

Truman und die Zugänge zur Informatik

Christian Siefkes

Technische Universität Berlin
Sekt. FR 6-2
Franklinstr. 28/29, 10587 Berlin

error@cs.tu-berlin.de
<http://tal.cs.tu-berlin.de/error/>

6. März 1999

Inhaltsverzeichnis

1	Trumans Welt	2
2	Das Informatikstudium	2
3	Wissenschaft und Pluralismus	3
3.1	Paradigmenwechsel	4
3.2	Anarchie und Erkenntnis	5
4	Weltmodelle: Konstruktion von Wissenschaft und Wirklichkeit	7
4.1	Modellbildung	9
4.2	Projektstudium	9
5	Türen schließen sich	11
6	Trumans Uni	12
	Literatur	13

»Die Umwelt,
so wie wir sie wahrnehmen,
ist unsere Erfindung.«
Heinz von Foerster [9,S. 40]

»We accept the reality of the world
with which we are presented.«
Christof, Macher der Truman Show [15]

1 Trumans Welt

Truman Burbank, der Held der *Truman Show*, lebt in einer Welt, die durchkonstruiert ist bis zum letzten. Sein Lebensablauf folgt fest eingespielten Bahnen, Beruf und Freizeit sind nahezu standardisiert. Alles ist komfortabel eingerichtet. Probleme werden entweder routiniert gelöst oder verdrängt.

Schließlich erkennt Truman, daß nicht alles so sein müßte, wie es ist. Die Regeln, denen sein Leben und seine Welt unterworfen sind, sind nicht selbstverständlich, sind nicht die einzig vorstellbaren – sie sind *gemacht*, von Menschen, die ein Interesse daran haben (weil sie Truman und seine Welt so in ihrer Fernsehserie vermarkten können).

Immer wieder hört Truman, daß der Ort wo er lebt (Seahaven Island) und die Art wie er lebt die besten, ja eigentlich die einzig richtigen sind. Jede Alternative wäre schlechter oder einfach Unsinn.

Truman erkennt erst spät, daß er nicht alles als selbstverständlich hinnehmen muß, von dem ihm erzählt wird, daß es selbstverständlich ist. Er bricht aus und entdeckt eine andere Welt, von deren Existenz er zuvor nichts ahnen konnte.¹

2 Das Informatikstudium

Das heutige Informatikstudium weist gewisse Ähnlichkeiten mit Trumans Welt auf; es ist (insbesondere im Grundstudium) ähnlich eng begrenzt.

»Inzwischen wissen wir ja, was Informatik ist« – genau diese Erkenntnis an die Studierenden weiterzugeben, sei Ziel des Informatikstudiums. So begründet ein Informatikprofessor an der TU Berlin die Tatsache, daß

¹ Für kritische und anregende Bemerkungen zu meinem Text danke ich Hans-Jörg Burtschick, Frank Eckert, Joachim Korb, Lionel Pöffel, Katharina Schmidt-Brücken, Dirk Siefkes, Martin Siefkes und Bettina Törpel.

in der geplanten neuen Studien- und Prüfungsordnung das Grundstudium vollständig aus Pflichtveranstaltungen besteht. Alle Alternativen seien schließlich schlechter oder einfach Unsinn.

Kein Wunder, daß da Vorhaben, die alternative Zugänge zur Informatik öffnen wollen, nicht nur mißtrauisch beäugt, sondern regelrecht bekämpft werden. Obwohl: wenn man den eigenen Ansatz für so überlegen hält, woher dann die Angst vor – angeblich schlechteren – Alternativen?

3 Wissenschaft und Pluralismus

Für mich, als gerade ins Hauptstudium gekommenen Informatikstudenten, war die Stelle als Tutor bei dem Studienreformprojekt »Geschichte als Zugang zur Informatik«² vor allem aus wissenschaftspolitischen Gründen reizvoll. Zwar hatte ich mich schon im Wahlbereich des Grundstudiums mit Wissenschaftsgeschichte beschäftigt – doch primär war es nicht die Geschichte, die mich lockte, sondern die Hoffnung, andere Zugangsweisen, Perspektiven, Methoden zu vermitteln als im Grundstudium üblich.

Dabei umfassen die Ziele des Projekts sehr Pragmatisches:

Zum wissenschaftlichen Arbeiten gehört es, Texte kritisch zu lesen und selbst zu schreiben, Quellen auszuwerten, Ergebnisse mündlich und schriftlich darzustellen, erste eigene Forschungsthesen zu bilden. Auch das korrekte Zitieren und die richtige Strukturierung wissenschaftlicher Texte sind nicht angeboren. All das wird im Informatikstudium zwar verlangt (spätestens bei der Diplomarbeit), aber kaum vermittelt.

Gruppenarbeit ist bei uns von der ersten Woche ein zentrales Element des Studiums (und einer seiner besten Aspekte), wird aber faktisch vorausgesetzt. Gibt es Probleme mit der Zusammenarbeit, muß die Gruppe alleine damit fertig werden.

Diese Lücken wollen wir füllen.

Im Beruf müssen sich Informatikerinnen³ in zahlreichen Bereichen bewähren, sie werden nicht nur mit Formalismen und Ingenieurmäßigem konfrontiert. Wer die Geschichte der benutzten Techniken kennt, kann ihren Sinn und ihre Verwendungsmöglichkeiten besser verstehen. Wer Ahnung hat von der Geschichte des eigenen Faches, vermeidet womöglich Irrwege und Fallen.

² vgl. Bettina Törpels Artikel in diesem Band

³ Wo immer ich in diesem Text die weibliche Form verwende, sind Männer eingeschlossen.

Diese pragmatischen Ziele sind vergleichbar mit Versuchen, Trumans Welt (Seahaven Island) erweitern und verbessern zu wollen, etwa durch den Bau eines Opernhauses oder einer Bibliothek. Mit unserem Projekt wollen wir jedoch noch mehr erreichen.

3.1 Paradigmenwechsel

Laut Kuhn [13] wechseln sich lange Ruheperioden des Forschens innerhalb eines konsistenten Paradigmas ab mit Revolutionen, bei denen das alte Paradigma als unbefriedigend über Bord geworfen und durch ein neues ersetzt wird. Kommt es zu einer Abweichung der Wirklichkeit von der Theorie, so geht die normale Wissenschaft von Rätseln aus, die sie zu lösen versucht, ohne ihr Paradigma anzutasten. Beim gehäuften Auftreten von unerklärlichen Anomalien kommt es schließlich zu einer Krise, einer Phase der Unsicherheit. Die Anomalien werden schließlich von einigen Wissenschaftlerinnen als Gegenbeispiele angesehen, die das alte Paradigma widerlegen. In außerordentlicher Forschung wird so, vorläufig abseits vom breiten Strom der Normalwissenschaft, nach neuen Theorien gesucht, die auch die Anomalien erklären können. Ist eine überzeugende Theorie gefunden und hat sich die Krise bis in weite Teile der Normalwissenschaft ausgebreitet, kommt es zur Revolution: Das neue Paradigma wird angenommen, die außerordentliche Forschung ist Normalwissenschaft geworden.

Kuhn [13, Kap. 10 u. 12] zeigt, daß Erfahrung theorieabhängig ist: Was Wissenschaftlerinnen beobachten, hängt zu einem guten Teil von ihren Erwartungen ab. Deshalb leben Wissenschaftlerinnen vor und nach einer Revolution gewissermaßen in verschiedenen Welten; sind die jeweils akzeptierten Theorien inkommensurabel: Sie können nicht miteinander verglichen werden, weil Ausgangspunkte und Grundbegriffe zu verschieden sind. Daher sieht er wissenschaftlichen Wandel weniger als Ergebnis kritischer Überlegungen, sondern religiösen Bekehrungen oder politischen Revolutionen vergleichbar; eher gesellschaftlich oder psychologisch als rational erklärbar.

Innerhalb der Informatik läßt sich dies etwa anhand der Übergänge von einem Programmierparadigma zu einem anderen nachvollziehen. In unserer ersten Veranstaltung⁴ haben wir uns insbesondere den Ursprung der strukturierten Programmierung angesehen (bei den Arbeitsgruppen, die sich mit der Entstehung der *Unified Modeling Language* (UML) beschäftigt haben, spielte auch der Wechsel zum objektorientierten Paradigma eine Rolle). Bei den Texten von Dijkstra [4, 5] scheint es weniger

⁴ »Computer und die Ordnung im Kopf: Geschichte graphischer Programmierhilfsmittel«, Integrierte Veranstaltung im Wintersemester 1998/99

um wissenschaftliche als vielmehr um Glaubensfragen zu gehen, bei den Debatten für [11] und wider [16] *GoTo* läßt sich die Inkommensurabilität der Positionen erahnen.

Bei einer Sitzung des Fachbereichsrats Informatik⁵ hat einer der Professoren, die gegen unser Projekt sind, einmal den Verdacht geäußert, wir wollten »heimlich einen neuen Grundstudiumszyklus aufbauen«, der letztlich die jetzigen obligatorischen Veranstaltungszyklen ersetzen sollte. Fürchtet man da etwa eine wissenschaftliche Revolution, in der die etablierte Rahmenordnungs-Informatik von einem neuen Paradigma verdrängt wird ...?

Unsere wahren Ziele sind natürlich viel bescheidener. Uns geht es nicht um noch mehr oder neue Pflichtveranstaltungen, sondern um mehr Möglichkeiten, mehr Auswahl. Wir wollen keinen ausschließlichen, sondern einen alternativen Zugang bieten – weniger Paradigmenwechsel als vielmehr Pluralismus im Feyerabend'schen Sinne.

3.2 Anarchie und Erkenntnis

Paul Feyerabend geht mit seinem erkenntnistheoretischen Anarchismus [6] über Kuhn hinaus: *Anything goes*, es kann keine objektiven Kriterien für oder gegen eine Theorie geben. Da Theorien wegen der Theorieabhängigkeit der Erfahrung nicht objektiv verglichen werden können, versuchen die Wissenschaftlerinnen mit Überredung und Propaganda, ihre Kolleginnen und die Öffentlichkeit zu ihren Ansätzen zu bekehren.

Er weist Kuhns Vorstellung von der Rolle der Normalwissenschaft als unrealistisch und zu starr zurück [7]: Es gibt keine monolithische Normalwissenschaft; die Wissenschaftsgeschichte ist weitgehend eine Geschichte konkurrierender Theorien. Theorienpluralismus gibt es also nicht nur in Phasen der Krise und außerordentlichen Forschung, sondern zu allen Zeiten. Kuhns Normalwissenschaft ist ein Mythos [7,S. 164f u. 168–170].

Krisen führen zudem nach Feyerabend nicht zur Entstehung alternativer Theorien, sondern werden erst durch deren Existenz ausgelöst [7,S. 169]. Die Probleme des Ptolemäischen Weltbilds seien etwa erst durch die Propagandistinnen der Kopernikanischen Alternative entdeckt und ernst genommen worden [6,S. 69–73].

Feyerabend hebt zwei methodologische Prinzipien hervor [7,S. 160–162 u. 166–169]:

⁵ Der mehrheitlich mit Professorinnen besetzte Fachbereichsrat entscheidet u.a. über Lehraufträge für Veranstaltungen, die Studien- und Prüfungsordnung sowie Reformprojekte wie unseres.

Dem *Prinzip des Proliferierens* zufolge sollen Wissenschaftlerinnen Wert legen auf die Einführung von Konkurrenztheorien und damit die Herstellung von Theorienpluralismus.

Das *Prinzip der Beharrlichkeit* verlangt, eine einmal aufgestellte Theorie hartnäckig zu verteidigen. Dabei sollen verschiedene Gruppen von Wissenschaftlerinnen verschiedene Theorien verteidigen, um so den Zustand des Theorienpluralismus zu bewahren.

Diese beiden Prinzipien sollen laut Feyerabend gleichzeitig in Kraft sein, so daß die Wissenschaft von der Konkurrenz mehrerer hartnäckig verteidigter Theorien geprägt wird. Das Wechselspiel zwischen Beharrlichkeit und Proliferieren ist nicht nur eine wissenschaftsgeschichtliche Tatsache, sondern auch methodologisch wünschenswert.

Für die Informatik als Disziplin trifft dies auch durchaus zu⁶. Um so auffälliger, daß im Informatik-(Grund-)Studium einige mehr oder weniger standardisierte Methoden dominieren – Alternativen gelten als unerwünscht (überholt, nicht bewährt, unbrauchbar). Für die Informatikstudierenden sind die benutzten Methoden praktisch »vom Himmel gefallen«; sie werden nur gelehrt, nicht begründet. Woher sie kommen, warum sie verwendet werden, in wessen Interesse sie liegen – das wird nie thematisiert.

In unseren Veranstaltungen werfen wir genau diese Fragen auf. Der geschichtliche Zugang kann Einblicke öffnen, die die Vorstellung »so wie es ist, mußte es kommen / ist es ideal« (primitive Evolutionsmodelle, Fortschrittsglauben) in Frage stellen. Das Verstehen der Hintergründe und Umstände der Entwicklung einer Technik kann so zu vertieftem Verständnis führen, aber auch zum Hinterfragen des Bestehenden. Es kann das kritische Denken anregen.

Im Umgang mit geschichtlichen Methoden lernen die Teilnehmenden die Arbeits- und Herangehensweisen eines anderen Fachs kennen, wo vieles anders ist (die formalen und inhaltlichen Ansprüche an Texte etwa). Der Gültigkeitsanspruch des Gewohnten relativiert sich so und macht einem Methodenpluralismus Platz. Die geschichtliche Sicht ist dabei kein Selbstzweck⁷, sondern dient zum Verstehen und Bewerten der Inhalte des eigenen Faches.

⁶ man vergleiche etwa die Fachorganisationen *Gesellschaft für Informatik* und *Forum InformatikerInnen für Frieden und gesellschaftliche Verantwortung*

⁷ obwohl sie sich auch als solcher lohnt!

4 Weltmodelle: Konstruktion von Wissenschaft und Wirklichkeit

Die Bildung und Benutzung von Modellen spielt eine entscheidende Rolle in der Informatik. Das ist nichts Neues. Schon 1971 wurde als Aufgabe der Informatik genannt, »durch Abstraktion und Modellbildung von speziellen Gegebenheiten sowohl der technischen Realisierung existierender Datenverarbeitungsanlagen als auch von Besonderheiten spezieller Anwendungen abzusehen und dadurch zu den allgemeinen Gesetzen, die der Informationsverarbeitung zugrunde liegen, vorzustoßen« [10,S. 4].

Ursprünglich gehörte es zu den Zielen unseres Projekts, ein Modellierungspraktikum anzubieten. Praktika sind Lehrveranstaltungen im Grundstudium, in deren Mittelpunkt das selbständige Arbeiten (in Gruppen) mit bestimmten Techniken steht. Aufgrund der verringerten Personalausstattung, mit der das Projekt bewilligt wurde, mußten wir dieses Ziel abspecken. Unsere Hoffnung ist nun, in unserer letzten Veranstaltung Ideen und Konzepte für ein solches Praktikum zu diskutieren und zu erproben. Die Studierenden sollen mit historischen und aktuellen (Modellier-)Werkzeugen und Techniken arbeiten. Dabei soll es auch um die Frage gehen, ob und wieweit die eigenen Ziele und Vorgehensweisen von dem benutzten Werkzeug beeinflußt werden.

Anregungen kommen dafür aus einem Projekt zum Thema Modellierung [8], das vor einigen Jahren an der TU Berlin stattfand. Die Autorinnen unterscheiden zwei Sichtweisen von Modellierung [8,insb. Kap 1 u. 12]:

In der *Abbildperspektive* wird eine Eins-zu-eins-Verknüpfung zwischen Modell und Realität unterstellt: Das Modell soll ein neutrales und objektiv interpretierbares Bild eines Teils der Realität – des Originals – darstellen. Um beurteilen zu können, ob das Modell ein Abbild des Originals ist, muß das Original unabhängig vom Modell existieren, so daß beide verglichen werden können. Vorbedingung dafür ist die Annahme, daß es nur eine Realität gibt, daß alle in derselben Realität leben und daß diese Realität objektiv erkennbar ist.

Die *Konstruktionsperspektive* geht dagegen davon aus, daß der Akt des Modellierens die Realität verändert. Die Modelle, die man benutzt, werden selbst Teil der Wirklichkeit, in und mit der man leben muß. Realität ist immer erfahrene Realität, abhängig vom erfahrenden Subjekt. Jedes Original ist immer schon Ergebnis einer Modellierung. Mit der Modellierung wird nicht nur das Modell, sondern auch das Original (um)konstruiert. Modelle sind mehr als nur Spiegelbilder, sie wirken selbst. Diese Perspektive kommt also ohne die Grundannahmen der Vergleichbarkeit und Unabhängigkeit aus.

Die Erfahrungen der Nutzerinnen sind für die Abbildperspektive uninteressant; der Problematik der Entwicklung immer komplexerer Systeme werden ausgefeilte formale Methoden entgegengesetzt.

In der Konstruktionsperspektive wird das Modell nicht anhand seiner strukturellen Ähnlichkeit zum Original beurteilt, sondern anhand seiner Angemessenheit. Erfüllt es seinen Zweck, ist es angemessen. Entscheidend sind daher jetzt die Zwecke, für die ein Modell konstruiert wird.

Zudem stellt sich die Frage nach der Angemessenheit des Originals. Auch das Original kann Änderungen unterworfen werden, wenn es seine Zwecke nicht erfüllt. ›Letzte Instanz‹ ist nicht mehr das Original, sondern die ModelliererIn. Modellieren heißt damit, Modell und Original solange zu konstruieren, bis bestimmte Interessen der Modellierenden befriedigt sind. Die Qualität des Modells hängt nur indirekt vom Original, direkt aber von den Interessen der Modellierenden ab.

Fischer et al. [8,S. 14 u. 136–138] postulieren nicht die unbedingte Überlegenheit der Konstruktions- bzw. der Abbildperspektive, sondern vermuten, daß in manchen Situationen erstere (etwa für die Erstellung erster Modelle, den Schritt vor der Formalisierung), in anderen letztere (für die Transformation einer Spezifikation in ein Programm) überlegen sein könnte.

Im normalen Informatikstudium wird dagegen fast ausschließlich die Abbildperspektive verwendet. Unter anderem zeigt sich dies daran, daß im Grundstudium in aller Regel mathematisch formulierte oder leicht formulierbare Einzelaufgaben programmiert werden, um anschließend die Korrektheit zwischen Programm und mathematischer Spezifikation zu beweisen. Hier geht es um Funktionalität und Funktionsweisen.

Im Grundstudium wird gelernt, wie man von der Formalisierung zum Programm kommt. In unserem Projekt geht es allerdings um den Schritt vor der Formalisierung: das Verstehen und Umsetzen, das Bilden von Modellen im weiteren Sinne. Daher dürfen wir die Konstruktionsperspektive nicht vernachlässigen, daher müssen wir auch nach Interessen und Zwecken fragen.

Eine weitere Vermutung von Fischer et al. [8,S. 15] erklärt den ›Grundlagenstreit‹ der Einordnung der Informatik als Ingenieur- oder Sozialwissenschaft als abhängig von der vertretenen Perspektive. Daß gerade an der TU Berlin ganz stark das Ingenieurmäßige betont wird⁸, bekräftigt die hiesige Dominanz der Abbildperspektive.

⁸ so beklagt einer der Autoren der neuen StuPO (siehe Abschnitt 5): »Zu wenige haben am Ende [des Grundstudiums] gelernt, wie ein Ingenieur zu denken« [14,S. 5]

4.1 Modellbildung

Neben diesen expliziten Modellen spielen aber auch implizite Modelle (Sichtweisen, Weltbilder, Werte) eine wichtige Rolle – nicht nur in der Wissenschaft [12, Abschnitt 4 u. 5.1]. Implizite Modelle entstehen aus unseren Sinneseindrücken und durch Überlieferung (Traditionen, Moral, Eltern und andere Vertrauenspersonen). Sie werden aber nicht nur durch unsere Wahrnehmung aufgebaut, sondern sie beeinflussen diese auch. Was zu unserem Weltbild paßt, wird leichter als ›wahr‹ akzeptiert. Das implizite Modell bildet so eine Art ›WahrnehmungsfILTER‹.

Die expliziten Modelle hängen von impliziten ab. Die Ziele (Zwecke) eines expliziten Modells basieren auf der Weltsicht der Modellierenden (oder ihrer Auftraggeberin). Dieser Einfluß persönlicher Motive und Weltbilder kann bewußt oder unbewußt erfolgen.

Umgekehrt wirken explizite Modelle auch auf implizite: Erfahrungen mit expliziten Modellen werden in die eigenen impliziten Modellvorstellungen übernommen.

Auf diese Weise haben Modellierende Macht über die Vorstellungen der Anwenderinnen. Daher gibt der KIF-Arbeitskreis Modellbildung [12, Abschnitt 5.2] Empfehlungen für Modelliererinnen:

- Verdeutlicht die Bedingungen, die in das Modell eingeflossen sind.
- Seid euch bewußt, daß das eigene Modell nur eins von mehreren möglichen ist.
- Legt offen, für welchen Bereich und für welchen Zweck ein Modell entstanden ist. Vorsicht bei Übertragungen!

Die Autorinnen fordern auf, »zu der eigenen Subjektivität zu stehen und alle Voraussetzungen, Methoden und Annahmen, die einem selbst bewußt sind, explizit zu benennen.« Sie rufen auf zur »Offenheit gegenüber anderen Modellen. Modelle, die aus unterschiedlichen Fachdisziplinen stammen, die auf verschiedenen Ebenen angesiedelt sind oder mit verschiedenen Modellierungstechniken entwickelt wurden, schließen sich nicht zwangsläufig aus [...] Sie können sich sogar ergänzen« [12, Abschnitt 5.3].

4.2 Projektstudium

Als wir noch vorhatten, ein Grundstudiums-Praktikum anzubieten, wollten wir die Teilnehmenden mit Methoden aus verschiedenen Zeiten modellieren und dann die Modellierungsprozesse und -ergebnisse vergleichen lassen: Wie wirkt sich das Werkzeug auf die Fragestellung aus?

Wie(weit) erklärt sich diese Wirkung aus Entstehung und Zweck der Methode? Praktisches Arbeiten allein reicht also nicht aus; sinnvollerweise würde die Veranstaltung eine Mischung aus praktischem und wissenschaftlichem Arbeiten. Möglicherweise modelliert eine Gruppe mit einer bestimmten Technik wie etwa UML (praktisches Arbeiten), eine andere erarbeitet die Entstehung und Geschichte dieser Technik (wissenschaftliches Arbeiten) – oder ein und dieselbe Gruppe macht beides nacheinander oder im Wechsel, vielleicht im Lauf von zwei Semestern.

Die Forderung nach einer solchen Vermischung der Lernformen findet sich auch in der StuPO 2000++ [1]. An diesem studentischen Vorschlag einer alternativen Studien- und Prüfungsordnung Informatik habe ich während und nach dem letzten Streik im Wintersemester 1997/98 mitgearbeitet. Über Projekte schreiben wir dort:

»Wichtig ist, daß das Projekt einen Lernprozeß vom Erwerben von Grundlagen über praktische und theoretische Arbeit bis zum Erstellen eines Ergebnisses umfaßt. Das Problem soll unter verschiedenen Gesichtspunkten beleuchtet werden. Dazu gehören verschiedene Fachmeinungen, aber auch wesentlich Fragen nach Folgenabschätzung, Benutzerfreundlichkeit u.ä. Es sollen verschiedene Arbeitsformen praktiziert und eingeübt werden: Diskussion in größeren Gruppen, Kleingruppenarbeit, Arbeit am Rechner, Referate halten, Arbeitsbericht erstellen, etc.« [1,S. 11]

In der StuPO 2000++ fordern wir auch eine andere Art des Studierens:

»Das wichtigste Studienziel ist nicht informatisches Wissen, sondern die Fähigkeit, in wechselnden Situationen informatische Kompetenzen zu erwerben. Dieser Herausforderung können Lehrende und Lernende nur begegnen, wenn sie daran arbeiten, Probleme als Anstöße zum Lernen zu begreifen. Aus diesem Grund lassen wir die Studentinnen projektorientiert studieren. Lernen und Lehren als lebendigen und nicht formalen Vorgang anzusehen, ist gerade für Informatikerinnen nicht einfach, aber um so wichtiger.

Bevorzugte Veranstaltungsform sind daher Projekte, da so mehr inhaltliche und soziale Kompetenzen vermittelt werden als bei Vorlesungen und Frontalunterricht.« [1,S. 10]

Ein konsequentes Projektstudium würde zu einer völlig anderen Studienstruktur führen. Die heutige Zerstückelung – in jedem Semester mehrere unabhängige Veranstaltungen, in vielen davon eine Klausur

oder mehrere Übungsblätter mit Aufgaben und Teilaufgaben – würde abgelöst durch einige große, zusammenhängende Projekte, an deren Ende jeweils ein greifbares Ergebnis stünde. So würden Zusammenhänge sichtbar, die in der Fragmentierung des Standardstudiums untergehen. Freilich bekommt man auf diese Weise nicht ›den Diplom-Informatiker‹ mit einheitlichem Wissen und genormtem Kopf.

Zudem erfordert die Umsetzung solcher Ideen Mut und Konsequenz. Der Vorschlag einer Tutorin »Man könnte das ganze Grundstudium an einem Projekt aufziehen, z.B. *Modellierung eines Schiffshebewerks*« wurde dankend aufgegriffen: Man machte daraus eine von acht Aufgaben auf einem von zehn Übungsblättern in einer von vier Lehrveranstaltungen eines der vier Grundstudiumszyklen ...

In unserem kleinen Projekt können wir eine solche Umsetzung nicht leisten. Aber Anregungen geben, erste Schritte auf neuen Wegen gehen, das können wir hoffentlich.

5 Türen schließen sich

Ausbruchsversuche sind unerwünscht und werden nicht geduldet. Beim immensen Verschulungsgrad des Informatik-Grundstudiums an vielen anderen Universitäten oder in den ›richtigen‹ Ingenieurstudiengängen sind alternative Ansätze schon jetzt nicht möglich. Zumindest können sie nicht angerechnet werden, da keine Semesterwochenstunden dafür vorgesehen sind. Oft genug sorgt die immer weiter ansteigende Arbeitsbelastung dafür, daß zusätzliche (Uni-)Aktivitäten praktisch unmöglich werden (worunter nicht nur nichtanrechenbare Veranstaltungen leiden, sondern etwa auch die Hochschulpolitik).

An der TU Berlin gibt es da relativ viel Freiheit – bisher. Doch demnächst soll eine neue Studien- und Prüfungsordnung (StuPO) Informatik in Kraft treten, nach der das Grundstudium komplett aus Pflichtveranstaltungen besteht [2; 3]. Dann ist es mit Experimenten wie dem unseren vorbei. Daß als durchschnittliche Arbeitsbelastung 47 Stunden pro Woche vorgesehen sind, dürfte die Leidenschaft der Studierenden, Zusatzveranstaltungen zu besuchen, nicht unbedingt erhöhen. Überdies wird das Grundstudium ausschließlich mit Informatik-Inhalten gefüllt: Zu Blicken über den Tellerrand oder Interdisziplinarität wird keine Möglichkeit mehr geboten. Damit dürfte die ›Gefahr‹, daß die Informatikerinnen der Zukunft noch aus dem vorgegebenen Schema ausbrechen, erheblich reduziert werden ...

6 Trumans Uni

Nach dreißig Jahren fallen Truman kleine Schäden und Inkonsistenzen in seiner Welt auf. Er macht sich auf die Suche nach Alternativen. Seine Suche ist mühsam und höchst gefährlich; über viele gutgemeinte Tips muß er sich dabei hinwegsetzen. Schließlich stößt Truman an die Grenze seiner Welt. Er entscheidet sich, lieber neue Wege zu begehen, als in seiner bequem eingerichteten Welt in Sicherheit weiterzuleben. Der Ausgang, die Tür nach draußen, öffnet sich auf leichten Druck (schließlich handelt es sich um einen Hollywoodfilm mit Happy End).

Das Informatikstudium in Deutschland ist etwa so alt wie Truman. Wenn einige der ›Bewohnerinnen‹ dieses Studiums erkennen, daß es in seiner jetzigen Form nicht die einzig denkbare, auch nicht unbedingt die beste Möglichkeit darstellt, und einige Zeit mit dem Suchen nach Alternativen verbringen, wäre ein wichtiges Ziel unseres Projekts erreicht.

Leider aber werden die Türen nach draußen immer fester verschlossen. Es sind eben nicht nur die Macher einer Fernsehsendung, die daran interessiert sind, daß dem so ist.

Wir werden an unserem Fachbereich das letzte derartige Projekt auf absehbare Zeit sein. Heute versteht man unter Studienreform etwas ganz anderes: mehr Effizienz, weniger Verzögerungen (Mitbestimmung), mehr Berufsbildung (Schulung) statt Ausbildung. Bei den jetzigen Mehrheitsverhältnissen in Fachbereichs- und Unigremien wäre ein Projekt wie das unsere nicht mehr durchsetzbar (seinerzeit war es schon knapp genug gegen die Mehrheit der Professoren).

Mit der neuen Studien- und Prüfungsordnung wird es im Grundstudium keinerlei Wahlfreiheit, nichts Interdisziplinäres oder Informatikfremdes mehr geben. So wird unser Zugang bald selbst Geschichte sein; Truman muß zurück in seine enge Welt.

Literatur

- [1] AG Studien-Utopien (1998): Offenes Studium mit der StuPO 2000++. Auch online verfügbar:
<http://autos.cs.tu-berlin.de/fb13ini/StuPO/stupo2000pp/>
Zugriff am 28. Februar 1999.
- [2] Biedl, Albrecht und Bernd Mahr (1999): Prüfungsordnung für den Studiengang Informatik an der Technischen Universität Berlin. Entwurf vom 20. Januar 1999. Auch online verfügbar:
<http://www.cs.tu-berlin.de/cs/fbv/pa/nepo.ps>
Zugriff am 1. März 1999.
- [3] Biedl, Albrecht und Bernd Mahr (1999): Studienordnung für den Studiengang Informatik an der Technischen Universität Berlin. Entwurf vom 20. Januar 1999. Auch online verfügbar:
<http://www.cs.tu-berlin.de/cs/fbv/pa/nesto.ps>
Zugriff am 1. März 1999.
- [4] Dijkstra, Edsger W. (1969): Structured Programming. In: *Software Engineering Techniques*, Hg. J. N. Buxton und B. Randell. Rom: NATO Science Committee, S. 84–88.
- [5] Dijkstra, Edsger W. (1979): Go To Statement Considered Harmful. In: *Classics in Software Engineering*, Hg. Edward Nash Yourdon, New York: Yourdon, S. 29–33.
- [6] Feyerabend, Paul K. (1976): *Wider den Methodenzwang. Skizze einer anarchistischen Erkenntnistheorie*. Frankfurt am Main: Suhrkamp.
- [7] Feyerabend, Paul K. (1978): Kuhns ›Struktur wissenschaftlicher Revolutionen‹. Ein Trostbüchlein für Spezialisten? In: *Der wissenschaftstheoretische Realismus und die Autorität der Wissenschaften*, Bd. 1 der *Ausgewählten Schriften*, Braunschweig – Wiesbaden: Vieweg, S. 153–204.
- [8] Fischer, Martin; Gernot Grube und Fanny-Michaela Reisin, Hg. (1995): *Abbild oder Konstruktion – Modellierungsperspektiven in der Informatik*. KIT Report 125. Berlin: TU Berlin. Auch online verfügbar:
<ftp://ftp.cs.tu-berlin.de/pub/local/kit/documents/KIT-Reports/r125.pdf>
Zugriff am 28. Februar 1999.
- [9] von Foerster, Heinz (1985): Das Konstruieren einer Wirklichkeit. In: *Die erfundene Wirklichkeit. Wie wissen wir, was wir zu wissen glauben?*

- Beiträge zum Konstruktivismus*, Hg. Paul Watzlawick, München – Zürich: Piper, S. 39–60.
- [10] Friedrich, Jürgen; Thomas Herrmann; Max Peschek und Arno Rolf, Hg. (1995): *Informatik und Gesellschaft*. Heidelberg – Berlin – Oxford: Spektrum.
- [11] Hopkins, Martin E. (1979): A Case for the GOTO. In: *Classics in Software Engineering*, Hg. Edward Nash Yourdon, New York: Yourdon, S. 101–109.
- [12] KIF-Arbeitskreis Modellbildung (1999): (Zwischen)Ergebnisse des AK Modellbildung. Stand 5.3.1999. Noch unveröffentlichte Dokumentation eines Arbeitskreises der Konferenz der Informatik-Fachschaften (KIF).
- [13] Kuhn, Thomas S. (1976): *Die Struktur wissenschaftlicher Revolutionen*. Frankfurt am Main: Suhrkamp, 2. Aufl.
- [14] Mahr, Bernd (1998): Gedanken zu einer neuen StuPO. *i-coup. Zeitschrift für Informatik und Lehre*, 3:5–6.
- [15] Weir, Peter, Regie (1998): *The Truman Show*. Spielfilm. USA: Paramount. Mit Jim Carrey, Ed Harris, Laura Linney u.a.
- [16] Wulf, W. A. (1979): A Case Against the GOTO. In: *Classics in Software Engineering*, Hg. Edward Nash Yourdon, New York: Yourdon, S. 85–98.