

Können AI-Programme als Theorien betrachtet werden? Ein wissenschaftsphilosophischer Beitrag

Manhart, Klaus

Veröffentlichungsversion / Published Version

Sammelwerksbeitrag / collection article

Empfohlene Zitierung / Suggested Citation:

Manhart, K. (1989). Können AI-Programme als Theorien betrachtet werden? Ein wissenschaftsphilosophischer Beitrag. In J. Retti, & K. Leidlmair (Hrsg.), 5. *Osterreichische Artificial-Intelligence-Tagung, Igls/Tirol, 28.-31. März 1989: Proceedings* (S. 346-358). Berlin u.a.: Springer. <https://nbn-resolving.org/urn:nbn:de:0168-ssoar-58499>

Nutzungsbedingungen:

Dieser Text wird unter einer Deposit-Lizenz (Keine Weiterverbreitung - keine Bearbeitung) zur Verfügung gestellt. Gewährt wird ein nicht exklusives, nicht übertragbares, persönliches und beschränktes Recht auf Nutzung dieses Dokuments. Dieses Dokument ist ausschließlich für den persönlichen, nicht-kommerziellen Gebrauch bestimmt. Auf sämtlichen Kopien dieses Dokuments müssen alle Urheberrechtshinweise und sonstigen Hinweise auf gesetzlichen Schutz beibehalten werden. Sie dürfen dieses Dokument nicht in irgendeiner Weise abändern, noch dürfen Sie dieses Dokument für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, aufführen, vertreiben oder anderweitig nutzen.

Mit der Verwendung dieses Dokuments erkennen Sie die Nutzungsbedingungen an.

Terms of use:

This document is made available under Deposit Licence (No Redistribution - no modifications). We grant a non-exclusive, non-transferable, individual and limited right to using this document. This document is solely intended for your personal, non-commercial use. All of the copies of this documents must retain all copyright information and other information regarding legal protection. You are not allowed to alter this document in any way, to copy it for public or commercial purposes, to exhibit the document in public, to perform, distribute or otherwise use the document in public.

By using this particular document, you accept the above-stated conditions of use.

Können AI-Programme als Theorien betrachtet werden?

Ein wissenschaftsphilosophischer Beitrag

Klaus Manhart

Institut für Höhere Studien, Wien

Zusammenfassung

Eine Vielzahl von AI-Forschern betrachtet erfolgreiche AI-Programme als Theorien über die modellierten Phänomene. Der vorliegende Beitrag untersucht, inwieweit diese Auffassung gerechtfertigt ist. Während bei der Umsetzung einer vorliegenden Theorie in ein Computerprogramm dem Programm i.a. ein Theoriestatus nicht abzuleugnen ist, können nicht alle erfolgreichen AI-Programme auch als Theorien betrachtet werden. Da Theorien erklärungskräftig sind und Erklärungen nur mit Hilfe von Gesetzen möglich sind, müssen "theoretische" Programme gesetzesartige Verallgemeinerungen enthalten. Sich daraus ergebende Probleme werden am Schluß dieses Aufsatzes diskutiert.

1. Einleitung

Während viele theoretische AI-Forscher die Ansicht vertreten, daß AI-Programme lediglich künstliche Realisationen empirischer Systeme sind, findet man in der AI-Literatur immer wieder Bemerkungen, wonach funktionierende Programme als Theorien über das Input-Output-Verhalten des unterliegenden empirischen Systems betrachtet werden. Funktionierende Programme werden danach nicht aufgefaßt als bloße Modelle, sondern als Theorien über das betrachtete System. In einem der weitverbreitetsten und einflußreichsten AI-Lehrbücher findet man beispielsweise die folgende Bemerkung (Winston 1984: 12):

"Occasionally after seeing what a program can do someone will ask for the theory behind it. Often the correct response is that the program is the theory" (Hervorhebung von Winston).

Ähnliche Bemerkungen finden sich gehäuft in Boden (1977), z.B.:

"A functioning program is a theory that is intended more as a movie of mind than a portrait of it..."(Boden 1977: 34).

An anderer Stelle heißt es über ein Programm, das neurotisches Verhalten simuliert:

"... the program represents a psychological theory of neurosis" (Boden 1977: 6).

Die AI-Pioniere Newell und Simon schreiben zu einem Programm, das menschliches Problemlösungsverhalten nachbilden soll: "Die Theorie löst die Aufgaben, die sie erklärt." (nach Weizenbaum 1978: 197). Nach dieser Auffassung ist offensichtlich das Programm die Theorie, welche empirische Phänomene erklärt, und gleichzeitig operiert das Programm auf diesen Phänomenen!

Eine Theorie zu verstehen, hieße dann einfach, das Programmlisting zu lesen:

"Because the algorithm for this model embodies a theory, the most direct way of understanding the theory is to read the listing of the algorithm" (Colby 1973: 266).

Die Auffassung, wonach funktionierende Programme als Theorien betrachtet werden können, ist im übrigen nicht beschränkt auf die AI, sondern findet sich auch außerhalb davon (vgl. die Bemerkung in Lehman 1977: 6-7).

Solche Thesen werden andererseits in Frage gestellt durch Gegenbeispiele wie Weizenbaums (1978) mittelalterlichen Uhrwerkbauer, der niemals die Newton'schen Gesetze entdecken würde, wenn er lediglich immer komplexere Uhrwerkmechanismen bauen würde.

Verschiedene Autoren (z.B. Lischka/Diederich 1987) beklagen deshalb, daß trotz 30-jähriger Tradition der Computermodellierung der methodische und wissenschaftstheoretische Status solcher Programme immer noch unzureichend definiert ist.

Bevor versucht werden soll, Licht ins Dunkel zu bringen, sei der Unterschied zwischen Theorien und Modellen herausgearbeitet.

2. Theorie und Modell

Unter Wissenschaftsphilosophen gibt es keine allgemein akzeptierte Antwort auf die Frage, was genau eine Theorie ist. Ohne in spezifische Details zu gehen, besteht bei den meisten Wissenschaftstheoretikern aber ein Konsens darüber, daß eine Theorie als eine Menge allgemeiner **gesetzesartiger** Aussagen betrachtet werden kann, die von Daten und Beobachtungen abstrahiert. Theorien haben unter anderem vier wichtige Eigenschaften (Lehman 1977: 3):

- sie sind eng verknüpft mit beobachtbaren Phänomenen;
- sie versuchen, eine Erklärung des beobachteten Verhaltens zu geben (zum Begriff der Erklärung vgl. unten);
- sie liefern testbare Voraussagen;
- sie beinhalten Aussagen über nicht-beobachtbare Prozesse (hypothetische Konstrukte).

Theorien sollten insbesondere 4 wichtigen Kriterien genügen (Lehman 1977: 227):

- Theorien sollten möglichst allgemein sein, d.h. erklärt eine Theorie T1 mehr Phänomene als eine andere Theorie T2, so sollte T1 vor T2 vorgezogen werden;
- Theorien sollte falsifizierbar sein, d.h. es müssen testbare Vorhersagen generierbar sein, die eine Theorie widerlegen können;
- Eine Theorie sollte präzise sein; sie sollte eine präzise Sprache verwenden und die Vorhersagen sollten genau sein;
- Eine Theorie sollte möglichst einfach sein, d.h. mit einem Minimum von Annahmen auskommen und möglichst wenig hypothetische Konstrukte postulieren.

Detaillierte formale Analysen über den Struktur von Theorien finden sich in Stegmüller (1969b).

Modelle abstrahieren von wesentlichen Elementen empirischer Phänomene und Theorien. Unter einem Modell wollen wir einerseits vereinfachte Formen empirischer Systeme verstehen, andererseits konkretere Formen abstrakter Theorien (Apter 1970: 22). Modelle werden manchmal aus Theorien abgeleitet und manchmal aus empirischen Phänomenen. Obwohl in der Literatur nicht üblich, erscheint es in gewissen Zusammenhängen wichtig, diesen Unterschied deutlich zu machen: entsprechend dem Abstraktionsgrad sprechen wir deshalb von empirischen oder theoretischen Modellen.

Bezüglich (theoretischem) Modell und Theorie bestehen folgende wichtige Unterschiede (Lehmann 1977: 4):

- ein Modell ist i.a. eine vereinfachte Repräsentation einer Theorie, aber nicht selbst die Theorie;
- Modelle sind konkreter und "greifbarer": anders als bei Theorien können Objekte und Variable manipuliert und die Effekte beobachtet werden;

- ein Modell ist eine Anwendung einer Theorie auf eine spezifische Situation. Während Theorien umfassend anwendbar sind, sind Modelle beschränkt auf spezielle Ausschnitte;
- Modelle sind nicht erklärend, sondern nur eine Vereinfachung und Abstraktion bestimmter Schlüsselemente einer Theorie. Ein Modell erlaubt die Konsequenzen einer Theorie zu erforschen, aber nicht, empirische Phänomene zu erklären.

Ein AI-Modell ist ein Modell in Form eines Computerprogramms, das Eingaben einliest und Ausgaben erzeugt. Ein solches Modell realisiert mathematisch gesprochen eine Funktion f von der Menge der Inputdaten I in die Menge der Outputdaten O :

$f: I \rightarrow O$

Das Programm P bildet hierbei den "Geist" G - oder andere empirische Phänomene - ab:

$i_m \in I_m \quad \text{--->} \quad \boxed{P} \quad \text{--->} \quad o_m \in O_m$

$i_a \in I_a \quad \text{--->} \quad \boxed{G} \quad \text{--->} \quad o_a \in O_a$

Abb.1: AI-Modellschema

i und o sind dabei irgendwelche, nach mathematischen oder linguistischen Regeln konstruierte Zeichenketten. In der AI werden in der Regel keine mathematischen Formeln als i und o verwendet, sondern linguistische Zeichenketten.

Was ist die Beziehung zwischen diesem empirischen System G und dem potentiellen Modell P ? Ist P ein Modell von G und beobachtet man eine von G produzierte spezifische i - o -Relation, so sollte gelten (' \approx ' steht hierbei für die Ähnlichkeitsrelation):

Wenn $i_a = i_m$ und $f_p(i_a) = o_a$ dann $o_a \approx o_m$

Die Ähnlichkeitsrelation ' \approx ' kann hierbei eventuell mit einem modifizierten Turing-Test erfaßt werden kann (vgl. hierzu Abelson 1968). P und G werden danach als i - o -äquivalent betrachtet, wenn die symbolischen i - o -Paare von P unter bestimmten Bedingungen nicht unterscheidbar sind von den symbolischen i - o -Paaren von G unter ähnlichen Bedingungen.

3. Top-down- und Bottom-up-Verfahren

Bei der Problemstellung, inwieweit ein Programm eine Theorie verkörpern kann, sind grundsätzlich 2 Ansätze zu berücksichtigen.

Im Top-down-Ansatz geht man von einer vorliegenden Theorie aus und versucht, diese in ein lauffähiges Computerprogramm zu bringen. Das Bottom-up-Verfahren wurzelt mehr in empirischen Phänomenen: man schreibt experimentelle Programme, verfeinert diese und betrachtet das Ergebnis u.U. - so einige Verfechter dieses Ansatzes - als Theorie.

Im Top-Down-Verfahren spricht im Prinzip nichts dagegen, eine vorliegende Theorie - sofern sie nicht allzu vage formuliert ist - in einer Programmiersprache zu "formalisieren". So wie Wissenschaftstheoretiker oder wissenschaftstheoretisch ausgebildete Sozialwissenschaftler versuchen, Theorien in einer formalen Sprache zu rekonstruieren (Mengentheorie, Logik, Graphentheorie, Differentialgleichungen etc.) kann eine Theorie auch in eine Programmiersprache abgebildet werden. Eine Programmiersprache ist nichts weiter als eine formale Sprache, die allerdings nicht zwangsläufig auch eine angemessene Theoriensprache ist. Von einer Sprache, in der eine Theorie formuliert ist, erwartet man, daß sie genau die für eine Theorie relevanten Aussagen beinhaltet, eine Programmiersprache muß aber erhebliche, theoretisch völlig irrelevante Teile enthalten. Glücklicherweise liegen in neuerer Zeit Programmiersprachen vor, die solche unwesentlichen Teile minimieren (siehe weiter unten).

Den oben angeführten Zitaten unterliegt aber das Bottom-up-Verfahren und damit die wesentlich schärfere Aussage, nach der alle - oder zumindest viele - erfolgreiche AI-Programme Theorien verkörpern würden.

Logisch betrachtet lassen sich die Ansichten über den Status von Programmen, mit abnehmender Stärke, in 3 Klassen von Aussagen unterscheiden:

- (A1) Alle i-o-äquivalenten (AI-)Programme sind auch Theorien über das entsprechende empirische System.
- (A2) Es gibt, unter bestimmten Voraussetzungen, (AI-)Programme, die als Theorien betrachtet werden können.
- (A3) Kein (AI-) Programm ist eine Theorie.

(A3) kann aufgrund der obigen Argumentation als widerlegt betrachtet werden. Offensichtlich entspricht (A2) der Auffassung von Winston (angezeigt durch die Partikel 'often'), aber es gibt auch Interpretationen, die (A1) zuzuordnen sind.

Mir ist z.Z. keine größere Arbeit bekannt, die den Status von AI-Programmen explizit untersucht; allenfalls findet man verstreut Bemerkungen hierzu in der AI-Literatur. Eine Ausnahme sind 2 kleine Papers von Kobsa (1982 und 1984a), an die ich anknüpfen will.

4. Kobsa's Argumentation

Kobsa (1984a) repliziert zunächst die Auffassung (A1), also derjenigen AI-Forscher, die alle funktionierenden Programme als Theorien betrachten:

AI-Forscher haben i.a. keinen formalen Theorienbegriff. Eine Theorie wird als eine Menge von Generalisierungen betrachtet und eine Erklärung ist eine Subsumption individueller Fälle unter die Theorie. Da jedes von einem AI-Programm erzeugbare i-o-Paar subsummierbar ist unter das Programm, stellt das Programm eine Verallgemeinerung des i-o-Paars dar. Das Programm - so die Schlußfolgerung der "Theorie = Programm-Auffassung" - ist deshalb eine Theorie über diese i-o-Paare und die Deduktion eine Erklärung.

Tatsächlich, so Kobsa, sind AI-Programme nicht Theorien über Strukturen, Mechanismen und Prozesse, sondern AI-Programme sind diese Strukturen, Mechanismen, Prozesse. AI-Programme bilden die Strukturen des realen Systems ab, sind also, in unserer Terminologie, empirische Modelle.

Was ermöglicht es einem Programm - nach Kobsa - ein bestimmtes Verhalten zu zeigen?

Bei der Erklärung der Fähigkeiten von Programmen beziehen sich Computerwissenschaftler gewöhnlich auf die Funktionen von Teilen des Modells. Das AI-Modell wäre dann eine funktionale Dekomposition: es kann in verschiedene Teile mit spezifischen Funktionen aufgeteilt werden, deren Zusammenwirken das Resultat erzeugt. Teile können durch funktional äquivalente Teile ausgetauscht werden; es kann somit verschiedene, funktional äquivalente Rekonstruktionsmöglichkeiten (u.U.unendlich viele) geben, die das gleiche i-o-Paar liefern.

Kobsa's Schlußfolgerung: Ein AI-Modell ist eine Instanziierung irgendeiner solchen funktionalen Dekomposition; es ist, wie der Name sagt, ein "Modell einer Theorie" und eine Rekonstruktion dessen, wie aus einem i ein o erzeugt wird; dies ist aber etwas ganz anderes als eine Theorie.

Es soll nun versucht werden zu zeigen, daß die von Kobsa eingangs dargestellte Argumentation der Theorie=Programm-Auffassung im Ansatz richtig, aber in der Konsequenz falsch ist. Ich werde dabei auf das formale Erklärungsschema in der analytischen Wissenschaftstheorie rekurrieren und Parallelen zwischen diesem Schema und dem obigen AI-Modellschema aufzeigen, die, unter gewissen Bedingungen, die Auffassung (A2) unterstützen. Auf Kobsa's Argumentation soll am Schluß dieses Abschnitts eingegangen werden.

5. Das Erklärungsschema der analytischen Wissenschaftstheorie

Wenn ein Programm eine Theorie ist, so ist es zwangsläufig auch erklärungsfähig: es muß erklären können, warum ein Ereignis - der Output - gültig ist. Nach dem weitgehend akzeptierten Hempel-Oppenheim-(H-O)-Schema gilt ein Ereignis dann als erklärt, wenn das zu erklärende Phänomen logisch aus Gesetzen und Anfangsbedingungen deduziert werden kann (Stegmüller 1969a). Gesetze und Anfangsbedingungen sind das Explanans, das zu erklärende Phänomen das Explanandum. Das Schema kann formal wie folgt repräsentiert werden:

A_1, A_2, \dots, A_n	Anfangsbedingungen	}	Explanans
G_1, G_2, \dots, G_n	Gesetze		
E	zu erklärendes Ereignis	}	Explanandum

Bedingungen für adäquate und gute Erklärungen sind:

- (B1) Das Argument, welches vom Explanans zum Explanandum führt, muß (logisch) korrekt sein.
 (B2) Das Explanans muß mindestens ein allgemeines Gesetz enthalten.
 (B3) Das Explanans muß einen empirischen Gehalt haben.
 (B5) Die Sätze, aus denen das Explanans besteht, müssen wahr sein.
 Für eine ausführliche Analyse dieser Bedingungen vgl. Stegmüller (1969a: 86-90).

Der Erklärung, warum ein Stück Kupferdraht - nennen wir es a - Elektrizität leitet, entspräche das folgende H-O-Schema (wir benutzen einfache prädikatenlogische Formeln):

$(x) K(x) \rightarrow E(x)$	Kupfer leitet Elektrizität
$K(a)$	a besteht aus Kupfer
<hr/>	
$E(a)$	a leitet Elektrizität

Die Frage, warum a Elektrizität leitet, wird dadurch erklärt, daß a aus Kupfer besteht und alles Kupfer Elektrizität leitet. Natürlich kann das Gesetz, wonach Kupfer Elektrizität leitet, wiederum durch ein allgemeineres Gesetz erklärt werden.

Während bei dem Versuch, ein Phänomen E zu erklären, E und A_i gegeben und Gesetze gesucht sind, kann das gleiche Schema zur Prognose benützt werden, wenn A_i und G_i gegeben sind. Aus A_i und G_i kann E als Prognose deduziert und mit empirischen Beobachtungen verglichen werden.

6. Gesetzmäßigkeiten als Bedingung theoretischer Programme

Zwischen dem H-O-Schema und dem obigen AI-Modellschema besteht eine auffallende strukturelle Ähnlichkeit. Dies wird in folgender Darstellung deutlicher:

A	i_m
G	P
—	—
E	o_m

Danach entspricht dem Antezedens der Modellinput, den Gesetzen das Programm und dem Explanans der Modelloutput: die Analogie legt die Auffassung nahe, daß auch ein Programm zur Erklärung benutzt werden kann. Entscheidend ist aber offensichtlich, daß das Programm als minimale Voraussetzung gesetzesartige Verallgemeinerungen enthält ohne die keine Erklärungsleistung und kein theoretischer Status gegeben sind.

Während aber dem Übergang von $\{A,G\}$ zu $\{E\}$ ein logischer Schluß unterliegt, unterliegt dem Übergang von $\{i_m,P\}$ zu $\{o_m\}$ kein logischer Schluß, sondern ergibt sich aus dem Algorithmus des Programms. Dieser algorithmische Übergang sollte ohne aufwendige Änderungen in eine logische Deduktion umgesetzt werden können.

Eine bloße Verhaltensähnlichkeit eines Modells mit seinem Objekt genügt nicht zum Theoriestatus - wie die von Kobsa replizierte Argumentation der Theorie=Programm-Auffassung nahelegt. Selbst ein Strukturmodell, das die Relationen innerhalb eines empirischen Systems widerspiegelt, hat nicht ohne weiteres theoretischen Charakter. Seine Relationen geben vielleicht die realen Variablenbeziehungen mit ausreichender Ähnlichkeit wieder, aber die Wiedergabe empirischer Beziehungen ist nicht gleichbedeutend mit der Formulierung allgemeiner Gesetzaussagen. Die empirische Gültigkeit eines Modells schließt nicht dessen Eignung zur Erklärung ein, wohl aber umgekehrt (Harbordt 1974: 169).

In der oben gegebenen Argumentation der Theorie=Programm-Anhänger wird - bildlich gesprochen - von den Wirkungen auf die Ursachen geschlossen: Rekonstruiert man die von Kobsa angegebene Argumentation, so ergibt sich folgende Kette:

- Singularitäten von i-o-Paaren sind subsumierbar unter ein Programm
- also ist das Programm eine Verallgemeinerung
- also ist das Programm eine Theorie.

Ein Programm kann aber keine Theorie sein, weil i-o-Paare subsumierbar sind unter dieses Programm, sondern es ist eine Theorie, falls es Verallgemeinerungen enthält und aufgrund dieser Verallgemeinerungen i-o-Paare subsumierbar sind unter das Programm.

Colby ist der einzige mir bekannte AI-Forscher, der explizit eine analoge Position vertritt:

"This model (für einen Paranoiker, Anm. Verf.) is considered a theoretical model because its inner structure embodies an explanatory account of the complex phenomena of paranoid communicative behavior. An explanatory account contains statements of lawlike generalisations" (Colby 1973: 265, Hervorh. Verf.).

Wenn Programme als Theorien betrachtet werden können, dann schaffen Computer eine völlig neue Beziehung zwischen Theorie und Modell. Eine Theorie in Form eines Computerprogramms ist sowohl eine Theorie als auch - in einen Computer eingegeben und von diesem bearbeitet - ein Modell, auf das die Theorie Anwendung findet. (Weizenbaum 1987: 197). In diesem Sinn ist das obige Zitat von Newell und Simon zu verstehen "Eine Theorie löst die Aufgaben, die sie erklärt." Eine Theorie - als eine Menge von Aussagen - kann nichts lösen, aber ein Modell. Theorie und Modell können in einem Computerprogramm "vereint" werden.

Das Verhältnis Theorie-Programm-Modell bringt Weizenbaum gut zum Ausdruck:

"Feigenbaums Programm ist, wie ich schon gesagt hatte, das Modell einer psychologischen Theorie, nämlich darüber, wie Menschen mit der Aufgabe fertig werden, Nonsense-Silben zu lernen. Ich hatte darauf hingewiesen, daß das Programm selbst ebenfalls eine Theorie ist; wenn man es beispielsweise einem Psychologen gibt, der mit der Programmiersprache vertraut ist, in der sie geschrieben ist, so darf man erwarten, daß er sie verstehen wird. Was es jedoch zur Theorie macht, ist der Umstand, daß es bestimmte Prinzipien aufstellt, aus denen Konsequenzen gezogen werden können. Diese Prinzipien liegen in Form eines Computerprogramms vor, und ihre Konsequenzen lassen sich am Verhalten des Programms ablesen, d.h. an der Art und Weise, wie der Computer das Programm liest" (Weizenbaum 1978: 235-236).

Es dürfte nicht überinterpretiert sein, Weizenbaum's "Prinzipien" als Gesetze zu identifizieren, deren Vorliegen einem Programm einen Theoriestatus verleihen können.

7. Einige problematische Aspekte

Diese Ergebnisse werfen eine Reihe von Fragen auf, die kurz angeschnitten werden sollen:

(P1) Was genau ist ein Gesetz?

(P2) Gibt es psychologische Gesetze?

(P3) Widerspricht die Funktionssichtweise dem Gesetzeskriterium?

(P4) Wie läßt sich genau verifizieren, wann ein Programm theoretisch ist und wann ein empirisches Modell?

(P1) Ein Unterscheidungskriterium zwischen Gesetzen und Nicht-Gesetzen zu finden ist selbst im naturwissenschaftlichen Fall wissenschaftstheoretisch nicht befriedigend gelöst. Die Postulierung von Gesetzen als allgemeine, raum-zeitlich unbeschränkt gültige Allaussagen wirft eine Fülle von Problemen auf, die hier nicht angeschnitten werden können (Stegmüller 1969a). Weitere Probleme kommen hinzu, wenn die Gesetze nicht streng deterministisch, sondern statistischer Art sind, also etwa: alle x mit der Eigenschaft F haben die Eigenschaft G zu 90%. (Stegmüller widmet dem Problem statistischer Erklärungen einen eigenen Band). Abgesehen von wissenschaftsphilosophischen Spitzfindigkeiten ist aber intuitiv klar, was unter einem Gesetz zu verstehen ist; unter Naturwissenschaftlern herrscht ein einstimmiger Konsens über Aussagen, die sie als Gesetze akzeptieren.

(P2) In Psychologie und Sozialwissenschaften ist immer wieder die Existenz von Gesetzmäßigkeiten im Sinn der Naturwissenschaften bestritten worden, insbesondere von hermeneutisch-geisteswissenschaftlichen Schulen. Aber selbst diese verwenden in ihren "theoretischen" Aussagen, wenn auch nur implizit, Verallgemeinerungen. Jede Theorie beruht eo ipso auf Verallgemeinerungen, ansonsten wäre es keine Theorie. Allerdings sind sozialwissenschaftliche Gesetzmäßigkeiten nicht von der Exaktheit und Allgemeinheit naturwissenschaftlicher Gesetze; soziale Regularitäten müssen oft statistisch formuliert werden, sind von beschränkter raum-zeitlicher Allgemeinheit (middle-range-theories) und sind insbesondere oft ungenau, rudimentär, partiell und skizzenhaft (Stegmüller 1969a, Beispiele finden sich S.105-116). Soweit möglich, sollten theoretische Aussagen präzisiert und vervollständigt werden, in vielen Fällen ist dies aber nicht möglich, da sozialen Phänomenen per se eine gewisse Unschärfe und Vagheit eigen ist. Glücklicherweise sind aber in der AI Methoden entwickelt worden, die es erlauben, auch vages und ungenaues Wissen darzustellen.

(P3) Nach der Auffassung von Kobsa (1984a) ist ein Programm eine funktionale Dekomposition in Teile, deren Zusammenwirken den Output bestimmt:

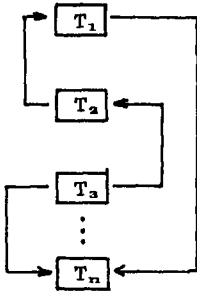


Abb.2: Funktionale Dekomposition von Programmen

Es gibt Teile, die Ein- Ausgabeoperationen durchführen, Teile, die Listenelemente einordnen, umstellen etc. Die Teile wirken zusammen, indem sie sich gegenseitig aufrufen etc. und all dies Zusammenwirken bestimmt schließlich einen Output. Schließt diese Funktionssichtweise aus, daß ein Programm eine Theorie ist? Nein, wenn sich Teile als gesetzesartige Aussagen identifizieren lassen, die in einer relevanten Beziehung zum empirischen System stehen (dies müßte weiter präzisiert werden) und jeder, der der Programmiersprache mächtig ist, sofort die Gesetze der Theorie angeben kann.

Die Betrachtung eines Programms als einer Theorie kann allerdings durch die verwendete Sprache gefördert oder behindert werden.

Traditionelle Programmiersprachen sind wenig adäquat zur Darstellung von Theorien, da sie viel prozeduralen "Ballast" enthalten, der die Einsicht in den theoretischen Kern verschleiert oder zumindest behindert. Wenn im Detail spezifiziert werden muß, wie ein Programm ein Problem bearbeitet, so wird es schwer fallen, die theoretischen Aussagen zu erkennen, da diese u.U. wiederum selbst als Zusammenwirken verschiedener Teile entstehen. Dies gilt m.E. auch für LISP, da sie relativ spezifizieraufwendig und maschinennah ist. Bei der derzeit höchsten Programmiersprache PROLOG läßt sich aber die prozedurale Wie-Sichtweise weitgehend durch eine Was-Sichtweise ersetzen, in der direkt Verallgemeinerungen angegeben werden können ohne auf spezifische Algorithmen, die diese Verallgemeinerungen erzeugen, zu achten. In PROLOG-Programmen fällt es auch leicht, die Funktionssichtweise ganz in den Hintergrund zu drängen.

(P4) Eine einfache Dichotomie zwischen theoretischen und nicht-theoretischen Programmen wird sich nicht ziehen lassen, sondern es wird ein Kontinuum geben mit rein theoretischen Programmen und rein empirischen Modellen als Extrempunkte. Einige Markierungspunkte lassen sich in etwa wie folgt benennen:

 reine
 Theorie-
 programme

theoret.
 Modelle
 (Simulations-
 charakter)

empir.
 Modelle

Empirische Modelle als künstliche Realisationen empirischer Systeme enthalten keine allgemeineren Gesetzmäßigkeiten oder deuten solche allenfalls an, da der Weg immer vom empirischen System zum Modell geht und eine möglichst gute i-o-Äquivalenz - und ev. auch Strukturäquivalenz - angestrebt wird. Ergebnisse dieser Simulationen können aber Theorien vielfach initiieren. Die meisten AI-Modelle sind in diesem Bereich angesiedelt.

Theoretische Modelle mit einer Zahl von Gesetzmäßigkeiten können vielleicht als eine unvollständige oder vereinfachte Theorie betrachtet werden, wobei aber der Simulations- und Experimentiercharakter im Vordergrund steht.

Mit reinen Theorieprogrammen meine ich Programme ohne ausgeprägten Simulations- und Experimentiercharakter, die aber eine Theorie vollständig und korrekt wiedergeben. Solche Computer-Theorien werden üblicherweise aus bestehenden Theorien konstruiert bzw. bestehende Theorien können anstatt in einer üblichen formalen Sprache in einer Programmiersprache rekonstruiert werden. Der Vorteil dieses Verfahrens liegt hauptsächlich in der Präzisierung und Veranschaulichung, d.h. man sieht, wie eine Theorie "arbeitet".

LITERATUR

- Abelson, R.P. (1968), *Simulation of Social Behavior*, in: Lindzey, G./ Aronson, E. (ed.) *The Handbook of Social Psychology*, Vol.2, 2.Aufl., Reading, Mass: Addison-Wesley
- Apter, M.J. (1970), *The Computer Simulation of Behaviour*, New York: Harper
- Boden, M. (1977), *Artificial Intelligence and Natural Man*, Brighton: Harvester
- Colby, K.M. (1973), *Simulations of Belief Systems*, in: Schank, R.C./Colby, K.M. (ed.), *Computer Models of Thought and Language*, San Francisco: Freeman
- Harbordt, S. (1974), *Computersimulation in den Sozialwissenschaften*, Bd.1, Hamburg: Rowohlt
- Kobsa, A. (1982), *On Regarding AI Programs as Theories*, in: Trapp R. (ed.), *Cybernetics and System Research*, Amsterdam: North-Holland
- Kobsa, A. (1984a), *What is Explained by AI Models?*, in: *Communication and Cognition*, 2-3, S.49-65
- Kobsa, A. (1984b), *Künstliche Intelligenz und Kognitive Psychologie*, in: Retzl, J. (ed.), *Artificial Intelligence - Eine Einführung*, Stuttgart: Teubner
- Lehman, R.S. (1977), *Computer Simulation and Modeling: An Introduction*, New York: Wiley
- Lischka, C./Diederich, J. (1987), *Gegenstand und Methode der Kognitionswissenschaft*, in: *GMD-Spiegel*, 2-3, S.21-32
- Stegmüller, W. (1969a), *Probleme und Resultate der Wissenschaftstheorie und Analytischen Philosophie*. Bd.I. *Wissenschaftliche Erklärung und Begründung*, Berlin: Springer
- Stegmüller, W. (1969b), *Probleme und Resultate der Wissenschaftstheorie und Analytischen Philosophie*. Bd.II. *Theorie und Erfahrung*, Berlin: Springer
- Weizenbaum, J. (1978), *Die Macht der Computer und die Ohnmacht der Vernunft*, Frankfurt a.M.: Suhrkamp
- Winston, P. (1984), *Artificial Intelligence*, 2.Aufl., Reading, Mass: Addison-Wesley