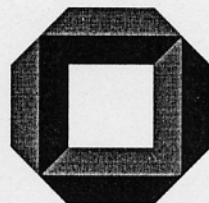


Informatik als
Handwerk, Technik, Wissenschaft

Roland Vollmar

Interner Bericht 2003-10



Universität Karlsruhe

Fakultät für Informatik

ISSN 1432 - 7864

Vorbemerkungen

1) Ein Teil der Ausführungen ist aus den Artikeln [V102] und [V202] mitunter wörtlich übernommen und stützt sich auf [G97].

2) Der Stock von Erkenntnissen und Methoden in der Informatik läßt sich (noch) nicht mit dem von klassischen, sprich älteren Wissenschaften, wie z.B. Mathematik oder Physik, vergleichen. Sein rapides Wachstum in den vergangenen vierzig Jahren zeigt die Informatik – von teilweise kuriosen Moden abgesehen – auf dem richtigen Weg.¹ Wenn auch im folgenden gelegentlich auf die Informatikgeschichte und auf einzelne Einflüsse zu ihrer Entstehung hingewiesen wird, geschieht dies nur der illustrierenden Beispiele wegen, insgesamt wird aber auf eine „Momentaufnahme“ Bezug genommen. Es wird weder unterschieden zwischen der Informatik zu den verschiedenen durch die Hardware bestimmten Computer“generationen“ noch zu den gewandelten Haupteinsatzbereichen von Computern, wie sie z.B. Ceruzzi [C99] skizziert² - schlicht deshalb, weil dabei keine prinzipiellen Veränderungen wahrzunehmen sind. Wahrscheinlich treten sie noch nicht einmal bei einer Verfügbarkeit von Quantencomputern hinreichender Größe in einschneidender Form auf, abgesehen von der anderen Art der Rechnerarchitektur und der Notwendigkeit der Entwicklung geeigneter Algorithmen und damit zusammenhängender Probleme.

Erfindungen und Innovationen setzen sich nur dann durch, wenn ihre Zeit gekommen ist, d.h. wenn die entsprechenden Bedürfnisse auftreten. Ebenso wenig wie auf diesen Aspekt wird hier auf die sozialen

¹ Solche Überzeugungen werden schon viel früher geäußert. Im Vorwort der 2. Jahrestagung der Gesellschaft für Informatik [GI72] heißt es: „Über die Bedeutung, welche der Informatik in Wissenschaft, Industrie heute zukommt, braucht an dieser Stelle nichts mehr gesagt werden.“

² „I have identified several major turning points [...]. They include the computer's transformation in the late 1940s from a specialized instrument for science to a commercial product, the emergence of small systems in the late 1960s, the advent of personal computing in the 1970s, and the spread of networking after 1985. [...]"

Auswirkungen und die durch den Computereinsatz bedingten gesellschaftlichen Änderungen eingegangen.

Zusammenfassung

Der Hauptteil dieses Artikels ist der Herausarbeitung der Charakteristika der Informatik gewidmet. Daran wird gezeigt, daß sie nicht nur eine eigenständige Wissenschaft ist, sondern durch handwerkliche und technische Vorgehensweisen geprägt wird. Dies soll nicht dazu dienen, sie von anderen Fächern abzugrenzen, sondern zur Verbesserung des Verständnisses für sie und um Richtungen ihrer wünschbaren weiteren Entwicklung aufzuzeigen.

Prolog: Konrad Zuse als Künstler, Handwerker, Wissenschaftler und Techniker

Konrad Zuses künstlerische Seite soll einmal *nicht* an seinem Wirken als Maler expressionistischer Abstraktionen, seiner wohlbekannten Porträts oder der weniger verbreiteten Karikaturen demonstriert werden, sondern an vielen seiner ästhetisch herausragenden mechanischen Entwürfe, von den Schaltbausteinen und dem Speicher der Z1 bis zu seinem letzten Werk, dem Helixturm.

Zugleich mögen diese Beispiele, deren Zahl leicht zu vergrößern wäre, hinreichen, um Konrad Zuse zu Recht auch als Handwerker zu bezeichnen. Sind doch - anders als in der Produktion - im Handwerk eigene Entscheidungen und eigenständige Innovationen unabdingbar, nach dem Selbstverständnis der Handwerker spielen für sie "kreative und künstlerische Fähigkeiten eine wichtige Rolle" [An()]. Auch wenn Konrad Zuse keine Rechner entworfen und gebaut hätte, wenn er keinen Parallelrechner konzipiert hätte und wenn er sich nicht innovativ mit Prozeßsteuerung beschäftigt hätte, würde er dennoch in der Geschichte der Wissenschaft einen besonderen Platz einnehmen, allein durch seinen "Plankalkül"; stellt dieser doch die erste höhere Programmiersprache dar.

Konrad Zuse war auch Techniker, was fast als Selbstverständlichkeit angesehen werden könnte. Zieht man z.B. die Definition von "Technik" aus der Encyclopaedia Britannica [EB99], nämlich „the application of scientific knowledge to the practical aims of human life“, zu Rate, dann ist eine derartige Zuordnung evident, sogar dann wenn sein Bauingenieurstudium und seine Beschäftigung als Statiker (und auch seine Selbsteinschätzung) außer Betracht bleiben.

Hätte man Konrad Zuses Werk gekannt und in seiner Bedeutung frühzeitig erkannt, wäre die "Informatik" - nicht eingeschränkt auf Computer Science - wohl 15 Jahre früher als geschehen in Deutschland initiiert worden.

War Konrad Zuse also geradezu ein "Muster" eines Informatikers - auch wenn er von sich später einmal sagte, daß er "mehr oder weniger erfolgreiche Versuche [anstelle], die Informatiker zu verstehen" [Z84]?

Das stark visuell entwickelte Vorstellungsvermögen Konrad Zuses prägte sowohl die Struktur des Z1-Speichers als auch die Darstellung des Plankalküls. Angesichts des schieren Arbeitsaufwandes zum Bau der Z1 hätte er wohl resigniert, wäre er das Problem nicht als Techniker angegangen und hätte er nicht über ein besonderes handwerkliches Geschick verfügt. Als nur *ein* Beispiel seiner außergewöhnlichen wissenschaftlichen Voraussicht sei sein „nichtnumerisches Programmieren“, z.B. von Schachproblemen, angeführt.

Das Zusammenwirken handwerklicher, technischer und wissenschaftlicher Vorgehensweisen, natürlich in unterschiedlichem Grade, sind konstituierend für die Informatik ^{1 2}. Diese Auffassung soll nicht dahingehend ausgelegt werden, als sei damit für die Informatik ein „Alleinstellungsmerkmal“ gegeben, aber sie kann so umfassender verstanden werden³.

¹ Für das Uhrmacherhandwerk wird in [J77] auch auf die Bedeutung einer nicht einseitigen Sicht hingewiesen: „Man tut gut daran, zu erkennen, daß Zeitmessung sowohl eine Kunst als auch eine Wissenschaft ist, und für ein richtiges Verständnis ist es wichtig, sie als beides gebührend zu würdigen.“

² Daß Handwerk und Technik als antagonistisch gesehen werden können, läßt sich aus dem folgenden Zitat ablesen ([F94]): „... war die ursprüngliche Konzeption und die künstlerische Richtung des Bauhauses weitgehend durch die Tradition William Morris' bestimmt, die eine Rückkehr zum Handwerk als Reaktion gegen die moderne Technologie fordert.“

³ Dies ist ähnlich, wie (nach [EB99]) für das Verständnis der Wissenschaft in der Renaissance gilt: „ It is impossible to speak knowledgeably about Renaissance **science** without first understanding the Renaissance concept of **art**.“

Charakteristika der Informatik

Um "Informatik" zu erklären und abzugrenzen, scheint es auf den ersten Blick ausreichend, auf die Literatur zu verweisen. Es gibt nämlich nicht nur mindestens ein Buch mit dem Titel "Was ist Informatik ?" [R91], sondern sogar ein nicht sonderlich altes mit dem Titel "Das ist Informatik" [De01]. Allerdings sei nicht verschwiegen, daß in einem darin erhaltenen Beitrag von W. Coy festgestellt wird: "Eine stabile Definition der Informatik ist noch nicht gefunden."

Daß in der (sich artikulierenden) Öffentlichkeit eine gewisse Unsicherheit hinsichtlich der Informatik herrscht, zeigen bereits deutlich die doch mitunter sehr differierenden Zahlen der künftig benötigten Informatiker oder kommt z.B. zum Ausdruck in der noch nicht allzu alten Klage von Reuter [R01]: "Geblichen ist das Mißverständnis unter Studienanfängern der Informatik, die glauben, ein bißchen Programmiererfahrung auf dem heimischen PC qualifiziere sie in besonderem Maße für das Studieren."

Woran liegt dies - bei einem Fach, das doch schon etwa 30 Jahre als eigenständiger Studiengang an den deutschen Hochschulen etabliert ist?

Ein Grund ist wohl darin zu sehen, daß es immer noch nur eine relativ geringe Zahl von professionell ausgebildeten Informatikerinnen und Informatikern gibt, so daß nur wenige Leute einen Informatik-Profi von Angesicht zu Angesicht kennenlernten. Und zum anderen hängen die Schwierigkeiten mit einem klaren Berufsbild - fast paradoxerweise - mit dem Erfolg der Informatik zusammen: die Informatik spielt eine so bedeutende Rolle in so vielen Anwendungsgebieten, daß die Absolventinnen und Absolventen über einen breiten Bereich von Firmen, Verwaltungen und Forschungsinstitute verteilt sind - eine Tatsache, die es ihnen auch in wirtschaftlich weniger guten Zeiten erleichterte, einen Arbeitsplatz zu finden. Zum anderen verhinderte die

geringe Größe "reiner Informatik-Unternehmen" in Deutschland die Herausarbeitung eines klaren, einfachen Bildes.

Der obige Hinweis auf die Anwendungsgebiete könnte fälschlicherweise den Eindruck aufkommen lassen, daß dieses Durchdringen lediglich die elektronische Datenverarbeitung betreffe. Lassen Sie mich deshalb einige eher handgreifliche Erfolge der Informatik aufzählen und durch das Skizzieren einiger mir wichtig erscheinenden Gebiete "mein Bild" der Informatik entwerfen:

An erster Stelle seien die Computer genannt. Die Steigerung ihrer Geschwindigkeit und die Erhöhung ihrer Speicherkapazität bei gleichzeitiger enormer Preisreduktion bedeuten einen ungeheuren Erfolg. Dadurch wurde es u.a. möglich, erschwingliche Autos mit ABS und Airbags auszustatten und z.B. hochwertige, bequeme Hörgeräte zu bauen. Aber auch die Programmierung oder besser gesagt, die Software-Erstellung braucht sich nicht zu verstecken: Programme, die global arbeitende Buchungssysteme ermöglichen, die (einigermaßen) zuverlässige Kommunikation über das Internet gewährleisten und z.B. solche, die den weitgehend ortsunabhängigen Einsatz von Mobiltelefonen erlauben, sind hierbei zu nennen. Die Verwendung von Programmen zum Entwurf von Häusern und Maschinen, einschl. Computern selbst, zur Auffindung von Lagerstätten, zur Wettervorhersage und Klimamodellierung, in der Astronomie, zum Einsparen teurer Experimente, z.B. in Crashtests, zur „Konstruktion“ von Molekülen wird bereits als selbstverständlich betrachtet. Daß all dies nicht der Informatik allein gutzuschreiben ist, versteht sich von selbst.

All diesen Entwicklungen liegen aber Haltungen und Ergebnisse aus den folgenden Bereichen zugrunde, die mit zur neuen Methodologie der Informatik beitragen:

1) *Algorithmisierung*: Das Auffinden und Erfinden von Algorithmen ist nicht nur die Grundlage zur Lösung von Problemen, sondern verhilft

zum tieferen Verständnis der verschiedensten Phänomene, Probleme und Prozesse.¹

2) *Formalisierung*: Formale Beschreibungen und Symbolmanipulationssysteme haben insbesondere für die Spezifikation realer Systeme und ihre Verhaltensweisen eine besondere Bedeutung gewonnen.

3) *Virtualisierung*: Die Möglichkeit, auf einem beliebigen (Universal-) Rechner die Arbeitsweise eines beliebigen anderen nachvollziehen (simulieren) zu können – und dies, falls gewünscht, über mehrere Stufen – („virtuelle Maschinen“) und Modelle (realer oder irrealer) Systeme im Rechner darzustellen und agieren zu lassen („virtuelle Realität“) erleichtert und beschleunigt Entwicklungsarbeiten resp. ermöglicht ein (ungefährliches) Trainieren in (potentiell) risikoreicher Umgebung.²

4) *Komplexitätsuntersuchungen*: Die Einsicht, daß formale Objekte und Prozesse eine inhärente Komplexität besitzen, zu deren Messung unterschiedliche Ressourcenmaße herangezogen werden können, trägt wesentlich zu ihrem Verständnis bei und hat weitreichende Auswirkungen bez. ihrer Anwendbarkeit.

5) *Untersuchung komplexer Systeme*: Während lange Zeit in den Naturwissenschaften Fortschritte dadurch erzielt wurden, daß Grundphänomene isoliert untersucht wurden, erlauben Informatikmethoden das Herangehen an interagierende, stark miteinander verkoppelte Systeme.

¹ Die herausragende Bedeutung der Algorithmisierung wird z.B. in [G98] betont: „Another possible name for the field, which is not intended to cover the full scope of CS but, rather, its heart and basis, is ‚algorithmics‘.“

Es soll aber nicht verschwiegen werden, daß zumindest in manchen Gebieten, in denen die Informatik Eingang gefunden hat, „geschichtete Verfahren“ besser als rein algorithmische geeignet scheinen. Wie z.B. in der Medizin in manchen Fällen sich eine Abfolge: Anamnese, Röntgen- und CT-Untersuchungen, Ausprobieren verschiedener Medikamente, ...als nützlich erweist, wird in der Bearbeitung komplexer realer Probleme ein interaktives Vorgehen einem rein algorithmischen adäquater sein.

² Als Beispiele seien für den ersten Fall das Ausführen von Mehrprozessoralgorithmen auf einem Einprozessorsystem und für den zweiten Flug- und Fahrzeugsimulatoren erwähnt.

Wesentlich zum Fortschritt beigetragen haben Methoden der *Simulation* und *Visualisierung* ¹.

Die Informatik selbst hat es ja auch ganz wesentlich mit höchst komplexen, nämlich chaotischen Systemen zu tun. Wie Hartmanis [H95] sagt, ist die die Berechnungen ausführende Hardware universell und chaotisch: Die kleinsten Veränderungen der Befehle oder der Daten können beliebig große Differenzen der Resultate nach sich ziehen ².

Antoniou et al. [A00] betonen, daß der Begriff des komplexen Systems nicht neu ist, aber erst mit Informatik-Methoden "in Angriff genommen" werden konnte: "Some of the central characteristics of complexity, including instability and chaos, were known to both Maxwell and Poincaré. But they lacked the mathematical tools to make any progress with these topics. Only with advances in IT was it possible to develop the new mathematics that the science of complexity requires."

Abstrahierungen von realen Systemen erlauben eine Modellbildung, die sich in Programme übersetzen läßt und durch deren (wiederholte und parametrisierte) Ausführung Rückschlüsse auf die realen Systeme gezogen und diesen innewohnende Gesetze aufgedeckt werden können. (Noch) nicht existierende Systeme können untersucht werden, wobei sogar in sog. virtuellen Welten die physikalischen und/oder biologischen Gesetzmäßigkeiten modifiziert werden können. Beschleunigung oder Verlangsamung von Vorgängen ist ebenso möglich wie z.B. deren Umkehrung.

Ergebnisse solcher Simulationen sind, wie auch die von umfangreichen Berechnungen, z.B. der Wettervorhersage, oder der statistischen Analyse sehr großer Datenmengen (z.B. beim sog. data

¹ Dabei ist es schon manchmal nicht mehr möglich die in Galisons Buchtitel „Image and Logic“ [Ga97] vorgenommene Unterscheidung zu machen.

² Man denke z.B. nur an eine Vorzeichenumkehr oder an eine Endlosschleife anstelle eines definierten Resultates.

mining) nur dann nutzbar, wenn sie in einer dem Menschen entsprechenden Form vorliegen. Dies gewährleisten Werkzeuge zur Visualisierung bzw. im dynamischen Fall, z.B. bei Crashtests, der Animation.

Hiermit ist nur ein kleiner Teil von dem angesprochen, was das informatische Vorgehen zu einer neuen Methodologie werden ließ. In den Natur- und Ingenieurwissenschaften bildet es u.a. nach Hartmanis und Lin [H92] neben theoretischem und experimentellem Vorgehen die dritte Säule der wissenschaftlichen Arbeitsweise. Die Informatik erweitert die durch Theorie und Experiment gebotenen Möglichkeiten beträchtlich, insbesondere in den bisher nicht zugänglichen Bereichen komplexer Systeme, wofür hier die Telekommunikationsnetze als ein Beispiel angeführt seien. Komplexe Vorgänge werden verstehbarer, es können Voraussagen über ihr (künftiges) Verhalten gemacht werden, die auch dazu benutzt werden können, entsprechende physikalische, chemische, biologische und technische Prozesse zu optimieren; in den Worten des Rubbia-Reports:

„Bringing closely together simulation (a simplified abstraction of reality) and modelling (a conceptual artificial representation of the real world) it offers the triple power of explanation, prediction and optimisation.“

Ein Gebiet wird aber nicht allein durch die materiellen Gegebenheiten bestimmt, sondern auch durch die Auffassungen der in ihm Tätigen von seinen Zielen.

Daß den Natur- und Ingenieurwissenschaften zugetraut wird, Vorhersagen zu machen, ja daß dies bei oberflächlicher Betrachtung als ihr Charakteristikum angesehen wird, braucht wohl nicht ausführlich erläutert zu werden. So selbstverständlich ist es uns, daß z.B. die Astronomie Methoden entwickelt hat, eine Sonnenfinsternis vorherzusagen bzw. voraus zu berechnen und daß z.B. eine Brücke nicht erst gebaut werden muß, um festzustellen, ob sie bestimmte Lasten tragen kann.

Hingewiesen sei auf die Unterschiedlichkeit dieser beiden Beispiele: Während bei dem ersten ein natürliches System analysiert und dann modelliert wird, steht beim zweiten die Modellierung eines zu bauenden Systems, das gewisse Eigenschaften aufweisen soll, im Vordergrund. In der Informatik wird man es, ebenso wie in den (übrigen) Ingenieurwissenschaften, hauptsächlich mit Aufgaben letzterer Art zu tun haben. Insofern ist es auch naheliegend, deren Konstruktionsprinzipien, vor allem Modularisierung und Hierarchisierung, zu adaptieren - *ein* Beispiel für die Befruchtung, die die Informatik von dort erfahren kann.

S. Toulmin hat in "Voraussicht und Verstehen" [T81] sehr schön dargelegt, daß wir mit den (richtigen) Voraussagen allein nicht zufrieden sind, sondern daß wir (zusätzlich) Verständnis gewinnen wollen. Ein überzeugendes Beispiel ist das der Pharmakologie. Ist man zunächst vielleicht froh, ein Medikament gegen ein bestimmtes Leiden entdeckt zu haben und empirisch seine Wirksamkeit gezeigt zu haben, wird man doch nicht ruhen, bis zumindest im Grundsatz der Wirkungsmechanismus aufgeklärt ist.

Was bedeutet aber "Verständnis" in der Informatik ? ¹ Im "Etymologischen Wörterbuch" heißt es in diesem Zusammenhang u.a. "Von der geistigen Beherrschung einer Sache geht die Entwicklung zum richtigen Erfassen eines geistigen Zusammenhangs:..."

Für die Informatik sei dies dahingehend paraphrasiert, daß über das Beherrschen von Ergebnissen und Methoden hinaus, was mit Handhabungs- oder Anwendungswissen bezeichnet sei, eine Einsicht in Zusammenhänge und Grenzen anzustreben ist.

¹ In einer eher erzählenden Weise geht Heisenberg [H69] auf das Verstehen in der modernen Physik ein. In diesem Zusammenhang sei noch auf die eher ungewöhnliche Auffassung von W. Pauli (zitiert nach [vM941] hingewiesen: „...daß die Befriedigung und Beglückung, die der Mensch beim Verstehen, d.h. bei Bewusstwerdung einer neuen Erkenntnis, empfindet darauf beruht, daß die Einstellung zu Vorgängen oder Objekten der Außenwelt mit im Unbewussten präexistenten inneren Bildern zur Deckung kommt.“

Erinnert sei einmal an die Situation, in der man einen längeren Beweis vor sich hatte: Wenn man auch noch Schritt für Schritt "versteht", hat man am Ende doch oft das Gefühl, "das Wesentliche" nicht begriffen zu haben, also eine Einsicht in den Gesamtzusammenhang nicht erhalten zu haben. Ähnlich geht es, wenn man zu nah vor dem Werk eines Pointillisten steht: Man sieht Farbpunkte, aber kein Bild. Während sich letzteres leicht durch Zurücktreten erhalten lässt, gibt es zum Verstehen von Beweisen keinen Königsweg.

Über *Einsicht*, ein anderer Begriff für „Verständnis“, zu verfügen, setzt Kenntnisse voraus - und je weniger die den "Zufälligkeiten technischer Entwicklungen" (F.L.Bauer) unterworfen sind, desto haltbarer und länger nutzbar sind sie zweifellos. Deshalb plädiere ich für das Vermitteln theoretischer Ergebnisse, auch weil ich die Auffassung Hadamards teile: "Die formale Strenge dient dazu, die Eroberungen der Intuition zu rechtfertigen." Allerdings sollten die sich nicht auf solche aus der sog. "theoretischen Informatik" - nebenbei gesagt, ein Rudiment einer überholten Strukturierung des Faches - beschränken, sondern Prinzipien und Methoden der Analyse und der Konstruktion komplexer Systeme, seien sie natürlich, technisch oder gedanklich gegeben, ebenso umfassen wie die äußerst wichtige Algorithmik.

Computer und Handwerk

Zur Einstimmung ein Zitat von F.J. Dyson [DF00]: "Die Naturwissenschaft entstand aus der Verschmelzung zweier alter Traditionen, der Tradition des philosophischen Denkens, die im antiken Griechenland begann, und der Tradition des Handwerks, die noch älter ist und ihre Blütezeit im europäischen Mittelalter erlebte. Die Philosophie lieferte der Wissenschaft die Begriffe, das Handwerk stellte die Werkzeuge bereit. Bis Ende des 19. Jahrhunderts entwickelten Wissenschaft und handwerkliche Manufaktur sich auf getrennten Wegen. Sie liehen sich zwar häufig Werkzeuge beieinander aus, doch beide blieben

eigenständig. Erst im 20. Jahrhundert gingen Wissenschaft und handwerkliche Industrie eine unlösbare Verbindung ein. [...]

In jeder menschlichen Kultur schaffen Hand und Kopf gemeinsam den Stil, der eine Zivilisation prägt. In allen Zivilisationen nimmt der Handwerker einen Ehrenplatz ein [...]. Unsere eigene Zivilisation bildet da keine Ausnahme."

Die *Encyclopaedia Britannica* verzichtet auf eine Definition von Handwerk, wohingegen in einem alten Lexikon [BL55] ¹ eine zu finden ist. Der Begriff soll in der üblichen, naiven Weise verstanden werden - auch in seiner Abgrenzung zu "Technik" und "industrielle Produktion". Hervorgehoben sei hier nur, daß - wie bereits erwähnt - nach dem Selbstverständnis der Handwerker, für sie auch "kreative und künstlerische Fähigkeiten eine wichtige Rolle" spielen. Anders als in der Produktion sind im Handwerk eigene Entscheidungen und eigenständige Innovationen unabdingbar.

Bevor die Informatik als Fach etabliert wurde, gab es schon Rechenmaschinen. Charles Babbage, der Mitte des 19. Jahrhunderts in seinem Entwurf der "Analytical Engine" alle wesentlichen Konzepte der späteren vollautomatischen Rechner vorgesehen hatte (- wenn auch nicht das Ablegen der Programme im Speicher), war von einer erstaunlichen handwerklichen und künstlerischen Begabung - wie schon ein Blick auf Zeichnungen von ihm beweist -. Mit Clement als Mechaniker arbeitete er eng zusammen, mußte letztlich aber an der zeitbedingten mangelnden Fähigkeit zur Fertigung hochpräziser Mechanik scheitern (wenn dann evtl. auch immer noch der Antrieb durch eine Dampfmaschine problematisch gewesen wäre). Nebenbei bemerkten die Arbeiten an seinen beiden Maschinen einen

¹ Handwerk, Gegenstände mittels einer durch mehrjährige Ausbildung [...] erlangten Fertigkeit herzustellen, bei d. man sich zwar bestimmter Werkzeuge bedient, aber die Handarbeit die Hauptrolle spielt und die Maschine in den Hintergrund tritt.

außerordentlich positiven Effekt bez. der Verbesserung der englischen Feinmechanik.

Ein Sprung um fast 100 Jahre führt zum ersten von dem großen deutschen Computerpionier Konrad Zuse gebauten Automaten, der Z1, den er zwischen 1936 und 1938 zusammen mit Freunden realisierte. Erstaunlicherweise war es ebenfalls noch ein mechanisches Gebilde. Durch die geniale Idee, eine binär arbeitende Maschine zu bauen, konnte er auf die mechanisch sehr aufwendig zu fertigenden Zahnräder (von Babbage), Staffelwalzen oder Sprossenräder verzichten und mit gelochten Blechen auskommen. Diese wurden in unterschiedlicher Version sowohl für den zuerst (1936) fertiggestellten Speicher als auch für Steuer- und Rechenwerk verwandt. Die Konzeption der Maschine erwies sich als korrekt, sie arbeitete aber wegen leicht vorkommender Verkantungen unzuverlässig.

Wie Zemanek bemerkt, war "[d]as Wichtigste für den Computer-Pionier [...] der Wille, etwas Brauchbares zu entwickeln [...]. Das Ziel war erreicht, wenn der Computer rechnete; meist kam dann sofort das nächste Modell an die Reihe." [Z01]

Bei der Entwicklung der Z1 ist das Handwerkliche nicht nur beim exakten Herstellen Tausender von gelochten Blechen zum Ausdruck gekommen, sondern viel beeindruckender in der Konzeption, die in der besten Tradition der Handwerkskunst stand, wie sie z.B. im 18. Jahrhundert durch John Harrison, den Konstrukteur mehrerer Uhren, die das Längengradproblem lösen halfen, exemplifiziert werden kann. Damit war aber die handwerkliche Entwicklung noch nicht am Ende. Statt jetzt auf die weiteren Zuseschen Maschinen Z3 und Z4 einzugehen, sei nochmals F.J. Dyson zitiert: "Der Bau von Computern begann als handwerkliche Tätigkeit; John von Neumann leitete eine wilde Truppe von Mathematikern und Ingenieuren am Institute for Advanced Study in Princeton, und ähnliche Gruppen bauten anderswo ähnliche Maschinen. In seinem Buch *The Soul of a New Machine*

beschreibt Tracy Kidder den Geist einer solchen handwerklichen Industrie, die bis in unsere Zeit überlebt hat. Aber die von Kidder beschriebene Maschine und das Ingenieurteam, das sie entwickelte, haben nicht überlebt. Die Maschine war ein Fehlschlag; sie konnte mit den von der Sun Corporation entwickelten Workstations nicht konkurrieren, und das Team, das sie baute, löste sich auf. Einige Jahre später gestanden Seymour Cray und Danny Hillis ihr Scheitern ein; sie waren die letzten jenes Schlags unabhängiger Unternehmer, die Computer im handwerklichen Stil produzierten. Sie konnten sich auf dem Markt nicht gegen die Großunternehmen behaupten. In der Computerbranche geht die Zeit der handwerklichen Produktion dem Ende entgegen."

Diese abschließende Feststellung bezieht sich - auch bei Dyson - nur auf den Bau von Computern.

Daß mit dem Begriff „Handwerk“ vorsichtig umgegangen werden muß, belege das folgende Zitat: „Stecken, Stanzen und Schalten sind also die relativ handwerklichen Interaktionsstile dieser hardwarezentrierten Mensch-Computer-Schnittstelle, bei der sich der Mensch weitestgehend nach der Arbeitsweise seiner Maschine richten muß.“ [W02] ¹. Hier ist daran erinnert, daß in der Frühzeit Programme durch Kabelverbindungen oder durch Stellungen von (Kipp-)Schaltern repräsentiert waren und Daten über Lochstreifen oder –karten eingegeben wurden. Den Begriff „handwerklich“ in diesem Zusammenhang zu benutzen, deckt sich nicht mit der hier vertretenen Auffassung von „Handwerk“; die genannten Tätigkeiten entsprechen eher der typischen Vorgehensweise einer Maschinen*bedienung* und nicht der (kreativen) des handwerklichen Schaffens (Die Erstellung der diesen Vorgängen

¹ Auf eine noch frühere Form des „Programmierens“ kommt Zemanek [Ze01] im Zusammenhang mit Musikautomaten zu sprechen – und stellt dabei auch noch eine (wohl nicht ganz ernst gemeinte) Verbindung zum Handwerk her: „ ... bietet sich als Programmspeicher die Stifftrommel fast von selbst an [...]. Daß das Programmieren mit dem Schmiedehammer begann, sollte dem Elektroniker eine Warnung sein – das Bit ist etwas Klobiges, auch wenn man es noch so kurz und klein macht.“

zugrundeliegenden Pläne, sprich das Programmieren, muß dabei selbstverständlich anders gewertet werden.)

Andererseits enthält natürlich jedes Gebiet „handwerkliche“ Elemente, wobei dieser Begriff aber in einer eher übertragenen Bedeutung benutzt wird. Nach [W831] sagt Heisenberg: „Physik ist ein ehrliches Handwerk; erst wenn du das gelernt hast, darfst du darüber philosophieren.“ Ähnlich ist es mit dem „Technischen“ bestellt: Am a.O. sagt von Weizsäcker: „In allem Technischen der theoretischen Physik war er mir hoffnungslos überlegen; fing er an zu rechnen, so hörte ich auf und wartete das Resultat ab.“

Dem „Rechnen“ in der Physik entspricht in der Informatik am ehesten das Programmieren, was sich auch im Reden vom „Programmierhandwerk“ ausdrückt.

Informatik als Wissenschaft

Wenn im folgenden überproportional oft auf Informatik als "Wissenschaft" eingegangen wird, hat dies damit zu tun, daß dieser Aspekt derzeit am ausgeprägtesten ist. Über die Berechtigung "Informatik" als solche zu sehen, sind schon zu Beginn genügend Belege angeführt worden. Generell ist eine Wissenschaft weniger durch ihren Bestand an Methoden und Resultaten gekennzeichnet, als durch ihre Ansätze zur Problemformulierung und -lösung ¹.

Hier geht es für die Informatik mehr um Alleinstellungsmerkmale und ihre Einordnung in das Gefüge der "klassischen Natur- und Ingenieurwissenschaften".

¹ Nach [EB99] ist Wissenschaft „any system of knowledge that is concerned with the physical world and its phenomena and that entails unbiased observations and systematic experimentation. In general, a science involves a pursuit of knowledge covering general truths or the operations of fundamental laws“.

Genügt ein - wenn auch komplexes - technisches Gebilde zur Begründung einer Wissenschaft?

Als Beispiel für ein solches seien Autos betrachtet: Sie sind geradezu allgegenwärtig, sie beeinflussen entscheidend die Wirtschaft, sie ersparen dem Menschen Zeit, sie sind (einigermaßen) zuverlässig und sie sind, bezogen auf ihre Leistung, billig.

Es gibt aber keine Wissenschaft, die sich ausschließlich mit Autos beschäftigt, sondern man widmet sich ihnen innerhalb des Maschinenbaus.

Worin liegt der Unterschied zum Computer?

In dem einführenden Lehrbuch von Goldschlager/Lister ist außer drei kennzeichnenden Eigenschaften, nämlich

- Geschwindigkeit
- Zuverlässigkeit
- Kosten,

die auch für Autos gelten, als viertes Merkmal

- die Fähigkeit zur Speicherung großer Informationsmengen

aufgeführt. Dabei nicht explizit erwähnt, ist

- die Fähigkeit von Computern, (codierte) Algorithmen auszuführen (wohl weil im Buch dieser Aspekt den größten Raum einnimmt).

(Im Hinblick auf den Informatikunterricht in der Schule stellen Hartmann und Nievergelt [H02] einen ähnlichen Vergleich an.¹⁾)

Besonders eindrucksvoll sind die Unterschiede noch nicht; entscheidend sind die Auswirkungen des Computereinsatzes.

Dazu sei grob skizziert, warum Computer in der Industrie, der Wirtschaft und der Wissenschaft eine solche Verbreitung gefunden haben (die allerdings ungleich weniger durchgreifend gewesen wäre, wenn nicht zuerst durch die Transistorisierung und dann durch die fortschreitende Miniaturisierung aus den geradezu monströsen Rechnern, bezogen auf Größe und Preis, fingernagelgroße und wenige hundert Euro teure Prozessoren geworden wären (und dies ist u.a. das Verdienst von Materialwissenschaften und Physik)).

Änderungen in der Wissenschaft

Der Ausdruck "Universalrechenautomat" liefert zugleich einen Hinweis auf seine Einsatzmöglichkeiten: Relativ ausführlich wird darauf eingegangen, welche Änderungen durch die Rechner in der Wissenschaft bewirkt wurden, weil sie exemplarisch auch für die anderen Bereiche sind.

In der Wissenschaft, vor allem an den Universitäten, wurde der Rechner zunächst seiner Bezeichnung entsprechend eingesetzt. So wird z.B. in Furger und Heintz über die Situation an der ETH Zürich, die seit 1950 Erfahrungen mit der Zuseschen Z4 gesammelt hatte, gesagt:

¹ "Es wird argumentiert, Autofahren lerne man auch nicht in der Schule, genausowenig müsse der Computer im Unterricht thematisiert werden. Dabei unterscheiden sich die Computernutzung und das Autofahren grundlegend. Das Auto hat nur einen Zweck, die Fortbewegung von Punkt A zu Punkt B. Der Computer ist ein Werkzeug mit einem sehr breiten Anwendungsspektrum.

Beim Autofahren können nur wenige Parameter beeinflusst werden, hauptsächlich Gas, Bremse und Steuerrad. Im Gegensatz dazu ist das 'Armaturenbrett' einer modernen Textverarbeitung wie Word ein Vielfaches komplexer. Wer einen Autotyp fahren kann, vermag recht einfach auf einen anderen Autotyp umzusteigen. Der Umstieg vom Release x.y auf Release x.y+1 einer Textverarbeitung bedingt meistens einige Stunden Einarbeitung."

"Der Bau der ERMETH hatte eine primär instrumentelle Funktion: Es ging darum, das 'programmgesteuerte Rechnen' und mit ihm die angewandte Mathematik zu fördern, und dazu brauchte man einen Computer."

Aiken ist vor allem am Einsatz bei der Lösung physikalischer Probleme interessiert.

John von Neumann sieht 1954, als er sich schon etwa zehn Jahre mit Rechnern und ihren Anwendungen befaßt hatte, einen breiteren Einsatzbereich als dies die Bezeichnung „Computer“ zum Ausdruck bringt: "[...] der Zweck einer Rechenmaschine [ist] lediglich, eine menschliche Tätigkeit, die man selbstverständlich auch ohne maschinelle Hilfe durchführen könnte, nämlich das Lösen von mathematischen Problemen durch Rechnen, zu beschleunigen [...]. Es handelt sich dabei größtenteils um das Auflösen von Problemen, die entweder aus der reinen Mathematik oder aus der angewandten Mathematik oder aus den angrenzenden Wissensgebieten - Physik, Chemie usw. - herrühren, wobei das Problem unter Umständen gar nicht ein Rechenproblem ist." [vN63]

Er deutet dabei auf eine höchst bedeutsame Nutzung des Rechners hin, nämlich *Modelle* zu bearbeiten, sowohl Modelle physischer Strukturen oder Prozesse, wirtschaftlicher Abläufe oder abstrakt gegebener Systeme. V. Cherniavsky identifizierte gar die Informatik mit "interpretierbarer Modellierung" (und betonte die Sichtweise „Virtualität“ (s. z.B. [S01])).

Ein anderer, für die Durchsetzung der Rechner entscheidender Punkt ist der des "Hoffähigmachens" numerischer Lösungen - im Gegensatz zu "geschlossenen Lösungen" der (vor-maschinellen) Mathematik. Auch dieser Aspekt wird schon von John von Neumann deutlich gesehen.

Aber es werden nicht "nur" numerische Lösungen akzeptiert in Fällen, in denen analytische nicht erreichbar sind, sondern auch Probleme wurden mit exakten Methoden behandelbar, z.B. im Zusammenhang mit Matrizen, an deren Lösung in der vor-maschinellen Zeit ihrer schiereren Größe wegen nicht zu denken war - natürlich trug auch das rapide steigende Speichervolumen dazu bei -. So sind z.B. Gleichungssysteme (sogar nichtlineare) mit mehreren Millionen von Unbekannten, wie sie z.B. im Maschinenbau bei Finite-Element-Methoden auftreten, lösbar geworden (Oosterlee et al. [O97]).

Und gerade auch die Möglichkeit, nichtlineare Probleme zu lösen, kann nicht hoch genug eingeschätzt werden.

Die Verfügbarkeit von Rechnern führte zu grundlegenden Änderungen in der Methodik der Angewandten Mathematik und hatte z.B. den Einsatz von Monte-Carlo-Methoden und von probabilistischen Algorithmen zur Folge.

F. J. Dyson nennt noch ein Beispiel: "Die wichtigsten unter den neuen Instrumenten, die Wolszczans Entdeckung ermöglichten, waren Computerprogramme. Der technische Fortschritt im Bereich der Astronomie hat heute mehr mit Software als mit Hardware zu tun."¹

Generell läßt sich für die Natur- und Ingenieurwissenschaften feststellen, daß sie durch die Nutzung von Computern in Geräten, zur Steuerung von Experimenten, zur Auswertung experimenteller Daten und nicht zuletzt durch Simulationen und Berechnungen von Modellen nicht nur eine Änderung erfahren, sondern entscheidende Schritte voran zu machen in der Lage waren. Ähnliches gilt auch für andere Wissen-

¹ Generell wird dies in [M719] unter dem Begriff „Technik“ angesprochen: „... Die verbreitetste Einschätzung des Verhältnisses von T. und Wissenschaft (v.a. Naturwissenschaft) ist naturalist. ausgerichtet. Sie nimmt an, der Mensch mache sich mit Hilfe der T. die Natur bzw. ihre Gesetze nutzbar, indem er wissenschaftl., d.h. theoret. Wissen anwendet. Diese Auffassung verkennt, daß die modernen Erfahrungswissenschaften in ihren Forschungsmöglichkeiten vom augenblickl. Stand der Beobachtungs-, Experimentier- und Meß-T. sowie der Informations- und anderer T.en abhängen. Selbst Naturforschung im engsten Sinn (z.B. in der Astronomie und in der Biologie) hängt in ihren Resultaten immer auch vom Stand der Beobachtungs-T. ab (z.B. vom Auflösungsvermögen der Fernrohre oder Mikroskope). ...“

schaften, z.B. die medizinische Forschung, worauf hier allerdings nicht eingegangen werde.

Änderungen in der Wirtschaft

Die Nutzung von Rechenanlagen in der Wirtschaft ergab sich fast zwangsläufig: Große und mittlere Unternehmen waren an den Einsatz von Lochkartenmaschinen gewöhnt (und auch darauf angewiesen), und so verlangten die größeren Möglichkeiten, die sich durch ihren Anschluß an Rechenautomaten boten, kein entscheidendes Umdenken. Zunächst stand die Automatisierung der Buchhaltung, der Lohnabrechnung und dann des gesamten Bestell- und Rechnungswesens an, wobei *ein* zentraler Rechner das Herz bildete. Alle Daten liefen in Form von Lochkarten und später von Magnetbändern zu ihm und Ergebnislisten wurden manuell weiterverarbeitet. Aber auch in diesem Umfeld erwiesen sich die Möglichkeiten der Modellbildung und der Verwendung von Modellen im Computer als höchst hilfreich bei der Integration der einzelnen Abläufe, so daß insbesondere mit dem Aufkommen von PCs und lokalen Netzen auf dieses Hilfsmittel nicht mehr verzichtet werden kann. (Die Auswirkungen des Internet auf die Wirtschaft, hier auf E-Commerce eingeschränkt, sind noch eher bescheiden, was sich allerdings in wenigen Jahren geändert haben dürfte.) Hier soll nicht weiter auf die Beeinflussung der Wirtschaft durch den Computer eingegangen werden, kennt doch jeder Konsument und (Bank-)Kunde Vor- und Nachteile aus eigenem Erleben.

Neben dieser konsequenten Übernahme von Geschäftsabläufen in Computer, werden sie auch für neuartige Aufgaben eingesetzt. Die Einbeziehung von Methoden des Operations Research und der Prozeßautomatisierung führte zu immer mächtigeren Systemen, wobei auch solche zu nennen sind, die zur Neuentwicklung höchst

komplexer technischer Produkte durch vernetzte, global positionierte Teams beitragen.

Prozeßautomatisierung

Hinter dem Begriff der "Prozeßautomatisierung" steckt ebenfalls eine ungeheuer erfolgreiche Entwicklung, bei der der Rechnereinsatz das integrative Element zwischen Elektrotechnik, Maschinenbau und Informatik darstellt.

Nach mechanischer Regelung z.B. bei Wasserspeichern oder der Dampfmaschine wurden zunächst spezielle elektrische Regler eingesetzt. Rechner, insbesondere in ihrer geringfügig modifizierten Form als Prozeßrechner, erlaubten vor allem durch ihre Geschwindigkeit, ihr Speichervolumen und die Fähigkeit nichtlinear zu regeln, einen wesentlichen Sprung. Sie werden zur Steuerung und Regelung von Produktionsabläufen, in praktisch allen Bereichen eingesetzt, zuerst wohl bei der Papierherstellung und in Walzwerken. (Robotern müßte ein eigener Artikel gewidmet werden.) Mit der Fähigkeit der Computer zur Bildverarbeitung und Mustererkennung erweiterten sich die Nutzungsfelder, wobei vor allem auch im militärischen Bereich naheliegende Anwendungen zu finden sind. Dort machte man von Spezialrechnern, wie z.B. dem *Colossus* in England, frühzeitig Gebrauch beim "Knacken" von verschlüsselten Nachrichten und generell beim Chiffrieren und Dechiffrieren, wie überhaupt die modernen Anwendungen der Kryptographie ohne Rechner nicht denkbar sind.

Weniger spektakulär, aber von größerem Einfluß auf unser tägliches Leben sind die Rechner in den sog. "eingebetteten Systemen". In Haushalts- und Bürogeräten, in der Unterhaltungselektronik, in Geräten zur Unterstützung bei Behinderungen oder auch in Autos sind

eine große Zahl von Rechnern vorhanden, ohne daß sie sichtbar oder sonst bewußt wahrnehmbar würden.

Anwendungen der Informatik wurden im Vergleich zu ihrem Umfang und zu ihrer sowohl wirtschaftlichen als auch das tägliche Leben beeinflussenden Bedeutung recht cursorisch abgehandelt. Dies geschieht unter dem Gesichtspunkt, daß Vollständigkeit angesichts des immer noch kontinuierlichen Eindringens in neue Bereiche ohnehin nicht erreicht werden kann und sie andererseits bekannt sind – wenn auch, wie gerade dargelegt, nicht in vollem Umfang.

Die Einordnung der Informatik in das Wissenschaftsgefüge trägt zu ihrer weiteren Charakterisierung bei, wobei unter den Pionieren keine Einigkeit herrscht – was verständlich ist, kamen sie doch in verschiedenster Weise zu dem Gebiet – aber auch nicht unter den Experten.

Vor dem Zitieren entsprechender auf die Informatik gemünzten Auffassungen sei noch kurz eine grundsätzliche Aussage von Heisenberg (zitiert nach [W832]) angeführt: „Es gibt die Dinge, über die man sich einigen kann, und die Dinge, die uns etwas bedeuten. Von den Dingen, über die man sich einigen kann, handelt die Wissenschaft. Die Dinge, die uns etwas bedeuten, spricht die Kunst aus.“ (An anderer Stelle [W831] fehlt in dem entsprechenden Zitat jeweils „uns“.) Solange man Informatik als „reine Wissenschaft“, d.h. ohne Bezug auf Anwendungen betrachtet, kann man sie der ersten Rubrik zuordnen, da sie, wie noch erwähnt werden wird, als „materialisierte Logik“ aufgefaßt werden kann – und dies ist ein Bereich, über den man diskutieren kann, wie man an den jahrhundertelangen Kontroversen über die materielle Implikation sehen kann. Und versteht man „bedeuten“ als „sinnstiftend wirken“, ändert sich auch bei der Hinzunahme der Anwendungen nichts.

F.L. Bauer sieht die Informatik als Ingenieurwissenschaft, allerdings in besonderer Nähe zur Mathematik [B91]: "Innerhalb der gesamten Geisteswissenschaft ist eben die Mathematik die einzige exakt zu nen-

nende Spielart; sie steht deshalb von den Ingenieurwissenschaften der Informatik am nächsten, der einzigen, die sich mit immateriellem, mit 'physikfreiem' *ingenium* befaßt. Das verbindet Mathematik und Informatik, macht sie zu Geschwistern."

Zemanek sagt dazu [Z92]: "Die Nähe zu Gehirn und Geist macht die Informationstechnik zu einer Brücke zwischen Natur- und Geisteswissenschaften und gibt ihr selbst geisteswissenschaftliche Züge, mehr noch: Geisteswissenschaftliche Natur; sie ist mehr als Technik, so wie sie mehr als Mathematik ist, auch wenn Technik und Mathematik ihr Anfang waren und ihr Bild gestalteten."

Und bereits 1981 schrieb Ganzhorn [G81]: "Mit der ‚Information‘ als Element technischer Einrichtungen hat sich unversehens ein neuer [...] Bereich aufgetan, der typisch interdisziplinär seinen Platz zwischen Mathematik, Logik, Ingenieurwissenschaften und Linguistik finden mag. Nachdem jedoch auch biologische Prozesse wesentlich informatorisch bestimmt sind, wird jeder Versuch einer Abgrenzung zum sachlich schwer vertretbaren Dogma [...]".

Unabhängig davon, welchen Aspekt der Informatik man betonen möchte, ob man sie als Strukturwissenschaft, als eine Werkzeugwissenschaft des Geistes, als Schlüssel- oder als Querschnittsdisziplin sieht (Näheres dazu findet sich in [C97].), vermeiden sollte man den Versuch der Abgrenzung gegenüber anderen Gebieten, sondern bemüht sein, die positiven Seiten der Verwandtschaft zu entwickeln.

Hardware und Software unter handwerklichen und technischen Aspekten

Im folgenden wird der Computer aus verschiedenen Winkeln betrachtet und stellt dabei scheinbar sehr unterschiedliche Geräte dar:

Der Computer ist

- eine symbolverarbeitende Maschine und lässt sich damit z.B. zu Buchhaltungszwecken als Auskunftssystem, zur Programmübersetzung und zu linguistischen Untersuchungen einsetzen
- eine Rechenmaschine, die dank ihrer Geschwindigkeit und ihres Speichervolumens die Möglichkeiten zur Modellierung und damit Beherrschbarkeit komplexer Systeme ungeheuer erweitern half
- eine logische Maschine, die (in noch bescheidenem Umfang) Beweise finden kann und in Interaktion mit dem Menschen diesen Vorgang eindrucksvoll unterstützt ¹
- ein Gerät zur Darstellung des sonst Unsichtbaren, z.B. in der Astronomie, beim Rastertunnelmikroskop oder dem Kernspintomographen
- ein Automat, der versehen mit Sensoren und Aktuatoren, z.B. autonom (als Roboter) handeln kann und mit dem Menschen über Sprachein- und -ausgabe kommuniziert
- ein Medium, insbesondere als Teil „universeller Netze“ ².

Auf die letztgenannte Sicht, die z.B. Coy mit dem Aufsatztitel "Automat - Werkzeug – Medium" [C95] auf eine kurze Form gebracht hat, sei hier nur durch zwei Zitate eingegangen, da sich so der Computer nicht unter "Werkzeug" subsumieren lässt: Er bemerkt dazu u.a.: "Die Idee vollständiger Automation und Algorithmierung scheitert an der umfassenden Durchdringung nahezu aller Arbeitsbereiche, denn die vollständige Formalisierung eines Arbeitsbereichs ist eine eher seltene Gegebenheit. Die Sicht des Computers als "Automat" [...] stößt an ihre Grenzen.

¹ J. von Neumann [vN63] stellt lapidar fest: "Es handelt sich bei der Ziffernmaschine um eine logische Maschine." Und Zuse bemerkt (zitiert nach Kaufmann [K74]): „Die Relaismaschine war gewissermaßen von vornherein materialisierte Logik.“

² Dieser Ausdruck soll (in gewisser Analogie zu „universellen Maschinen“) auf solche Netze wie z.B. das Internet, hinweisen, die vielfältige Funktionen und Inhalte bereitstellen und über LANs oder für spezielle Aufgaben zusammengestellte „Grids“ hinausgehen.

Was sich statt dessen mit der ökonomischen Dezentralisierung des Computers durch die PCs zeigt, ist die Möglichkeit, Computer und Programme als unterstützendes Arbeitsmittel einzusetzen." Und später: "Mit 'Automat' oder 'Werkzeug' werden natürlich keine physischen Eigenschaften des Computers beschrieben, es sind Sichtweisen auf das gleiche technische Artefakt, das aus unterschiedlicher Sicht nicht gleich aussieht."

Daß es zumindest noch eine andere Sichtweise gibt, bringt er folgendermaßen zum Ausdruck: "...Weder das Leitbild 'Automat' noch das Leitbild 'Werkzeug' kann diese kulturellen Wirkungen der Informatik als vernetzte Elemente in einer Medienwelt hinreichend erklären. 'Think of the computer, not as a tool, but as a medium!' fordert Brenda Laurel [...] - aber vielleicht sind diese beiden Sichten gar nicht unvereinbar, sondern unabhängig voneinander."

Eine unterschiedliche Sicht auf den Computer im (schulischen) Unterricht haben Hartmann und Nievergelt [H02]: „Der Computer nimmt in der Ausbildung verschiedene Rollen ein. Zum ersten ist er selber Gegenstand des Unterrichtes, zum zweiten ein Werkzeug im Arbeitsalltag und zum dritten ein Medium zur Über- und Vermittlung von Informationen."

Daß es natürlich durchaus sinnvoll sein kann, sich auf die eine oder die andere Sichtweise zu beschränken, kann man z.B. an Grundlagenuntersuchungen zur Berechnungsuniversalität sehen, bei der die Automatensicht hinreicht oder andererseits an einem Aufsatz von Zemanek mit dem Titel "Werkzeug Computer: Verstärker von Intelligenz und Sorglosigkeit" [Z96].

Der bereits mehrfach gebrauchte Begriff "Universalrechenautomat" wird gerechtfertigt durch vorstehende Charakterisierung – auch wenn die Bezeichnung in einem naiven Sinne verstanden wird.

Die vier erstgenannten Punkte beschreiben den Werkzeugcharakter des Computers, eines universell einsetzbaren Werkzeuges. Er hat, ähnlich wie das Messer, eines der wohl ersten Werkzeuge des Menschen, nicht nur die unterschiedlichsten Einsatzgebiete, sondern (daraus resultierend) auch die unterschiedlichsten Erscheinungsformen, vom Mikroprozessor über PC, "Main frame"-Rechner, Prozeßrechner bis zu Größt- und Parallelrechner. Ebensowenig wie man von einem Menschen erwarten würde, gleich gut mit dem ganzen Spektrum von Messern, vom Miniskalpell bis zum Schneider von Metallplatten, hantieren zu können, ebensowenig werden diese verschiedenartigen Computer einfach benutzbar sein. (Und hier zeigt sich noch ein weiterer Unterschied zum Auto: Der Führerschein bestätigt ja gerade das Fahrenkönnen eines beliebigen Autos (zumindest einer Klasse)).

Die "universelle" Fähigkeit des Computers ist seine Stärke, aber auch in gewisser Weise seine Schwäche: Sie macht es erforderlich, ihn an die gewünschte Aufgabenstellung anzupassen, sprich, ihn zu programmieren. Auch dies ist nichts prinzipiell Neues: "Long before the word *software* came into general use, there was a need to describe longer, complicated sequences of calculation. For this purpose, it was patent to support the writeup of intermediate results on paper by forms indicating the course of calculation. [...]. Konrad Zuse, in 1935, was lead by a blank of this kind used in calculating the superposition of two rectangular moment areas to his first ideas about a formula-controlled computer [...]." [B00]

Für die frühen Computer wurde die Programmerstellung (bis in die sechziger Jahre) noch nicht als problematisch angesehen, einfach deshalb, weil der wissenschaftlich ausgebildete Personenkreis, dem sie zugänglich waren, mit den Algorithmen auch die Programme entwarf bzw. gewisse Standardroutinen von den Rechnerentwicklern kostenlos mitgeliefert erhielt.

In diesem Zusammenhang sei bemerkt, daß die Informatik im Gegensatz zu anderen Ingenieurwissenschaften nicht aus einem oder mehreren Handwerken hervorging. Dies mag auch *ein* Grund dafür gewesen sein, daß in ihr lange Zeit Ausbildungsberufe auf unterschiedlichen Ebenen kaum angeboten bzw. gefragt waren, was ihr sicherlich nicht zum Vorteil gereichte, allein schon deshalb, weil vieles von „Dilettanten“ erledigt werden mußte und weil „handwerklich Begabten“ keine Perspektiven eröffnet wurden. Eine weitere Ursache mag darin gelegen haben, daß der Umgang mit Computern in deren Jugendjahren gleichsam als „Geheimwissenschaft“ betrachtet wurde, deren Vermittlung weder als wünschenswert noch als leicht möglich angesehen wurde.

Die Haltung zur Programmierung änderte sich mit dem oben beschriebenen Einsatz des Computers in immer neuen Gebieten, und mit zunehmender Komplexität der Anwendungen wuchsen die Programme. Das spiegelt sich u.a. in der Wortwahl wider: D. Knuth nannte sein berühmtes Werk „The Art of Computer Programming“ [K76]. Wenn die Bezeichnung wohl auch metaphorisch zu verstehen ist, so wurde es doch bald offensichtlich, daß Programmieren zu einer "Technik" (bzw. einer „Wissenschaft“ [Gr81]) entwickelt werden mußte. Jetzt wird sie mit dem Begriff "Software Engineering" bezeichnet.

Der Handwerk- und Technikcharakter der Informatik tritt vor allem im Softwarebereich zutage¹: Ebenso wie manche „klassischen“ Handwerker, z.B. Schmiede oder Uhrmacher, stellt sie ihre Werkzeuge her und vermittelt gleichfalls das richtige Umgehen mit ihnen. In der Informatik werden nicht nur die Rechner konzipiert und zumindest viele Vorbereitungen zu ihrem "Bau" getroffen, sondern sie vermittelt auch das Wissen um die Möglichkeiten und die Grenzen ihres Einsatzes und sie lehrt, wie das "Universalwerkzeug Computer" durch

¹ Für diese Auffassung kann ich z.B. auch Brooks [Br96] als Kronzeuge anführen, obwohl ich seine Charakterisierung für zu eng halte: „...the computer scientist is a *toolsmith* – no more, but no less. It is an honourable calling.“

Programmierung dem jeweiligen Problem angepaßt werden kann. Zemanek wies 1993 darauf hin, daß in diesem Bereich noch „viel zu tun ist“, was nicht der Software-Industrie allein überlassen werden könne [Z93]¹ – und er stellt dabei auch den Bezug zum Handwerk her. Jeder Nutzer eines Computers wird wohl dieser Aussage auch heute noch zustimmen.

Nun einige Aussagen zur Software-Erstellung in komprimierter Form:

- Die Vielfalt der Einsatzgebiete des Computers verlangt das Erstellen immer neuer Programme.
- Die gleiche Wirkung kann durch (unendlich) viele Programme erzielt werden.
- Für Programme können die unterschiedlichsten Komplexitätsmaße sinnvoll sein, z.B. Zeit- oder Speicherbedarf, Zeilenzahl, Verständlichkeit, Erweiterbarkeit, Wiederverwendbarkeit, ... und Kombinationen davon.

¹Hier möchte ich einen Exkurs in die Gegenwart machen und einen Aspekt der Computertechnik anpeilen, der vielleicht auch heute wieder Universitätsaufgabe sein könnte: Der Aspekt der Programmierwerkzeuge, der Softwarefabrik. Während nämlich die Hardware alle Vorteile der Kombination von Physik, Elektronik und Fertigungstechnik zu äußerst preiswerter Herstellung ausnützt, wird die Software handwerklich gefertigt, aber nicht mit den Vorteilen des Handwerks, vom persönlichen Kontakt auszugehen und die Erfahrung für die Wünsche und Nöte des Benützers einsetzen zu können; und weil Software-Fertigung so mathematisch aussieht, nimmt man sie auch als Tummelplatz von Genies und Quasigenies hin. Man redet zwar von SoftwareEngineering (man darf nicht übersehen, daß dies ein von Mathematikern geprägter und initiiertes Ausdruck ist), aber von einer Produktionsstrecke in Analogie zur Hardware wird noch lange keine Rede sein können. Ohne auf Einzelheiten einzugehen: Werkshallen und erst recht Produktionsstrecken beziehen Effektivität und Verlässlichkeit aus den verwendeten Werkzeugmaschinen. Für die Software gibt es zwar den Ausdruck "tools", aber keine universell befriedigende Werkzeugmaschinen der Programmierung. Alle Versuche, Softwarefabriken in Gang zu nehmen, sind gescheitert [...]. In der Software-Produktion fehlt der Informationstechnik noch ein wesentliches Element: Die beherrschte Beziehung zwischen Werkzeug und Produkt. Es klafft eine Lücke mit teuren Folgen infolge von Unverlässlichkeit, Einzelfertigungspreis, improvisierter Dokumentation und krassen Liefertermin-Überschreitungen.

Vielleicht wäre hier die Rückkehr zu vom kommerziellen Druck freien Universitätsvorgangsweisen der rechte Weg - die akademische Durchuntersuchung des Software-Herstellungsvorgangs auf seine Automatisierbarkeit. [...] Man müßte den Entwurfs- und Herstellungsvorgang bis in die Grundlagen analysieren und langfristig daran arbeiten - was sich die Industrie tatsächlich nicht leisten kann.

Nach F.J. Dyson ist die Software-Industrie eine handwerklich geprägte Industrie: "Die Computer-Software entstand als Hilfsmittel der Wissenschaft und breitete sich später auf alle Bereiche der Industrie und des Handels aus. Überall, wo ernsthaft mit Computern gearbeitet wurde, lernten junge Menschen, Programme zu schreiben und anzuwenden. Trotz Microsoft und anderer Großunternehmen ist die Software-Industrie bis heute in weiten Teilen handwerklich strukturiert.

Wegen der gewaltigen Vielfalt spezieller Anwendungen wird es stets Raum für Einzelne geben, die auf der Grundlage ihres ganz besonderen Wissens Software entwickeln. Es wird stets Marktnischen geben, in denen kleine Software-Hersteller überleben können. Das Handwerk der Software-Entwicklung wird nicht aussterben. Und das Gewerbe der kreativen Anwendung von Software wird sogar noch stärker florieren als deren Entwicklung. Die Vielfalt der Programme und die Fähigkeit, sie einzusetzen, haben ein ganzes Spektrum handwerklich strukturierter Industrien hervorgebracht, vom Desktop-Publishing bis hin zur computergestützten Konstruktion und Produktion."¹

Mit der Ad-hoc-Fertigung von Programmen als Mittel, den Computer in die Lage zu versetzen, einzelne Algorithmen "auszuführen", ist es aber bei wohl allen - zumindest nicht-wissenschaftlichen - Anwendungen nicht getan.

Ohne das Handwerk abwerten zu wollen, bin ich überzeugt, daß sich die „handwerklich strukturierte“ Software-Industrie (-und diese Einschätzung Dysons teile ich hinsichtlich der derzeitigen Situation) in eine „technisch-orientierte“ wandeln muß. Zumindest müssen aber

¹ Daß aus der Wissenschaft Anforderungen an das Handwerk gestellt werden, ist natürlich auch nichts Neues. Exemplarisch stehe dafür das Folgende: „Von den angewandten mathematischen Wissenschaften verlangte besonders die Astronomie, die unter Männern wie Bernhard Walther, Johann Werner und Regiomontan in Nürnberg eine hohe Blüte erreichte, feinste Instrumente, die geschickte Schlosser und Schmiede anfertigten. Ebenso war es mit feinen Meßinstrumenten, die von den Zirkelschmieden hergestellt wurden, und exakten Waagen, die aus der Werkstatt der Waagemacher kamen.“ [S77]

die bewährten Vorgehensweisen des Handwerks auch für diesen Bereich übernommen werden. Ein gutes Beispiel dafür sind die „Entwurfsmuster“ für Software, die in Analogie zu den im Handwerk – für Schlosser, Weber bis zu Turmuhren- und Orgelbauer und in der Bildenden Kunst – erwähnt seien nur das Göttinger Musterbuch und das von de Grassi - bekannten Musterbüchern zu sehen sind (–von den bekanntesten, nämlich solchen für Köche einmal ganz abgesehen).

Eine solche Entwicklung wird allein schon der Masse der zu produzierenden *qualitativ hochwertigen* Software wegen notwendig sein; auch die Analogie z.B. zur Autoproduktion legt dies nahe. Der Bedarf an *solcher* Software scheint mir immer noch größer als daß er mit dem vorhandenen Personal realisiert werden könnte.

Nach [W01] gilt für Software: " - Jedes Software-System [...] kann nur als Teil eines Ganzen betrachtet werden. [...]

- Entwicklung von Software-Systemen findet nie mehr ohne Bezug zu anderen Software-Systemen statt. [...]

- Software-Systeme erreichen keinen stationären Zustand. [...]

Das Konstruieren von Software aus existierenden Komponenten erfordert die Analyse dieser Komponenten auf Paßfähigkeit zu anderen Komponenten sowie die Entwicklung von Schnittstellen zwischen diesen Komponenten. Die Analyse von Software kann sich nicht auf den Abgleich syntaktischer Bedingungen an den Schnittstellen beschränken, sondern muß semantisch bedingte Wechselwirkungen zwischen den Komponenten berücksichtigen.

Diese Liste von Aufgaben ist keineswegs erschöpfend. Sie sollte aber ausreichen, um zu verdeutlichen, daß das Software Engineering sich fundamental von der Programmierung unterscheidet."

Es darf nicht sein, daß jeder, der programmieren kann, etwas „Neues“ zusammen“bastelt“. Und auch der Einsatz von Werkzeugen („case tools“) hat beträchtliche Nachteile: Verhindert er doch, daß Software-Erstellung von der Pike auf gelernt werden muß. Ähnlich wie jeder Maschinenbaustudent früher im Praktikum beim Feilen einen sinnlichen Eindruck vom Material gewinnen mußte, sollte im Informatik-Studium auf eine „programming literacy“ Wert gelegt werden. Der Mangel daran hat ähnliche Konsequenzen, wie die, die in Klagen von Bauingenieuren gehörte: Das Handhabungswissen von Programmsystemen zum Entwurf auch komplexer Bauten sei relativ einfach zu vermitteln; was fehle, sei die Erfahrung, die nur bei längerem „training on the job“ erworben werde und deren Fehlen verhindere, daß (für Geübte offensichtliche) Schwachpunkte erkannt würden.

Andererseits ist der Aufbau von Software-Systemen, die größenordnungsmäßig eine Million Zeilen Code enthalten, nicht in hinreichender Güte ohne verbesserte Werkzeuge und ohne Computerhilfe zu bewerkstelligen. Ich hege allerdings Zweifel daran, daß dabei die Automatisierung so weit reichen kann, wie dies z.B. beim Compilerbau der Fall ist, nicht nur weil die Komplexität solcher Systeme inhärent größer ist, sondern weil ja in den meisten Anwendungen die menschlichen Nutzer mit ihren divergierenden Ansichten und Bedürfnissen, vielleicht auch denen, zumindest Grundstrukturen der Software zu verstehen und deren Grenzen verdeutlicht zu erhalten, einbezogen werden müssen.

Nach Tauber [T01] dienen in industrieller Software "die Hälfte bis Dreiviertel der Programmzeilen nicht dem eigentlichen Anwendungsproblem, sondern der Robustheit und Stabilität, der komfortablen Benutzbarkeit und der sicheren Betreibbarkeit." ¹

Letztere zu erhöhen und zumindest Software für sicherheitskritische Anwendungen zu verifizieren, ist eine Forderung an die Informatik als *Wissenschaft*, wobei nicht unterschlagen sei, daß halb-automatische Beweiser in relativ kleinen Programmen schon erfolgreich eingesetzt werden. In anderen Bereichen des Software-Engineering ist die Informatik von ihrer *technischen Seite* her am Zuge.

Handwerke wirken aufeinander und auf Technik und Wissenschaft ein. So nutzten z.B. Kanonemacher das Know-how der Glockengießer, nach der *Encyclopaedia Britannica* hatten die aus der Entwicklung der Windmühlen gewonnenen Kenntnisse über Mechanik u.a. Einfluß auf die Herstellung von Uhren, und das Schmiedehandwerk förderte das Aufkommen der Metallurgie.

Im Bereich der Software-Erstellung werden am ehesten die Wechselwirkungen von handwerklichem Vorgehen und wissenschaftlichen Erkenntnissen deutlich. Aber auch auf dem Gebiet der Automatentheorie und der Formalen Sprachen läßt sich beobachten, wie Resultate der Forschung in die informatische Technik übernommen werden.

In [H01] ist dazu Folgendes bemerkt: "While applications of automata and language theory to compilers are now so well understood that they are normally covered in a compiler course, there are a variety of more recent uses, including model-checking algorithms to verify

¹ Nach eigener Erfahrung früherer Programmfertigungen würde es mich nicht wundern, wenn auch in kommerzieller Software ein Teil nie angesprochen wird, entweder weil man Erweiterungen geplant, dann aber nicht ausgeführt hat oder weil man z.B. zwei verschiedene Algorithmen zur Lösung eines Problems implementierte und sich nicht definitiv entschließen konnte, einem den Vorzug zu geben. Damit scheint eine ähnliche Situation vorzuliegen wie in der Biologie: „...erweist sich der allergrößte Teil unseres vorgeblichen Erbgutes aus solcher Sicht eher als Erbmüll, in dem zwar vermutlich, wie in den Abfallhaufen der Vorzeit, in denen Archäologen so gerne wühlen, allerhand Information über die biogenetische Vorgeschichte des Menschen steckt“ [M01]

protocols and document-description languages that are patterned on context-free grammars."

Der Versuch einer Trennung der Informatik in technische und wissenschaftliche Bereiche ist nicht erfolgreich vorzunehmen: Charakterisiert man (wohl zu naiv) die Technik als „durch Erfindungen geprägt“ und ordnet man der Wissenschaft „Entdeckungen“ zu, so hat man einerseits die Schwierigkeit, daß die Wirkbereiche dieser beiden Begriffe umstritten sind¹, und daß man zum andern daran zweifeln kann, daß es in der Informatik überhaupt Entdeckungen gibt – und damit ein Widerspruch zu oben Gesagtem aufträte. Das konstituierende Element der Informatik, „der“ Computer, ist sicherlich nicht entdeckt worden. (Folgt man Daumas [D75], dann müßte ergründet werden, ob er eine Erfindung oder eine Innovation darstellt.) Entdeckungen der Informatik wird man am ehesten in der Theorie erwarten. Ist aber z.B. die Feststellung, daß unentscheidbare Probleme existieren, eine Entdeckung? Ihre Existenz hängt ja davon ab, ob man die Adäquatheit der Explikation des Berechenbarkeitsbegriffs (z.B. durch die Turing-Maschine) akzeptiert. Aussagen der Theorie beruhen auf einem „gesetzten“ Axiomensystem, also auf etwas, auf das man sich nach Heisenberg (s. oben) „nur“ einigen kann. In den Naturwissenschaften liegt eine andere Situation vor, aber auch dort muß man vorsichtig sein: Die „Entdeckung“ der meisten Transurane z.B. kommt erst durch ihre „Konstruktion“ zustande, da sie auf der Erde nicht (mehr) vorkommen.

Technik und vor allem Handwerk berücksichtigen bei ihren Hervorbringungen ästhetische Ansprüche. Deshalb wird im folgenden gezeigt, daß dies auch für die Informatik gilt und sie sich auch unter diesem Aspekt entsprechend einordnen läßt. Folgte man Duddeck [Du01], so wäre (auch) die Informatik als „Technikwissenschaft“ eine

¹ S. z.B. [D75]: „Die langen Kontroversen über die Bedeutung der Worte Entdeckung und Erfindung sind bekannt. Vor langer Zeit begonnen, sind sie auch heute noch nicht abgeschlossen.“

Kunst¹ - und damit, zumindest wenn man die klassische im Auge hat, mit Ästhetik zu assoziieren. In der Encyclopaedia Britannica wird ausgeführt, daß oft die Grenze zwischen Wissenschaft und Kunst nicht klar zu ziehen sei². Nach Goethe ist „Kunst: eine andere Natur, auch geheimnisvoll, aber verständlicher; denn sie entspringt aus dem Verstande.“ [Go98], woraus zu folgern ist, daß sich Wissenschaft und Kunst – zumindest nicht von vornherein – auszuschließen brauchen.

Es dürfte klar sein, daß Personen, die in der Informatik tätig sind, *daneben* aber als Künstler (im üblichen Sinne, d.h. als Komponisten, Maler, ...) wirken als Zeugen für „Ästhetik in der Informatik“ nicht herangezogen werden dürfen – wie oben schon für Konrad Zuse vermerkt -. Auch Werke, bei deren *Produktion* Computer eingesetzt wurden, sollen nicht als Beleg dienen. Ebenso wenig darf eine Selbstaussage, wie sie z.B. im Buchtitel „The Art of Computer Programming“ [K76] oder anderen ähnlichen gegeben ist, als Beweis genutzt werden, womit nicht gesagt sei, daß der Bereich der Programmierung außer Acht gelassen werden dürfe.

Nach einem Blick auf die „Definition“ von Kunst in [EB99]³ werde das „Hervorbringen ästhetischer Objekte“ in der Informatik diskutiert. Nicht-funktional bedingte Ästhetik, wie sie z.B. im (damals modernen) Teakholz-Gehäuse der TR4 oder der Rauchglas-Eleganz der CM2 zum Ausdruck kam, bleibe unberücksichtigt. In der Informatik spielen sowohl beim Erstellen von Modellen als auch beim Entwerfen von Algorithmen und Schaltungen wie auch beim Fertigen von

¹ „Technik ist das griechische *Techne*, die Kunst, das Können, die Geschicklichkeit. [...] Bis in die Renaissance hinein waren alle Techniken Künste: Die Kunst des Wasserbaus, des Ackerbaus, des Schmiedens, die Kunst des Brückenbaus, die Bergbaukünste.“

² [...] whether one considers architecture, medicine, chess, or cookery, there is no clear frontier where the realm of **science** ends and that of **art** begins: the creative person is a citizen of both. [EB99]

³ Art, the use of skill and imagination in the creation of aesthetic objects, environments, or experiences that can be shared with others. The term art may also designate one of a number of modes of expression conventionally categorized by the medium utilized or the form of the product; thus we speak of painting, sculpture, filmmaking, music,...

Programmen - wobei diese Aufzählung nicht als erschöpfend anzusehen ist¹- ästhetische Gesichtspunkte eine Rolle.

Natürlich läßt sich über die Ästhetik von Objekten trefflich streiten; in allen Kunstgattungen sind viele Beispiele zu finden, deren Ein- und Wertschätzung sich im Laufe der Zeit geändert haben – und zwar in beide Richtungen – und solche, die sogar zu *einem* Zeitpunkt auch unter Kennern massive Ablehnung und begeisterte Zustimmung fanden.

Der Einfachheit der Betrachtung wegen sei im folgenden „ästhetisch“ und „schön“ synonym gebraucht. Damit ist noch nicht viel gewonnen, da „Schönheit“ philosophisch gesehen nicht leichter handhabbar ist², aber den Vorteil hat, intuitiv einsichtiger zu sein.

Die folgenden Einschätzungen sind natürlich höchst subjektiv, sie sollen aber auch nur beispielhaft zeigen, in welchem Sinne „Schönheit“ verstanden wird.

In der Mathematik hält man häufig kürzere Beweise für schöner als längere (für dieselbe Aussage)^{3 4}. Für Programme kann dies in einer ersten Näherung ebenso gelten, allerdings darf dabei die Kürze nicht

¹ So fiel mir zum ersten Mal eine außerinformatische Wertung in dem Buchtitel „Jewels of Formal Language Theory“ [S81] auf.

² Die Schönheit ist, so lautete die eine der antiken Definitionen, die richtige Übereinstimmung der Teile miteinander und mit dem Ganzen. [H71]

³ Daß dies nicht generell so gesehen wird, kommt in folgendem Zitat zum Ausdruck [B96]: „In der Mathematik scheint es so zu sein, daß Schönheit eng mit Einfachheit zusammenhängt. Manche gehen so weit, diese beiden Begriffe zu identifizieren: Schönheit = Einfachheit.“

Auch hier beobachtete Penrose genauer. Seiner Meinung nach ist in der Mathematik nicht Einfachheit als solche schön, sondern vor allem *unerwartete* Einfachheit.“

⁴ In der Mathematik wird die Bezugnahme auf „Schönheit“ nicht gescheut, wie aus zahlreichen Zitaten im Internet hervorgeht oder in Buchtiteln [H99] oder explizit zum Ausdruck kommt. Als Beispiel für letzteres ein Zitat aus [B96]: „Der berühmte englische Mathematiker G.H. Hardy (1877-1947) vertrat prononciert die Meinung, daß die Schönheit der eigentliche Maßstab für die Mathematik sei. [...] schreibt er: „Ein Mathematiker schafft, ähnlich wie ein Maler oder ein Dichter, Strukturen [...] Die Strukturen, die ein Mathematiker schafft, müssen so wie die der Maler und Dichter schön sein ... Schönheit ist der erste Test: Für häßliche Mathematik gibt es keinen dauerhaften Platz auf der Welt.““

Für die Informatik war die „Ausbeute“ wesentlich karger – und eigentlich nur in einer Werbeaussage war ein entsprechendes (Selbst-)Bewußtsein vorhanden [S02]: „Software von SOLYP funktioniert nicht nur technisch perfekt, sie ist auch schön [...]“

mit Unstrukturiertheit erkaufft sein. Als unschön sind sicherlich im allgem. solche Programme anzusehen, die sich „entwickelt haben“ (oft nur ein euphemistischer Ausdruck für „zusammengeschustert werden“), d.h. ohne Gesamtkonzeption erstellt und nach und nach erweitert und modifiziert wurden.^{1 2} Für sie gilt wohl das o.a. Zitat von Markl [M01] über die Erbsubstanz.

Bei Algorithmen hängt die Schönheit auch von der Formulierung ab, die geprägt werden kann durch die Möglichkeiten der verwendeten Programmiersprache: So sind rekursive Algorithmen eleganter und schöner als ihre iterative Form. Algorithmen zur Fakultätsberechnung und zum Quicksort aus [G95], die in Gofer notiert sind, stellen Beispiele dafür dar.

An „innerinformatischen“ Modellen betrachte ich die Turing-Maschine – als Explikation des Berechenbarkeitsbegriffes – als schöner als z.B. Markow-Algorithmen, obwohl letztere als Symbolmanipulationsvorschriften der Informatik-Vorgehensweise adäquater sind. Meine Vorliebe für die Turing-Maschine wird nicht zuletzt beeinflusst durch die überzeugende anthropomorphe Motivation, die Turing [T36] bot.

Modelle realer Systeme haben es im allgemeinen schwerer, ästhetischen Ansprüchen zu genügen, müssen sie doch, wenn sie nicht ganz grob sind, die üblicherweise komplexe Realität widerspiegeln. Dies gilt sogar für noch relativ abstrakte Modelle wie z.B. die Typen von Chomsky-Grammatiken.

Bildliche Repräsentationen von Computerbausteinen, seien es graphische von Schaltungen oder photographische von Chips offenbaren ihren ästhetischen Reiz auf den ersten Blick: Strukturen

¹ „In fact, the ecology of programming is such that overall, programmers spend over 80 percent of their time modifying code, not writing it“, said Selfridge in 1995.[DG99]

² By almost any standard, the software industry qualifies as a Darwinian process – from the generation of software (you combine existing code segments, then execute and weed out the bugs) to the management of the industry as a whole: eat, be eaten, or merge. [DG99]

wie z.B. Speicherchips haben wegen ihres homogenen, modularen Aufbaus ihre eigene Schönheit, die auch zu tun hat mit ihrer Durchschaubarkeit. Dabei findet Beachtung, was im ersten Absatz des folgenden Zitats [Z01] formuliert ist: „Eine Entwurfstugend, die nicht zur Konsistenz gehört, aber bei ihrer Verwirklichung hilft, ist die Orthogonalität, die Vermeidung unnötiger Kopplungen und Querabhängigkeiten. Wenn man einen Zug oder eine Einzelheit des Entwurfs ändern muß, sollten keine anderen betroffen sein. Eine weitere Tugend ist der Verzicht auf mutwillige Beifügungen, ohne in Geiz zu verfallen.

Über dem Ganzen muß stets aber auch die Effizienz stehen, die Ökonomie und die Umwelt- und Benutzerfreundlichkeit. Es ist ein ganzer Berg einander widersprechender Forderungen, dem man nicht mit einem Algorithmus auf den Leib rücken kann, sondern wo höheres menschliches Empfinden erforderlich ist. Die heute übliche Abwertung der Emotion bringt die Gefahr des Blindwerdens für Abwägungen, die eine Ebene höher liegen als Berechnung und rationale Entscheidung. Entwurf ist eine heikle Mischung aus strengem Denken bis Rechnen und klarem Fühlen all dessen, was jenseits der formalen Erfassung liegt. Mit sehr befriedigendem Erfolg habe ich die Lehren des altrömischen Architekten Vitruvius in den Systementwurf übersetzt: Dabei entwickelt sich ein Feingefühl für die Verstöße gegen gute Architektur, denen man heute auf allen Gebieten begegnet.

Ich wage vorauszusagen, daß auf lange Sicht in der Informatik das Gleiche passieren wird wie im Bauwesen: Das Studium wird in Entwurfslehre und in Herstellungslehre geteilt werden. Die Architektur programmierter Systeme sowohl im Innern des Computers als auch für alle Anwendungen wird zu einer gepflegten Kunst werden müssen.“

Der zweite Teil der Ausführungen Zemaneks verweist auf eine „höhere Schönheit“.

Statt einer Zusammenfassung

"Ein Handwerker kann und will jedoch seine Tätigkeiten nicht wie der Spezialist in der Industrie in Funktionen aufteilen. Er muß vielen in gleicher Weise gerecht werden: den stofflichen wie den technischen, den kaufmännischen wie den organisatorischen." [S77] Dies könnte und sollte - entsprechend modifiziert - auch als (Zusatz-)Tätigkeitsbeschreibung für Informatikerinnen und Informatiker gelten.

Wenn Francis Bacon empfiehlt, daß Wissenschaftler die Methoden des Handwerks studieren sollten und Handwerker mehr von der Wissenschaft lernen sollten, so kann dies dahingehend paraphrasiert werden, daß Informatiker jeweils die handwerklichen, technischen und wissenschaftlichen Seiten ihres Faches im Auge behalten sollten.¹

¹ In [MP01] wird das Zusammenwirken im Titel genannten Attribute der Informatik als ein Kennzeichen der „modernen“ Wissenschaft gesehen: „Der englische Physiker Robert Boyle [...] übertrug den von Bacon theoretisch entwickelten Empirismus in die Praxis eines systematischen Experimentierens. Mit der empirischen Wende der Wissenschaft intensivierten sich die Kontakte zwischen Handwerker-Ingenieuren und theoretischen Wissenschaftlern. Nach und nach wurde so nicht nur die soziale Grenze zwischen Wissenschaftlern und technischen Praktikern durchlässig; auch ihre Denkweisen begannen sich gegenseitig zu durchdringen. Durch diesen Brückenschlag entstand auf der einen Seite die moderne Naturwissenschaft, auf der anderen Seite die schrittweise Verwissenschaftlichung der Technikentwicklung.“

Literatur

- [A00] Antoniou, I., Reeve, M., Stenning, V.
The information society as a complex system
J. Universal Computer Science 6, 2000, 272-288
- [An()] o.V.
7000 Jahre Handwerk und Technik
Pawlak, Herrsching, o.J.
- [B00] Bauer, F.L.
A computer pioneers talk: Pioneering work in software during the
fifties in Central Europe
In: [I00], 11-20
- [B91] Bauer, F.L.
Informatik und Algebra
In: Broy, M. (Hrsg.) Informatik und Mathematik, Springer, Berlin, 1991,
28-40
- [B96] Beutelsbacher, A.
„In Mathe war ich immer schlecht...“
Vieweg, Braunschweig, 1996
- [Br96] Brooks, F.P.
The computer scientist as toolsmith II
Communications of the ACM 39, 1996, 61-68
- [BL55] o. V.
Beckmanns Neues Weltlexikon
Freytag, München, 1955
- [C99] Cerruzzi, P.E.
A History of Modern Computing
The MIT Press, Cambridge, MA, 3rd printing, 1999
- [C95] Coy, W.
Automat – Werkzeug – Medium
Informatik-Spektrum 18, 1995, 31-38

- [C97] Coy, W.
Defining discipline
In: [F97], 21-35
- [D75] Daumas, M.
Technikgeschichte: ihr Gegenstand, ihre Grenzen, ihre Methoden
In: Hausen, K., Rürup, R. (Hrsg.) [HR75], 31-45
- [De01] Desel, J. (Hrsg.)
Das ist Informatik
Springer, Berlin, 2001
- [Du01] Duddeck, H.
„Aus Schaden wird man klug ... ? Wie Technik Wissen gewinnt
Abhandlungen der Braunschweigischen Wissenschaftlichen Gesellschaft, L (2001), Braunschweig, 2001, 135-159
- [DF00] Dyson, F.J.
Die Sonne, das Genom und das Internet
Fischer, Frankfurt, 2000
- [DG99] Dyson, G.
Darwin Among the Machines
Penguin Books, London, 1999
- [EB99] Encyclopaedia Britannica
Encyclopaedia Britannica Inc., Chicago, 1999
- [F94] Forman, P.
Weimarer Kultur, Kausalität und Quantentheorie 1918 – 1927
In: [vM942], 61-179
- [F97] Freksa, C., Jantzen, M., Valk, R. (Eds.)
Foundations of Computer Science – Potential – Theory - Cognition
Springer, 1997
- [G98] Gal-Ezer, J., Harel, D.
What (else) should CS educators know?
Communications of the ACM 41, 1998, 77-84

- [Ga97] Galison, P.
Image and Logic – A Material Culture of Microphysics
The University of Chicago Presss, Chicago, 1997
- [G81] Ganzhorn, K.E., Schulz, K.M., Walter, W.
Datenverarbeitungssysteme
Springer, Berlin, 1981
- [GI72] GI – 2. Jahrestagung, Karlsruhe, 2.-4. Oktober 1972
herausgegeben im Auftrag der Gesellschaft für Informatik von
P. Deussen
Springer, Berlin, 1973
- [Go98] Goethe, J.W. von
Werke, Kommentare und Register
Hamburger Ausgabe in 14 Bänden
dtv, 1998
- [G95] Goos, G.
Vorlesungen über Informatik, Band 1
Springer, Berlin, 1995
- [Gr81] Gries, D.
The Science of Programming
Springer, Berlin, 1981
- [G97] Gruska, J., Vollmar, R.
Towards adjusting informatics education to information
In: [F97], 49-67
- [H95] Hartmanis, J.
Turing Award Lecture: On computational complexity and the nature of
Computer Science
ACM Computing Surveys, 27, 1995, 7-16
- [H92] Hartmanis, J., Lin, H. (Eds.)
Computing the Future
National Academy Press, Washington, D.C., 1992

- [H02] Hartmann, W., Nievergelt, J.
Informatik und Bildung zwischen Wandel und Beständigkeit
Informatik-Spektrum 16, 2002, 465-476
- [HR75] Hausen, K., Rürup, R. (Hrsg.)
Moderne Technikgeschichte
Kiepenheuer&Witsch, Köln, 1975
- [H69] Heisenberg, W.
Der Teil und das Ganze
Piper, München, 1969
- [H71] Heisenberg, W.
Die Bedeutung des Schönen in der exakten Naturwissenschaft
In: Bayerische Akademie der Schönen Künste (Hrsg.)
Ensemble 2
Oldenbourg, 1971, 228-243
- [H99] Hoffmann, P.
Der Mann, der die Zahlen liebte – Die erstaunliche Geschichte des
Paul Erdős und die Suche nach der Schönheit in der Mathematik
Ullstein, Berlin, 1999
- [H01] Hopcroft, J.E., Motwani, R., Ullman, J.D.
Introduction to Automata Theory, Languages, and Computation
Addison-Wesley, Boston, 2nd ed., 2001
- [I00] ICHC 2000: Mapping the History of Computing – Software Issues
Heinz Nixdorf MuseumsForum Paderborn, April 5-8, 2000
- [J77] Jagger, C.
Wunderwerk Uhr
Zollikon, 1977
- [K74] Kaufmann, H.
Die Ahnen des Computers
Econ, Düsseldorf, 1974

- [K01] Kowar, H.
Die Wiener Flötenuhr
Technisches Museum Wien, 2001
- [K76] Knuth, D.
The Art of Computer Programming
Addison-Wesley, Boston, 1976
- [M01] Markl, H.
Das Genom, nicht der Mensch wurde entziffert
Humboldt Kosmos 77, 2001, 18-19
- [MP01] Max-Planck-Gesellschaft
Verantwortliches Handeln in der Wissenschaft: Analysen und
Empfehlungen
Max-Planck-Forum 3, München, 2001
- [M719] Meyers Enzyklopädisches Lexikon
Bibliographisches Institut, Mannheim, 9. Aufl., 1971-1979
- [O97] Oosterlee, K., Schüller, A., Trottenberg, U.
Durchbruch im Wissenschaftlichen Rechnen durch adaptive Mehr-
gitterverfahren auf Parallelrechnern
Der GMD-Spiegel 3, 1997, 15 -18
- [R93] Rafeiner, O. (Hrsg.)
Patente, Marken, Muster, Märkte – Der gewerbliche Rechtsschutz
international
Manz'sche Verlags- und Universitätsbuchhandlung, Wien, 1993
- [R91] Rechenberg, P.
Was ist Informatik?
Hanser, München, 1991
- [R01] Reuter, A.
Was lehren wir eigentlich, wenn wir Informatik lehren?
In: [De01], 33-46

- [S81] Salomaa, A.
Jewels of Formal Language Theory
Computer Science Press, Rockville, 1981
- [S96] Schinzel, B. (Hrsg.)
Schnittstellen
Vieweg, Braunschweig, 1996
- [S77] Sinz, H.
Das Handwerk
Econ, Düsseldorf, 1977
- [S02] SOLYP Informatik GmbH
<http://www.solyp.de/2960.html>, 2002
- [S01] Stiege, G.
Virtualität in der Informatik
Carolo-Wilhelmina XXXVI, 2001, 10-13
- [T01] Tauber, D.
Software-Entwicklung im industriellen Maßstab
In: [De01], 85-98
- [T81] Toulmin, S.
Voraussicht und Verstehen
Suhrkamp, Frankfurt/M., 1981
- [T36] Turing, A.
On computable numbers with an application to the Entscheidungsproblem.
Proc. London Math. Soc. (2), 42, 230-265, 1936-7.
Correction, *ibidem*, 43, 544-546, 1937
- [V102] Vollmar, R.
Von Zielen und Grenzen der Informatik
Abhandlungen der Braunschweigischen Wissenschaftlichen Gesellschaft, LI (2002), 9-24

- [V202] Vollmar, R.
Seit wann gibt es Informatik?
Abhandlungen der Braunschweigischen Wissenschaftlichen Gesellschaft, LI (2002), 25-47
- [vM941] von Meyenn, K.
Ist die Quantentheorie milieubedingt?
In: [vM942], 3-58
- [vM942] von Meyenn, K. (Hrsg.)
Quantenmechanik und Weimarer Republik
Vieweg, Braunschweig, 1994
- [vN63] von Neumann, J.
Collected Works (ed. by A.H. Taub)
Pergamon Press, Oxford, 1963
- [W83] von Weizsäcker, C. F.
Wahrnehmung der Neuzeit
Hanser, 4. Aufl., 1983
- [W831] von Weizsäcker, C. F.
Bohr und Heisenberg. Eine Erinnerung aus dem Jahr 1932
In: [W83], 134-146
- [W832] von Weizsäcker, C. F.
Heisenberg und Heidegger über das Schöne und die Kunst
In: [W83], 147-170
- [W01] Weber, H.
Sind Informatiker auch gute Software-Ingenieure?
In: [De01], 67-74
- [W02] Wurster, C.
Computers - Eine illustrierte Geschichte
(oder: Computer - Eine illustrierte Geschichte
Umschlag und Titelblatt unterschiedlich!)
Taschen, Köln, 2002 (?)

- [Z93] Zemanek, H.
Das Mailüfterl – ein österreichischer Aufbruch ins Computerzeitalter
In: [R93], 142-160
- [Z92] Zemanek, H.
Das geistige Umfeld der Informationstechnik
Springer, Berlin, 1992
- [Z96] Zemanek, H.
Werkzeug Computer: Verstärker von Intelligenz und Sorglosigkeit
In: [S96], 31-47
- [Z01] Zemanek, H.
Vom Mailüfterl zum Internet
Geschichte, Perspektiven und Kritik der Informationstechnik
Picus, Wien, 2001
- [Ze01] Zemanek, H.
Automaten des 18. Jahrhunderts
Anhang in [K01], 237-270
- [Z84] Zuse, K.
Der Computer - Mein Lebenswerk
Springer, Berlin, 1984