als kognitive Wende bezeichnet. Charakteristisch hierfür ist das so genannte kognitivistische Paradigma: Psychische Phänomene lassen sich als „Informationsverarbeitung“ rekonstruieren. Diese Richtung basiert auf der Metapher des Computermodells des Geistes. Sie wird später als Kognitivismus bezeichnet.

Ein spezielles Interesse bestand in der Erforschung des Problemlösens. Das führte unter anderem zum General Problem Solver - GPS - aber auch zur künstlichen Intelligenz - KI. Wenn Handeln, wie beim GPS und im Kognitivismus, als Problemlösen und Problemlösen als das Finden eines Weges von einem Ausgangszustand zum Zielzustand verstanden

<sup>4</sup> Ausgehend von den Erfahrungen bei der Volkszählung wurde dieses Verfahren von staatlicher bis hin zu betrieblicher Verwaltung wie unterschiedlichste Statistiken, Lagerhaltung, Kostenerfassung oder Lohnabrechnung angewandt.

<sup>5</sup> siehe G. Pickert „Der kognitivistische Ansatz in der angewandten Informatik“ 2007

wird, dann ist hier Problemlösen das erfolgreiche Suchen in diesem Problemraum. Dabei wird davon ausgegangen, dass alle Operatoren des Problemraumes bekannt sind und alle Zwischenzustände sich gegebenenfalls damit ermitteln lassen.<sup>6</sup>

Genau diesen Problembegriff unterstellen auch die theoretische Informatik und die KI. **Software** wird folglich als Automatisierung eines Algorithmus, der wiederum selbst eine Problemlösung (im obigen Sinn) repräsentiert, verstanden.

Diese Art der Charakterisierung von Software ist heute weit verbreitet und wird gerne in der theoretischen Informatik bevorzugt, da sie explizit auf dem Begriff des Algorithmus basiert.

## 2.4 Der Zusammenhang zwischen diesen Ansätzen

Der kognitivistische Ansatz kann als ein spezieller gegenstandsorientierter Modellansatz der kognitiven Psychologie verstanden werden (siehe Klix 1971, G. Pickert 2007).

Im meinem Buch „Systemanalyse menschlichen Handeln“ habe ich gezeigt, wie sich der tätigkeits-theoretisch orientierte Ansatz in einen gegenstandsorientierten Modellansatz (überführen und) einbetten lässt.

Das führt zu der Aussage: Software basiert auf einem Modell, unabhängig davon, ob man sich dessen bewusst ist oder nicht.

Aus Sicht der Tätigkeitstheorie sind gegenstandsorientierte Modelle Gegenstand oder Mittel innerer orientierender Tätigkeit. Auf diese Weise lässt sich der gegenstandsorientierte Modellansatz tätigkeits-theoretisch einbetten.

Beide Ansätze sind demnach gleichberechtigt.

## 3. Unterstützung von Entscheiden als exemplarisches Beispiel<sup>7</sup>

Bei der Erforschung der Unterstützung von Entscheidungssituationen können alle drei Ansätze eine Rolle spielen.

Entscheidungssituationen sind dadurch charakterisiert, dass es mehrere Alternativen oder Handlungsmöglichkeiten (Strategien) gibt, um ein Ziel zu erreichen.

Es geht nun darum, sich für eine (geeignete) Strategie beziehungsweise Alternative zu entscheiden. Dieses kann auf intuitive Weise, durch Nutzung von Erfahrungen, durch Delegieren auf den Zufall oder wissenschaftlich vorbereitet (unter anderem durch bewusste Bildung von Varianten mit Hilfe von Modellen und Prüfung dieser auf ihre Eignung) erfolgen.

Nur die wissenschaftlich vorbereitet Entscheidung werde ich im Folgenden betrachten. Im Rahmen der Bewältigung einer solchen Entscheidungssituation können folgende Situationen auftreten

### a) **Problemsituation oder Orientierungsphase**

Sie ist dadurch charakterisiert, dass uns bewusst wird, dass wir nicht genügend über eine Situation wissen, um sie bewältigen zu können.

Die Beschreibung dieses Nichtwissens, dieser Defizite in unserem Wissen wird auch Problembeschreibung genannt.

Im Falle einer Entscheidungssituation kann sich das Wissensdefizit auf das Ziel oder den Gegenstand der Entscheidung aber auch auf Handlungen, die das Ziel ermöglichen, einschließlich von Kriterien zur Bewertung solcher Handlungsmöglich-

---

<sup>6</sup> Dörner (1987) spricht in solchen Fällen statt von Problem von Aufgabe. Sind dagegen Operatoren nicht bekannt, dann liegt nach Dörner ein Problem vor. Solche Operatoren müssen dann synthetisiert werden (z.B. durch Entdecken oder Analogiebildung). Dieses ist in der Regel ein kreativer Prozess.

<sup>7</sup> Das ist eine Thematik, die uns, Tadeusz Kasprzak und mich, immer wieder beschäftigt hat. Daher möchte ich sie hier als ein exemplarisches Beispiel benutzen.

keiten, beziehen.

Eine solche Situation ist "bewältigt", wenn das Problem beschrieben/formuliert ist.

**b) Problembearbeitungssituation oder Problemlösungsphase**

Sie ist dadurch charakterisiert, dass es ein formuliertes Problem gibt, das wir lösen wollen, uns jedoch eine Methode zur Lösung des Problems nicht oder nur unvollständig bekannt ist.

Hier geht es darum: Wie löse ich das Problem?

Eine solche Situation ist "bewältigt", wenn das Problem gelöst ist.

**c) Entscheidungssituation (im engeren Sinn)**

Hat man aus der ursprünglichen Entscheidungssituation alle Anteile, die einer Problemlösung zugeführt werden konnten, herausgefiltert, so bleibt in der dann entstehenden Entscheidungssituation im engeren Sinn nur noch ein subjektiv entscheidbarer Anteil übrig - die eigentliche Entscheidung!

Hier geht es darum: Für welche Alternative entscheide ich mich (im Sinne einer subjektiven Wertung)?

Eine solche Situation ist "bewältigt", wenn eine Alternative/Variante gewählt wurde.

**d) Realisierungssituation**

Sie ist dadurch charakterisiert, dass die ausgewählte Variante umzusetzen ist.

Eine solche Situation ist "bewältigt", wenn diese Variante umgesetzt ist.

Bei der Umsetzung kann es passieren, dass erneut eine Entscheidung zu treffen ist, so dass man in Bezug auf die Umsetzung wieder bei der Orientierungsphase landet.

Inwieweit können nun die obigen Ansätze die Bewältigung der Situationen in a) bis c) unterstützen?

1. Bildung von Varianten in Bezug auf den Gegenstand der Entscheidung  
Hier hat der Modellansatz seine Stärken.  
Dabei geht es um die Bildung/die Erzeugung/das Finden von Alternativen insbesondere mit Hilfe von Modellen (einschl. Simulation).  
Die Unterstützung bezieht sich hier nur auf b).  
In Bezug auf die Bildung von Varianten mit Hilfe von Modellexperimenten (Simulation) wird vollständige Mittelbarkeit und Reproduzierbarkeit vorausgesetzt.
2. Bildung von Handlungsmöglichkeiten zur Erreichung des Zieles  
Es geht hier um folgende Frage: Welche Handlungsmöglichkeiten gibt es, um von einer Ausgangssituation zum Ziel zu kommen? Unterstellt man, dass alle Handlungsmöglichkeiten im Sinne eines Problemraumes (siehe 2.3.) bekannt sind beziehungsweise konstruiert werden können, dann geht es nur noch um das Suchen nach beziehungsweise Finden von geeigneten Wegen von einer Ausgangssituation zu einer Zielsituation in einem solchen Problemraum.  
Dieses führt zum kognitivistischen Ansatz.  
Die Unterstützung bezieht sich hier auch nur auf b).
3. „Entscheiden“ mit Hilfe von Entscheidungsmodellen einschl. Optimierung beziehungsweise Spieltheorie  
Hier wird Entscheiden auf Problemlösen (im Sinne von b)) zurückgeführt.  
Wenn eine eindeutige Lösung gefunden werden kann, entfällt eine Entscheidung (im Sinne von c)). Stattdessen wird eine Aufgabe im Sinne von Dörner (1987) gelöst. Dieses führt indirekt zum kognitivistischen Ansatz.  
Die Unterstützung bezieht sich, wie beim kognitivistischen Ansatz nur auf b).  
Hier wird die Entscheidung auf die Auswahl von Modellen und Verfahren vorverlagert. Inwieweit diese jedoch tatsächlich geeignet sind, wird häufig nicht problematisiert.

4. Sind folgende Fragen von Bedeutung:  
 Was ist (von einer Handlung/ Tätigkeit) operationalisierbar?  
 Wie kann man den nicht operationalisierbaren, den nicht subjektfreien, den wertungsbezogenen Anteil berücksichtigen?  
 Was ist automatisierbar?  
 Was ist automatisierungswürdig?  
 dann sollte man den tätigkeitstheoretisch orientierten Ansatz bevorzugen.  
 Er unterstützt b) und berücksichtigt c).  
 Leider spielen in der Informatik die wertungsbezogenen Anteile oft nur eine untergeordnete Rolle. Wie unter anderem bei UML2 deutlich wird, gehen sie im Laufe des Spezifikationsprozesses verloren (siehe G. Müller 2007).

An diesem exemplarischen Beispiel kann man schon deutlich erkennen, dass alle drei Ansätze unterschiedliche Perspektiven beleuchten, so dass es sinnvoll erscheint, sich nicht nur auf einen Ansatz zu stützen, um größere Probleme zu vermeiden.

## Literatur

- Coy, W. (Hrsg.), Sichtweisen der Informatik Braunschweig, Wiesbaden: Vieweg 1992
- Coy, Wolfgang und Bonsiepen, Lena. 1989. Erfahrung und Berechnung. Kritik der Expertensystemtechnik. Berlin : Springer, 1989.
- Dahme, Ch.; Raeithel, A.: Ein tätigkeitstheoretischer Ansatz zur Entwicklung von brauchbarer Software, in: Informatik-Spektrum · 20 · Heft 1 · Februar 1997, siehe auch: „Modele informacyjne procesów gospodarczych“ red. Tadeusz Kasprzak, wyd. WNE UW, Warszawa, 1998
- Dahme, Ch.: Systemanalyse menschlichen Handelns - Grundlagen und Ansätze zur Modellbildung, Westdeutscher Verlag 1997, Opladen
- Dahme, Ch.: Wissenschaftstheoretische Positionen in bezug auf die Gestaltung von Software. In: Fuchs-Kittowski, K. / Parthey, H. / Umstätter, W. / Wagner-Döbler, R. (Hrsg.): Organisationsinformatik und Digitale Bibliothek in der Wissenschaft: Wissenschaftsforschung Jahrbuch 2000. Berlin: Gesellschaft für Wissenschaftsforschung 2001. S. 167-178
- Dahme, Ch.: Methodologische und theoretische Voraussetzungen für die Analyse komplexer Entscheidungssituationen, Diss. B, Humboldt-Universität zu Berlin, 1987
- Floyd, C. et al. (Hrsg.): Software Development and Reality Construction, Berlin: Springer 1992.
- Dörner, Dietrich: Problemlösen als Informationsverarbeitung. Stuttgart etc. : Kohlhammer, 1987. – 3. Aufl.
- Dörner, Dietrich ; Bick, Thomas: Lohhausen vom Umgang mit Unbestimmtheit und Komplexität. Bern etc. : Hans Huber, 1983. – 2. Aufl.
- Kasprzak, Tadeusz: Entscheidungssysteme in der Leitung eines Betriebes. In: Wissenschaftswissenschaftliche Beiträge, Heft 1, Berlin 1978, S. 157 - 176
- Kasprzak, Tadeusz (Red.nauk.): Badania operacyjne i systemowe Teorie Decyzji, Warszawa, WNE UW, 1987
- Kasprzak, Tadeusz (Red.nauk.): Modele informacyjne procesów gospodarczych, Warszawa, WNE UW, 1998
- Klix, Friedhart: Information und Verhalten. Berlin : Dt. Verlag der Wissenschaft, 1971. – 4. Aufl
- Klix, Friedhart: Erwachendes Denken eine Entwicklungsgeschichte der menschlichen Intelligenz. Berlin : Dt. Verlag der Wissenschaft, 1980. – 1. Aufl.
- Müller, G.: Anwenderorientierte Softwarespezifikation mit der UML 2 aus tätigkeitstheoretischer Sicht, Diplomarbeit, Humboldt-Universität zu Berlin, Institut für Informatik, 2007
- Newell, Allen ; Simon, Herbert A.: GPS, A Program that Simulates Human Thought. In: Billing, H. (Hrsg.): Lernende Automaten. München : Oldenbourg, 1961, S. 109–124
- Newell, Allen ; Simon, Herbert A.: The Theory of Human Problem Solving. In: Collins, A. (Hrsg.) ; Smith, E. E. (Hrsg.): Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence. San Mateo : Kaufmann, 1988, S. 33–51
- Pickert, G.: Der kognitivistische Ansatz in der angewandten Informatik, Diplomarbeit, Humboldt-Universität zu Berlin, Institut für Informatik, 2007
- Raeithel, A., Selbstorganisation, Kooperation, Zeichenprozess. Hrsg. v. Ch. Dahme. Opladen: Westdeutscher Verlag 1998.
- Turing, A. M.: Computing Machinery and Intelligence. In: Collins, A. (Hrsg.) ; Smith, E. E. (Hrsg.): Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence. San Mateo : Kaufmann, 1988, S. 6–19