

Service Didactics / Dienstleistungsdidaktik

Volker Claus

Institute of Formal Methods in Computer Science
Faculty 5, University of Stuttgart
Universitätsstraße 38, D-70569 Stuttgart, Germany
claus@informatik.uni-stuttgart.deAbteilung

Abstract: This paper asks for a faster lead-in to Computing Science for the growing number of students who have to learn information processing as a complementary science or for a foundation degree (UK). Nowadays the human lifespan limits more and more the fundamentals which are necessary for the respective profession. Therefore the skills and the essential knowledge must be restricted to the principal subject, and all other knowledge has to be taught without deep explanation. Complementary sciences and even areas of the major subject will be trained by "drill-and-practice" but in such a way that a future insight in the scientific principles always remains possible and that more topics as today are to be dealt with. This methodology of teaching and learning is characterised by:

- to convey extensive knowledge and skill (using recipes for training),
- to follow certain paradigms of the major subject,
- to train the operation and the use of systems, and to explain their prospects and risks,
- to teach some underlying theoretical foundations and rules (in a reduced manner).

These methods are called "basic recipe" (Grundrezeptik). It is a central part of the "service didactics" which have to be developed for optimizing the efficient, quick and serious usage of a science in another context. Service didactics must elaborate a minimal but stable scaffolding of the field, design excellent supervised courses, develop models for teaching, evaluations and assessments etc. An important technique will be the "virtual laboratory" for intensive learning in limited time. Each laboratory will be built up and updated by scientists from different faculties. In future the courses of more and more studies will be oriented towards this basic recipe. Computing / Computer Science seems to be a good candidate for investigations and the European Bologna process with its reordering of the educational system can immediately be used for many experiments.

1 Grundrezeptik für Dienstleistungen

1.1 Ausgangslage. Menschen scheinen nach Ansicht der Bildungspolitik mit dem Schulende weitgehend mit den notwendigen Grundfertigkeiten und Arbeitsweisen vertraut zu sein und benötigen vor allem allgemeine Befähigungen und berufsbezogenes Wissen. Stattdessen setzen die deutschen Universitäten jedoch auf grundlagenorientiertes Wissen (fälschlich oft "theorieorientiert" genannt) und ver(sch)wenden hierbei viel Ausbildungszeit für Inhalte, die für den künftigen Beruf weniger relevant sind. Daher müssen neue Qualifikationen und praxisbezogenes Vorgehen festgelegt werden, die von den Erstsemestern sicher begierig aufgegriffen würden und zugleich positive Nebeneffekte wie beispielsweise Studienzeitverkürzung und Aufbau eines Systems für lebenslanges Lernen hätten.

Die Realität im Fach Informatik sieht im ersten Studienjahr an deutschen Universitäten meist wie folgt aus: Orientierungslosigkeit, Konsumhaltung, Abheftmentalität und die Annahme, im Wesentlichen werde die reine Anwesenheit in Veranstaltungen – wie seinerzeit das Abitur – auch das Diplom oder den Bachelor beschern, dominieren bei mehr als der Hälfte der Erstsemester. Um hiergegen anzugehen und zugleich das oft sehr unterschiedliche Vorwissen auszugleichen, konzentrieren sich die Universitäten im Grundstudium auf die Vermittlung struktureller Einsichten und grundlegender Prinzipien. Da zu wenig Lehrpersonal vorhanden ist, erfolgt dies in Massenveranstaltungen. Dies mag für motivierte Hauptfachstudierende kein Hindernis sein, sofern ein regelmäßiger Übungsbetrieb angeboten wird. Was geschieht jedoch mit den noch nicht motivierten Studierenden und mit denjenigen, die Informatik nur als Hilfswissenschaft benötigen? Es nützt wenig, diese Studierenden mit abstrakten und formalen Konzepten zu konfrontieren, die sie (oft mangels mathematischer Vorkenntnisse) kaum verstehen und die daher keinen Bezug zum Einsatz der Informatik zu haben scheinen. Für diese Studierenden müssten vielmehr betreuungsintensive, zielorientierte und arbeitsaufwändige Kurse angeboten werden, um Methoden, Arbeitstechniken und Vorgehensmuster in den Erstsemestern herauszubilden und die "realistischen" Einsatzmöglichkeiten der Informatik zu beleuchten. Eigentlich wäre ein solches Angebot eine Chance für jedes Gebiet, seine Bedeutung den Studierenden klar zu machen. Stattdessen möchte jedoch jedes noch so nebensächliche Fach einen möglichst tiefen Einblick in seine Methoden und Denkweisen vermittelt sehen und lehrt daher Inhalte, die weit über den Charakter einer Hilfswissenschaft hinausgehen.

Warum ist dies oft so? Warum nimmt die Universität die Realität bei ihren Anfänger(inne)n kaum zur Kenntnis? Dabei gibt es Vorbilder, z.B. die "Mathematik für Wirtschaftswissenschaften". Hier werden Darstellungen und einfache Vorgehensweisen besprochen und niemand versucht, die Hörschaft mit der Wissenschaft Mathematik vertraut zu machen. Dagegen lagert über Informatik-Veranstaltungen, die von Informatiker(innen) für fremde Studiengänge angeboten werden, stets ein grundlagenorientierter Geist, der a) nicht rüberkommt, b) nicht richtig geprüft werden kann und c) die Zeit für den sinnvollen Einsatz der Methoden stiehlt. Es ist dieser erhobene Zeigefinger ("Wir müssen Ihnen hier auch etwas beibringen, was uns außerordentlich zuwider ist und nur die Mittelalterlichkeit heutiger Anwendungen beweist."), der so treffend von Wilhelm Busch im Lehrer Lämpel gezeichnet wurde und den wir über Bord werfen müssen.

1.2 Folgerung: Wir brauchen also eine doppelte Didaktik:

- a) eine Didaktik für den Informatik-Nachwuchs (damit befassen wir uns hier nicht),
- b) eine Didaktik für andere Studiengänge, die die Informatik als Hilfswissenschaft dringend benötigen.

Zunächst wird man versucht sein, die Didaktik b) als Verkümmern der Didaktik a) aufzufassen und deren Entwicklung daher dem zugehörigen Personenkreis zu übertragen. Das wäre grundfalsch und würde nichts Neues hervorbringen. Vielmehr muss hier ohne missionarischen Eifer eine *Dienstleistungsdidaktik* auf hohem und effizientem Niveau kreiert werden, die die Einsatzbeispiele (und diese prägen die Vorstellung der Studierenden von der Informatik!) aus dem Hauptfach bezieht, die dortige Methodik stets im Blick behält und in (Labor-) Anwendungen die beiden Wissenschaftsgebiete zusammenführt. Andererseits dürfen die Stringenz und der innere Aufbau der Informatik nicht verbogen werden, sondern die grundsätzliche Struktur und die fundamentalen Ideen müssen formal korrekt in einem begleitenden Hintergrundprogramm mitgeführt werden, wobei die Betonung auf "Hintergrund" liegt. Diese noch im Detail zu entwickelnde und zu fundierende Mischung aus rezeptartiger Vermittlung, Vertiefung an Anwendungsszenarien, Herausarbeitung der Nützlichkeit und des konstruktiven Einsatzes sowie der Mitvermittlung eines wissenschaftlich sauberen Grundgerüsts im Hintergrund wird hier "Grundrezeptik" genannt. (Den zugehörigen wissenschaftlichen Überbau kann man als Grundrezeptologie bezeichnen.)

1.3 Hinweis zu den Begriffen: *Rezeptik* und *Rezeptologie* sind keine gängigen Wörter. Rezeptik kann die Vermittlung von Wissen und Verhaltensweisen mit Hilfe eines schematischen, rezeptartigen Vorgehens, Rezeptologie die Methodik der Entwicklung und des Einsatzes solcher rezeptartiger Vermittlungstechniken bezeichnen. Während das Wort "Rezeptik" bisher ungebräuchlich ist, wird das Wort Rezeptologie hin und wieder für die schematische Wiedergabe von unverständlichen Aussagen verwendet, also im Sinne eines auswendig gelernten Plapperns. (Rezeptik ist nicht zu verwechseln mit Rezeptorik; letztere bezeichnet das Vermögen zur Aufnahme von Informationen über Rezeptoren, kann aber auch für ein System von Rezeptoren, z.B. in der Haut oder bei Überwachungssystemen, stehen.)

1.4 Intelligenzzeugcharakter: Wir können erworbenes Wissen dazu nutzen, unser Selbstwertgefühl und unsere Vorstellungen von der Welt zu vervollkommen - gänzlich ohne irgendwelchen wirtschaftlichen oder persönlichen Nutzen. Gerade weil die "Dinge als solche" innere Zusammenhänge besitzen und erforschbaren Gesetzmäßigkeiten genügen, stehen diese zweckfreien Kenntnisse im Zentrum der wissenschaftlichen Ausbildung, und dies muss auch weiterhin für das jeweilige Hauptfach die zentrale Richtschnur sein. Zum Nutzen der Menschheit setzen wir das Wissen um in Produkte, Produktionsverfahren, Abläufe oder Dienstleistungen und müssen also das Zweckfreie um das Nützliche ergänzen, wobei durch die wachsende Interdisziplinarität das Wissen neben dem Verständnis und den Methoden des Einsatzes um die weitere Dimension der Fächer wächst. Mit der Wissensexplosion in den einzelnen Fachgebieten, mit den immer neuen Anwendungsbereichen und mit den diversen grundlegenden Einsichten lässt sich eine umfassende Bildung an den Universitäten aber nicht mehr in einem begrenzten Zeitraum von, sagen wir, fünf Jahren realisieren. Es muss daher eine Konzentration auf wenige Hauptgebiete erfolgen, und alles, was aus anderen Wissenschaften kommt, ist den Studierenden wie ein Werkzeugkasten zur Verfügung zu stellen, der aber nicht ungeordnet sein darf, sondern eine logische Struktur besitzt. Das heißt: Ein ständig wachsender Teil des Wissens und Könnens muss wie in einem Intensivtraining eingeübt werden, ein wenig nach dem Prinzip "quick and dirty", wobei es Aufgabe der Grundrezeptik ist, das "dirty" trotz "quick" so seriös wie möglich zu gestalten, schon allein um ein lebenslanges Lernen (LLL) nicht zu behindern. Denn ebenso wie Informatiker mit den Anwendern später in Teams reden und daher deren Sprache verstehen müssen, so müssen die Anwender gewisse Grundideen der informatischen Lösungen begreifen.

Dieser Werkzeugkasten hat mit dem klassischen Handwerk nichts mehr zu tun. Er bewegt sich vollständig im Virtuellen, im Geistigen. Er dient als Intelligenzverstärker, und informatisches Werkzeug wird von uns nun Intelligenzzeug genannt. Der Intelligenzzeugkasten muss so umfangreich ausgestattet sein, dass viele in der Zukunft auftretende Probleme hierdurch gelöst werden können. In den verschiedenen Hochschulen wird man diesen Zukunftszeitraum unterschiedlich lang definieren, wobei zugleich Maßnahmen entwickelt werden müssen, um alte Intelligenzzeugkästen regelmäßig zu aktualisieren und um Neues zu ergänzen (z.B. durch Reengineering). Trendmäßig müssen und können die künftigen Absolventen dann wesentlich umfangreichere Gebiete als heute überschauen. Das vertiefte Wissen wird seinen Platz im Gehirn der Fachleute mit immer mehr unscharfem Wissen über einen sich ständig vergrößernden Intelligenzzeugkasten teilen. Als Folge wird das Zweckfreie, das wir heute zu den Garanten des Kreativitätszuwachses zählen, bezüglich der Hilfswissenschaften ständig an Boden verlieren (Ausnahmen bestätigen auch hier die Regel) bzw. auf sehr hohem Niveau nur noch wenigen vorbehalten bleiben. Die Konsequenz für die Ausbildung in der Informatik wird es sein, die Studierenden zu befähigen, Intelligenzzeugkästen zu ihrem Alltagsinstrumentarium zu machen, *aber* hierbei die logische Struktur, die in den Kästen steckt, und den sinnvollen Einsatz mitzuvermitteln.

1.5 Hinweis auf *eine mögliche* Realisierung: Für die Lehre müssen nun die Inhalte, die Methoden, die Intelligenzzeugkästen, die Lehrveranstaltungsformen, die virtuellen Labore und die Prüfungsformen festgelegt werden. Die Inhalte müssen gemeinsam mit den Hauptfach-Experten erarbeitet und für die Labore aufbereitet werden, und sie sind in Kursen zu lehren, die anfangs sehr betreuungsintensiv sind, aber im Laufe des Semesters immer mehr Selbstständigkeit von den Teilnehmern fordern. Ein typischer vierstündiger Kurs über "Datenstrukturen", dessen Inhalte heute in einer 4-stündigen Vorlesung mit 2 Stunden Übungen gelehrt werden, könnte dann eine höchstens einstündige Vorlesung, die den Orientierungsrahmen bildet, ein hiermit eng verzahntes einstündiges Praktikum, in dem neue Inhalte in Kleinstgruppen durchdacht und erprobt werden, ein mindestens einstündiges Labor sowie eine einstündige Übung, in der die Lösungen von individuell zu bearbeitenden Übungsaufgaben besprochen werden, umfassen. Ergänzt wird der Kurs um einzelne Stunden zur Bedienung des benutzten Intelligenzzeugkastens.

Entscheidet man sich für eine derartige Kursstruktur, so könnte man den Kurs an einem Praktikum und Labor in den Naturwissenschaften orientieren: Er findet in einem Computer-Raum statt; ein oder zwei Studierende sitzen an jedem Computer-Arbeitsplatz; in einer Praktikumsanleitung werden die zu bearbeitenden Aufgaben beschrieben; im Labor werden die Szenarien illustriert und vertieft; zu den Aufgaben und ihren Lösungen werden Praktikumsberichte geschrieben; die Aufgaben betonen den Einsatz des Intelligenzzeugkastens, für den der Computer-Arbeitsplatz benötigt wird; das Praktikum wird unter Anleitung in der Universität durchgeführt (und nicht zuhause!); durch die Diskussionen am Arbeitsplatz finden jederzeit Wissensabprüfungen der Studierenden durch die Betreuer statt. Die Übungen beziehen sich in der Regel auf bereits bearbeitete Praktikumsaufgaben und betonen die Formalismen und übergreifenden Prinzipien. Man könnte auch ein gemeinsames Diskussionsforum einrichten. Die Erstellung des Kurses und die jährliche Aktualisierung sind aufwändig, die Abwicklung sollte durch Tutoren oder Studenten aus höheren Semestern erfolgen. Der Betreuungs- und Finanzaufwand würde sich gegenüber der heutigen Situation deutlich erhöhen, weil Vorlesungsanteile durch betreute Übungen und Praktika ersetzt werden.

2. Ein Beispiel (Details siehe Anhang)

2.1 Was ist daran neu? Wie unterscheidet sich dies von Trainingskursen, die mit Trial-and-Error-Vorgehen oder drill-and-practice-Techniken heute in der Praxis alltäglich sind?

Der Unterschied liegt in zwei Bereichen: Zum einen wollen wir den sinnvollen Einsatz der Informatik im Hauptfach lehren, was über geeignete Szenarien in einem virtuellen Labor gesteuert wird. Es darf also nicht nur eine umfangreiche Klassenbibliothek vorgestellt und exemplarisch erläutert werden, sondern es sind die Anforderungen an die Informatik aus dem Hauptfach heraus zu formulieren und an typischen Beispielen mit dem Intelligenzzeug umzusetzen. Zum anderen müssen die wichtigsten Prinzipien der Informatik gelehrt werden, d.h., es ist ein fachlich korrektes, aber stark abgemagertes Strukturgerüst der Informatik, mit dem sich möglichst viele Konzepte und Werkzeuge erklären lassen, zu entwickeln. Und hier kommt die (schon klassische?) Schuldidaktik der Informatik ins Spiel; denn ähnliche Überlegungen gelten auch für den Schulunterricht. Allerdings muss dieses auf universitärem Niveau angesiedelt sein.

2.2 AVL-Bäume: Betrachten wir ein Beispiel. In der Stuttgarter Grundvorlesung Informatik stellen wir zurzeit (= im Juni 2004) den Studierenden höhenbalancierte Bäume mit den zugehörigen Darstellungen, Eigenschaften, Algorithmen und programmiersprachlichen Realisierungen vor. Deren Mechanismen und Abläufe empfinden viele Hörer, die nicht Informatik oder Softwaretechnik studieren, als spannend und auch als überraschend. Allerdings bezieht sich dies nur auf die 30% der Vorlesung, in denen der Aufbau und die Manipulationen an Beispielen demonstriert werden. Die übrigen 70% sind Varianten, Analyse der Rotationen, die penible Neuberechnung der Balancen, Fibonacci-bäume, semantisch korrekte Umsetzungen in Prozeduren und Beweise über die maximale oder die durchschnittliche Tiefe solcher Bäume. In der Summe ist dieses Wissen ein "Wert an sich", durch den Wissensstrukturen und Erkenntnisse, also langlebige Modelle aufgebaut werden, die die Hauptfachstudierenden befähigen, in ihrem weiteren Leben Speicherungs- und Verarbeitungsmethoden einzusetzen, zu bewerten und weiterzuentwickeln, was aber die Nebenfach-Informatiker nicht sonderlich reizt. Was hier genau behandelt wird und wie eine Veränderung aussehen könnte, ist im Anhang aufgelistet.

2.3 Die Praxis glaubt jedoch nicht an den Wert dieser Inhalte; denn für konkrete Anwendungen (etwa im Zusammenhang mit Datenbanken) sind höhenbalancierte Strukturen durch vordefinierte Klassen längst für jeden zugreifbar, ohne dass irgendjemand eine Ahnung von der Theorie und den Formalismen oder von den geheim gehaltenen Realisierungen haben müsste. Kein Anwender muss diese Strukturen kennen - und kann sie dennoch für seine Probleme nutzen. Aus Sicht des Hauptfachs sind obige 70% der Lehrinhalte weitgehend Zeitverschwendung; von ihnen sind höchstens Prinzipien und einige "Parameter" (Zeitverhalten, Speicher-Overhead, Initialisierungen, ...) interessant, um die existierenden Intelligenzzeuge aus den längst gründlich durchstöberten Intelligenzzeugkästen (Klassen, Bibliotheksprogramme, aus dem Netz abrufbare Methoden, Vermittlungssysteme, eigene Ergänzungen usw.) bereits gut einsetzen zu können.

Dennoch vermitteln wir *allen* Studierenden dieses Basiswissen, das aus Sicht der Praxis nur für die Theoretiker und für die Implementierer von Bibliotheksprogrammen und objektorientierten Klassen wichtig ist. Wir tun dies aus der Überzeugung heraus, ohne Hintergrundwissen sei eine Bewertung und damit ein sinnvoller Einsatz kaum möglich. Aber: Stimmt diese Annahme auch für die Nicht-Informatiker? Und wenn sie stimmt, ist sie dann nicht irrelevant angesichts des Zeitarguments? Die betroffenen Studierenden sehen dies deutlich pragmatischer: Sie erwarten einen Überblick mit einigen Eigenschaften und ein Verständnis für gewisse Einsatz-Szenarien, und mehrheitlich möchten sie die Verfahren keinesfalls selbst implementieren, sondern aus erprobten Standardisierungen übernehmen. Warum oktroyieren wir ihnen also unsere Weisheiten auf? Warum bieten wir den Nicht-Informatikern keine ("trotzdem fundierte") Praxis- oder Einsatz-Version, also keine Dienstleistungs-Variante der Informatik an, wo wir gleichzeitig behaupten, unsere Informatik sei bestens anwendbar?

3. Die Bedeutung einer Dienstleistungsdidaktik

3.1 Didaktik wird oft nur unter dem Aspekt der Vermittlung von Inhalten und Methoden an Lehramtsstudierende gesehen. Diese verkürzte Sicht dient gerne als Begründung dafür, warum eine Fachdidaktik nichts in einer fachwissenschaftlich orientierten Fakultät zu suchen habe. Aber die Didaktik hat eine viel tiefer liegende Auswirkung: Sie klärt die Struktur eines Faches und ist die eigentliche Begründung für das Humboldtsche Ideal der Einheit von Forschung und Lehre.

Wer forscht, soll auch lehren, und lehren sollte nur, wer auch in der Forschung engagiert ist. Mit dieser These begründen die Universitäten ihre (sicherlich sinnvollen) Privilegien wie das Promotionsrecht oder das gegenüber den Schulen deutlich reduzierte Lehrdeputat. Aber stimmt diese These von der Einheit von Forschung und Lehre? In Max-Planck-Instituten, in der Fraunhofer-Gesellschaft und in den industriellen Forschungsinstituten wird kaum systematisch gelehrt (am liebsten werden Diplomarbeiten betreut) und auch die DFG-Mitarbeiter und die Mitglieder von Graduiertenkollegs sollen möglichst nicht lehren.

Die These der Einheit von Forschung und Lehre gilt bei näherer Betrachtung nur für Professor(inn)en an Universitäten, wo sie den hochqualifizierten wissenschaftlichen Nachwuchs ausbilden sollen. Dies bedarf eines genauen Überblicks über den Stand der Erkenntnisse; daher sind hier die Forscher gefordert. Das gelehrt Wissen wiederum darf kein Sammelsurium sein, da es dann nicht in vernünftiger Zeit vermittelt werden kann. Es bedarf also einer Systematik des Fachs, einer Herausarbeitung der Prinzipien, einer Analyse der grundlegenden Methoden, einer Beschreibung von Problemlöseverfahren und der Synthese usw. Alle diese Zielsetzungen gehören aber zu den Aufgaben einer Didaktik. Daher muss die Didaktik als zentrale Komponente eines universitären Faches angesehen werden. (Warum die Realität anders aussieht, wäre ein eigener Artikel.)

3.2 Einschub: Wissenschaft und die Bedeutung ihrer Didaktik. Damit ein Gebiet zu einer Wissenschaft werden kann, muss es nach meiner Vorstellung mindestens vier Erfordernisse erfüllen:

- Es muss neue Inhalte und eine eigene Methodik hervorbringen,
- es muss eine eigene unverständliche Sprache herausgebildet haben,
- es muss eine eigene Didaktik besitzen,
- es muss Reflexion (und möglichst auch Selbstironie) entwickelt haben.

(Ein Bezug zur Realität und eine Nützlichkeit liegen in der Regel ebenfalls vor, jedoch sind auch rein zweckfreie Wissenschaften denkbar.)

Informatik hat bereits in den 1970er Jahren den Sprung zur eigenständigen Wissenschaft geschafft, selbst wenn sich die Didaktik und die selbstkritischen Töne noch auf recht wenige Wissenschaftler/innen beschränkten. Der mit jeder Wissenschaft verbundene Aspekt der "Unverständlichkeit" hat sich in der Informatik und ihren Anwendungsgebieten gut entwickelt. Um diese Neigung zur Isolation auszugleichen, steht oben die Forderung nach einer eigenen Didaktik, die diverse zentrale Aufgaben zu erfüllen hat:

- Klärung der Grundbegriffe (z.B. fundamentale Ideen) und Zusammenstellung typischer und eingängiger Beispiele (Paradigmen),
- systematischer Aufbau des Faches und Ordnung im Methoden-Bereich, Ausstrahlung auf andere Fächer,
- Bewertung der Inhalte und Teilgebiete (bzgl. Schwierigkeit, Nützlichkeit, Anwendung, Auswirkung, ...),
- Verständlich-Machen von Schwierigem und Unverständlichem,
- Konzepte zur Vermittlung und Prüfung von Inhalten, Methoden und Techniken und Erarbeitung einer Methodik des Unterrichtens (unterschiedlich für verschiedene Altersstufen und Zielgruppen),
- Erstellung von Curricula und Rahmenempfehlungen, Zielorientierung der Ausbildung,
- optimal wäre ein durchgehendes Gesamtkonzept "vom Kindergarten bis zur Pensionierung",
- hierzu zählen auch eine Dienstleistungsdidaktik, um der erforderlichen Interdisziplinarität und Solidarität (im Sinne von wechselseitiger Hilfe unter den Wissenschaften) Rechnung zu tragen, und eine Weiterbildungsdidaktik (basierend auf einer Erwachsenenpädagogik), um auf den rasanten Austausch von Wissen zu reagieren.

3.3 Während die hierdurch umrissene Didaktik eine gewisse Unabhängigkeit beanspruchen kann, wird sie in der Regel an Universitäten auf die Betreuung von Lehramtsstudiengängen ausgerichtet, wodurch sie eine konkrete, für die Gesellschaft wichtige Rolle erhält. Generell ist eine "zweckfrei angelegte Didaktik" unüblich, und so stehen auch für den Bereich der Dienstleistungsdidaktik keine universitären Institute a priori zur Verfügung. Dennoch sprechen viele *Gründe dafür, Grundrezeptik zu betreiben*:

- Ständig wachsende Erkenntnisse bei konstanter oder sinkender Ausbildungszeit erzwingen eine deutlich verbesserte Aufbereitung bei gleicher Reduzierung auf Wesentliches und Nützlich.

- Die interdisziplinäre Zusammenarbeit erfordert reduzierte Grundgerüste jedes Faches als schnellen Einstieg für den wechselseitigen Dialog.
- Der Einsatz der Informatik bedingt rezeptartige Verhaltensweisen und Bedienungsanleitungen für die immer unverständlicher werdenden Intelligenzzeuge.
- Typische Beispiele und Szenarien dienen hierbei der Orientierung und Bewertung.
- In einer immer mehr mit Hilfsmitteln sich füllenden Welt bedarf es guter übergreifender Prinzipien und Richtlinien, nach denen sich möglichst viele Instrumentarien zu richten haben.
- Die entwickelten Kurse sind eine ideale Basis für Weiterbildungsmaßnahmen und LLL. Ein umfassendes Weiterbildungssystem ist unverzichtbar, wenn möglichst alle Beschäftigten (wie politisch beschlossen) bis zum 67. Lebensjahr produktiv arbeiten sollen.
- Die notwendige Rückkopplung der Anwendung auf die Informatik würde zugleich gestärkt.
- Schließlich können und sollen hiermit auch allgemeine Arbeitstechniken vermittelt werden; denn mangelhafte Arbeitstechniken sind ein Hauptgrund für schlechte Leistungen in den ersten Semestern.

3.4 Allein schon mit dem Argument der begrenzten Ausbildungszeit bei ständig wachsenden und zu lehrenden Inhalten und mit der zunehmenden Interdisziplinarität müssen neue Veranstaltungen - basierend auf einer Grundrezeptik - in Zukunft verstärkt entwickelt und angeboten werden. Die durch das 12-jährige Gymnasium weiter absinkenden Eingangqualifikationen der Studierenden (trotz Eignungsfeststellungsverfahren wird dies unvermeidbar sein) werden den Druck auf die Hilfswissenschaften, schnell und einsatzorientiert auszubilden, verstärken. Den neuen Formen des Schnelleinstiegs in ein Gebiet, aber zusammen mit einer grundlagenorientierte Komponente zur Sicherung des LLL, also der Grundrezeptik, prognostizieren wir daher eine ständig wachsende Bedeutung. Wir unterstützen jedoch nicht den gedankenlosen Schnelleinstieg, sondern eine Ausbildung, die auf wissenschaftlicher Arbeitsmethodik aufbaut und diese auch vermittelt, die durch virtuelle Laborkomponenten den sinnvollen Einsatz präsentiert und die es den Teilnehmer(inne)n erlaubt, später zum einen mit den Hauptfachstudierenden der Informatik in einen Dialog treten und zum anderen sogar relativ leicht in das Informatikstudium wechseln zu können.

Dies ist zugleich (und vorwiegend!) eine didaktische Herausforderung: Wie kann man gleichzeitig mit dem rezeptartig vorgetragenen Stoff den intellektuellen Gehalt und möglichst viele Ideen in die Köpfe transportieren? Eine Hauptschwierigkeit wird vermutlich in der überfachlichen Zusammenarbeit liegen: Professor(inn)en, die wechselweise von den Gebieten der anderen nur eine begrenzte Kenntnis haben, müssen ein geschlossenes Kurskonzept mit neuen Präsentations- und Übungsmethoden planen und umsetzen. Vielleicht ist dies aber auch eine große Chance, die Interdisziplinarität zu fördern, manche Berührungspunkte abzubauen und den Stoff für die Anwender deutlich attraktiver zu gestalten.

3.5 Der laufende "Bologna"-Prozess (zusammen mit der 12-jährigen Schulzeit) wird nach unserer Überzeugung diesen Trend ebenfalls stärken. Wenn der künftige Bachelor – was sich abzeichnet – nicht ein um Zusatzqualifikationen erweitertes Vordiplom, sondern ein um berufspraktische Kenntnisse erweitertes Abitur wird, so müssen alle Hilfsfächer kompakt studierbar werden und es sind die wichtigsten Inhalte besser zu vermitteln, um das Niveau nicht ganz wegbrechen zu lassen. Zugleich müssen die Betreuung deutlich erweitert und die Langlebigkeit möglichst vieler Inhalte gewahrt werden. Daher ist jetzt auch der geeignete Zeitpunkt, um eine Informatik-Grundrezeptik im Detail zu konzipieren und voran zu treiben.

4. Die Umsetzung in den Universitäten?

4.1 Das erste Problem: kein Geld. Der Trend - so unsere Überzeugung - wird schon rasch zur Forderung nach einer Grundrezeptik führen. Da kein Geld für neue Initiativen vorhanden ist, wird man deshalb von *allen* Universitätsmitgliedern verlangen, sich hier "kostenneutral" einzuarbeiten und die geforderten Leistungen ohne professionelle Kenntnisse irgendwie zu erbringen. Vor derartigen Entwicklungen ist nur dringend zu warnen, da sie im Endeffekt deutlich teurer als ein systematisches Vorgehen sind oder sich als kontraproduktiv erweisen. Als Beispiele seien aus der Informatik die folgenden Absätze a und b angeführt.

a. Bessere Lehre. Um den Erfolg der Ausbildung zu erhöhen, wird seit vielen Jahren die "Stärkung der Lehre" gefordert. Gemeint sind hiermit zwei Dinge: Zum einen sollen die Hochschulen der Lehre (gegenüber der Forschung) mehr Zeit widmen und zum anderen soll die Lehre besser werden. Letzteres, d.h. *bessere Lehre*, zielt vordergründig auf die bessere Aufbereitung und Darstellung der zu vermittelnden Inhalte, wodurch man sich leichteres, schnelleres, einsichtsvolleres usw. Lernen verspricht. In der Praxis heißt bessere Lehre aber keineswegs „besseres Lernen“, besseres Verständnis oder kürzere Studienzeiten. Hiergegen spricht vor allem die natürliche Trägheit der Menschen: Je besser aufbereitet ein Stoff dargeboten wird, um so geringer ist der Anreiz, sich noch mit ihm zu beschäftigen und umso eher verharrt der Lernende in der trügerischen Vorstellung, er habe den Stoff schon beim ersten Zuhören vollständig verstanden. So erwies sich "bessere Lehre" oft als doppelt belastender Faktor: deutlich höherer Erstellungsaufwand bei den Unterrichtsmaterialien (in der Informatik ist der Faktor fünf keine Seltenheit) und deutlich höherer Betreuungsaufwand, um die Studierenden von der inhaltlichen Tiefe und den innewohnenden Ideen wirklich zu überzeugen. Das bisher meist kontraproduktive Ergebnis: immer weniger Zeit für Forschungsaufgaben mit der zu befürchtenden negativen Langzeitwirkung für die Qualität der Ausbildung.

b. Informatik als Schulfach. Da in den 1980-er Jahren keine Lehramtsstudiengänge eingerichtet und daher keine Lehrkräfte für Informatik ausgebildet wurden, wurde das Schulfach Informatik fachfremden Lehrern übertragen. So kamen viele Erstsemester mit völlig falschen Vorstellungen über die Informatik an die Universitäten, was mit zu den sehr hohen Abbruchquoten in diesem Studienfach führte und noch führt. Man hatte auf die Selbstorganisation der Schulen gesetzt, aber den Lehrkräften keine Zeit gelassen, die Grundlagen der Informatik tatsächlich zu erwerben. Diese Fehlentwicklung hat der Volkswirtschaft vermutlich einen weitaus höheren Schaden zugefügt, als durch Lehramtsstudiengänge Kosten entstanden wären. Übrigens: Eine Auswirkung dieses politischen Versäumnisses besteht darin, dass heute die Universitäten für die Abbruchquoten verantwortlich gemacht werden.

Wegen solcher Erfahrungen sind die Universitäten gut beraten, zuerst finanzielle Garantien zu fordern, bevor sie auf irgendeinem Gebiet aktiv werden. Leider bezieht sich dies auch auf alle sinnvollen und erforderlichen Maßnahmen, wie zum Beispiel die Entwicklung und Analyse einer Grundrezeptik. Man kann also davon ausgehen, dass noch so gute Argumente keine Reaktionen auslösen werden.

4.2 Das zweite Problem: die Kapazitäten. Vermutlich gibt es viele Informatiker(innen) in den Universitäten, die eine Dienstleistungsdidaktik und entsprechende Veranstaltungen unterstützen würden. Die Vergangenheit lehrt aber, dass sinnvolle Dinge (wie beispielsweise die Informatikdidaktik selbst) Jahrzehnte für ihre Einrichtung benötigen.

Solche Verzögerungen oder Verhinderungen beruhen meist auf Kapazitätsargumenten. Bei gedeckelten Haushalten kann Neues nur auf Kosten von Bestehendem installiert werden. Es müssten Lehrstühle umgewidmet und kapazitatativ relevante Veranstaltungen aus dem Kerngebiet der Informatik zugunsten der neuen (vielleicht nur im Wahlbereich angesiedelten) Veranstaltungen aufgegeben werden. Kein Fakultätsrat wird solchen Plänen in der Regel zustimmen (schon aus strukturellen Gründen), es sei denn, neue Stellen würden hierfür *von außen* bereitgestellt oder bereits gestrichene Stellen könnten hierdurch gerettet werden.

Zugleich spielt das Lehrdeputat eine zentrale Rolle. Bei 8 bzw. 9 Stunden pro Woche bleibt wenig Zeit, um innovative Neuorientierungen in Angriff zu nehmen oder gar sinnvoll umzusetzen. Man müsste folglich versuchen, dies durch Examensarbeiten oder gezielte Ringvorlesungen voranzutreiben, was aber auch große Verzögerungen mit sich bringen wird.

4.3 Das dritte Problem: Auch inhaltlich dominieren die Bedenken. Es wird argumentiert werden, dass durch solche Kurse falsche Denkweisen gefördert, kommende Entwicklungen der Informatik verpasst und die derart Ausgebildeten der raschen Verkalkung preisgegeben werden. Für viele Informatiker(innen) kommen solche nützlichen, aber anspruchslosen Veranstaltungen deshalb nicht in Frage. Zusätzlich wird ins Feld geführt, dass andere Fächer der Informatik ja auch nicht entgegen kommen: Keine betriebswirtschaftliche oder ingenieurwissenschaftliche Veranstaltung wird auf die Bedürfnisse der Informatik zugeschnitten. Technische Universitäten sind darüber hinaus besonders zögerlich, wenn mit neu zu schaffenden Professuren nicht unmittelbar die engere Fachwissenschaft angemessen abgerundet wird.

Man darf auch das "Ver-Bachelorn" nicht vergessen, genannt nach der Aussicht, dass die, die in Zukunft in der Bachelorausbildung tätig sind, deutlich wenig qualifiziert sein dürfen als die, die in Masterstudiengängen mitwirken. Natürlich wird jede(r) die Befürchtung hegen, automatisch an wissenschaftlichem Ansehen zu verlieren, wenn vorwiegend eine Grundrezeptik-Lehre erbracht wird. Vermutlich werden diese Personen finanziell und rechtlich sogar noch schlechter als die im Bachelorstudiengang Tätigen eingestuft werden.

4.4 Es spricht zwar alles für Dienstleistungskurse, aber auch alles gegen eine geordnete Förderung der Grundrezeptik durch die Universitäten. Also wird man zunächst nichts tun. Das soll aber die Didaktik nicht daran hindern, selbst Untersuchungen durchzuführen, sofern sie entsprechenden Lehr- und Lernmethodiken überhaupt etwas abgewinnen kann.

Das Ziel solcher Untersuchungen wird die Konzipierung und Skizzierung einer Informatik-Grundrezeptik sein. Welche Bereiche sollen die neuen Veranstaltungen und virtuellen Labore abdecken und wie kann man sie qualitativ hochwertig installieren? Nach welchem Schema soll man sie aufbauen, wie werden sie mit dem Hauptfach verzahnt und mit welchem Erstellungsaufwand muss gerechnet werden? Wie führt man hier effizient und schnell Prüfungen durch? Welche Rolle spielt die Lehrkraft und welche Anteile lassen sich automatisieren? Wie kann man die Inhalte in Weiterbildungsmaßnahmen, aber vielleicht auch in propädeutische Kurse für Gymnasiasten einfließen lassen?

5. Schlussbemerkung

In der heutigen Diskussion über die Verbesserung der Ausbildung kommen zwei zentrale Begriffe so gut wie gar nicht mehr vor, nämlich *Zweckfreiheit* und *Reifung*. Der künftige Student soll schnell und berufsbezogen sein Studium durchheilen, um rasch eine volkswirtschaftliche Bedeutung zu erlangen. Dies ist zwar aus Sicht der Alterspyramide verständlich, wird aber nach unserer Ansicht den ersehnten Effekt langfristig nicht bewirken.

Wir erleben immer wieder, wie Studierende nach längerem Zögern plötzlich Tritt fassen und sich konzentriert ihrem Studium zuwenden. Wir sehen, wie Menschen Ideen oft nur über Abstraktionen entdecken, die frei von jeder Anwendung sind. Wir haben auch viele Studierende, die Veranstaltungen besuchen, die nichts mit ihren Prüfungen, sondern nur etwas mit ihrem Interesse zu tun haben. Wir wissen, dass man Menschen Muße geben muss, damit sie etwas vollständig durchdringen und damit sie Selbstbewusstsein erwerben. Ein als verloren hingestelltes Jahr kommt dadurch mit vielfachem Nutzen wieder herein. Von diesem Menschentyp benötigt eine Gesellschaft viele Personen.

Grundrezeptik kann daher auch dazu dienen, den Freiraum für die Reifung, die vor allem im Hauptfach stattfinden muss, zu schaffen. Ein solches übergeordnetes Ziel streben wir an, wohl wissend, dass die heutigen Universitäten (im Gegensatz zu den Professor(inn)en?) solchen Argumenten wenig zugänglich sind. Aber wir geben die Hoffnung auf langfristig eintretende Einsicht noch nicht verloren.

Anhang 1: AVL-Bäume (heutige Präsentation)

Aufbau grob, vergleiche 1.5.

a) Vorlesung

Vortragsdauer: Etwa 84 Minuten. Jedem Thema sind unten die erforderlichen Minuten vorangestellt worden. Gut 5 Minuten der Veranstaltung gehen durch Tafelwischen, Beamer-Bedienungen und Herstellen von Ruhe und Aufmerksamkeit verloren.

- 5 Definition AVL-Baum (höhenbalancierter Baum, Spezialfall der Binärbäume)
- 9 Beispiele sowie zwei Beispiele von Binärbäumen, die nur wegen eines Blattes nicht AVL-Bäume sind, sowie umgangssprachliche Erläuterung der Rotationen (Vertikalverschiebung von Knoten)
- 14 Die vier Typen von Rotationen mit den sich ergebenden Balancen
- 8 Programmstück zur Neuberechnung der Balancen, mit etwas Verifikation
- 4 Vorgehen beim Suchen, Einfügen und Löschen
- 10 Das Löschen im Detail mit den Problemen beim Aufwärtssteigen
- 6 Programmiersprachliche Hinweise (auch rekursive Prozedur)
- 7 Einschub (Wiederholung?): Definition und Eigenschaften der Fibonaccizahlen
- 11 Beweis der maximalen Tiefe eines AVL-Baums mit n Knoten; gemessene Tiefen in der Praxis
- 7 Fibonaccibäume
- 3 Höhenbalancierte Bäume, die nicht gewichtsbalanciert sind

b) zugehörige Übungen

In den Übungen werden mehrere konkrete Beispiele durchgesprochen (Studentendatei, Briefmarken und Merkmalsklassifikationen, CD-ROM-Sammlung: Vor-/Nachteile der AVL-Struktur); Sortieren mit AVL-Bäumen und Vergleichsmessungen mit anderen Suchbäumen; die Operationen beim Einfügen und Löschen werden im Einzelnen durchgegangen und programmiersprachlich im Detail ausformuliert; Auswirkungen, wenn beim Löschen stets der in-order-Nachfolger gewählt wird?; Verallgemeinerung auf Bäume, deren Balancen zwischen -2 und 2 (oder $-k$ bis $+k$) liegen. Die Aufgaben werden so ausgewählt, dass (oft mit Ausnahme einer anspruchsvollen oder theoriebetonten Aufgabe) in den Übungsgruppen eine Präsenzzeit von rund 50 Minuten eingehalten werden kann. Die Vorbereitungszeit und die Zeit für die Bearbeitung der AVL-Übungsaufgaben werden für jeden Studierenden mit rund 135 Minuten angesetzt.

c) Materialien hierzu im Netz

Ältere Ansätze sind nachlesbar unter Kapitel 3.2.4 "Einführung in die Informatik II" (Folien 111-140):

<http://www.informatik.uni-stuttgart.de/fmi/fk/lehre/ss04/info2/default.htm>

Übungen stehen in Blatt 8 auf der gleichen Webseite.

Man findet unter dem Stichwort "AVL-Baum" viele Ausarbeitungen dieses Themengebiets im Internet. Die besten Darstellungen stehen allerdings in den einschlägigen Lehrbüchern.

Anhang 2: AVL-Bäume (aus Sicht einer Grundrezeptik, Erprobungsvorschlag)

a) Vorlesung mit integriertem Praktikum (135 Minuten, falls integriert realisiert)

Die Vorlesung findet im Praktikumsraum statt. Es sind höchstens 16 Teilnehmer anwesend. Der Professor hält rund 15 Minuten Vorlesung, dann folgen rund 30 Minuten betreutes Arbeiten am Rechner. Daran schließen sich rund 20 Minuten Vorlesung an, gefolgt von 70 Minuten Arbeiten am Rechner.

a1 Vorbereitung im Praktikumsheft: Rund 6 Seiten über AVL-Bäume und Beispiele sind vorab von jedem Teilnehmer durchzulesen. (Inhalt: Charakteristikum eines AVL-Baums, diverse Beispiele, Einfügevorgang mit zwei Rotationen und der Aufforderung, weitere Rotationen selbst zu entdecken.) Dies wird als bekannt vorausgesetzt. Das Arbeiten mit einer Klassenbibliothek (z.B. in Java) ist ebenfalls bekannt.

a2 Vortrag: 15 Minuten. Das Herstellen von Ruhe und Aufmerksamkeit ist in der Regel hier nicht erforderlich; Tafelwischen und Technikbedienung werden in überlappend zu anderen Teilen durchgeführt. Vor jedes Thema sind die geschätzten Minuten gestellt:

- 2 Definition AVL-Baum (höhenbalancierter binärer Suchbaum)

- 4 Beispiele für Einfügungen mit den zwei fehlenden Rotationen
- 4 Illustration der vier Typen von Rotationen und ihre Auswirkung an Beispielen
- 2 ein gewichtsbalancierter Baum, der nicht höhenbalanciert ist
- 3 Beispiel aus der Anwendung (Studentendatei oder Lexikon oder Lagerbestand oder ...)

a3: Gemeinsame Arbeit der Anwesenden in Zweiergruppen: Umgang mit AVL-Bäumen (30 Minuten).

Die Klasse AVL-Baum wird inspiziert. Es ist eine kleine vorbereitete Datei in einen AVL-Baum einzulesen. Das Ergebnis wird auf Tiefe untersucht. In diesem Baum werden Elemente gelöscht. Es werden größere vorbereitete Dateien eingelesen und einmal in einer linearen Liste, einmal in einem AVL-Baum abgelegt. Suchzeiten und Löschvorgänge werden experimentell verglichen (hierzu müssen vorab in Java Simulationstools geschrieben werden, um die Wirkungen sichtbar zu machen). Die Ergebnisse sind zu beschreiben und zu interpretieren. (Varianten sind ggf. zu konstruieren.)

a4 Vortrag: 20 Minuten

- 12 die Grundoperation für das Löschen und ihre Implementierung
- 3 Präziser Formalismus
- 5 Formulierung einiger Eigenschaften hiermit (max. Tiefe, Laufzeiten, Speicherbedarf)

a5: Gemeinsame Arbeit der Anwesenden in Zweiergruppen (70 Minuten): Modifikation oder Erstellung eines Programms zur Bestimmung der Anzahl der Elemente in einem AVL-Baum und zur Berechnung der maximalen Tiefe. (Falls die Teilnehmer nicht fertig werden: bis zum nächsten Termin austesten). Eventuell noch Tests zum Bereich "Löschen" (hierbei zufällig in-order-Vorgänger und -Nachfolger wählen, Tests tatsächlich durchführen lassen).

Es wird erwartet, dass die Studierenden noch etwa 2 weitere Stunden pro Woche außerhalb der betreuten Zeit am Rechner verbringen.

b) zugehörige Übungen (45 Minuten Präsenzzeit)

In der Regel 2 bis 3 Aufgaben je Übung. Bei AVL erfolgt eine Anwendung auf ein konkretes Problem aus dem Hauptfachbereich der Teilnehmer(innen). Eine weitere Aufgabe dient den Grundlagen (z.B. möglichst dünne AVL-Bäume). Eine dritte Aufgabe könnte sich mit dem Baumsortieren mittels AVL-Bäumen befassen. Die individuelle Vorbereitungszeit und die Zeit für die häusliche Bearbeitung der Übungsaufgaben sind mit rund 90 Minuten anzusetzen. In den Übungen präsentieren die Studierenden ihre Lösungen. Hierfür sind mindestens 30 Minuten der Präsenzzeit vorgesehen. Weitere 15 Minuten können den Aufgaben oder Problemen des Praktikums gewidmet werden.

c) Hinweise

Für die Grundrezeptik sind Vorlesungen ungeeignet. Hier müssen sich kurze vorlesungsartige Teile mit längeren praktischen Bearbeitungen abwechseln; die Hintergründe sollen in den Übungen inhaltlich reflektiert und von den Studierenden vor der Gruppe vorge-tragen werden.

Andere Vorgehensweisen sind denkbar, z.B. eine langfristige Orientierung an einem Anwendungsproblem über mehrere Wochen. Dies setzt umfangreiche Vorbereitungen voraus, die dann aber über mehrere Jahre (etwa 5 Jahre) in leichter Variation wiederholt werden können. Der Kurs sollte über diesen Zeitraum fest bei einem Lehrstuhl angesie-delt sein. Varianten müssen ohnehin eingebaut werden, da elektronisch erstellte Lösun-gen sich sehr schnell über Studentengenerationen vererben.

Es liegen natürlich bereits Erfahrungen aus "Kompaktkursen" vor, die aber den hohen Laboranteil nicht besitzen.

Kapazitatativ ein solcher Kurs als 4-stündige Übung in 15er-Gruppen einzustufen.

Als Gesamtaufwand für die Studierenden setzen wir an:

Besuch der Vorlesung, des Praktikums und der Übungen: insgesamt 3 Zeitstunden.

Weitere Zeiten am Rechner: 2 Zeitstunden.

Arbeit zuhause gemeinsam und/oder allein: insgesamt 3 Zeitstunden.

Gesamtaufwand: 8 Zeitstunden je Woche. Bei 15 Wochen im Semester sind dies ins-gesamt 120 Stunden. Dies entspricht 4 ECTS-Punkten. (Zum Vergleich: Die derzeitige 4V+2Ü-stündige Vorlesung, die allerdings auch mehr Breite besitzt, entspricht 9 ECTS-Punkten. Im Schnitt müsste man eine Reduktion der ECTS-Punktzahl auf 50% bis 75% erreichen können bzw. den vermittelten Stoff entsprechend auf weitere Inhalte des Intel-ligenzeugkastens ausweiten. Vorsicht bei der Umsetzung: Es ist davon auszugehen, dass die jeweiligen Hauptfächer begeistert reagieren werden.)

Für die Realisierung sollte man auch Modelle durchdenken, bei dem Studierende höherer Semester als Mentoren für jüngere Semester agieren.