

# "Welchen Wert haben theoretische Grundlagen in der Berufspraxis? Was Theorie leisten kann und soll."

## Universalismus, Abstraktion und Modellbildung in der Informatik

Britta Schinzel

Unter Theorie verstehen wir in der Informatik gemeinhin die mathematischen (logischen algebraischen, kombinatorischen) Modellbildungen, deren sich die Informatik bedient, und ihre Prozessierung in Algorithmen, auf Automaten, Rechnern, sowie deren Erfassung und Beobachtung in semantischer (für den Korrektheitsnachweis) und komplexitätstheoretischer Hinsicht (zur Abschätzung der Ressourcenfragen).

Dazu gehört auch die Fundierung der o.g. Elemente in der Mathematik.<sup>1</sup>

Zunächst möchte ich skizzieren, welchen Wert die theoretische Informatik für die Professionalisierung der Disziplin insgesamt und so auch vermittelt für die informatische Praxis hat. Dann gehe ich auf den methodischen Universalismus der Informatik für die verschiedensten

---

<sup>1</sup> Ergänzungswürdig erscheint eine solche Theorieauffassung nach W. Coy [ W. Coy et al (Hrsg.): Sichtweisen der Informatik, Vieweg, Braunschweig, 1992] in Hinsicht auf geistes- und gesellschaftswissenschaftliche Fundierungen der Begrifflichkeiten und Objekte der Informatik deshalb, weil die Informatik im Unterschied zur Mathematik ihre Erkenntnisgegenstände und Problemlösungen in der sozialen Realität (z.B. von Arbeit, Freizeit und Organisationen) plazierte, wo sie nicht nur die abstrakten formalen Bedeutungen, sondern auch gegenständliche Bedeutungen in der realen Welt, umgangssprachliche Bedeutungen im sozialen Leben erlangen. Da informatische Begriffe, wie Programm, Daten, Abstraktion, Information, Dämonen u.s.w. in verschiedensten Bedeutungen gebraucht werden, erscheint eine Fundierung z.B. zur sprachlichen Vereinheitlichung und zur Kommunikationsfähigkeit notwendig. Mehr noch erweisen sich Sprachschöpfungen zur Kommunikation und zur Dokumentation der abstrakten informatischen Objekte, vor allem jener neuen Begriffe, die über die Benutzungsschnittstellen an Anwender übergeben werden, als unverzichtbar, und sie tauchen auch laufend neu auf, sind semantisch diffus, erfahren Bedeutungsverschiebungen. Da sie mit performativen Wirkungen verknüpft sind, führen solche Kontingenzen zu erheblichen Reibungen und Zerrungen in den IT einbettenden Zusammenhängen.

So wird eine (geisteswissenschaftliche) Theorie des Formalen immer wichtiger für die Praxis.

Anwendungen ein, um einen Einblick in die Grenzen der informatischen Problemlösungsfähigkeiten zu gewinnen. Schließlich werden kurz für die Professionalisierung der InformatikerInnen notwendige Fähigkeiten angesprochen, die auch über die formalen, theoretischen hinausgehen.

## **I. Probleme der Disziplin Informatik**

### **1. Dynamik der Entwicklung**

Als Wissenschaftsdisziplin hat sich die Informatik in der kurzen Zeit ihres Bestehens enorm verändert und ausgeweitet. Die Dynamik der Entwicklung in der Informatik und in der Software und Hardware führt zu dem rasch verfallenden aktuellen Wissen in der Informatik. Programmierparadigmen, Programmiersprachen, Software, Hardware, Umgebungen, Betriebssysteme usw. wechseln mit großer Geschwindigkeit und bilden vergängliches, stets neu zu erarbeitendes Wissen.

Anders als die Mathematik hat die junge Informatik bereits mit dem Problem des Historischen, Gewachsenen der Technik zu kämpfen: Altlasten verhindern die Umsetzung der Innovationen aus Forschung und Entwicklung in die Software-Praxis, da die Anbindungen an das Bestehende nicht umstandslos und schonend genug (Brüche erzeugen Instabilitäten und Fehler) gelingen.

Die Entwicklungsdynamik ist so groß, daß man von „Halbwertszeiten“ des Verfalls von Studieninhalten innerhalb von 5 Jahren spricht. Dazu haben die rasante Entwicklung und Verbreitung der Informations- und Kommunikationstechnologien in der Praxis beigetragen und die Tatsache, daß die Informatik in immer neue und breitere Anwendungsbereiche eindringt.<sup>2</sup>

---

2

Auch wegen dieses stark expansiven Charakters ist nach W. Coy [s.a.a.O.] eine geisteswiss. Fundierung der Informatik unbedingt nötig, da sich so die Frage nach der durch die Wissenschaft Informatik leistbaren Integration von technischen, geistes- und gesellschaftswissenschaftlichen Grundlagen und Grundannahmen, also die Frage nach ihren wissenschaftlichen Fähigkeiten und Grenzen (sachlich, methodisch, verantwortbarer Einsatz) verschärft.

Bisher ist es der Informatik jedoch nicht gelungen, die Vielfalt von Wissensgebieten, die durch sie berührt werden, interdisziplinär mit sich zu verknüpfen. Vielmehr hat sie bisher auf die Anforderungen aus verschiedenen Praxisbereichen sowie auf die Diskussion um die Folgen des Computereinsatzes mit der teilweisen Integration von Wissenschaftsinhalten aus anderen Disziplinen reagiert. Im Ergebnis zeigt sich, daß es ihr zunehmend schwerer fällt, ihre Grenzen zu finden. Gleichzeitig aber fehlt ihr, auch aus den o.g. Gründen, ein stabiles Zentrum.

## **2. Kern der Informatik und Professionalisierung**

Sicher ist der Rechner medialer Angelpunkt der Informatik, doch ist er keineswegs inhaltlich zentral. Die Softwareproduktion steht zwar im Mittelpunkt, aber gerade sie ist sehr stark der Veränderung, fast möchte ich sagen, Moden unterworfen und methodisch noch nicht zufriedenstellend ausgereift, daß sie einen auch nur halbwegs stabilen Bereich der Informatik bildete. Zur Herstellung von Wissensmonopolen für die Professionalisierung müßte sich so etwas wie ein invarianter Kern an Erkenntnisinteressen, Methoden, Begriffen und Konzepten herauschälen und unabhängig von der raschen Entwicklung bewahren lassen, wenn das Fach in seiner jetzigen Struktur überdauern will. Es erhebt sich somit die Frage, was den unveränderlichen Kern der Informatik darstellt, sowohl mit Bezug auf die Erkenntnisinteressen als auch bezüglich der Stabilität.

## **II. Theorie als stabiler Kern**

Invarianten informatischen Wissens zeigen sich aber doch im Bereich der Theorie: große Teile der theoretischen Informatik bilden gleichbleibendes Grundwissen, das zwar auch noch verschiedenen Moden, Schulen und unterschiedlichen Bewertungen unterworfen ist, wovon dennoch große Teile zur Professionalität von InformatikerInnen beitragen. Prinzipielle Ergebnisse der Automatentheorie und der formalen Sprachen, der Rekursionstheorie, der klassischen und nichtklassischen Logiken, der Semantik von Programmiersprachen, der Komplexitätstheorie, Algorithmentheorie und der konkreten Algorithmen und ihrer Komplexitätslokalisierung sind festes Grundwissen der Informatik, das immer wieder und an den verschiedensten Stellen gebraucht wird und damit zum Grundwissen der Informatikausbildung

gehört. (1, 4)

Werden aber theoretische Grundlagen im Studium nicht ausreichend vermittelt und erlernt, so sind sie ex post in der betrieblichen Praxis kaum mehr nachzuholen. (1, 2, 4)

Theoretische Informatik bildet somit eine Invariante und gehört damit zum unveränderlichen, wenn auch zu erweiternden Kern der Informatik.

Doch nicht nur für die Lehre, auch für die Forschung kommen Desiderate an Theorie-Entwicklung aus der Praxis der Softwareproduktion. Erweiterungen der theoretischen Basis sind unverzichtbar notwendig z.B. für bessere Verifikationsmethoden, sauberere semantische Durchdringung der informatischen Modellbildungen und Sprachen, etc.

Auch die theoretische Informatik ist also noch nicht ausgereift. Für Basisprobleme der Informatik existieren noch keine theoretischen Lösungen (auch wenn sie nicht im Bereich der Unmöglichkeit durch Ressourcenüberschreitung liegen), viele Theorie-Teile leben unverbunden und unvereinbar nebeneinander, ohne daß eine Einordnung oder Bewertung aus praktischer Sicht geleistet wäre (z.B. die verschiedenen Theorien zum Lernen).

### **III. Professionalisierung durch Theorie**

Die theoretisch-logische Durchdringung informatischer Methoden und Modellbildungen wird im Zuge der Professionalisierung immer wichtiger für die informatische Praxis.

Dies sieht man an Qualitätssprüngen in vielen Bereichen: der zunehmenden mathematischen Durchdringung von Chipdesign und Rechnerarchitektur, sowie deren Fehlererkennung und Verifikation, der logischen Fundierung der Semantik in den Bereichen Programmiersprachen und Übersetzer, Datenbanken, KI, u.s.w. Es scheint, daß im Bereich der Qualitätssicherung, der Spezifikation und der partiellen Verifikation besonders für sicherheitskritische Bereiche sich bessere theoretische Methoden entwickeln und durchzusetzen beginnen.

Theoretische Durchdringung hilft auch (aber natürlich nicht nur) zur Erfüllung der Anforderungen an informatische Problemlösungen aus Sicht einer sozialverträglichen Technikgestaltung:

Genauere Spezifikation, saubere Modellierung mit klarer Semantik erleichtern (im Gegensatz zum munteren Drauflosprogrammieren) partielle Korrektheitseinblicke, die Transparenz und das Erkennen der Grenzen eines Systems und sind damit Grundvoraussetzung für Konsistenz und

Vollständigkeit, Erklärungsfähigkeit des Systems, die Nachvollziehbarkeit der Systemantworten durch den Benutzer und gute Dokumentationen.

All diese Eigenschaften sind notwendige (wenn auch keinesfalls hinreichende) Bedingungen zur Verbesserung der durch das Wort "dauernde Softwarekrise" beschriebenen Situation der Softwarepraxis.

#### **IV. Methodischer Universalismus**

In Analogie zu mathematischen Grundlagen haben auch die theoretisch-informatischen Gebiete universalistische Eigenschaften: sie sind verhältnismäßig unabhängig vom Konkreten in dem formalisierenden Fach Informatik und in seiner Softwareerstellung einsetzbar. Da die Methoden der Abstraktion und Modellbildung, wie der Algorithmisierung in allen Anwendungen sinnvoll sein können, ist die Einübung in diese Methodik unabhängig vom Anwendungskontext notwendig.

Diese Eigenschaft der Abstraktion ins Formale, Sinnentleerte (im Sinne der Herkunft der Probleme aus der realen Welt, nicht im Sinne etwa einer wahrheitsfunktionalen Semantik) erlaubt erst die uniforme Anwendbarkeit informatischer Problemlösungen in den verschiedensten Kontexten und Wissensbereichen.<sup>3</sup> In vielen Anwendungsbereichen werden allerdings durchaus spezifische Methoden entwickelt, so etwa in Mustererkennung und Robotik und in anderen Bereichen der KI.

Die Anwendbarkeit und die wirkliche Plazierung der informatischen Problemlösungen in der realen Welt unterscheidet die Informatik von der Mathematik. Dies macht auch den Unterschied bei der Frage nach den Erkenntnisinteressen der Informatik aus: sind jene der Mathematik in Eigenschaften, Bedeutungsfülle und deren Konsequenzen für die beweisbaren Strukturen der von ihr selbst geschaffenen mathematischen Sprache zu suchen, so genügt eine solche Bestimmung für die Informatik nicht. Hier wird der Bereich des Formalen überschritten zum Konkreten

---

<sup>3</sup> Es gehört allerdings zu den professionalisierten Defiziten der InformatikerInnen, die Anbindungen solcher formalisierter Problemlösungen an die Anwendungskontexte nicht oder zu wenig zu berücksichtigen. Mit der Entwicklung von Benutzungsschnittstellen hat hier ein Umdenken zwar eingesetzt, aber es betrifft zunächst nur die Oberfläche der Verbindung Formales/Nichtformales.

sozialer und technischer Semantiken. Erkenntnisinteressen also hat die Informatik so auch in den erwähnten konkreten Anwendungsbereichen, aber mehr noch in der Beurteilung und Behandlung der Grenze bzw. der Lücke zwischen Formalem und dem Nichtformalem, der weichen, fließenden Realität des Sozialen, der Organisationen, der Arbeit u.s.w., in der sie ihre Produkte plaziert

## **V. Erkennen der Grenzen der Machbarkeit mit informatischen Mitteln**

Eine wichtige indirekte Wirkung der Ergebnisse der theoretischen Informatik liegt für die Studierenden in dem Erkennen des universellen Teils der Einschränkung der Machbarkeit mit Hilfe informatischer, formalisierender Methoden. Dies wird durch die Ergebnisse der Logik, der Komplexitätstheorie und der Algorithmenanalyse deutlich. Die Gödelschen Unentscheidbarkeitsresultate lassen sich für eine große Zahl grundlegender informatischer Probleme durchziehen (wie z.B. die allgemeine (uniforme) Frage der Erreichbarkeit irgendeines Programmteils in einem oder jedem Programmlauf). Auch wenn entscheidbar, so sind selbst einfache Fragen der Erfüllbarkeit einer aussagenlogischen Formel, des scheduling oder von kürzesten Wegen u.dgl., uniform gestellt, nicht sequentiell ausführbar auf von Neumann-Rechnern mit verfügbaren Ressourcen an Zeit und Hardware. Auch für parallele und verteilte Systeme sind viele Aufgaben nicht effizienter lösbar als sequentiell.

Dies alles sollte die Studierenden an die Eingeschränktheit ihrer methodischen Möglichkeiten erinnern und sie von allzu großen universalistischen Problemlösungsversuchen abhalten. Solche Ergebnisse und Forschungen können auch dem weitverbreiteten, durch Unkenntnis und z.B. durch Versprechungen der KI (harte und weiche KI-These) genährten Allmachbarkeitswahn entgegenwirken. (3)

Dies sind allerdings keineswegs die einzigen Grenzen, vor die sich informatische Problemlösungsversuche gestellt sehen: z.B. entstehen mit Software grundsätzliche Probleme, die sich weder durch bessere Theorie noch durch besseres engineering weggestalten lassen. Diese haben mit dem Gegensatz zwischen Formalem bzw. Formalisiertem und der Einbettung der Formalismen in nichformalisierbare Zusammenhänge, wie dies soziale Strukturen,

Arbeitsumgebungen und Organisationen sind, zu tun. Durch eindeutige deterministische Problemlösungen entstehen feste Verbindungen in einer veränderlichen Umwelt, welche die Veränderung behindern ("Softwarezement"). Diese Gegensätze sind der Informatik inhärent. Es gibt jedoch verschiedene Formen und Möglichkeiten, mit diesen Widersprüchen umzugehen, z.B. Software vorwiegend für Routinen und wenig für veränderliche Strukturen einzusetzen, kleine austauschbare Systeme zu entwickeln, oder dynamische, adaptierbare, situationsorientierte Benutzungsschnittstellen einzubauen, oder sei es auch, im Einzelfall von einer informationstechnischen Problemlösung einmal Abstand zu nehmen.

Darüber hinaus setzen die kognitiven Schwierigkeiten des Menschen, mit unvorstellbaren Größen, diskreten Phänomenen (z.B. Nichtrobustheit) und komplexen Strukturen (z.B. des Zusammenspiels der Wirkungen von Teilen von Programmen, der Unverständlichkeit der Wirkung stark rekursiver Programmstücke) explizit umzugehen, der Durchschaubarkeit und damit Kontrollierbarkeit (denn nie läßt sich alles automatisch kontrollieren) und dem Verständnis der Abläufe und Folgen enge Grenzen.

Gerade dadurch ist die Informatik als Technik eine radikale Neuheit und diese, so Dijkstra, sollte nicht verniedlicht, sondern die Fremdheit sollte gelehrt werden.

## **VI. Entwicklung notwendiger Fähigkeiten**

Die Qualitätskriterien für das Studium müssen in der Professionalisierung des Fachs und in der informatischen Praxis gesucht werden. Dabei muß auch dem äußerst dynamischen Charakter der Informatik Rechnung getragen werden, letzterem möglicherweise dadurch, daß es selbst als Teil eines Prozesses konzipiert wird. Ein Studium der Informatik sollte nicht nur Wissen und Fertigkeiten vermitteln, vielmehr sollten diese Inhalte und Methoden Transportmittel zum Erwerb bestimmter Fähigkeiten sein. Ausgangspunkt für die Ausbildung müssen jene Kenntnisse, Fähigkeiten und Fertigkeiten sein, mit denen die Absolventen den Anforderungen ihres Berufes gerecht werden können. Welche dies sein sollten und wie sie vermittelt werden können, dies anzudenken machte sich eine Diskussion der Tagung "Theorie der Informatik" in

Zu diesen Fähigkeiten gehört die Fähigkeit der Abstraktion, der möglichst adäquaten formalen Modellbildung und der möglichst effizienten Algorithmisierung aus den o.g. Gründen, welche gerade in den theoretischen Fächern geübt werden.

Zu diesen Fähigkeiten gehören aber weiter und unter anderem

- die Fähigkeit, in Zusammenhängen zu denken und Gesamtsichten zu erwerben
- nicht zwanghaft nach Lösungen zu suchen, sondern auch abschätzen zu können, ob und wann die Lösung eines Problems außerhalb des eigenen Könnens oder des eigenen Fachgebietes liegt
- Widersprüche und Konflikte erkennen und aushalten zu können
- kommunikative Kompetenzen und Selbstreflexionsfähigkeit zu erwerben
- eine Streitkultur zu entwickeln, um Positionen zu finden, die es erlauben, allein und gemeinsam Verantwortung wahrnehmen zu können.

Leider verstellt die Einübung und Fixierung aufs Formale und die informatischen Kompetenzen im engeren Sinne oft den Blick auf gerade diese so notwendigen sprachlichen und kommunikativen Kompetenzen und führt so zu defizitären informatischen Produkten und damit auch zu einem volkswirtschaftlichen Schaden.<sup>5</sup>

---

<sup>4</sup> Dabei waren folgende Fragen leitend:

1. Wie muß ein Curriculum verändert werden, damit Formalisieren als sinnvoller Prozeß gelehrt werden kann?
2. Wie muß ein Curriculum verändert werden, damit die spezifischen Interessen von Frauen berücksichtigt werden?
3. Wie sollen Studierende ausgebildet werden, damit sie den Einsatz von Computern in Arbeitszusammenhängen gut gestalten können?
4. Wie sollen Studierende ausgebildet werden, damit sie ihre Verantwortung wahrnehmen können?

<sup>5</sup> Empirische Untersuchungen aber haben gezeigt, daß über zwei Drittel der Arbeitszeit von Software-EntwicklerInnen aus Kommunikation besteht