

Walter
Hehl



Wechsel- wirkung

Wie Prinzipien der Software
die Philosophie verändern



Springer Vieweg

Wechselwirkung

Walter Hehl

Wechselwirkung

Wie Prinzipien der Software die
Philosophie verändern

Walter Hehl
Thalwil
Schweiz

ISBN 978-3-662-48113-4 ISBN 978-3-662-48114-1 (eBook)
DOI 10.1007/978-3-662-48114-1

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Springer Vieweg

© Springer-Verlag Berlin Heidelberg 2016

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften. Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen.

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

Springer-Verlag Berlin Heidelberg ist Teil der Fachverlagsgruppe Springer Science+Business Media
(www.springer.com)

Vorwort

Die fruchtbarsten Entwicklungen haben sich überall dort ergeben, wo zwei unterschiedliche Arten des Denkens zusammentrafen.

Werner Heisenberg, Physiker, 1901–1976

Software ist ein Technikbereich, der wenig sichtbar, aber wirtschaftlich immens wichtig und verbreitet ist: Alles, was komplex ist und agiert, ist Software. Nach vielen Berufsjahren u. a. als Manager für „Advanced Software“, für „Advanced Systems“ und auch als Professor auf Zeit am Lehrstuhl für Softwaretechnologie an der TU Dresden wurde mir diese Bedeutung klar und ich bedauerte es, dass Konzepte aus der Software nicht allgemeiner bekannt sind.

Das Eingangszitat wird oft in esoterischen Kreisen verwendet. Es geht allerdings weiter mit *„wenn eine echte Wechselwirkung stattfindet“*: Genau dies ist bei Software und Philosophie der Fall. Es gibt viele Ideen, die es wert wären, auch ausserhalb der Softwaregemeinde weiter gegeben zu werden. Eine Reihe der auftauchenden Fragen, Begriffe und Antworten sind philosophischer oder nahezu philosophischer Natur, die in Software konstruktiv, pragmatisch und funktionierend angegangen werden. Sollte es nicht fruchtbar sein für philosophisch Interessierte, einen Blick über den Zaun zu werfen? Die Software scheint mir ein nüchterner, technischer Nachbar zur Philosophie zu sein – schliesslich befassen sich beide Gebiete mit dem Bau von Systemen und mit (auf eine gewisse Art) geistigen Objekten. Es würde mich freuen, wenn das Buch zur Akzeptanz der Software als einem wesentlichen Träger der Welt neben der materiellen Säule beitragen könnte und wenn die dargestellten Ideen in ein paar Jahren als selbstverständlich angesehen würden.

Auf der anderen Seite sollten vielleicht auch Informatiker die Brücke zu philosophischen Gedanken überschreiten und die Sicht darauf von einem Ingenieur und Physiker hören wollen: Also schrieb ich dieses Buch. Eine kleine Freude hat mir die Suche nach mehr oder weniger passenden Zitaten zur Auflockerung gemacht: Ich hoffe, sie machen auch der Leserin oder dem Leser intellektuelles Vergnügen.

Zum Begriff „Software“: Es geht natürlich auch um den Computer, aber erstens ist „Computer“ nur das Gerät (wer würde z. B. ein Buch über „Philosophie und Astronomie“ als „Philosophie und Teleskop“-Buch bezeichnen), und zweitens ist Software wirklich das Wichtigere. Software beschreibt das Werden, nicht nur „das Sein“: Es ist der Stoff, der macht, im Computer, ja im erweiterten Sinn sogar vieles im Menschen.

Einleitung

Hören Sie auf die Technologie, und verstehen Sie, was sie sagt.
Carver Mead, Computeringenieur, geb. 1934

Philosophie und Software sind vielleicht ein überraschendes Paar: Philosophie versucht mit Weisheit die Welt und den Menschen zu verstehen, Software baut all das in der Welt, was handelt. Aber im erweiterten Sinn läuft Software auch im biologischen Leben und in unserem Intellekt: Philosophie verhält sich zur Software ähnlich wie Architektur zur Bau- statik und Bauphysik: Es gibt Gemeinsames und gemeinsame Aufgaben.

Zunächst geben wir einen Abriss der Philosophie aus Ingenieurssicht, der hoffentlich bei den Fachphilosophen keinen Unmut auslöst, aber in diese 2500 Jahre alte Disziplin kurz einführen soll. Software ist der neue Stoff der modernen Wirtschaft und Gesellschaft (und erweitert auch der belebten Natur), führt aber ein Leben am Rande des Intellektuellen. Sie muss in ihrer philosophischen Bedeutung erst akzeptiert werden (ihre praktische Bedeutung ist nicht zu übersehen). Aber sie hat viele Facetten und grundsätzliche Aufgaben, die sie (mit Information zusammen) sogar zu einer Säule der Welt machen, so wichtig wie die materielle Säule.

Software und ihre Konzepte können auf zwei Ebenen philosophische Bedeutung haben, eine kleine Bedeutung als Analogon und Lehrmittel, eine grosse Bedeutung unmittelbar für die Weltvorstellung und Welterklärung als die zweite Säule neben der physischen „materiellen“ Welt als erster Säule.

Als kleine Bedeutung sehen wir uns philosophische Ideen an, die sich in der normalen Softwaretechnik finden, etwa den Begriff der Komplexität, oder in geänderter Form wiederfinden, konkreter, formeller aber weniger mit dem Anspruch auf absolute Erkenntnis, etwa die Ontologie oder die Denkmuster. Wie baut man grosse Systeme? Danach wird es eventuell origineller und kontroverser im Kapitel über Emergenz und Evolution: Wie entsteht Neues mit und um Software? Was ist Emergenz? Verschiedene Stufen der Kreativität werden erläutert – von sich selbständernden Programmen und neuronalen Netzen über den Affen, der Shakespeares Hamlet erzeugen soll, zum mittelalterlichen katalanischen Philosophen Ramon Llull und seinem Wissensrad. Dazu gehört der Zufall – künstlich, kryptisch oder ganz echt. Das Ergebnis ist ein zunächst dualistisches Weltmodell (plus Zu-

fall und plus Mathematik): die „It-Welt“ der Atome (und des übrigen Quasi-Materiellen) und die „Bit-Welt“ (die Software im allgemeinen Sinn).

Im Kapitel Bewusstsein wird strömendes Computing eingeführt als zugehörige technische Definition mit dem „Bewusstsein“ der selbstfahrenden Autos. Es ist im Zusammenhang mit Software offensichtlich, dass wir die Computermetapher des Gehirns zumindest als Bild verwenden. In diesem Sinne lösen sich philosophische Probleme wie das Bewusstsein und die „Qualia“ (das spezifische Empfinden z. B. von Farben) philosophisch auf und werden zu wissenschaftlich-technischen Fragestellungen. Dafür spannen wir das Bewusstsein mit Informationstechnologie (d. h. dem Internet) weltweit auf („transzendieren“). Es wird zur „Noosphäre“ und zum „Weltgehirn“.

Eine ähnliche Klärung geschieht mit dem Verstehen und der Intelligenz: Wir zeigen, wie Software „versteht“ und etwas „weiss“ und warum menschliche Hybris (dass nur wir es „richtig“ können) überhaupt nicht angebracht ist; wir diskutieren die Fähigkeiten und Grenzen von „Watson“, dem Quizsystem der IBM, und begründen, warum die Entwicklung an den schönen und beliebten menschlichen Imitationstests (z. B. dem „Turing-Test“) vorbeigeht. Wir erläutern, dass Maschinelle Intelligenz (und sogar die Seele) Sorites-Probleme sind (und was Sorites-Probleme sind), und dass man mit vielem rechnen kann, z. B. mit Dominosteinen und, respektlos gesagt, „mit Fleisch“, nach einer wunderbaren Novelle von Terry Bisson. Bei der Seele als Gesamtsoftware unserer Innenprozesse (philosophisch das Leib-Seele-Problem) betrachten wir die Schichten der IT, von Neurosimulation „unten“ bis zur psychologischen „Seele“ und ihrer Simulation „oben“.

Dazu kommen weitere kontroverse Themen zwischen Philosophie und Informationstechnologie (bzw. innerhalb der Philosophie) wie der sog. „freie Wille“ (mit der falschen Homunkulustheorie), unsere Identität (ein eigentliches Softwareproblem, das wir mit dem philosophischen Schiff-des-Theseus verbinden) und exotische philosophische Softwareideen wie das Gehirn-im-Tank und der Mind-Upload: Wie stellt man fest, dass die Welt nicht eine grosse Simulation ist? Unsere Antwort ist übrigens: man kann es nicht, solange der Weltprogrammierer oder der Weltcomputer keinen Fehler macht.

Als eine Vorahnung des Dualismus der beiden Welten Materie und Software in der Antike stellen wir eine begriffliche Verbindung her mit der Bewegungslehre des Aristoteles. Das grosse Weltbild mit Materie, Software und Geist entsteht im letzten Kapitel als Abwandlung der Drei-Welten-Theorie des britisch-österreichischen Philosophen Karl Popper. Dazu erweitern wir das dualistische System um eine zu Popper modifizierte 3. Säule, den Kern des wirklich Geistigen. Vieles, was im Alltag als geistig gilt, etwa Wissen und Sprache, ist ja im Wesentlichen zu Software und Konstruktion geworden (oder wird es gerade in unserer Zeit, zur unnötigen Kränkung vieler) und ist eigentlich nicht geistig sondern „nur“ intellektuell. Aber es gibt das Geistige, beinahe trotzig gesagt: Es muss es geben, insbesondere Liebe und Kunst jenseits von Physik und Chemie und emotionaler Software. Dazu kommen als Grundelemente die Mathematik und der unbedingt notwendige Zufall als Einbrechen einer Aktion von ausserhalb des Systems, allerdings nur im Rahmen der Naturgesetze. Wir versuchen zu zeigen, dass Software die konstruktivistische Seite der Welt und des Weltverstehens ist. Sie zeigt nicht nur, sondern baut und betreibt die

Welt. Andererseits ist sie die funktionalistische Seite (sie liefert die benötigten komplexen Funktionen, ob Bildverarbeitung oder Sehen oder Leben selbst).

Wir haben uns bemüht, auch Gegenargumente zu unserer Sicht von Software und Computer (zusammengefasst im neokartesischen Spruch „*Procedo Software, ergo sum*“) zu berücksichtigen und zu widerlegen. Zu den im Folgenden vorgestellten Ingenieurs- und Philosophiekonzepten sind Diskussionen erwünscht.

Dank

Es ist mir eine grosse Freude, meine Erfahrungen aus Physik, Software und Philosophie weitergeben zu dürfen. Ich bedanke mich vor allem bei Hermann Engesser, dem Leiter der Computerfachbücher beim Springer-Verlag Heidelberg, für die freundliche Aufnahme der sicher ungewöhnlichen Idee eines philosophischen Computerbuchs, für Hinweise und Diskussionen. Dazu kam die Ermutigung durch unzählige Gesprächspartner, die einerseits von Einzelthemen angetan waren (wie dem Gehirn-im-Tank) und andererseits von der Gesamtkonzeption der Verbindung von Philosophie und Software.

Für die Anregung, über Computer überhaupt selbst zu philosophieren, möchte ich mich vor allem beim ehemaligen Kollegen Albert Endres bedanken. An dieser Stelle sei auf den kontinuierlichen Fluss an Ideen zur Informatik in seinem Blog Bertals Blog hingewiesen. Als ersten, schon historischen Anstoss zu Philosophie und Software möchte ich die Stuttgarter Vorlesung im Studium Generale von Max Bense erwähnen mit dem Titel „Kybernetik und Ästhetik“. Ohne dass ich viel verstanden hätte, hat sie mich unvergänglich beeindruckt. Einige Ideen des Buches konnte ich mit Herrn Gert Folkers vom Collegium Helveticum angeregt diskutieren, dafür herzlichen Dank.

Gerade bei der Vielfalt der Themen und den Sprüngen zwischen verschiedenen Bereichen von antiker Philosophie bis zu moderner Computertechnologie – immer mit dem gemeinsamen Nenner „Software“ – war die „gute alte“ Wikipedia sehr hilfreich.

Ich danke vor allem meiner Frau Edith für ihre Geduld; zum einen bei ihrem sorgfältigen Korrekturlesen dieses Textes, zum andern mich über zwei Jahre hinweg sammeln und schreiben zu lassen.

Inhaltsverzeichnis

1 Was ist Philosophie?	1
1.1 Eine Frage, die „es“ schon ist	1
1.2 Prä- und Postphilosophie	3
1.3 Philosophie der Systeme: Architekturen, Ontologie des Seins und des Werdens, Epistemologie	8
1.4 Philosophische Kuriositäten: Paradoxien	17
2 Was ist Software?	23
2.1 Die Definitionen von Software und Information	23
2.2 Atome und Software als zwei Säulen: Ein Dualismus zur Konstruktion der Welt	32
2.3 Komplexität und Software	41
2.4 Arten und Eigenschaften von Software – nützliche Beobachtungen	46
3 Philosophisches aus und für Software und Softwaretechnologie	55
3.1 Software als Ingenieursdisziplin	55
3.2 Sein und Seiendes: Einige technologische Konzepte. Von Objekten, Klassen und Geheimnissen	57
3.3 Ideen für und aus interagierenden Systemen: Von Hollywood und fünf Philosophen	61
3.3.1 Rechnerkommunikation: Hollywood-Prinzip	63
3.3.2 Rechnerkommunikation: Körnigkeit von Arbeiten und Kohärenz von Systemen	63
3.3.3 Synchronizität: „Tödliche“ und „lebendige“ Umarmungen und die fünf essenden Philosophen	66
3.4 Wie baut man grosse Systeme: Gedanken zur und Ideen aus der Gesamtarchitektur	69
3.4.1 Warum Architektur?	69
3.4.2 Architekturprinzipien für Softwaresysteme	70
3.5 Die Idee der Denk- und Entwurfsmuster	76
3.6 Agiles zur Managementphilosophie und Software	79

4 Philosophisches mit Software: Emergenz, Evolution und Weltmodell (Ontologie)	85
4.1 Was ist Emergenz?	85
4.2 Emergenz durch vorgegebene Software	88
4.3 Der Zufall als Schöpfer: Philosophische und physikalische Arten des Zufalls	92
4.4 Kreativität durch Kombinatorik und „Brute Force“ – Software	100
4.5 Emergenz und Evolution durch emergente Software	105
4.5.1 Selbstmodifizierender Code	106
4.5.2 Neuronale Netze: Vom Falschen zum Richtigen, ohne Verstehen, manchmal ohne Wissen	107
4.5.3 Evolutionäre Verfahren: Emergenz durch Evolution in der angewandten Mathematik	110
4.5.4 „Die“ Evolution und Software	111
4.6 Konstruktiver Zufall	125
4.6.1 Das Wachstum von Schneekristallen als anorganisches Lehrstück	126
4.6.2 Weltmodelle mit Zufall	127
5 Strömende Software und Bewusstsein	129
5.1 Definition des Bewusstseins	129
5.2 Strömendes Computing und Software für Komplexe Events	131
5.3 Menschliches und künstliches Bewusstsein	136
5.4 Weltbewusstsein und Noosphäre	143
5.5 Die technologische Singularität	147
5.6 Physisches Weltbewusstsein und „Weltgehirn“	150
6 Wissen, Verstehen und Intelligenz mit Software	155
6.1 Wissen und Verstehen mit Softwarearchitektur	155
6.2 Software weiss Wissen: Google, Watson und weiter	162
6.3 Das Sorites-Problem und die Intelligenz der Computer	172
6.3.1 Denken und Computer	173
6.3.2 Verstehen und Computer	174
6.3.3 Sorites, menschliche und maschinelle Intelligenz	176
7 Es muss nicht Silizium (oder Ähnliches) sein	193
7.1 Konventionelle und unkonventionelle Computertechnologien	193
7.2 Gehirntechnologie	199
8 Software der Seele: Weil wir Computer sind	205
8.1 Die Seele: Von Aristoteles zum Softwaresystem	205
8.2 Hierarchische Mustererkennung als zentraler Prozess	211
8.3 Computer und Psyche: Simulation der Seele	217

9 Freier Wille, Gehirn-im-Tank, Identität	225
9.1 Der freie Wille und der Homunkulus	225
9.2 Identität, Präsentität und das Schiff des Theseus	232
9.2.1 Was ist unsere Identität?	232
9.2.2 Wir sind Schiffe des Theseus	236
9.3 Das Gehirn im Tank und „Mind Upload“	241
9.3.1 Die simulierte Aussenwelt und das Universum aus Software	241
9.3.2 „Sich hochladen“: Der Mind Upload	246
10 Und wo bleibt der Geist?	251
10.1 Das Problem mit dem Geist, vor allem in der deutschen Sprache	251
10.2 Die drei Welten – Karl Popper revisited	255
11 Wechselwirkungen	265
11.1 Softwareismus	265
11.2 Denkfallen und Irrtümer	270
11.3 Ohne Software kein Weltverständnis: Ein Drei-Welten-Modell	274
Anhang	279
Glossar	281
Literatur	289
Sachverzeichnis	291

Wir geben verschiedene (simple) Definitionen des Begriffs „Philosophie“ und diskutieren Philosophie in Bezug auf Naturvorstellungen und Stand der Technik als Zeitphänomen. Im Sinne der Definition als „nicht-triviale Gedanken jenseits von Wissenschaft“ definieren wir eine „Präphilosophie“ vor der wissenschaftlichen Erschließung eines Wissensbereichs und die „Postphilosophie“ danach. Dies wurde z. B. im Bereich der Naturphilosophie und der Atomlehre erlebt mit Vertretern wie Demokrit einerseits und Werner Heisenberg andererseits. Heute findet dieser Übergang bei den Informations- und Softwaresystemen statt und erfasst damit Begriffe wie Komplexität, Wissen, Bewusstsein, Identität, ja Leben und vieles mehr. Früher verbal und kontrovers diskutiert, heute oft formalisiert, quantifiziert und sogar nachgebaut. Wir weisen hier auf Philosophiebereiche hin, die von der Entwicklung von Software und Informatik besonders betroffen werden. Als kleine Wechselwirkung von Philosophie und Software führen wir in eine Reihe von klassischen philosophischen Paradoxien ein, die in der IT pragmatisch gelöst werden. Ein Beispiel ist das Barbier-Problem von Bertrand Russell.

1.1 Eine Frage, die „es“ schon ist

Philosophie ist ein liebevoller Umgang mit der Weisheit.
Dante Alighieri, Dichter, 1265–1321

Philosophie ist die Wissenschaft, über die man nicht reden kann, ohne sie selbst zu betreiben.
Friedrich von Weizsäcker, Astrophysiker, 1912–2007

Wissenschaft ist das, was man weiss; Philosophie ist das, was man nicht weiss.
Bertrand Russell, Philosoph, 1872–1970

Die grosse philosophische Frage:

Deep Thought: „Die Antwort nach dem Leben, dem Universum und dem ganzen Rest ist ...“

Die Philosophen: „Ja?...“

Deep Thought: „ist ...“ Die Philosophen (etwas lauter): „Ja?...“

Deep Thought: „ist ...“ Die Philosophen (richtig laut): „Ja?...“

Deep Thought: „42“ Philosoph 1: „Die bringen uns um, weisst Du das?“

Douglas Adams, Schriftsteller, 1952–2001

Mit der Frage nach der Definition von Philosophie sind wir schon mittendrin in der philosophischen Diskussion und damit in der Philosophie: Viele Berufs- oder akademische Philosophen und manche philosophische Autodidakten haben ihre eigene Definition – und dazu ihre eigenen Philosophien. Einig ist man sich in der Herkunft des Wortes und dessen Ursprung: Die Philosophie (altgriechisch φιλοσοφία philosophía) ist in wörtlicher Bedeutung die „Liebe zur Weisheit“.

Wir werden „Weisheit“ an Hand einer Softwaregraphik und mit einer kleinen Formulierungshilfe vom Wikipediaartikel „Weisheit“ (07/2015) definieren als „die Fähigkeit, bei Problemen und Herausforderungen eine nachhaltige und schlüssige Handlungsweise zu finden“ mit dem Zusatz „ohne die darunterliegenden Details zu verstehen“. Dabei ist die Betonung auf nachhaltig wesentlich, denn Handeln ohne zu verstehen ist zunächst sehr gebräuchlich, aber noch nicht weise, eher das Gegenteil. Der Weise erahnt das Wesentliche für ein Problem, einschliesslich der Vorhersage der Wirkungen. Diese Definitionen von Philosophie und Weisheit mit dem Unverstehen der unmittelbaren Zusammenhänge als Zusatzbedingung bedeutet, dass Philosophie per Definition überall auftreten kann, wo

- ein Sujet (noch) unverstanden ist,
 - die Grenzen eines Verständnisbereichs überschritten sind,
- aber
- noch rational gehandelt werden kann und soll, dies etwa im Gegensatz zur Religion.

Diese Definition schliesst einen leichtfertigen Umgang mit „dem Ganzen“ aus, sei es in der Betrachtung der Welt, sei es in der Medizin als Wunschkonzept: Philosophie wie Medizin sollen das Ganze ihres Feldes sehen, aber bei hinreichendem Verstehen der Unter-systeme. Sowohl in mancher „Philosophie“ wie in mancher „Medizin“ wird dieses Prinzip verletzt. Andererseits liegt es gerade in der menschlichen Natur (zumindest eines Teiles der Menschheit), Grenzen und Unverstandenes als besonders attraktiv zu empfinden – und dabei eine eigene Wahrheit zu finden, auch wenn die Grenzen der Rationalität selbst überschritten werden.

Aus den obigen Bedingungen der prinzipiellen Unklarheit folgt, dass philosophische Überlegungen

- in vielen Bereichen, auch des täglichen Lebens entstehen,
- und häufig zu verschiedensten Antworten, sogar zu verschiedenen Fragen, bei verschiedenen Autoren führen.

Schon der griechische Philosoph Aristoteles (384–322 v. Chr.) bemerkt, dass selbst gegen die Philosophie zu sprechen eine Art Philosophie ist.

Personen als Autoritäten sind dadurch im akademischen Arbeitsbetrieb der Philosophie wesentlich wichtiger als in Wissenschaft und Technik. In der Wissenschaft sollten z. B. das Experiment und die Konsistenz mit der zugehörigen Theorie den Ausschlag geben. Einstein als Autorität ist in der Naturwissenschaft eine grosse Ausnahme. In der Philosophie bleiben häufig auch konträre Gedanken als Ideen nebeneinander stehen. Wir werden dieses Verhalten mit der Entwicklung der angewandten Informatik vergleichen, bei der die Werkzeuge zum Bau und Verständnis zusammen mit dem Fortschritt und der Grösse der Systeme selbst in ihrer Mächtigkeit wachsen – und das Ganze muss funktionieren.

Viele Definitionen und Sätze der Philosophie sind Bonmots. Der wohl – zumindest bei jungen Leuten – berühmteste und „fundamentalste“ Philosophiespruch der Moderne stammt vom Schriftsteller und Dramatiker Douglas Adams. Es ist das obige verkürzte Zitat aus dem Roman „Per Anhalter durch die Galaxis“ aus den Jahren 1978 bis 1992. Der Spruch ist die Antwort auf die vom Autor Douglas Adams gestellte grosse philosophische Frage „nach dem Leben, dem Universum und dem ganzen Rest“ („life, the universe and everything“).

Der Computer des Universums gibt die ultrakurze Antwort „42“ – aber niemand kann mit der Antwort etwas anfangen, weil niemand weiß, wie die eigentliche „grosse philosophische Frage“ lautet: Sie wäre wohl die beste Definition von Philosophie. Trotzdem – es ist der Beginn einer Vielzahl von Deutungen und Spekulationen über die Zahl 42. Dieses Zitat und diese Formel für Philosophie haben Kultstatus erreicht. Softwareentwickler verwenden die Zahl „42“ als Füllzahl an Stelle des profanen „4711“ oder „0815“.

1.2 Prä- und Postphilosophie

Wir dürfen kaum sagen, dass wir weiter seien als Platon. Nur im Material der wissenschaftlichen Erkenntnisse, die er benutzt, sind wir weiter. Im Philosophieren selbst sind wir noch kaum wieder bei ihm angelangt.

Karl Jaspers, Philosoph, 1883–1969

Durch bloßes logisches Denken vermögen wir keinerlei Wissen über die Erfahrungswelt zu erlangen; alles Wissen über die Wirklichkeit geht von der Erfahrung aus und mündet in ihr.

Albert Einstein, Physiker, 1879–1955

Philosophie baut auf dem Stand der Wissenschaft und Technik ihrer Zeit auf. Die aktuelle Philosophie kann jeweils aktuelle Ergebnisse der Wissenschaft und aktuelle Entwicklungen in Gesellschaft und Technik als Grundlage mitverwenden. Dies gilt für die Philosophie der Natur wie für die des Wissens oder des Menschen. Entsprechende Umwälzungen mit direkter Auswirkung auf Philosophie fanden z. B. in Physik und Astronomie statt für die Naturphilosophie und Kosmologie und für die Gedanken zur Gesellschaft durch die technologische und soziologische Entwicklung der Industrie und Wirtschaft.

Die philosophischen Gedanken zu einem Kontext in den Anfängen einer wissenschaftlichen Erschliessung dieses Bereichs oder der technischen Entwicklung unterscheiden sich naturgemäss erheblich von Gedanken in einem fortgeschrittenen Stadium. Wir bezeichnen die Gedanken im Vorstadium als „Präphilosophie“, jene im hinreichend fortgeschrittenen Stadium des Gebiets als „Postphilosophie“ für diesen Kontext. Zur Illustration von Philosophie „davor“ und „danach“, d. h. vor einer Umwälzung im Verstehen und nach dieser Umwälzung, betrachten wir kurz zwei physikalische Begriffe, den Atombegriff in der Antike verglichen mit dem der Neuzeit, und die Identität von Dingen bei Leibniz und in der modernen Physik.

Akzeptiert man die Endlichkeit einer Unterteilung von Materie (d. h. von kleinsten Teilchen), so folgt recht schlüssig die Vorstellung von verschiedenen Formen und Wechselwirkungen (z. B. Häkchen) zur Erzeugung der Elemente, der Stoffe und der Geschmäcker. Allerdings ist die Forderung der Endlichkeit der Materie nicht trivial; noch Samuel Hahnemann, der Begründer der Homöopathie, bleibt 1810 beim Kontinuum und verdünnt seine Präparate als „Potenzierung“ nahezu *ad infinitum*, nach heutigem Verständnis weit unter die Feinheit der Atome. Für einige griechische Philosophen ist allerdings umgekehrt die ewige Fortsetzung der Teilung philosophisch suspekt (s. u. das Paradox von Zeno), und die Endlichkeit ist vorzuziehen. Erst die Physiker und Mathematiker des 16. bis 19. Jahrhunderts haben gelernt, mit „dem Unendlichen“ konsequent umzugehen.

Insgesamt erscheinen diese Modelle von „Atomen“ (nach Demokrit von Abdera, ca. 460–370 v. Chr.) oder der „Anu“ (in der indischen Vaisheshika-Katāna-Schule, ca. ein Jahrhundert v. Chr.) uns heute als grossartige philosophische Spekulationen. Dazu gehörten auch die Idee des Bewegungsmediums (des Vakuums) und bei Katāna eine Unterstruktur der räumlichen Teilchen (der Anu) in Gruppen von Unterteilchen (den Paramanu). Es war für über zwei Jahrtausende Spekulation – aus heutiger Sicht eine geniale Prä-Philosophie mit viel Glück, da es in die Richtung der späteren Atomvorstellung ging und dabei bis zum 19. Jahrhundert ohne die Chance eines Beweises war. Dazu war die Philosophie der Atome angefeindet (Platon soll sie gehasst haben), ja nahezu vergessen. In Tab. 1.1 finden sich einige wissenschaftliche Meilensteine des atomaren Konzepts bis zur Post-Philosophie.

Nach der Vorahnung der Atome und indirekten Hinweisen auf ihre Existenz in Chemie und Physik beginnen mit dem noch heute vertrauten Bohrschen Atommodell die Überlegungen (und Auseinandersetzungen) zur Bedeutung der entdeckten neuen Gesetze mit den Quanteneffekten im atomaren Bereich. Die neuen Beobachtungen widersprachen hart den aus unserer Umwelt gewohnten Gesetzen und erschütterten philosophische Grundbegriffe wie die Kausalität mit Aussagen zur prinzipiellen Unbestimmtheit und des Zufalls.

Die Diskussion hat neue Experimente provoziert, die wieder neue Erkenntnisse hervorbrachten. Philosophen und Physiker verarbeiten diese Umwälzung in „Postphilosophie“ oder „Postphilosophien“, etwa Werner Heisenberg in seinem Buch wie „Physik und Philosophie“ (1959); es ist zumindest aus der Sicht eines Physikers ein philosophisches Buch.

Die Umwälzungen durch die Informationstechnologien sind für menschliche Aspekte vielleicht noch wichtiger. Sie erfassen alle Bereiche mit Information und Informations-

Tab. 1.1 Der wissenschaftliche Weg zum Atom

Zeit	Entdeckung und Entdecker	Bedeutung
1803	Stöchiometrische Gesetze (John Dalton)	Beweis des „Digitalen“ Verhalten der Materie bei chemischen Reaktionen
1811	Avogadro Gesetz (Amedeo Avogadro)	Feste Anzahlen von Atomen pro cm ³ Gas, es gibt Moleküle
1865	Loschmidt	Bestimmung der Grösse und der Anzahl von Atomen pro Gramm
1905	Indirekte Beobachtung durch Brownsche Bewegung (Robert Brown/Albert Einstein)	Erster direkter physikalischer Beweis der Existenz von Atomen
1913	Semiklassisches Atommodell (Niels Bohr)	Erstes Verständnis vom Atomaufbau von verschiedenen chemischen Elementen

verarbeitung: direkt durch den Bau von Systemen und indirekt durch das Verständnis der Informationsprozesse in der Natur wie in der Genetik, der Evolution und den Vorgängen im Gehirn. Hier besteht die unmittelbare Möglichkeit, dass viele unserer Vorstellungen – analog der Entwicklung in der Physik – umgestossen werden und ganz neue Konzepte auftauchen.

Der Begriff der Identität von Dingen ist ein Einzelbeispiel der Veränderung der Diskussion um einen philosophischen Begriff vor und nach (oder während) der naturwissenschaftlichen oder technischen Erfassung. Es geht um die philosophischen Begriffe der Gleichheit und Ununterscheidbarkeit oder um die „Identikalität“: „*Der Status der Ähnlichkeit von etwas zu etwas anderem bis zum gleich erscheinen.*“ (Quelle: *Urban Dictionary*).

Die klassische philosophische Diskussion und Analyse zur „Identität von Ununterscheidbarem“ und umgekehrt von der „Ununterscheidbarkeit von Identischem“ stammt vor allem von Gottfried Wilhelm Leibniz (1646–1716). Seine Fragen sind etwa:

- Kann es überhaupt identische Dinge (wie zwei gleiche Blätter) geben?
- Wie können verschiedene Blätter an einem Baum aus ein und demselben Prinzip entstehen?
- Kann Gott identische Dinge schaffen?

Für Leibniz hängen Identität, Ununterscheidbarkeit und (fehlender) Grund für die Entstehung von ununterscheidbaren Objekten eng zusammen. Seine Schlussfolgerung ist, dass es „keine Sinnendinge geben könne, die absolut ununterscheidbar seien“, z. B. keine zwei identischen Wassertropfen oder identische Blätter. Dazu berichtet er von einem Garten-spaziergang im Park von Herrenhausen mit einer „grossen Prinzessin“, bei der „ein intellektueller Edelmann“ versuchte, für die Diskussion zwei gleiche Blätter zu finden – aber man sah doch immer einen Unterschied.

Durch die Entwicklung der Physik mit Atomen, Elementarteilchen und insbesondere der Quantenphysik hat die Frage der Identität neue Tiefe erhalten. Identität und Ununterscheidbarkeit sind fundamentale Aspekte der Physik geworden und hängen eng mit Kausalität, Zeit und Ort zusammen: Atome können identisch sein, sind aber noch lokal identifizierbar, die Nanotechnologie baut identische Körperchen. Elektronen und Photonen verlieren sogar den konkreten Ort und erzeugen kuriose, nie gedachte Effekte bis hin zur „spukhaften“ Quantenverschränkung. Der Begriff der „Identität des Ununterscheidbaren“ von Leibniz bekommt als Fermi- oder Bosestatistik und als Pauli-Prinzip eine unmittelbare fundamentale Bedeutung. Das philosophische Problem der Identität von Objekten ist zu Physik geworden. Wir werden unten versuchen, das Problem der menschlichen Identität zu erfassen und zum Teil zu Informationstechnologie werden zu lassen.

Die Aufgaben, die sich Leibniz gestellt hatte, sind also viel tiefer als gedacht – aber seine „Prä-Philosophie“ der Identität ist auch für die heutigen Post-Philosophen (und für die Physiker) ein Ausgangspunkt.

Die konkreten Verfahren zur Identitätsbestimmung mit Eigenschaften wie Namen, Adressen oder Fähigkeiten des Objekts sind Aufgaben aus dem Bereich der Informatik; es sind technische Aufgaben bei der Erstellung von Netzen und beim Lauf von Programmen. Wie identifiziert man ein Objekt, wie schreibt man ihm Funktionen und Attribute zu, wie vergleicht man oder „vererbt“ man Eigenschaften u. v. m. in der Software. Die Gedanken von Leibniz zur Identität zweier Objekte und zur Ununterscheidbarkeit werden stillschweigend verwendet, etwa als sog. Gesetz von Leibniz:

1. *Wenn x identisch mit y ist, dann hat y auch jede Eigenschaft, die x hat, und umgekehrt,*
2. *Wenn y jede Eigenschaft von x hat, und x jede Eigenschaft von y , dann ist x identisch mit y .*

Aus der „Prä-Philosophie“ erwächst in der Softwaretechnologie eine ausgefeilte präzise technische Unterdisziplin mit Systemen, die mit Millionen, ja Milliarden von Objekten umgehen. Im nächsten Kapitel betrachten wir einige Bauregeln dafür. Es ergeben sich neue philosophische und soziologische Fragen, etwa zu den Eigenschaften und inneren Gesetzen derartiger ultragrosser Mengen von Objekten mit der Bildung von Schwärmen und der Verbreitung von Ideen.

Während im Falle der Physik und der Neurowissenschaften das Stadium der atomistischen Post-Philosophie durch unser „echtes“ Eindringen in atomare Grössenordnungen geschieht, ergibt sich dies im Falle der Informatik vor allem durch den Akt des Bauens an sich. Wir sind nicht nur Zuschauer, wir analysieren uns nicht nur mühsam selbst aus uns heraus, sondern verstehen uns und die Welt durch und im Zusammenhang mit dem Bauen. Hier zwei Zitate, die das Verstehen durch Konstruieren betreffen, einmal vom chinesischen Philosophen Konfuzius (551–479 Chr., zumindest zugeschrieben):

Ich höre und vergesse, ich sehe und behalte, ich tue es und verstehe.

Eine „philosophischere“ Sequenz des Umgangs mit Wissen kommt ebenfalls aus dem asiatischen Raum und spricht in unserem Sinn unmittelbar die Post-Philosophie an:

„*Shu – Ha – Ri*“ hier übersetzt mit „*Lernen – Loslassen – Transzendieren*“.

Es ist das Konzept zur Vollendung in der japanischen Kampfkunst Aikido, das Kampfsport, Religion und Philosophie vereinigen möchte. Wir denken, die Zeit ist reif, einige Themen der Informationsverarbeitung „loszulassen“, weil wir hinreichend komplexe Systeme bauen können. Dazu die nüchterne Ingenieursmantra vom amerikanischen Nobelpreisträger Richard Feynman (1918–1988):

„What I cannot create, I do not understand“ –
 „Was ich nicht machen kann, verstehe ich nicht.“

Software ist die Disziplin vom Erschaffen funktionierender komplexer Systeme und damit zu ihrem Verstehen. Ende des 18. Jahrhunderts hatte man schon das Gefühl, jetzt die Welt als Bauwerk und als Ganzes zu verstehen, genauer als ein grosses Uhrwerk. Der philosophische Grundgedanke (aus der Astronomie stammend, glänzend von Pierre Simon Laplace formuliert als dem „Laplaceschen Dämon“, der alles weiss) war ein deterministischer Materialismus: Das Geschehen der Welt war durch die solide Materie, deren Anfangsbedingungen und ihre Bewegungsgesetze bestimmt. Unser Alltagsdenken ist immer noch so klassisch geprägt. Seit dem 19. Jahrhundert wissen wir, dass Materie nicht „solide“ ist, sondern mit abstrakten elektrischen und magnetischen Wechselwirkungen „funktioniert“, und auch nicht hart deterministisch ist, sondern chaotisch und unscharf. Das Wort „Determinismus“ ist ein schlechtes Wort geworden und ist zu vermeiden, obwohl die Welt den Naturgesetzen folgt – aber eben nicht wie ein lineares Uhrwerk.

Wir erklären die Welt mit zwei notwendigen Säulen:

- Physik und den Turm von Wissenschaften (und Technologien) darüber, etwa die Chemie,
- Informatik und den Turm von Technologien (und Wissenschaften) darüber.

Software ist in der Säule „Informatik“ der Stoff, der Vorgänge abwickelt und sie abweichen lässt vom natürlichen Lauf. Ein einfaches Geschoss fliegt ballistisch (es gehört nur zur ersten Säule), ein gesteuertes Geschoss dagegen mit zusätzlicher Energie auf einer (nahezu) willkürlichen Bahn mit Software, die den Antrieb steuert (Säule 2 und Säule 1).

Beide Säulen sind von immenser Bedeutung: Nur die unbelebte Welt (etwa die Sterne) ist allein Säule 1, in der belebten und technischen Welt treffen sich beide z. B.

- im Computer in den Transistoren und in der Software,
- im Menschen in der Biologie mit Genetik und Leben; vielleicht weniger offensichtlich, im intellektuellen Leben mit Gehirn, Gefühlen und Sprache.

Die Kenntnis von der grundsätzlichen Bedeutung der Informatik als Weltsäule ist noch kaum verbreitet: Vorläuferideen sind etwa „Der rechnende Raum“ (1946) bei Konrad

Zuse oder die Theorie der „zellulären Automaten“ von Stanislaw Ulam und John von Neumann (etwa 1940). Die philosophischen und physikalischen „tiefen“ Zusammenhänge der beiden Säulen sind nach unserer Ansicht nicht verstanden, haben aber ganz konkrete Auswirkungen für Computer, die sehr wenig Energie verbrauchen sollen. Ein physikalischer Zusammenhang ist der Energieverbrauch bei Informatikoperationen: Jedes Bit, das gelöscht werden muss, verbraucht nach Rolf Landauer (1961) eine winzige, aber nicht unterschreitbare Energie. Unsere aktuellen Computer brauchen allerdings noch das hundert- oder tausendfache davon.

Amüsanterweise erinnert die Existenz der beiden Säulen an den klassischen philosophischen Konflikt „Materialismus“ und „Idealismus“: Die Materie ist durch Quantentheorie und Elementarteilchen nicht mehr hart, und die Gedanken und Ideen sind nicht mehr freischwebend, sondern angewandt und eng verbunden mit der Materie. Zur Information gehört wenigstens ein kleines physikalisches Element wie ein Atom, ein Lichtquant oder ein Quantenzustand pro Bit. Wir kennen kein Wort, das diese dualistische Weltgrundlage treffend beschreibt: Solange die angewandte Informatik aus kleinen Programmen in grossen Kisten bestand, war der Gedanke einer Zweisäulentheorie der Welt auch nur spekulativ. Angesichts der praktischen Bedeutung der IT gilt das Interesse primär den „Apps“ und den Infrastrukturen der Unternehmen und ähnlichem und nicht der Bedeutung für das Weltverständnis. Aus Physikersicht könnte man heute die Informatiksäule als die neue „Metaphysik“ ansehen im Sinne des griechischen Wortes *μετά μετά* „danach“, „hinter“, „jenseits“.

Heute ist Software (auf hinreichend schneller Hardware) der „Stoff“, der diese Umwälzungen und Erkenntnisse durch Informatik realisiert. Beispiele von zu erwartenden (oder bereits laufenden) Veränderungen von philosophischem Gehalt und damit zu einer Postphilosophie betreffen etwa die Begriffe

- Identität (von uns, von den Dingen, der Welt),
- Wissen und Erkennen,
- Bewusstsein,
- Sprache,
- Intelligenz und das Verhältnis zum Geist,
- Komplexität und Aufbau von Systemen, insbesondere von sehr grossen Systemen.

Dieses Buch will den Einfluss der „zweiten Säule (Informatik)“ auf die Philosophie diskutieren. In den nächsten Unterkapiteln zeigen wir Bereiche der Philosophie auf, die technisch von Softwarekenntnissen gewinnen können.

1.3 Philosophie der Systeme: Architekturen, Ontologie des Seins und des Werdens, Epistemologie

„Architektur beruht auf drei Prinzipien: Firmitas (Festigkeit, Stabilität), Utilitas (Zweckmässigkeit, Nützlichkeit) und Venustas (Anmut, Schönheit).“

Vitruv, römischer Architekt unter Kaiser Augustus, 1. Jahrhundert v. Chr.

„Ich verstehe unter einer Architektonik die Kunst der Systeme ... ich verstehe aber unter einem Systeme die Einheit der mannigfaltigen Erkenntnisse unter einer Idee.“

Immanuel Kant, Philosoph, in „Kritik der reinen Vernunft“, 1781

Wir betrachten in diesem Abschnitt recht oberflächlich und un-fachphilosophisch einige philosophische systemnahe Begriffe, die sinnvoll in die Informatik übertragen werden konnten und können. Sie verbreiten dort „intellektuellen Glanz“ oder erlauben zumindest lehrreiche Analog- und Rückschlüsse von der Informatik in die Philosophie und umgekehrt.

Eine Hauptaufgabe der Philosophie ist es, die Erkenntnisse der Welt zu einem System zusammenzustellen. Dabei ist „*das Ganze also gegliedert (articulatio) und nicht gehäuft (coacervatio)*“, schreibt Immanuel Kant in der Kritik der reinen Vernunft. Und weiter zunächst „*wie Gewürme, durch eine generatio aequivoca, aus dem bloßen Zusammenfluß von aufgesammelten Begriffen, anfangs verstümmelt, mit der Zeit vollständig.*“ In der Software gibt es „für das Gewürme“ das Wort „Spaghetticode“ für entsprechende zusammengeschiedene oder zusammengeflickte Programme.

Die Gliederung und die konzeptuelle Zusammenfassung werden allgemein durch die Struktur gegeben, genauer durch eine Architektur des Systems. Kant schreibt weiter:

... architektonisch, um der Verwandtschaft willen und der Ableitung von einem einigen obersten und inneren Zwecke, der das Ganze allererst möglich macht, kann dasjenige entspringen, was wir Wissenschaft nennen, dessen Schema den Umriß (monogramma)] und die Einteilung des Ganzen in Glieder, der Idee gemäß, d. i. a priori enthalten, und dieses von allen anderen sicher und nach Prinzipien unterscheiden muß.

Kant fasst den Begriff „Wissenschaft“ recht weit. In der Übertragung auf die Informatik geht es um die konkrete Architektur des Betriebssystems für einen Organismus, im kommerziellen Bereich etwa um die Organisation eines Unternehmens. Zentral für die Architektur ist die Analyse aller Grössen („Entitäten“) im System: konkrete und abstrakte Gegenstände, Eigenschaften, Sachverhalte, Ereignisse, Prozesse und ihre strukturellen Beziehungen untereinander. Das klassische Begriffspaar für diese Analyse (seit Heraklit und Aristoteles) ist die berühmte Unterscheidung von „Sein“ und „Seiend“, die sich als Grundsatzdiskussion durch die Geschichte der Philosophie zieht. Wir werden diese Diskussion im Sinne der Software mit „Werden“ ergänzen:

„Seiendes“ (that which has being) bezeichnet Konkretes wie existierende Dinge oder Individuen, z. B. Sie (den momentanen Leser).

„Sein“ (being) bezeichnet allgemein die Existenz dieser Dinge ohne Bezug auf die individuellen (äusseren) Eigenschaften.

Als Beispiel der Unterscheidung die Kategorie „Mensch“. Sie, der Leser, sind ein „Seiendes“ – eine körperliche, universelle Repräsentierung. Aristoteles schreibt hier zum Sein: „*Das Mensch-sein ist ein Vernünftig-Wesen-sein*“ als bewusste Betonung von „*der Mensch ist ein vernünftiges Wesen*“. In der Geschichte der Philosophie wird dieses Wesentliche (das „Sein“) auf fiktive Vorstellungen von „Substanz“ oder „Gott“ zurückge-

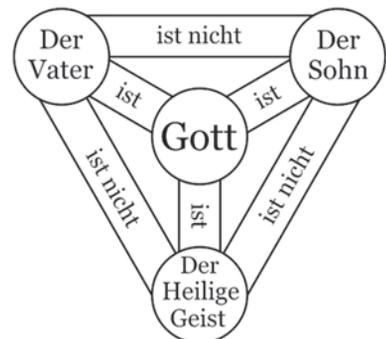
führt. Aus der konkreten Sicht des Softwareentwurfs ist es die Sammlung der fundamentalen Eigenschaften, Fähigkeiten und Anwendungen auf die „Kategorie“ Mensch oder eines Exemplars „Mensch“.

Diese allgemeinen Überlegungen heißen in der Philosophie (und konkret in der Softwaretechnik) *Ontologie*. In der Philosophie bezieht sich die *Ontologie* auf „die Welt“ in der Einzahl – also die ganze Welt, aber jeweils verschieden von verschiedenen Philosophen betrachtet. Es geht im Prinzip um das Wesen der Dinge in einem absoluten Sinn. In der Anwendung des Begriffs in der Software geht es im Allgemeinen um die formale Beschreibung der Welt von gegebenen Aufgaben und Aufgabenbereichen. *Ontologie* ist dabei ein recht neues Kunstwort aus dem 17. Jahrhundert (griechisch *ὄν*, *on*, „seiend“, als Partizip Präsens zu εἶναι, *einai*, „sein“, und *λόγος*, *logos*, „Lehre, Wort“), seit etwa 1990 auch in der Informatik verwendet. Ein kleines, aber wunderbares christliches Beispiel einer *Ontologie* aus dem 13. Jahrhundert ist die Inschrift der „Schilder des Glaubens“ (Abb. 1.1), das die Interpretation der christlichen Dreifaltigkeit und damit Gottes visualisiert, schematisiert und strukturiert. Das Bild enthält zwölf logische Aussagen wie „Vater ist Gott“ usf.

Eine triviale Untermenge einer *Ontologie* ist eine *Taxonomie*, eine hierarchische Anordnung und Auflistung von Objekten wie aus der Biologie bekannt als Klassifikation von Pflanzen oder Tieren. *Taxonomien* stehen häufig am Beginn einer Wissenschaft oder auch am Anfang des Entwurfs eines Softwaresystems. In der Softwaretechnologie wird das Architekturprinzip zu einer regelrechten Konstruktionslehre ausgebaut, naturgemäß viel präziser und fassbarer als die verbalen philosophischen Systeme mit (aus Ingenieurssicht) vagen Begriffen, persönlichen und zeitbedingten Eigenheiten und Mehrdeutigkeiten.

Ontologie beschreibt Grundelemente der Welt wie die Substanzen, d. h. statisch. Dazu passt die Vorstellung des Aristoteles von der Welt als unvergänglich und ursprungslos. Aber selbst der Sternenhimmel hatte sich schon in der Antike als nicht unveränderlich erwiesen (Hipparch hatte im 2. Jahrhundert die Bewegung des Frühlingspunktes entdeckt). Eine andere extreme Haltung vertritt Heraklit von Ephesus (ca. 520–ca. 460 v. Chr.) in seiner Lehre von allen Dingen in Bewegung:

Abb. 1.1 Eine *Ontologie*: Die Inschriften des Schilds der christlichen Dreifaltigkeit nach dem Athanasischen Glauben. (Aus Wikimedia Commons, nach Francis McLloyd und Anon Moos)



„Pánta chorei kai oudèn ménei“ – „Alles bewegt sich fort und nichts bleibt.“

Oder das berühmte Flussgleichnis: „*Wir steigen in denselben Fluss und doch nicht in denselben, wir sind es und wir sind es nicht.*“

Die Metapher vom fließenden Wasser im festen Flussbett mit dem „Sein und Werden“ passt gut zur Realität, besser als Statik. Nach Jürgen-Eckardt Pleines, (2002) sagt Heraklit:

Denn was wäre Sein ohne Werden? – eine unerkennbare, gestaltlose Masse ohne Struktur und Leben; und was wäre Werden ohne Sein? – eine unerkennbare Bewegung ohne Richtung und Zweck, eine Veränderung von nichts zu nichts.

Dem Werden entspricht in mehr technischer Bezeichnungsweise die Menge der Prozesse zwischen den Objekten und passt damit sehr gut zum Begriff der Software. Es ist allerdings kein passives Werden, sondern es ist Bauen und Werden. Jede grössere Softwareentwicklung ist ein Werden in einem Strom. Der Projektmanager versucht, den Strom im Flussbett zu halten – aber manchmal verlagert sich das Bett und es wird ein anderes Projekt daraus, manchmal versandet es sogar. Softwareprojekte strukturieren und bauen die „Welt“ unter menschlicher Leitung ganz ähnlich wie die Evolution „Software“ baut und Strukturen schafft (wenn dort auch der Zweck nicht nachfragbar ist). Ein Begriff, der per Definition den Konflikt von Sein und Werden adressiert, ist der Begriff der Identität, die wir in Kap. 9 beschreiben.

In der Moderne gibt es eine Philosophierichtung, die bewusst das Werden in den Mittelpunkt stellt, die Prozessphilosophie oder die „Ontologie des Werdens“. Die bekanntesten Vertreter sind die amerikanischen Philosophen Alfred North Whitehead (1861–1947) und Nicholas Rescher (geb. 1928). Obwohl viele der Aussagen der Prozessphilosophen metaphysisch sind, literarisch ja theologisch, sind Grundbegriffe dieser Denkrichtung wie „Prozess“ und „Netzwerke“ (von Prozessen), technisch gesehen, Softwarebegriffe. Software baut die Prozesse und betreibt sie in ungeahnter Komplexität und in vielen Bereichen von Natur und Gesellschaft. Software umfasst das „Sein“ und das „Werden“ in verschiedenen Stufen getrennt oder eng verschränkt, besonders schön im „Strömenden Computing“ wie es im Kapitel „Bewusstsein“ diskutiert wird. Es ist die Vermutung des Autors, dass man an den Eigenschaften der nüchternen kommerziellen Softwareentwicklung Hilfen und Wegweiser finden kann für die Entwicklung der (Prozess-) Philosophie.

Die philosophische Ergänzung zur Ontologie, die das existierende Sein beschreibt, ist die Epistemologie oder Erkenntnislehre, die den Weg zur Beschreibung der Welt untersucht, einschliesslich des „Höchsten“. Es geht um Fragen wie: Woher haben wir die Kenntnis der Objekte? Wie grenzen wir sie untereinander ab? Wie gehen wir dazu systematisch vor? Haben wir die „richtigen“ Objekte gefunden? Ist diese Einteilung optimal? Haben wir Widersprüche?

Das Wort Epistemologie selbst ist mit dem Griechischen ἐπιστήμη *episteme* „Wissen“, „Wissenschaft“ oder „wahre“ Erkenntnis in 19. Jahrhundert geprägt worden als Parallele zur Ontologie. Bei der Software haben wir beim Entwurf eines Systems diese „episte-

mologische“ Aufgabe zu lösen und die richtigen Grössen und die Operationen zwischen ihnen zu definieren. Im Sinne der Prozessphilosophie und erst recht der Software ist die Epistemologie von der klassischen Statik („Wie erfahren wir, was existiert?“) zu erweitern auf Interaktionen und Prozesse („Was läuft ab?“ „Wo kommt es her?“ „Wie funktioniert es im Detail?“). In der Technik ist der harte Kern dieser praktischen Epistemologie das „Reverse Engineering“ (etwa von alten Programmen, die niemand mehr versteht, s. Kap. 8).

In der Software sind verschiedene Arten von Wissen, wie sie in der philosophischen Literatur be- und umschrieben werden, zu finden und werden präzise verwendet, explizit oder implizit. Einige der prägnanten Kurzformen sind:

Wissen, dass („knowledge that“), Wissen, wie („knowledge how“), Wissen direkt („acquaintance-knowledge“) und Wissen indirekt („by description“).

Zur Illustration hier Beispiele zu den Wissenstypen

„Wissen, dass“ $2 + 2$ das Ergebnis 4 ergibt,

„Wissen, wie“ wie man addiert,

die Art und Weise der Wissenserlangung (Bertrand Russell, 1912),

„Wissen direkt“ es im Programm unter eigener Kontrolle zu errechnen,

„Wissen indirekt“ es über ein Beispiel zu wissen, es von aussen zu importieren, oder bezogen auf den Zeitpunkt

„Wissen a priori (vorher)“ bei naivem Kenntnisstand von einem Phänomen und dessen Ursachen,

„Wissen a posteriori (nachher)“ nach dem Zeitpunkt des Erlangens der wesentlichen Erkenntnisse.

Am Beispiel der Wissenschaftsphilosophie im Vergleich zur Software lässt sich die prinzipielle Äquivalenz von Wissen in Philosophie und Wissen in einer Software zeigen (Tab. 1.2).

In der Software wird ein wachsender Anteil von Wissen importiert, sowohl als Methodik als auch direkt als arbeitsfähige Funktion. Der fertige Code ist das absolute Wissen. Er bestimmt die wirklichen Abläufe. Dazu hat man bei der Software (meistens) das Wissen, was man (oder der Kunde) eigentlich mit dem Ganzen erreichen will in Form eines Pflichtenhefts (den „Specs“).

Die strukturelle Verwandtschaft des (Natur-) Wissenschaftssystems und von üblichen kommerziellen oder technischen Softwaresystemen beruht auf einem gemeinsamen epistemologischen Prinzip, dem Konstruktivismus. Es handelt sich nicht um eine geisteswis-

Tab. 1.2 Schematischer Vergleich „Wissen in Wissenschaft“ und „Wissen in einer Software“

	Wissen, dass	Wissen, wie	Wissen, direkt	Wissen, indirekt	Wissen, a priori	Wissen, a posteriori
Wissenschaft	Schon Bekanntes	Gesetzmässigkeit	Eigene Experimente	Fremde/ historische Experimente	Bekannte Grundlagen	Dazugelertes
Software	Erfahrung	Operationen, Vorgehen	Selbst kodiert	Importierter Kode	Gegebene Information	Erfahrung

senschaftliche und strukturell „sanfte“ konstruktivistische Richtlinie wie etwa in der Erziehung oder für das Lernen, sondern um den exakten und formalen Aufbau von Systemen:

- Konstruktion des Systems Naturwissenschaften mit der Korrektur der menschlichen Ideen durch Experimente,
- Konstruktion von Software nach den Vorgaben und der Grundidee und Korrektur bis zum Funktionieren nach dem Pflichtenheft (der „Spec“).

Als Sinnbild des Konstruktivismus zeigt Abb. 1.2 eine Pagode, die grosse Wildganspagode von Xi'an, als ein Symbol für hierarchische Architekturen für Wissen wie für Software-systeme. Die chinesische Pagode enthält zumeist im Erdgeschoss das Hauptbildnis Budhas und in der Mitte eine durchgehende Säule, die Spitze symbolisiert das Himmelreich.



Abb. 1.2 Die grosse Wildganspagode von Xi'an in China als Sinnbild des Konstruktivismus. (Bild: Wikimedia Commons „Giant Wild Goose pagoda“, Autor: Kgbkbgkb)

Als Analogon für ein Softwaresystem könnte das Erdgeschoss der Pagode der Hardware der Computer entsprechen, die mittlere Säule der tragenden Idee des Programms und die Spitze dem Management oder dem Auftraggeber. Eine Struktur in Schichten ist ein wesentliches Prinzip zum Aufbau grosser Systeme und umgekehrt auch eine Voraussetzung für deren Verständlichkeit.

Konstruktion ist die philosophische Grundlage der Leonardo-Welt des Philosophen Jürgen Mittelstrass. Die Konstruktion des wissenschaftlichen Systems durch Menschen bedeutet keineswegs, dass das Ergebnis willkürlich ist und die Aussagen subjektiv wären. Die menschlichen Ideen werden von der Natur unbarmherzig abgelehnt und korrigiert. Der Konstruktionsprozess und der Wissenschaftsbetrieb selbst sind menschlich – das Ergebnis ist jenseits des Menschlichen und beschreibt die Gesetze des Kosmos und der atomaren Welt mit unvorstellbarer Präzision, z. B. repräsentiert durch die etwa 15 geltenden Ziffern der Werte einer Atomuhr. Funktioniert der Korrekturmechanismus für falsche Aussagen nicht oder unzureichend, wie es naturgemäss in Bereichen der Medizin oder Soziologie der Fall ist, so bleibt man in der menschlichen Grauzone stehen und die Aussagen sind zweifelhaft¹: Wer kann genau sagen, was „gesund“ ist?

Bei der Konstruktion von kommerzieller Software ist der Mensch der „Intelligente Designer“ im Sinne des Kreationismus, jedenfalls im Allgemeinen und im Prinzip – trotz vieler Fehler in kommerzieller Software muss man ihn so nennen (aber es gibt ihn nicht in der Evolution). Allerdings ist es auch möglich, Code selbst automatisch zu verändern oder gar zu erzeugen und gerade nach dem Evolutionsprinzip für einen Zweck die beste Variante auszuwählen oder selbst auswählen zu lassen. Der grösste Teil des Programmcodes wird sowieso automatisch erzeugt. Wir hatten bei der Entwicklung neuer Prozessoren tagelang automatisch primitive künstliche Programme erzeugt, die vom Inhalt her meistens sinnlos oder gar verboten waren mit sinnlosen oder nicht erlaubten Folgen von Operationen – die aber wunderbar geeignet waren, um neue Hardware auf Baufehler zu testen.

Da die Evolution des biologischen Lebens auf der Erde die Entwicklung einer grossen und wachsenden „natürlichen“ Software ist, bei der die Umgebung die Korrekturen negativ ausliest, bekommt die Softwareentwicklung für uns als Menschen eine unmittelbare Bedeutung. Der amerikanische Mathematiker Gregory Chaitin (geb. 1947) hat dies prägnant formuliert:

„Evolution is an up-hill random walk in software space“, etwa zu übersetzen als
„Die Evolution ist ein aufwärtsführender Zufallsweg im Raum der möglichen Software“.

Heute wird die Evolution allerdings auch vom Menschen selbst unmittelbar als dem „intelligentem Designer“, weitergeführt. Dies geschieht etwa im Craig-Venter-Institut im Bereich der synthetischen Genomik, beim Bau synthetischer und der Veränderung natürlicher Gene – zusätzlich zur (vermutlich) weiter existierenden Evolution ohnehin.

Mit Hilfe konstruktivistischer Epistemologie definieren wir auch „Verstehen“ im harten Sinn. Konstruktivistisch heisst in der Wissenschaft, dass wir zum Verstehen einen

¹ Unsicherheit und Sicherheit von Aussagen werden im Buch „Die unheimliche Beschleunigung des Wissens“ (2012) vom gleichen Autor besprochen.

Turm von Wissen aufbauen von Grundlagen bis zur Oberfläche mit dem Phänomen selbst – und alle Schichten sind dafür notwendig. Genau dies ist das Vorgehen bei der Konstruktion von Software: In der Tat haben wir den strikten Aufbau in Schichten als Prinzip zur Beherrschung der Komplexität gelernt.

Wir sehen in der Wissenschaft „echtes“ Verstehen proaktiv: Es muss die Zukunft erklärt werden, Erklärungen *a posteriori* sind nur beschränkt tauglich. Wer versteht, kann ein Phänomen, etwa den Fall eines Körpers, vorhersagen. Bei der Erklärung *a posteriori* gibt es leicht Schlupflöcher, sei es unbewusst, sei es per Methode etwa durch Anpassung von inneren Parametern in einem Modell. Der dänische Physiker Niels Bohr hat genau dies im oft zitierten Spruch (vielen anderen zugesprochen und oft nur als oberflächliches Bonmot verstanden) ausdrücken wollen:

Vorhersagen sind schwierig, vor allem in die Zukunft.

Wir sagen, dass wir ein Phänomen „stark“ verstehen wenn

- wir hinreichend erfolgreiche Vorhersagen des Phänomens machen können,
- diese Vorhersagen auf Grund eines kausalen Systems gemacht werden,
- die Rolle des Zufalls hinreichend berücksichtigt ist.

Dazu gehört also nicht nur ein beschreibendes Modell (durch die Naturgesetze, etwa das Gravitationsgesetz zwischen Massen), sondern sogar die aktive Anwendung des Modells, etwa die Lösung von Differentialgleichungen, analytisch, d. h. von Hand, oder im Grossen per Computer.

Wir definieren dieses wissenschaftstheoretische Verstehen ganz analog und entsprechend einer Softwarekonstruktion im nächsten Abschnitt. Eine erfolgreiche wissenschaftliche Vorhersage entspricht dort den Resultaten des Programmlaufs, das kausale System dem selbstgemachten Stapel (dem „Stack“) der Software. Der Prozess zur Entwicklung von (menschengemachter) Software mit Designphase entspricht dem Try-und-Error-Prozess beim Bau der Wissenschaft, die eingebaute Logik der Software den Gesetzen der Natur und die Ausführungsphase des Programms vergleichen wir mit der Durchführung der Experimente. Diese Analogie ist kein Zufall – die Aussagen der Wissenschaft können als grosses Softwaresystem aufgefasst werden. Dies ist unmittelbar einsichtig bei astronomischer Software zur Berechnung der Positionen von Gestirnen – Planeten, Mond, Doppelsternpositionen etc. zu einem Zeitpunkt soweit die Gesetze in Zukunft und Vergangenheit reichen (oder besser die Genauigkeit unserer Ausgangswerte). Diese Software bildet die Gesetze der Himmelsmechanik ab. Überall, wo etwas entsteht, wächst oder sich verändert, werden Instruktionen ausgeführt oder man kann es in der Form von Instruktionen und deren Ausführung sehen. Software kombiniert damit den Aufbau und das Verstehen, die Ontologie und die Epistemologie. Am Beispiel des Sonnensystems beschreiben Software und Information den Bau und den Status des Systems und Software „verstehet“ die Vorgänge und Bewegungen im feinen Detail und damit auch im Allgemeinen.

Aus dem Satz des Heraklit „*panta rhei*“ – „*alles fließt*“ wird der ergänzende Satz: „*alles, was fließt, ist Software*“, in pseudo-antidem Griechisch „*πάντα ῥεῖ, λογισμικό*“.

Dann gilt für die Wahrheit und das Verstehen der gleiche Grundsatz für die Software in kommerziellen Programmen wie in der Biologie, aber auch für die Physik, für den Bau einer Maschine, ja für einen Vorgang im Unternehmen mit seiner Betriebssoftware (z. B. von SAP). Der Programmierer Robert C. Marti hat es für die Praxis gesagt, aber es ist universell gültig:

„Truth can only be found in one place: the code“ – „Die Wahrheit findet man nur an einer Stelle: Im Programm selber.“

Genau dies hat man mit der Entschlüsselung des Genoms ja auch versucht und arbeitet weiter daran! Auch im Unternehmen geschieht genau das, was die Unternehmenssoftware mit den codierten Prozessen vorgibt (auch wenn man etwas anderes erwarten würde), es sei denn, ein mutiger Manager überschreibt ihre Wirkung durch seine Autorität.

Das von uns definierte „harte“ Verstehen durch Bauen beinhaltet, dass alle Komponenten und ihre Beziehungen verstanden und wohldefiniert, ja formal sind. Die Hermeneutik (altgr. ἐρμηνεύω *hermēneuō* ‚erkläre‘, ‚lege aus‘) löst die allgemeine philosophische Aufgabe, in einer nicht (vollständig) verstandenen oder verschwommenen, nicht rigoros durchkonstruierten Umgebung die Rolle eines Objekts, eines Begriffs oder einer Geschichte im ganzen Zusammenhang zu verstehen. Die klassische Aufgabe der Hermeneutik ist die Auslegung von Texten, vor allem von Texten mit „verschlüsseltem“ Inhalt – absichtlich wie bei einem Orakel, vielleicht absichtlich wie in der Bibel und wohl unbeabsichtigt wie im Wortlaut von Gesetzen:

‘In Christus?’ Noch so ein nicht gerade leicht zu verstehendes Bild!
Markus Langer, Dominikaner, Predigt 2009

Speziell bei der Auslegung spezifischer religiöser Texte wie der Bibel, der Tora oder dem Koran spricht man von Exegese. Das Verstehen historischer Texte bedeutet das doppelte Einpassen einer Textstelle:

- einmal in das Gefüge der Ursprungszeit mit allen Nebenbedingungen wie die Einbindung in eine kulturelle Umgebung, in eine bestimmte Zeit mit bestimmten Allegorien, Abweichungen von Regeln, Selbstbezug usw.,
- zum anderen in die heutige Denk- und Begriffswelt.

Der Abstand dieser beiden Welten heisst „hermeneutische Distanz“. Diese Distanz ist z. B. bei der Unterhaltung von zwei kommunizierenden Fachleuten der gleichen Zeit und Kultur klein, im Dialog zwischen einem Laien und einem Fachmann über einen Fachartikel über ein Patent.

Da dieses „Verstehen“ das Zusammenwirken vieler Begriffe aus der Wissensdomäne verlangt, bietet sich das Bild eines Netzwerks als das einfachste Modell einer Wechsel-

wirkung an. Das Wort „Text“ heisst ja selbst auch schon „Netz“ nach dem lateinischen „*textus*“ – Gewebe. Wir haben den speziellen Sinn eines Phänomens erfasst, wenn wir ein hinreichendes Netzwerk von Begriffen (und Referenzen zu Begriffen) haben, in das sich das Phänomen (die Textstelle z. B.) einfügt.

Das wohl erste historische Netzwerk zum Verstehen stammt vom antiken indischen Philosophen Panini (im 5. oder 4. Jahrhundert v. Chr.). Panini schuf eine Grammatik der Sprache Sanskrit mit zugehörigen Begriffen und beschrieb dazu insgesamt 3959 Regeln! Seine Arbeit war ein Vorläufer der Linguistik und einer modernen Programmiersprache (und Software, Vikram Chandra, 2014).

Das Knüpfen von Netzen mit Wissensknoten und Beziehungen untereinander ist eine Aufgabe der praktischen Informationsverarbeitung. Das „Verstehen“ der allgemeinen Welt, gekennzeichnet etwa durch das Wissen des Gebildeten, ist allerdings ein recht grosses Netz – und die Zahl der Verknüpfungen ist es erst recht. Aber es ist technisch eine wichtige Aufgabe der praktischen Informatik mit z. B. dem Internet als Kern. Im allgemeinen Fall der Betrachtung und Interpretation aus einer Welt heraus auf eine ganz andere (wie ein biblisches Gleichnis in die moderne Sprache und unsere Welt zu transponieren) geht es darum, das Ursprungsnetz „sinngemäss“ in das neue Netz zu transponieren. In den IT-Anwendungen zu Verstehen und Übersetzung wird dies direkt versucht.

Abbildung 1.3 zeigt als Beispiel die Beziehungen von Patenten eines engen Fachbereichs der Nanotechnologie als Netzwerk in einer Grafik. Es enthält die Ontologie eines kleinen beschränkten Wissensbereichs. Die informationstechnischen Wissenssysteme wachsen auf breiter Front und die enthaltenen Fakten übersteigen alles Wissen eines Einzelnen um Grössenordnungen.

1.4 Philosophische Kuriositäten: Paradoxien

Der Anfang aller tiefen Philosophie ist immer die Anerkennung von Schwierigkeiten, die scheinbar jeden Weg zur Klarheit blockieren – ist immer das Gefühl des Paradoxen.
Hilary Putnam, Philosoph, geb. 1926

Für viele Menschen ist „Philosophie“ und „philosophisch sein“ synonym mit der Erörterung von Fragen aus der Antike, die damals nicht klar zu beantworten waren oder noch nicht sind. Im Zusammenhang mit Technologie geht es uns zunächst um an und für sich konkrete Probleme. Auch die paradoxe Behauptung des Zenon von Elea (490–430 v. Chr.) beschreibt eine konkrete Situation. Hier berichtet von Aristoteles:

Achilles will mit einer Schildkröte ein Wettrennen machen. Da die Schildkröte langsamer ist, bekommt sie einen Vorsprung. Nun geht das Rennen los. Achilles erreicht den Punkt an dem die Schildkröte zum Startzeitpunkt ist. In der Zwischenzeit ist sie aber auch ein Stück vorangekommen. Nun erreicht Achilles diesen Punkt und die Schildkröte ist wieder ein Stück weiter. So setzt sich das Rennen fort und Achilles wird seinen Gegner nicht einholen können, da dieser, wenn Achilles den jeweils letzten Punkt des Tieres erreicht, schon wieder ein Stück vorwärts ist.

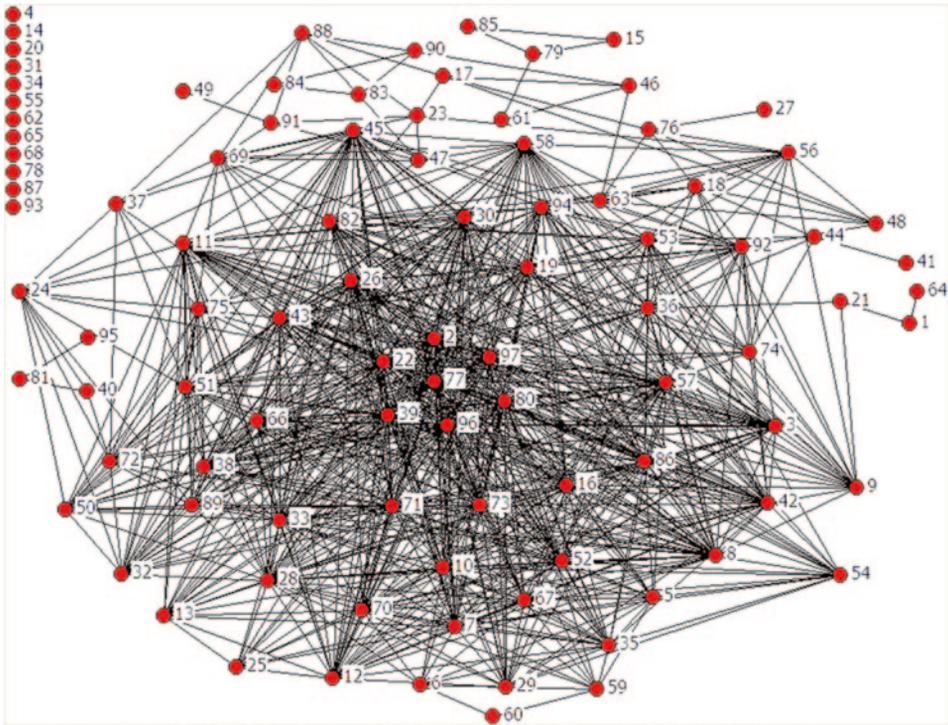


Abb. 1.3 Ein Wissensnetzwerk als Resultat der Analyse von Patenttexten. (Bild aus „Carbon Nano-tube Fabrication: Patent Analysis, Pao-Long Chang et al.“, in Nanotechnology and Nanomaterials, TechOpen 2011)

Sieht man im Staunen und Wundern den Anfang der Philosophie, etwa wie hier Platon:

Das Staunen ist die Einstellung eines Mannes, der die Weisheit wahrhaft liebt, ja es gibt keinen anderen Anfang der Philosophie als diesen,

so stehen solche paradoxe Aussagen, aber auch komplexe „Denksportaufgaben“ mit unerwarteten Ergebnissen, in der Tat am Anfang der Philosophie. Auch das obige Schildkrötenrennen ist auch heute noch tieforschend, obwohl auflösbar:

- trivial: Einfach durch die Erfahrung der Bewegung des eigenen Körpers im Raum wird das Paradoxon ad absurdum geführt, denn es gibt ja offensichtlich Bewegung,
- mathematisch: durch die Einführung des Grenzwertbegriffs,
- physikalisch: im Kontinuum des Raumes durch die gewohnte Mathematik („Infinitesimalrechnung“),
oder bei der Annahme eines diskreten Raumes, der sich nicht beliebig weiterteilen lässt, und der Existenz einer endlichen Länge (und endlichen Zeit) durch diskrete Mathematik,