

**DAS 4-SCHRITTE-MODELL:
GRUNDLAGE FÜR EIN KOMPETENZORIENTIERTES
E-LEARNING**

ABHANDLUNG

zur Erlangung des Titels

DOKTOR DER WISSENSCHAFTEN

der

ETH ZÜRICH

vorgelegt von

LUKAS EMANUEL FÄSSLER

Dipl. phil. nat., Universität Bern

geboren am 07.05.1972

von Appenzell AI

Angenommen auf Antrag von
Prof. Dr. Hans Hinterberger
Prof. Dr. Juraj Hromkovic

Vorwort

Diese Dissertation ist im Rahmen meiner über sechsjährigen Tätigkeit als Assistent und Projektmitarbeiter bei Prof. Hans Hinterberger am Institut für Computational Science an der ETH Zürich hervorgegangen. Ausgangslage war das E-Learning-Projekt „Information Processing for Domain Scientists“, unterstützt vom Fonds zur Förderung innovativer Lehrprojekte (Filep) des Rektors der ETH Zürich, das im Jahr 2000, also während der ersten E-Learning-Welle, gestartet wurde. Die Tatsache, dass wir einen eigenen operativen Unterricht mit jährlich über 600 Studierenden durchführen mussten, hat uns schon früh gezeigt, dass E-Learning wesentlich mehr bedeutet, als sich mit Übertragungsraten, Hardware und Software auseinanderzusetzen. Bei uns stand von allem Anfang an die Frage im Zentrum, wie wir durch den Einsatz des Computers im Unterricht gegenüber traditionellen Unterrichtsmethoden einen Mehrwert generieren und das individuelle Lernen an und mit dem Computer über die Zeitspanne eines ganzen Semesters effektiv unterstützen können. Zusätzlich standen wir vor der Herausforderung, einer nicht technisch orientierten Zielgruppe der Naturwissenschaftlerinnen und Naturwissenschaftler das Fach Informatik näher zu bringen. Nach Abschluss dieses ersten Projekts zeigte sich, dass unser Unterrichtskonzept bei den Studierenden so beliebt war, dass sie in der Nachfolgeveranstaltung, einer Programmiervorlesung, ebenfalls eine solche Computerunterstützung nach dem Vorbild des ersten ICT-Kurses wünschten. Ab 2005 wurden wir zusätzlich beauftragt, den Einführungsunterricht in Informatik des Departements Bau-, Umwelt und Geomatik zu übernehmen, diesmal unter der Leitung von Dr. Bettina Bauer-Messmer. Auf diese Weise ist ein computergestützter Lehrgang in Informatik mit aktuell über 1100 Studierenden entstanden, der vom Departement Informatik als Dienstleistung für die Departemente Agrar- und Lebensmittelwissenschaften, Bau- Umwelt- und Geomatikwissenschaften, Biologie, Chemie und Angewandte Biowissenschaften, Erdwissenschaften und Umweltwissenschaften angeboten wird.

Diese Dissertation soll einen Blick hinter die Kulissen dieses computergestützten Informatikunterrichts ermöglichen. Sie fasst die Grundlagen des in unserer Gruppe aufgebauten und betriebenen multidisziplinären *Educational Engineerings*, das die Entwicklung von operativ einsetzbaren Lernmodulen zu unterschiedlichen Themen des Informatikunterrichts zum Ziel hat, zusammen.

Mein Dank gilt vorab dem Fonds Filep des Rektors der ETH Zürich für die grosszügige finanzielle Unterstützung unserer Projekte, ohne die diese Arbeit nicht möglich gewesen wäre.

Speziell danken möchte ich folgenden Personen:

Hans Hinterberger für die vorzügliche Betreuung dieser Arbeit und das exzellente Arbeitsklima in seiner Gruppe. Er hat durch seine Offenheit und seinen Mut, in dieses multidisziplinäre Gebiet des E-Learnings vorzustossen, diese Arbeit möglich gemacht.

Juraj Hromkovic für die Übernahme des Korreferats,

Markus Dahinden für die vielen hilfreichen Diskussionen und die tatkräftige Unterstützung beim Tagesgeschäft im Unterricht,

Marco Wyss für die Unterstützung bei Fragen im psychologischen und pädagogischen Bereich,

Simone Fässler für das Lektorat dieser Arbeit,

der ganzen Gruppe für die tolle Atmosphäre,

Susanne Imper für ihre Geduld und die liebevolle Unterstützung,

den über 5000 Studierenden, die in den vergangenen 6 Jahren unseren Unterricht aktiv mitgestaltet und die Lernumgebungen belebt haben.

Inhaltsverzeichnis

Zusammenfassung	1
Abstract.....	3
Kapitel 1: Einleitung und Übersicht	5
1.1. Einsatz des Computers zum Unterrichten von Informatik-Kompetenzen in einem interdisziplinären Umfeld	5
1.2. Zielsetzungen und wissenschaftlicher Beitrag dieser Arbeit	5
1.2.1 Ebene Hochschuldidaktik: Einführung konstruktivistischer Methoden für grosse Klassen.....	7
1.2.2 Ebene Informatikunterricht: Unterrichten einer interdisziplinären Enabling Science.....	7
1.2.3 Ebene E-Learning: Einsatz von Lernsoftware zur Steuerung von Lernprozessen für den Aufbau von Kompetenzen.....	8
1.3. Formale Ziele und Outputs dieser Arbeit.....	10
1.4. Entwicklungskontext dieser Arbeit.....	12
1.5. Gliederung dieser Arbeit.....	13
Teil I: Grundlagen	
Kapitel 2: Kompetenzorientierter Unterricht	19
2.1. Einleitung.....	19
2.2. Generelle Merkmale einer Kompetenz	20
2.3. Aneignung von Wissen.....	22
2.3.1 Wissensaufbau und dessen Organisation in Konzepten	22
2.3.2 Verschiedene Arten von Wissen	23
2.3.3 Unterschiede bei der Aneignung verschiedener Wissensarten.....	27
2.3.4 Aneignung von Wissen unterschiedlicher Dimensionen als Konstruktionsprozess	30
2.4. Erhöhen der Unterrichtsqualität durch die Kompetenzorientierung	31

2.4.1 Nachhaltigkeit und Transfer des Wissens	31
2.4.2 Nachhaltiger, transferfreundlicher Unterricht.....	32
2.5. Kompetenzniveaus.....	33
2.5.1 Kurskompetenz und Fachkompetenz	33
2.5.2 Kognitive Prozess-Dimensionen einer Kompetenz.....	34
2.6. Fazit: Modell für einen kompetenzorientierten Unterricht	37
Kapitel 3: ICT-Kompetenzen für die wissenschaftliche Tätigkeit	39
3.1. Informatik als Enabling-Science in der wissenschaftlichen Tätigkeit.....	39
3.2. Kompetenzen im Umgang mit digitalen Daten und Informationen	40
3.2.1 Information-Literacy: Kompetenz im Umgang mit Information.....	41
3.2.2 Professioneller Umgang mit digitalen Daten	43
3.2.3 Computer-Fluency	44
3.3. Herstellen eines realistischen Kontexts zur Aneignung von ICT-Kompetenzen....	48
3.3.1 Der Computer als Informationsarbeitsplatz	48
3.3.2 Einsatz des Informationsarbeitsplatzes im Unterricht	49
3.3.3 ICT-Kompetenzen für die Arbeit im Informationsarbeitsplatz	50
3.3.4 Entwicklung einer Lernumgebung zum Aufbau von ICT-Kompetenzen im Informationsarbeitsplatz	52
3.4. Ressourcenabklärung bei Informatik-Einführungskursen an der ETH Zürich.....	53
3.4.1 Verfügbarkeit von Hardware bei den Studierenden.....	54
3.4.2 Verfügbarkeit von ICT-Kompetenzen bei den Studierenden.....	54
3.4.3 Kursprogramm zum Aufbau von ICT-Kompetenzen im Informationsarbeitsplatz	58
Kapitel 4: Kompetenzorientierte Lernumgebungen.....	59
4.1. Einleitung	59
4.2. Eigenschaften von kompetenzorientierten Lernumgebungen mit indirekter Instruktion.....	60
4.2.1 Lernumgebungen mit indirekter Instruktion	60
4.2.2 Kompetenzorientierung mit indirekter Instruktion	61
4.2.3 Lernmethoden und didaktische Modelle.....	64
4.3. Die Entwicklung von Lernumgebungen als System-Engineering.....	65

4.3.1 System-Engineering von Lernumgebungen.....	66
4.4. Modell eines kompetenzorientierten Lernsystems	68
4.4.1 Elemente eines kompetenzorientierten Lernsystems	68
4.4.2 Lerneinheit	69
4.4.3 Personengebundene Elemente.....	71
4.5. Entwicklung Lernprozess-unterstützender Strukturen	73
4.5.1 Ergonomische Entwicklung und Optimierung von Systemstrukturen.....	73
4.5.2 Anforderungen an Lernprozess-unterstützende Strukturen.....	74
4.5.3 Das Cognitive Apprenticeship-Modell	77
4.6. Fazit: Anforderungen an die Entwicklung einer kompetenzorientierten Lernumgebung.....	79
Kapitel 5: Problembasiertes Lernen.....	81
5.1. PBL als Lehr- und Lernmethode.....	81
5.1.1 Definitionen von PBL.....	81
5.1.2 Hohe Transfer-Wirksamkeit.....	83
5.1.3 Eignung für selbstgesteuertes Lernen	83
5.1.4 Einsatz von PBL zur Aneignung von ICT-Kompetenzen im Informationsarbeitsplatz.....	83
5.2. Aufbau von Kompetenzen durch PBL	84
5.2.1 Problemaufgaben als Vehikel der Aneignung von Kompetenzen.....	84
5.2.2 Typen von Problemaufgaben.....	87
5.2.3 Optimieren der Output-Kompetenz durch eine Kombination von Aufgabentypen.....	89
5.2.4 Entwicklung problembasierter Kurse.....	89
5.3. Motivationale und emotionale Faktoren des PBL.....	91
5.3.1 Erfolgserlebnisse als positive Verstärker für den Lernprozess	92
5.3.2 Motivationale Grössen beim Lösen von Problemen	94
5.3.3 Berücksichtigung motivationaler Grössen bei der Implementierung von PBL.....	97
5.4. Fazit: Effektives und motivierendes PBL.....	99
Kapitel 6: Selbstgesteuerte, problembasierte Instruktion	101
6.1. Einleitung.....	101

6.2. Anleitungen als Instruktionsmedium.....	101
6.3. Aufbau eines aktiven Wissensnetzes.....	103
6.3.1 Eine Instruktion für verschiedene Wissensdimensionen	103
6.3.2 Konstruktive Begriffsbildung	104
6.4. Elemente einer selbstgesteuerten, problembasierten Anleitung.....	106
6.4.1 Interaktion von Anleitung, Handlung und Überprüfung.....	106
6.4.2 Abgleich von Ist und Soll als Handlung.....	107
6.4.3 Motivationale und emotionale Aspekte während der Instruktion	108
6.5. Einsatz des Computers als Instruktionsmittel.....	111
6.5.1 <i>E.Tutorials</i> ®: Applikation für selbstgesteuerte, problembasierte Instruktionen.....	112

Teil II: Beschreibung und Umsetzung der didaktischen Modelle

Kapitel 7: Didaktische Modelle	117
7.1. Einleitung.....	117
7.1.1 Zeiträume zum Aufbau von Kompetenzen	118
7.1.2 Zwei Modelle für den Aufbau von Kompetenzen	119
7.2. Das 4-Schritte-Modell	120
7.2.1 In 4 Schritten zu aktivem, transferfähigem Wissen.....	120
7.2.2 Zeitlicher Ablauf des 4-Schritte-Modells.....	122
7.2.3 Einsatz von Medien im 4-Schritte-Modell	123
7.2.4 Einsatz von Hyperlinks im <i>E.Tutorial</i> ®	128
7.3. Das Anwendungsebenen-Modell.....	132
7.4. Evaluation des didaktischen Modells	133
7.5. Fazit: Didaktische Modelle für den kompetenzorientierten Unterricht.....	135
Kapitel 8: Entwicklung und Umsetzung kompetenzorientierter Lerneinheiten	137
8.1. Einleitung	137
8.1.1 Überblick.....	137
8.1.2 Entwicklung von kompetenzorientierten Lerneinheiten.....	137
8.2. Entwicklungsprozesse	138
8.2.1 Überblick über die Entwicklungsprozesse	139

8.2.2 Übersicht über die eingesetzten Hilfsmittel	143
8.2.3 Fallstudie: Umsetzung der didaktischen Modelle an einem Beispiel.....	143
8.3. Prozess A: Kompetenzorientierte Problemkonstruktion	144
8.3.1 Prozess A1: Festlegen des Zeitgefässes im Unterricht.....	144
8.3.2 Prozess A2: Zielgruppen-Analyse.....	145
8.3.3 Prozess A3: Kompetenz-Analyse	147
8.3.4 Prozess A4: Auswahl der Informatikmittel	147
8.3.5 Output Prozess A: Problemstellungen	150
8.4. Prozess B: Anwendungsebenen-Modell.....	151
8.4.1 Entscheidung 1: Ist die Aufteilung in Teilkompetenzen erforderlich?	151
8.4.2 Output Prozess B: Problemserie zum Aufbau einer (Teil-)Kompetenz	152
8.5. Prozess C: Erstellen der Lernmaterialien für das 4-Schritte-Modell.....	153
8.5.1 Prozess C1: Formulieren der Aufgabenstellung	155
8.5.2 Prozess C2: Formulieren der Begleitunterlagen	155
8.5.3 Prozess C3: Problembasierte Instruktion.....	156
8.5.4 Output Prozess C3: <i>E.Tutorial</i> ®	162
8.6. Prozess D: Qualitäts-Feedback.....	166
8.6.1 Prozess D1: Verbesserung der Lernmaterialien.....	166
8.6.2 Prozess D2: Kompetenzkontrolle	167
8.6.3 Prozess D3: Implementierung der Lerneinheit in den Gesamtkurs	169
8.7. Fazit.....	172

Teil III: Einsatz, Evaluation und Weiterentwicklung

Kapitel 9: Einsatz der computergestützten Lernumgebung im Unterricht	177
9.1. Einleitung	177
9.2. Kursimplementierung: von den Lernmaterialien zum operativen Kurs	177
9.2.1 Ablaufstrukturen.....	178
9.2.2 Rollen und Aufgaben der am Unterricht im 4-Schritte-Modell beteiligten Personen.....	180
9.3. Übersicht über Aufgaben, Personen und Ablaufstrukturen für das Unterrichten eines Kurses im 4-Schritte-Modell	181
9.3.1 Aufgaben	181

9.3.2 Personen und Verantwortungen	181
9.3.3 Ablaufstrukturen	182
9.4. Zeitlicher Ablauf und Umsetzung der Aufgaben von Kursen im 4-Schritte- Modell	183
9.4.1 Überblick	183
9.4.2 Kursspezifische Informationen bereitstellen (Aufgabe A)	185
9.4.3 Lernmaterialien bereitstellen, verteilen und unterhalten (Aufgabe B).....	186
9.4.4 Assistenzpersonen-Monitoring durchführen (Aufgabe C)	188
9.4.5 Leistungskontrollen und -nachweise organisieren und durchführen (Aufgabe D)	189
9.4.6 Qualitätskontrolle eines Kurses durchführen (Aufgabe E)	192
9.4.7 Präsenzveranstaltungen organisieren und halten (Aufgaben F1-6)	192
9.5. Fazit.....	194
Kapitel 10: Evaluation der Lernumgebungen	197
10.1. Einführung	197
10.2. In dieser Arbeit durchgeführte Evaluationen.....	198
10.2.1 Übersicht	198
10.2.2 Datenerhebungen im Unterricht.....	198
10.3. Outputanalyse: Anwendungsorientierte Leistungskontrolle als Mass für das Vorhandensein von anwendbaren Wissensstrukturen	201
10.3.1 Übersicht.....	201
10.3.2 Einführung	201
10.3.3 Leistungskontrolle als Evaluationsinstrument.....	202
10.3.4 Entwicklungsverlauf von Unterricht und Leistungskontrolle	204
10.3.5 Resultate der Outputanalyse	206
10.3.6 Diskussion der Resultate der Outputanalyse.....	207
10.4. Prozessanalyse: Überprüfung der Implementierungsqualität von Problem- aufgaben in problembasierten Kursen.....	210
10.4.1 Übersicht	210
10.4.2 Einführung	210
10.4.3 Beispiel eines Implementierungsprozesses: Programmierkurs	211
10.4.4 Durchführung der Prozessanalyse	212
10.4.5 Resultate der Prozessanalyse	213

10.4.6 Diskussion der Resultate der Prozessanalyse	215
10.4.7 Folgerungen aus den Prozessanalysen	217
10.5. Weiterführende Evaluationen	220
Kapitel 11: Schlussfolgerungen und Ausblick	221
11.1. Schlussfolgerungen	221
11.1.1 Wichtige Rolle der Informatik als Enabling Science	221
11.1.2 Qualitätsverbesserung des Informatik-Grundlagenunterrichts durch kompetenzorientiertes E-Learning	222
11.1.3 Entwickeln von Lernumgebungen als multidisziplinäre Disziplin	222
11.2. Grundlagen für die Weiterentwicklung der Lernumgebungen	223
11.3. Weiterentwicklung der Lernumgebungen „Einsatz von ICT“ und „Programmieren“	225
11.3.1 Entwicklung eines Systems für die Leistungskontrolle	225
11.3.2 Anpassen der Lernumgebungen auf weitere Zielgruppen	226
11.3.3 Erweitern bereits bestehender Lernumgebungen durch neue Inhalte	227
11.4. Entwicklung neuer computergestützter Lernumgebungen	228
11.4.1 Informatikunterricht	228
11.4.2 Fachunterricht	229
11.4.3 Industrie	230
 Anhang	
Anhang A: Kursprogramme	235
A1. ICT-Kurs	235
A2. Programmierkurs	237
Anhang B: Programmcode Programmierkurs	239
B.1 E.Tutorial® 1: Elektronisches Schloss	239
B.2 E.Tutorial® 2: Berechnung von Standard-Papierformaten	240
B.3 E.Tutorial® 3: Geburtstags-Paradoxon	240
B.4 E.Tutorial® 4: Galton-Board	241
B.5 E.Tutorial® 5: Barcode	242
B.6 E.Tutorial® 6: BMI	242

Anhang C: Fragebogen	243
C.1 Vorumfrage (ICT-Kurs)	243
C.2 Nachumfrage (ICT-Kurs).....	246
 Verzeichnisse	
Abbildungsverzeichnis	251
Tabellenverzeichnis	257
Literaturverzeichnis	261
Lebenslauf	266

Zusammenfassung

Diese Dissertation beschäftigt mit folgenden Herausforderungen des universitären Einführungsunterrichts im Bereich der Informations- und Kommunikationstechnologien: erstens die Vermittlung von aktivem statt tragem Wissen, zweitens die Motivierung der Studierenden von nicht technisch orientierten Fachrichtungen für das Fach Informatik und drittens die Individualisierung des Unterrichts trotz grosser Klassen.

Um diese Zielsetzungen erreichen zu können, soll der Computer Werkzeuge und Instruktionen zur Verfügung stellen, damit sich die Lernenden selbstgesteuert und in einem realistischen Kontext Kompetenzen aneignen können. Kernelement dieser Lernumgebungen bilden eine Reihe authentischer Problemstellungen aus der zukünftigen wissenschaftlichen Tätigkeit der Studierenden. Damit sich die Lernenden individuell und weitgehend unabhängig von ihrem Vorwissen auf das selbständige Lösen solcher Problemaufgaben vorbereiten können, wurde eine E-Learning-Applikation mit dem Namen *E.Tutorial*[®] zur Durchführung problem- und handlungsbasierter Instruktionen entwickelt, bei der sich Anleitung, Werkzeug und Überprüfungsmöglichkeit auf einer Arbeitsfläche befinden. Um diese E-Learning-Applikation in die Strukturen einer Blended Learning-Umgebung einzubetten, wurde ein kompetenzorientiertes didaktisches Modell, das *4-Schritte-Modell*, entwickelt und umgesetzt. Es ermöglicht einen effizienten und effektiven Wissenstransfer für ausgewählte Konzepte in den vier Schritten *SEE*, *TRY*, *DO* und *EXPLAIN*. Die Konzepte werden zunächst computergestützt und problembasiert instruiert (Phasen *SEE* und *TRY* mittels *E.Tutorial*[®]), anschliessend in einer neuen Problemstellung selbständig mit dem Computer angewendet (Phase *DO*) und zum Schluss die eigene Lösung einem Experten erklärt (Phase *EXPLAIN*).

Ein zentraler Schritt für den Aufbau solcher Lernumgebungen betrifft die strukturierte Aufarbeitung der Lerninhalte zu operativ einsetzbaren Lernmodulen. Dazu gehört die Integration der zu vermittelnden Konzepte in eine für die Zielgruppe geeignete Problemstellung sowie die Entwicklung von Lernmaterialien für den Unterricht im 4-Schritte-Modell bestehend aus *E.Tutorials*[®], Begleitunterlagen und Aufgabenstellungen. In dieser Dissertation wird ein Szenario für die Entwicklung von Lernmodulen zum Aufbau kompetenz- und problembasierter Lernumgebungen am Beispiel eines Programmierkurses für Naturwissenschaftlerinnen und Naturwissenschaftler vorgestellt. Die Lernumgebungen zeichnen sich durch hohe Stabilität im Einsatz und Beliebtheit bei den Studierenden aus. Grosse Klassen verlangen jedoch für die Steuerung und Kontrolle der Phasen des 4-Schritte- Modells effiziente Ablaufstrukturen. Eine wichtige

Erkenntnis war, dass auf einen Kontakt zwischen Lehrenden und Lernenden nicht gänzlich verzichtet werden kann (Blended Learning).

Die Resultate der im Rahmen dieser Arbeit durchgeführten Qualitätskontrolle bestätigen die Überzeugung, dass es möglich ist, mit Hilfe von E-Learning-Materialien eine deutliche Qualitätsverbesserung des Unterrichts herbeizuführen. Dies zeigt sich zum einen in der Fähigkeit der Studierenden, ihr angeeignetes Wissen zur Lösung neuer Probleme anzuwenden und zum anderen in einem positiven Motivationsverlauf der Studierenden während eines ganzen Semesters.

Die durch diese Arbeit geschaffenen Grundlagen zur Entwicklung computergestützter, kompetenzorientierter Lernumgebungen bleiben nicht auf den Informatikunterricht beschränkt, sondern können überall dort eingesetzt werden, wo die Aneignung von Kompetenzen direkt am und mit dem Computer von Interesse ist.

Abstract

This dissertation is concerned with the following tasks of introductory university teaching in the area of information technology: first, to convey active rather than inert knowledge; second, to motivate students from non-technical subjects in the area of information technology; and third, to individualize teaching despite large classes.

To meet these challenges the computer must make available tools and instructions which enable learners to acquire competences on their own, and in a realistic context. The core element of the learning environment in this project is a series of authentic problems drawn from the students' future scientific activities. In order that learners may prepare themselves to solve problems autonomously, individually and more or less independently from previous knowledge, an e-learning application called *E.Tutorial*[®], with instructions, tools and verification procedure all found on the same desktop, has been developed for students to work through problem- and activity-based instructions. To embed this e-learning application in the structure of a Blended Learning Environment, a competence-oriented didactic model, the *4-Step Model*, was developed and implemented. It facilitates efficient and effective transfer of knowledge involving selected concepts in the four steps *SEE*, *TRY*, *DO* and *EXPLAIN*. The concepts are first introduced by computer in a problem-based manner (phases *SEE* and *TRY*) by means of the *E.Tutorial*[®]. They are then applied independently to a new problem using the computer (the *DO* phase). Finally, the independent solution is explained to an expert (the *EXPLAIN* phase).

A central step in setting up this type of learning environment involves structuring the learning content into operatively deployable learning modules. To this process belong the integration of the to-be-conveyed concepts into a problem appropriate to the target group, and the development of learning materials for 4-Step-Model teaching comprising *E.Tutorials*[®], accompanying texts and tasks. This dissertation presents a scenario for the development of learning modules which set up competence- and problem-based learning environments and there deploy a programming course for scientists as an example. These learning environments demonstrate great stability in use and are popular with students. Large classes, however, require efficient procedural structures for steering and control of the 4-step-model phases. An important realization was that personal contact between instructor and learner remains fundamental and may not be entirely dispensed with (Blended Learning).

The results of the quality control assessment carried out in the framework of this project confirm the conviction that a significant improvement in the quality of teaching may be achieved with the help of e-learning materials. This is seen on the one hand in the students' ability to apply their acquired knowledge to solve new problems, and on the other in their high level of motivation over the whole semester.

The bases for developing computer-supported, competence-oriented learning environments elaborated in this project do not just apply to information technology teaching, but may be deployed in any area where acquiring competences directly on and using the computer is of interest.

Kapitel 1

Einleitung und Übersicht

1.1. Einsatz des Computers zum Unterrichten von Informatik-Kompetenzen in einem interdisziplinären Umfeld

Im universitären Unterricht ist Informatik für viele Fachbereiche Teil der Grundausbildung und nimmt den Status einer wichtigen *Enabling Science* (Englisch für „befähigend“ oder „ermächtigend“) ein, die den professionellen und effizienten Einsatz des Computers zum Lösen von Problemen zum Ziel hat. Die Ausübung einer solchen Kompetenz im Umgang mit dem Computer setzt Wissen über Konzepte und Methoden der Informatik voraus und ist von der Fähigkeit abhängig, dieses Wissen beim Lösen von Problemen in der realen Welt einzusetzen. Der Computer spielt im Informatikunterricht eine ganz besondere Rolle: er ist gleichzeitig Lerninhalt und Gegenstand des Unterrichts.

In dieser Arbeit sollen Grundlagen für *kompetenzorientierte Lernumgebungen* für den Informatikunterricht aufgezeigt werden, deren Strukturen auf effiziente Art und Weise die Aneignung von Informatik-Kompetenzen unterstützen. Die Lernenden sollen direkt am Computer problembasiert in die Konzepte und Methoden der Informatik eingeführt werden und gleichzeitig den Computer als Werkzeug nutzen, um gegebene Probleme zu lösen, indem sie die Konzepte und Methoden anwenden.

Dieses einführende Kapitel gibt einen Überblick über die verschiedenen Entwicklungsbereiche und den Entwicklungskontext dieser multidisziplinären Arbeit und stellt die formalen Ziele und Outputs vor.

1.2. Zielsetzungen und wissenschaftlicher Beitrag dieser Arbeit

Diese Arbeit ist multidisziplinär. Im Zentrum steht ein *Educational Engineering*, dessen Ziel die Entwicklung von kompetenzorientierten Lernumgebungen für den operativen Einsatz im universitären Unterricht ist (Abb. 1.1). Ein Educational Engineering besteht aus einer inhaltlichen und einer funktionellen Komponente. Zur inhaltlichen Komponente gehört die Aufarbeitung und Integration der zu vermittelnden Lerninhalte (in unserem Fall der Informatik) in die Lernumgebung. Die funktionelle Komponente setzt Konzepte aus unterschiedlichen Disziplinen (Psychologie, Didaktik, Pädagogik, Neurobiologie usw.) innerhalb der Lernumgebung um, mit dem Ziel, Lernprozesse im

Unterricht effektiv zu unterstützen und zu optimieren. Der Einsatz des Computers als Lernmedium soll zusammen mit Lernsoftware zu jenen Aktivitäten führen, die eine Anreicherung von Kompetenzen ermöglichen.

Mit dieser Arbeit wurden Ziele in unterschiedlichen Bereichen verfolgt (Abb. 1.1):

- 1. Eine Qualitätsverbesserung des universitären Grundlagenunterrichts durch die Einführung konstruktivistischer Methoden für grosse Klassen (Abschnitt 1.2.1)
- 2. Das Unterrichten von Lerninhalten der Informatik als interdisziplinäre Enabling Science für diverse Fachrichtungen (Abschnitt 1.2.2)
- 3. Der Einsatz des Computers im Unterricht als Lernmedium und die Entwicklung von Lernsoftware zur Steuerung von Lernprozessen für den Aufbau von Kompetenzen (Abschnitt 1.2.3)

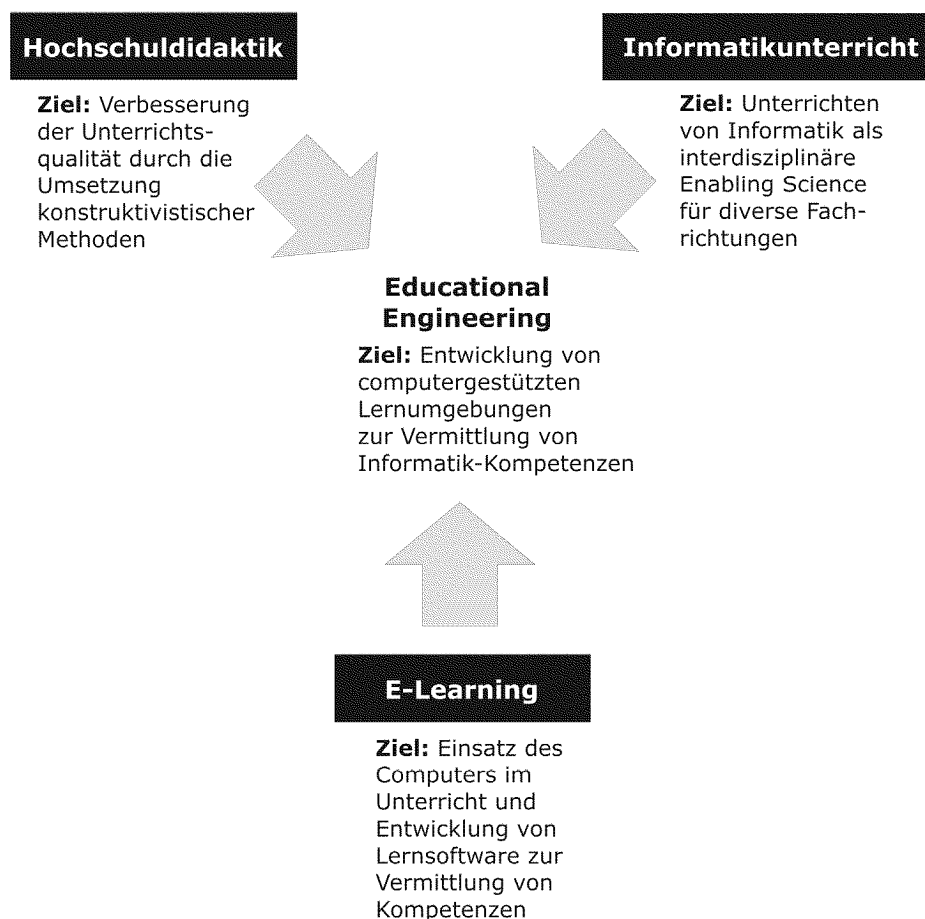


Abb. 1.1: Zielsetzungen dieser Arbeit in den Bereichen Hochschuldidaktik, Informatikunterricht und E-Learning. Übergeordnetes Ziel ist die Ausübung eines Educational Engineerings für die Entwicklung computergestützter Lernumgebungen zum Unterrichten von Informatik-Kompetenzen.

1.2.1 Ebene Hochschuldidaktik: Einführung konstruktivistischer Methoden für grosse Klassen

Grundlagenunterricht auf universitärer Stufe ist häufig durch folgende Eigenschaften geprägt:

- grosse Klassen
- anspruchsvoller Inhalt mit hohem Theorieanteil
- wenig Praxisbezug
- unzureichende Gelegenheiten zum Einüben von Grundlagen
- wenig Selbststeuerungsmöglichkeiten der Lernenden

Im Unterricht mit diesen Eigenschaften werden Lerninhalte zu einem grossen Teil durch Darbieten und Erklären vermittelt. Die Lernenden sind hierbei vornehmlich in einer passiven und stark von aussen angeleiteten Rolle. Es ist bekannt, dass unter diesen Umständen angeeignetes Wissen häufig träge bleibt und in neuen Situationen bei der Lösung von Problemen nicht aktiv eingesetzt werden kann [Mandel 1993, Renkl 1996]. Damit Lernende angeeignetes Wissen später auch anwenden und weiterentwickeln können, sind *konstruktivistische Methoden* in Betracht zu ziehen [Riedl & Schelten 2006]. Diese sind vermehrt handlungsorientiert, individualisiert und realitätsnah. Für die Lernenden bedeutet dies, dass sie sich in eine aktivere Position begeben und sich intensiv und eigenständig mit den Lerninhalten auseinandersetzen müssen. Die Funktion des Lehrenden beschränkt sich darauf, die Lernenden zu unterstützen und zu beraten. Für die zu entwickelnde Lernumgebung stellt sich die Frage, wie konstruktivistische Prozesse für den Unterricht mit grossen Klassen und inhaltlich anspruchsvollen Lerninhalten organisiert und mit zu schaffenden Strukturen sinnvoll unterstützt werden können, damit ein effektives und motivierendes Lernen ermöglicht wird.

1.2.2 Ebene Informatikunterricht: Unterrichten einer interdisziplinären Enabling Science

Für viele Disziplinen (wie z.B. die Naturwissenschaften) ist heute die Ausübung wissenschaftlicher Tätigkeiten stark mit Konzepten und Methoden der Informatik (wie z.B. Datenverwaltung, Netzwerke und Programmierung) verknüpft. Softwareschulungen, wie sie heute zahlreich angeboten werden, reichen hierfür nicht aus, weil sie meistens keine Konzepte vermitteln. Andererseits ist es für das Verständnis von Informatik-Konzepten notwendig, dass diese zusammen mit einem Kontext vermittelt, respektive gelernt werden. Das Unterrichten einer solchen Enabling Science verlangt deshalb nach neuen interdisziplinären Ansätzen der Kompetenzvermittlung: Konzepte der Informatik sollen handlungsbasiert in einem realen Kontext angeeignet werden. Damit die

Lernenden die Konzepte und Methoden direkt mit praktischen Fertigkeiten verknüpfen können, ist die Arbeit in einem Informatik-Labor sinnvoll. Dies hat den Vorteil, dass das Wissen mit realen Handlungen verknüpft werden kann und dadurch nachhaltiger gelernt wird [Reusser 2005].

Das Lernen durch die Arbeit in diesem Informatik-Labor sollte mit Vorteil so konzipiert sein, dass es nicht nur im Computerraum einer Schule, sondern auf jedem beliebigen Computer (z.B. dem privaten Laptop der Lernenden) betrieben werden kann.

1.2.3 Ebene E-Learning: Einsatz von Lernsoftware zur Steuerung von Lernprozessen für den Aufbau von Kompetenzen

Beim E-Learning interessiert der Einsatz des Computers als Lernmedium und die Möglichkeit, mit Hilfe der internetbasierten Kommunikationsmöglichkeiten Lernen orts- und zeitunabhängig zu gestalten. Bei der Entwicklung von Lernsoftware steht die Interaktion zwischen den Lernenden und den durch den Computer repräsentierten Lernobjekten im Vordergrund, mit dem Ziel, den individuellen Lernprozess zu unterstützen und vermehrt auch selber steuern zu lassen.

Schulmeister hat eine Taxonomie zur Klassierung der Interaktivität von Lernsoftware eingeführt [Schulmeister 2002]. Mit Interaktivität sind die Möglichkeiten der Lernenden gemeint, im Zusammenhang mit einer E-Learning-Applikation ein Lernobjekt zu manipulieren. Für den Interaktivitätsgrad ist entscheidend, wie viel Handlungsfreiheit den Lernenden eingeräumt wird (Tab. 1.1).

Ein hoher Level an Interaktivität wird gemäss Schulmeisters Taxonomie dann erreicht, wenn die Lernenden Objekte nicht nur betrachten und manipulieren (Tab. 1.1, Stufen 1 bis 4), sondern den Inhalt der Repräsentation selber konstruieren sowie Prozesse generieren und steuern können (Tab. 1.1, Stufen 5 und 6). Die beiden höchsten Stufen erlauben eine selbständige Konstruktion der Lernobjekte durch eigenes Handeln. Die Stufe 6 lässt individualisierte Lösungen der Lernenden zu und behandelt diese durch situativ sinnvolle Rückmeldungen. Mit zunehmender Handlungsfreiheit der Lernenden steigt auch die Anforderung an das Lernprogramm, sich an die individuellen Situationen anpassen zu können.

Um mit dem Einsatz des Computers im Unterricht gegenüber traditioneller Unterrichtsmedien einen Mehrwert zu generieren, sind höhere Stufen der Interaktivität anzustreben (Stufen 4 und höher). Lernprogramme, die auf höhere Interaktivitätsstufen abzielen, haben die Eigenschaft, dass sie Lernobjekte nicht bloss darstellen, sondern mit Werkzeugen und Problemstellungen Hilfsmittel für die Lernprozesssteuerung im Sinne von Learning-Management-Systemen zur Verfügung stellen.

	Beschreibung der Stufe	Lernobjekte	Handlungsfreiheit der Lernenden in Bezug auf das Lernobjekt
Stufe 1	Computer als Medium zum Betrachten und Rezipieren von Objekten	vorgefertigt	Auf die Darstellungsform kann kein Einfluss genommen werden.
Stufe 2	Multiple Darstellungen betrachten und rezipieren	vorgefertigt	Verschiedene Darstellungsformen sind wählbar.
Stufe 3	Repräsentationsform variieren	vorgefertigt	Repräsentationsform kann frei variiert werden.
Stufe 4	Den Inhalt der Komponente beeinflussen: Variation durch Parameter- oder Datenvariation	durch Lernende generiert	Freie Eingabe von Parametern und Daten ist möglich.
Stufe 5	Das Objekt bzw. den Inhalt der Repräsentation konstruieren und Prozesse generieren	durch Lernende generiert	Werkzeuge ermöglichen eigene Konstruktion des Lernobjekts.
Stufe 6	Konstruktive und manipulative Handlungen mit situationsabhängigen Rückmeldungen	durch Lernende generiert	Erlaubt individuelle Lösungen. Lernsoftware generiert situativ sinnvolle Rückmeldungen.

Tab. 1.1: Taxonomie zur Klassierung der Interaktivität von Lernsoftware [Schulmeister 2002].

Die Stufe 5 ist in der Taxonomie die höchste Stufe, welche direkt die Handlungsfreiheit der Lernenden im Umgang mit Lernobjekten betrifft. Auf Stufe 6 wird die Möglichkeit angestrebt, auf die Handlungen der Lernenden eine intelligente Rückmeldung zu erteilen. Sollen die Rückmeldungen vom Lernprogramm selber generiert werden, ergeben sich Verständnis- und Interpretationsprobleme, weil Computer kein Sinnverstehen betreiben können. Bei der konstruktiven Aneignung von Kompetenzen stellen individuelle Rückmeldungen (*Feedbacks*) ein unverzichtbares Element zwischenmenschlicher Kommunikation und sozialer Interaktion dar, die sich als zu komplex erweisen, um von einem Computer erledigt zu werden.

Grosses Potential sehen wir hingegen bei den Möglichkeiten, Lernobjekte von den Lernenden mit Hilfe von Computerwerkzeugen selber aktiv zu konstruieren und zu verändern (Tab. 1.1, Stufe 5). Für die Entwicklung von Lernsoftware stellt sich dabei die Frage, wie die Lernenden mit sinnvollen Problemstellungen, Konzepten und Rückmeldungen versorgt werden können, damit sie mit geeigneten Werkzeugen authentische Probleme effizient und unabhängig lösen können. Auf diese Weise soll den Lernenden ein maximaler Handlungsspielraum im Umgang mit den Lernobjekten ermöglicht werden. Um dabei Überforderungen zu vermeiden und stattdessen Erfolgserlebnisse

zu ermöglichen, ist eine Form von Instruktionen zu schaffen, welche die Lernenden auf diese anspruchsvolle Aufgabe vorbereitet.

1.3. Formale Ziele und Outputs dieser Arbeit

Bei dieser Arbeit wurde das Ziel verfolgt, eine computergestützte Lernumgebung mit einem hohen Interaktivitätsgrad zur Vermittlung von Informatik-Kompetenzen zu entwickeln und im Unterricht einzusetzen. Abb. 1.2 fasst die zur Entwicklung einer solchen Lernumgebung wichtigen Schritte sowie die im Zusammenhang mit dieser Arbeit entstandenen Outputs zusammen.

Ausgangslage: Lernen an authentischen Situationen

Der Computer soll als Medium dort eingesetzt werden, wo seine Stärken liegen: als Arbeitsmittel beim Lösen von Problemen im Arbeitsprozess. Auf diese Weise können die Studierenden mit authentischen Situationen konfrontiert werden, die sie am und mit dem Computer lösen (Abb. 1.2). Die dadurch erreichte Handlungsorientierung stellt eine wichtige Ausgangsbasis für die Aneignung von Kompetenzen dar. Die Lernenden konstruieren durch Handeln selber Lernobjekte und Prozesse (Interaktivitätsstufe 5, siehe Tab. 1.1) und lernen diese in der realen Welt einzusetzen.

Output 1: E-Learning-Applikation für die computergestützte, problembasierte Instruktion

Um die Lernenden unabhängig von Ort, Zeit und Lehrperson auf die Anforderungen realer Situationen vorzubereiten, sollen problembasierte Instruktionen direkt am Computer durchgeführt werden. Zu diesem Zweck soll eine E-Learning-Applikation entwickelt werden, die eine problemorientierte Instruktion und deren handlungsorientierte Lösung durch die Bedienung eines Werkzeugs auf einer Bildschirmoberfläche erlaubt (Abb. 1.2, Output 1).

Um aus der entwickelten E-Learning-Applikation eine operativ einsetzbare Lernumgebung zu gestalten, sind zusätzlich folgende Entwicklungsaufgaben durchzuführen:

- die Entwicklung eines *didaktischen Modells*, welches die computer- und problembasierte Instruktion in sinnvolle Unterrichtsstrukturen und einen zeitlichen Ablauf einbindet, um ein effizientes und effektives Lernen zu garantieren (Abb. 1.2, Output 2)
- ein *Modell zur strukturierten Aufbereitung von Lerninhalten* für das Erstellen von Lernmodulen zur Vermittlung definierter Kompetenzen (Abb. 1.2, Output 3).

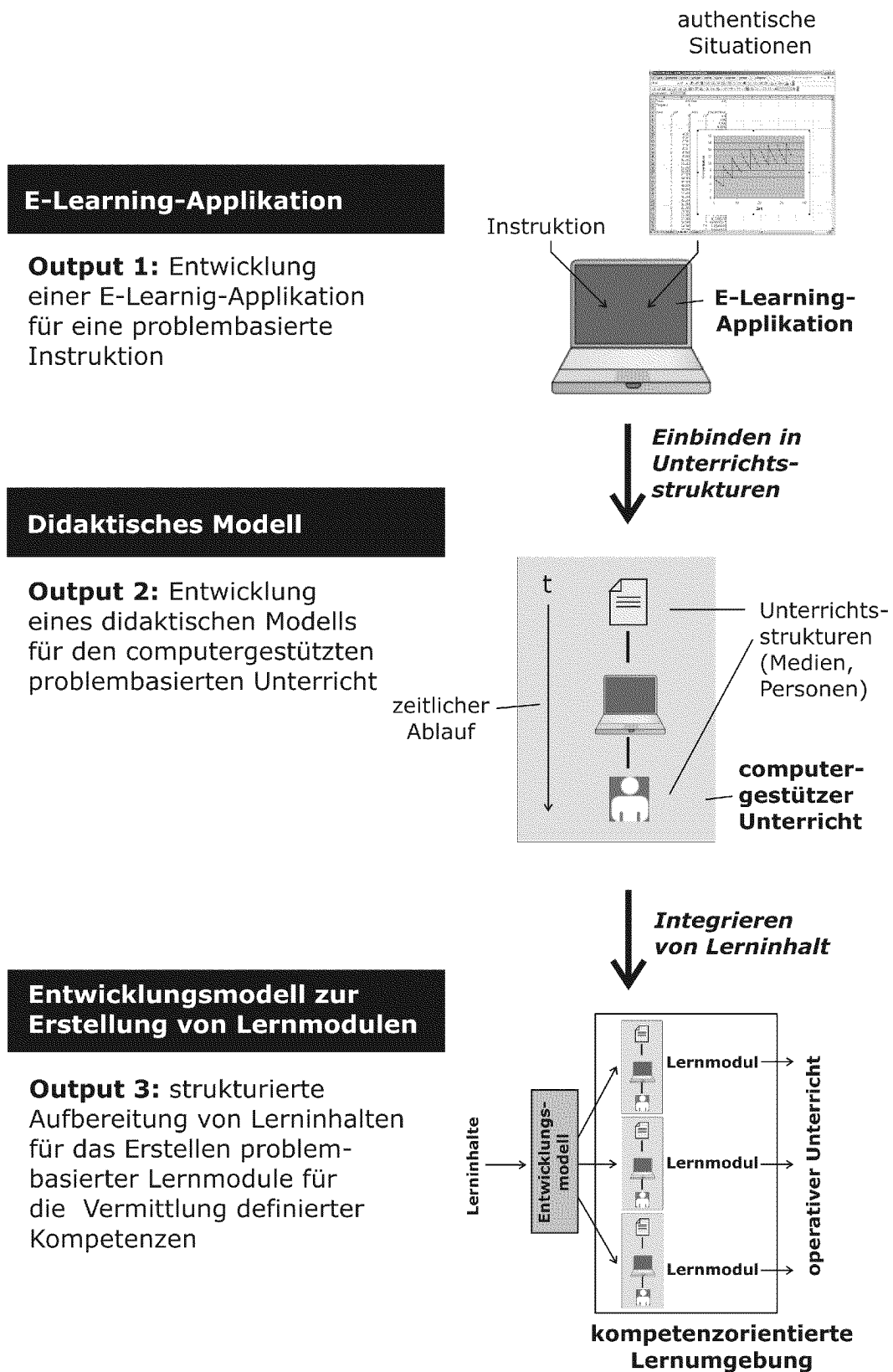


Abb. 1.2: Outputs dieser Arbeit. Beim Aufbau einer computergestützten und kompetenzorientierten Lernumgebung sind folgende Outputs entstanden: 1. eine E-Learning Applikation für die problem-basierte Instruktion für den Umgang mit authentischen Situationen, 2. ein didaktisches Modell für den computergestützten, problembasierten Unterricht und 3. ein Entwicklungsmodell zur Aufbereitung von Lerninhalten für die Erstellung problembasierter Lernmodule.

Output 2: Didaktisches Modell für den computergestützten, problembasierten Unterricht

Didaktische Modelle organisieren die Aktivitäten der Lernenden und Lehrenden sowie den Einsatz von Medien und Werkzeugen mit dem Ziel, den Lernprozess effizient und effektiv zu gestalten. Sie integrieren den Computer zusammen mit der E-Learning-Applikation in die Strukturen eines bestimmten Unterrichtsablaufs. Das zu entwickelnde didaktische Modell soll trotz Unabhängigkeit von Ort, Zeit und Lehrperson nicht auf zwischenmenschliche Kontakte verzichten, Rückmeldungen und Überprüfungsmöglichkeiten zu unterschiedlichen Zeitpunkten des Lernprozesses erlauben sowie die Selbständigkeit der Lernenden ermöglichen und fördern.

Output 3: Entwicklungsmodell für die strukturierte Aufbereitung von Lerninhalten für das Erstellen problembasierter Lernmodule

Ein zentraler Schritt beim Aufbau einer Lernumgebung ist die strukturierte Aufbereitung von Lerninhalten zu funktionellen Lernmodulen, die im operativen Unterricht zusammen mit dem didaktischen Modell und dem Einsatz der E-Learning-Applikation eine definierte Kompetenz generieren. Bei einer computergestützten, problembasierten Lernumgebung umfasst dies folgende Aufgaben:

- die Konstruktion und Auswahl von geeigneten Problemaufgaben zu ausgewählten Konzepten
- das Erstellen von Instruktionsmaterialien für die computergestützte, problembasierte Instruktion, die sich auf die ausgewählten Problemaufgaben beziehen.

Ein Entwicklungsmodell soll den Entwicklerinnen und Entwicklern kompetenzorientierter Lernumgebungen helfen, die Modularisierung des Unterrichts an realistischen Kompetenzen auszurichten und die Effektivität und Effizienz erstellter Lernmaterialien zu überprüfen und zu verbessern.

1.4. Entwicklungskontext dieser Arbeit

Die in der vorliegenden Arbeit entwickelten Lernumgebungen sind im Zusammenhang mit der Durchführung von sechs Informatik-Einführungskursen entstanden, die das Institut für Computational Science der ETH Zürich jedes Jahr für die Studierenden der Natur- und Bauingenieurwissenschaften mit über 1100 Studierenden durchführt (Tab. 1.2). Es gibt Studiengänge, deren Studierende ein einsemestriges Programm absolvieren, und solche mit einem zweisemestrigem Programm. Die Kurse sind auf einen Arbeitsaufwand von je 60-90 Stunden (2 bis 3 ECTS-Punkte) angelegt (weitere Details siehe Kapitel 3, Abschnitt 3.4.3).

Kurs-Nr.*	Titel des Kurses	Zielgruppe/ Fachrichtung	Anzahl Studierende**
551-0839-00L	Einsatz von Informatikmitteln (Kurs 1)	UWIS, AGRL, ERDW	284
251-0842-00L	Anwendungsnahes Programmieren (Kurs 2, Variante a)	UWIS, AGRL, ERDW	144
251-0842-00L	Programmieren und Problemlösen (Kurs 2, Variante b)	UWIS, AGRL, ERDW	30
252-0432-00L	Informatik für Biologie und Pharmazeutische Wissenschaften	BIOL, CHAB	278
251-0845-00L	Informatik I (Kurs 1)	BAUG	210
251-0846-00L	Informatik II (Kurs 2)	BAUG	196

Tab. 1.2: Informatik-Einführungskurse der Natur- und Bauingenieurwissenschaften an der ETH Zürich. Abkürzungen: AGRL: Agrar- und Lebensmittelwissenschaften, BAUG: Bau, Umwelt und Geomatik, BIOL: Biologie, Bewegungswissenschaften und Sport, CHAB: Chemie und angewandte Biowissenschaften, ERDW: Erdwissenschaften, UWIS: Umweltnaturwissenschaften. *Kurs-Nummer gemäss Vorlesungsverzeichnis ETH Zürich, 2007, **Anzahl Studierende 2006/07.

1.5. Gliederung dieser Arbeit

Diese Arbeit ist in drei Teile gegliedert. Im ersten Teil werden die Grundlagen zur Entwicklung computergestützter, kompetenzorientierter Lernumgebungen für den Informatikunterricht für angehende Wissenschaftlerinnen und Wissenschaftler behandelt. Im zweiten Teil stehen die Beschreibung und Umsetzung der umgesetzten didaktischen Modelle im Zentrum. In Teil drei folgen der Einsatz, die Evaluationsresultate und anschliessend die Diskussion über die Möglichkeiten der Weiterentwicklung der Lernumgebungen.

Teil I: Grundlagen

Kapitel 2: Kompetenzorientierter Unterricht

Das Unterrichten von definierten Kompetenzen soll eine Qualitätsverbesserung des Grundlagenunterrichts bringen, die im traditionellen instruktionsorientierten Unterricht nicht zu erreichen ist. Es soll das Problem vom trägen Wissen, das im theoriebasierten Unterricht entsteht, angehen und angeeignetes Wissen aktiver und nachhaltiger machen. In diesem Kapitel wird auf den Begriff der Kompetenz und die Bedingungen, unter welchen Kompetenzen angeeignet werden, eingegangen. Zudem

wird ein Mass zur Bestimmung verschiedener zu erreichender Kompetenzniveaus im Unterricht beigezogen.

Kapitel 3: ICT-Kompetenzen für die wissenschaftliche Tätigkeit

In der wissenschaftlichen Ausbildung vieler Fachrichtungen hat die Informatik die Stellung einer wichtigen Enabling Science inne. In diesem Kapitel wird diskutiert, welche Kompetenzen im Umgang mit den Informations- und Kommunikationstechnologien (ICT) für angehende Wissenschaftlerinnen und Wissenschaftler von Interesse sind. Die Vermittlung von ICT-Kompetenzen erfordert das Herstellen eines realistischen Kontextes. Um dies zu erreichen, haben wir den Begriff des Informationsarbeitsplatzes eingeführt, bei dem der Computer als universelles Arbeitsinstrument einer professionellen wissenschaftlichen Datenverarbeitung betrachtet wird. Damit sollen sich die Lernenden mit entsprechender Unterstützung ausgewählte ICT-Kompetenzen aneignen können.

Kapitel 4: Kompetenzorientierte Lernumgebungen

An die Entwicklung kompetenzorientierter Lernumgebungen, die einen möglichst selbstgesteuerten Aufbau von Kompetenzen unabhängig von Ort, Zeit und Lehrperson in individuellem Tempo ermöglichen sollen, stellen sich spezielle Anforderungen. Kompetenzorientierte Lernumgebungen sind komplexe Systeme, deren Eigenschaften und Anforderungen in diesem Kapitel anhand eines Modells diskutiert werden.

Kapitel 5: Problembasiertes Lernen

Durch die Lernmethode des Problembasierten Lernens (PBL) werden die Lernenden aktiv am Unterricht beteiligt. Sie ermöglicht die Aneignung von Kompetenzen aus Konzepten. Die Implementierung der Lernmethode in den Unterricht zu einem problembasierten Kurs erweist sich als grosse Herausforderung, sollen die Lernenden dadurch gefordert, aber nicht überfordert werden. Im Zentrum dieses Implementierungsprozesses steht die Konstruktion von Problemaufgaben. Deren konstruktionsrelevanten Grössen werden in diesem Kapitel vorgestellt.

Kapitel 6: Selbstgesteuerte, problembasierte Instruktion

Damit eine möglichst grosse Anzahl der Lernenden die maximale Wirkung aus der Lernmethode des problembasierten Lernens ziehen kann, müssen sich Novizen auf das Lösen von umfangreichen Aufgaben vorbereiten können. Selbstgesteuerte, problembasierte Anleitungen haben sich als effiziente Form erwiesen, sich die nötigen Wissensstrukturen zur Lösung von umfangreichen Problemen anzueignen. Zur Umsetzung von selbstgesteuerten, problembasierten Instruktionen mit Hilfe des Computers wurde eine Applikation mit dem Namen *E.Tutorial*[®] entwickelt, bei der sich Anleitung, Anwendung und Überprüfungsmöglichkeit auf derselben Arbeitsfläche befinden.

Teil II: Beschreibung und Umsetzung der didaktischen Modelle

Kapitel 7: Didaktische Modelle

Im Rahmen dieser Arbeit wurden zwei didaktische Modelle entwickelt und umgesetzt, die den problembasierten Einsatz des Computers für den Aufbau von Kompetenzen im Unterricht organisieren. Neben den zeitlichen Abläufen werden in diesem Kapitel die Sequenzierung und der Einsatz von Medien während der einzelnen Unterrichtsphasen der umgesetzten didaktischen Modelle besprochen.

Kapitel 8: Entwicklung und Umsetzung einer kompetenzorientierten Lernumgebung

In diesem Kapitel steht die Umsetzung der im 7. Kapitel beschriebenen didaktischen Modelle im Zentrum. Es werden sämtliche Entwicklungsprozesse am Beispiel eines Programmierkurses für Naturwissenschaftlerinnen und Naturwissenschaftler beschrieben und in einem Entwicklungsmodell zusammengefasst.

Teil III: Einsatz, Evaluation und Weiterentwicklung

Kapitel 9: Einsatz der computergestützten Lernumgebung im Unterricht

Der Einsatz einer kompetenzorientierten, computergestützten Lernumgebung hat Auswirkungen auf den gesamten Unterricht. In diesem Kapitel wird besprochen, wie der Ablauf des Unterrichts nach Abschluss der Entwicklung der Lernumgebung im Detail abläuft und wie sich die Arbeit der Lehrenden dadurch verändert hat.

Kapitel 10: Evaluation der Lernumgebungen

Neben den routinemässig durchgeführten Evaluationen zur Verbesserung der einzelnen Lernmodule haben wir auch Evaluationen durchgeführt, deren Fokus auf das ganze Lernsystem gerichtet ist. In diesem Kapitel werden die Resultate von zwei Evaluationen vorgestellt und diskutiert. Dabei geht es hauptsächlich um die Frage, ob mit der Entwicklung der kompetenzorientierten Lernumgebung unter dem Einsatz von E-Learning die Unterrichtsqualität erhöht werden konnte.

Kapitel 11: Schlussfolgerungen und Ausblick

Der Abschluss dieser Arbeit bildet eine Zusammenfassung der wichtigsten Erkenntnisse. Anschliessend soll diskutiert werden, in welche Richtung sich die im Rahmen dieser Arbeit entstandene Lernumgebung weiterentwickeln lässt und welches Potential in der Entwicklung neuer, nach dem gleichen didaktischen Modell erstellter Lernumgebungen steckt.

Teil I

Grundlagen

Kapitel 2

Kompetenzorientierter Unterricht

2.1. Einleitung

Als Kompetenz (*competence*) wird die Fähigkeit eines Menschen bezeichnet, bestimmte Aufgaben in einem Fachgebiet selbstständig durchzuführen [Erpenbeck 2003]. Kompetente Personen verfügen über Wissensstrukturen, welche sie zur Lösung von Problemen in einem bestimmten Bereich einsetzen können. Eine Kompetenz ist daher immer auch an eine Problemstellung mit einem Kontext sowie an bewusste und gerichtete kognitive Prozesse und von aussen beobachtbare Handlungen gebunden (Abb. 2.1). Letzteres wird auch als *Handlungskompetenz* bezeichnet. Handeln ist im Gegensatz zu „Tun“ oder „Verhalten“ zielgerichtet und bewusst [Wahl 2006].

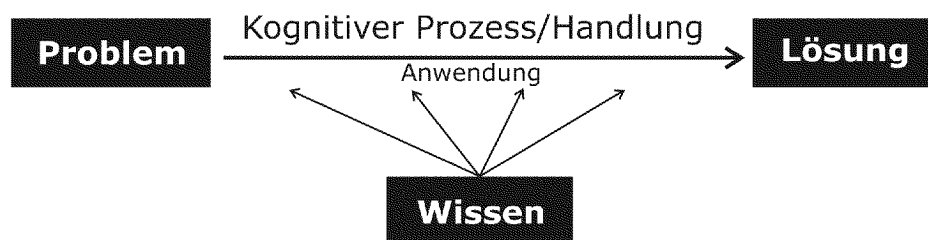


Abb. 2.1: Kompetenz als Fähigkeit, durch die Anwendung von Wissen kognitive Prozesse und Handlungen auszuführen, um Probleme in einem bestimmten Bereich lösen zu können.

Das Problem von trägem Wissen im Unterricht

In vielen Lernsituationen von Schule und Universität wird davon ausgegangen, dass die Lernenden das ihnen auf theoretischem Wege übermittelte Wissen nachfolgend spontan in geeigneten Problemsituationen anwenden. Bei Untersuchungen unter Medizern, Ingenieuren, Juristen und Lehrpersonen wurde aber erkannt, dass ihnen trotz langer schulischer Ausbildung oft die Fähigkeit fehlt, das schulisch erworbene Wissen auf die Probleme im praktischen Teil einer Ausbildung oder in Beruf anzuwenden [Gräsel 1993].

Träges Wissen (*inert knowledge*) ist Wissen, das zwar im Unterricht erworben wurde, aber beim Lösen einer Problemsituation nicht zur Verfügung steht [Mandel 1993, Renkl 1996]. Es hat sich gezeigt, dass träges Wissen vor allem deshalb entsteht, weil erworbenes Wissen zu wenig mit der Praxis in Verbindung gebracht wird. Die zeitliche und örtliche Trennung zwischen der Aneignung von Konzepten und Fertigkeiten und der

späteren Anwendung in der Realität, wie sie im Unterricht oft geschieht, erweist sich als zu gross. Weil beim instruktionsorientierten Unterricht gewöhnlich vor allem das Ziel der Aufnahme und Reproduktion von Fakten und Konzepten verfolgt wird, wird sie zu wenig mit Fertigkeiten und der praktischen Anwendung verknüpft, was den Aufbau eines Wissens- und Erfahrungsschatzes verhindert. Bei der späteren Anwendung sind die Lernenden nicht in der Lage, die instruktional gelernten Konzepte zu erkennen und weiterzuentwickeln. Um träges Wissen zu verhindern, muss versucht werden, Konzepte, Fertigkeiten und Anwendung bereits beim Erlernen der Grundlagen zu verbinden, indem sie als Teile einer bestimmten Kompetenz betrachtet werden.

Vermittlung von Kompetenzen im Grundlagenunterricht

Der Ansatz, bereits im Grundlagenunterricht definierte Kompetenzen zu vermitteln, hat zum Ziel, die Qualität des Unterrichts zu verbessern. Die Lernenden sollen Wissen nicht bloss wiedergeben können, sondern das Gelernte zur Lösung von Problemen einsetzen und anwenden können. Wissen, welches im Zusammenhang mit Kompetenzen und dem Lösen von Problemen angeeignet wurde, erweist sich als weniger träge und kann beim Lösen künftiger Probleme besser angewendet werden [Gruber 2000].

In diesem Kapitel soll folgenden Fragen nachgegangen werden:

- Was zeichnet kompetente Personen aus? (Abschnitt 2.2)
- Über welche Arten von Wissen verfügen kompetente Personen? (Abschnitt 2.3)
- Weshalb soll kompetenzorientierter Unterricht eine höhere Unterrichtsqualität bringen? (Abschnitt 2.4)
- Welche Kompetenzniveaus können im Unterricht erreicht werden? (Abschnitt 2.5)

2.2. Generelle Merkmale einer Kompetenz

Kompetente Personen unterscheiden sich von Novizen bei der Auseinandersetzung mit Problemen durch generelle Merkmale. Diese Merkmale umfassen unter anderem folgende Punkte [Mietzel 2003]:

- Kompetente Personen verfügen über umfangreiches bereichsgebundenes Wissen. Dieses besteht aus unterschiedlichen Arten von Wissen, so genannten Wissensdimensionen (siehe nächster Abschnitt).
- Kompetente Personen können ausserordentlich schnell Gegebenheiten erkennen und als sinnvolle Einheit wahrnehmen. Dabei hilft die Fähigkeit, sinnvolle Muster, so genannte Schemata, erkennen zu können.

- Viele grundlegende Prozesse laufen sehr schnell ab. Viele Verfahrensweisen sind automatisiert. Es ist kein „Nachdenken“ mehr erforderlich.
- Kompetente Personen achten auf relevante Merkmale, während Novizen auch Irrelevantes beachten.
- Kompetente Personen wenden im Vergleich zu Novizen mehr Zeit auf, um Situationen genau zu analysieren.
- Kompetente Personen sind im Gegensatz zu Novizen in der Lage, dank ihrer Erfahrung jene Massnahme als Lösungsstrategie auszuwählen, die mit höchster Wahrscheinlichkeit zum Ziel führt.
- Kompetenten Personen gelingt es im Vergleich zu Novizen sehr viel besser, ihre eigenen kognitiven Prozesse zu kontrollieren. Sie kommentieren häufig spontan ihre eigenen Denkprozesse sowie ihr Wissen.

Beim Betrachten dieser Merkmale fällt auf, dass der Vorteil der kompetenten Personen gegenüber der Novizen auf einen grossen Erfahrungshintergrund zurückzuführen ist, der vor allem in der praktischen und reflektierenden Auseinandersetzung mit dem zu lösenden Problem erworben worden ist. Die daraus abstrahierten Muster (Schemata) und das Vorhandensein von Wissensnetzen führen dazu, dass Probleme effizient und professionell gelöst werden können.

Die wesentlichen Merkmale einer Kompetenz sollen anhand eines Beispiels aus dem Alltag, der Kompetenz „Autofahren“, betrachtet werden. Diese Kompetenz manifestiert sich als Handlung, bei der sich eine Person im Strassenverkehr selbständig als kompetente, sichere Autofahrerin oder Autofahrer verhalten kann. Eine kompetente Person muss sich dem sich ständig verändernden „Problem“ Strassenverkehr in sehr kurzer Zeit anpassen können. Dabei bringt die Person neben einem soliden Wissensnetz Erfahrung im Umgang mit Situationen mit, auf die später zurückgegriffen werden kann. Dieses Wissensnetz besteht aus verschiedenen Arten von Wissen. Faktenwissen alleine würde bei der Ausübung dieser Kompetenz wenig nützen. Viel mehr muss Wissen über Regeln zusammen mit verschiedenen Fertigkeiten und Fähigkeiten (z.B. Parkieren, Autobahnfahren etc.) bei sich ständig ändernden Bedingungen flexibel angewendet werden können.

Das Erlangen einer Kompetenz ist stets mit einem erheblichen Zeitaufwand verbunden, weil neben dem Aufbau eines Wissensnetzes auch Erfahrungen im praktischen Umgang eine zentrale Rolle spielen. Wie wichtig Erfahrungen und Reflexion im Lernprozess sind, ist am Beispiel der Kompetenz „Schachspielen“ zu sehen. Die Spielregeln hat man schnell einmal gelernt, diese reichen aber alleine nicht aus, um ein erfolgreiches Spiel zu spielen. Zwischen dem Wissen über die Regeln und deren Anwendung im Spiel lie-

gen viele Schritte. Um die Kompetenz eines guten Spielers zu erlangen, ist viel Spielpraxis und Erfahrung nötig.

2.3. Aneignung von Wissen

Der Aufbau einer Kompetenz geht mit der Aneignung von Wissen einher. Deshalb wollen wir uns zunächst mit dem Begriff *Wissen* näher beschäftigen. Für Wissen gibt es viele Definitionen. Wir möchten uns mit folgenden Aspekten einer Wissensdefinition auseinandersetzen:

- Wie wird Wissen aufgebaut und organisiert? (Abschnitt 2.3.1)
- Welche Arten von Wissen können bei kompetenten Personen unterschieden werden? (Abschnitt 2.3.2)
- Welche Unterschiede gibt es bei der Aneignung der verschiedenen Wissensarten? (Abschnitt 2.3.3)
- Aneignung von Wissen als Konstruktionsprozess (Abschnitt 2.3.4)

2.3.1 Wissensaufbau und dessen Organisation in Konzepten

Wissen besitzt die Eigenschaft, dass es stets an einen Kontext und an einen Träger, zum Beispiel an eine Person oder eine Organisation, gebunden ist [Hey 2004]. Die Aneignung von Wissen ist immer mit einem individuellen Erkenntnisprozess verbunden. Wissen entsteht, wenn *Information* von einem Individuum in einen Bedeutungskontext gesetzt wird. Wissen muss im Gehirn eines jeden Lernenden immer wieder neu geschaffen werden [Roth 2003]. Um Wissen von Personen unabhängig zu machen, werden durch Abstrahierungs- und Verallgemeinerungsprozesse *Konzepte* gebildet. Auf diese Weise lässt sich Wissen in Form von Konzepten organisieren, kommunizieren und unter anderen Bedingungen in einem neuen Kontext anwenden (Abb. 2.2).

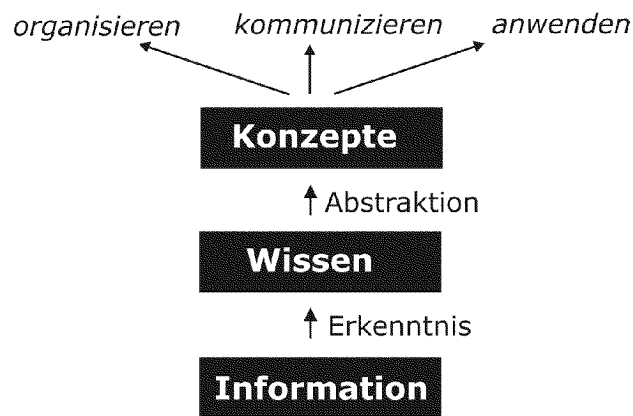


Abb. 2.2: Der Aufbau von Wissen aus Informationen ist an einen individuellen Erkenntnisprozess gebunden und muss für jedes Individuum neu geschaffen werden. Durch Abstrahieren entstehen Konzepte, durch die sich Wissen organisieren, kommunizieren und anwenden lässt.

Weil der Aufbau von Wissen und Konzepten an Erkenntnis- und Abstraktionsprozesse gebunden ist, können sie nicht direkt „vermittelt“, sondern lediglich in Form von Information mitgeteilt werden. Sie müssen für die Vermittlung in sinnvolle Informationen „zerlegt“ und bereitgestellt werden, die den Lernenden helfen, durch eigene Erkenntnis- und Abstraktionsprozesse ihr individuelles Wissen zu bilden (siehe 2.3.4 „Aneignung von Wissen als Konstruktionsprozess“).

2.3.2 Verschiedene Arten von Wissen

Kompetente Personen verfügen über ein Netz von verschiedenen Arten von Wissen: *deklaratives*, *prozedurales*, *konditionales* und *metakognitives* Wissen [Anderson 2001, Mietzel 2003] (Tab. 2.1).

Art des Wissens	Beschreibung	Erklärung
Deklaratives Wissen	Wissen über Fakten, Konzepte und Regeln	Wissen dass... (<i>know-what</i>)
Prozedurales Wissen	Wissen über Abläufe und Prozesse	Wissen wie... (<i>know-how</i>)
Konditionales Wissen	Wissen über die situative Abhängigkeit von Handlungen	Wo und wann ist das Wissen zur Lösung des Problems einsetzbar? (<i>know-why</i>)
Metakognitives Wissen	Wissen und Kontrolle über das eigene Wissen und Strategien	Kenne ich zur Lösung des Problems eine geeignete Strategie und wie setze ich sie um? (<i>know what I do</i>)

Tab. 2.1: Verschiedene Arten von Wissen [Anderson 2001, Mietzel 2003].

Diese Wissensarten können zu verschiedenen Wissensdimensionen *Fakten und Konzepte*, *Fertigkeiten*, *Fähigkeiten* und *Metakognition* verknüpft werden (Abb. 2.3). Eine Wissensdimension kann aus mehreren Wissensarten bestehen. Zur Ausübung einer Fertigkeit kann beispielsweise sowohl deklaratives als auch prozedurales Wissen benötigt werden. Jede Wissensdimension hat eine deklarative Ebene.

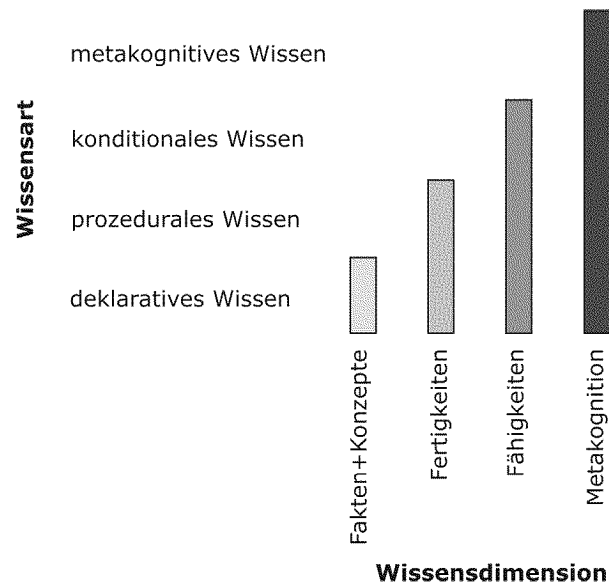


Abb. 2.3: Verknüpfung verschiedener Wissensarten zu verschiedenen Wissensdimensionen einer Kompetenz.

Im Folgenden soll auf die einzelnen Wissensdimensionen eingegangen und anschließend jede anhand eines Beispiels illustriert werden.

Dimension 1: Fakten und Konzepte (*Facts, Concepts*)

Es können verschiedene Formen von *deklarativem Wissen* unterschieden werden, dazu gehören: Wissen über Fakten, Konzepte und ihre Eigenschaften, semantische Beziehungen, Regeln und Metawissen. Faktenwissen hat die Form von isolierten Begriffen und Informationen (*wissen dass...*). Es entspricht der elementaren Einheit an Information, die Experten einsetzen, um über ihre akademische Disziplin zu kommunizieren und Information systematisch zu organisieren. Faktenwissen sind die Terminologien und Begriffe, welche die Lernenden kennen müssen, wenn sie in einer Disziplin Probleme kommunizieren und lösen wollen. Beim Konzeptwissen ist die Form der Information organisierter und verknüpfter als bei isoliertem Faktenwissen. Konzeptwissen umfasst einerseits Wissen über Kategorien, Klassifikationen und Beziehungen und andererseits generellere Prinzipien, Verallgemeinerungen, Modelle, Theorien und Strukturen [Alexander 1991, Anderson 2003].

Konzepte spielen eine wichtige Rolle bei der Organisation von Information für den Unterricht (siehe Abb. 2.2).

Dimension 2: Fertigkeiten (*Skills*)

Prozedurales Wissen ist das Wissen über Abläufe und Prozesse (*wissen wie...*). Es hat meistens die Form von Sequenzen und Schritten, die ausgeführt werden müssen. Prozedurales Wissen umfasst Wissen über Fertigkeiten, Algorithmen, Techniken und Methoden [Alexander 1991, Anderson 2003]. Diese Art von Wissen ist für das Ausführen von Handlungen wichtig. Es macht die Konzepte aus Dimension 1 anwendbar. Fertigkeiten sind häufig kontextspezifisch und an eine Disziplin gebunden.

Dimension 3: Fähigkeiten (*Capabilities*)

Konditionales Wissen betrifft die situative Abhängigkeit von Handlungen (*wo und wann ist das Konzept einsetzbar...?*). Es beinhaltet höhere Denkprozesse wie Problemlösen, logisches Denken, Fehlerbehebung, Umgang mit Komplexität etc. Die Entwicklung von Fähigkeiten ist von persönlichen Erfahrungen, aber auch von Reflexionen über die eigenen Denkprozesse abhängig.

Dimension 4: Metakognition (*Metacognition*)

Metakognitives Wissen ist das Wissen und die Kontrolle über das eigene geistige System [Favell 1979]. Lernende, die sich ihrer eigenen Gedanken bewusst sind, haben einen entscheidenden Vorteil: Sie erhalten dadurch die Möglichkeit, ihre Denkprozesse zu erkennen, zu kontrollieren und nötigenfalls zu korrigieren (*wissen was ich tue*). Zwei Arten von Metakognition werden unterschieden: Das metakognitive Wissen (deklarativ) und die metakognitive Kontrolle (exekutiv) [Kaiser 1999] (Abb. 2.4).

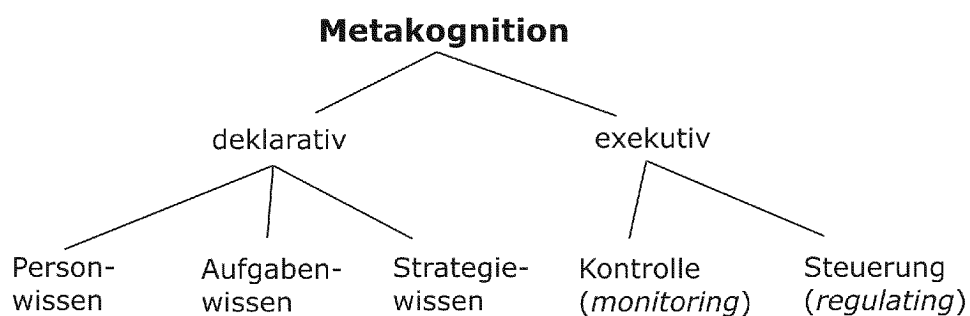


Abb. 2.4: Unterschied zwischen metakognitivem Wissen (deklarativ) und metakognitiver Kontrolle (exekutiv) [nach Kaiser 1999].

Personenwissen umfasst Wissen über das eigene Denken sowie Kenntnisse über das Denken anderer Menschen. Aufgabenwissen ist das Wissen über die Beschaffenheit von Informationen und über die spezifischen Anforderungen, die zu erfüllen sind (z. B. über den Schwierigkeitsgrad oder das Ziel einer Aufgabe). Strategiewissen erlaubt es, Lösungswege in ihrer Eignung für die Bewältigung einer Aufgabe einzuschätzen und alternative Lösungsmöglichkeiten in ihrer jeweiligen Wirksamkeit bewerten zu können.

Die aktive Kontrolle (*self-monitoring*) (wie weit bin ich mit der Aufgabe oder bin ich auf dem richtigen Weg?) und die zielgerichtete Regulation von kognitiven Prozessen (*self-regulation*) (Planung, Regulierung und Bewertung der Bearbeitungsaktivitäten) sind zwei Merkmale der metakognitiven Kontrolle.

Beispiel verschiedener Wissensdimensionen einer Kompetenz

Als nächstes sollen die Wissensdimensionen anhand der Kompetenz „Programmieren“ betrachtet werden. Bei den meisten Programmiersprachen kommt das Konzept der wiederholten Programmausführung durch Schleifen (*Loops*) vor. Die Behandlung dieses Konzepts ist deshalb auch Bestandteil fast jeder Programmierintroduction. Der kompetente Einsatz dieses Konzepts zur Lösung eines Programmierproblems erfordert die Aneignung von Wissen in allen Dimensionen.

Die *erste Dimension* (Fakten und Konzepte) des Konzepts „Schleifen“ umfasst das Wissen, was Schleifen sind und welche Bestandteile und Eigenschaften dazu gehören. Dieses Wissen sollte so generell sein, dass Schleifen auch in anderen Programmiersprachen erkannt und eingesetzt werden können. Dieses Wissen allein reicht aber nicht aus, um ein konkretes Problem zu lösen, das mit Hilfe einer Schleife in einer spezifischen Programmiersprache gelöst werden soll.

Die *zweite Dimension* (Fertigkeiten) des Konzepts „Schleife“ umfasst das Wissen, wie eine Schleife in einer bestimmten Programmiersprache geschrieben wird. Dieses Wissen ist zwar an die Syntax einer Programmiersprache gebunden, kann aber mit etwas Erfahrung auch auf eine andere Programmiersprache übertragen werden. Mit Hilfe des Wissens der deklarativen und prozeduralen Dimension können zwar kleine Probleme unter Anwendung des Konzepts gelöst werden, komplexere Probleme erfordern hingegen Wissen, das eine flexiblere Anwendung des Konzepts notwendig macht.

Die *dritte Dimension* (Fähigkeiten) des Konzepts „Schleife“ ist das Wissen, wann und wo bei einem gestellten Problem eine Schleife zum Einsatz kommt. Es setzt das Vorhandensein von Konzeptwissen und Fertigkeiten voraus. Schleifen treten dabei als Teil eines Algorithmus in Kombination mit anderen Konzepten (z.B. Kontrollvariablen, Bedingungsprüfungen usw.) auf, wie sie in realen Programmen vorkommen. Dieses Wissen beruht auf Erfahrungen mit ähnlichen Problemen und erfordert eine hohe Flexibilität im Umgang mit dem Konzept.

Die *vierte Dimension* (Metakognition) ist das Wissen über verschiedene Strategien, wie das Konzept „Schleife“ zur Lösung eines Problems eingesetzt werden kann. Es umfasst Wissen über die Kontrolle und die Regulation der Ausführung der Strategie der „Iteration“ zur Lösung eines Problems sowie die Überwachung des eigenen Verständnisses, während der Problemlösung. Dabei wird auch Wissen über eigene Stärken und Schwächen beim Anwenden der Strategie aktiviert sowie Wissen über die eigene Moti-

vation, das gestellte Problem fertig zu lösen. Dieses Wissen beruht unter anderem auf Erfahrung und der Reflexion vergangener Problemlösungsprozesse.

Wie bei der Kompetenz „Programmieren“ mit dem Konzept „Schleife“ verdeutlicht wurde, können die einzelnen Wissensarten nicht losgelöst voneinander betrachtet werden. Sie treten bei einer kompetenten Person beim Lösen eines Problems vielmehr als *Wissensnetz* auf. Deklaratives Wissen hat dabei den höchsten Allgemeingrad, prozedurales und konditionales Wissen sind häufig stark an den Kontext des Problems gebunden und metakognitives Wissen an die Person und die Bedingungen, die während der Kompetenzerneuerung herrschen.

2.3.3 Unterschiede bei der Aneignung verschiedener Wissensarten

Die Aneignung verschiedener Wissensarten einer Kompetenz ist mit einigen speziellen Eigenschaften der Informationsverarbeitung des Gehirns verbunden. Unter diesem Aspekt sollen folgende zwei Punkte betrachtet werden:

- 1. die Entstehung von *implizitem* und *explizitem Wissen*
- 2. die Aneignung von Wissen über Begriffe

1. Entstehung von implizitem und explizitem Wissen

Nicht jede Wissensart ist im gleichen Masse verbalisierbar. Das heisst, obwohl bestimmte Wissensarten angeeignet wurden und vorhanden sind, sind sie nicht bewusst und können nicht in Worte gefasst werden [Polanyi 1985]. Der Grund dafür ist, dass beispielsweise deklarative und prozedurale Information im Gehirn auf unterschiedliche Art und Weise verarbeitet werden. Prozedural gespeicherte Information macht generell mehr Mühe, in Sprache umgesetzt zu werden als deklarative Information. Sie steht als *implizites Wissen* zur Verfügung (Tab. 2.2).

	explizites Wissen	implizites Wissen
Wissensdimension	deklarativ	prozedural
Verbalisierbarkeit	gross	klein
Form	Fakten	Regeln
Entwicklung	sprunghaft (durch Ereignisse)	stetig (durch Üben)
Output dieses Wissens	Wiedergabe/Mitteilung	Anwendung/Können

Tab. 2.2: Die unterschiedliche Verarbeitung von deklarativer und prozeduraler Information im Gehirn führt zu implizitem und explizitem Wissen. Implizites Wissen kann sekundär zu explizitem Wissen werden (siehe auch Abb. 2.5).

Ein Beispiel für implizites Wissen sind Regeln der Grammatik der Muttersprache, die wir im Alltag fehlerfrei anwenden können, ohne dass wir die dazugehörigen Regeln explizit nennen können. Implizites Wissen ist primär dazu da, genutzt oder angewandt

zu werden. Bei dieser Art von Wissen steht das Können im Vordergrund, nicht dessen Verbalisierung. Unterschiede gibt es auch in der Geschwindigkeit, mit der sich diese beiden Wissensarten entwickeln. Während das explizite Wissen sprunghaft einsetzt, entwickelt sich das implizite stetig (Abb. 2.5). Wie kommen diese Unterschiede zu Stande?

Aus kognitionspsychologischer und neurobiologischer Sicht wird Lernen als aktive Verarbeitung von Information im dafür spezialisierten Organ „Gehirn“ gesehen, indem vor dem Hintergrund von Vorwissen neue Information verarbeitet wird. Dies führt zur Repräsentation von Information der Aussenwelt durch Neuronen [Spitzer 1996]. Dadurch soll ein Abbild der Umgebung geschaffen werden, mit dem Ziel, das Verhalten des Individuums erfolgreich zu steuern. Eine Voraussetzung dafür ist eine aktive Auseinandersetzung mit der Umgebung. Je länger, intensiver, vielfältiger und tiefer die Auseinandersetzung, desto differenzierter das Abbild der Umgebung.

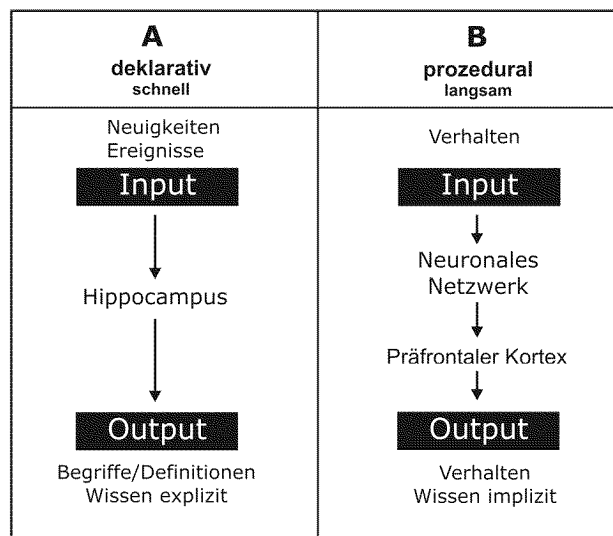


Abb. 2.5: Unterschiedliche Wege der Aneignung von deklarativem und prozeduralem Wissen. Während Neuigkeiten und Ereignisse schnell als deklarative Information abgespeichert und als explizites Faktenwissen zur Verfügung stehen (A), ist prozedurale Information als Synapsenstärken zwischen Neuronen im neuronalen Netzwerk des Gehirns gespeichert, die nur durch Üben langsam verändert werden können (B). Zudem ist das Wissen implizit gespeichert und muss durch spezielle Aktivitäten zugänglich gemacht werden.

Ereignisse und Fakten, die vom Gehirn als äusserst wichtig klassiert werden, können wir bereits nach einmaligem Hören oder Sehen behalten und stehen explizit zur Verfügung. Dabei spielt der Hippocampus eine wichtige Rolle bei der Überführung der Gedächtnisinhalte aus dem Kurz- ins Langzeitgedächtnis (Abb. 2.5, A) [Spitzer 2002]. Demgegenüber ist das Lernen *prozeduraler Information* ein langsamer Prozess. Diese Information ist im Gehirn in Form von Synapsenstärken zwischen Neuronen im neuronalen Netzwerk gespeichert (Abb. 2.5, B) [Spitzer 2002]. Bei einem bestimmten Input wird ein bestimmter Output produziert, ohne dabei eine explizite Regel zu verwenden. Durch wiederholtes Üben verändern sich die Synapsenstärken. Das Gehirn kann durch

das Verarbeiten von Beispielen Regelmäßigkeiten erkennen und daraus Regeln ableiten. Diese Regeln sind im präfrontalen Kortex der Grosshirnrinde repräsentiert und stehen zur Steuerung zukünftigen Verhaltens zur Verfügung. Implizites Wissen kann jedoch im Nachhinein zu explizitem Wissen werden, indem die Prozesse reflektiert und verbalisiert werden. Dieses deklarative Wissen ist mit prozeduralem verknüpft, was sich dadurch äussert, dass Regeln nicht nur wiedergegeben, sondern auch angewandt werden können.

2. Die Aneignung von Wissen über Begriffe

Aus der Sicht des Lernenden bedeutet ein Einstieg in ein Fachgebiet die Bewältigung einer grossen Flut an neuen Fachbegriffen. Diese miteinander in Beziehung zu bringen, fällt gerade Anfängerinnen und Anfängern schwer. Eine Verknüpfung auf einer rein deklarativen Ebene verunmöglicht es den Lernenden, Erfahrungen zu sammeln und erschwert eine spätere Anwendung. Die Fakten und Konzepte werden dadurch kontextunabhängig gelernt und können nicht verinnerlicht werden.

Begriffe werden am besten als kognitive Werkzeuge (*cognitive tool*) eingeführt, welche in einer bestimmten alltäglichen Situation verwendet werden können [Dewey 1991]. Dabei reicht es wie bei der Verwendung eines Werkzeuges nicht aus, wenn man weiss, *was* es ist, man benötigt zusätzlich wissen, *wie* man damit umgeht. Noch besser ist, wenn Begriffe aus einer realen Handlung selbst entwickelt werden können. Dabei wird sichergestellt, dass das *was* und das *wie* zusammengehören und dass das prozedurale und konditionale Wissen ebenfalls mitgelernt wird. Das Verständnis der Lernenden ist stets in ihre Erfahrungen eingebettet und enthält den Bezug zur konkreten Situation. Dieser Ansatz ermöglicht, dass sich die Vorstellung über einen Begriff entwickelt und zusammen mit Erfahrungen ausdifferenziert werden kann (siehe Beispiel „Schleifen“).

Es macht deshalb einen wesentlichen Unterschied, ob Begriffe bloss über die Aufnahme von Begriffsdefinitionen oder zusammen mit einer realen Handlung eingeübt werden. Im ersten Fall findet die Begriffsaneignung auf rein deklarativer Ebene statt. Der Zeitaufwand dafür ist zwar relativ klein, die Lernenden werden aber Schwierigkeiten haben, den Begriff auszdifferenzieren, und sie werden die Begriffe schneller wieder vergessen. Begriffe, die zusammen mit einer Handlung und einer konkreten Situation eingeübt werden, verknüpfen hingegen mehrere Wissensdimensionen. Die dafür aufgewendete Zeit ist zwar grösser, die Lernenden werden die Begriffe aber besser und länger behalten können.

Ein allgemeines begriffliches Verständnis über ein Ereignis oder einen Gegenstand, das auf der Grundlage vorausgegangener Erfahrungen entstanden ist, wird als so genanntes *Schema* im Gehirn abgelegt [Cohen 1989]. Schemata können als kognitive Repräsentationen von Begriffen verstanden werden. Sie sind kontextspezifisch, es spielt also eine Rolle, unter welchen Umständen ein Mensch dieses Wissen erworben hat. Es stellt stets eine Verringerung der Komplexität dar und ermöglicht dem Indivi-

duum, auf die äusserst komplexe Welt zu reagieren. Um den Widererkennungseffekt möglichst gross zu halten, wird vom Gedächtnis eine Art Prototyp eingesetzt. Auf diese Weise lernen auch Kinder ihre Umwelt in Sprache zu fassen, indem sie zunächst auf der Grundlage klarer Beispiele Begriffe lernen und diese als Prototypen speichern [Rosch & Lloyd 1978]. Diese Prototypen werden dann als Ausgangspunkt verwendet, um die Begriffe weiterzuentwickeln und auszudifferenzieren (siehe Kapitel 6.3.2 „Konstruktive Begriffsbildung“).

2.3.4 Aneignung von Wissen unterschiedlicher Dimensionen als Konstruktionsprozess

Kompetente Personen verfügen über Wissen verschiedener Dimensionen, die miteinander vernetzt und unterschiedlich bewusst sind. Die Aneignung dieses Wissens ist an einen *individuellen Konstruktionsprozess* gebunden. Die Lernenden verarbeiten Informationen aktiv, machen Erfahrungen, verbinden neue Information mit bereits bestehenden Wissensstrukturen und bauen diese schrittweise auf und aus. Die Perspektive, dass Wissen das Produkt eines aktiven und individuellen Konstruktionsprozesses ist, entspricht der *konstruktivistischen Philosophie*. Dabei interessiert weniger, wie Wissen vermittelt, sondern wie Wissen durch den Lernenden konstruiert wird. Die wichtigsten Prozessmerkmale konstruktivistischen Lernens sind in Tab. 2.3 zusammengefasst [Reinmann-Rothmeier & Mandel 1997].

Die Rollen von Lehrenden und Lernenden verändern sich durch dieses Paradigma grundlegend. Der konstruktivistische Aufbau von Wissen bedeutet für den Lernenden mehr Aktivität und Selbstverantwortung, für den Lehrenden das Einnehmen einer neuen Rolle. Statt zu lehren, müssen Strukturen, Informationen und Zeitgefässe zur Verfügung gestellt werden, die dem Lernenden einen Aufbau von Kompetenzen ermöglichen (siehe Kapitel 4 „Kompetenzorientierte Lernumgebungen“).

Prozessmerkmale des Lernens	Erklärung
Lernen als konstruktiver Prozess	Lernen ist in jedem Fall konstruktiv: Ohne den individuellen Erfahrungs- und Wissenshintergrund und eigene Interpretation finden keine kognitiven Prozesse statt.
Lernen als aktiver Prozess	Lernen geschieht über die aktive Beteiligung der Lernenden. Dazu gehört, dass die Lernenden an dem, was sie tun und wie sie es tun, Interesse haben oder entwickeln.
Lernen als selbstgesteuerter Prozess	Bei jedem Lernen übernehmen die Lernenden Steuerungs- und Kontrollprozesse, das Ausmass eigener Steuerung und Kontrolle variiert je nach Lernsituation.
Lernen als problemorientierter Prozess	Lernen erfolgt in spezifischen, auf das Anwendungsfeld bezogenen, situativen Kontexten, so dass jeder Lernprozess auch als situiert oder aufgrund der Offenheit der Fragen und des Kontexts als problemorientiert gelten kann.
Lernen als sozialer Prozess	Lernen ist ein sozialer Prozess. Lernen ist ein interaktives Geschehen in einer sozialen Gruppe, die als Erfahrungs- und Lerngemeinschaft fungiert.

Tab. 2.3: Prozessmerkmale des Konstruktivistischen Lernens [nach Reinmann-Rothmeiner & Mandel 1997]

2.4. Erhöhen der Unterrichtsqualität durch die Kompetenzorientierung

Die beiden Hauptgründe, weshalb der Aufbau von Kompetenzen im Grundlagenunterricht von Interesse ist, stehen im Zusammenhang mit den zwei wichtigsten Qualitätsaspekten des Unterrichts: der *Nachhaltigkeit* und dem *Transfer* von Wissen.

2.4.1 Nachhaltigkeit und Transfer des Wissens

Nachhaltigkeit des Wissens

Unter nachhaltigem Wissen verstehen wir Wissen, welches über die Dauer des Unterrichts hinaus Bestand hat und auch nach längerer Zeit noch vorhanden ist. Bei Konzepten und Begriffen, die bloss auswendig gelernt und wiedergegeben werden, ist die Nachhaltigkeit nicht sehr gross. Wissen ist nur nachhaltig, wenn die gelernten Konzepte von den Lernenden auch nach Abschluss einer Lerneinheit weiterentwickelt werden können. Dafür brauchen Konzepte für eine Zielgruppe eine gewisse Relevanz und einen sinnvollen Kontext, den die Zielgruppe mit grosser Wahrscheinlichkeit in ihrem (späteren) Alltag auch antrifft. Zudem brauchen die Lernenden Fähigkeiten und Strategien, um die Konzepte in ihrem Kontext zu erkennen und nötige Informationen zu beschaffen, um ihr individuelles Wissen weiterzuentwickeln. Damit Konzepte selb-

ständig weiterentwickelt werden können, müssen sie mit einer bestimmten kontextuellen Verankerung verarbeitet und verinnerlicht werden.

Unter dem Begriff des *situierten Wissens* ist die Nicht-Trennbarkeit von Wissen und Anwendung gemeint [Collins 1989]. Wissensinhalte sind immer bis zu einem bestimmten Grad an einen zeitlichen, räumlichen und örtlichen Kontext gebunden, in dem sie erworben worden sind. Wissen, welches in einen Kontext gesetzt wurde, kann im Gedächtnis besser gespeichert und abgerufen werden als Wissen, das ohne Kontext erworben wurde. Das bedeutet, dass das Aneignen von Konzepten dann nachhaltiger wird und weniger träge bleibt, wenn sie in der Umgebung eingeübt werden, in der sie in der Realität auch vorkommen [Reusser 2005].

Transfer von Wissen

Unter Transfer von Wissen (*knowledge transfer*) verstehen wir die Fähigkeit, Wissen zur Lösung von Problemen der realen Welt einzusetzen, um neue Fragen zu beantworten oder in einem Gebiet weiterzulernen [Mayer & Wittrock 1996]. Die Lernenden sollen nicht bloss wiedergeben können, was sie gelernt haben, sondern die Bedeutung ihres angeeigneten Wissens kennen und fähig sein, das Gelernte zur Lösung von Problemen einzusetzen.

Die Schwierigkeit besteht darin, dass in der realen Welt die Konzepte nicht nur in Beziehung mit anderen Konzepten stehen, sondern dass der Kontext, in dem das Konzept auftritt, ebenfalls analysiert und bewältigt werden muss. Neben Konzeptwissen und Fertigkeiten spielen beim Lösen der Probleme Fähigkeiten und metakognitives Wissen eine zentrale Rolle.

Die Lernenden müssen zusätzlich zu den Konzepten den Umgang mit der Komplexität der realen Welt erlernen. Für die Lehrenden steht nicht die Darstellung der Komplexität, sondern der Aufbau einer Kompetenz zur Bewältigung der Komplexität im Zentrum. Dieser Ansatz setzt beim Lernenden Individuum an. Oft werden Konzepte unterrichtet, als handle es sich um eine Wahrheit, die nur in einer abstrakten Sprache vermittelbar ist. Aus Konzepten werden aber nur Kompetenzen, wenn sie von den Lernenden durch eigene Erkenntnis- und Abstrahierungsprozesse selber konstruiert werden.

2.4.2 Nachhaltiger, transferfreundlicher Unterricht

Marini & Genereux nennen drei wesentliche Elemente eines nachhaltigen und transferfreundlichen Unterrichts (Tab. 2.4) [Marini & Genereux 1995]. Während der erste Punkt vor allem damit zu tun hat, auf welche Art und Weise Konzepte im Unterricht verarbeitet worden sind, spielen beim zweiten und dritten Punkt die Eigenschaften der zu lösenden Probleme und die Fähigkeiten der Lernenden im Umgang mit Problemen eine wesentliche Rolle.

Transfer-Element	Erklärung
Vermittelte Konzepte und ihre Verarbeitungstiefe	Mit welcher Verarbeitungstiefe müssen Konzepte von den Lernenden verarbeitet worden sein, damit sie später angewandt werden können?
Die Eigenschaft der zu lösenden Probleme und ihr Kontext	Eignen sich die gewählten Probleme, damit die Lernenden die zugrunde liegenden Prinzipien und Konzepte erkennen können?
Lernende und ihre Problemlöse-Fähigkeiten	Bringen die Lernenden die Wissensstrukturen in der nötigen Vernetztheit und hinreichend Motivation mit, um ein Problem lösen zu können?

Tab. 2.4: Elemente eines transferfreundlichen Unterrichts [Marini & Genereux 1995].

Um einen Transfer von Konzepten erreichen zu können, müssen sie zuvor mit einer bestimmten Tiefe, die Erkenntnis und eigenaktives Abstrahieren miteinschließt, verarbeitet worden sein. Der Unterricht muss so gestaltet werden, dass die Lernenden sich transferfähiges Konzeptwissen aneignen können (siehe kognitive Prozessdimensionen im Abschnitt 2.5.2).

Transfer bedeutet, dass die Lernenden ihr erworbenes Wissen einsetzen können, um Probleme zu lösen. Ist das Lösen von Problemen bereits zentrales Element des Unterrichts, wird auch der „Praxischock“ weniger stark ausfallen. Die Lernenden sind sich nicht nur gewöhnt, Probleme in einem Fachbereich anzugehen, sie werden später die gelernten Konzepte auch besser anwenden können.

Die Lernmethode des *Problembasierten Lernens* integriert das Lösen von Problemen in den Unterricht. Der Aufbau einer Wissensbasis wird dabei vom späteren Anwendungsfeld nicht losgelöst, was den Transfer von Wissen unterstützen und die Entstehung von tragem Wissen verhindern soll. Das Problembasierte Lernen ist Thema des Kapitels 5.

2.5. Kompetenzniveaus

Im Folgenden wollen wir uns verschiedenen Kompetenzniveaus zuwenden, die bei einer bestimmten Kompetenz erreicht werden können. Als erstes wollen wir zwischen einer Kurs- und einer Fachkompetenz unterscheiden (Abschnitt 2.5.1), dann soll ein Mass zur Bestimmung verschiedener Kompetenzniveaus beigezogen werden (Abschnitt 2.5.2).

2.5.1 Kurskompetenz und Fachkompetenz

Während eine Fachkompetenz meist über Jahre aufgebaut wird und deshalb mit viel praktischer Erfahrung verbunden ist, kann eine Kompetenz, die in einem Kurs von einem Semester Dauer (in der Regel etwa 14 Wochen) erworben wurde, nicht das selbe Niveau aufweisen. Das Zeitvolumen, welches im Unterricht zur Aneignung einer Kom-

petenz zur Verfügung steht, ist in jedem Fall begrenzt. Die *Kurskompetenz* entspricht daher jener Kompetenz, welche in der zur Verfügung stehenden Zeit von den Kursteilnehmerinnen und -teilnehmern aufgebaut werden kann (Abb. 2.6). Die Konzepte, die in einem Kurs zum Aufbau einer Kurskompetenz zum Einsatz kommen, werden Schlüsselkonzepte (*Key-concepts*) genannt. Ein Kompetenzlevel, der in einem Kurs angeeignet wurde, kann nie dem Level einer Fachkompetenz gleichkommen. So gesehen kann in einem Kurs ein vollständiger Transfer von Wissen gar nicht erreicht werden. Die Kurskompetenz soll aber ausreichen, um bestimmte Aufgaben auf einem definierten Kompetenzlevel (siehe nächster Abschnitt) selbständig auszuführen, und eine gute Ausgangslage darstellen, um die aufgebaute Kompetenz nach dem Kurs weiterzuentwickeln. Der Output des Unterrichts sollte sich deshalb an einem maximalen, aber realistischen Kurskompetenzlevel orientieren, der eine gute Grundlage dafür bildet, zusammen mit praktischer Erfahrung und ausgewählten Weiterbildungen einen Fachkompetenzlevel zu erreichen.

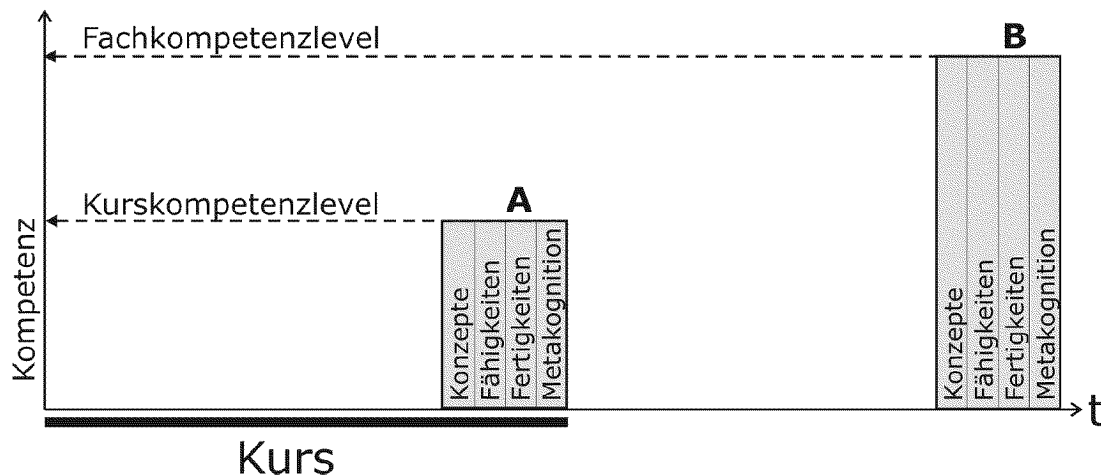


Abb. 2.6: Unterschied zwischen Kurs- (A) und Fachkompetenz (B). Beide bestehen aus einem Netz von Konzepten, Fähigkeiten und Fertigkeiten, sie unterscheiden sich jedoch in der Fähigkeit, ein Problem eines bestimmten Kompetenzlevels lösen zu können.

2.5.2 Kognitive Prozess-Dimensionen einer Kompetenz

Um im Unterricht Nachhaltigkeit und Transfer zu erreichen, müssen kognitive Prozess-Dimensionen erreicht werden, die über die Wiedergabe von Unterrichtsinhalten hinausgehen. Als Mass für solche Prozess-Dimensionen eignet sich Blooms *taxonomy of educational goals* [Bloom 1956], die durch Anderson und seine Mitarbeiter erweitert wurde [Anderson 2001]. Nach dieser Taxonomie können Lernende sechs kognitive Prozess-Dimensionen (K-Dimensionen) erreichen: *erinnern*, *verstehen*, *anwenden*, *analysieren*, *evaluieren* und *kreieren* (Tab. 2.5).

	Kategorie	Unterkategorien	Erklärung
K1	erinnern (<i>remember</i>)	erkennen wiedergeben	Abrufen von relevanter Information aus dem Gedächtnis
K2	verstehen (<i>understand</i>)	interpretieren klassieren veranschaulichen zusammenfassen vergleichen erklären	Konstruieren von Sinngehalt aus Instruktionen-Informationen
K3	anwenden (<i>apply</i>)	ausführen implementieren	Ausführen oder anwenden in einer gegebenen Situation
K4	analysieren (<i>analyze</i>)	differenzieren organisieren dekonstruieren	Material in seine Einzelteile zerlegen und ermitteln, wie die Teile zusammengehören
K5	evaluieren (<i>evaluate</i>)	kontrollieren kritisieren	Anhand von Kriterien und Standards Entscheidungen treffen
K6	kreieren (<i>create</i>)	Hypothesen bilden planen konstruieren	Elemente zusammensetzen, um ein funktionierendes Ganzes zu formen, Elemente in neue Muster und Strukturen reorganisieren

Tab. 2.5: Blooms kognitive Prozess-Dimensionen nach Anderson und Mitarbeiter [Anderson 2001].

Die Prozess-Dimensionen, die als Handlung auf einem bestimmten Kompetenzlevel betrachtet werden, sind der Output des Unterrichts. Sie sind nicht mit Unterrichtsaktivitäten zu verwechseln. Die Taxonomie ist aufbauend. Je höher die Taxonomiestufe, desto mehr Aufwand ist für die Konstruktion des damit verbundenen Wissensnetzes aufzuwenden. K1 ist das Abrufen von Information aus dem Gedächtnis, die Levels K2 bis K6 betreffen den Transfer von Wissen.

Je nach Taxonomiestufe ändert sich auch der Umgang mit den Konzepten. Konzepte dienen den Lernenden beim Aufbau von Kompetenzen als Gerüst zum Aneignen und zum Transfer von langlebigen, flexibel einsetzbaren Wissensstrukturen. Sie müssen deshalb als aktives Wissen unterschiedlicher Dimensionen zur Lösung von neuen Problemen eingesetzt werden können. Je höher der K-Level, desto mehr Flexibilität ist im Umgang mit einem Konzept erforderlich.

Die Kombination der verschiedenen kognitiven Prozess-Dimensionen mit den Wissensdimensionen Faktenwissen, Konzeptwissen, Prozedurales Wissen und Metakognitives Wissen ergibt folgende Taxonomie-Tabelle (Abb. 2.7):

Kognitive Prozesse

		Erinnern K1	Verstehen K2	Anwenden K3	Analysieren K4	Evaluiieren K5	Kreieren K6	
Wissensdimension	Faktenwissen	A		B		C		explizit
	Konzeptwissen							
	Prozedurales Wissen							implizit
	Metakognitives Wissen							
		<i>bleibt träge</i>		<i>transferfähig</i>		<i>längerfristige Zielsetzung</i>		
		Individuelle Erkenntnisse						
		Anwendungskontext						

Abb. 2.7: Taxonomie-Tabelle nach Anderson [abgeändert nach Anderson 2001]. Im Unterricht können verschiedene Kompetenzlevels mit Wissen unterschiedlicher Dimensionen erreicht werden. Bereich A: Zeitraum Minuten bis Stunden. Das Wissen auf den Levels K1 und K2 bleibt beim späteren Anwenden träge. Bereich B: Zeitraum Stunden bis Wochen. Das Wissen auf den Kompetenzlevels K3 und K4 ist transferfähig. Bereich C: Zeitraum Monate bis Jahre. Das Wissen auf den Levels K5 und K6 dient als längerfristige Zielsetzung in Studium oder Ausbildung. Zur Aneignung von Wissen auf den Levels K2 und höher sind individuelle Erkenntnisse notwendig. Zur Aneignung von K3 und höher ist ein Anwendungskontext notwendig.

Beim Unterricht, der auf die Kompetenzlevels K1 und K2 abzielt, bleibt das Wissen bei der späteren Anwendung träge, weil es zu wenig vernetzt und angewandt werden muss (Abb. 2.7, Bereich A). Die benötigte Zeit, um einen bestimmten Level zu erreichen, nimmt von Level zu Level zu. Um einen Kompetenzlevel grösser als K2 zu erreichen, sind individuelle Erkenntnisprozesse notwendig, ab Level K3 braucht es einen Anwendungskontext, um das nötige Wissen anzueignen. Kompetenzlevels über K4 würden zum Aufbau so viel Zeit in Anspruch nehmen, dass sie in einem Einführungskurs von ein bis zwei Semestern kaum erreicht werden können. Diese Kompetenzlevels eignen sich als Zielsetzung für eine ganze Ausbildung oder ein ganzes Studium und können erst auf der Stufe Master- oder Diplomarbeit erreicht werden (Abb. 2.7, Bereich C). Für Einführungskurse steht deshalb das Erreichen der Kompetenzlevels K3 und K4 im Zentrum des Interesses, weil sie einerseits transferfähiges Wissen hervorbringen und vom Zeitaufwand her innerhalb von Wochen bewältigt werden können (Abb. 2.7, Bereich B).

Die Wissensdimensionen, die für jeden kognitiven Level erreicht werden können, unterscheiden sich in ihrer Verbalisierbarkeit (siehe Abschnitt 2.3.3 implizites und explizites Wissen). Das bedeutet, dass eine Wissensdimension auf einem bestimmten kognitiven Level zwar erreicht werden kann, aber ohne dass das Wissen explizit zur Verfügung steht. Das Wissen dieser Dimension kann mittels geeigneter Methoden zu explizitem Wissen werden (siehe Kapitel 7 „Didaktische Modelle“).

Die Taxonomie-Tabelle von Anderson gibt Auskunft über die Qualität, mit der Konzepte unterrichtet werden. Weil das Erreichen höherer Kompetenzlevels im Unterricht mehr Zeit in Anspruch nimmt, kann nicht dieselbe Anzahl Konzepte unterrichtet werden wie im traditionellen, informellen Unterricht. Das bedeutet, dass für den K3/K4-basierten Unterricht eine Auswahl an Konzepten getroffen werden muss. Über die Auswahl gibt die Taxonomie keine Auskunft. Diese ist das Produkt von fachlichen Überlegungen. Konzepte sollten so ausgewählt werden, dass der Unterricht Kompetenzen hervorbringt, die für eine bestimmte Zielgruppe einen Sinn ergeben. Die Auswahl von Konzepten zum Aufbau von ICT-Kompetenzen ist Thema des nächsten Kapitels.

2.6. Fazit: Modell für einen kompetenzorientierten Unterricht

Unterricht kann nur nachhaltiges, transferfähiges Wissen hervorbringen, wenn es im Zusammenhang mit der Aneignung von Kompetenzen steht. Kompetenzen lassen sich, wie Wissen auch, nicht direkt von Individuum zu Individuum weitergeben. Deren Aufbau ist von individuellen Konstruktions- und Erkenntnisprozessen abhängig. Deshalb rückt die Entwicklung *kompetenzorientierter Lernumgebungen* ins Zentrum des Interesses, mit deren Hilfe die Lernenden definierte Kompetenzen individuell aufbauen können (Abb. 2.8). Die Lernumgebung hat zum einen die Aufgabe, die Aneignung von Wissensnetzen zu ermöglichen, und zum anderen, das gelernte Wissen in die reale Welt zu transferieren. Um dies zu erreichen, muss eine Auswahl an Konzepten zunächst handlungsbasiert eingeübt werden (Abb. 2.8, A) und anschliessend in einer realen Umgebung angewandt werden (Abb. 2.8, B). Die durch Handlungen entstehenden Lernprodukte dienen einerseits dazu, eigene Fortschritte sichtbar und erlebbar zu machen, und bilden andererseits die Grundlage für eine Reflexion des Gelernten, um auch implizites Wissen zugänglich zu machen.

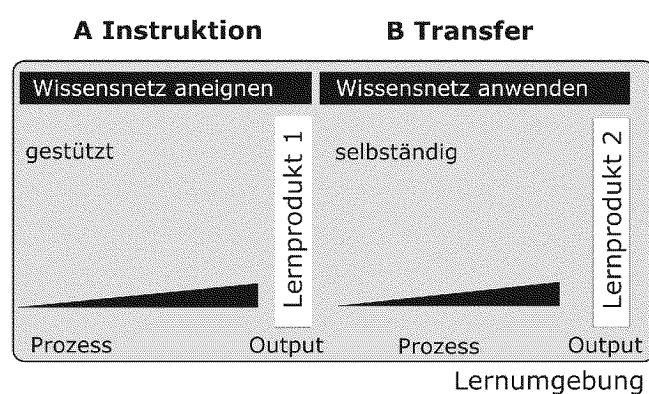


Abb. 2.8: Modell für einen kompetenzorientierten Unterricht. Während der Instruktion (A) steht die Aneignung eines Wissensnetzes im Zentrum und während der Anwendung der Transfer (B). Die durch Handlungen entstehen Lernprodukte dienen als Grundlage für eine Reflexion des Gelernten.

ICT-Kompetenzen für die wissenschaftliche Tätigkeit

3.1. Informatik als Enabling-Science in der wissenschaftlichen Tätigkeit

Die wissenschaftliche Tätigkeit innerhalb einer modernen Wissensorganisation wie zum Beispiel den Naturwissenschaften erfordert heute professionelle Kompetenzen im Umgang mit digitalen Daten und Informationen. Während Kompetenzen im Umgang mit digitaler Information vor allem durch den Einsatz von Informations- und Kommunikations- Technologien (ICT) geprägt sind, macht der professionelle Umgang mit digitalen Daten die Anwendung von Konzepten und Methoden der Informatik notwendig. Für die Ausübung dieser wissenschaftlichen Tätigkeit nimmt die Informatik die Stellung einer *Enabling-Science* ein (Abb. 3.1).

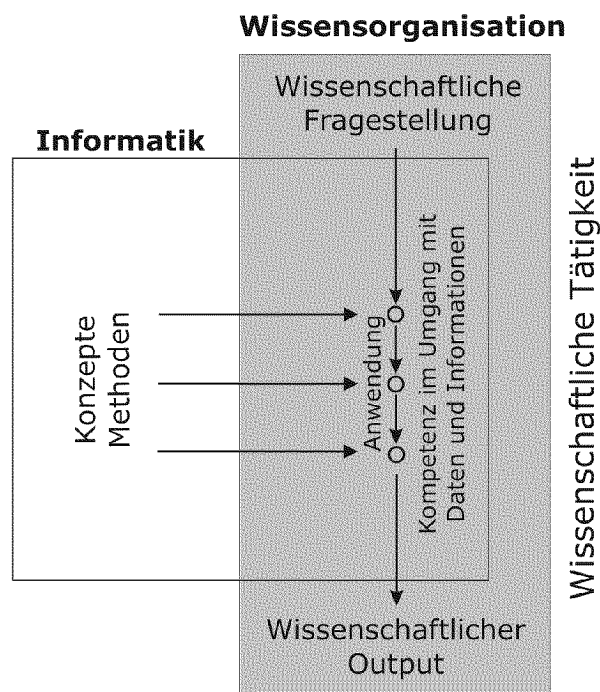


Abb. 3.1: Informatik als Enabling-Science für die wissenschaftliche Tätigkeit innerhalb einer Wissensorganisation (z.B. Naturwissenschaften): Das Beantworten einer wissenschaftlichen Fragestellung erfordert Kompetenzen für den professionellen Umgang mit digitalen Daten und Informationen. Diese Kompetenzen umfassen die Anwendung von Konzepten und Methoden der Informatik.

Wie im Kaptitel 2 ausführlich beschrieben, erfordert die Aneignung von Kompetenzen eine realistische und authentische Umgebung, weil durch den Realitätsbezug verschiedene Wissensarten als Einheit gelernt und reale Handlungen erzeugt werden können.

In diesem Kapitel gehen wir folgenden Fragen nach:

- Welche Konzepte, Fertigkeiten und Fähigkeiten sind zur Aneignung einer ICT-Kompetenz für die wissenschaftliche Tätigkeit von Interesse? (Abschnitt 3.2)
- Wie kann zur Vermittlung, respektive Aneignung von ICT-Kompetenzen für die wissenschaftliche Tätigkeit ein realistischer Kontext hergestellt werden? (Abschnitt 3.3)
- Welche Ressourcen (Hardware, Kompetenzen) können bei den Studierenden im universitären Bereich vorausgesetzt werden? (Abschnitt 3.4)

3.2. Kompetenzen im Umgang mit digitalen Daten und Informationen

In der wissenschaftlichen Tätigkeit entsteht neue Information und Wissen, indem Daten erhoben und interpretiert werden (siehe auch Kapitel 2 „Kompetenzorientierter Unterricht“). Die Ausübung der wissenschaftlichen Tätigkeit ist somit massgeblich von fachunabhängigen Kompetenzen auf der Ebene von Information und Daten abhängig (Abb. 3.2).

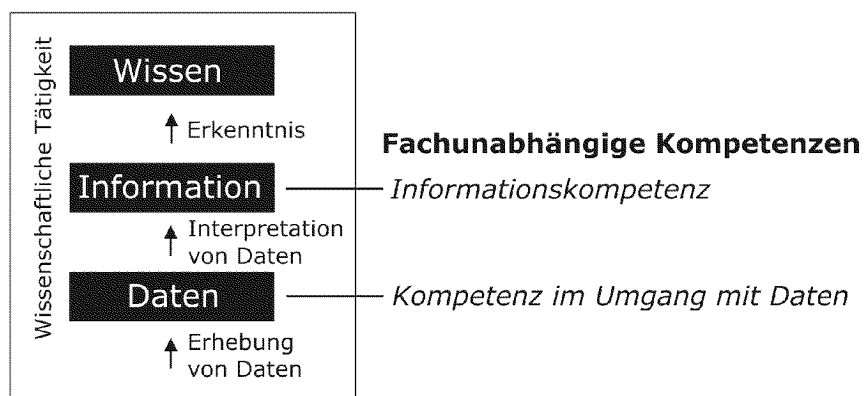


Abb. 3.2: Fachunabhängige Kompetenzen beim Aufbau von Wissen in der wissenschaftlichen Tätigkeit (z.B. den Naturwissenschaften): Das Erheben und interpretieren von Daten führt zu neuer fachspezifischer Information und Wissen. Hierfür sind professionelle Kompetenzen im Umgang mit digitalen Daten unter dem Einsatz von Konzepten und Methoden der Informatik notwendig.

1. Informationskompetenz

Seit Informationen immer häufiger in digitaler Form vorliegen, hat sich der Fokus der Informationskompetenz von eigentlichen „Bibliotheks-Fertigkeiten“ auf Fähigkeiten und Kompetenzen im Umgang mit dem global vernetzten Computer verlagert. Kompetenzen im Umgang mit Information gehören in einer Wissensgesellschaft zu den Schlüsselkompetenzen (Abb. 3.2, Ebene Information).

2. Kompetenz im Umgang mit Daten

Die wissenschaftliche Tätigkeit erfordert neben der Kompetenz im Umgang mit Information die Aneignung zusätzlicher Kompetenzen. Sie umfassen den Umgang mit digitalen Daten (Abb. 3.2, Ebene Daten). Während der Umgang mit digitaler Information heute weitgehend ohne Informatik-Konzepte bewältigt werden kann, sind für den Umgang mit digitalen Daten Verständnis und Anwendung von Konzepten und Methoden der Informatik erforderlich.

Die wissenschaftliche Tätigkeit verlangt Kompetenzen, welche beispielsweise die Verwaltung, Strukturierung und Interpretation von Daten erfordern. Zur Ausübung dieser Kompetenzen, ist die Aneignung eines umfassenden und flexiblen Wissensnetzes unter Anwendung von Konzepten erforderlich.

3.2.1 Information-Literacy: Kompetenz im Umgang mit Information

Von der American Library Association wurde die Informationskompetenz (engl. Information-Literacy) wie folgt definiert [ALA 1989]:

To be information literate, a person must be able to recognize when information is needed and have the ability to locate, evaluate and use effectively the needed information.

Die Informationskompetenz umfasst eine Reihe von fachunabhängigen Fähigkeiten, welche den kompetenten, effizienten und verantwortungsbewussten Umgang mit Information ermöglichen. Dazu gehört nicht nur die selbstorganisierte Erschließung von Wissen durch Information, sondern auch das Erweitern vorhandener Fähigkeiten sowie die zielgruppenorientierte Kommunikation der Information. Tab. 3.1 fasst einzelne Komponenten der Information-Literacy zusammen [Educational Testing Services ETS 2005]:

Fähigkeit	Definition
Erkennen (<i>define</i>)	Fähigkeit, Informationsbedarf zu erkennen.
Lokalisieren (<i>access</i>)	Fähigkeit, Information zu lokalisieren.
Organisieren (<i>manage</i>)	Fähigkeit, Information zu organisieren.
Integrieren (<i>integrate</i>)	Fähigkeit, Information zu interpretieren. Dazu gehört die Synthese, das Zusammenfassen und Vergleichen von Information verschiedener Quellen.
Evaluieren (<i>evaluate</i>)	Fähigkeit, Information zielgerichtet nach spezifischen Kriterien zur Bewältigung der Problemstellung zu selektieren.
Anwenden (<i>create</i>)	Fähigkeit, Information anzupassen und anzuwenden.
Kommunizieren (<i>comunicate</i>)	Fähigkeit, Information gezielt und zielgruppenorientiert zu kommunizieren.

Tab. 3.1: Komponenten der Information-Literacy [Educational Testing Services ETS 2005].

Die einzelnen Fähigkeiten sind jeweils abhängig von einem Kontext und setzen ein fundiertes Wissen und Erfahrung in einem Fachbereich voraus. Aus diesem Grund geht die Aneignung solcher fachunabhängiger Fähigkeiten im Umgang mit Information auch stets mit der Aneignung von Wissen und Kompetenzen in einem Fachbereich einher [ALA 1989].

Wissen verändert sich und wächst innerhalb eines Fachbereichs sehr schnell. Damit Mitglieder einer Wissensorganisation (Wissenschaftlerinnen und Wissenschaftler, Fachlehrerinnen und Fachlehrer usw.) diese Schnelllebigkeit bewältigen können, stellen Fähigkeiten im Umgang mit Information eine Kernkompetenz dar und liefern die Voraussetzung für ein lebenslanges Lernen in einem Fachbereich.

ICT-Literacy: Kompetenzen im Umgang mit digitaler Information

Der Umgang mit digitaler Information unter dem Einsatz von Informations- und Kommunikations- Technologien (*ICT-Literacy*) stellt eine Spezialisierung der Information-Literacy dar. Durch die Möglichkeiten, mit dem Computer Information zu manipulieren und kommunizieren, hat die ICT-Literacy eine neue Dimension erhalten. Der professionelle Umgang mit Information und Technologie stellt neue Anforderungen an die individuellen kognitiven Fähigkeiten und Fertigkeiten, um gezielt individuelle Probleme lösen zu können. Für die ICT-Literacy ist man heute jedoch kaum noch gezwungen, Informatik-Konzepte zu lernen und anzuwenden. Sie kann grösstenteils mit der Bedienung von Texteditoren, E-Mail und Webbrowsern ausgeübt werden.

3.2.2 Professioneller Umgang mit digitalen Daten

Daten als Grundlage von Information können als „Rohstoff“ der Wissenschaft zur Erarbeitung neuer Erkenntnisse betrachtet werden (folglich Abb. 3.2). Für Wissenschaftlerinnen und Wissenschaftler gehört deshalb der professionelle Umgang mit Daten zu den Grundkompetenzen ihrer Tätigkeit. Seit der Digitalisierung und Entwicklung neuer Messmethoden sind Wissenschaftlerinnen und Wissenschaftler nicht nur mit sehr grossen Datenmengen konfrontiert, sie haben auch Konsequenzen für die Anforderungen an ihre Fähigkeiten und Fertigkeiten in der computergestützten Datenverarbeitung.

In Abb. 3.3 sind einzelne ausgewählte Kompetenzen im Umgang mit digitalen Daten dargestellt. Diese reichen vom Erfassen der Daten bis hin zu deren Einsatz im wissenschaftlichen Output. Heutige Messgeräte liefern häufig nur noch Daten in digitaler Form (Abb. 3.3, A). Um die dadurch erhaltenen grossen Datenmengen bewältigen zu können, sind professionelle Konzepte der Datenverwaltung notwendig (B). Für die Analyse (C) und das Visualisieren von Daten (D) sowie das Simulieren von Prozessen (E) sind spezialisierte Informatikmittel einzusetzen, welche über die Verwendung von Standardsoftware hinausgehen. Um den Funktionsumfang von Informatikmitteln anzupassen und zu erweitern, sind Programmierkenntnisse erforderlich und hilfreich (siehe weiter unten).

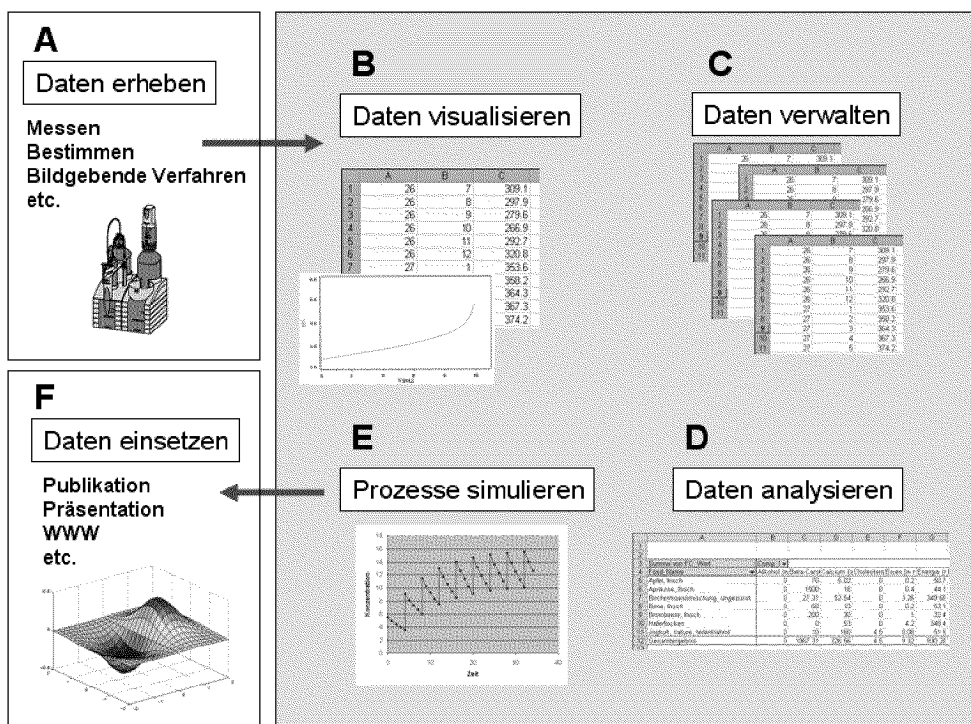


Abb. 3.3: Kompetenzen beim professionellen Umgang mit digitalen Daten in der wissenschaftlichen Tätigkeit. Daten werden in digitaler Form erhoben (A), visualisiert (B), verwaltet (C), analysiert (D) sowie als Prozesse simuliert (E), um als wissenschaftlicher Output in unterschiedlichster Form als Publikation, Präsentation, im World Wide Web usw. (F) eingesetzt werden zu können.

3.2.3 Computer-Fluency

Für die Kompetenz im Umgang mit dem Computer führte Lawrence Snyder den Begriff der *Computer-Fluency* ein, die weiter und tiefer gehen soll als die ICT-Literacy [Snyder 2004]. Er ist der Meinung, dass der professionelle Umgang mit digitalen Daten und ICT ein Verständnis von elementaren Informatik-Konzepten zusammen mit ausgewählten Fertigkeiten und Fähigkeiten erfordert. Die Entwicklung einer Computer-Fluency ist ein zeitaufwändiger Prozess, der über ein Studium hinaus zu verfolgen ist und danach stetig weiterentwickelt werden muss. Weil Informationstechnologie sehr schnellen Änderungen unterworfen ist, ist eine Orientierung an langlebigen, grundlegenden Ideen und Konzepten notwendig, die möglichst von Technologie und Anwendung unabhängig sind. Sie sollen einerseits helfen, in Zukunft neue, veränderte Fertigkeiten und Fähigkeiten schneller zu erlernen, und andererseits die Grundlage für eine Flexibilisierung des individuellen Wissens bilden.

Im Folgenden soll auf Konzepte, Fertigkeiten und Fähigkeiten im Umgang mit dem Computer eingegangen werden die vom *Comitee on Information Technology Literacy* entwickelt wurde [National Research Council (U.S.) Committee on Information Technology Literacy 1999].

1. Konzepte

Tab. 3.2 zeigt eine Reihe von Konzepten (*Concepts*) der Computer-Fluency [für Details siehe National Research Council (U.S.) Committee on Information Technology Literacy 1999]. Die Vermittlung und Aneignung der Konzepte der Liste ist sehr anspruchsvoll und kann je nach Komplexität des Konzepts Jahre in Anspruch nehmen (folglich Kapitel 2 „Kompetenzorientierter Unterricht“). Die Idee ist nicht, dass die Konzepte in einem Einführungskurs abschliessend behandelt werden. Sie sollen vielmehr dem Lernprozess während eines Kurses eine Richtung und eine Idee geben, um auch nach einem Kurs weiterentwickelt werden zu können. Sie bilden die Grundlage für Problemlöseprozesse im Umgang mit digitalen Daten.

Konzepte der Computer-Fluency	
1.	Grundlagen von Computern und Programmen
2.	Organisation von Informationssystemen
3.	Grundlagen von Netzwerken
4.	Digitale Repräsentation von Information
5.	Organisation von Information
6.	Modellieren und Abstraktion
7.	Algorithmisches Denken und Programmierung
8.	Universalität
9.	Limiten der Informationstechnologie
10.	Gesellschaftliche Folgen von Informationstechnologie

Tab. 3.2: Konzepte der Computer-Fluency nach Snyder [für Details siehe National Research Council (U.S.) Committee on Information Technology Literacy 1999].

2. Fertigkeiten

Fertigkeiten (*Skills*) der Computer-Fluency umfassen den Umgang mit Computern und Informatikmitteln. Weil Fertigkeiten an konkrete Applikationen gebunden sind, verändern sich diese auch mit deren Weiterentwicklung. Die Fähigkeit zur Aneignung von neuen Fertigkeiten ist für die wissenschaftliche Tätigkeit heute von grosser Wichtigkeit. In Tab. 3.3 sind Fertigkeiten der Computer-Fluency aufgelistet. Die Aneignung und Ausübung der Fertigkeiten 1 bis 7 kann heute weitgehend ohne Informatik-Konzepte bewältigt werden. Für die Fertigkeiten 8 bis 10 ist hingegen die Aneignung und Anwendung von Informatik-Konzeptwissen erforderlich.

Fertigkeiten der Computer-Fluency	
1.	Einen Computer in Betrieb nehmen.
2.	Bedienung eines Betriebssystems.
3.	Mit Hilfe einer Textverarbeitung ein Text-Dokument erstellen.
4.	Mit Hilfe geeigneter Software aussagekräftige Diagramme und Illustrationen erstellen.
5.	Ein Computer mit einem Netzwerk verbinden.
6.	Mit Hilfe des Internets Informationen suchen.
7.	Den Computer zum Kommunizieren einsetzen.
8.	Eine Tabellenkalkulation einsetzen, um Prozesse zu modellieren.
9.	Eine Datenbank einsetzen, um Information zu verwalten.
10.	Ein vorhandenes Programm eigenen Bedürfnissen anpassen.

Tab. 3.3: Fertigkeiten der Computer-Fluency nach Snyder [für Details siehe National Research Council (U.S.) Committee on Information Technology Literacy 1999].

3. Fähigkeiten

Die Fähigkeit (*Capability*), einen Problemlöseprozess unter Einsatz von Informations- und Kommunikationstechnologien in komplexen und längerfristigen Situationen zu bewältigen, macht die Aneignung höherer Denkprozesse notwendig (folglich Kapitel 2 „Kompetenzorientierter Unterricht“). Die Fähigkeiten bringen Personen dazu, Informationsmittel zu ihrem Vorteil zu nutzen und unerwartete Probleme zu meistern. In Tab. 3.4 sind Fähigkeiten der Computer-Fluency aufgelistet. Viele der Fähigkeiten gehören zu generellen Problemlöseprozessen und sind auch aus anderen Disziplinen bekannt.

Fähigkeiten der Computer-Fluency	
1.	Sich mit längerfristigen Lösungsansätzen befassen
2.	Umgang mit Komplexität
3.	Testen von Lösungen
4.	Fehler beheben beim Lösen von Problemen
5.	Beschaffen und Organisieren von Information
6.	Kollaborieren
7.	Information weitergeben
8.	Sich auf Unerwartetes einstellen
9.	Sich auf neue Technologien einstellen
10.	Abstrakt über Informationstechnologien denken

Tab. 3.4: Fähigkeiten der Computer-Fluency nach Snyder [für Details siehe National Research Council (U.S.) Committee on Information Technology Literacy 1999].

Aufbau einer Computer-Fluency in 4 Schritten

Fähigkeiten im Umgang mit dem Computer als Computer-Fluency und ICT-Kompetenz werden nur dann langlebig und anwendbar, wenn Konzepte, Fertigkeiten und Fähigkeiten als Einheit und nicht isoliert voneinander betrachtet werden (folglich Kapitel 2 „Kompetenzorientierter Unterricht“). Damit die Konstruktion eines solchen Wissensnetzes in der realen Welt auch für Novizen möglich wird, müssen die einzelnen Wissensdimensionen schrittweise eingeführt und miteinander verknüpft werden. Die beste und effektivste Möglichkeit, unterschiedliche Arten von Wissen miteinander in Beziehung zu bringen, ist deren Aneignung in möglichst authentischen und realistischen Situationen. Hierfür haben wir das Konzept des *Informationsarbeitsplatzes* eingeführt. Der Computer wird dabei als Zentrum der wissenschaftlichen Datenverarbeitung betrachtet und bietet die Möglichkeit zur Integration von Konzepten, Fähigkeiten und Fertigkeiten für das Selbstgesteuerte Lernen (siehe nächster Abschnitt).

Folgender Ablauf hat sich beim Aufbau der Computer-Fluency bewährt und als sehr effektiv herausgestellt (Tab. 3.5) [Faessler 2006, Hinterberger 2007]:

Ausgangslage zur Aneignung der Computer-Fluency ist eine möglichst realistische Problemstellung. Sie stellt den „Raum“ zur Aneignung von Wissen aller Wissensdimensionen zur Verfügung. Konzepte als wichtige Organisations- und Hilfsstruktur bei der Entwicklung einer Kompetenz werden zunächst isoliert betrachtet (Dimension 1). Um aus diesen Konzepten eine Kompetenz aufzubauen, ist die Anwendung der Konzepte zusammen mit Fertigkeiten notwendig (Dimension 2). Durch die zunehmende Erfahrung und Selbständigkeit beim wiederholten Problemlösen können Fähigkeiten angeeignet werden (Dimension 3). Eine Reflexion des gesamten Problemlöseprozesses ermöglicht die Aneignung von metakognitivem Wissen (Dimension 4).

Dimension / Phase		Erklärung
1.	Konzepte isoliert betrachten	Konzepte, welche zur Lösung der Problemstellung angeeignet werden müssen, werden isoliert betrachtet.
2.	Verknüpfung der Konzepte mit Fertigkeiten	Konzepte werden zusammen mit Fertigkeiten eingeübt.
3.	Entwicklung von Fähigkeiten	Durch Wiederholung und zunehmende Selbständigkeit entstehen Fähigkeiten.
4.	Durch Reflexion zur Metakognition	Die Reflexion des ganzen Problemlösungsprozesses ermöglicht die Aneignung von metakognitivem Wissen.

Tab. 3.5: Schrittweise Verknüpfung von Wissensdimensionen zum Aufbau einer Computer-Fluency beim Lösen eines realen Problems. Siehe auch Kapitel 7 „Didaktische Modelle“.

Dieser Ablauf bildet die Grundlage zur Umsetzung des im Rahmen dieser Arbeit entwickelten *4-Schritte-Modells*. Die detaillierte Beschreibung und Umsetzung dieses didaktischen Modells für den Aufbau einer Computer-Fluency ist Thema der Kapitel 7 und 8.

3.3. Herstellen eines realistischen Kontexts zur Aneignung von ICT-Kompetenzen

3.3.1 Der Computer als Informationsarbeitsplatz

Mit der Einführung des Begriffs *Informationsarbeitsplatz* wird der Computer als Zentrum der wissenschaftlichen Datenverarbeitung betrachtet. Er dient als universelles Arbeitsinstrument für die professionelle wissenschaftliche Tätigkeit und ermöglicht eine vielfältige computergestützte Verarbeitung von Daten und Information. Im Informationsarbeitsplatz lassen sich die Konzepte und Methoden der Informatik zusammen mit Fertigkeiten in der Bedienung von Informatikmitteln in einem realen, interdisziplinären Bereich anwenden. Die kompetente Bedienung des Informationsarbeitsplatzes erfordert die Aneignung einer umfassenden ICT-Kompetenz (Abb. 3.4).

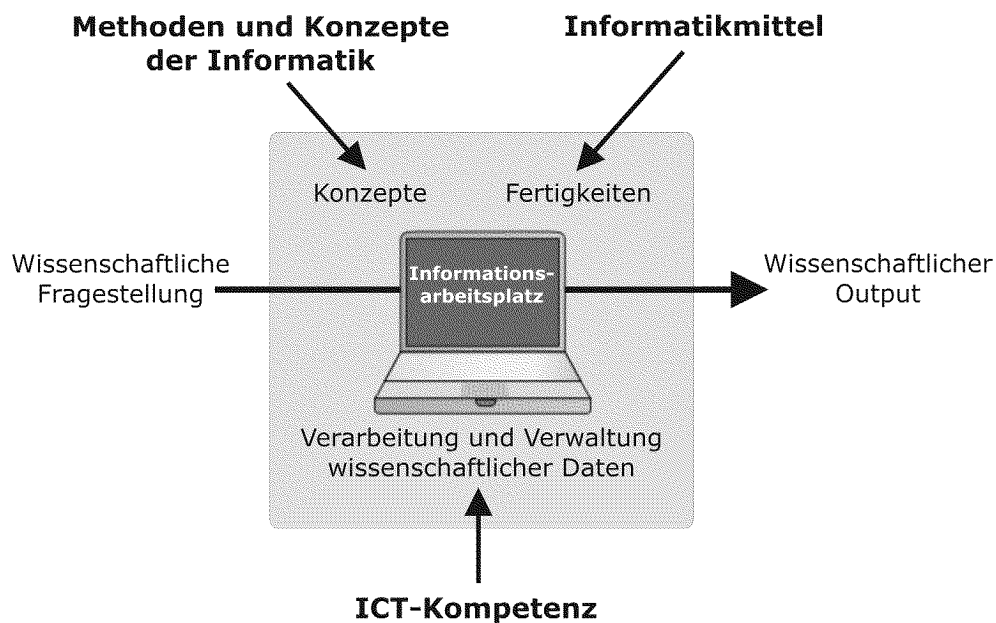


Abb. 3.4: Im Informationsarbeitsplatz lassen sich Konzepte und Methoden der Informatik zusammen mit Informatikmitteln nutzen, um wissenschaftliche Daten in einem interdisziplinären Feld zu verarbeiten und zu verwalten. Hierfür ist eine umfassende ICT-Kompetenz anzueignen.

Die technische und preisliche Entwicklung im Computer-Markt hat es möglich gemacht, dass Informationsarbeitsplätze in Form von leistungsfähigen Geräten weit verbreitet, vernetzt und relativ günstig verfügbar sind (siehe Ressourcenabklärung Abschnitt 3.4). Als Informatikmittel stehen leistungsfähige und umfangreiche Pakete wie Tabellenkalkulationen, Webpublishing-Editoren, Datenbanksysteme und Programmierumgebungen zur Verfügung (siehe Ressourcenabklärung Abschnitt 3.4). Die Verfügbarkeit von Hard- und Software garantiert jedoch nicht, dass der Informationsarbeitsplatz auch zur Lösung von Problemen eingesetzt werden kann. Als dritte Ressource spielt die

Brainware als Grundkompetenz des wissenschaftlichen Arbeitens eine immer wichtigere Rolle (folglich Abb. 3.1). Diese ist vor allem davon abhängig, ob die Anwenderinnen und Anwender eines Informationsarbeitsplatzes in der Lage sind, Konzepte der Informatik zur Anwendung zu bringen und sich Fertigkeiten im Umgang mit Informatikmitteln anzueignen.

Das Konzept des Informationsarbeitsplatzes kann im Unterricht vielfältig genutzt werden. Die beiden wichtigsten Argumente sind sicherlich die Möglichkeiten, den Studierenden von nicht technisch orientierten Studiengängen eine reale Welt zur Aneignung von ICT-Kompetenzen zur Verfügung zu stellen, und Informatik-Konzepte in einem interdisziplinären Umfeld zu unterrichten, in welchem das Fach für die Zielgruppe einen Sinn hat. Der Computer erhält dadurch im Unterricht nicht nur die Aufgabe, Informationen zu verteilen, sondern gleichzeitig ähnlich dem Gang in ein Labor als Arbeitsmittel zur Lösung konkreter Probleme zu dienen. Durch die Selbststeuerung von Handlungen können gleichzeitig auch Lernprozesse individualisiert werden.

3.3.2 Einsatz des Informationsarbeitsplatzes im Unterricht

Für den Einsatz des Informationsarbeitsplatzes zur Vermittlung von Informatik-Kompetenzen sprechen unter anderem folgende drei Argumente:

1. Lernen in einer realen Umgebung

Die Studierenden lernen ICT-Kompetenzen im Informationsarbeitsplatz in einer realen Lernumgebung und nicht in einer abgeschirmten und künstlichen Welt. Sie treffen dabei auf jene Welt, in der ein Computer in Wirklichkeit und in ihrer beruflichen Zukunft auch tatsächlich eingesetzt wird. Dadurch erhalten Konzepte und Kompetenzen einen Realitätsbezug, und es können reale Handlungen erzeugt werden. Da Konzepte und deren Anwendung nicht isoliert voneinander betrachtet werden, bleibt das auf diese Weise angeeignete Wissen weniger träge. Es wird in Zukunft besser transferiert werden können (folglich Kapitel 2 „Kompetenzorientierter Unterricht“). Der Unterschied zum Lernen in einer simulierten artifiziellen Welt besteht darin, dass das Lernen in einer authentischen Situation die Realität nicht bloss abbildet. Dadurch bleiben die Aktionen im Umgang mit der Situation kontextverbunden.

2. Interdisziplinärer Ansatz des Informatikunterrichts

Durch das Lernen im Informationsarbeitsplatz ergibt sich für die Studierenden den Vorteil, dass Kompetenzen direkt an Problemstellungen handlungsorientiert erarbeitet werden können. Werden Informatik-Konzepte in einen interdisziplinären Kontext gestellt, kann nicht nur ein besseres Verständnis erreicht werden, die Informatik-Konzepte erhalten dadurch aus der Sicht der Lernenden von nicht technisch orientierten Studiengängen einen Sinn und eine Berechtigung Teil ihrer Grundausbildung zu sein.

3. Möglichkeit zum selbstgesteuerten, handlungsbasierten Lernen

Der Einsatz des Informationsarbeitsplatzes im Unterricht kann den Studierenden den nötigen Aktivitätsraum liefern, im Lernprozess mehr Selbststeuerung und Verantwortung zu übernehmen, weil sie den Informationsarbeitsplatz selber bedienen müssen. Der Unterricht kann dadurch auch in grossen Klassen individualisiert werden. Damit diese Aufgabe von den Lernenden übernommen werden kann, ist der Informationsarbeitsplatz mit Instruktionen und Problemstellungen zu ergänzen (siehe Abschnitt 3.3.4).

3.3.3 ICT-Kompetenzen für die Arbeit im Informationsarbeitsplatz

Ein Informationsarbeitsplatz stellt verschiedene Werkzeuge zur Verfügung, mit denen unterschiedliche Aufgaben im Umgang mit digitalen Daten und Informationen erledigt werden können (siehe Abb. 3.3). Dazu gehören Aufgaben wie:

- Daten und Informationen verarbeiten, analysieren, verwalten und präsentieren
- Daten und Informationen mit dem Rest der Welt austauschen
- Informatikmittel den eigenen Bedürfnissen anpassen.

Die zur Bewältigung dieser Aufgaben eingesetzten Informatikmittel müssen der Aufgabe entsprechend ausgewählt werden. Für kleinere, einfachere Aufgaben reicht in den meisten Fällen der Einsatz von Standard-Informatikmitteln aus. Bei umfangreicheren Aufgaben kommen diese Informatikmittel häufig an ihre Grenzen, und es müssen spezialisierte Informatikmittel eingesetzt werden.

Gruppen von Kompetenzen im Informationsarbeitsplatz

Folgende Gruppen von Kompetenzen sind für den Einsatz im Informationsarbeitsplatz von Interesse:

A. Rechnernetze und Kommunikation

Dass Computer mit anderen Rechnern über ein Rechnernetz kommunizieren können, ist zu einer Selbstverständlichkeit geworden. Ein unabdingbarer Bestandteil eines Informationsarbeitsplatzes sind Softwarewerkzeuge, um mit anderen Computern zu kommunizieren. Im Informationsarbeitsplatz können mit speziellen Editoren Informationen in Form von Webseiten erstellt und durch das Verschieben auf einen Webserver veröffentlicht werden.

B. Modellierung

Eine wichtige Anwendung für viele Disziplinen ist die Modellbildung und Simulation von Prozessen mit Hilfe numerischer Methoden. Im Informationsarbeitsplatz steht die Tabellenkalkulation als universelles Informatikmittel für eine vielfältige Datenverarbeitung zur Verfügung. Für speziellere Aufgaben der Prozessmodellierung müssen spezialisierte Informatikmittel eingesetzt werden können.

C. Visualisierung

Die Fähigkeit, Daten mit Hilfe von geeigneten Werkzeugen im Informationsarbeitsplatz als aussagekräftige Diagramme darzustellen, dient der Präsentation von Information. Bei grossen, mehrdimensionalen Datenmengen sind die klassischen Darstellungsmethoden überfordert, und es sind spezielle Visualisierungsmethoden einzusetzen.

D. Datenverwaltung

Im Informationsarbeitsplatz müssen grosse Mengen komplexer Daten verwaltet werden können. Computer sind auf mathematischer Logik aufgebaute Geräte, die nur zweckmässig und zuverlässig organisierte Daten verarbeiten können. Die Verwaltung kleiner Mengen von Daten ist mit einer Tabellenkalkulation als Listen und Tabellen möglich. Ab einer bestimmten Grösse einer Datensammlung ist der Einsatz eines Datenbanksystems sinnvoll.

E. Programmierung

Die Fähigkeit, Computerprogramme zu erstellen, kann in allen Bereichen des Informatikarbeitsplatzes eingesetzt werden. Durch eine geeignete Programmierung lässt sich die Funktionalität des Informationsarbeitsplatzes erweitern.

Dazu gehören:

- Makroprogramme für das Automatisieren von Aufgaben innerhalb eines Anwendungsprogramms
- Skript-Programme für Web-Anwendungen
- Anwendungsunabhängige Programme als zusätzliche Informatikmittel.

Im Informationsarbeitsplatz steht das Erweitern und Anpassen von bestehenden Programmen an die individuellen Bedürfnisse im Zentrum.

3.3.4 Entwicklung einer Lernumgebung zum Aufbau von ICT-Kompetenzen im Informationsarbeitsplatz

Der Aufbau von ICT-Kompetenzen im Zusammenhang mit einem Informationsarbeitsplatz wird durch die Entwicklung folgender Elemente unterstützt (Abb. 3.5):

- authentische Situationen mit realen Daten (1)
- Problemstellungen, die das Anwenden von ICT-Konzepten, Fertigkeiten und Fähigkeiten notwendig machen (2)
- individualisierte Instruktionen zur Aneignung von ICT-Konzepten und Fertigkeiten (3)

Daraus ergibt sich eine kompetenzorientierte Lernumgebung zur Aneignung von ICT-Kompetenzen.

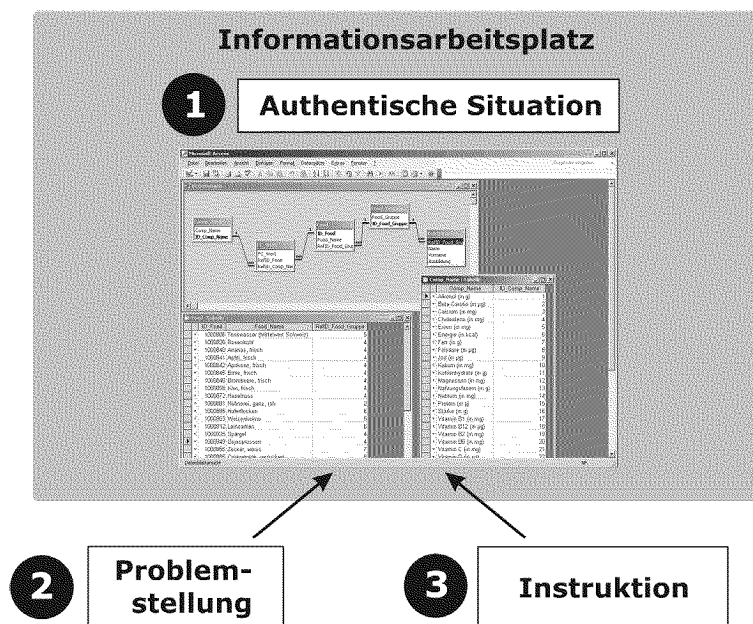


Abb. 3.5: Einsatz des Informationsarbeitsplatzes im Unterricht. Eine authentische Situation mit realen digitalen Daten (z.B. Schweizerische Nährwertdatenbank) (1) muss im Informationsarbeitsplatz von den Lernenden durch eine Problemstellung (2) in eine vorgegebene Richtung verändert werden. Die zur Lösung der Problemstellung nötigen Konzepte und Fertigkeiten (z.B. Anwendung der Konzepte einer relationalen Datenbank und Bedienung einer Datenbank-Applikation) sollen sich die Lernenden mittels Instruktion (3) direkt im Informationsarbeitsplatz aneignen können. (Das Beispiel stammt aus dem ETH-Kurs „Einsatz von Informatikmitteln“ 551-o839-00L).

1. Authentische Situationen mit realen Daten

Authentische, reale Situationen machen aus dem Informationsarbeitsplatz eine interessante Welt und stellen gleichzeitig einen starken Praxisbezug her. Im Idealfall stammen die zu bearbeitenden Daten aus einem Gebiet der Studierenden, z.B. den Na-

turwissenschaften. In Abb. 3.5 ist ein Beispiel einer solchen authentischen Situation dargestellt. Die Studierenden lernen, mit Daten der Schweizerischen Nährwertdatenbank Daten zu verwalten. Es wird darauf verzichtet, lediglich über Daten zu sprechen oder isolierte Software-Fertigkeiten zu üben. Die Lernenden kommen mit den Daten direkt in Kontakt und erhalten die Möglichkeit, den Umgang damit ohne reale Konsequenzen zu üben. Die authentische Situation stellt den Lernenden den Aktivitätsrahmen und den Raum zum Aufbau eines Wissensnetzes zur Verfügung und ermöglicht es ihnen, eigene Erfahrungen beim Bedienen des Informationsarbeitsplatzes zu machen.

2. Problemstellungen, die das Anwenden von ICT-Kompetenzen notwendig machen

Eine Problemstellung soll Aktivität der Studierenden in den Informationsarbeitsplatz bringen, indem die Situation aus Punkt 1 mit einer Zielsetzung in eine bestimmte Richtung bearbeitet und angepasst werden muss. Zur Lösung des Problems im Informationsarbeitsplatz müssen Informatikmittel bedient und Informatik-Konzepte angewendet werden. Durch die Entwicklung solcher Problemstellungen müssen die zu lernenden Informatik-Konzepte angewendet werden, ohne dass sie aus dem Sinnzusammenhang der Situation des Informationsarbeitsplatzes gelöst werden. Die Entwicklung von Problemstellungen ist Thema des Kapitels 5 „Problembasiertes Lernen“.

3. Individualisierte Instruktionen zur Aneignung von ICT-Konzepten und Fertigkeiten

Das Lösen der Problemstellung aus Punkt 2 in einer authentischen Situation aus Punkt 1 erfordert die Mobilisierung von Wissen in einer Handlung. Dieses Wissensnetz aus Konzepten und Fertigkeiten muss zunächst in einer Instruktion angeeignet werden können. Dies geschieht mit Vorteil bereits handlungsbasiert. Die Entwicklung individualisierter und problembasierter Instruktionen ist Thema des Kapitels 6 „Selbstgesteuerte, problembasierte Instruktion“.

3.4. Ressourcenabklärung bei Informatik-Einführungskursen an der ETH Zürich

Die selbstgesteuerte Aneignung von ICT-Kompetenzen im Informationsarbeitsplatz ist von folgenden Ressourcen abhängig:

- In welchem Umfang ist bei den Studierenden eigene *Hardware* (eigene Computer, Laptops) verfügbar? (Abschnitt 3.4.1)
- Welche *ICT-Kompetenzen* können bei den Studierenden vorausgesetzt werden und welche nicht? (Abschnitt 3.4.2)

Um das Vorhandensein dieser Ressourcen zu prüfen, wurden über mehrere Jahre Vorumfragen bei sechs Informatik-Einführungskursen der Natur- und Bauingenieurwissenschaften an der ETH Zürich mit jährlich über 1100 Studierenden durchgeführt (siehe Tab. 1.2).

3.4.1 Verfügbarkeit von Hardware bei den Studierenden

Resultate der Umfrage

In einer über mehrere Jahre durchgeführten Umfrage unter Studierenden in 6 verschiedenen Informatik-Einführungskursen der Natur- und Bauingenieurwissenschaften an der ETH Zürich (Tab. 1.2) gaben durchschnittlich über 99% der Studierenden an, einen eigenen Computer zu haben. Unter ihnen hatten im Jahre 2007 durchschnittlich 70% einen eigenen Laptop zur Verfügung (Abb. 3.6). Dabei gab es Unterschiede, in welchem Semester die Studierenden sind. Während im ersten Semester durchschnittlich 53% einen eigenen Laptop zu besaßen, waren es im zweiten 66% und im vierten Semester bereits 75% der Studierenden. Zwischen den untersuchten Studiengängen konnten keine nennenswerten Unterschiede festgestellt werden. Der Anteil der Studierenden, die bereits zu Beginn des Studiums einen Laptop zur Verfügung hatten, ist in den Jahren 2005 bis 2007 gestiegen, und es ist zu erwarten, dass diese Werte noch weiter steigen werden.

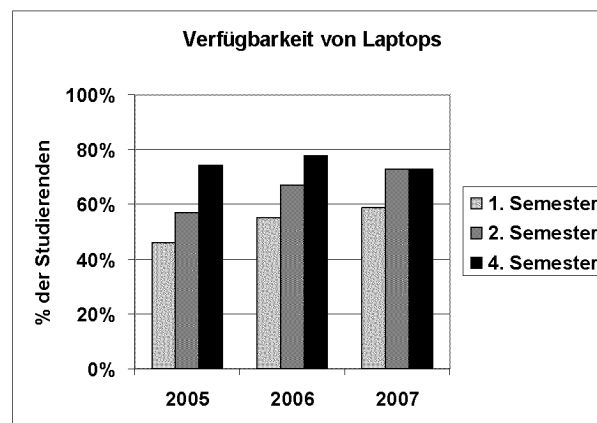


Abb. 3.6: Verfügbarkeit von Laptops gemäss Umfragen unter den Studierenden in den Jahren 2005 bis 2007. Die Studierenden stammen aus den Fachrichtungen Biologie, Bewegungswissenschaften und Sport, Pharmazie (2. Semester), Umweltnaturwissenschaften, Erdwissenschaften, Agrar- und Lebensmittelwissenschaften (1. und 4. Semester), Bauingenieurwissenschaften, Geomatik und Planung, Umweltingenieurwissenschaften (1. und 2. Semester). Die durchschnittliche Anzahl Studierende pro Umfrage beträgt 196.

3.4.2 Verfügbarkeit von ICT-Kompetenzen bei den Studierenden

Studierende bringen aus ihrer Vorbildung einiges an Vorwissen und Erfahrung im Umgang mit einem Informationsarbeitsplatz mit. Andere Bereiche hingegen sind für die Studierenden weitgehend unbekannt. Um abzuklären, inwiefern angehende Studie-

rende der Natur- und Bauingenieurwissenschaften an der ETH Zürich ihre Kenntnisse in den einzelnen Kompetenzbereichen einschätzen, und an welchen Orten sie ihr bisheriges Wissen im Fach Informatik angeeignet haben, wurden über eine Zeitspanne von mehreren Jahren Vorumfragen unter Studierenden im ersten und zweiten Semester durchgeführt.

Resultate der Vorumfragen

Die Resultate der Vorumfragen sind den Abb. 3.7 und Abb. 3.8 zu entnehmen. Abb. 3.7 zeigt das von den Studierenden eingeschätzte Vorwissen in einzelnen Kompetenzbereichen des Informationsarbeitsplatzes. Die in einem elektronischen Fragebogen gestellten Fragen (Tab. 3.6) lehnen sich an die Fertigkeiten der Computer-Fluency von Snyder (Tab. 3.3) an. Abb. 3.8 zeigt die von den Studierenden angegebenen Orte der Wissensaneignung im Fach Informatik.

Wie würden Sie auf einer Skala von 1 bis 5 Ihre Kenntnisse in folgenden Tätigkeitsgebieten einschätzen? (mehrere Antworten möglich) (5=sehr sicher, 4=sicher, 3=einigermassen sicher, 2=unsicher, 1=sehr unsicher)	
1.	Einen neuen Computer oder Laptop in Betrieb nehmen
2.	Einen Computer an ein Netzwerk anschliessen
3.	Mit einer Textverarbeitung (z.B. Word) ein Textdokument erstellen
4.	Ein Illustrations- oder Grafikprogramm einsetzen, um eine Illustration zu erstellen
5.	Mit Hilfe des Internets Informationen beschaffen
6.	Information auf einer Webseite auf dem Internet verfügbar machen
7.	Einen Computer zum Kommunizieren einsetzen
8.	In einer Tabellenkalkulation einfache Prozesse modellieren
9.	Aus einer Reihe von Werten ein aussagekräftiges Diagramm erstellen
10.	Ein Datenbanksystem einsetzen, um Information zugänglich zu machen
11.	Mit Hilfe von Makros Programmabläufe automatisieren
12.	Ein Computer-Programm einer Programmiersprache den eigenen Bedürfnissen anpassen

Tab. 3.6: Unter den Studierenden durchgeführte Vorumfrage während 4 Semestern in den Jahren 2006 und 2007. Die Resultate sind der Abb. 3.7 zu entnehmen.

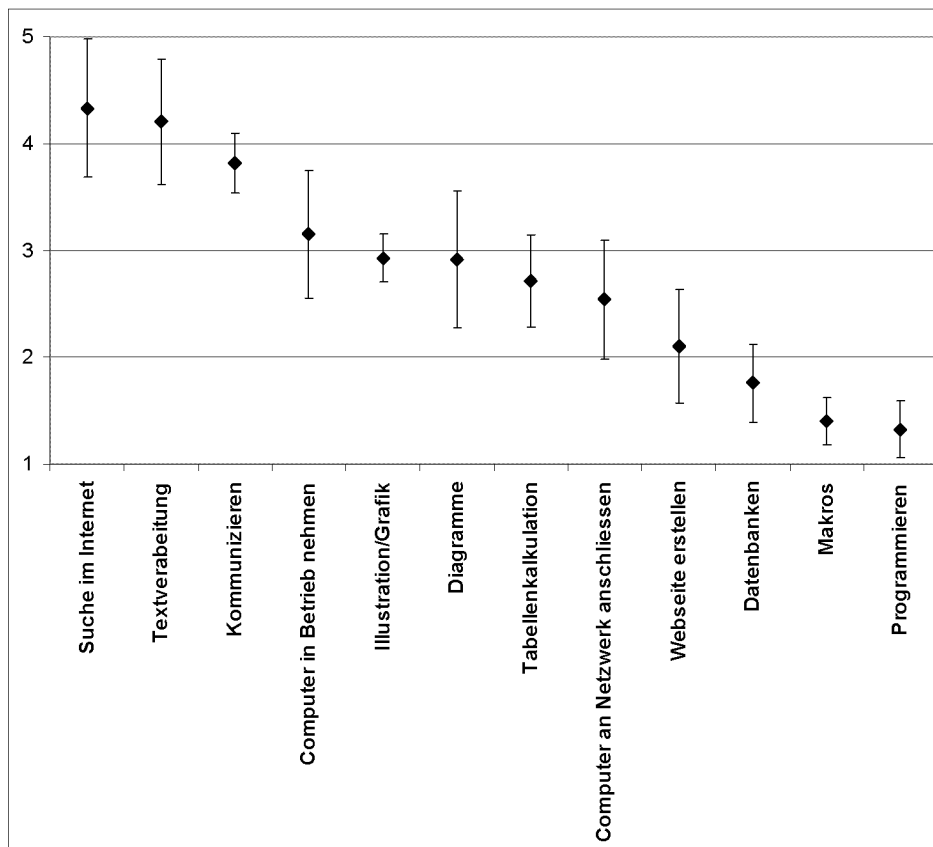


Abb. 3.7: Mittelwert und Standardabweichung der Selbsteinschätzung des Vorwissens der Studierenden in unterschiedlichen Tätigkeitsgebieten des Informationsarbeitsplatzes, sortiert nach dem mittleren eingeschätzten Sicherheitsgrad. Kodierung der Antworten: 5=sehr sicher, 4=sicher, 3=einigermaßen sicher, 2=unsicher, 1=sehr unsicher. N=438 im ersten und zweiten Semester im Jahr 2006/07, durchschnittliche Rücklaufquote: 57%. Die befragten Studierenden stammen aus den Fachrichtungen Biologie, Pharmazie, Bewegungswissenschaften und Sport, Umweltnaturwissenschaften, Erdwissenschaften, Agrar- und Lebensmittelwissenschaften, Bauingenieurwissenschaften, Geomatik und Planung, Umweltingenieurwissenschaften.

Die Resultate der Vorumfrage zeigen, dass es sehr grosse Unterschiede gibt im selbst eingeschätzten Sicherheitsgrad der einzelnen Kompetenzbereiche eines Informationsarbeitsplatzes. Die Mittelwerte variieren zwischen 4.5 (Suche im Internet) und 1.5. (Programmieren). Sicher fühlen sich die Studierenden in den Bereichen „Suche im Internet“, „Textverarbeitung“ und „Kommunizieren“ (Werte > 3.5). Auf der anderen Seite geben die Studierenden in den Bereichen „Webseite erstellen“, „Datenbanksystem einsetzen“, „Makros“ und „Programmieren“ an, unsicher zu sein (Werte < 2.5).

Auch wenn die Resultate nicht die tatsächliche Kompetenz in den einzelnen Gebieten repräsentieren, kann daraus geschlossen werden, dass es einige Gebiete gibt, bei denen die Studierenden der Natur- und Bauingenieurwissenschaften über wenig bis gar keine Grundlagen verfügen. Typischerweise betrifft dies alle Kompetenzbereiche, die eine Anwendung von Informatik-Konzepten erfordern: „Webseite erstellen“, „Daten verwalten“ und „Programmieren“. Die Kompetenz-Bereiche „Suche im Internet“, „Textverarbeitung“ und „Kommunikation“ setzen die Studierenden bereits mehrheitlich in ihrem

Alltag ein. Diese Bereiche kommen heute grösstenteils ohne Anwendung von Informatik-Konzepten aus.

Über 84% der Studierenden geben an, sich ihr Informatikwissen selber durch Ausprobieren von Applikationen angeeignet zu haben (Abb. 3.8). An zweiter Stelle folgt der Unterricht in einer öffentlichen Schule (72%), gefolgt von Wissen von Freunden und Bekannten (54%). Nur wenige der Studierenden haben ihr Wissen in Software-Schulungen oder an einem Arbeitsplatz erworben.

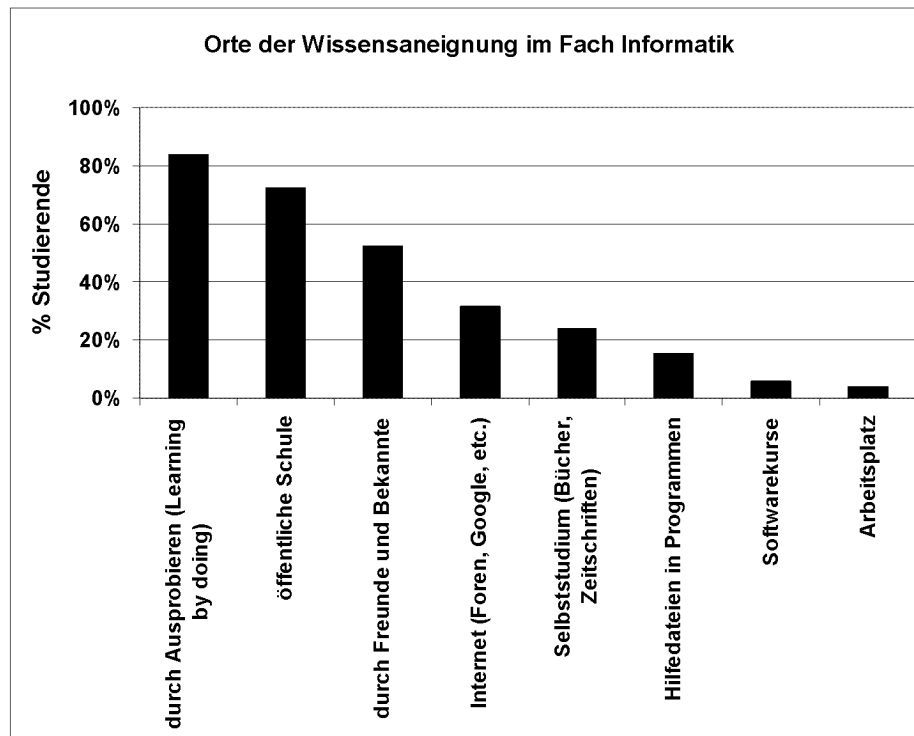


Abb. 3.8: Vorumfrage über die Orte der Wissensaneignung im Fach Informatik. Die Frage lautete im Fragebogen: *Ich habe mein Wissen in Informatik bisher an folgenden Orten angeeignet (mehrere Antworten möglich)*. N=657 im ersten und zweiten Semester in den Jahren 2005 bis 2007. Die befragten Studierenden stammen aus den Fachrichtungen Biologie, Pharmazie, Bewegungswissenschaften und Sport, Umweltnaturwissenschaften, Erdwissenschaften, Agrar- und Lebensmittelwissenschaften.

Folgerungen aus den Vorumfragen

Die Resultate dieser Vorumfragen zeigen, dass im Grundstudium der Natur- und Bauingenieurwissenschaften im universitären Bereich Bedarf besteht, Kompetenzen im Informationsarbeitsplatz, die den Umgang mit digitalen Daten betreffen, zu unterrichten. Die Verfügbarkeit von Hardware kann heute als ausreichend bezeichnet werden, die Studierenden müssen aber lernen, die Geräte für die wissenschaftliche Arbeit einzusetzen. Themen wie Datenbanken, Programmieren und Publizieren über Internet erfordern die Aneignung von Informatik-Konzepten und können nicht auf die selbe Art und Weise gelernt werden, wie das im Umgang mit E-Mail, Textverarbeitung und der Internetsuche geschieht. Lernen durch Ausprobieren von Computer-Applikationen und

durch Tipps von Freunden und Bekannten reicht hierfür nicht aus. Es muss ein Grundlagenunterricht geschaffen werden, der die Aneignung und Anwendung von Konzepten zum Ziel hat.

3.4.3 Kursprogramm zum Aufbau von ICT-Kompetenzen im Informationsarbeitsplatz

Abb. 3.9 zeigt den Aufbau des Informatikunterrichts im Grundstudium einiger Studiengänge der ETH Zürich. Ein ICT-Kurs behandelt die Kompetenz-Gruppen A bis E (siehe Abschnitt 3.3.3), mit dem Ziel, den Informationsarbeitsplatz bedienen zu können. Ein Programmierkurs vertieft die Kompetenzgruppe E, mit dem Ziel, den Funktionsumfang des Informationsarbeitsplatzes den eigenen Bedürfnissen entsprechend zu erweitern. Details zum Kursprogramm mit Angaben zu Aufgaben, Konzepten und Fertigkeiten sowie den eingesetzten Werkzeugen und der realen Situation im Informationsarbeitsplatz sind dem Anhang A zu entnehmen.

Beim Zusammenstellen des Kursprogramms wurden die Schwerpunkte bewusst bei jenen Kompetenzen gesetzt, bei denen ein Grossteil der Studierenden noch wenig Vorwissen hat (Abb. 3.7) und zu deren Aneignung Informatik-Konzepte angewendet werden müssen (Tab. 3.2). Diese Kompetenzen sind ausserhalb der Schule nur schwierig anzueignen, weil sie ein Monitoring voraussetzen und ein hohes Zeitvolumen in Anspruch nehmen.

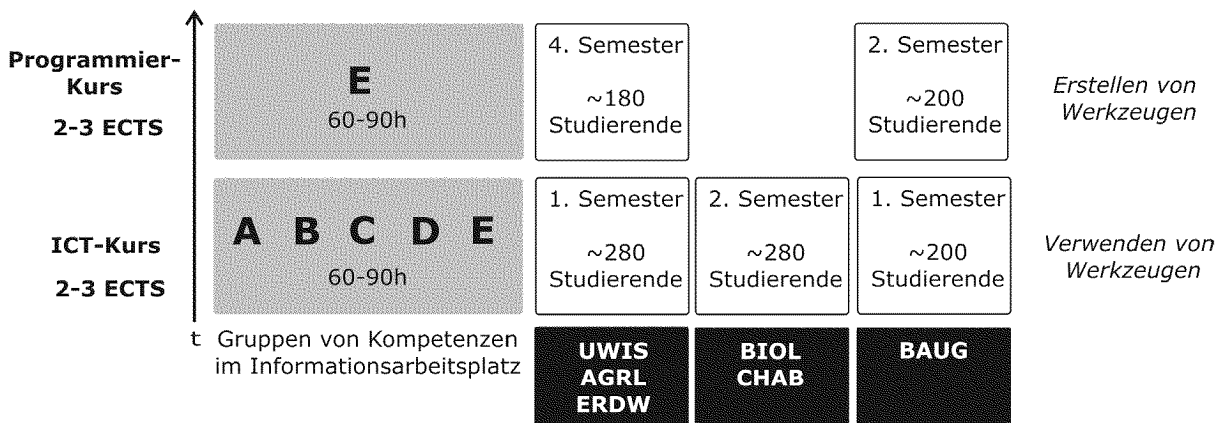


Abb. 3.9: Aufbau des Informatikunterrichts im Grundstudium einiger Studiengänge der ETH Zürich mit zwei Kursen von je einem Semester Dauer (14 Wochen, 2 bis 3 ECTS-Punkte). In einem ICT-Kurs werden ICT-Kompetenzen der Gruppen A bis E behandelt. Ein Programmierkurs hat die Kompetenzgruppe E zum Thema. Abkürzungen: *Gruppen von Kompetenzen*: A: Rechnernetze und Kommunikation, B: Datenmodellierung, C: Datenvisualisierung, D: Datenverwaltung, E: Programmierung. *Zielgruppen*: AGRL: Agrar- und Lebensmittelwissenschaften, BAUG: Bau, Umwelt und Geomatik, BIOL: Biologie, Bewegungswissenschaften und Sport, CHAB: Chemie und angewandte Biowissenschaften, ERDW: Erdwissenschaften, UWIS: Umweltnaturwissenschaften.

Kapitel 4

Kompetenzorientierte Lernumgebungen

4.1. Einleitung

In diesem Kapitel wenden wir uns den Eigenschaften von kompetenzorientierten Lernumgebungen und den Anforderungen an deren Entwicklung zu. Unter dem Begriff *Lernumgebung* werden Systeme verstanden, in denen Lehr- und Lernprozesse stattfinden. Zu einer Lernumgebung gehören die Lernenden, die Lehrenden und der Unterrichtsinhalt sowie ein Ort und eine Zeit, wo das Lernen stattfindet. Um den Lernprozess zu individualisieren und den Aufbau von Kompetenzen zu ermöglichen, sollten Lernumgebungen unabhängig von Ort, Zeit und Person funktionieren können (folglich Kapitel 2 „Kompetenzorientierter Unterricht“).

Das Didaktische Dreieck

Die Lehrenden, Lernenden und der Lerninhalt stehen im Unterricht miteinander in einer charakteristischen Beziehung, dem *Didaktischen Dreieck* (Abb. 4.1).

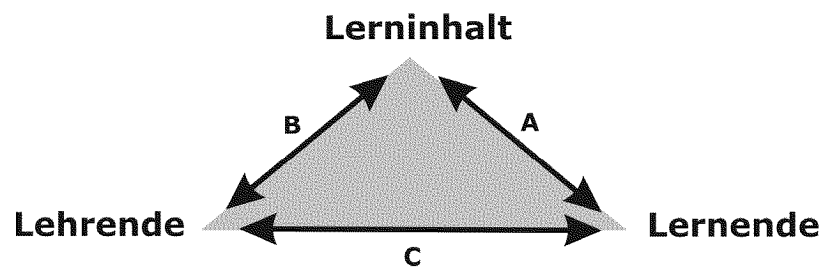


Abb. 4.1: Das Didaktische Dreieck: Unterricht als Interaktionen zwischen Lehrenden, Lernenden und dem Lerninhalt. Interaktionen A, B und C siehe Text.

Im Didaktischen Dreieck einer Lernumgebung finden folgende Interaktionen statt:

- Der Lerninhalt wird von den Lernenden gelernt (Interaktion A) und von den Lehrenden gelehrt (Interaktion B).
- Die Lernenden lernen den Lerninhalt (Interaktion A) durch Instruktionen der Lehrenden (Interaktion C).
- Die Lehrenden vermitteln Lerninhalt (Interaktion B) durch Instruktionen und überprüfen das Verständnis der Lernenden (Interaktion C).

Je nach Bedingungen (Zielgruppe, Grösse der Klassen etc.) und Ziel des Unterrichts ergeben sich im didaktischen Dreieck innerhalb einer Lernumgebung unterschiedliche Interaktionsmuster zwischen Lehrenden, Lernenden und dem Lerninhalt, welche die Lehr- und Lernprozesse beeinflussen.

Wir interessieren uns im Folgenden für Lernumgebungen mit speziellen Bedingungen, welche die Interaktionen innerhalb des didaktischen Dreiecks beeinflussen:

- Lernumgebungen mit *indirekter Instruktion* (Abschnitt 4.2.1)
- *Kompetenzorientierung* mit indirekter Instruktion (Abschnitt 4.2.2)

In diesem Kapitel soll folgenden Fragen nachgegangen werden:

- Welches sind die Eigenschaften von kompetenzorientierten Lernumgebungen mit indirekter Instruktion? (Abschnitt 4.2)
- Mit welchen Methoden können funktionierende Lernumgebungen entwickelt werden? (Abschnitt 4.3)
- Durch welche Elemente lässt sich eine kompetenzorientierte Lernumgebung als Modell beschreiben? (Abschnitt 4.4)
- Welches sind die Besonderheiten der Entwicklung von lernprozessunterstützenden Strukturen einer kompetenzorientierten Lernumgebung? (Abschnitt 4.5)

4.2. Eigenschaften von kompetenzorientierten Lernumgebungen mit indirekter Instruktion

4.2.1 Lernumgebungen mit indirekter Instruktion

Um Lernumgebungen zu individualisieren, sollen *Lernmaterialien* und *Medien* entwickelt und eingesetzt werden, mit deren Hilfe die Lernenden weitgehend unabhängig von Ort, Zeit und Lehrpersonen lernen können. Die Lernprozesse können dadurch vollständig individualisiert und die Interaktion A in Abb. 4.1 zwischen den Lernenden und dem Lerninhalt intensiviert werden. Es entsteht eine Lernumgebung mit *indirekter Instruktion*, was die Interaktionen im Didaktischen Dreieck wie folgt verändert:

1. Veränderung für die Interaktion A im Didaktischen Dreieck

Um Lernprozesse zu individualisieren und gleichzeitig die Interaktion zwischen den Lernenden und dem Inhalt zu intensivieren, wird die zu lernende Information vermehrt indirekt über Lernmaterialien und Medien statt direkt durch die Lehrperson vermittelt (Abb. 4.1, Interaktion A). Es kommen Lernmethoden und Werkzeuge zum Einsatz, welche die Eigenaktivität der Lernenden erhöhen. Die Lernenden sind dadurch gezwungen,

Handlungen auszuführen, Lernprozesse zunehmend selber zu steuern und zu überprüfen. Dadurch rückt die Entwicklung von lernendenzentrierten Lernmaterialien (Materialien für das Selbststudium) ins Zentrum des Interesses, mit deren Hilfe sich die Lernenden selber Kompetenzen aneignen können. Dank der zunehmenden Selbststeuerung können auch grosse Klassen auf qualitativ hohem Niveau individuell unterrichtet werden.

2. Veränderungen für die Interaktionen B und C im Didaktischen Dreieck

Der Einsatz von Lernmaterialien für ein individuelles Lernen verändert die Beziehung zwischen den Lehrenden und den Lernenden (Abb. 4.1, Interaktion C) grundlegend. Die Lehrenden nehmen die Rolle als Entwicklerinnen und Entwickler von Lernumgebungen und Lernmaterialien ein (Abb. 4.1, Interaktion B), und es stellen sich mit indirekter Instruktion neue Herausforderungen bei der Leistungskontrolle (Abb. 4.1, Interaktion C). Die Aufgaben der direkten Instruktion und Überprüfung rücken in den Hintergrund. Das Erstellen von Unterrichtsmaterialien und deren Integration in eine Lernumgebung wird professionalisiert und erfordert die Aneignung neuer Kompetenzen durch die Lehrpersonen (siehe nächster Abschnitt).

4.2.2 Kompetenzorientierung mit indirekter Instruktion

Die Wissensvermittlung mit indirekter Instruktion kennt man seit der Erfindung der Schrift. Wie verhält es sich aber mit der Vermittlung von *Kompetenzen* durch indirekte Instruktion? Wie im Kapitel 2 „Kompetenzorientierter Unterricht“ erläutert, ist die Aneignung von Kompetenzen von einem aktiven individuellen Konstruktionsprozess der Lernenden zusammen mit Strukturen zur Organisation von kognitiven Prozessen, Handlungen und Reflexionen abhängig. Würden in einer Lernumgebung mit indirekter Instruktion bloss isolierte Informationen und Lerninhalte bereitgestellt oder präsentiert werden, wäre das einfach ein Informationssystem. Dies reicht zum Aufbau von Kompetenzen nicht aus. Lerninhalte müssen zu *Learning Management Systemen* aufbereitet werden (Abb. 4.2). Diese enthalten neben den Lerninhalten unter anderem zusätzlich:

- Werkzeuge
- ein Anwendungskontext
- handlungssteuernde Strukturen
- Möglichkeiten zur (Selbst-) Überprüfung
- ein Monitoring der Studierenden und Feedback



Abb. 4.2: Indirekte Instruktion in einer kompetenzorientierten Lernumgebung: Lerninhalte werden durch die Lehrenden so zu einem Learning Management System aufbereitet, dass Lernende zunehmend selbstgesteuert definierte Kompetenzen aufbauen können. Learning Management Systeme werden zum zentralen Element zwischen Lehrenden und Lernenden. Die Aufbereitung von Lerninhalten entspricht in Abb. 4.1 der Interaktion B, der Kompetenzaufbau durch die bereitgestellten Elemente des Learning Management Systems der Interaktion A.

Unterschiede zwischen Informations- und Lernsystemen

Aus den unterschiedlichen Zielsetzungen von Informations- und Lernsystemen ergeben sich unterschiedliche Anforderungen an deren Entwicklung. Bei einem Informationssystem steht die Zugänglichkeit von Information im Vordergrund. Bei der Entwicklung eines Lernsystems soll der gezielte Aufbau von Kompetenzen mit Hilfe von Informationen unterstützt werden. Bei beiden Systemen werden die gleichen Medien eingesetzt, um Informationen zu verbreiten und darzustellen. Sie unterscheiden sich jedoch darin, dass ein Lernsystem neben Information zusätzlich handlungssteuernde Strukturen besitzt, mit deren Hilfe die Lernenden ein definiertes Kompetenzniveau (folglich Kapitel 2.5) erreichen können (Abb. 4.3). Bei einem Lernsystem steht somit die Interaktion zwischen den Lernenden und dem Lerninhalt im Zentrum. Die didaktische Aufbereitung von Information in einem Medium zu einem kompetenzorientierten Lernsystem erfordert deshalb eine ergonomisch orientierte Entwicklung von Lernmaterialien, mit dem Ziel, dass Information von Lernenden auf effiziente Art und Weise gelernt werden kann (siehe Abschnitt 4.5 „Entwicklung Lernprozess-unterstützender Strukturen“).

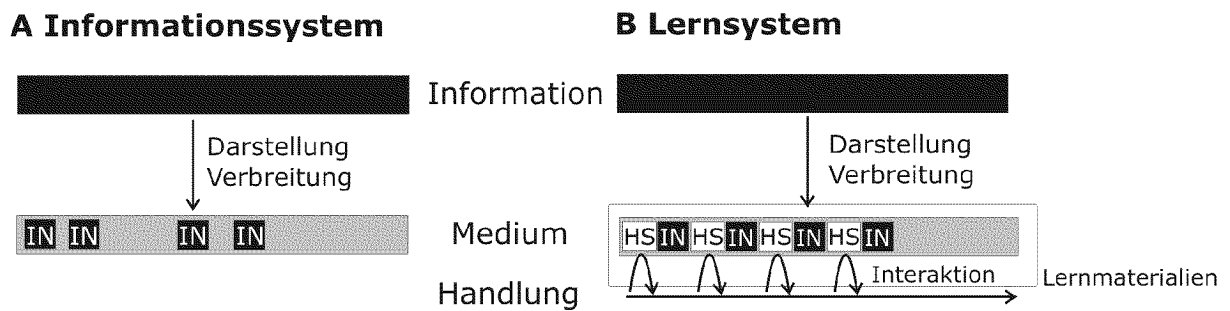


Abb. 4.3: Unterschiede zwischen einem Informationssystem (A) und einem kompetenzorientierten Lernsystem (B). In beiden Systemen wird mit Hilfe eines Mediums Information verbreitet und dargestellt. Bei einem Lernsystem dienen darüber hinaus zusätzliche Strukturen der Steuerung von Handlungen der Lernenden mit dem Ziel einer Kompetenzaneignung. Die Information wird durch die Arbeit mit den Lernmaterialien direkt mit Handlungen der Lernenden verknüpft. Abkürzungen: IN: Information, HS: Handlungssteuernde Strukturen.

Die Rolle der Medien in einer kompetenzorientierten Lernumgebung

Seit der Digitalisierung haben sich die Möglichkeiten der Gestaltung und Verbreitung von Medien stark verändert. Neue Formate ermöglichen vielfältige Multimedia-Anwendungen, und das Internet steht als effizientes Verteilsystem von Information in unterschiedlichen Medienformaten zur breiten Verfügung. Es gibt viele Versuche, auf der Stufe der Medien, beispielsweise durch Animationen und Simulationen, Aktivität in die zu vermittelnde Information zu bringen. Für den Aufbau von Kompetenzen sind aber vor allem Aktivitäten der Lernenden in der Auseinandersetzung mit Lerninhalten notwendig (Abb. 4.1, Beziehung A und Tab. 1.1, Stufe 5). Damit Lernmedien den Lernprozess sinnvoll unterstützen, ist deshalb nicht deren Eigenaktivität entscheidend, sondern ihre Eigenschaft, Aktivitäten der Lernenden im Lernprozess zu fördern und den Lernprozess in Gang zu bringen. Das bedeutet, dass die Medien mit geeigneten *Lernmethoden* in Verbindung gebracht und zu Lernmaterialien ausgebaut werden müssen, die eine starke Interaktion zwischen den Lernenden und dem Inhalt ermöglichen.

Beim *Blended Learning* steht der Einsatz einer geeigneten Kombination von Medien zur Umsetzung von Lernmethoden im Vordergrund, mit dem Ziel, Aktivitäten der Lernenden zu erzeugen, die ein optimales Lernen ermöglichen. Das daraus resultierende didaktische Modell beschreibt den zeitlichen Ablauf sowie den Einsatz der Medien und Lernmethoden zum Aufbau von Kompetenzen (siehe nächster Abschnitt 4.2.3 sowie Kapitel 7).

Handlungsorientierter Ansatz

Durch einen handlungsorientierten Ansatz unter Einbezug von Werkzeugen wird nicht nur Lerninhalt sequenziert. Die Lernumgebung wird auch Lernenden-gesteuert und -zentriert, indem diese ihre eigene Handlung von A nach B bringen müssen. Die dafür nötige Unterstützung sollen die Lernenden ebenfalls durch das Medium erhalten (folglich Abb. 4.3). Dieses *learning by doing* hat den Vorteil, dass Lerninhalte dann beigezo-

gen werden können, wenn es der Stand der Handlung bei den Lernenden verlangt. Gleichzeitig werden die Lerninhalte laufend mit einer Handlung verknüpft, was ein aktives Lernen garantiert. Aus der Sicht der Lernenden macht dadurch sowohl die Aneignung des Lerninhalts als auch die für das Lernen notwendige Aktivität Sinn.

Die handlungssteuernden Strukturen des Mediums sollen sich am individuellen Lernprozess der Lernenden orientieren und soweit variiert werden können, dass die Interaktion zunächst intensiviert und dann schrittweise zurückgenommen werden kann, damit Selbständigkeit entstehen kann (siehe Abschnitt 4.5 „Entwicklung Lernprozess-unterstützender Strukturen“ und „Cognitive Apprenticeship-Modell“). Das Resultat ist eine Intensivierung der Interaktion zwischen den Lernenden und dem Lerninhalt (Abb. 4.1, Interaktion B). Der durch eigenes Handeln eingeübte Inhalt bleibt in einer späteren Anwendung weniger träge und ist deshalb nachhaltiger (siehe Kapitel 2 „Kompetenzorientierter Unterricht“).

4.2.3 Lernmethoden und didaktische Modelle

1. Implementierung von Lernmethoden in eine Lernumgebung

Um in einer Lernumgebung Aktivitäten und Handlungen von Lernenden zu erzeugen, ist die Implementierung von *Lernmethoden* notwendig. Aktivitäten von Lernenden alleine garantieren jedoch noch keine Aneignung von Kompetenzen. Häufig gibt man sich bei der Implementierung von Lernmethoden in eine Lernumgebung bereits damit zufrieden, wenn irgendwelche Studierenden-Aktivitäten erreicht werden. Die erzeugten Aktivitäten erzielen aber erst dann ihren Zweck, wenn sie in direktem Zusammenhang mit der Kompetenzaneignung stehen und die Lernenden den Nutzen ihrer Aktivitäten als sinnvoll, effektiv und als Teil ihrer Kompetenzaneignung betrachten. Damit Lernende im Lernprozess Selbständigkeit entwickeln und die Steuerung ihres Lernprozesses übernehmen können, müssen sie mit einer Lernmethode vertraut sein und deren Wirkung als effizient und effektiv einschätzen. Dieses Vertrauen in eine Methode muss zuerst aufgebaut werden. Erst wenn eine Lernmethode in einer Lernumgebung wiederholt eingesetzt wird und Lernende gute Erfahrungen damit machen, werden sie Sicherheit und Selbständigkeit im Umgang mit einem Inhalt entwickeln. Das heisst, dass eine Lernmethode wiederholt eingesetzt werden muss, damit ihre Effektivität und Wirksamkeit zum Tragen kommt. Treffen die Lernenden in einer Lernumgebung auf zu viele verschiedene und/oder schlecht implementierte Lernmethoden, so führt dies oft zu Unsicherheiten, die der Entstehung von Selbständigkeit entgegenwirken. Dies steht im Kontrast zum Ansatz, in einer Lernumgebung eine möglichst grosse Methodenvielfalt zu gewährleisten.

In Kapitel 5 wird die detaillierte Implementierung der Lernmethode des problembasierenden Lernens beschrieben.

2. Didaktische Modelle

Wie in den vorangehenden Abschnitten erläutert, garantiert der isolierte Einsatz von Lernmethoden und Medien alleine noch keine Aneignung von Kompetenzen. Hierfür ist die Umsetzung eines didaktischen Konzepts in einem Unterrichtsgefäß notwendig. Lernmaterialien müssen mit der Entwicklung und dem Einsatz von Medien sowie mit Aktivitäten einer Lernmethode in einer Lernumgebung kombiniert werden (folglich Abb. 4.3). Dabei spielt neben der Kombination von Medien und Methoden auch der zeitliche Ablauf eine wesentliche Rolle, mit der diese im Unterricht eingesetzt werden.

Das *didaktische Modell* beschreibt die Organisationsstruktur und die zeitlichen Abläufe des Medieneinsatzes in einer Lernumgebung. Es stellt sich dabei die Frage, wie Medien und Lernmethoden in einer Lernumgebung ausgewählt und sequenziert werden müssen, damit Lernende effektiv lernen können, um das Ziel der Aneignung einer Kompetenz zu erreichen.

Im englischsprachigen Raum hat sich hierfür der Begriff des Instruktions-Designs (*Instructional Design*) etabliert [Gagné 1974]. Im deutschsprachigen Raum hat sich diese Bezeichnung nie richtig durchgesetzt. Die Unterschiede bei der Verwendung der beiden Begriffe haben wahrscheinlich mit der geisteswissenschaftlichen Tradition der Didaktik im deutschsprachigen Raum zu tun und mit dem negativen Image des Begriffs „Instruktion“ im Rahmen der Diskussion um den Konstruktivismus. In dieser Arbeit soll der Begriff des „didaktischen Modells“ verwendet werden.

Die Wahl verschiedener Medien zur Umsetzung zweier im Rahmen dieser Arbeit entwickelter didaktischer Modelle ist Thema des Kapitels 7 „Didaktische Modelle“.

4.3. Die Entwicklung von Lernumgebungen als System-Engineering

Eine Lernumgebung kann als System betrachtet werden, das die Lernenden, die Lehrenden und die zu vermittelnde Information umfasst (siehe Abb. 4.1) und Strukturen zur Verfügung stellt, mit dem Ziel, dass sich Lernende aus Information individuell Kompetenzen aneignen können (siehe Abb. 4.2). Die neue Aufgabe der Lehrenden, solche Systeme zu entwickeln, erfordert eine Erweiterung und Professionalisierung einer ganzen Reihe von Autoren-Kompetenzen und eröffnet andererseits Möglichkeiten, die entstehenden Systeme wissenschaftlich zu betrachten und zu diskutieren.

Bei der Entwicklung unserer Lernumgebung sind wir nicht bewusst einem existierenden didaktischen Modell gefolgt. Wir haben vielmehr ein existierendes System immer wieder eingesetzt und optimiert. Im Verlaufe der Zeit ist daraus ein abstrahiertes didaktisches Modell entstanden, welches Grundlage einerseits für die Weiterentwicklung der Lernumgebung und andererseits für eine wissenschaftliche Betrachtungsweise lieferte. Der Ansatz des *System-Engineerings*, der Entwicklung und der stetigen Verbesserung von Systemen, deckt sich mit der kybernetischen Sicht in der

Didaktik, die zum Ziel hat, durch Optimierungsstrategien Lernprozesse in Lernumgebungen so effizient wie möglich zu gestalten [Berner 1999].

4.3.1 System-Engineering von Lernumgebungen

Die Entwicklung ganzer Lernumgebungen ist mit einem umfangreichen Entwicklungsprozess verbunden. Es ist unmöglich, ein derart komplexes System mit allen Strukturen und Prozessen gleich in einer ersten und endgültigen Version korrekt umzusetzen. Es muss vielmehr ein Entwicklungsprozess gefunden werden, der die stetige Verbesserung der Lernumgebung unter Berücksichtigung vieler Aspekte aus vielen verschiedenen Disziplinen ermöglicht. Dies setzt voraus, dass die Lernumgebung immer wieder operativ eingesetzt, evaluiert und verbessert wird.

Wilson und Davis haben in ihrem Buch „Redesigning education“ vorgeschlagen, die Qualität der Lehre dadurch zu verbessern, dass Lernumgebungen entwickelt und eingesetzt werden. Durch die stetige Verbesserung sollen immer wieder aktualisierte Versionen hervorgebracht werden [Wilson & Davis 1994].

Entwicklung von Systemen mit komplexen Anforderungen

Die Entwicklung komplexer Systeme, wie beispielsweise eines modernen Verkehrsflugzeugs, konnte auch nicht von einem Tag auf den anderen geplant und umgesetzt werden. Neue Typen wurden aus bereits existierenden Typen entwickelt. Die Systemanforderung, flugfähig zu sein, war bereits in den ersten Versionen umgesetzt, in der Entwicklung von einem Typ zum nächsten mussten aber zusätzliche Aspekte aus verschiedenen Disziplinen (z.B. Ökonomie, Materialtechnik, Sicherheit, Ökologie etc.) berücksichtigt und integriert werden. Durch das stetige Anpassen liess sich das System laufend weiterentwickeln. Zudem entstanden Typen, die nur für ganz spezifische Aufgaben gebaut wurden. Viele allgemeine Strukturen eines Systems lassen sich deshalb nur im Nachhinein durch seine Entstehungsgeschichte verstehen, erklären und beschreiben.

Das Vorgehen des System-Engineerings bei der Entwicklung von Lernumgebungen stellt sicher, dass neben dem Fachinhalt auch psychologische, pädagogische und soziale Aspekte berücksichtigt werden. Diese interdisziplinären Aspekte brauchen in der Lernumgebung keine sichtbaren Strukturen zu sein, sie sollen vielmehr im Lernprozess ihre Funktion ausüben. Im daraus abstrahierten didaktischen Modell hingegen sollen diese Aspekte dokumentiert, diskutiert und reflektiert werden können. Es dient als Ausgangslage für eine allfällige Optimierung des ganzen Lernsystems.

Abb. 4.4 stellt ein solches Entwicklungsszenario einer Lernumgebung durch den Ansatz des System-Engineerings dar. Dabei entstehen zwei Outputs:

- die im Unterricht einsetzbare *Lernumgebung*

- ein *didaktisches Modell*, dessen Reflexion und Diskussion als wissenschaftlicher Output des Entwicklungsprozesses verwertet werden kann.

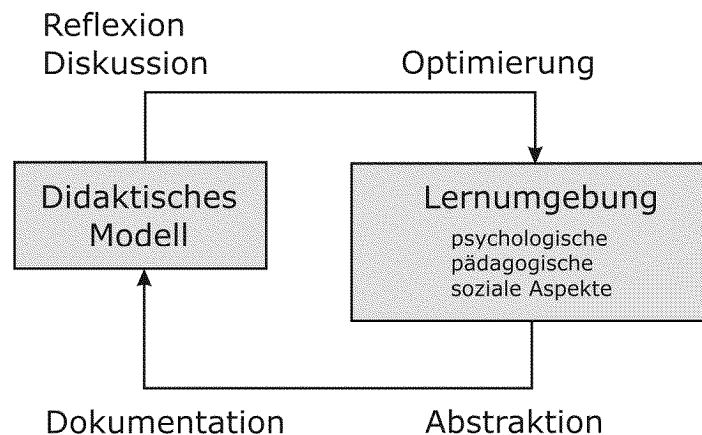


Abb. 4.4: Methoden des System-Engineerings für die Entwicklung einer in der Praxis einsetzbaren Lernumgebung. Das didaktische Modell stellt eine Abstraktion der gesamten Lernumgebung dar, die auch psychologische, pädagogische und soziale Aspekte enthält. Es dient als Dokumentation und Diskussionsgrundlage zur Optimierung der gesamten Lernumgebung.

Realistische statt idealisierte Umsetzung didaktischer Modelle

Durch den Ansatz des System-Engineerings bei der Entwicklung von Lernumgebungen soll ein realistischer und sinnvoller Regelkreis zwischen einem didaktischen Modell und der Umsetzung in der Lernumgebung entstehen. Der Abgleich zwischen dem didaktischen Modell und seiner Umsetzung in der Lernumgebung geschieht in kleinen Schritten. Das didaktische Modell enthält Idealvorstellungen, in der Lernumgebung müssen diese auf bestimmte Bedingungen (Zielgruppe, Zeit usw.) angepasst und umgesetzt werden. Auf diese Weise entstehen mit der Zeit realistische Modelle statt unumsetzbare Ideale.

Ein Merkmal von didaktischen Modellen, welche auf diese Weise entstehen, ist, dass deren Übertragbarkeit auf andere Gebiete und Bedingungen beschränkt bleibt, weil sie durch den direkten Praxisbezug an Kontexte und Zielgruppen gebunden sind. Für die Entwicklung neuer Lernumgebungen sind aber letztlich funktionierende Lernumgebungen von grösserem Interesse als solche, die zwar ein Ideal repräsentieren, aber nicht umsetzbar sind oder nicht funktionieren.

Die Beschreibung und Umsetzung zweier didaktischer Modelle (*4-Schritte-Modell* und *Anwendungsebenen-Modell*), die im Zusammenhang mit dieser Arbeit entstanden sind, ist Inhalt des Kapitels 7 „Didaktische Modelle“.

4.4. Modell eines kompetenzorientierten Lernsystems

Ziel einer kompetenzorientierten Lernumgebung ist der individuelle Aufbau von Kompetenzen. Damit ein Lernprozess in Gang kommen kann und sich Lernende mit Hilfe der vom Lernsystem bereitgestellten Strukturen Kompetenzen aneignen können, müssen Fragen der Aktivität, Verantwortung, Kontrolle, Motivation und Interaktion innerhalb des Systems geklärt werden.

In diesem Abschnitt soll der Frage nachgegangen werden, welche Elemente ein kompetenzorientiertes Lernsystem charakterisieren und die individuellen Lernprozesse beeinflussen. Sie sind Teil eines Modells, das als Grundlage für die Entwicklung Lernprozess unterstützender Systemstrukturen dienen soll, die im nächsten Abschnitt diskutiert werden.

4.4.1 Elemente eines kompetenzorientierten Lernsystems

In einem kompetenzorientierten Lernsystem können zwei Gruppen von Elementen unterschieden werden:

1. Lerneinheit (Abschnitt 4.4.2)

Mit einer ersten Gruppe von Elementen wird eine *Lerneinheit* eines kompetenzorientierten Lernsystems unabhängig von beteiligten Personen charakterisiert. Diese Größen werden meistens dann genannt, wenn eine Lerneinheit oder ein Kurs beschrieben werden muss (z.B. in einem Vorlesungsverzeichnis). Dazu gehören folgende Größen:

- Zeit
- Kompetenzniveau
- Anzahl und Struktur verarbeiteter Konzepte
- Leistungskontrolle

2. Personen (Abschnitt 4.4.3)

Mit einer zweiten Gruppe von Elementen werden *Eigenschaften beteiligter Personen* (Lehrende und Lernende) und deren Interaktion in einem kompetenzorientierten Lernsystem beschreiben. Dazu gehören:

- Verantwortungen
- Aktivitätsmuster der Lernenden
- Kontrolle und Steuerung
- Motivation und Selbstvertrauen der Lernenden

4.4.2 Lerneinheit

In Abb. 4.5 sind die Beziehungen zwischen den Elementen einer kompetenzorientierten Lernumgebung dargestellt. Zu einer Lerneinheit gehören die zur Verfügung stehende Zeit (1), das durch den Lernprozess (hier als Lernkurve dargestellt) erreichte und durch die Leistungskontrolle geprüfte Kompetenzniveau (2) und die Anzahl der im Lernprozess verarbeiteten Konzepte (4).

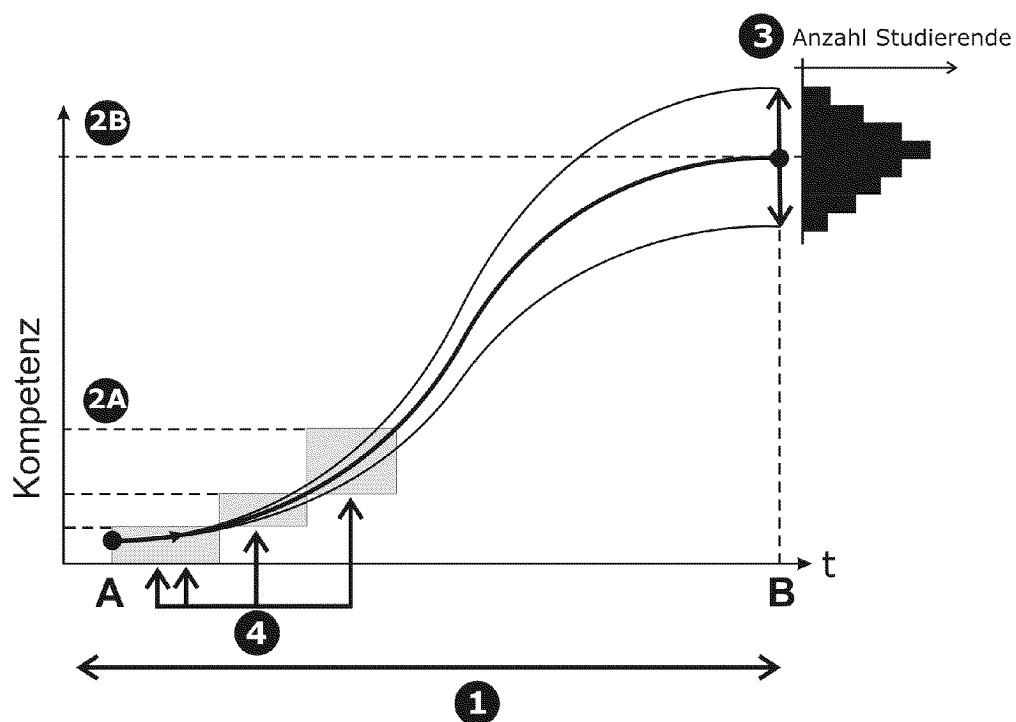


Abb. 4.5: Beziehungen zwischen Elementen einer kompetenzorientierten Lernumgebung. Für eine Lerneinheit steht eine bestimmte Zeit (1) zur Verfügung, in deren Verlauf sich individuelle Lernprozesse als Lernkurven von Zeitpunkt A nach Zeitpunkt B entwickeln. Nach Abschluss des Kurses (Punkt B) erreichen die Lernenden ein durchschnittliches Kurs-Kompetenzniveau (2B) mit einer bestimmten Verteilung der Lernenden entlang der Kompetenzachse (3). Diese kann durch eine Leistungskontrolle zum Zeitpunkt B erhoben werden. In verschiedenen Lerneinheiten werden eine bestimmte Anzahl Konzepte behandelt und verarbeitet (4), was zu einem bestimmten Teil-Kompetenzniveau (2A) führt.

1. Zeit

Die Gesamtzeit, die für eine Unterrichts- oder Lerneinheit zur Verfügung steht, ist auf einen bestimmten Zeitraum begrenzt und richtet sich nach den curricularen Vorgaben. Gemessen wird sie nach dem Wert des ECTS-Kreditsystems [Europäische Kommission 2004]. Ein Kreditpunkt entspricht dabei einem Arbeitspensum von ungefähr 25 bis 30 Stunden. Bei einem kompetenzorientierten System umfasst diese Zeit nicht bloss die Zeit zur Instruktion und Überprüfung von Konzepten, sondern schliesst deren Anwendung und Verknüpfung mit anderen Konzepten zu Kompetenzen mit ein. Weil kompetenzorientierter Unterricht zeitintensiv ist, hat die zur Verfügung stehende Zeit einen

starken Einfluss auf den zu erreichenden Kompetenzlevel und die Anzahl der Konzepte, die während der Lerneinheit vermittelt werden können.

2. Kompetenzniveau

Das Kompetenzniveau einer Unterrichts- oder Lerneinheit einer kompetenzorientierten Lernumgebung entspricht der Kompetenz, welche durch den individuellen Lernprozess in der zur Verfügung stehenden Zeit durch die Verknüpfung unterschiedlicher Teilkompetenzen erreicht werden kann. Dieses Kompetenzniveau kommt nach Abschluss der Lerneinheit als Unterrichts-Output z.B. an einer Prüfung durch eine Leistung zum Ausdruck (folglich Lernziel-Taxonomie von Bloom in Kapitel 2.5.2). Weil der Lernprozess in einer kompetenzorientierten Lernumgebung individuell ist, produziert ein Lernsystem in einer Gruppe von Lernenden eine gewisse Variation im Kompetenzniveau (Abb. 4.5, 3). Eine Charakteristik eines Lernsystems ist deshalb nicht nur ein bestimmtes Kompetenzniveau, sondern auch dessen Variation in der Interaktion mit einer Gruppe von Lernenden. Weil die Zeit zum Erreichen eines Kompetenzniveaus in einer Lerneinheit begrenzt ist, ist es nicht möglich, in einem Kurs Expertenniveau zu erreichen (folglich Abb. 2.6). Die Lernenden sollen aber mit Hilfe der Lernumgebung die Möglichkeit erhalten, ihre Zeit beim Lernen effizient zu nutzen und aus einem ausgewählten Set an Grundkonzepten eine Kompetenz auf einem bestimmten Niveau zu entwickeln.

3. Anzahl und Struktur verarbeiteter Konzepte

In einer Unterrichts- oder Lerneinheit wird eine bestimmte Anzahl von Konzepten behandelt. Die Anzahl Konzepte, die konzeptorientiert vermittelt werden können, richtet sich nach der zur Verfügung stehenden Zeit und der Komplexität der Konzepte. Konzepte werden meistens zu Beginn des Lernprozesses eingeführt, indem sie isoliert betrachtet werden. Bei einer Kompetenz treten Konzepte aber in Kombination auf. Für den kompetenzorientierten Unterricht bedeutet dies, dass Konzepte nicht bloss eingeführt, sondern auch miteinander verknüpft und von den Lernenden verinnerlicht werden müssen.

4. Leistungskontrolle

Die Leistungskontrolle erhebt das nach Abschluss der Lerneinheit erreichte Kompetenzniveau (folglich Kapitel 2.5). Sie dient einerseits dazu, zu prüfen, ob die Lernenden ein definiertes Kompetenzniveau erreicht haben, und kann andererseits dazu eingesetzt werden, eine kompetenzorientierte Lernumgebung inhaltlich zu evaluieren (siehe Kapitel 10.3 „Anwendungsorientierte Leistungskontrolle als Mass für das Vorhandensein von anwendbaren Wissensstrukturen“).

4.4.3 Personengebundene Elemente

1. Verantwortungen

Die Verantwortung für Inhalt und Aufbau von Lernmaterialien liegt bei den Lehrenden, respektive den Entwicklern der Lernumgebung. Die Verantwortung für die Verwendung der Lernmaterialien für den individuellen Lernprozess sollten die Lernenden hingegen selber tragen können. Dies ist nur möglich, wenn die Lernenden im Verlaufe ihres Lernprozesses Selbständigkeit entwickeln können (siehe „Cognitive Apprenticeship Modell“ Kapitel 4.5.3 und Kapitel 6 „Selbstgesteuerte, problembasierte Instruktion“).

2. Aktivitätsmuster der Lernenden

Die Aneignung von Kompetenzen ist von einer aktiven Verarbeitung von Information durch die Lernenden abhängig (folglich Kapitel 2 „Kompetenzorientierter Unterricht“). Das maximale Kompetenzniveau wird dann erreicht, wenn die Aktivität der Lernenden über die ganze Unterrichtszeit möglichst hoch ist. Weil die Aktivität der Lernenden durch die Lernmethode generiert wird, ist sie so zu wählen, dass die Lernenden bereits von Beginn weg zu Aktivitäten herausgefordert werden. Aktivitäten der Lernenden alleine garantieren aber noch keine Aneignung von Kompetenzen (folglich Kapitel 4.2.3 „Lernmethoden und didaktische Modelle“). Die Aktivitäten müssen so mit der Aneignung und Anwendung von Konzepten einhergehen, dass daraus zunächst Teilkompetenzen und später Kompetenzen aufgebaut werden (siehe Kapitel 5 „Problembasiertes Lernen“ und 7 „Didaktische Modelle“).

3. Kontrolle und Steuerung

Für den Verlauf des Lernprozesses ist es von entscheidender Bedeutung, wie und durch wen ein Lernsystem zu unterschiedlichen Zeitpunkten gesteuert und kontrolliert wird. In einem von Lehrenden überkontrollierten Lernsystem können sich Abhängigkeiten ergeben, die dazu führen, dass Lernende keine Selbständigkeit entwickeln. Überlässt man hingegen die Lernenden im System sich selber, kommt der Lernprozess bei einem gewissen Anteil nicht in Gang. Sie werden im späteren Verlauf durch die hohen Anforderungen überfordert, was zu Frustration und Unsicherheiten führt. In einem System ganz ohne äussere Kontrolle durch die Lehrenden entwickelt sich ebenfalls keine Selbständigkeit. Das bedeutet, dass die Kontrolle und Steuerung in einem idealen Lernsystem über die Systemstrukturen variiert und schrittweise dem Lernenden übertragen wird (siehe Abschnitt 4.5 „Entwicklung Lernprozess-unterstützender Strukturen“).

4. Motivation und Selbstvertrauen der Lernenden

Durch die indirekte Instruktion müssen die Lernenden in einer kompetenzorientierten Lernumgebung zunehmend die Steuerung ihres Lernprozesses übernehmen. Voraus-

setzung dafür ist *Motivation* und *Selbstvertrauen* beim Aneignen einer Kompetenz. Diese können nicht als gegeben betrachtet werden, sondern sollen durch die Interaktion mit den Systemstrukturen entstehen können. Das heisst, dass sich im Verlaufe des Lernprozesses eine ganze Reihe *motivationaler Faktoren* in eine definierte Richtung entwickeln sollen, um Motivation und Selbstvertrauen aufzubauen und zu verstärken (Tab. 4.1). Für den Lernprozess ist es von entscheidender Bedeutung, inwiefern sich die Lernenden in dessen Verlauf selber motivieren können.

Psychologisches Konzept	Motivationaler Faktor	Erklärung
Psychologisches Anreizmodell	intrinsische Motivation	von innen gerichtete selbstgesteuerte Motivation
	extrinsische Motivation	von aussen erteilte Motivation
Kontrollüberzeugung	interaktionistische bis internale Kontrollüberzeugung	Ursache eines Ereignisses wird bei sich selber gesehen
Selbstwirksamkeit (<i>Self efficacy</i>)	positive Selbstwirksamkeitsüberzeugung	Überzeugung oder Erwartung, ein Problem zu meistern

Tab. 4.1: Wichtige motivationale Faktoren einer kompetenzorientierten Lernumgebung (weitere Details siehe Kapitel 5.3.2 „Motivationale Grössen beim Lösen von Problemen“).

Eine *intrinsische Motivation* mit *internaler Kontrollüberzeugung* kommt vor allem durch eigenes wirkungsvolles Handeln zustande. Die Handlungsorientierung in einer Lernumgebung ermöglicht es, dass sich die Lernenden intrinsisch motivieren können [Rheinberg 2000]. Eine von aussen erteilte, *extrinsische Motivation* ist als Bestätigung für die erfolgreiche Bewältigung einer Etappe des Lernprozesses für eine erfolgreiche Kompetenzaneignung ebenfalls von grosser Wichtigkeit.

Um zu erreichen, dass die Lernenden die Ursache für Fortschritte im Lernprozess bei sich selber sehen (*internale Kontrollüberzeugung*) und davon überzeugt sind, ein Problem mit den eigenen Mitteln meistern zu können (*positive Selbstwirksamkeit*), müssen die Lernenden Vertrauen in eine Lernmethode und die Lernprozess-unterstützenden Strukturen einer Lernumgebung entwickeln können. In einer idealen kompetenzorientierten Lernumgebung wird ein Lernprozess durch hauptsächlich intrinsische, von Zeit zu Zeit aber auch extrinsische Motivation positiv verstärkt. Zufriedenheit bei der Arbeit mit einer Lernumgebung und das Vertrauen, mit Hilfe der zur Verfügung gestellten Strukturen das geforderte Kompetenzniveau erreichen zu können, sind eine wichtige Voraussetzung für selbstgesteuertes Lernen.

Die Umsetzung dieser motivationalen Faktoren wird im Zusammenhang mit der Lernmethode des *Problembasierten Lernens* im nächsten Kapitel behandelt.

4.5. Entwicklung Lernprozess-unterstützender Strukturen

Der zentrale Prozess innerhalb eines konstruktivistischen Lernsystems ist der individuelle Lernprozess jedes einzelnen Individuums. Strukturen, welche diese individuellen Lernprozesse während der Kompetenzerwerb unterstützen, stehen deshalb im Zentrum der Entwicklung. Zu diesen Prozess-unterstützenden Strukturen gehören:

- *Lernmaterialien*, die ein aktives Lernen ermöglichen und den gezielten und unabhängigen Kompetenzerwerb unterstützen (siehe Abb. 4.3)
- *Leistungskontrollen*, mit Feedbacks zum Lernprozess oder den Lernprodukten der Studierenden (siehe Kapitel 9 „Einsatz der computergestützten Lernumgebung im Unterricht“).
- *Ablaufstrukturen*, welche die Verwaltung grösserer Gruppen von Lernenden innerhalb einer Lernumgebung ermöglichen (siehe Kapitel 9 „Einsatz der computergestützten Lernumgebung im Unterricht“).

Bei einer orts- und zeitunabhängigen indirekten Instruktion kommt den Systemstrukturen grosse Beachtung zu, weil sie den von aussen sichtbaren Teil des Lernsystems ausmachen und die Lernenden über diese Strukturen mit den Lerninhalten in Kontakt kommen (siehe Abb. 4.2 und Abb. 4.3). Diese Strukturen sind schliesslich auch jener Teil des gesamten Lernsystems, der von Lernenden und Lehrenden für Fortschritte beim Kompetenzerwerb verantwortlich gemacht wird.

4.5.1 Ergonomische Entwicklung und Optimierung von Systemstrukturen

Das Ziel einer ergonomischen Entwicklung und Optimierung von Objekten ist, dass sie von Menschen zur Lösung einer Aufgabe ideal eingesetzt werden können [ISO 9241, 2006]. Prozess-unterstützende Strukturen stehen im Zentrum einer solchen ergonomischen Entwicklung und Optimierung einer kompetenzorientierten Lernumgebung. Die Strukturen werden dahingehend entwickelt und verbessert, dass die Lernenden mit ihrer Hilfe Informationen zum Aufbau von Kompetenzen nutzen können (folglich Abb. 4.2). Die *Gebrauchstauglichkeit (Usability)* bezeichnet die Eignung eines Produkts zur Nutzung durch bestimmte Benutzende in einem bestimmten Benutzungskontext, um eine vorgegebene Aufgabe effektiv, effizient und zufriedenstellend zu lösen (Tab. 4.2) [ISO 9241-11, 1998].

Dimension der Usability	Erklärung
Effektivität (<i>Effectiveness</i>)	Erreichen die Benutzenden mit dem System ihre Ziele?
Effizienz (<i>Efficiency</i>)	Wie gut erreichen die Benutzenden in der zur Verfügung stehenden Zeit mit dem System ihre Ziele?
Zufriedenheit (<i>Satisfaction</i>)	Welche subjektive Zufriedenheit zeigen die Benutzenden bei der Bedienung?

Tab. 4.2: Dimensionen der Usability nach ISO 9241-11 [ISO 9241-11, 1998].

Usability von Unterrichtsstrukturen

Werden Unterrichtsstrukturen als Produkt betrachtet, mit dessen Hilfe Anwenderinnen und Anwender auf effektive, effiziente und zufriedenstellende Art und Weise Kompetenzen aneignen können, ist deren Usability auf diese Zielsetzung hin zu optimieren. Im traditionellen Grundlagenunterricht mit direkter Instruktion besuchen Studierende Vorlesungen und lesen Textbücher. Ein gut geschriebenes Lehrbuch oder Unterrichtsskript und eine gut gehaltene Vorlesung können als Systemstrukturen gute Usability-Werte erreichen, wenn die Unterrichtsstrukturen der Lernumgebung so auf die Zielsetzung hin optimiert sind, dass Lernende Konzepte als deklaratives Wissen wiedergeben können (Kompetenzlevel K1 und K2 Lernzieltaxonomie von Bloom, Kapitel 2.5.2). Ist das Ziel jedoch die Aneignung von Kompetenzen, welche die Aneignung von unterschiedlichen Wissensarten und die Anwendung des Lerninhalts miteinschließt (Kompetenzlevel K3 und höher), kommen die Strukturen einer solche Lernumgebung an ihre Grenzen. Ihre Usability-Werte fallen für diese Zielsetzung schlechter aus, weil sich mit dem Einsatz des Mediums Buch, sowie der direkten Instruktion (Vorlesung) praktische Erfahrung und Fähigkeiten kaum vermitteln lassen. Diese sind für den Aufbau von Kompetenzen aber von entscheidender Bedeutung (folglich Kapitel 2 „Kompetenzorientierter Unterricht“).

Es muss deshalb nach Lösungen gesucht werden, Lernprozess-unterstützende Strukturen zu schaffen, die eine Aneignung von Kompetenzen ermöglichen. Der Fokus einer solchen ergonomischen Entwicklung und Optimierung richtet sich auf die Interaktion der Lernenden mit den Strukturen der Lernumgebung bei der Kompetenzaneignung (siehe Abb. 4.1, Interaktion A).

4.5.2 Anforderungen an Lernprozess-unterstützende Strukturen

Es spielt eine Rolle, auf welcher Kompetenzstufe einer Ausbildung man die Anforderungen an Lernprozess-unterstützende Strukturen betrachtet. In einer Grundlagen-Lehrveranstaltung können und wissen die Lernenden am Anfang meist noch wenig, weil die Wissensnetze in einem Gebiet noch nicht ausgebildet sind. Die Lernenden sind auf Hilfen, Inputs und Feedbacks angewiesen, die durch Prozess-unterstützende Strukturen bereitgestellt werden müssen. Die Lernenden sollen dann aber durch Interaktionen mit den Prozess-unterstützenden Strukturen im Laufe ihrer Kompetenzaneignung

Fortschritte machen und immer mehr Selbständigkeit entwickeln können. Die Prozessunterstützenden Strukturen sollen so flexibel sein, dass der Lernprozess in Gang gebracht und die Entstehung von Selbständigkeit ermöglicht wird. Bei der Interaktion zwischen den Lernprozess-unterstützenden Strukturen und den Lernenden ist zu berücksichtigen, dass sich ein Teil des Lernsystems, nämlich die Lernenden selber, durch den durch das System hervorgerufenen Lernprozess verändern und weiter entwickeln.

Die Herausforderung bei der Entwicklung von Systemstrukturen für eine kompetenzorientierte Lernumgebung ist, dass sich die Strukturen den wechselnden Bedingungen während des Lernprozesses anpassen können. Die Lernenden brauchen am Anfang des Lernprozesses nicht dieselbe Menge und Qualität an Information wie im späteren Verlauf des Prozesses. Steht zu Beginn vor allem generische Input-Information im Zentrum, ist im späteren Verlauf individuelle Feedback-Information wichtig (Abb. 4.6).

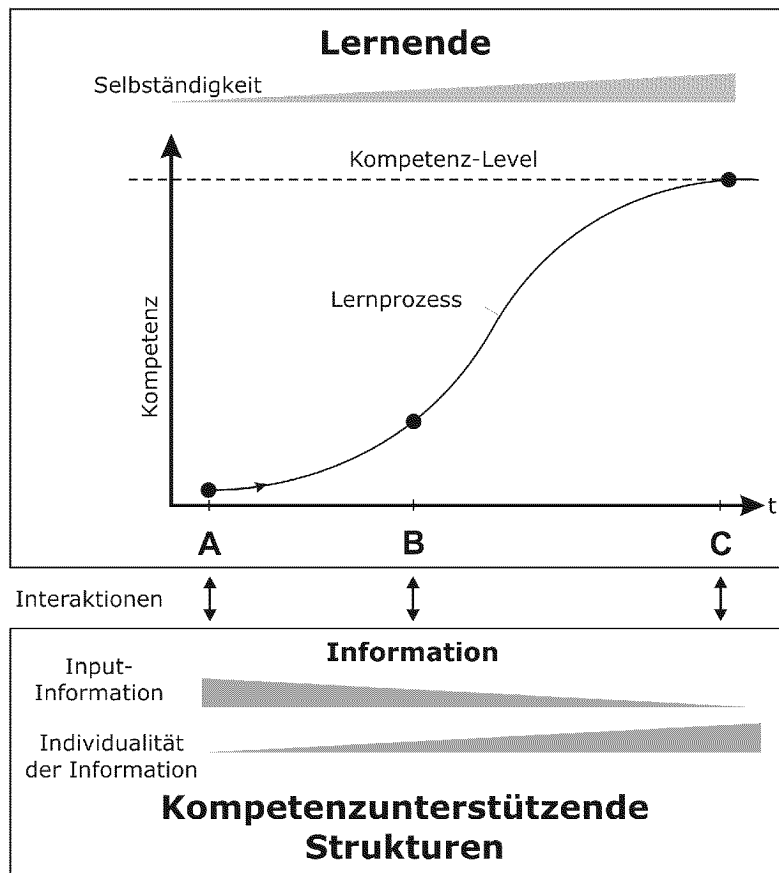


Abb. 4.6: Die Interaktion zwischen den Lernenden und den Systemstrukturen in einer kompetenzorientierten Lernumgebung verändert sich im Verlaufe der Kompetenzerwerbungs laufend. Bei der Entwicklung von kompetenzunterstützenden Strukturen muss berücksichtigt werden, dass die Lernenden zu Beginn des Lernprozesses (Zeitpunkt A) noch weitgehend unselbständig sind, weil noch wenige Kompetenzen vorhanden sind. Die Lernenden müssen mit viel Input-Information versorgt werden. Im Verlaufe des Lernprozesses (Zeitpunkt B) kann mit zunehmender Selbständigkeit der Lernenden die Menge der Information reduziert werden. Die Individualität der benötigten Informationen (z.B. für Feedback) nimmt hingegen zu. Dadurch verändert sich die Art der Interaktion zwischen den Prozessunterstützenden Strukturen und den Lernenden bis zum Erreichen des angestrebten Kompetenzlevels (Zeitpunkt C).

Effektive Systemstrukturen können nicht entwickelt werden, indem sie isoliert vom Gesamtsystem betrachtet werden. Voraussetzung für die erfolgreiche Entwicklung von Prozessunterstützenden Systemstrukturen ist deshalb, dass die Anforderungen an ein kompetenzorientiertes System und dessen Eigenschaften verstanden werden.

Im nächsten Abschnitt wollen wir uns deshalb mit dem didaktischen Modell des Cognitive Apprenticeship auseinandersetzen, das zum Ziel hat, Systemstrukturen zur Verfügung zu stellen, um einen idealen Transfer von Wissen und Können zu ermöglichen.

4.5.3 Das Cognitive Apprenticeship-Modell

Im Modell des *Cognitive Apprenticeship* werden Methoden der Berufslehre auf die Schule übertragen, indem effektive Bestandteile eines Meister-Lehrling-Verhältnisses auf kognitive Lernziele angewandt werden [Collins 1989]. Das Modell wurde mit dem Ziel entwickelt, einen interaktiven Wissenstransfer vom Experten zum Novizen zu ermöglichen. Das Cognitive Apprenticeship-Modell ist sowohl problem- als auch handlungsbasiert und berücksichtigt die Beziehung zwischen den Lehrenden und den Lernenden sowie die zunehmende Selbständigkeit der Lernenden. Die Lernenden kontrollieren ihre Lern- und Problemlöseprozesse weitgehend selber unter Beratung durch den Meister. Tab. 4.3 zeigt die wesentlichen Elemente des Cognitive Apprenticeship-Modells.

	Element	Erklärung
1.	Inhalt (<i>content</i>)	Neben Fakten-, Konzept- und prozeduralem Wissen soll auch metakognitives Wissen vermittelt werden, um in einem Gebiet Problemlöse-Fähigkeiten zu erlernen und zu verbessern [Collins 1989].
2.	Situiertes Lernen (<i>situated learning</i>)	Wissen sollte in jenem Kontext gelernt werden, in dem es auch in der realen Welt auftritt [Collins 1989].
3.	Modellieren und Erklären (<i>modeling and explaining</i>)	Ein Experte führt eine Problemlösung vor und kommentiert seine Strategie [Collins 1991].
4.	Coaching (<i>coaching</i>)	Lernende werden beim Problemlösen beobachtet und nach Bedarf mit Hinweisen und Hilfen unterstützt.
5.	Ausblendung (<i>fading</i>)	Die Lernenden werden mehr und mehr sich selber überlassen [Strittmatter & Niegemann 2000].
6.	Artikulation (<i>articulation</i>)	Es wird implizites Wissen verfügbar gemacht, indem Lernende ihre Handlungen reflektieren und ihre Entscheidungen und Strategien begründen müssen [Mandel 1997].
7.	Reflexion (<i>reflection</i>)	Studierende blicken auf ihre Leistung zurück, um den eigenen Fortschritt zu analysieren [Collins & Brown 1988].
8.	Erkunden (<i>exploration</i>)	Studierende lösen ein Problem eigenständig. Sie werden ermutigt, verschiedene Strategien und Hypothesen auszuprobieren, um ihre Effekte zu erkunden.
9.	Abfolge (<i>sequencing</i>)	Die Abfolge einer Instruktion erfolgt vom Einfachen zum Komplexen und mit zunehmender Vielfältigkeit.

Tab. 4.3: Elemente des Cognitive Apprenticeship [Collins 1989].

Problemlösen als Zentrum der Tätigkeit der Lernenden

Als Zentrum der Lernenden-Tätigkeit werden im Cognitive Apprenticeship-Modell Probleme aus einem Gebiet gelöst, um neben Fakten-, Konzept- und prozeduralem

Wissen auch metakognitives Wissen zu erwerben. Die Lernenden treffen dabei die Probleme und Konzepte in jenem Kontext an, in dem sie in der Realität auch auftreten. Wissensinhalte werden dadurch nicht aus ihrem Kontext gelöst und bleiben für die spätere Anwendung weniger träge. Eine Charakteristik des Cognitive Apprenticeship-Modells ist, dass die Lehrenden die Rolle eines Coaches einnehmen. Die Lehrenden zeigen den Lernenden zunächst, wie ein Problem zu lösen ist und helfen ihnen dann, das Problem selber zu lösen. Die Lehrenden nehmen ihre tutorielle Aktivität im Laufe der Zeit schrittweise zurück und fördern dadurch die Selbständigkeit der Lernenden beim Lösen von Problemen und die Interaktion der Lernenden mit dem Inhalt. Durch Artikulation und Reflexion des eigenen Lernprozesses werden einerseits implizites Wissen bewusst und andererseits die individuellen Fortschritte sichtbar gemacht.

In 4 Phasen zu Selbständigkeit und Kompetenz

Das Instruktionsmodell des Cognitive Apprenticeship ist, eine gute Implementierung vorausgesetzt, eine effiziente Form des Wissenstransfers vom Experten zum Lernenden. Es wechseln sich darin verschiedene Phasen mit unterschiedlichen Aktivitäten der Lehrenden und Lernenden ab (Tab. 4.4). Die Lehrpersonen lösen ein Problem vor und erklären ihr Vorgehen (Phase 1), die Lernenden beobachten und machen nach (Phase 2). Im weiteren Verlauf lösen die Lernenden weitere Probleme immer selbständiger und sind nur noch auf einzelne Hinweise und Tipps der Lehrenden angewiesen (Phase 3). Als letzter Schritt des Wissenstransfers müssen die Lernenden den ganzen Problemlösungsprozess zusammenfassen, indem sie ihn mit eigenen Worten erklären (Phase 4). Die Lehrenden hören zu und geben dazu ein Feedback.

	Lehrende	Lernende
Phase 1	Problem vorlösen und kommentieren	Beobachten, Problemlösungsprozess nachvollziehen
Phase 2	Hilfe beim Lösen des Problems, schrittweise Reduktion der Hilfe	Durch den Lehrenden gelöstes Problem nachlösen
Phase 3	Hinweise und Tipps zur Lösung des Problems, sonst keine Hilfe	Problem selbständig lösen
Phase 4	Zuhören und Feedback	Zusammenfassen des Problemlösungsprozesses

Tab. 4.4: Aktivitäten von Lehrenden und Lernenden während verschiedener Phasen des Apprenticeship-Modells (Tab. 4.3).

Schrittweise Verfeinerung der Interaktion zwischen Experten und Novizen

Beim Cognitive Apprenticeship-Modell wird die Interaktion zwischen einem Experten und einem Novizen schrittweise verfeinert. Im Zentrum steht ein Problemlöseprozess. Die Lehrpersonen machen zunächst ihre Gedankengänge für die Lernenden sichtbar,

im weiteren Verlauf werden die Rollen vertauscht und die Lernenden teilen ihre Gedankengänge den Lehrpersonen mit. Durch das Modell können Denkprozesse nicht nur sichtbar gemacht werden, es erlaubt Anfängerinnen und Anfängern auch, aktiv am Problemlösen teilzunehmen [Mandel 1997]. Mittels Unterstützung von Experten können die Lernenden trotz noch beschränkter kognitiver Ressourcen relevantes Wissen erwerben. Da diese Aneignung in einem authentischen Handlungskontext und im Austausch mit Experten geschieht, kann die kontextspezifische Bedeutung des Wissens erfahren werden. Gleichzeitig werden die Kommunikation in einer Fachsprache (Terminologie) und die damit verbundene Art, das eigene Tun zu reflektieren, gelernt. Durch das Cognitive Apprenticeship-Modell wachsen die Lernenden schrittweise in eine Expertenkultur hinein. Weil das zu lösende Problem und das Handeln der Lernenden im Zentrum stehen, wirkt die Interaktion zwischen Experten und Novizen weniger belehrend als eine traditionelle Lehrer-Schüler-Beziehung.

4.6. Fazit: Anforderungen an die Entwicklung einer kompetenzorientierten Lernumgebung

In diesem Kapitel haben wir kompetenzorientierte Lernumgebungen als umfangreiche und komplexe Systeme betrachtet. Der zentrale Prozess in diesem System ist die Interaktion zwischen den Lernenden und dem Lerninhalt (Abb. 4.1, Interaktion A). Diese Interaktion zeigt sich durch einen individuellen Lernprozess, der verschiedene Phasen mit zunehmender Selbständigkeit durchläuft. Systemstrukturen, welche diesen Prozess unterstützen, stehen im Zentrum der Entwicklung kompetenzorientierter Lernsysteme. Zur Kompetenzaneignung gehören neben dem Aufbau eines Wissensnetzes eine Reihe von psychologischen Faktoren (Motivation, Kontrolle, Steuerung), die ebenfalls über die Systemstrukturen ihre Wirkung auf den Lernprozess ausüben.

Im Zusammenhang mit der ergonomischen Entwicklung von Lernprozess-unterstützenden Strukturen einer Lernumgebung bei der Aneignung von Kompetenzen sind folgende Fragen zu klären:

1. Welche Phasen der Kompetenzaneignung sind in einem Lernsystem automatisierbar?

Das Ziel der Entwicklung einer stabilen Lernumgebung, mit der auch grosse Gruppen von Studierenden möglichst unabhängig von Ort, Zeit und Lehrpersonen individuell Grundkompetenzen erlernen können, führt zur Frage, welche Phasen der Kompetenzaneignung durch Lernmaterialien und andere Strukturen automatisierbar sind und wie diese sinnvoll mit elektronischen Mitteln unterstützt werden kann. Phasen mit hoher Individualität sind schwierig zu automatisieren (siehe Tab. 1.1, Stufe 6).

2. Wie können aus Konzepten Kompetenzen aufgebaut werden?

Konzepte können beim Aufbau von Kompetenzen nicht bloss bereitgestellt, verteilt oder mitgeteilt werden, sollen sie zum Lösen von Problemen eingesetzt werden können (folglich Abb. 4.2). Es stellt sich die Frage, wie diese Konzepte in Systemstrukturen integriert werden müssen, damit ein Lernprozess in Gang kommt und im weiteren Verlauf des Prozesses sinnvolle Kompetenzen aufgebaut werden können.

3. Wie wird ein kompetenzorientierter Lernprozess in Gang gebracht?

Zu Beginn eines Lernprozesses sind die Lernenden meist noch unselbständig, weil kein aktives Wissen zur Verfügung steht. Eine ganze Reihe von Prozessen soll bei einem möglichst grossen Anteil der Studierenden in Gang gebracht werden. Einerseits steht der Aufbau eines aktiven Wissensnetzes an und andererseits muss eine ganze Reihe psychologischer Faktoren dahingehend beeinflusst oder verändert werden, dass sie den Lernprozess positiv verstärken sowie Vertrauen und Zufriedenheit im Umgang mit der Lernumgebung schaffen.

4. Wie können Lernprozess-unterstützende Strukturen in eine Lernumgebung implementiert werden?

Die Anforderungen an Prozess-unterstützende Strukturen einer Lernumgebung sind komplex. Es stellt sich die Frage, wie sie entwickelt und innerhalb einer Lernumgebung zusammenspielen müssen, damit sie genügend flexibel bleiben, um den Lernprozess optimal zu unterstützen.

Diese Fragen müssen bei der Entwicklung und Umsetzung eines didaktischen Modells berücksichtigt werden, soll die zu entwickelnde Lernumgebung kompetenzorientiert werden. Die beiden im Rahmen dieser Arbeit entwickelten didaktischen Modelle werden in Kapitel 7 beschrieben und in Kapitel 8 in Lerneinheiten umgesetzt.

Kapitel 5

Problembasiertes Lernen

Bei der Methode des Problembasierten Lernens (*Problem based learning*, PBL) werden die Lernenden durch das Lösen von konkreten, praxisnahen Problemen aktiv am Unterricht beteiligt. Die Lernenden memorisieren nicht, sie konstruieren ihr Wissen mit Hilfe zur Verfügung stehender Information weitgehend selbständig. Die Rollen von Lehrenden und Lernenden verändern sich dadurch grundlegend. Während die Lernenden die Steuerung ihres Lernprozesses weitgehend übernehmen, verändert sich die Aufgabe der Lehrenden von der Wissenspräsentation in Richtung Lernprozessbegleitung. Eine andere neue Aufgabe ist die der Entwicklung von Lernumgebungen mit indirekter Instruktion durch die Konstruktion und Implementierung von Problemaufgaben.

5.1. PBL als Lehr- und Lernmethode

Die Lehrmethode des PBL wurde in den 50er Jahren als Instruktionsprinzip für die medizinische Ausbildung entwickelt, um Konzepte zusammen mit einer Anwendungssituation zu lehren und lernen und wurde seither auf viele andere Bereiche wie Naturwissenschaften, Rechtswissenschaften und die Ingenieurwissenschaften angewandt [Merrill 1994].

5.1.1 Definitionen von PBL

Folgende zwei Definitionen sollen die wesentlichen Merkmale des PBL als Lernmethode aufzeigen:

Problem-based learning is the process whereby a student learns by using a problem as a stimulus to discover what information is needed to understand and facilitate the resolution of a problem. The problem is encountered right at the beginning of the learning process [Guilbert 1977].

Problem-based learning is centered on the learning that emanates from a real problem. The learning is more important than the solution of the problem [Prpic & Hadgraft 1999].

Wie aus den beiden Definitionen ersichtlich wird, dient beim Problembasierten Lernen ein praxisnahes Problem oder ein Fall als Fokus für das Lernen. Dabei ist bei vielen Formen von PBL keine Einführung vorgesehen, die Lernenden werden meist gleich zu Be-

ginn mit einer Problemsituation konfrontiert, die mit Hilfe bereitgestellter Information gelöst werden soll. Die Aneignung von neuem Wissen soll durch den dabei entstehenden kognitiven Konflikt herausgefordert werden und die Lernenden dazu motivieren, mehr zu wissen, um das Problem oder den Fall zu verstehen und weiterzuentwickeln. Das Problem wird einerseits als motivierendes Mittel zur Erarbeitung neuen Wissens eingesetzt und dient andererseits dazu, den Konzepten einen Sinnzusammenhang zu geben. Auf diese Weise sollen neben der eigentlichen Problemlösung transferfähiges Wissen und fachspezifische Lern- und Denkstrategien erworben werden. Dieses Wissen wird durch einen individuellen Konstruktionsprozess bei einer engen Verbindung zur Handlung erworben. Das während dem Problemlöseprozess erarbeitete Produkt macht den Lernprozess sichtbar und dadurch auch implizites Wissen zugänglich (folglich Abb. 2.8).

Definierte Abläufe beim PBL

PBL ist dadurch gekennzeichnet, dass es sich an festgelegte Abläufe mit sich wiederholenden Phasen hält. Auf diese Weise können nicht nur inhaltliches Wissen, sondern auch generelle Problemlösungsstrategien und Herangehensweisen gelernt werden. Ein bekanntes solches Verfahren von PBL ist das „Siebensprung-Modell“ (*seven steps* oder *jumps*), das im Wesentlichen auf folgenden vier generellen Elementen eines Problemlösungsprozesses basiert [Steiner 2004]:

- Problemexposition
- Problemdiskussion
- Selbststudium
- Ergebnisdiskussion

In der ersten Phase wird den Lernenden ein Problem aus der Praxis dargelegt, für das eine Erklärung oder eine Lösung zu erarbeiten ist. In einem zweiten Schritt haben die Lernenden das Problem differenziert zu analysieren, indem sie Begriffe klären und diese mit ihrem Vorwissen abstimmen und ordnen, so dass mögliche Hypothesen formuliert werden können. Im dritten Schritt erarbeiten sich die Studierenden selbständig Antworten auf die in Schritt 2 formulierten Hypothesen. Der Problemlöseprozess wird durch eine Ergebnisdiskussion abgeschlossen, indem das erarbeitete Wissen vorgestellt, diskutiert und kritisch beurteilt wird.

Zwei Vorteile von PBL sollen im Folgenden herausgehoben werden:

- die hohe Transfer-Wirksamkeit (Abschnitt 5.1.2)
- die Eignung zum selbstgesteuerten Lernen (Abschnitt 5.1.3).

5.1.2 Hohe Transfer-Wirksamkeit

Zwei wesentliche Eigenschaften machen das Problembasierte Lernen, eine erfolgreiche Implementierung vorausgesetzt, zu einer transferwirksamen Lernmethode: Die *Kontextgebundenheit* und die *Erkenntnisorientierung*. Die Kontextgebundenheit meint, dass für den nachhaltigen Aufbau eines Wissensnetzes das spätere Anwendungsfeld miteingebunden wird (folglich „situiertes Wissen“, Kapitel 2.4.1). Der Kontext wird dadurch immer mitgelernt und ermöglicht, aktives Wissen anzueignen und wichtige inhaltliche Erfahrungen zu sammeln. Kontextualisiertes Wissen kann im Gedächtnis besser gespeichert und abgerufen werden als ohne Kontext gelerntes Wissen [Reusser 2005]. Zudem basiert PBL auf Erkenntnissen statt der Reproduktion von Inhalten, was ebenfalls der Entstehung von tragem Wissen entgegenwirkt [Gruber 2000].

5.1.3 Eignung für selbstgesteuertes Lernen

Eine Voraussetzung dafür, dass in einer Lernumgebung unabhängig von Ort, Zeit und Person gelernt werden kann, sind Möglichkeiten zur Selbststeuerung durch die Lernenden. Die Implementierung von PBL ist eine Möglichkeit, in einer Lernumgebung selbstgesteuertes Lernen zu erreichen. Damit sich auch Novizen in einer Lernumgebung selber steuern können, bedarf es der Implementierung von geeigneten Strukturen und Prozessen in Form eines didaktischen Modells, die eine Selbststeuerung positiv unterstützen und aufbauen.

Für die Implementierung von selbstgesteuertem PBL in eine Lernumgebung soll die Lernmethode unter folgenden zwei Aspekten betrachtet werden:

- Aufbau von Kompetenzen durch PBL (Abschnitt 5.2)
- Motivationale und emotionale Faktoren des PBL (Abschnitt 5.3)

5.1.4 Einsatz von PBL zur Aneignung von ICT-Kompetenzen im Informationsarbeitsplatz

Folgende Punkte sprechen für den Einsatz von PBL zur Aneignung von ICT-Kompetenzen im Informationsarbeitsplatz (folglich Kapitel 3 „ICT-Kompetenzen für die wissenschaftliche Tätigkeit“):

1. Aneignung von ICT-Kompetenzen in realen Situationen

Das Lernen von ICT-Kompetenzen im Informationsarbeitsplatz ermöglicht die Aneignung der Kompetenzen in jenem Kontext, in dem die Lernenden sie in der Realität auch antreffen oder antreffen werden. So wie die Kompetenz „Autofahren“ mit Vorteil im realen Verkehr gelernt wird, werden Informatik-Kompetenzen am besten in Kombi-

nation mit realen Problemen eingeübt. Die durch die Problemstellung erzeugten realen Handlungen sind eine Voraussetzung für die Aneignung von ICT-Kompetenzen.

2. Integrative Funktion für eine kombinierte Aneignung von ICT-Konzepten, Fertigkeiten und Fähigkeiten

Statt Konzepte, Fertigkeiten und Fähigkeiten isoliert voneinander in Theorie- und Anwendungskursen zu lehren, ermöglicht die Problemorientierung deren kombinierte Aneignung im Informationsarbeitsplatz. Weil die Aneignung von Konzepten, Fertigkeiten und Fähigkeiten von Anfang an im Zusammenhang mit einer Anwendungssituation geschieht, sehen die Lernenden, wofür sie lernen und wie das Gelernte angewendet werden kann. Das angeeignete Wissen wird später weniger träge bleiben. Die kombinierte Aneignung verschiedener Wissensdimensionen durch geeignete Problemstellungen ermöglicht darüber hinaus das Lernen von generellen Problemlösungsstrategien (siehe Kapitel 2.3.2 „Verschiedene Arten von Wissen“).

3. Wissenschaftliche Tätigkeit als Berufsziel der Lernenden

Werden zur Aneignung von ICT-Kompetenzen Probleme aus dem Fachgebiet der Lernenden (z.B. den Naturwissenschaften) verwendet, werden sie von dieser Zielgruppe als Teil ihrer späteren wissenschaftlichen Tätigkeit verstanden und als Beitrag zu ihrer Berufsbildung gesehen.

5.2. Aufbau von Kompetenzen durch PBL

5.2.1 Problemaufgaben als Vehikel der Aneignung von Kompetenzen

Beim Aufbau von Kompetenzen durch PBL sind Problemaufgaben nicht nur Fokus des Lernens, sondern dienen darüber hinaus als Struktur oder Gerüst für die Aneignung der Kompetenzen [Reusser 2005]. Durch das Lösen der Problemaufgaben werden Konzepte zusammen mit Fertigkeiten und Fähigkeiten in Handlungen umgesetzt. Finden die Handlungen in Kombination mit einem geeigneten didaktischen Modell statt, führen sie mit den Erfahrungen der Lernenden zu einem Erkenntnisgewinn und gemeinsam mit anderen Konzepten zu einer Kompetenz (Abb. 5.1).

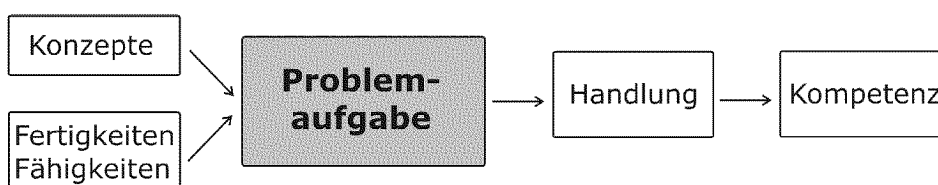


Abb. 5.1: Kompetenzaneignung durch PBL: Die Problemaufgabe setzt Konzepte, Fertigkeiten und Fähigkeiten in Handlungen und Kompetenzen um.

Beim Aufbau von Kompetenzen durch PBL kommt der Entwicklung von Problemaufgaben und deren Implementierung zu einem problembasierten Kurs eine Schlüsselrolle zu. In einem problembasierten Kurs wird eine Reihe von Problemaufgaben zur Aneignung einer Kompetenz eingesetzt. Wir möchten uns deshalb zunächst einigen Grössen zuwenden, die im Zusammenhang mit dem Aufbau von Kompetenzen mittels PBL eine Rolle spielen. Dazu gehören der *Aufgabenraum* und der *Definitionsgrad* einer eingesetzten Problemaufgabe. Diese Grössen führen zu unterschiedlichen Aufgabentypen. Die Konstruktion und die Auswahl von Problemaufgaben werden im Rahmen des Kapitels 8 „Entwicklung und Umsetzung kompetenzorientierter Lerneinheiten“ behandelt.

1. Aufgabenraum einer Problemaufgabe

Abb. 5.2 und Tab. 5.1 fassen einige wichtige Grössen der Kompetenzaneignung mittels Problemaufgabe zusammen. Zur Lösung der Problemaufgabe ist eine durchschnittlich minimale Anfangskompetenz (*Input-Kompetenz*) erforderlich. Im Verlaufe des Problemlösungsprozesses muss eine bestimmte Anzahl an Konzepten, Fertigkeiten und Fähigkeiten angewendet werden, was mit einem Zeitaufwand verbunden ist. Nach dem Lösen der Problemaufgabe weisen die Lernenden aufgrund der gemachten Erfahrungen beim Anwenden der Konzepte, Fertigkeiten und Fähigkeiten eine Kompetenz auf, die höher liegt als die Input-Kompetenz. Sie kann als durch eine Problemaufgabe generierte *Output-Kompetenz* bezeichnet werden. Das Niveau dieser Output-Kompetenz hängt ab von der Zeit, die zur Lösung einer Problemaufgabe aufzuwenden ist. Je höher die Output-Kompetenz, desto mehr Zeit ist für die Bearbeitung einer Problemaufgabe aufzuwenden. Daraus ergibt sich der Aufgabenraum einer im PBL eingesetzten Problemaufgabe (siehe auch Abb. 5.4, A).

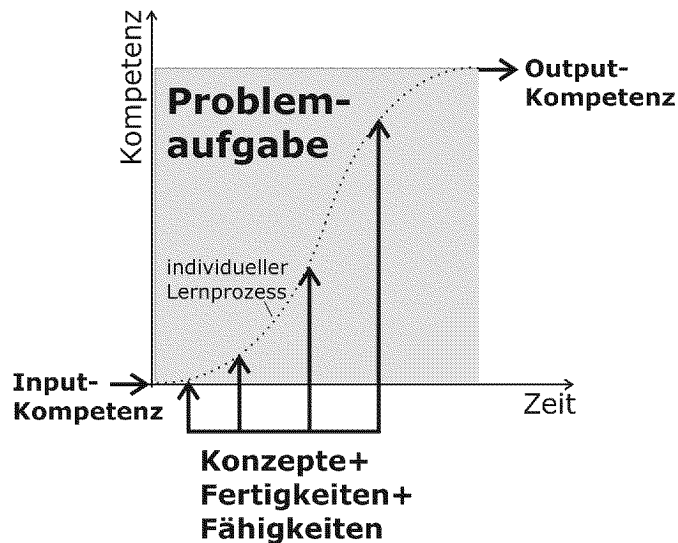


Abb. 5.2: Beziehung verschiedener Größen beim Aufbau einer Kompetenz durch eine Problemaufgabe beim PBL. Das Lösen einer Problemaufgabe erfordert eine Input-Kompetenz, das Beiziehen und Anwenden von Konzepten, Fertigkeiten und Fähigkeiten sowie eine aufzuwendende Zeit. Der durch das Lösen der Problemaufgabe erzeugte individuelle Lernprozess (hier als Lernkurve dargestellt) führt zu einer bestimmten Output-Kompetenz.

Grösse	Erklärung
Input-Kompetenz	Anfangskompetenz, die zum Verständnis einer Aufgabe erforderlich ist und den Lernprozess durch PBL ermöglicht.
Output-Kompetenz	Kompetenz-Level, welches durch das Lösen der Problemaufgabe generiert wird.
Anzahl umgesetzter Konzepte, Fertigkeiten und Fähigkeiten	Konzepte, Fertigkeiten und Fähigkeiten, die zur Lösung der Problemaufgabe angewendet werden müssen.
Bearbeitungszeit	Zeit, die zur Lösung einer Problemaufgabe aufgewendet werden muss.

Tab. 5.1: Grössen einer Problemaufgabe beim kompetenzorientierten PBL.

2. Definitionsgrad von Problemaufgaben

Eine weitere Charakteristik einer Problemaufgabe ist ihr Definitionsgrad. Klar definierte (*well-defined*) Probleme haben im Gegensatz zu unklar definierten (*ill-defined*) ein bekanntes zu erreichendes Ziel [Aebli 1983]. Lernende sind sich aus ihrer Schullaufbahn gewohnt, vor allem mit klar definierten Problemen umzugehen. In der späteren Praxis hingegen sind sie mit vielen unklar definierten Problemen konfrontiert. Ein Grund, weshalb die Schule ungenügend auf die Anforderungen der Praxis vorbereitet, liegt darin, dass die Lernenden zu wenig mit unklar definierten Problemen konfrontiert werden. Bei den Studierenden sind unklar definierte Probleme nicht sehr beliebt, weil sie mit den Unsicherheiten über den einzuschlagenden Weg umgehen müssen. Unklar definierte Probleme setzen Vorwissen und Erfahrung voraus.

5.2.2 Typen von Problemaufgaben

Je nach Ziel, welches durch den Einsatz einer Problemaufgabe beim PBL verfolgt werden soll, können zwei Typen von Problemaufgaben mit unterschiedlicher Grösse und verschiedenem Definitionsgrad eingesetzt werden. Dabei wird unterschieden, ob ein Problem dazu dient, durch die Problembearbeitung neues Wissen anzueignen (*problembasierter Wissenserwerb*, Typ A) oder bereits gelerntes Wissen im realen Umfeld anzuwenden (*problembasierte Wissensanwendung*, Typ B) (Abb. 5.3). Beispiele zu den beiden Aufgabentypen sind im Kapitel 8 „Entwicklung und Umsetzung kompetenzorientierter Lerneinheiten“ zu finden.

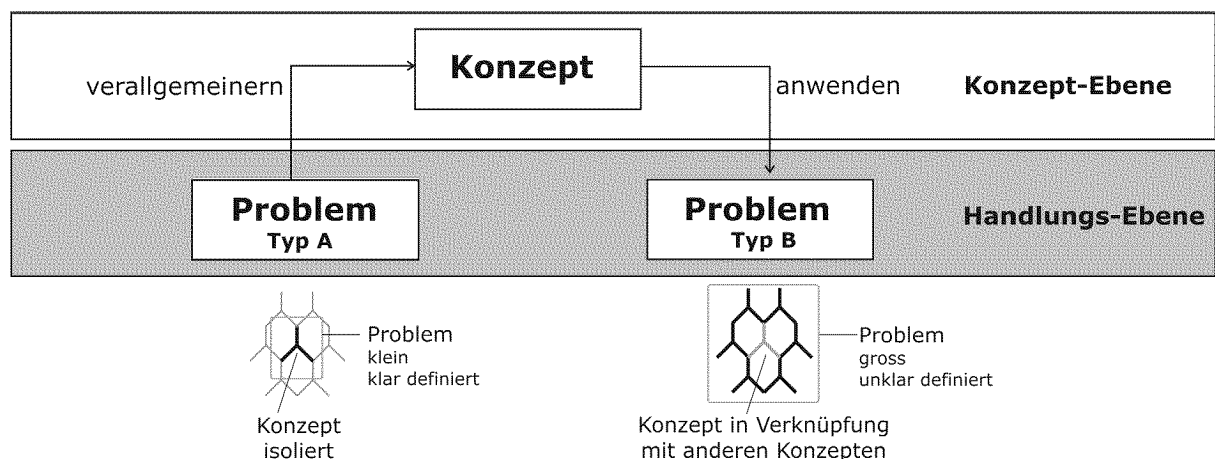


Abb. 5.3: Typen von Problemaufgaben beim PBL. Beim problembasierten Wissenserwerb (Typ A) werden einzelne Konzepte durch wiederholtes Üben verallgemeinert und auf diese Weise angeeignet und verinnerlicht. Die Konzepte werden bei diesem Aufgabentyp isoliert betrachtet. Bei der problembasierten Wissensanwendung (Typ B) werden Konzepte in einer realen Problemstellung angewendet. Die Konzepte werden bei diesem Aufgabentyp in Verknüpfungen mit anderen Konzepten betrachtet.

1. Problembasierter Wissenserwerb (Typ A)

Damit beim problembasierten Wissenserwerb aus einer Aufgabe die wesentlichen Merkmale extrahiert werden können, sollen kleine, einfache und klar definierte Probleme gelöst werden. Die Probleme sollen die zu lernenden Konzepte optimal repräsentieren. Novizen sind nur in der Lage, Konzepte aus einem Problem zu extrahieren, wenn die Problemaufgabe diese klar widerspiegelt. Das Problem muss hierfür auch eine klare innere Struktur aufweisen. Widersprüchliche oder zusammenhangslose Erfahrungen führen nicht zum Erlernen der Konzepte. Diese Art der Wissensaneignung wird auf die Fähigkeit des Gehirns zurückgeführt, aus Problemen allgemeine Regeln und Strukturen zu extrahieren (folglich Kapitel 2.3.3 „Unterschiede bei der Aneignung verschiedener Wissensarten“). Durch wiederholtes Anwenden in unterschiedlichen Kontexten werden die Konzepte weiter differenziert und flexibilisiert. Weil bei diesem Aufgabentyp die Konzepte isoliert betrachtet werden, ist das daraus generierte Wissen in seiner Anwendbarkeit begrenzt. Allzu kleine Aufgaben mit hohem Abstraktionsgrad und ohne Praxisbezug führen jedoch auch nicht zu einer Kompetenzaneignung, weil

damit keine Erfahrungen gemacht werden können. Der optimale Wirkungsgrad dieses Aufgabentyps liegt deshalb bei einem kleinen Aufgabenvolumen mit minimalem Praxisbezug, das die verlangte Input-Kompetenz möglichst tief hält (Abb. 5.4, Typ A).

2. Problembasierte Anwendung (Typ B)

Bei der problembasierten Anwendung werden Konzepte in einem praxisnahen Problem in jenem Kontext angewendet, in dem sie auch in der realen Welt vorkommen. Konzepte treten in Kombination mit anderen Konzepten und mit einem starken Bezug zur Praxis auf. Um diese Form der Problemaufgaben erfolgreich lösen zu können, müssen die Lernenden auf ein bereits vorhandenes Netz von Grundlagenwissen zurückgreifen können. Die minimale Input-Kompetenz ist demnach höher als beim ersten Aufgabentyp (Abb. 5.4, B). Konzepte werden durch die Arbeit mit dem Problem gefestigt und weiter flexibilisiert. Bei diesem Aufgabentyp ist das Volumen der Aufgabe durch den Praxisbezug grösser, er führt aber auch zu einer höheren Output-Kompetenz mit aktivem, anwendbarem Wissen. Überschreitet eine Aufgabe ein gewisses Volumen, werden die Lernenden durch die Anzahl der anzuwendenden Konzepte überfordert und der Wirkungsgrad dieses Aufgabentyps nimmt wieder ab. Der optimale Wirkungsgrad dieses Aufgabentyps liegt bei der Aufgabengrösse, die in der zur Verfügung stehenden Zeit bewältigt werden kann (Abb. 5.4, B).

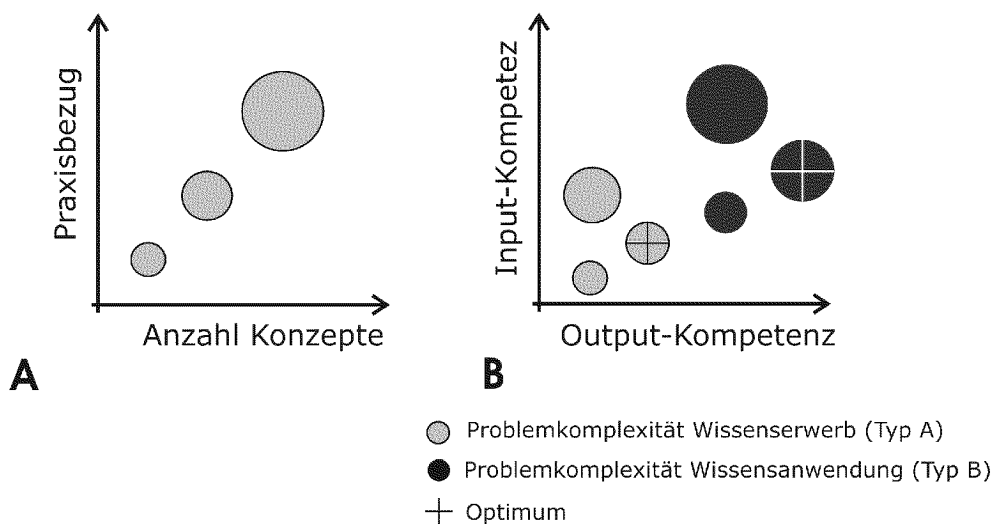


Abb. 5.4: Aufgabengrösse, Komplexität und Wirkungsgrad von Problemaufgaben beim PBL. A: Die Aufgabengrösse und Komplexität wächst mit zunehmendem Praxisbezug und der Anzahl der involvierten Konzepte. B: Wirkungsgrad von Problemaufgaben beim PBL. Typ A mit kleinem Aufgabenraum hat eine niedrigere Input-Kompetenz, minimalen Praxisbezug und enthält einzelne isolierte Konzepte. Typ B mit grossem Aufgabenraum erfordert eine höhere Input-Kompetenz, hat grösseren Praxisbezug und enthält mehrere Konzepte. Mit Aufgabentyp B kann eine höhere Output-Kompetenz erreicht werden als mit Typ A, er setzt aber Vorwissen voraus.

5.2.3 Optimieren der Output-Kompetenz durch eine Kombination von Aufgabentypen

Um eine höhere Output-Kompetenz mit aktivem, transferfähigem Wissen zu erreichen, die Lernenden aber durch die erhöhte Input-Kompetenz nicht zu überfordern, hat sich ein kombinierter Einsatz beider Aufgabentypen als sehr effizient erwiesen (Abb. 5.5). Zunächst werden in einem oder mehreren kleinen Problemen ausgewählte Konzepte durch kleine Aufgaben vom Typ A eingeübt. Dabei steht das Einüben einzelner isolierter Konzepte zusammen mit Fertigkeiten und Fähigkeiten im Vordergrund. In einem zweiten, umfangreicheren Problem steht das praxisbezogene Anwenden der Konzepte im Zentrum. Durch diesen Einsatz verschiedener Typen von Problemaufgaben werden die Vor- und Nachteile der beiden Aufgabentypen ideal genutzt (folglich Abb. 5.4 B).

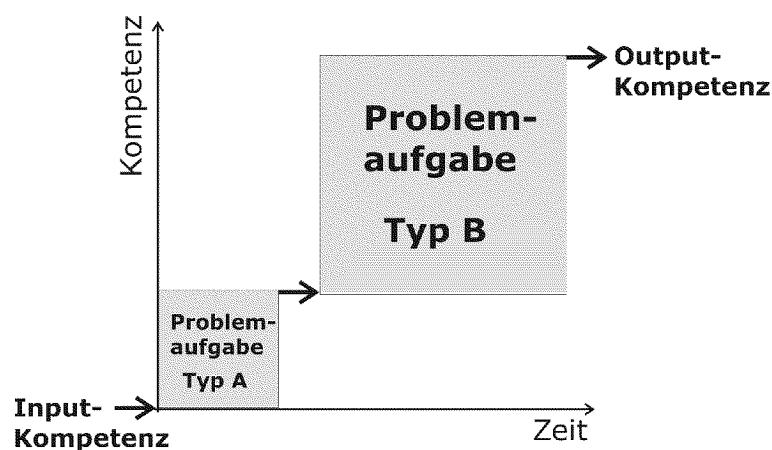


Abb. 5.5: Optimieren der Output-Kompetenz durch den kombinierten Einsatz zweier Aufgabentypen. Durch die Folge einer oder mehrerer kleiner konzeptbasierter Aufgaben vom Typ A und einer grossen Anwendungsaufgabe vom Typ B kann die Output-Kompetenz erhöht werden.

Das im Rahmen dieser Arbeit entwickelte didaktische Modell mit dem Namen *4-Schritte-Modell* setzt Problemaufgaben in dieser Abfolge ein, um in der zur Verfügung stehenden Zeit ein maximales Kompetenz-Level bei gleichzeitig niederschwelligem Zugang zu erzielen. Es wird im Rahmen des Kapitels 7 „Didaktische Modelle“ vorgestellt und im Kapitel 8 „Entwicklung und Umsetzung kompetenzorientierter Lerneinheiten“ an einem konkreten Beispiel erläutert.

5.2.4 Entwicklung problembasierter Kurse

Wie bereits im Kapitel 4 „Kompetenzorientierte Lernumgebungen“ erläutert, reicht es bei der Implementierung von Lernmethoden nicht aus, eine Reihe von zusammenhangslosen Problemaufgaben zu formulieren, abzugeben und lösen zu lassen, um aus einem Kurs einen problembasierten Kurs zu machen. PBL muss in eine Lernumgebung implementiert werden. Das bedeutet, dass die verschiedenen Aufgaben so aufeinander abgestimmt werden müssen, dass sie einen übergeordneten Lernprozess unterstützen und sich am Ziel der Kurskompetenz orientieren (Abb. 5.6). Diese Abstimmung der

Problemaufgaben wird vor allem durch die Auswahl und Konstruktion von neuen Aufgaben zusammen mit problembasierten Instruktionen erreicht (Details zur Umsetzung siehe Kapitel 8 „Entwicklung und Umsetzung kompetenzorientierter Lerneinheiten“).

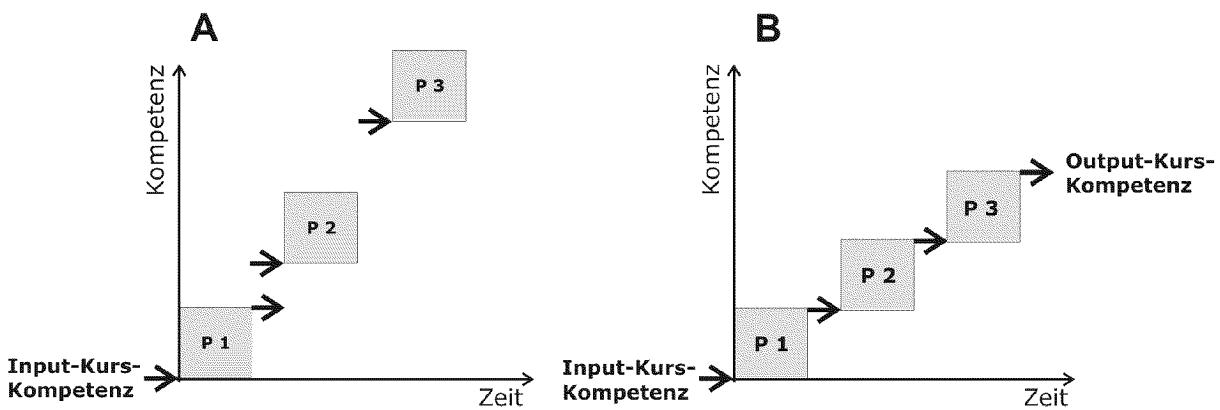


Abb. 5.6: Abstimmten der Kompetenzlevels einer Reihe von Problemaufgaben zu einer Kurs-Kompetenz. A: Sind die Input-Kompetenzen der Problemaufgaben nicht aufeinander abgestimmt, weil z.B. eine Folgeaufgabe eine viel höhere Input-Kompetenz erfordert, werden viele Lernende die Aufgaben nicht selbständig lösen und daher auch keine Kurskompetenz entwickeln können. B: Implementierung von PBL in einem Kurs führt dann zu einer brauchbaren übergeordneten Kurskompetenz, wenn die Input-Kompetenz der Problemaufgaben auf die Output-Kompetenz der vorangehenden Aufgabe abgestimmt ist.

Folgende zwei Punkte bilden die grosse Herausforderung bei der Implementierung von PBL:

1. Entwicklung übergeordneter Fähigkeiten und Kompetenzen durch eine Reihe von Problemaufgaben

Die Aneignung umfangreicherer Kompetenzen und Fähigkeiten (wie z.B. Programmieren) erfordert die Anwendung und Verknüpfung mehrerer miteinander in Beziehung stehender Konzepte, Fertigkeiten und Fähigkeiten (folglich Kapitel 3 „ICT-Kompetenzen für die wissenschaftliche Tätigkeit“). Sie muss durch eine Reihe von Problemaufgaben erreicht werden. Es stellt sich dabei die Frage, wie Problemaufgaben eingesetzt und miteinander in Beziehung gebracht werden können, damit auch übergeordnete Fähigkeiten und Kompetenzen entwickelt werden. Das zweite im Rahmen dieser Arbeit entwickelte didaktische Modell mit dem Namen *Anwendungsebenen-Modell* organisiert eine Reihe von kleineren Problemaufgaben für den Aufbau umfangreicherer Kompetenzen. Es wird in Kapitel 7 „Didaktische Modelle“ vorgestellt und in Kapitel 8 „Entwicklung und Umsetzung kompetenzorientierter Lerneinheiten“ an einem konkreten Beispiel erläutert.

2. Das Erreichen der Kurskompetenz für alle Lernenden ermöglichen

Die Implementierung von selbstgesteuertem PBL macht zwar die Entwicklung von Kompetenzen möglich, die Lernmethode erweist sich aber für die Lernenden als sehr

anspruchsvoll. Das kann dazu führen, dass Novizen beim Lösen von realen Problemen überfordert sind, weil ihnen das zur Lösung der Aufgabe notwendige Wissen grösstenteils fehlt. Ein gewisser Anteil der Lernenden gibt im Verlaufe eines Kurses auf, löst die Aufgabe nur oberflächlich oder kopiert sie von anderen Mitstudierenden.

Die Qualitätsansprüche einer Lernumgebung erfordern, dass mit einer Lernmethode ein möglichst grosser Anteil der Lernenden die geforderte Kurskompetenz zu einem bestimmten Grad unabhängig von Vorwissen und Arbeitstempo erreichen kann. Die Herausforderung besteht darin, PBL so einzusetzen, dass Novizen nicht überfordert und Personen, die bereits über Vorwissen verfügen, trotzdem gefordert sind.

Für den Lernprozess ist es entscheidend, dass die Lernenden die Aufgaben selber lösen, damit Erkenntnisse und Erfahrung im Lernprozess zum Tragen kommen können. Ob eine Problemaufgabe gelöst werden kann oder überhaupt angegangen wird, hängt aber nicht nur vom Vorwissen der Lernenden ab. Eine ganze Reihe motivationaler Faktoren beeinflusst den Lernprozess und entscheidet darüber, ob Lernende nach dem Lösen eines Problems weitere Probleme lösen und welche Einstellung sie dabei gegenüber dem Lerninhalt entwickeln.

Wir wollen uns deshalb im nächsten Abschnitt mit einigen motivationalen Aspekten des Problemlösens beschäftigen.

5.3. Motivationale und emotionale Faktoren des PBL

Die Problemaufgaben eines problembasierten Kurses helfen, einen Lernprozess in Gang zu bringen, indem die Lernenden Konzepte, Fertigkeiten und Fähigkeiten zur Lösung des Problems anwenden müssen. Ob Problemaufgaben von den Lernenden überhaupt angegangen werden, hängt davon ab, welche Erfahrungen die Lernenden in der Vergangenheit beim Lösen von ähnlichen Problemen gemacht haben, welche Rolle sie bei der Kompetenzerwerb einnehmen und wie sie sich dabei selber einschätzen. Eine Lernumgebung mit PBL kann nur lernendengesteuert werden, wenn sich die Lernenden als fähige, kompetent handelnde Individuen sehen und im Unterricht Möglichkeiten erhalten, dieses Bild von sich im Zusammenhang mit dem Lerninhalt zu entwickeln.

Motivationale und emotionale Faktoren sowie die Überzeugung, ein Problem selber meistern zu können, sind für die Implementierung von selbstgesteuertem PBL wichtige Grössen. Sie wirken sich nicht nur nachhaltig auf den laufenden Lernprozess aus, sie beeinflussen auch die Bedingungen, unter denen die nachfolgenden Probleme gelöst werden. Lösen die Lernenden die Probleme gerne, werden sie sich länger und intensiver mit dem Lerninhalt auseinandersetzen. Umgekehrt werden Lernende, die die Aufgaben weniger gerne lösen, sich oberflächlicher oder gar nicht mit dem Inhalt beschäftigen.

Ein positiver Verlauf dieser motivationalen Grössen wirkt weit über einen Einführungskurs hinaus. Sie beeinflussen das Bild, das die Lernenden von einem Fachbereich oder einem Lerninhalt haben. Lernende, die sich im Zusammenhang mit einem Fach oder einem Lerninhalt als fähig und kompetent einschätzen, werden sich auch nach einem Kurs vermehrt mit diesen Lerninhalten auseinandersetzen und weiterbilden. Auf diese Weise kann eine Basis für ein lebenslanges Lernen in einem Fachbereich gesetzt werden. Natürlich ist die Auswahl einer für eine Zielgruppe interessanten Aufgabe wichtig und für Lernende motivierend. Der für die Motivation wesentlichste Prozess beim PBL aber ist, dass die Lernenden durch das Lösen von Aufgaben Freude am eigenen Fortschritt und Lernprozess erhalten.

Zusammenfassend halten wir fest, dass folgende motivationalen und emotionalen Grössen für die Implementierung von PBL von Interesse sind:

- Erfolgserlebnisse beim Lösen von Problemen als positive Verstärker für den Lernprozess (siehe Abschnitt 5.3.1)
- Zufriedenheit, Kontrollüberzeugung und Selbstwirksamkeit (siehe Abschnitt 5.3.2)

5.3.1 Erfolgserlebnisse als positive Verstärker für den Lernprozess

Wie in Abb. 5.2 dargestellt, stellt jedes Problem eine bestimmte Anforderung an die Anfangs- oder Inputkompetenz der Lernenden. Ob eine Problemaufgabe selbständig lösbar ist und daraus gelernt werden kann, ist davon abhängig, ob die Anfangskompetenz der Lernenden den Anforderungen der Problemaufgabe entspricht (Abb. 5.7). Genügt die momentane Kompetenz der Lernenden den Anforderungen der Aufgabe, wird sie von den Lernenden selbständig gelöst werden können, was zu Erfolgserlebnissen, Gefühlen der Befriedigung und zum Aufbau von Selbstvertrauen in Bezug auf die zu lösende Aufgabe führt. Diese Bedingungen sind für das Lernen ideal und die Lernenden erweitern ihr individuelles Wissen und Können. Die Anforderungen einer nächsten Aufgabe können erhöht werden. Ein „virtuoser Kreis“ des Lernens kommt in Gang, der das weitere Lernen positiv verstärkt (Abb. 5.7, A).

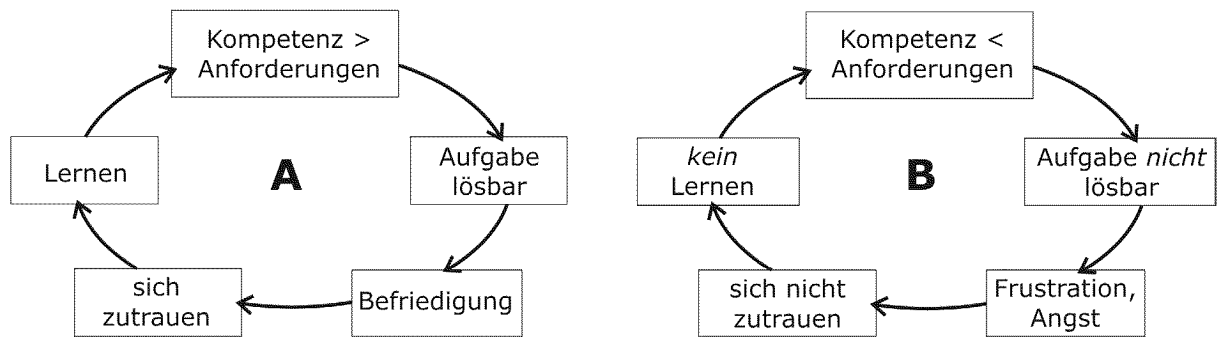


Abb. 5.7: Virtuoser Kreislauf (A) und Teufelskreis (B) beim PBL. Entsprechen die Anforderungen an eine Aufgabe der momentanen Kompetenz der Lernenden, wird das Lernen eines Lerninhalts mit Gefühlen der Zufriedenheit und Selbstvertrauen verbunden. Entsprechen umgekehrt die Anforderungen einer Problemaufgabe nicht der momentanen Kompetenz der Lernenden, wird das Lernen eines Lerninhalts mit Frustration und Angst verbunden, was ein weiteres Lernen verhindert [abgeändert nach Butterworth 1999].

Genügt umgekehrt die momentane Kompetenz der Lernenden den Anforderungen der Problemaufgabe nicht, wird die Aufgabe nicht selbstständig gelöst werden können. Frustration und ein Gefühl, den Anforderungen der Aufgaben nicht gewachsen zu sein, verhindern ein Lernen am Problem. Weil die Lernenden in einem nächsten Problem mit höheren Anforderungen konfrontiert sind, kommt ein „Teufelskreis“ in Gang, der das Selbstvertrauen schwächt und durch Gefühle der Frustration und Angst dominiert wird (Abb. 5.7, B). Die eigentlichen Vorteile des PBL bleiben aus. Haben sich die motivationalen Größen einmal in diese Richtung entwickelt, ist es schwierig, sie wieder in die andere Richtung zu lenken.

Wie aus Abb. 5.7 ersichtlich wird, garantieren Problemaufgaben beim PBL nicht, dass damit auch gelernt wird. Je nach Anforderungen der gestellten Aufgabe können beide Kreisläufe bei einem bestimmten Anteil der Lernenden in Gang gebracht werden. Die Herausforderung bei der Implementierung von PBL in eine Lernumgebung besteht darin, dass bei einem möglichst grossen Teil der Lernenden der „virtuose Kreislauf“ des Lernens in Gang gebracht wird, um beim Lösen von Problemaufgaben einen möglichst grossen Nutzen zu erzielen.

Damit PBL nicht mit Frustration und Angst verbunden wird, sondern Erfolgserlebnisse Selbstvertrauen und Zufriedenheit auslöst, ist es wichtig, dass die Anforderungen zum Lösen einer Aufgabe auf die momentane Kompetenz der Lernenden Rücksicht nimmt. Um in der zur Verfügung stehenden Zeit trotzdem ein maximales Kompetenzniveau zu erreichen, muss nach Lösungen gesucht werden, die Lernenden auf die Anforderungen komplexerer Problemaufgaben vorzubereiten (siehe Abschnitt 5.4 und Kapitel 6 „Selbstgesteuerte, problembasierte Instruktion“).

5.3.2 Motivationale Grössen beim Lösen von Problemen

Wie im letzten Abschnitt gezeigt, verändern sich beim Lösen von Problemaufgaben eine ganze Reihe psychologischer Grössen abhängig von den Anforderungen, die die Problemaufgaben an die Lernenden stellen. Je höher der Komplexitätsgrad der zu erreichenden Kompetenz ist, desto mehr Problemaufgaben müssen in einem PBL-Kurs eingesetzt werden. Soll gleichzeitig der Anteil der Selbststeuerung der Lernenden im Kurs hoch sein, spielen diese motivationalen Grössen für den individuellen Lernerfolg und die Qualität der problembasierten Lernumgebung eine entscheidende Rolle. Welche psychologischen Konstrukte sind dabei von Interesse?

Tab. 5.2 zeigt, wie sich die Grössen der psychologischen Konstrukte *Zufriedenheit*, *Kontrollüberzeugung* und *Selbstwirksamkeitserwartung* in Abhängigkeit von den gestellten Anforderungen und der Kompetenz entwickeln können. Im Folgenden sollen die einzelnen Grössen im Bezug auf das PBL betrachtet werden.

	Kompetenz > Anforderung	Kompetenz < Anforderung
Zufriedenheit	zufrieden	unzufrieden
Kontrollüberzeugung	internal/interaktionistisch	external/fatalistisch
Selbstwirksamkeitserwartung	positiv	negativ

Tab. 5.2: Motivationale Grössen beim PBL: Veränderungen einiger Grössen psychologischer Konstrukte in Abhängigkeit von der Kompetenz der Lernenden bezüglich der Anforderungen der Aufgabe.

1. Zufriedenheit

Zufriedenheit (*Satisfaction*) ist ein emotionaler Zustand, der sich aus einem Vergleich zwischen den Erwartungen und dem Grad der Erfüllung der Zielvorgaben ergibt [Bruggemann 1974]. Beim PBL entspricht das Soll den Anforderungen zur Lösung einer Problemaufgabe. Je nach dem, ob die Lernenden die Zielvorgaben der Problemstellung erreichen können und wie ihre Erwartungen dabei sind, entwickelt sich ein Gefühl von Zufriedenheit, respektive Unzufriedenheit. Bruggemann unterscheidet in einem Modell zur Arbeitszufriedenheit verschiedene Formen der empfundenen Zufriedenheit, respektive Unzufriedenheit (Tab. 5.3). Die einzelnen Formen der Zufriedenheit entwickeln sich abhängig davon, ob die vorgegebene Zielsetzung im Verhältnis zu den eigenen Erwartungen erfüllt werden kann. Je nach Form der Zufriedenheit kommt es bei den Lernenden zu einer Anpassung der Erwartungen, und es entscheidet sich, ob weitere Problemlöseversuche unternommen werden oder nicht.

Bei einer „stabilisierten Zufriedenheit“ kann die Zielsetzung erreicht werden, das Anspruchsniveau wird beibehalten und es werden neue Problemlöseversuche unternommen (Tab. 5.3). Entwickelt sich eine „progressive Zufriedenheit“, dann wollen sich die Lernenden im nächsten Problem steigern können, weil sie das gesetzte Ziel noch nicht erreicht haben, aber davon überzeugt sind, es mit Hilfe der nächsten Aufgaben

erreichen zu können. Ihre Zielvorstellungen und Erwartungen werden erhöht, was sich auf die Bedingungen beim Lösen des nächsten Problems auswirkt. Die in Tab. 5.3 erwähnte „Pseudo-Zufriedenheit“ kommt auf Grund von Fehleinschätzungen beim Erreichen der Zielsetzung zu Stande. Es wird eine Zufriedenheit entwickelt, obwohl die Zielsetzung nicht erreicht wurde.

Zufriedenheit beim Lösen von Problemen ist Voraussetzung dafür, dass in Zukunft neue Probleme angegangen werden. Dabei ist es wichtig, dass die Lernenden genaue Zielsetzungen erhalten und richtig und realistisch einschätzen können, ob sie diese Zielsetzungen erreicht haben.

	Form	Zielsetzung erreicht?	Erwartungen der Lernenden	Neue Problemlöse-versuche?
Zufriedenheit	progressive Zufriedenheit	nicht erreicht	erhöht	ja
	stabilisierte Zufriedenheit	erreicht	beibehalten	
Unzufriedenheit	pseudo-Zufriedenheit	nicht erreicht		nein
	konstruktive Unzufriedenheit			
	fixierte Unzufriedenheit			
	resignative Unzufriedenheit		gesenkt	ja

Tab. 5.3: Formen der Arbeitszufriedenheit [abgeändert nach Bruggemann 1974].

2. Kontrollüberzeugung

Mit Kontrollüberzeugung oder -bewusstsein (*Locus of control*) ist die Überzeugung eines Individuums gemeint, dass ein Zusammenhang zwischen Handlungen oder Vorkommnissen und darauf folgenden Ereignissen besteht [Rotter 1966]. Sie basiert auf der Annahme, dass Menschen Ereignisse in Zusammenhängen interpretieren und nach Gründen suchen, die den Lauf der Geschehnisse erklären. An diesem psychologischen Konstrukt ist nicht entscheidend, ob tatsächlich ein Zusammenhang zwischen einer Handlung und einem Ereignis besteht, sondern die persönliche Sicht der Dinge.

Es können verschiedene Dimensionen der Kontrollüberzeugung unterschieden werden (Tab. 5.4). Die wichtigsten beiden Dimensionen sind die Unterscheidung zwischen internaler und externaler Kontrollüberzeugung [Rotter 1990]. Bei der internalen Kontrollüberzeugung werden die Gründe für das eigene Verhalten bei der eigenen Per-

son gesucht, bei der externalen Kontrollüberzeugung werden äussere Einflussfaktoren dafür verantwortlich gemacht. Interaktionistisch meint, dass Ereignisse durch eine Interaktion von Person- und Situationsfaktoren bestimmt sind [Hoff 1992].

Dimensionen der Kontrollüberzeugung	Erklärung
internal	Eigene Wahl, Bedürfnisse, Selbstwirksamkeit, Eigenaktivität, Unabhängigkeit
interaktionistisch	Gelegenheiten wahrnehmen, Entscheidungen oder Handeln von anderen beeinflussen, Umwelt und Person beeinflussen sich reziprok, eigene und fremde Einflüsse wirken unabhängig, an Gegebenheiten anpassen
external	Starker Einfluss anderer Personen, starker Einfluss der Situation, keine eigene Kontrolle
fatalistisch	Zufall, Glück

Tab. 5.4: Dimensionen der Kontrollüberzeugung.

Die Kontrollüberzeugung ist für die Implementierung von PBL deshalb wichtig, weil bei Lernenden, die in einem problembasierten Kurs davon überzeugt sind, dass sie mit den zur Verfügung gestellten Mitteln den Lernprozess selber in die Hand nehmen und mitgestalten können, der Lernerfolg höher sein wird, als wenn sie das Gefühl haben, unter starkem Einfluss anderer Personen zu stehen.

Aufgrund der starken Handlungsorientierung des PBL müssen Lerninhalte in Handlungen umgesetzt und als Kompetenz sichtbar gemacht werden. Durch diese Verknüpfung der Lerninhalte mit dem eigenen Handeln kann eine interne Kontrollüberzeugung aufgebaut werden.

Sind die zu lösenden Probleme zu schwierig, so dass sie nicht selbständig genug gelöst werden können, sind die Lernenden auf Hilfe angewiesen. Ist die Abhängigkeit von anderen Personen (Lehrpersonen, Mitstudierende, Freunde etc.) beim Lösen der Probleme gross, kann sich eine Kontrollüberzeugung entwickeln, die unter starkem Einfluss anderer Personen steht (externale Kontrollüberzeugung).

Das Nebeneinander von Freiräumen und Einschränkungen im Problemlöseprozess sowie gegebene Abhängigkeiten zwischen Lehrenden und Lernenden führen in der Realität oft zu einer interaktionistischen Kontrollüberzeugung, die den Lernenden ermöglicht, differenzierter zwischen eigenen und fremden Einflüssen zu unterscheiden.

3. Selbstwirksamkeitserwartung

Ob eine Problemaufgabe gelöst werden kann, hängt nicht nur davon ab, ob das nötige Wissen zur Lösung des Problems mobilisiert werden kann, sondern auch von der Überzeugung, ein Problem lösen zu können. Lernende, die überzeugt sind, selbst etwas be-

wirken zu können, die ihren Fähigkeiten vertrauen und Probleme durch das eigene und selbstständige Handeln beeinflussen können, haben eine hohe Selbstwirksamkeitserwartung (*Self efficacy*) [Bandura 1997]. Die Selbstwirksamkeit muss nichts damit zu tun haben, was die Personen wirklich zu leisten in der Lage sind. Trotzdem steuert sie das Handeln der Lernenden stark. Die Selbstwirksamkeitserwartung ist mit Erfahrungen beim Lösen von ähnlichen Problemaufgaben verknüpft. Hat man in der Vergangenheit bei bestimmten Aufgaben Erfolgserlebnisse erzielt, wird das Vertrauen gestärkt, bei ähnlichen Anforderungssituationen auch erfolgreich zu sein. Umgekehrt gilt auch, dass die wahrgenommene Selbstwirksamkeit eines Menschen sinkt, wenn mit einer bestimmten Aufgabe Misserfolge erzielt werden. Bei einer negativen Selbstwirksamkeitserwartung kann es vorkommen, dass eine Aufgabe nicht gelöst werden kann, nicht weil die Anforderungen zu hoch sind, sondern weil die Lernenden sich aufgrund früherer Erfahrungen nicht zutrauen, sie erfolgreich zu bewältigen. Eine positive Selbstwirksamkeitserwartung ist Voraussetzung dafür, dass in einer Folge von Problemen mit zunehmender Komplexität die Aufgaben überhaupt angegangen und selbstständig gelöst werden. Sie wird dann erreicht, wenn die Erwartungen der Lernenden positiv beeinflusst werden können.

Zusammenfassend gesagt wird beim selbstgesteuerten PBL dann am meisten gelernt, wenn die Lernenden beim Lösen der Probleme die Anforderungen einer Aufgabe erfüllen, Zufriedenheit entwickeln, den Grund für Fortschritte im Lernprozess bei ihrem persönlichen Einsatz sehen und sich durch die Erfolgserlebnisse zutrauen, weitere Probleme erfolgreich zu meistern.

5.3.3 Berücksichtigung motivationaler Größen bei der Implementierung von PBL

Um mit PBL eine maximale Wirkung zu erzielen, müssen sich eine ganze Reihe motivationaler Größen diverser psychologischer Konstrukte in eine für das Lernen günstige Richtung entwickeln (folglich letzter Abschnitt). Die Implementierung von PBL würde aber kaum erfolgreich verlaufen, würde jedes psychologische Konstrukt einzeln erfasst und umgesetzt werden. Es stellt sich daher die Frage, wie eine erfolgreiche Implementierung von PBL erreicht und mit welchen Qualitätskriterien diese überprüft werden kann?

1. Vorgehen für eine erfolgreiche Implementierung von PBL

Aus der Sicht der Entwicklerinnen und Entwickler eines problembasierten Kurses geschieht die Implementierung von PBL unter Berücksichtigung der psychologischen Größen über die Aufbereitung des Lerninhalts. Dies wird einerseits über die Auswahl und Konstruktion anspruchsvoller Problemaufgaben umgesetzt und andererseits über die ergonomische Entwicklung von Lernprozess-unterstützenden Systemstrukturen der Lernumgebung im Zusammenhang mit einem didaktischen Modell.

Folgendes Vorgehen hat sich bei für die Entwicklung eines problembasierten Kurses bewährt:

- 1. Auswahl, beziehungsweise Konstruktion von anspruchsvollen Aufgaben mit einem Bezug zur Praxis, bei welchen ausgewählte Konzepte angewendet werden müssen.
- 2. Entwicklung von Lernmaterialien mit dem Ziel, dass die Aufgabe aus Punkt 1 für einen möglichst grossen Teil der Lernenden selbständig lösbar wird und zu deren Zufriedenheit und Motivation führt.
- 3. Wiederholter produktiver Einsatz der Lernmaterialien im Unterricht, damit sie effizient und zielgerichtet verbessert werden können.

Die im Rahmen dieser Arbeit entwickelten didaktischen Modelle werden im Kapitel 7 vorgestellt und die Konstruktionsprozesse zu deren Implementierung von PBL ist Inhalt des Kapitels 8.

2. Qualitätskriterien zur Überprüfung einer Implementierung von PBL

Für die Entwicklung und Qualitätskontrolle eines problembasierten Kurses sind Grössen zu erfassen, die etwas über die Qualität der Implementierung von PBL zu unterschiedlichen Zeitpunkten des Kurses aussagen. Folgende Grössen sind dabei von Interesse und relativ einfach zu erfassen: Die von den Lernenden für eine Problemaufgabe investierte Arbeitszeit, der von ihnen eingeschätzte Schwierigkeitsgrad der Aufgabe und ihre Motivation oder Zufriedenheit beim Lösen der Aufgabe (Tab. 5.5).

Grösse	Mögliche Frage
Investierte Arbeitszeit	Wie lange haben Sie an dieser Problemaufgabe gearbeitet?
Eingeschätzter Schwierigkeitsgrad	Wie stufen Sie den Schwierigkeitsgrad dieser Problemaufgabe ein?
Motivation / Zufriedenheit	Wie gerne haben Sie dieses Problem gelöst?

Tab. 5.5: Grössen zur Erfassung der Implementierungsqualität einer Problemaufgabe beim PBL [Bauer-Messmer 2005, Faessler 2006].

Die erfassten Werte liefern einerseits Information über die Wirkung einer einzelnen Problemaufgabe. Andererseits besteht die Möglichkeit, die Werte einer Problemaufgabe in Bezug zu den Werten anderer Problemaufgaben zu stellen, um die Effektivität von Problemaufgaben in einem problembasierten Kurs zu erfassen. Für diesen Fall eignet sich die Durchführung einer *Prozessanalyse*, bei der für die erfassten Werte an Stelle von Absolutgrössen Veränderungen über die Zeit im Zentrum stehen [Fisseni 2004]. Die Grössen aus Tab. 5.5 sind für die Durchführung einer Prozessanalyse jeweils in Bezug zum vorgängigen Problem zu stellen.

Die Resultate der im Rahmen dieser Arbeit ausgeführten Prozessanalyse sind dem Kapitel 10 „Evaluation der Lernumgebungen“ zu entnehmen.

5.4. Fazit: Effektives und motivierendes PBL

Die grosse Stärke von PBL ist seine Transfer-Wirksamkeit. Zu diesem Zweck müssen die gewählten Probleme einen starken Bezug zur Praxis haben. Dies geht mit dem Konzept des *Inverted Curriculums* einher, bei dem sich die Lernenden Kompetenzen aneignen, indem sie komplexe Probleme von aussen nach innen betrachten und schrittweise versuchen, dieses zu verstehen und zu modifizieren [Meyer 1993]. Solche Aufgaben sind meist umfangreich und setzen voraus, dass die Lernenden sich bereits spezifische Konzepte, Fertigkeiten und Fähigkeiten angeeignet haben und darüber verfügen können (Abb. 5.4). PBL ist nur dann effektiv und motivierend, wenn es zu Zufriedenheit und Selbstvertrauen beim Lösen der Probleme führt. Dies ist dann der Fall, wenn die Lernenden durch eine Problemaufgabe gefordert sind und die Aufgabe selbständig und unabhängig bewältigen können (Abb. 5.7). Damit alle Studierenden unabhängig vom Vorwissen die Möglichkeit erhalten, diese praxisnahen Probleme selbständig zu lösen, ist die *Instruktion* von ausgewählten Konzepten, Fertigkeiten und Fähigkeiten unerlässlich. Diese Instruktion ist mit Vorteil ebenfalls problembasiert, weil dadurch Konzepte und Fertigkeiten nicht voneinander getrennt werden und aktives, anwendbares Wissen angeeignet werden kann (Abb. 5.8).

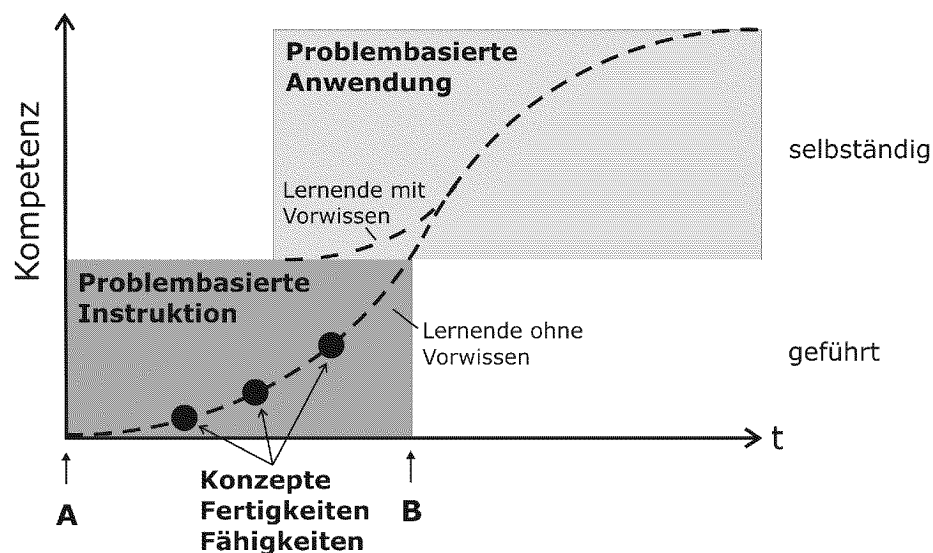


Abb. 5.8: Problembasierte Instruktion und Anwendung als effektive und motivierende Form des PBL: Konzepte, Fertigkeiten und Fähigkeiten, die zur Lösung einer praxisnahen, selbständig zu lösenden Problemaufgabe angeeignet werden müssen, werden zunächst unter Anleitung in einer problembasierten Instruktion eingeübt. Lernende, die bereits über Vorwissen verfügen, können auch erst beim selbständigen Teil (Zeitpunkt B) beginnen.

Durch diese Anordnung wird PBL nicht nur effektiver, es kann dadurch auch ein grösserer Anteil der Studierenden unterstützt und auf das für die Prüfung verbindliche

Kompetenzniveau gebracht werden. Lernende mit ungenügendem Vorwissen beginnen bei der problembasierten Instruktion (Zeitpunkt A). Lernende, die bereits über Vorwissen verfügen, können bei der problembasierten Anwendung einsteigen und ihr Wissen durch die selbständige Lösung der Aufgabe unter Beweis stellen. Realisieren diese Studierenden, dass sie beim Lösen der Aufgabe doch Wissenslücken haben, können sie jederzeit auf die problembasierte Instruktion oder Teile davon zurückgreifen.

Das im Rahmen dieser Arbeit entwickelte *4-Schritte-Modell* setzt Problemaufgaben in dieser Abfolge ein. Es wird im Kapitel 7 „Didaktische Modelle“ vorgestellt. Im nächsten Kapitel gehen wir der Frage nach, welche Aufgaben und Funktionen eine solche problembasierte Instruktion erfüllen muss.

Kapitel 6

Selbstgesteuerte, problembasierte Instruktion

6.1. Einleitung

Im Zusammenhang mit PBL ist es naheliegend, die Instruktion ebenfalls problembasiert auszurichten, um erstens den Lernprozess in Gang zu bringen und zweitens die Lernenden auf das Lösen von anspruchsvollen Aufgaben mit Realitätsbezug vorzubereiten (folglich Kapitel 5.4 „Effektives und motivierendes PBL“, Abb. 5.8). Das Lösen von Problemaufgaben soll zu Erfolgserlebnissen führen und dafür sorgen, dass mehr gelernt und ein höheres Kompetenzniveau erreicht wird. Orientiert sich die Instruktion ebenfalls an Problemen, die von den Lernenden unter Anleitung selber gelöst werden, so wird aktives, anwendbares Wissen aufgebaut, das beim anschließenden Lösen einer selbständigen Aufgabe eingesetzt werden kann (folglich Kapitel 2 „Kompetenzorientierter Unterricht“).

Aufgaben einer problembasierten Instruktion

Der problembasierten Instruktion fallen beim Aufbau der Kompetenzen vor allem folgende Aufgaben und Funktionen zu:

- Den individuellen Lernprozess starten, Sicherheit und Selbständigkeit verleihen.
- Wichtige Begriffe und ein Wissensnetz aus Konzepten, Fertigkeiten und Fähigkeiten aufbauen.
- Durch die Möglichkeit der Selbststeuerung soll die Instruktion unabhängig von Ort, Zeit und Lehrpersonen im eigenen Tempo durchgeführt werden können.
- Durch die Handlungsorientierung kann die Instruktion nachhaltig intrinsisch motivieren und wirkt dadurch nicht belehrend.

6.2. Anleitungen als Instruktionsmedium

Bei der problembasierten Instruktion dient die Problemaufgabe einerseits als Anschauungsobjekt für die Aneignung von Konzepten und Fertigkeiten und andererseits als

Mittel zur Strukturierung der zu vermittelnden Information. Im Zentrum der Entwicklung von Instruktionmaterialien für den problembasierten Wissenserwerb stehen Anleitungen (*Tutorials*), mit deren Hilfe die Lernenden die Lösung eines Problems im eigenen Tempo nachkonstruieren können. Die Instruktion hat dabei die Form eines geleiteten Weges durch die reale und komplexe Welt einer Problemstellung, mit dem Ziel, auf eine grössere, selbständige Arbeit vorzubereiten. Dieser Weg soll aber nicht bestritten werden, indem ein Experte ihn erklärt oder beschreibt, sondern indem er durch die Lernenden selber im eignen Tempo zurückgelegt wird, die mit Hilfe der Anleitung den Gedankengängen folgen und Handlungen ausführen. Die Lernenden lösen die Aufgabe wie sie ein Experte lösen würde. Lernen durch Nachahmen kann sehr effektiv sein, wenn dabei die Aneignung eigener Kompetenzen im Zentrum steht und die Selbständigkeit im Verlaufe der Zeit stetig erhöht wird (folglich Kapitel 4.5.3 „Kognitive Apprenticeship-Modell“).

Durch die Anleitung werden die zur Lösung des gestellten Problems erforderlichen Konzepte und Fertigkeiten nicht nur bereitgestellt, sondern in die Reihenfolge eines sinnvollen Handlungsablaufs gestellt. Konzepte können dabei als *Bauteile (Was)* und Fertigkeiten als das *Wissen über die Bedienung von Werkzeugen (Wie)* zur Lösung eines Problems betrachtet werden (Abb. 6.1).

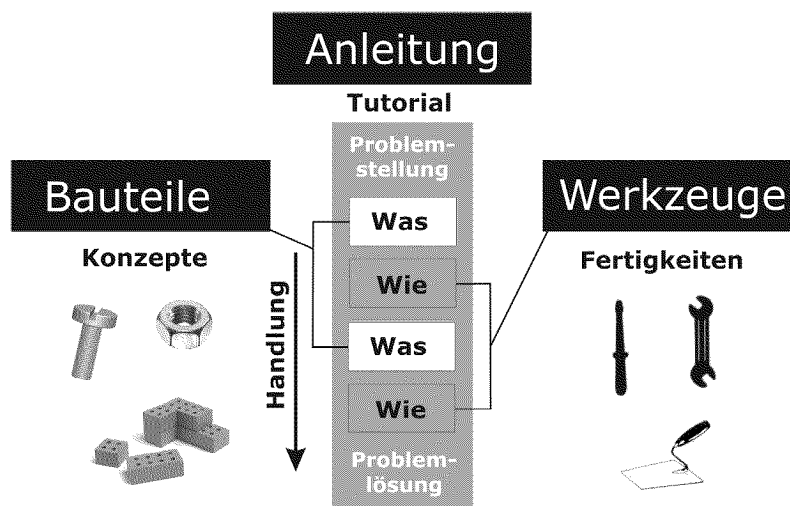


Abb. 6.1: Anleitung als Mittel zur Strukturierung bei der Vermittlung von Informationen für unterschiedliche Wissensarten. Bei der Anleitung (*Tutorial*) können Konzepte als Bauteile (*Was*) und Fertigkeiten als Wissen über die Bedienung von Werkzeugen (*Wie*) betrachtet werden. In Kombination führen sie zu einer Handlung und können zur Lösung von Problemen eingesetzt werden.

Indem die Lernenden mit den zu vermittelnden Konzepten und Fertigkeiten eigene Handlungen ausführen und ein eigenes Produkt kreieren, beziehen sie die damit verbundene Information auf sich. Die Instruktion der Konzepte wird dadurch nicht als fremdbestimmt, sondern als persönliche Kompetenzaneignung gewertet (folglich Kapitel 5.3.2 „internale Kontrollüberzeugung“). Das Resultat ist der Aufbau eines Wissensnetzes als direkte Verknüpfung von deklarativem und prozeduralem Wissen in

Form *kognitiver Werkzeuge*, das in der späteren Anwendung als aktives Wissensnetz zur Verfügung steht (siehe „konstruktive Begriffsbildung“ Abschnitt 6.3.2).

Texte, die Konzepte und Fertigkeiten als praktische Anwendung beschreiben, sind oft sehr umfangreich. Hier seien als Beispiel die häufig sehr umfangreichen Bücher der angewandten Informatik erwähnt. Es besteht dabei die Gefahr, dass die Lernenden wegen Übermass an Information die Konzepte nicht mehr erkennen können. Zudem erzeugt das Lesen über Fertigkeiten und Fähigkeiten kein Wissen dieser Dimensionen. Durch die Form der Anleitung als Instruktionsmedium die dadurch ausgeführten Handlungen entstehen echte Fertigkeiten und Fähigkeiten, die nur durch eigenes Handeln entstehen können (folglich Kapitel 2 „Kompetenzorientierter Unterricht“). Die Herausforderung bei der Entwicklung von handlungsbasierten Anleitungen besteht darin, dass einerseits ein stetiger Handlungsfluss entstehen und andererseits Wissen unterschiedlicher Dimensionen erzeugt und gesichert werden soll.

Mit dem Computer stehen mächtige Werkzeuge zur Lösung von Problemen vieler Disziplinen zur Verfügung. Je komplexer die Probleme, desto mehr ICT-Konzepte müssen zur Anwendung gebracht werden (folglich Kapitel 3 „ICT-Kompetenzen für die wissenschaftliche Tätigkeit“). Da die Bedienung dieser Computerwerkzeuge am Computer stattfindet, ist es naheliegend, diese Anleitungen mit den (ICT-)Konzepten und den Informationen zur Bedienung der Werkzeuge als elektronische Anleitungen (*E.Tutorials*[®], siehe Abschnitt 6.5.1) möglichst nahe zueinander auf denselben Bildschirm zu bringen.

6.3. Aufbau eines aktiven Wissensnetzes

Die beiden wichtigen Aufgaben einer Instruktion sind der Aufbau eines aktiven Wissensnetzes (Abschnitt 6.3.1) und die Einführung von Begriffen (Abschnitt 6.3.2). Soll angeeignetes Wissen später zur Lösung von Problemen angewendet werden können, muss bei der Instruktion eine Form gewählt werden, bei der das Wissen nicht träge bleibt (folglich Kapitel 2.1 „Träges Wissen“) und das Wissen über Begriffe - auch nach dem Kurs - stetig erweitert werden kann.

6.3.1 Eine Instruktion für verschiedene Wissensdimensionen

Im traditionellen Unterricht werden die unterschiedlichen Wissensdimensionen meistens sequenziert und in verschiedenen dafür spezialisierten Unterrichtsgefäßen (Vorlesung, Übung, Anwendung) vermittelt (Abb. 6.2, Sequenzierte Instruktion). Die Instruktion beginnt meistens mit der mündlichen oder schriftlichen Präsentation der Konzepte. Dieses Vorgehen hat den Vorteil, dass man sich auf die wesentlichen allgemeinen Prinzipien, die „Theorie“, konzentrieren kann, ohne sich gleichzeitig in den praxisrelevanten Details zu verlieren. Bloss auf theoretischer Ebene angeeignetes Wissen bleibt aber oft beim späteren Anwenden träge, weil es bei der Aneignung zu wenig

mit prozeduralem Wissen verknüpft werden konnte (folglich Kapitel 2 „Kompetenzorientierter Unterricht“). In nachfolgenden Übungsstunden werden zusätzlich Fertigkeiten trainiert. Es zeigt sich aber, dass die von Konzepten isolierte Aneignung von Fertigkeiten ebenfalls nicht zu aktivem Wissen führt, weil die Lernenden nicht in der Lage sind, das Wissen unterschiedlicher Dimensionen miteinander zu verknüpfen. Die Folge der sequenzierten Vermittlung von Konzepten und Fertigkeiten ist, dass die beiden Wissensdimensionen nicht als Teil der selben Kompetenz wahrgenommen werden und der Kompetenzaufbau daher häufig ausbleibt.

Eine Verknüpfung von Konzepten und Fertigkeiten lässt sich erreichen, indem beide gleichzeitig in einem Beispiel angewendet werden (Abb. 6.2, Anwendungsorientierte Instruktion). Statt alle Konzepte zunächst zu beschreiben, dann Fertigkeiten zu üben und erst danach (meistens erst nach Monaten oder Jahren) in der realen Welt anzuwenden, sollen die Konzepte bereits während der Instruktion gleich zusammen mit den zugehörigen Fertigkeiten eingeübt und angewendet werden. Auf diese Weise angeeignetes Wissen soll auf den selbständigen Teil des PBL vorbereiten und den Kompetenzaufbau in Gang bringen.

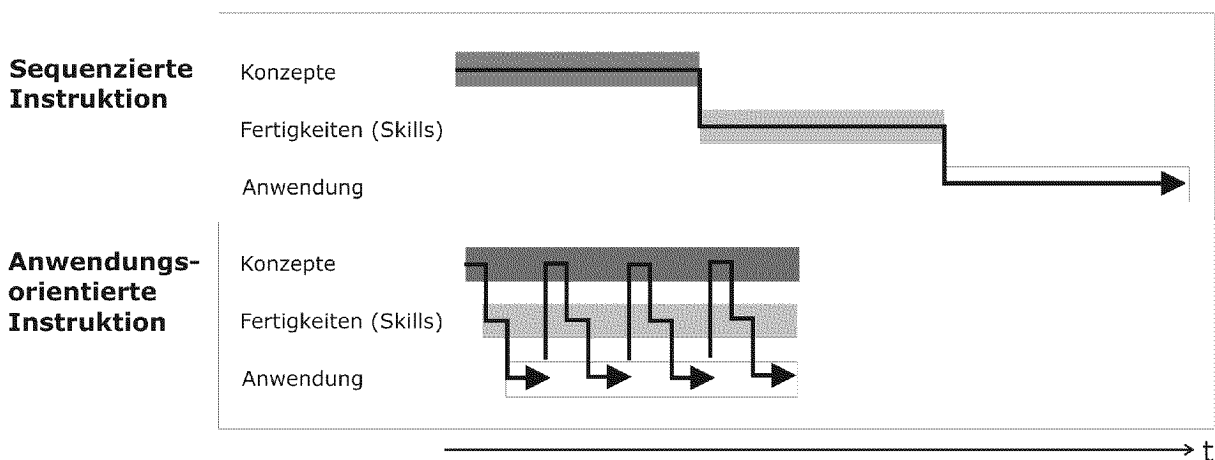


Abb. 6.2: Ablauf der Instruktion durch die Vermittlung von Konzepten und Fertigkeiten sowie deren praktische Anwendung. Sequenzierte Instruktion: Konzepte, Fertigkeiten und deren Anwendung geschehen zeitlich nacheinander. Das erworbene Wissen bleibt träge. Anwendungsorientierte, problembasierte Instruktion: Die einzelnen Konzepte werden durch die sofortige Anwendung mit Fertigkeiten verknüpft. Das Wissen steht bei der späteren Anwendung als aktives Wissen zur Verfügung.

6.3.2 Konstruktive Begriffsbildung

Eine weitere wichtige Aufgabe der Instruktion ist die Einführung von Begriffen. Damit ist die Einführung jener Begriffe gemeint, von denen sich die Lernenden im Verlaufe einer Lehrveranstaltung ein differenziertes Bild aneignen sollen. Traditionellerweise werden die Grundbegriffe in Vorlesung und Text als allgemeingültige Definitionen eingeführt. Soll ein Begriff von den Lernenden nicht bloss als Definition wiedergegeben, sondern als differenziertes kognitives Werkzeug zur Lösung von Problemen eingesetzt werden können, ist die Aneignung und Verknüpfung von Wissen aller Wissens-

dimensionen notwendig (folglich Kapitel 2.3.3 „Unterschiede bei der Aneignung verschiedener Wissensarten“). Es stellt sich die Frage, wie ein Begriffsverständnis dieser Qualität aufgebaut werden kann (siehe Kapitel 2 „Aneignung von Wissen über Begriffe“).

Durch die Problem- und Handlungsorientierung der Instruktion ergibt sich die Möglichkeit einer *konstruktiven Begriffsbildung*. Das bedeutet, dass ausgewählte Begriffe so in die Anleitung eingebaut werden, dass sie im Zusammenhang mit dem zu lösenden Problem wiederholt auftreten, mit Erfahrungen verbunden und mit Wissen anderer Dimensionen verknüpft werden (Abb. 6.3).

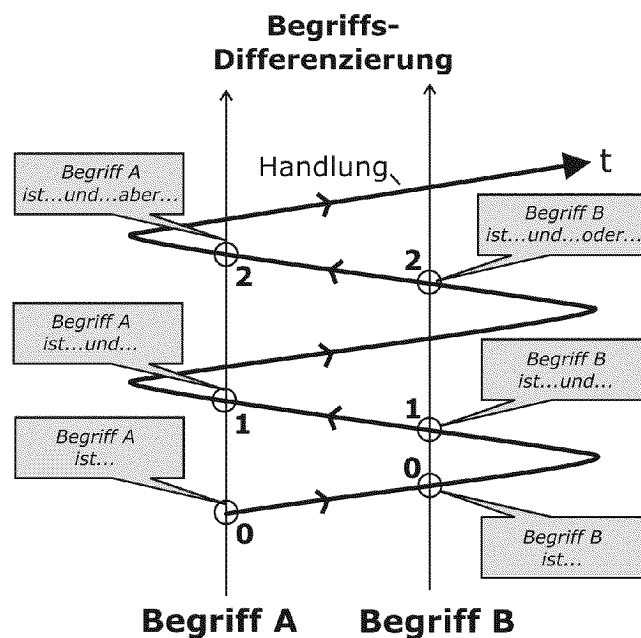


Abb. 6.3: Konstruktiver Aufbau der Begriffe A und B in einer handlungs- und problemorientierten Instruktion. Die Begriffsbildung wird zum Zeitpunkt 0 in der Anleitung gestartet. Auf dem Weg zur Kompetenz kann der Begriff aus einer realen Handlung heraus schrittweise ausdifferenziert werden. Das Wissen über einen Begriff wächst dadurch mit der Erfahrung und den Fähigkeiten der Lernenden.

Um mit Hilfe der Anleitung Begriffe einzuführen und lernendenzentriert ausdifferenzieren zu lassen, ist eine Startration an Erklärung notwendig. Dieser Anfangspunkt der Begriffskonstruktion hat nicht den Anspruch, vollständig zu sein. Er enthält die Informationen, die für das zu lösende Problem von Interesse sind. Zu diesem Zweck erweist sich die Verwendung von Metaphern und Bildern an Stelle von komplexen Begriffskonstrukten als sehr effizient. Die dadurch erreichte Einfachheit ermöglicht, dass die Lernenden sehr schnell mit der praktischen Umsetzung beginnen können. Dies verhindert, dass die Anleitung die Form eines umfangreichen Textbuches annimmt, bei dem auf Anwendungsebene doziert wird. Prozedurales Wissen im Zusammenhang mit einem Begriff muss mit realen Handlungen verbunden sein (folglich Abschnitt 6.2 „Anleitungen als Instruktionsmedium“). Durch die Handlungsorientierung und dem damit verbundenen Wissenserwerb werden die Begriffe durch die Lernenden selber gebildet, mit Erfahrungen verknüpft und laufend ausdifferenziert. Lässt man Lernende

zu unterschiedlichen Zeitpunkten im Lernprozess Begriffe verbalisieren, fallen diese auf konstruktive Weise gebildeten Definitionen nicht nur immer ausführlicher und differenzierter aus, sie sind auch mit Wissen unterschiedlicher Dimensionen verbunden (Abb. 6.3).

Auf konstruktive Weise erworbene Begriffsbilder haben den Vorteil, dass sie für die Lernenden über die Instruktion hinweg erweiterbar sind und durch die Handlungsorientierung einen Bezug zur Praxis haben. Sie bleiben allerdings auch stark mit dem Kontext des erarbeiteten Problems verbunden, was der völlig freien Übertragbarkeit auf neue Situationen entgegenwirken kann [Mietzel 2003].

6.4. Elemente einer selbstgesteuerten, problembasierten Anleitung

Bei einer selbstgesteuerten, problembasierten Instruktion in Form einer Anleitung ist das zu lösende Problem in kleinere, überprüfbare Schritte aufzuteilen. Die Kompetenz soll durch eigenes Handeln schrittweise aufgebaut werden, um auf das spätere, selbständig zu lösende Problem vorzubereiten. Die Anleitung muss dabei zwei Aufgaben erfüllen: erstens die zur Lösung des Schrittes notwendigen Konzepte und Fertigkeiten bereitstellen und zweitens eine Möglichkeit bieten, die erfolgreiche Ausführung der Handlung als Teilschritt der Aufgabe selbständig zu überprüfen.

6.4.1 Interaktion von Anleitung, Handlung und Überprüfung

Um die problembasierte Instruktion handlungsorientiert und selbstgesteuert zu gestalten, muss die Anleitung in starker Interaktion mit der Handlung und deren Überprüfung stehen. Die Anleitung umfasst die Problemstellung und die Einführung von Konzepten und Fertigkeiten. Die unmittelbare Umsetzung des Problems in einer Anwendung führt den Handlungsschritt aus. Eine Überprüfungsmöglichkeit stellt sicher, dass der Handlungsschritt von den Lernenden durch Vergleich mit einer für den gegebenen Handlungsschritt relevanten Lösung richtig umgesetzt und abgeschlossen wird, bevor in der Anleitung der nächste Schritt angegangen wird (Abb. 6.4).

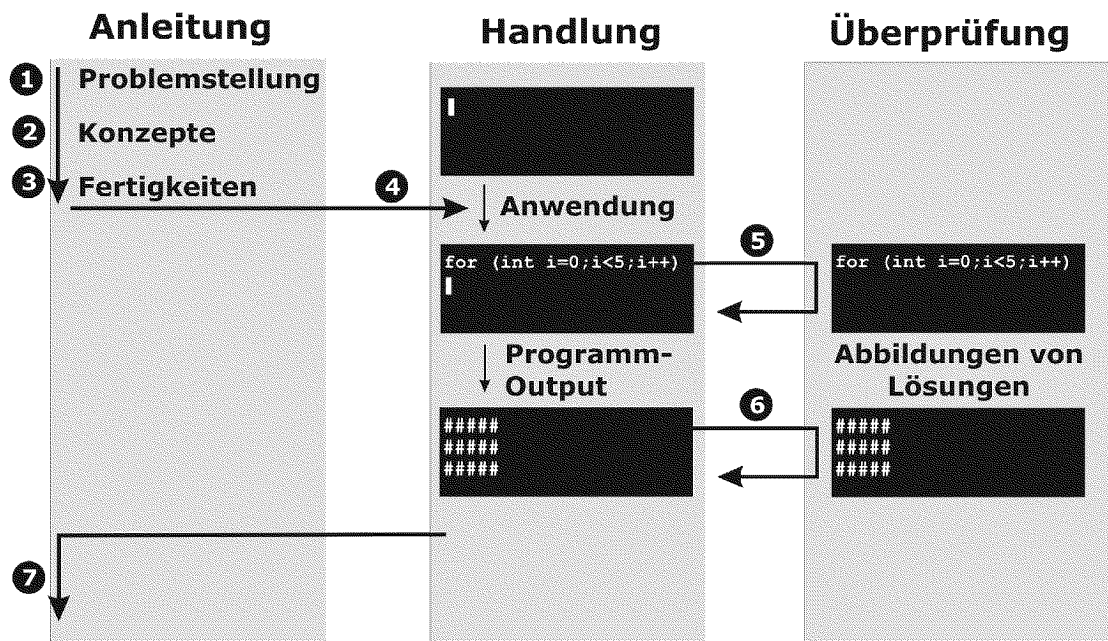


Abb. 6.4: Zusammenspiel zwischen Anleitung, Handlungsschritt und Überprüfung zur Lösung eines Problems einer selbstgesteuerten, problembasierten Instruktion. In der Anleitung macht eine Problemstellung (1) die Aneignung von Konzepten und Fertigkeiten (2, 3) (z.B. wiederholte Programmausführung) notwendig. Die unmittelbare Umsetzung in einer Anwendung (4) führt zum Handlungsschritt (z.B. der Programmierung einer Schleife in einer Programmierumgebung). Durch die Möglichkeit der Überprüfung von Handlungsschritten (5) oder des Programm-Outputs (6) (z.B. mittels Vergleich mit einer Vorlage des gewünschten Resultats) können die Lernenden selber feststellen, ob der Schritt abgeschlossen ist oder ob zur Erfüllung des Schrittes weitere Handlungen notwendig sind. Stimmt das Resultat der Handlung mit jenem der Überprüfungsmöglichkeit überein, können weitere Schritte der Anleitung angegangen werden (7).

6.4.2 Abgleich von Ist und Soll als Handlung

Durch die Kombination von Anleitung und Anwendung sind die Lernenden gezwungen, eigene Handlungen auszuführen. Der Anfangs- oder Ist-Zustand entspricht dabei dem Zustand der zu erlernenden Anwendung vor dem Handlungsschritt (Abb. 6.5). Diese Anwendung ist durch eine Handlung der Lernenden veränderbar. Die Anleitung zusammen mit den Elementen Aufgabenstellung, Konzepten und Fertigkeiten weist den Lernenden den Weg Richtung Soll-Zustand der Anwendung für den gegebenen Handlungsschritt. Umsetzen müssen sie den Schritt aber selber. Durch die Möglichkeit, Soll-Zustände in Form von Bildern als statische „Wegpunkte“ anzuzeigen (siehe Abb. 6.4), können die Lernenden ihre Handlung selber steuern, indem sie die Differenz zwischen gewünschtem (Soll) und tatsächlichem (Ist) Output durch eine Handlung ausgleichen (Abb. 6.5).

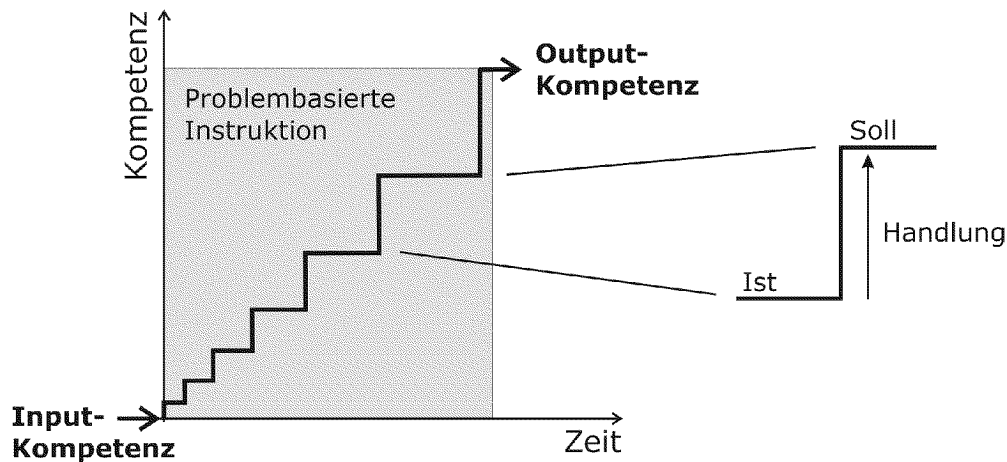


Abb. 6.5: Aufbau und Selbstüberprüfung einer (Teil-)Kompetenz einer problembasierten Instruktion mittels Anleitung. Überprüfungsmöglichkeiten helfen zusammen mit der Anleitung beim Abgleich von Ist- und Soll-Zustand, deren Differenz durch Handlungen der Lernenden ausgeglichen werden kann. Die Anleitung beginnt mit kleinen Schritten, welche mit zunehmender Selbständigkeit der Lernenden stetig vergrößert werden.

Damit die Schritte selbständig bewältigt werden können, dürfen sie zu Beginn nicht allzu gross sein. Dies hat damit zu tun, dass am Anfang zusätzlich Konzepte und Fertigkeiten eingeführt werden, eine Problemstellung verstanden werden muss, und gleichzeitig noch kein Wissensnetz und keine Erfahrung zur Verfügung stehen. Im Verlaufe der Instruktion kann die Stufengrösse aber stetig erhöht werden (siehe Abb. 6.5). Die im Anschluss an die Instruktion selbständig zu lösende Anwendungsaufgabe (siehe Abb. 5.8) entspricht im Idealfall einer weiteren Erhöhung der Stufengrösse. Diese Zunahme der Selbständigkeit, die fließend geschieht, führt dazu, dass die spätere Anwendung weniger als Praxisschock wahrgenommen wird. Der Übergang von der Instruktionsphase zur Anwendungsphase wird für die Lernenden erheblich erleichtert.

6.4.3 Motivationale und emotionale Aspekte während der Instruktion

Wie im Abschnitt „Motivationale und emotionale Faktoren beim PBL“ diskutiert, haben die beim Lösen von Problemen gemachten Erfahrungen in einem Fachbereich einen erheblichen Einfluss darauf, wie nachfolgend gestellte Probleme gelöst werden (siehe Abb. 5.7). Handelt es sich dabei um erste Erfahrungen, ist es umso wichtiger, wie diese ausfallen. Deshalb hat die problembasierte, angeleitete Instruktion neben der Vermittlung von Konzepten und Fertigkeiten zusätzlich die Aufgabe, eine Reihe psychologischer Faktoren in einen für den Lernprozess förderlichen Bereich zu entwickeln.

Folgende für den Lernprozess zentralen psychologischen Aspekte sollen zusammen mit der problembasierten Instruktion gefördert werden:

- 1. Sicherheit und Vertrauen
- 2. Erfolgserlebnisse und Selbstwirksamkeit

- 3. internale Kontrollüberzeugung und intrinsische Motivation.

Die Definitionen der psychologischen Konstrukte sind dem Kapitel 5.3.2 „Motivationale und emotionale Aspekte beim PBL“ zu entnehmen.

1. Sicherheit und Vertrauen

Die Lernenden investieren zur Aneignung einer Kompetenz Zeit und Energie und sind speziell zu Beginn einer indirekten Instruktion darüber im Ungewissen, ob mit Hilfe der vom Lernsystem bereitgestellten Strukturen und ihrem geleisteten Einsatz die erforderliche Zielsetzung und der Aufbau der Kompetenz erreicht werden können. Der Einstieg ist beim Aufbau einer Kompetenz immer mit Unsicherheiten verbunden. Das ist vergleichbar mit dem Ankommen in einer fremden Grossstadt, in der man sich noch nicht auskennt. Zu diesem Zeitpunkt sind weder Wissensnetz noch Erfahrung vorhanden, um sich in der fremden Welt selbständig bewegen zu können. Aus diesen Gründen ist die Schaffung von Strukturen wichtig, welche den Lernenden Sicherheit und Vertrauen verleihen, auf dem richtigen Weg Richtung Ziel zu sein und gleichzeitig die Aneignung von Wissensnetzen und Erfahrung fördern. Am schnellsten am Ziel wären in der unbekanntem Stadt sicher jene Personen, die sich mit einem Taxi ans Ziel fahren lassen. Diese werden aber kaum Selbständigkeit entwickeln können. Am schnellsten selbständig werden hingegen jene, die sich mit Hilfe eines Stadtplans selber orientieren. Der Stadtplan verleiht Sicherheit, indem der momentane Standort laufend überprüft und mit dem Zielort abgeglichen werden kann. Voraussetzung für das Vertrauen in die Karte ist, dass mit Hilfe der Karte das Ziel erreicht werden kann. Das heisst, sie muss genügend genau und aktuell sein. Bei einem nächsten Besuch dieses Stadtteils wird man sich schon viel selbständiger orientieren können. Die Karte muss weniger oft konsultiert werden, bis mit zunehmender Selbständigkeit und Sicherheit ganz darauf verzichtet werden kann.

Eine ideale Instruktionsanleitung übernimmt auf dem Weg von A nach B in einer komplexen Welt ebenfalls die Funktion einer Struktur, die Sicherheit verleiht. Über die visuellen Überprüfungsmöglichkeiten der eigenen Handlungsschritte in zu Beginn zeitlich kurzen Abständen (siehe Abb. 6.5) finden die Lernenden auf ihrem Weg Richtung Problemlösung trotz Eigenaktivität immer wieder selbständig zurück zum angeleiteten Weg. Wie beim Blick auf einen Stadtplan können sie sich auf dem Weg zu einem bestimmten Ziel hin orientieren.

Mit zunehmender Erfahrung wachsen die Wissensnetze, die Sicherheit nimmt zu und es kann zunehmend auf diese Hilfsstrukturen verzichtet werden.

2. Erfolgserlebnisse und Selbstwirksamkeit

Ein Lernsystem, das nur Forderungen stellt und dem die Lernenden trotz intensiver Bemühung in der zur Verfügung stehenden Zeit unmöglich genügen können, ist länger-

fristig frustrierend (folglich Kapitel 5.3.2 „Zufriedenheit“). Um zu vermeiden, dass die Lernenden durch zu hohe Zielsetzungen überfordert werden, ist es wichtig, dass sie bereits zu Beginn des Lernprozesses Möglichkeiten erhalten, für sie wahrnehmbare Fortschritte und Erfolgserlebnisse zu erzielen und dadurch einem geforderten Sollzustand zu genügen (Abb. 6.5). Solche Erfolgserlebnisse wirken sich nachhaltig positiv auf den weiteren Verlauf des Lernprozesses aus (folglich Kapitel 5.3.2 „Erfolgserlebnisse“).

Die Überzeugung, sich eine Kompetenz mit Hilfe der zur Verfügung gestellten Strukturen aneignen zu können, hängt sehr stark mit den ersten Erfahrungen beim Lernen eines bestimmten Inhalts zusammen. Weil die Instruktion am Anfang steht, wirken sich die Erfahrungen, die während der Instruktion gemacht werden, stark auf den weiteren Verlauf des Lernprozesses aus. Dabei ist es während der Instruktion nicht nur wichtig, dass laufend Fortschritte erzielt werden, sondern dass diese von den Lernenden auch als solche wahrgenommen werden.

Die Form der handlungsorientierten selbstgesteuerten Instruktion kann neben den Konzepten auch die Überzeugung vermitteln, sich eine bestimmte Kompetenz durch eigenes Handeln aneignen zu können. Diese positive Selbstwirksamkeitserwartung, die während der Instruktionsphase aufgebaut wird, hat einen starken Einfluss darauf, wie spätere selbständige Anwendungsaufgaben gelöst werden (folglich Kapitel 5.3.2 „Selbstwirksamkeitserwartung“).

3. Internale Kontrollüberzeugung und intrinsische Motivation

Weil die Lernenden bei der Anleitung der problembasierten Instruktion innerhalb von Leitplanken selber Handlungen ausführen und überprüfen, kann sie von den Lernenden selber gesteuert werden. Dabei wird die Handlung der Lernenden während des Lernprozesses von aussen nicht überprüft. Die Lernenden erhalten das Gefühl, selber Kontrolle über ihre Wissensaneignung zu haben und können eine internale Kontrollüberzeugung entwickeln.

Weil die Lernenden wissen, dass sie während der problembasierten Instruktion nicht überprüft werden und mit keinen Konsequenzen rechnen müssen, wagen sie mehr und machen dadurch auch Fehler. Durch die bereitgestellten Überprüfungsmöglichkeiten können sie diese Fehler aber selbständig beheben und sammeln dadurch für den Lernprozess wichtiges Wissen und Erfahrungen.

Der Beweggrund für das Handeln kommt bei der selbstgesteuerten, problembasierten Instruktion einerseits aus der eigenen Handlung (intrinsische Motivation) und liegt andererseits in der zu lösenden Problemstellung (Abb. 6.6 und Tab. 6.1). Diese ist Teil der indirekten Instruktion und von aussen gegeben. Die Motivation der Lernenden entsteht durch die Interaktion mit den Systemstrukturen der Lernumgebung. Obwohl die Problemaufgabe von aussen gegeben ist, wird die Motivation intrinsisch. Durch das eigene Handeln der Lernenden während der Instruktion wirkt sie nicht belehrend, und die

Lernenden beziehen das Problem auf sich und erhalten die Bestätigung, es selber bewältigt zu haben.

Die Überprüfung von aussen als Teil der Leistungskontrolle erfolgt erst nach Ende des Lernprozesses, indem die Lernenden die Möglichkeit haben, ihr gelöstes Problem zusammen mit ihrer erworbenen Kompetenz zu zeigen und in Form eines Leistungsnachweises anrechnen zu lassen (siehe Kapitel 7.2 „Das 4-Schritte-Modell“ und Kapitel 9 „Einsatz der computergestützten Lernumgebung im Unterricht“).

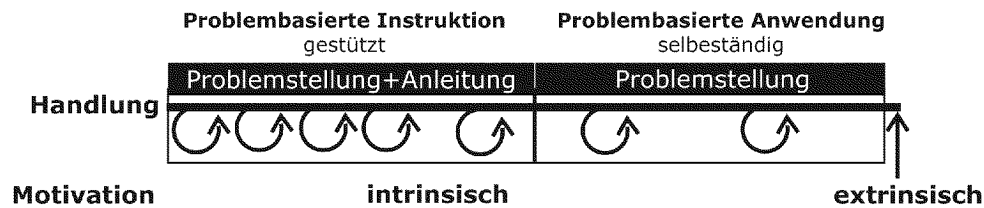


Abb. 6.6: Motivationsmodell beim PBL mit selbstgesteuerter, problembasierter Instruktion. Die durch Anleitung und Problemstellung erzeugten Handlungen während der Instruktionsphase ermöglichen, dass der Anreiz für das Handeln in der Tätigkeit selbst liegt. Die Lernenden motivieren sich intrinsisch. Dies ist eine Voraussetzung für selbständiges Handeln und Lernen während der Anwendungsphase. Die Kontrolle und Beurteilung von aussen durch andere Personen wird auf den Moment verschoben, wo die Probleme durch die Lernenden gelöst sind und der Lernprozess abgeschlossen ist.

Art der Überprüfung	Lernprozess	Lernprodukt
Was wird überprüft?	Einzelschritte	(Teil-)Kompetenz als Ganzes
Wer überprüft?	Selbstüberprüfung	Lehrende/Assistierende
Art der Motivation	intrinsisch	extrinsisch
Häufigkeit der Rückmeldung	mehrfach	1 Mal nach Abschluss

Tab. 6.1: Art der Motivierung beim PBL mit problembasierter Instruktion.

6.5. Einsatz des Computers als Instruktionsmittel

Eine selbstgesteuerte, problemorientierte Instruktion an einem Computer durchzuführen, ist gegenüber traditioneller Instruktionsformen und -medien (Buch, Vorlesung usw.) unter anderem aus folgenden Gründen interessant: Unabhängigkeit, Unterrichtsqualität, Studierendenmotivation und Realitätsnähe.

1. Unabhängigkeit der Instruktion von Ort, Zeit und Personen

Eine mittels Computer durchgeführte Instruktion hat den Vorteil, dass die Instruktion unabhängig von Ort, Zeit und Lehrpersonen durchgeführt werden kann. Sie erlaubt ein individuelles Lernen im eigenen Tempo und kann zu jeder Zeit unterbrochen und wieder aufgenommen werden.

2. Höhere Qualität des Unterrichts dank Automatisierung und Individualisierung

Die Instruktion als erstes Element der Kompetenzaneignung ist Startpunkt einer ganzen Reihe von Prozessen. Die Qualität dieser Phase des Lernprozesses spielt für dessen weiteren Verlauf eine zentrale Rolle. Mit der Durchführung der Instruktion am Computer kann sie auch in grossen Gruppen angewendet werden und trotzdem individuell eine hohe Qualität garantieren. Eine solche Automatisierung der Instruktion muss nicht bedeuten, dass beim Lernprozess auf zwischenmenschliche Kontakte verzichtet werden soll.

3. Intrinsische Motivation durch Handlungsorientierung

Über die Möglichkeit, eigene Handlungsschritte am Computer selber überprüfen zu können, kann ein hoher Grad an Selbststeuerung durch die Lernenden erreicht werden. Die dadurch erreichte intrinsische Motivierung der Lernenden wirkt sich nachhaltig positiv auf den Lernprozess aus (folglich Abschnitt „internale Kontrollüberzeugung und intrinsische Motivation“).

4. Aneignung von Kompetenzen in realen Situationen

Der Computer dient als Arbeitsmittel zur Lösung von Problemen verschiedener Disziplinen (folglich Kapitel 3 „ICT-Kompetenzen für die wissenschaftliche Tätigkeit“). Werden Problemlöswerkzeuge mit realen Situationen verknüpft, ergibt sich eine Lernumgebung zur Aneignung von Kompetenzen, die mit traditionellen Instruktionsformen nicht erreicht werden kann.

6.5.1 *E.Tutorials*[®]: Applikation für selbstgesteuerte, problembasierte Instruktionen

Zur Umsetzung von selbstgesteuerten, problembasierten Instruktionen mit Hilfe des Computers wurde eine Applikation mit dem Namen *E.Tutorial*[®] entwickelt, bei der sich Anleitung, Werkzeug und Überprüfbarkeit auf einer Arbeitsfläche befinden [Hinterberger 2004, Faessler 2004]. Diese Anordnung ermöglicht eine starke Interaktion zwischen den drei Elementen der problembasierten Instruktion (Abb. 6.7).

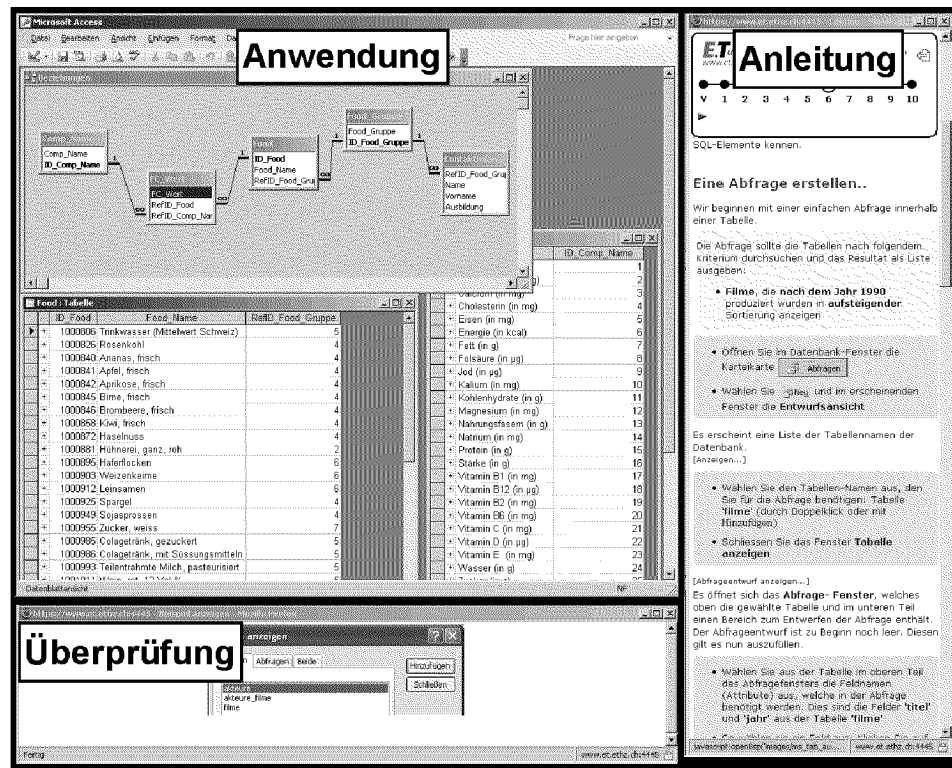


Abb. 6.7: *E.Tutorial*[®], eine Applikation für die elektronische, problembasierte Instruktion, bestehend aus Anleitung, Handlung und Überprüfung. Charakteristisch für ein *E.Tutorial*[®] ist die Aufteilung des Bildschirms in drei Fenster. Im Fenster oben links läuft eine Computeranwendung (z.B. ein Datenbankprogramm), mit der die Studierenden ein Problem (z.B. der Umgang mit relationalen Datenbanken) lösen müssen. Im zweiten Fenster am rechten Bildschirmrand erhalten sie Instruktionen zur Lösung einer konkreten Aufgabe. Im dritten Fenster können die Lernenden überprüfen, ob sie sich im Problemlösungsprozess auf dem richtigen Weg befinden. Die Überprüfungsmöglichkeiten haben die Form von visuellen Wegpunkten, mit deren Hilfe die Studierenden Zwischenresultate mit ihrer eigenen Lösung vergleichen können.

Der Einsatz von *E.Tutorials*[®] als Instruktionsmittel für den kompetenzorientierten Unterricht wird im Rahmen dieser Arbeit in den folgenden Kapiteln behandelt:

- Kapitel 7 „Didaktische Modelle“: Einsatz von Medien zur Umsetzung des 4-Schritte-Modells
- Kapitel 8 „Entwicklung und Umsetzung kompetenzorientierter Lerneinheiten“: Entwicklung von Instruktionsmaterialien für kompetenzorientierte Lernumgebungen
- Kapitel 9 „Einsatz der computergestützten Lernumgebung im Unterricht“: Unterrichten mit *E.Tutorials*[®]
- Kapitel 10 „Evaluationen der Lernumgebungen“: Evaluation des *E.Tutorial*[®]-basierten Unterrichts
- Kapitel 11 „Schlussfolgerungen und Ausblick“: Weiterentwicklungsmöglichkeiten.

Beschreibung und Umsetzung der didaktischen Modelle

Kapitel 7

Didaktische Modelle

7.1. Einleitung

Im ersten Teil dieser Arbeit wurde argumentiert, dass Unterrichtsmethoden und Medien alleine noch keine Kompetenzaneignung garantieren. Kompetenzen entstehen durch eine Organisationsstruktur, die im *didaktischen Modell* beschrieben wird (Details zum Begriff des didaktischen Modells siehe Kapitel 4.2.3 „Lernmethoden und didaktische Modelle“).

Der Aufbau von Kompetenzen im Unterricht muss letztlich in Unterrichts- oder Lerneinheiten organisiert werden, die jeweils verschiedene Teilaspekte einer Gesamtkompetenz betrachten. Um beispielsweise in der Kompetenz des Programmierens ein bestimmtes Niveau zu erreichen, könnte ein solcher Teilaspekt das Betrachten von Programmwiederholungen mit Schleifen oder das Speichern von Datenreihen in Arrays sein. Unterrichts- oder Lerneinheiten, die solche Teilaspekte behandeln, führen zu Teilkompetenzen wie zum Beispiel dem kompetenten Umgang mit Schleifen oder Arrays. Aus der Sicht der Konzepte sind es Kompetenzen, aus der Sicht der Gesamtkompetenz jedoch Teilkompetenzen.

Kompetenzen, die über einen ganzen Kurs hin angeeignet werden sollen, müssen demnach in sinnvolle Teilkompetenzen unterteilt werden. Aus diesen Teilkompetenzen sollte sich während einer Unterrichtseinheit durch den im didaktischen Modell beschriebenen Ablauf aus Unterrichtsmethoden und durch die Sequenzierung und Auswahl von Medien eine Kompetenz aufbauen lassen. Dies setzt voraus, dass die einzelnen Teile zusammenpassen.

Das didaktische Modell beschreibt für eine Lerneinheit:

- den zeitlichen Ablauf
- die Aktivitäten der Lernenden
- die Sequenzierung
- die Auswahl geeigneter Medien für einzelne Sequenzen

7.1.1 Zeiträume zum Aufbau von Kompetenzen

Der Aufbau einer Kompetenz setzt sich aus verschiedenen Teilaspekten einer Kompetenz zusammen. Jeder dieser Teilaspekte wird kompetenzorientiert behandelt, was aus der Sicht der Gesamtkompetenz jeweils zu einer Teilkompetenz führt. Der Aufbau von Kompetenzen findet somit auf drei verschiedenen Ebenen statt (Abb. 7.1 und Tab. 7.1):

- A) Die *Instruktion* mit dem Ziel der Aneignung von Konzepten und Fertigkeiten (Details siehe Kapitel 6 „Selbstgesteuerte, problembasierte Instruktion“)
- B) Die *Unterrichts- oder Lerneinheit*, bestehend aus einer problembasierten Instruktion (A) und einer Anwendungsphase, die zu einer Teilkompetenz führt (Details siehe Kapitel 5 „PBL“).
- C) Ein *Kurs*, bestehend aus einer bestimmten Anzahl Unterrichtseinheiten (B), die zur Kurskompetenz führen (Details siehe Kapitel 4 „Kompetenzorientierte Lernumgebungen“).

Die einzelnen Phasen stehen zueinander in einer hierarchischen Beziehung (siehe Abb. 7.1), was bedeutet, dass eine übergeordnete Phase die Eigenschaften der darunterliegenden Phase beeinflusst und umgekehrt eine Phase immer auch auf die Zielsetzung der nächst höheren Ebene vorbereiten muss.

Für die hier beschriebenen didaktischen Modelle sind daher die Übergänge von einer Ebene zur nächst höheren entscheidend:

- Wie werden aus Konzepten und Fertigkeiten der Instruktion Teilkompetenzen einer Unterrichtseinheit (Übergang Ebene A → Ebene B)?
- Wie wird aus Teilkompetenzen einer Unterrichtseinheit eine Kurskompetenz (Übergang Ebene B → Ebene C)?

Ebene	Zeitraum / Phase	Dauer	Zielebene / Wissensdimensionen
A	Instruktion	4-8 Stunden	Konzepte + Fertigkeiten
B	Unterrichts- oder Lerneinheit	2 Wochen	Teilkompetenz Konzepte + Fertigkeiten + Fähigkeiten
C	Kurs	14 Wochen	Kurskompetenz Teilkompetenz 1 + Teilkompetenz 2 + ...

Tab. 7.1: Phasen beim Aufbau einer Kompetenz. Die Verknüpfung von Konzepten und Fertigkeiten während der problembasierten Instruktion (Ebene A) und die anschließende problembasierte Anwendung führen zur Aneignung von Fähigkeiten auf der Ebene einer Teilkompetenz (Ebene B). Einzelne Teilkompetenzen fügen sich in einem Kurs zu einer Kurskompetenz zusammen (Ebene C). Die Dauer bezieht sich auf einen Semesterkurs mit 4 Wochenstunden Unterricht.

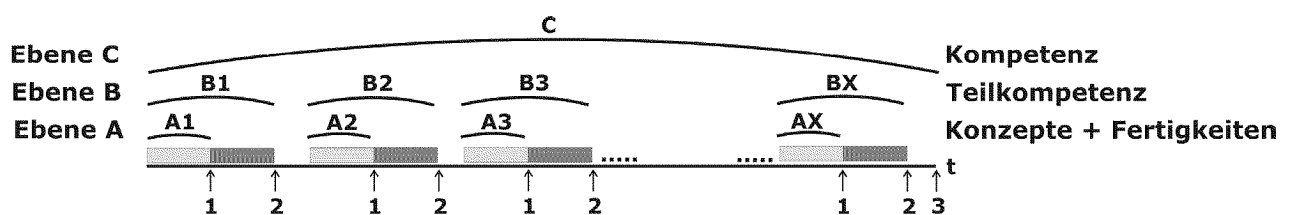


Abb. 7.1: Gliederung des Aufbaus von Kompetenzen in Phasen von unterschiedlichen Zeiträumen: A: Instruktion, B: Unterrichtseinheit, C: Kurs. Entwicklungsziele: Zeitpunkt 1: Konzepte + Fertigkeiten; Zeitpunkt 2: Teilkompetenz: Konzepte + Fertigkeiten + Fähigkeiten; Zeitpunkt 3: Kompetenz: Teilkompetenz 1 + Teilkompetenz 2 + usw.

Der kompetenzorientierte Aufbau von Teilaspekten einer Kompetenz durch eine Reihe von Unterrichtseinheiten steht im Zentrum der Entwicklung eines kompetenzorientierten Unterrichts. Diese führt zur sinnvollen und zielgerichteten Modularisierung des Grundlagenunterrichts und beeinflusst die Auswahl der behandelten Konzepte (für die detaillierte Umsetzung der Unterrichtseinheiten siehe Kapitel 8 „Entwicklung und Umsetzung kompetenzorientierter Lerneinheiten“).

7.1.2 Zwei Modelle für den Aufbau von Kompetenzen

Die Grundlage dieser Arbeit bilden zwei didaktische Modelle, die unter den Bedingungen des realen Unterrichts über mehrere Jahre entwickelt wurden: das *4-Schritte-Modell* und das *Anwendungsebenen-Modell*. Die beiden Modelle ergänzen sich bei der Vermittlung, respektive der Aneignung von Kompetenzen. Sie organisieren die Ebenenwechsel von der Instruktion zur Teilkompetenz (A→B) und von den Teilkompetenzen zur Kurskompetenz (B→C) (Tab. 7.2).

1. 4-Schritte-Modell

Mit Hilfe des 4-Schritte-Modells sollen sich die Lernenden Konzepte zusammen mit Fertigkeiten mittels selbstgesteuerter, problembasierter Instruktion aneignen, um sie

danach bei einem authentischen Problem anwenden zu können. Das Resultat ist aktives, anwendbares Wissen und Fähigkeiten auf der Ebene von Teilkompetenzen.

2. Anwendungsebenen-Modell

Das Anwendungsebenen-Modell stellt eine strukturierte Folge des obigen 4-Schritte-Modells dar. Die Lerneinheiten werden so aneinandergereiht, dass anspruchsvollere Fähigkeiten entwickelt werden können. Das Anwendungsebenen-Modell dient der Aneignung von Kompetenzen, die die Anwendung mehrerer Konzepte und Fähigkeiten in Kombination erfordern.

Ebenenwechsel	Modell	Ziel	Dauer	Vorgehen
A → B	4-Schritte-Modell	Einführen von Konzepten zusammen mit Fertigkeiten und Fähigkeiten	4 - 8 Stunden	problembasierte Instruktion und Anwendung
B → C	Anwendungsebenen-Modell	Entwicklung von anspruchsvolleren Fähigkeiten und Kompetenzen	4 - 8 Wochen	Folge von aufbauenden Problemen im 4-Schritte-Modell

Tab. 7.2: Zwei didaktische Modelle zur Organisation des Ebenenwechsels. Die Zeiträume beziehen sich auf einen Semesterkurs von 14 Wochen mit 4 Wochenstunden Unterricht.

Als nächstes sollen die beiden Modelle im Detail betrachtet werden. Das folgende Kapitel 8 zeigt deren detaillierte Umsetzung am Beispiel eines Kurses.

7.2. Das 4-Schritte-Modell

Das 4-Schritte-Modell organisiert problembasiertes Lernen mit zunehmender Selbständigkeit der Lernenden. Sie sollen sich mit Hilfe der im Modell eingesetzten Strukturen nicht bloss über Konzepte und Fertigkeiten informieren können, sondern diese soweit einüben, dass sie als aktives, transferfähiges Wissen zur Lösung von größeren Problemen zur Verfügung stehen (siehe dazu auch Kapitel 2.3.2 „Verschiedene Arten von Wissen“, 3.2.3 „Computer Fluency“, 5.4 „Effektives und motivierendes PBL“ und 6.3 „Aufbau eines aktiven Wissensnetzes“).

7.2.1 In 4 Schritten zu aktivem, transferfähigem Wissen

Beim 4-Schritte-Modell steht der problembasierte Wissenserwerb von Konzepten zusammen mit Fertigkeiten im Zentrum, um aktives Wissen für das selbständige Lösen eines Problems zur Verfügung zu stellen. Ein Konzept wird dadurch nicht bloss auf deklarativer Ebene, sondern als umfassendes Netz mit Wissen unterschiedlicher Dimen-

sionen verinnerlicht (siehe Abb. 7.2). Dieses Wissen soll später beim Lösen umfangreicherer Probleme als flexibel einsetzbares Wissen zur Verfügung stehen.

Das 4-Schritte-Modell besteht aus folgenden 4 Phasen (Tab. 7.3):

Schritte		Beschreibung
1	<i>SEE</i>	Die Lernenden haben die Möglichkeit, sich über die Konzepte und Probleme zu informieren.
2	<i>TRY</i>	Die Lernenden erhalten die Möglichkeit, die Konzepte unter Anleitung aktiv anzuwenden.
3	<i>DO</i>	Die Lernenden wenden die Konzepte selbständig an.
4	<i>EXPLAIN</i>	Die Lernenden müssen ihre Lösung mündlich zusammenfassend formulieren.

Tab. 7.3: Die 4 Phasen des 4-Schritte-Modells [Faessler 2006].

In Abb. 7.2 ist der schrittweise Aufbau eines Wissensnetzes in verschiedenen kognitiven Dimensionen dargestellt. Die Konzepte werden zunächst auf der deklarativen Ebene betrachtet (*SEE*) und danach zusammen mit einer Anwendung mit prozeduralem Wissen verknüpft (*TRY*). Durch das selbständige Anwenden der Konzepte in einem neuen Problem (*DO*) eignen sich die Studierenden zusätzlich konditionales Wissen an. Im abschliessenden Gespräch mit einer Assistenzperson (*EXPLAIN*) wird das Konzept zusammen mit dem Problemlösungsprozess zusammengefasst, indem die Studierenden einer Assistenzperson erklären, wie die Problemstellung mit Hilfe des Konzepts gelöst wurde.

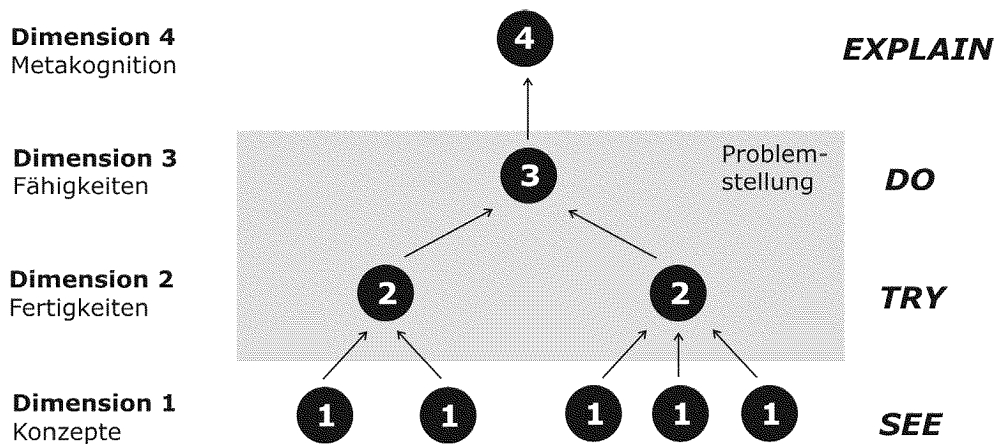


Abb. 7.2: Das 4-Schritte-Modell: In 4 Schritten zu aktivem und flexiblem Wissen unterschiedlicher Dimensionen. Eine realistische Problemstellung bildet den Raum für den Aufbau des Wissensnetzes. Schrittweiser Aufbau in 4 Phasen: 1. Konzepte einführen (*SEE*), 2. Konzepte und Fertigkeiten verknüpfen (*TRY*), 3. Entwickeln von Fähigkeiten durch wiederholtes Problemlösen (*DO*), 4. Entwicklung von meta-kognitivem Wissen durch Reflexion (*EXPLAIN*).

Konzeptwissen wird durch das 4-Schritte-Modell schrittweise verinnerlicht und um eine Wissensdimension erweitert. Das Resultat ist aktives, transferfähiges Wissen.

7.2.2 Zeitlicher Ablauf des 4-Schritte-Modells

Abb. 7.3 zeigt den zeitlichen Ablauf des 4-Schritte-Modells. Die Zeiträume beziehen sich auf einen Semesterkurs von 14 Wochen Dauer mit 4 Wochenstunden Unterricht. Details zu Auswahl und Einsatz der Medien im 4-Schritte-Modell ist Inhalt des nächsten Abschnitts.

1. Phase *SEE*

Die Lernenden erhalten in kompakter Form eine kurze Einführung auf Papier, in der die wesentlichen Konzepte und Begriffe beschrieben werden. Dies soll ihnen helfen, in späteren Phasen der Anwendung die zentralen Konzepte und Begriffe zu fokussieren und diese auszudifferenzieren (siehe auch Abb. 7.2). Dauer 15 bis 20 Minuten.

2. Phase *TRY*

Die Lernenden wenden die Konzepte der Phase *SEE* zusammen mit Fertigkeiten in einer realen Problemstellung an. Durch die Instruktion einer Anleitung wird ein Informatikmittel (z.B. eine Anwendungssoftware oder eine Programmierumgebung) durch eigenes Handeln aktiv bedient. Die einzelnen Schritte der Umsetzung können selber überprüft werden. Während dieser Phase kommt als Medium das *E.Tutorial*[®], ein Verbund aus Anleitung, Anwendung und Überprüfung, zum Einsatz (folglich Kapitel 6 „Selbstgesteuerte, problembasierte Instruktion“). Dauer 2 bis 5 Stunden.

3. Phase *DO*

Die Lernenden wenden die in der Phase *TRY* gelernten Konzepte an einem neuen, umfangreicheren Problem an (folglich Aufgabentypen Kapitel 5 „PBL“). Das Informatikmittel muss zur Lösung des Problems nun selbständig bedient werden. Die Aufgabenstellung erhalten die Studierenden auf Papier. Studierende mit ausreichendem Vorwissen können auch an dieser Stelle einsteigen. Dauer 3 bis 5 Stunden.

4. Phase *EXPLAIN*

Als Abschluss des 4-Schritte-Modells müssen die Lernenden in der Phase *EXPLAIN* in einer Face-to-Face Situation einer Assistenzperson in eigenen Worten erklären, wie sie das Problem der Phase *DO* gelöst haben. Dauer 5-10 Minuten.

Details zum Unterricht mit dem 4-Schritte-Modell siehe Kapitel 9 „Einsatz der computergestützten Lernumgebung im Unterricht“.

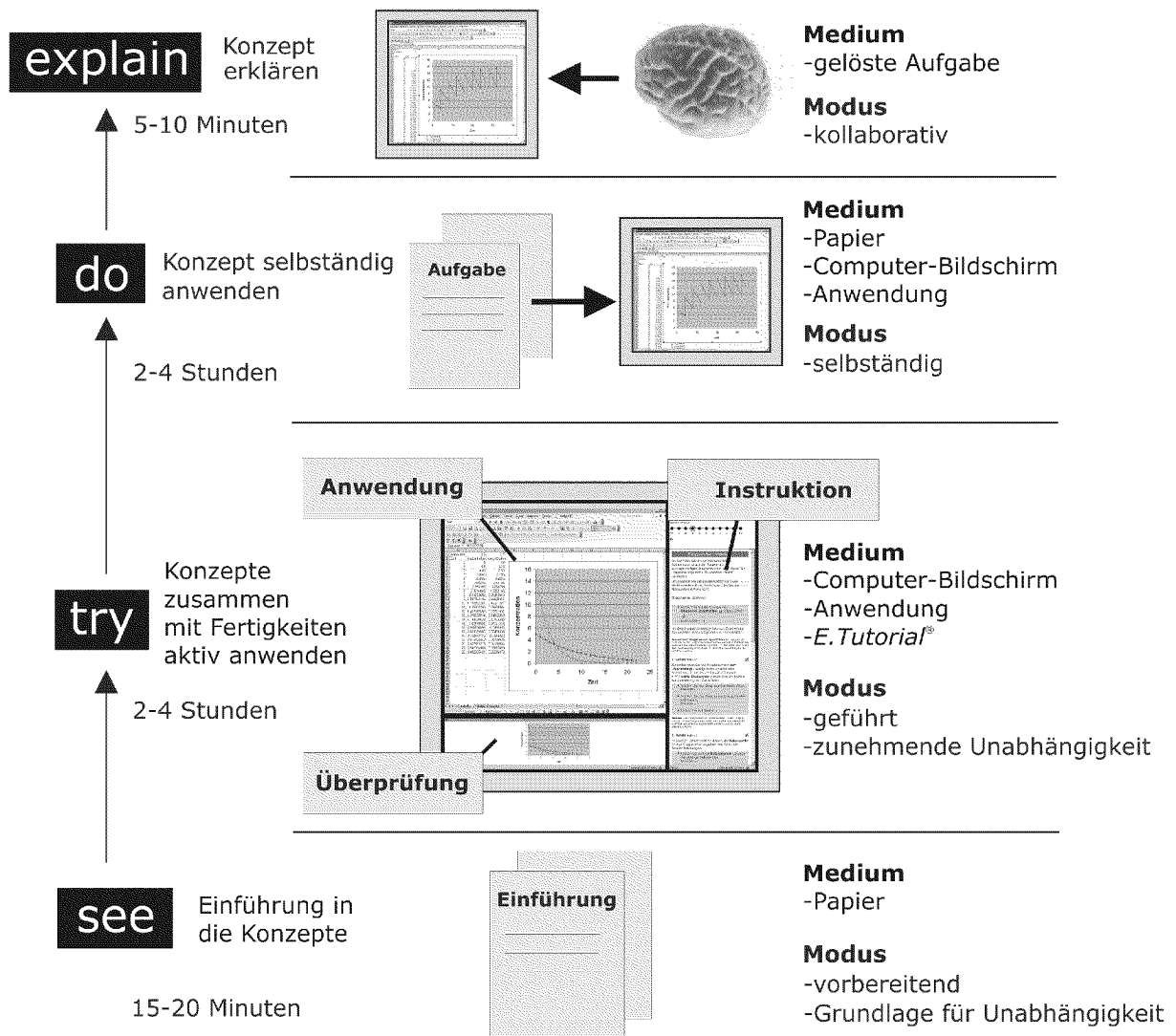


Abb. 7.3: Zeitlicher Ablauf des 4-Schritte-Modells und eingesetzte Medien. Details siehe Text [abgeändert nach Faessler 2006 und Hinterberger 2007].

7.2.3 Einsatz von Medien im 4-Schritte-Modell

Zur Wissensvermittlung steht heute eine Vielzahl unterschiedlicher Medien zur Verfügung. Wir haben uns deshalb überlegt, welches Medium bei den verschiedenen Phasen am wirkungsvollsten eingesetzt werden kann. Zur Umsetzung des 4-Schritte-Modells kommen während verschiedener Phasen unterschiedliche Medien zum Einsatz (siehe Abb. 7.4). Während der Phasen *SEE* und *TRY* sind die meisten Inputs und Hilfen erforderlich. Die Phasen *DO* und *EXPLAIN* erfordern eine selbständige Anwendung. Die durch Medien unterstützte indirekte Instruktion sollte daher bis zur Phase *DO* abgeschlossen sein (folglich Kapitel 4 „Kompetenzorientierte Lernumgebungen“). Als Elemente dieser Instruktion kommen während der Phase *SEE* eine textuelle Einführung und während der Phase *TRY* ein *E.Tutorial*® als interaktive Zusammenführung von Anleitung, Werkzeug und Überprüfungsmöglichkeit zum Einsatz (siehe Kapitel 6.5.1 „*E.Tutorials*®: Applikation für selbstgesteuerte, problembasierte Instruktionen“).

Im Folgenden wollen wir uns damit befassen, welche Medientypen und Präsentationsmedien im 4-Schritte-Modell in welcher Reihenfolge eingesetzt werden (Tab. 7.3). Durch den Einsatz des Computers beim Lösen von Problemen durch die Bedienung von Informatikmitteln findet das 4-Schritte-Modell bereits am Computer statt (siehe Kapitel 3.3 „Herstellen eines realistischen Kontexts zur Aneignung von ICT-Kompetenzen“). Dies hat starken Einfluss auf die Wahl und den Einsatz der Medien im 4-Schritte-Modell.

Phase	Element	Medien	
		Medientyp	Präsentationsmedium
<i>SEE</i>	informative Einführung der Konzepte	Text/Grafik	Papier
<i>TRY</i>	Anleitung	Hypertext/Grafik	Bildschirm
	Selbstüberprüfung	Hypertext/Grafik	Bildschirm
	Anwendung/Werkzeug	Informatikmittel	Bildschirm
<i>DO</i>	Problemstellung	Text/Bild	Papier
	Anwendung/Werkzeug	Informatikmittel	Bildschirm
<i>EXPLAIN</i>	mündlicher Bericht	Informatikmittel	Bildschirm

Tab. 7.4: Im 4-Schritte-Modell eingesetzte Medien.

1. Medientypen

Im Folgenden sollen Vor- und Nachteile verschiedener Medientypen für den Einsatz im 4-Schritte-Modell besprochen werden.

Text. Mit dem Medientyp *Text* lassen sich sowohl Anweisungen als auch Konzepte konkret und präzise beschreiben. Im 4-Schritte-Modell kommt Text einerseits während der Phase *SEE* als Beschreibung der Konzepte und andererseits während der Phasen *TRY* und *DO* als Beschreibung der Problemstellung zum Einsatz. Während der Phase *TRY* wird Text zusätzlich zum Verfassen von Einzelschritten der Anleitung beim Lösen von Problemen eingesetzt (Abb. 7.4). Text fördert im Gegensatz zu selbstaktiven Medientypen (wie z.B. Audio und Video) die Selbststeuerung, weil er im eigenen Tempo gelesen und ausgeführt werden kann (siehe weiter unten).

Beim Verfassen von Instruktionstexten für die Phase *TRY* stellt sich die Frage, wie Text präsentiert werden muss, damit die Selbststeuerung unterstützt werden kann. Diese Frage wird im Rahmen des Kapitels 8.5.3 „Formulieren von Instruktionstexten“ im Detail diskutiert.

Grafik. Der Medientyp *Grafik* kommt im 4-Schritte-Modell während der Phasen *SEE* und *TRY* vor allem in Kombination mit Text zur Illustration der Konzepte zum Einsatz (Abb. 7.4 und Abb. 7.5, A). Bei der Phase *TRY* werden Grafiken als Teil der handlungsorientierten Anleitung zusätzlich als Mittel zur Überprüfung von Handlungsschritten

eingesetzt, indem der Zustand des zu bedienenden Informatikmittels nach dem Beenden eines Schrittes abgebildet wird (Abb. 7.4). Die Lernenden können zur Lösung eines Teilschritts eines Problems das Informatikmittel so lange bearbeiten und verändern, bis der abgebildete Soll-Zustand erreicht wird. Grafiken unterstützen auf diese Weise das selbstgesteuerte Handeln der Lernenden (siehe Kapitel 6 „Selbstgesteuerte, problem-basierte Instruktion“).

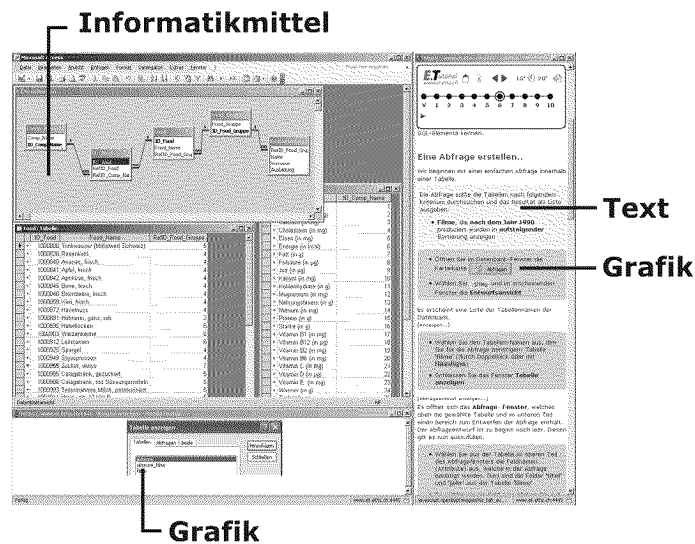


Abb. 7.4: Einsatz von Medien während der Instruktionsphase TRY zur Bedienung eines Informatikmittels im Anwendungsfenster (rechts) eines E.Tutorials®. Die Anleitung im Instruktionsfenster (links oben) enthält Text und Grafik. Als Überprüfungsmöglichkeit kommen im Überprüfungsfenster (links unten) Grafiken zum Einsatz.

Audio, Video, Animation. Zu Beginn der Entwicklung der E.Tutorials® wurden versuchsweise Videoanimationen eingesetzt. Die Lernenden hatten damals die Wahl zwischen statischen Elementen (Grafik und Text) und einer Video-Animation. Die Lernenden nutzten jedoch die Video-Animationen kaum, und sie haben sie in einem nachfolgenden E.Tutorial®, das keine Video-Instruktionen mehr enthielt, auch nicht vermisst (Abb. 7.5).

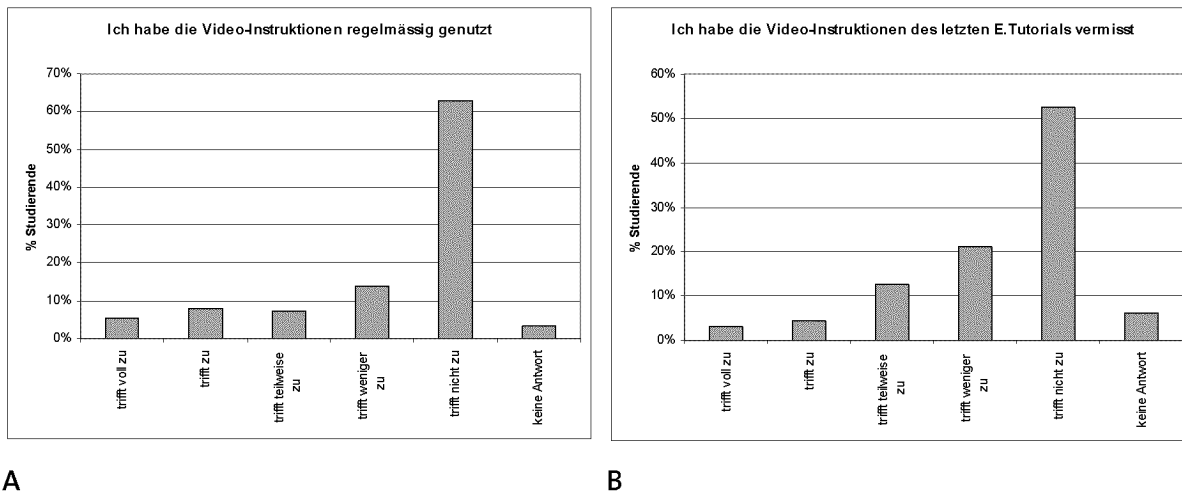


Abb. 7.5: Umfrage nach dem Einsatz zweier *E.Tutorials*[®] mit und ohne Video-Instruktionen im Überprüfungsfenster (siehe Abb. 7.4). Beim ersten *E.Tutorial*[®] hatten die Lernenden die freie Wahl zwischen Text-Instruktion und Video-Instruktion, beim zweiten wurde die Video-Instruktion weggelassen (ETH Zürich, Kurs 551-0839-00L „Einsatz von Informatikmitteln“, Wintersemester 2001/02, N=228).

Die Medientypen Audio und Video laufen im Gegensatz zu Text und Grafik in Echtzeit ab. Trotz der Dynamik dieser Medien eignen sie sich nicht für den Einsatz im 4-Schritte-Modell. Grund ist die Tatsache, dass die Lernenden ein Informatikmittel zur Lösung eines Problems selber bedienen und die Handlungen im eigenen Tempo ausgeführt werden müssen. Die Aktivität sollte deshalb nicht im Medium, sondern durch die Problemstellung und die damit verbundene Handlung der Lernenden selber erzeugt werden.

Audio, Video und Animationen geben nicht nur ein fremdes Tempo vor, es besteht auch die Gefahr der Informationsüberladung. Bei einer Video- oder Audio-Instruktion der auszuführenden Handlungsschritte haben die Lernenden Mühe, dem selbstaktiven Medium die erforderliche Information zur Ausführung eigener Handlungsschritte zu entnehmen, weil dabei auch jede Menge zusätzlicher, irrelevanter Information mitgezeigt wird. Die durch den problembasierten Ansatz geschaffene Aktivität geht durch den Einsatz selbstaktiver Medien wieder verloren.

2. Präsentationsmedien

Der Einsatz eines Computers im Unterricht muss nicht bedeuten, dass alle eingesetzten Medien in digitaler Form vorliegen und gezwungenermassen am Bildschirm präsentiert werden müssen. Für das Lesen von längeren Texten ist Papier als Medium (z.B. als gedrucktes Buch) nach wie vor dem Bildschirm überlegen. Als Gründe werden die Gewohnheit, schnellere Ermüdung und die langsamere Lesegeschwindigkeit am Bildschirm genannt [Nielson 1996, Schulmeister 1996]. Andere Untersuchungen zeigen, dass Internetseiten am Bildschirm meist nur gescannt und nicht Wort für Wort linear gelesen werden [Morkes & Nielson 1997]. Ein Indiz dafür könnte sein, dass längere Lehr-

texte am Bildschirm von den Lernenden häufig ausgedruckt werden. Die grosse Stärke des Computers ist sein Einsatz als Werkzeug und nicht die eines Präsentationsmediums.

Bei der Auswahl der Präsentationsmedien im 4-Schritte-Modell spielt neben diesen Argumenten vor allem die Tatsache eine Rolle, dass der grösste Teil der Bildschirmfläche bereits durch das zu bedienende Informatikmittel belegt ist. Es muss der aus ergonomischer Sicht ideale Mix aus analogen und digitalen Medien gefunden werden.

Im Folgenden soll der Einsatz von Papier und Bildschirm als Präsentationsmedien im 4-Schritte-Modell diskutiert werden.

Papier. Papier kommt im 4-Schritte-Modell als Präsentationsmedium von Text und Grafik während der Phasen *SEE* und *DO* zum Einsatz (Abb. 7.3). Diese längeren zusammenhängenden Texte lesen sich auf Papier nicht nur leichter als auf einem Bildschirm, sie können dadurch auch jederzeit zu Hand genommen und mit Hervorhebungen und Notizen versehen werden. Die Papierunterlagen der Phase *SEE* sind wie ein Stadtplan als Prozess-begleitendes Medium zu verstehen, das bei der Orientierung in einer fremden Welt helfen soll und mit zunehmender Unabhängigkeit immer mehr zur Seite gelegt werden kann. Die Begleitunterlagen helfen den Lernenden, während der praktischen Arbeit auf bestimmte Konzepte zu fokussieren und sie mit praktischen Erfahrungen zu verknüpfen.

Die sehr kurz gehaltene Darstellung der Konzepte der Phase *SEE* sollten dabei zu Beginn einmal komplett gelesen werden. Während der Arbeit am Computer in den Phasen *TRY* und *DO*, wo das praktische Anwenden der Konzepte im Zentrum steht, soll darauf jeder Zeit zurückgegriffen werden können. Papier hat zudem den Vorteil, dass es einfach mit persönlichen Notizen versehen werden kann. Papier wird im 4-Schritte-Modell also aus zwei Gründen eingesetzt, erstens, um die Arbeit am Computer-Bildschirm nicht durch weitere Fenster zu belegen, und zweitens, um die ergonomischen Vorteile des Mediums Papier zu nutzen.

Bildschirm. Die Hypertext-Technologie ermöglicht es, während der Phase *TRY* des 4-Schritte-Modells Anleitung, Werkzeug und Überprüfung nahe zueinander auf einen Bildschirm zu bringen (Abb. 7.4, folglich Kapitel 6 „Selbstgesteuerte, problembasierte Instruktion“). Die räumliche Nähe der drei Elemente des *E.Tutorials*[®] bringt durch die schrittweise Instruktion, sowie die direkte Ausführung und Überprüfung das Handeln der Lernenden in Fluss. Es wird ein Präsentationsmedium für Problemstellung, Bedienungsanleitung, Einführung der Konzepte und Umsetzung der Probleme geschaffen. Es stellt den Lernenden eine optimale handlungsbasierte Lernumgebung für den problembasierten Wissenserwerb auf einer Bildschirmfläche zur Verfügung. Arbeitsanweisungen und Konzepterklärungen können problemlos am Bildschirm gelesen

werden, falls die Texte kurz gefasst sind und das Lösen des Problems im Zentrum der Anleitung steht.

Die drei Fenster haben die Form von einzelnen HTML-Webseiten, die mit Hilfe von *JavaScript*-Code auf dem Bildschirm positioniert werden. Das zu bedienende Informatikmittel wird zusätzlich auf dem selben Bildschirm geöffnet und über dem entsprechenden Platzhalter (oben links) positioniert (Tab. 7.4 und Abb. 7.4).

Fenstername	Format	Medientyp
Anwendungsfenster	Abhängig vom eingesetzten Informatikmittel	Informatikmittel
Instruktionsfenster	Webseite (HTML)	vorwiegend Hypertext, einige Grafiken
Überprüfungsfenster	Webseite (HTML)	vorwiegend Grafiken, wenig Hypertext

Tab. 7.5: Medien und Funktion der drei Fenster des *E.Tutorials*®.

Die Anordnung der drei Fenster im *E.Tutorial*® wird von den Lernenden in Umfragen als sehr sinnvoll gewertet (Abb. 7.6).

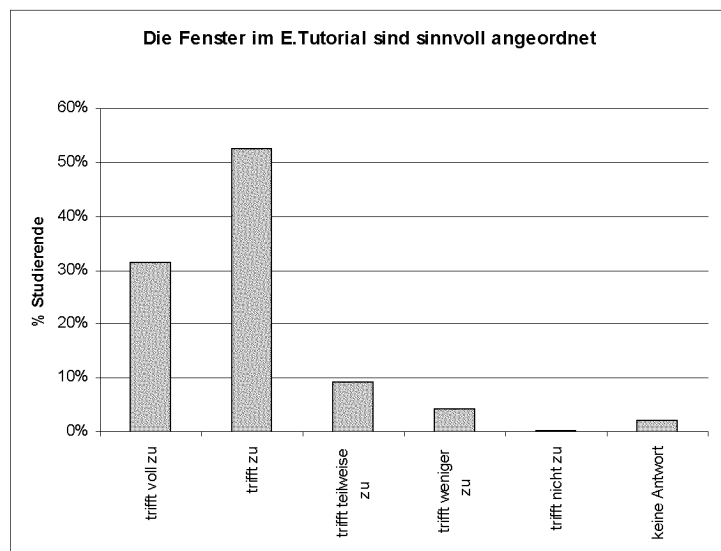


Abb. 7.6: Umfrage über die Anordnung der Fenster im *E.Tutorial*® (ETH Zürich, Kurs 551-0839-00L „Einsatz von Informatikmitteln“, Wintersemester 2001/02, N= 283).

7.2.4 Einsatz von Hyperlinks im *E.Tutorial*®

Hypertext bringt die Möglichkeit von Hyperlinks, Verknüpfungen, die einem Textelement oder einem grafischen Element hinterlegt sind und durch Anklicken aktiviert werden können. Durch diese Aktivierung springt die Leserin oder der Leser zu einem neuen Textteil. Es hat sich gezeigt, dass der Einsatz von Hyperlinks, obwohl er technisch einfach ist, funktionell und inhaltlich Sinn machen muss, damit er von den Lernenden genutzt wird.

Im Verlaufe der Entwicklung des 4-Schritte-Modells wurden Hyperlinks im Instruktionsfenster des *E.Tutorials*[®] der Phase TRY wie folgt eingesetzt (Abb. 7.7):

- 1. zur Navigation im *E.Tutorial*[®]
- 2. zum Anzeigen von Inhalt im Überprüfungsfenster
- 3. zum Anzeigen von Theorie und Glossar

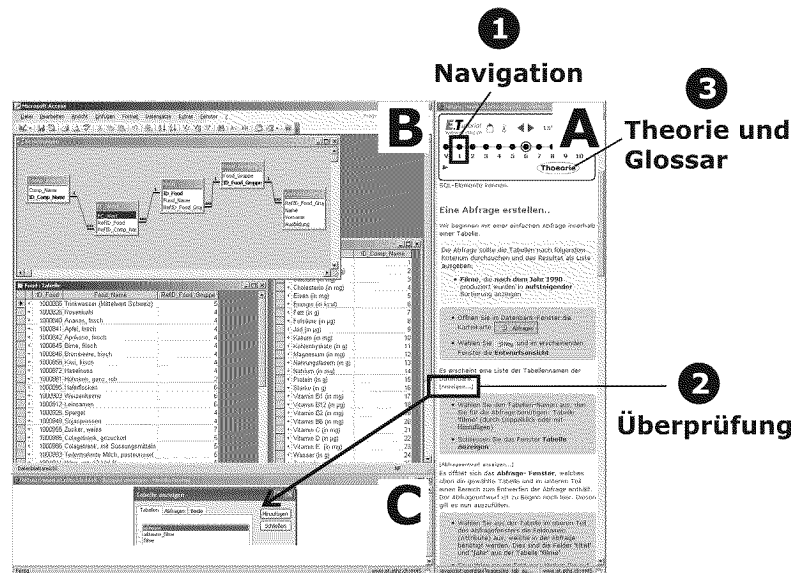


Abb. 7.7: Einsatz von Hyperlinks im *E.Tutorial*[®]: zur Navigation zwischen den einzelnen Lektionen des Instruktionsfensters (A) sowie zum Aufrufen von Überprüfungsgrafiken im Überprüfungsfenster (C). Links zu Glossar und Theorie aus dem Instruktionsfenster (A) haben sich nicht bewährt.

1. Navigation im *E.Tutorial*[®]

Der Inhalt des Instruktionsfensters des *E.Tutorials*[®] (Abb. 7.7, Fenster A) wird zur Gliederung des Unterrichts und zur besseren Übersichtlichkeit in verschiedene Lektionen aufgeteilt. Die einzelnen Lektionen werden entweder über einen Hyperlink am Ende einer Lektion („Weiter zur nächsten Lektion“) oder über das Navigationsmenü aufgerufen (Abb. 7.7, 1). Die Bedienung des *E.Tutorials*[®] wird von den Lernenden generell als intuitiv und einfach empfunden (siehe Abb. 7.8, A). Der jahrelange Einsatz hat gezeigt, dass dafür keinerlei Erklärungen oder Einführungen notwendig sind. Für das Installieren, Öffnen und Einrichten der Informatikmittel im Anwendungsfenster (Abb. 7.7, Fenster B), die oft auch zur Bedienung des *E.Tutorials*[®] gezählt werden, braucht es jedoch Hilfen und Erklärungen. Diese können direkt im Instruktionsfenster des *E.Tutorials*[®] (Abb. 7.7, Fenster A) erklärt werden.

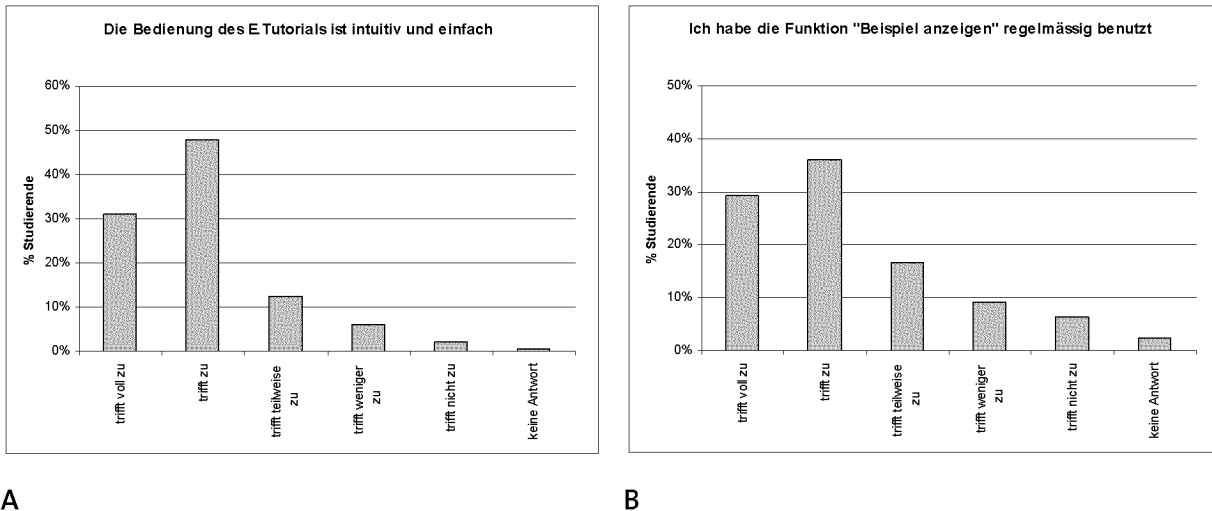


Abb. 7.8: Umfrage über die Nutzung der Überprüfungsmöglichkeit (ETH Zürich, Kurs 551-0839-00L „Einsatz von Informatikmitteln“, Wintersemester 2001/02, N= 283).

2. Anzeigen von Inhalt im Überprüfungsfenster

Mit Hilfe von Hyperlinks können im *E.Tutorial*[®] Hilfen und Überprüfungsgrafiken direkt aus der Anleitung heraus aufgerufen werden (Abb. 7.7, Fenster A → Fenster C). Diese wichtige Eigenschaft des *E.Tutorials*[®] ermöglicht ein Anzeigen des Soll-Zustandes eines Arbeitsschrittes im Überprüfungsfenster. Es hat sich gezeigt, dass diese Funktion von den Lernenden oft benutzt wird (Abb. 7.8, B). Sie dient den Lernenden dazu, sicherzustellen, ob ein Arbeitsschritt korrekt und vollständig umgesetzt wurde und verleiht zusätzlich Sicherheit und Erfolgserlebnisse (siehe auch Kapitel 6.4 „Elemente einer selbstgesteuerten, problembasierten Anleitung“). Da die Hilfe aufgerufen werden muss, stört sie diejenigen nicht, welche sie nicht benötigen.

3. Verlinkung mit Theorie und Glossar

Im Lauf der Entwicklung des *E.Tutorials*[®] wurde versucht, der praktischen Anleitung über Hyperlinks zusätzlich eine allgemeine Ebene zu integrieren. Dadurch sollte erreicht werden, dass die Lernenden von der Handlungsebene zurück auf die allgemeine theoretische Ebene finden. Dem Instruktionsfenster des *E.Tutorials*[®] (Abb. 7.7, Fenster A) wurden Hyperlinks integriert, die auf ein elektronisches Textbuch und ein Glossar verwiesen, in dem die wichtigsten Begriffe und Theorien nachgeschlagen werden konnten. Eine Umfrage aus dem Jahre 2001 hat ergeben, dass nur wenige Studierende diese Hyperlinks benutzt haben (Abb. 7.9, A).

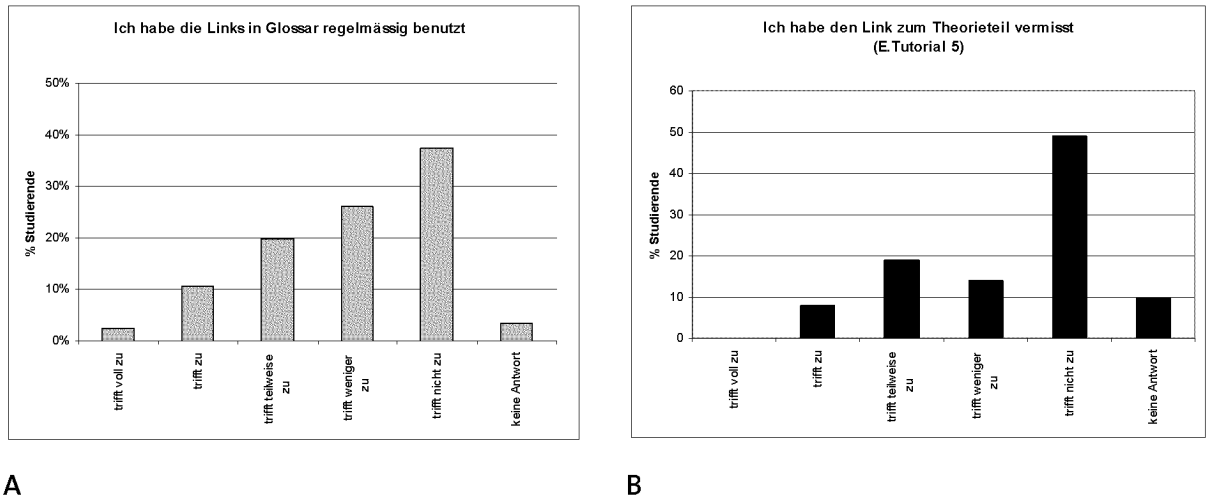


Abb. 7.9: Umfrage nach einem *E.Tutorial*[®] mit (A) und ohne (B) Verlinkung zu Theorie und Glossar. Beim *E.Tutorial*[®] in Abbildung A war die Theorie als Hyperlinks aus dem Instruktionsfenster heraus aufrufbar. Bei jenem in Abbildung B wurde auf die Hyperlinks verzichtet (ETH Zürich, Kurs 551-0839-00L „Einsatz von Informatikmitteln“, Wintersemester 2001/02, N=283 und Sommersemester 2002, N=128).

Ein Experte hat sich im Verlaufe der Zeit Wissensnetze mit einer assoziativen Struktur angeeignet (folglich Kapitel 2.2 „Generelle Merkmale einer Kompetenz“). Deshalb ist es nahe liegend, dass Lehrende Wissen als Netz und nicht als lineare Struktur präsentieren möchten. Hypertext bietet die technische Möglichkeit, Text von seiner linearen Struktur zu befreien. Die Hoffnung war gross, mit Hypertext ein Medium zu haben, um vernetzte Texte für vernetztes Denken zu Verfügung zu stellen. Man glaubte, wenn Lernende einen neuen Wissensgegenstand in einer vernetzten Umgebung kennen lernen, falle ihnen die Einordnung dieses Wissensgegenstandes in ihr eigenes vernetztes Wissen leichter. Es stellte sich die Frage, über welchen Weg Novizen neu angeeignetes Wissen vernetzen, wenn noch kein solches Netz vorhanden ist?

Gegen die Annahme, dass die vernetzte Wissenspräsentation zu vernetztem Wissen führt, spricht das Argument, dass Leserinnen und Leser von Hypertext sich ihre Leseleine in jedem Fall erst selber bahnen müssen. Durch die Navigation in Hypertext findet das Lesen ebenfalls in einer zeitlich sequentiellen Reihenfolge statt, so dass jeder faktische Weg letztlich doch wieder linear ist. Zudem wird häufig vergessen, dass durch Überblättern von gedrucktem Text ebenfalls die Möglichkeit besteht, die vorgegebene Sequenz zu durchbrechen. Gerade technische Anleitungen werden selten linear von vorne bis hinten gelesen.

Die Integration von vernetztem Wissen in ein Wissensnetz erweist sich als problematischer als die Integration linearen Wissens in das Wissensnetz. Dahinter steht die Annahme, dass zwei Netze schwieriger ineinander zu integrieren sind als eine lineare Struktur in ein bestehendes Netz [Kuhlen 1991].

Die Problemorientierung wurde in den Unterricht eingeführt, um die Aktivität der Lernenden zu erhöhen und einen Bezug zwischen der Theorie und der Praxis herzustellen.

Es hat sich gezeigt, dass das gezielte Verallgemeinern der Konzepte nicht durch die Schaffung einer Struktur auf der Ebene des Mediums (z.B. Hyperlinks zwischen Theorie und Praxis) alleine gelöst werden kann. Der Aufbau eines Wissensnetzes ist vielmehr mit einem Prozess verbunden und kann mit Hilfe des 4-Schritte-Modells ermöglicht werden (siehe Abschnitt 7.4).

7.3. Das Anwendungsebenen-Modell

Beim Anwendungsebenen-Modell wird ein längerfristiges Ziel verfolgt: die Entwicklung von Fähigkeiten und Kompetenzen, die sich aufgrund ihrer Komplexität erst über längere Zeit entwickeln. Um diese Kompetenzen erreichen zu können, muss eine ganze Reihe von Konzepten eingeführt und mit Wissen unterschiedlicher Dimensionen verknüpft werden (folglich Kapitel 2.3.2 „Verschiedene Arten von Wissen“ und Kapitel 3.2.3 „Computer-Fluency“). Das Anwendungsebenen-Modell organisiert den wiederholten Einsatz des 4-Schritte-Modells mit dem Ziel, komplexere Fähigkeiten und Kompetenzen aneignen zu können. Die zu lösenden Probleme des 4-Schritte-Modells werden dabei so aufeinander abgestimmt, dass eine übergeordnete Kompetenz erreicht werden kann (folglich Kapitel 5 „PBL“, Abb. 5.6).

In Abb. 7.10 ist der schlussendlich zu erreichende Kompetenzlevel als *Anwendungslevel B* dargestellt. Zur Lösung der Problemaufgabe dieses Kompetenzlevels (Abb. 7.10, P4) muss Wissen zur Anwendung gebracht werden, welches als Netz verschiedener miteinander in Beziehung stehender Konzepte zur Verfügung stehen muss. Um dieses Level mit Wissen unterschiedlicher Dimensionen zu erreichen, muss zunächst eine Auswahl von Konzepten zusammen mit Fertigkeiten isoliert auf dem *Anwendungslevel A* eingeübt werden (Abb. 7.10 und Tab. 7.5). Diese Einführung der Konzepte wird im 4-Schritte-Modell vorgenommen (siehe letzter Abschnitt). Auf diesem Level kommen kleinere Probleme (Abb. 7.10, P1 bis P3) zum Einsatz, welche die Konzepte isoliert repräsentieren und einführen (folglich unterschiedliche Typen von Problemaufgaben Kapitel 5.2.2). Wie viele Instruktionen auf dem Anwendungslevel A notwendig sind, bis eine Aufgabe auf dem Anwendungslevel B gelöst werden kann, ist von den Anforderungen des Levels B und der Komplexität der beteiligten Konzepte abhängig. Um beispielsweise ein arithmetisches Problem mit einer Tabellenkalkulation zu lösen, ist wahrscheinlich ein Instruktionsproblem ausreichend, während die Umsetzung eines Algorithmus in einem Programm die Einführung mehrerer Konzepte notwendig macht (siehe Umsetzung der Modelle im nächsten Kapitel). Beim Anwendungslevel B steht die Anwendung der selben Konzepte im Zentrum wie beim Anwendungslevel A, mit dem Unterschied, dass sie in diesem umfangreicheren Problem nicht mehr isoliert, sondern als Teil eines Problemlösungsprozesses auftreten. Dieser ist von den Lernenden nur dann zu bewältigen, wenn sie beim Lösen auf aktives, transferfähiges Wissen aus dem Anwendungslevel A zurückgreifen können.

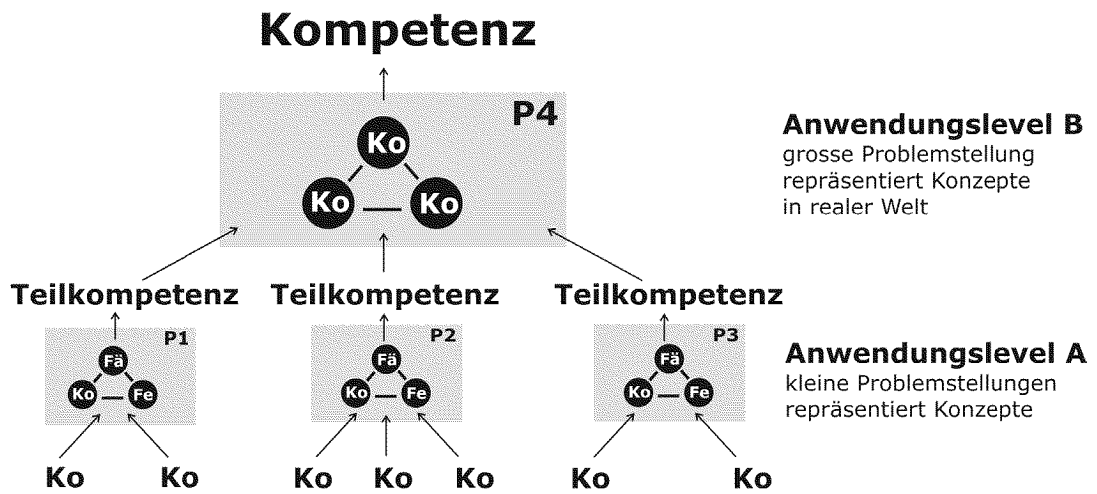


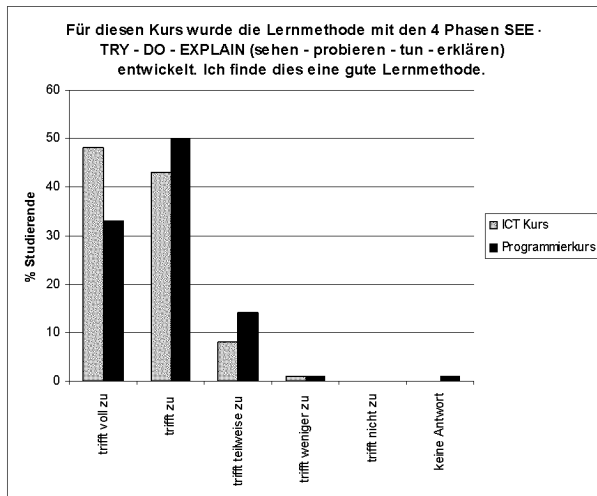
Abb. 7.10: Ablauf des Anwendungsebenen-Modells. Durch das Lösen einer Folge von kleineren Problemen (hier P1 bis P3) auf *Anwendungslevel A*, welche grundlegende Konzepte repräsentieren, wird das notwendige Wissen angeeignet, um eine umfangreiche Aufgabe der realen Welt (hier P4) auf *Anwendungslevel B* lösen zu können. Das Resultat ist die Aneignung von Teilkompetenzen, die in einer grösseren Aufgabe zu einer Kompetenz zusammengesetzt werden müssen. Abkürzungen: Ko: Konzepte, Fe: Fertigkeiten, Fä: Fähigkeiten.

	Konzepte	Aufgabetypp	Ziel
Anwendungslevel A	Einüben von einzelnen Konzepten zusammen mit Fertigkeiten	kleine Problem- aufgaben	Aufbau eines Wissens- netzes, Aneignung von transferfähigem Wissen
Anwendungslevel B	Anwenden mehrerer Konzepte in realer Umgebung	grosse Problem- aufgabe	Anwenden des erwor- benen Wissensnetzes

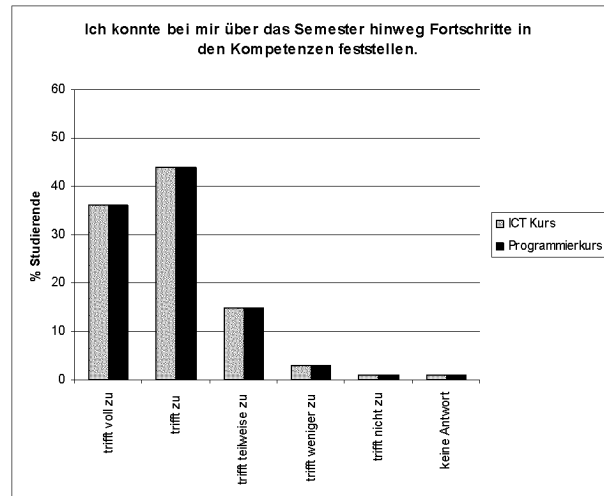
Tab. 7.6: Anwendungsebenen A und B des Anwendungsebenen-Modells.

7.4. Evaluation des didaktischen Modells

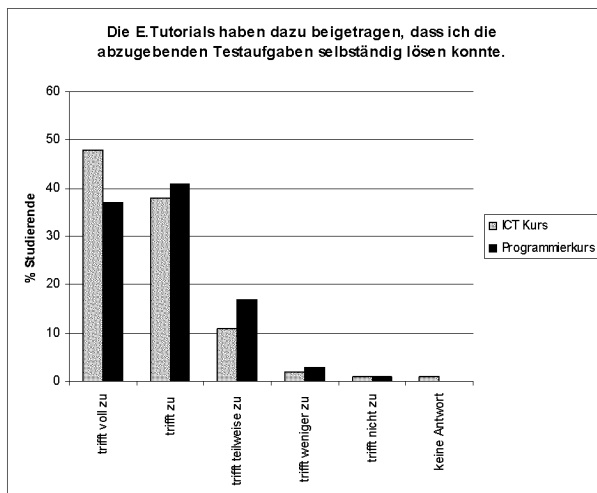
Das 4-Schritte-Modell wurde über mehrere Jahre als Teil der Unterrichtsevaluation in Form von Nachumfragen von den Studierenden beurteilt (siehe Kapitel 10 „Evaluation der Lernumgebungen“, 8.6 „Qualitäts-Feedback“ und 9.4.6 „Qualitätskontrolle eines Kurses“). Abb. 7.11 zeigt die Resultate zur Evaluation des 4-Schritte-Modells in den durchgeführten ICT- und Programmierkurse in den Jahren 2006 und 2007.



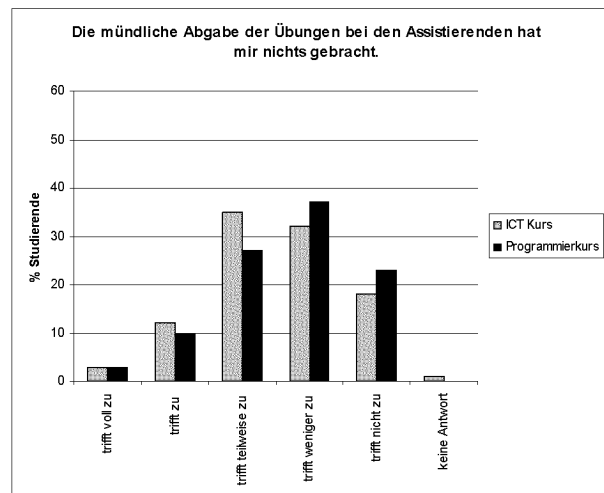
A



B



C



D

Abb. 7.11: Evaluation des didaktischen Modells in Nachumfragen unter Studierenden als Teil der Unterrichtsevaluation im Winter- und Sommersemester 2006 und 2007 (ICT-Kurse N= 215, Programmierkurse N=297).

Die Resultate zeigen, dass eine grosse Mehrheit der Studierenden (ICT Kurs 91%, Programmierkurs 83%), die mit einer der beiden Lernumgebungen unterrichtet wurden, das 4-Schritte-Modell als eine gute Lernmethode einstuft (Abb. 7.11, A). Jeweils 80% der Studierenden beider Kurse geben an, über das Semester hinweg Fortschritte in den Kompetenzen festgestellt zu haben (Abb. 7.11, B). Weiter finden 86% der Studierenden des ICT-Kurses und 78% des Programmierkurses, dass das *E.Tutorial*[®] dazu beigetragen hat, die abzugebende Testaufgabe selbständig lösen zu können (Abb. 7.11, C). 14% der Studierenden beider Kurse beantworteten die Aussage „die mündliche Abgabe der Übungen habe ihnen nichts gebracht“ mit „trifft voll zu“ oder „trifft zu“ (Abb. 7.11, D). 60% der Studierenden des ICT-Kurses und 51% der Studierenden des Programmierkurses gaben an, dies treffe „weniger“ oder „nicht zu“.

Zusammenfassend kann aus den durchgeführten Evaluationen gefolgert werden, dass der Unterricht im 4-Schritte-Modell bei den Studierenden nicht nur sehr beliebt ist, sie können auch deren Effektivität feststellen und sie beurteilen das *E.Tutorial*® als vorbereitendes Element für die darauf folgende selbständige Aufgabe. Die mündliche Abgabe bei den Assistierenden wird von einer Mehrheit der Studierenden ebenfalls als Teil des Lernprozesses betrachtet. Zwischen den beiden Lernumgebungen, die unterschiedliche Kompetenzen vermitteln, konnten keine nennenswerten Unterschiede festgestellt werden.

7.5. Fazit: Didaktische Modelle für den kompetenzorientierten Unterricht

Beim kompetenzorientierten Unterricht stehen didaktische Modelle im Zentrum, mit deren Hilfe ausgewählte Konzepte zu umfassenden Kompetenzen „umgewandelt“ werden können. Die Modelle legen die Aktivitäten der Lernenden sowie die Sequenzierung der Medien während sich wiederholenden Abläufen fest, die den Unterricht in Lerneinheiten oder Modulen organisieren (Abb. 7.12).

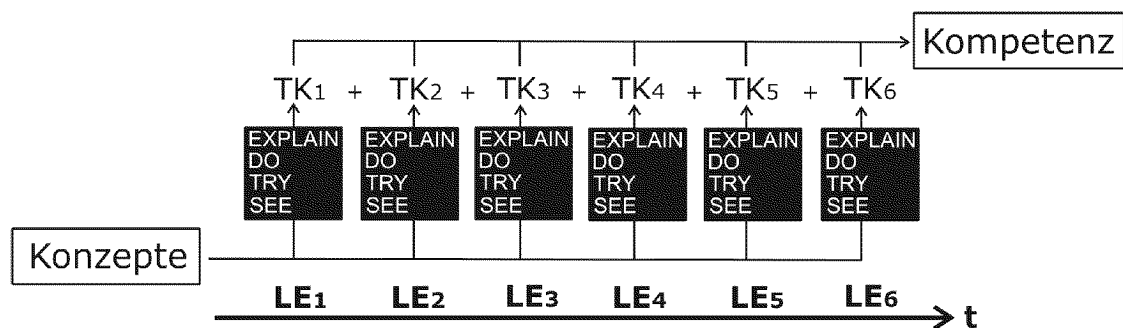


Abb. 7.12: Der Einsatz didaktischer Modelle führt zur aktiven Verarbeitung von Konzepten während Lerneinheiten (LE) und zum Aufbau von Teilkompetenzen (TK). Diese werden im Laufe eines Kurses zu Kompetenzen zusammengesetzt.

Während jeder Lerneinheit bringt das 4-Schritte-Modell eine Teilkompetenz hervor. Konzepte werden dadurch auf die Handlungsebene gebracht, verallgemeinert und mit einem Netz von Wissen unterschiedlicher Dimensionen ausdifferenziert. Die durch das 4-Schritte-Modell eingesetzten Medien und organisierten Aktivitäten der Lernenden ermöglichen eine Aneignung und Vernetzung von Wissen unterschiedlicher Dimensionen. Die Vernetzung geschieht schrittweise von der allgemeinen Konzeptebene über die Verknüpfung mit Fertigkeiten zur Entstehung von Fähigkeiten und Metakognition (siehe Abb. 7.2). Bei jedem Schritt kommt Wissen einer Dimension hinzu und bleibt dabei sowohl mit dem Konzept als auch mit individueller Erfahrung verknüpft. Das durch das 4-Schritte-Modell angeeignete Wissen ist daher nicht nur langlebiger, sondern auch flexibler in der Anwendung.

Der wichtige persönliche Kontakt zwischen den Lernenden und den Lehrenden wird beim 4-Schritte-Modell auf den letzten Schritt (*EXPLAIN*) beschränkt. Zu diesem Zeit-

punkt haben sich die Lernenden die notwendigen Wissensstrukturen bereits angeeignet, und es steht die Reflexion des Gelernten und ein Feedback durch eine Lehr- oder Assistenzperson im Zentrum. Welche Auswirkungen dies auf den Unterricht hat, ist dem Kapitel 8 „Einsatz der computergestützten Lernumgebung im Unterricht“ zu entnehmen.

Bei der Aneignung von anspruchsvolleren Konzepten, deren praktische Anwendung auch mit höheren Anforderungen an die zu leistende Kompetenz verbunden ist, kommt das 4-Schritte-Modell an seine Grenzen. In diesem Fall kommt das Anwendungsebenen-Modell zum Einsatz. Dabei bereitet eine Folge von problembasierten Konzept-einführungen nach dem 4-Schritte-Modell die Lernenden auf das Lösen der umfangreicheren Aufgabe vor (siehe Abb. 7.10). Durch den kombinierten Einsatz beider didaktischen Modelle kann ein Maximum an Wirksamkeit aus der Lernmethode des PBL herausgeholt und gleichzeitig die Zufriedenheit und Motivation der Lernenden gewährleistet werden.

Die detaillierte Umsetzung der beiden didaktischen Modelle zum Aufbau von ICT- und Programmierkompetenzen ist Inhalt des nächsten Kapitels.

Kapitel 8

Entwicklung und Umsetzung kompetenzorientierter Lerneinheiten

8.1. Einleitung

8.1.1 Überblick

In diesem Kapitel wenden wir uns der Umsetzung der im letzten Kapitel behandelten didaktischen Modelle zu. Nach einem Überblick über alle Entwicklungsprozesse zum Aufbau einer kompetenz- und problemorientierten Lernumgebung (Abschnitt 8.2) folgt eine detaillierte Umsetzung anhand eines Beispiels in Form einer Fallstudie aus dem Programmierunterricht für Naturwissenschaftlerinnen und Naturwissenschaftler (Abschnitte 8.3 bis 8.5).

8.1.2 Entwicklung von kompetenzorientierten Lerneinheiten

Im Zentrum der Entwicklung einer kompetenzorientierten Lernumgebung für den Unterricht im 4-Schritte-Modell steht die Gestaltung von Unterrichts- oder Lerneinheiten, bestehend aus einer problembasierten Instruktion, gefolgt von einer Anwendungsphase (siehe Kapitel 7 „Didaktische Modelle“, Tab. 7.1 und Abb. 5.8). Damit die problembasierte Instruktion und Anwendung innerhalb der Lernumgebung gut zusammenspielen und ein Kompetenzaufbau erreicht werden kann, hat sich die Entwicklung nahe am Unterricht nach folgendem Muster bewährt (Abb. 8.1):

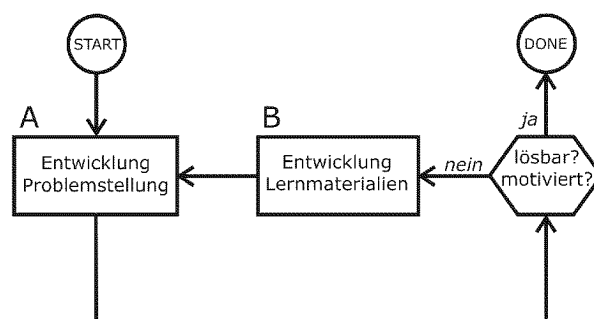


Abb. 8.1: Entwicklung einer kompetenz- und problemorientierten Lernumgebung. Ziel der Entwicklung ist es, Lernmaterialien in Form von problembasierten Instruktionen (B) zur Verfügung zu stellen, welche die Studierenden auf das Lösen von anspruchsvollen Problemaufgaben (A) vorbereiten. Die auf diese Weise erzielten Erfolgserlebnisse führen zu Zufriedenheit und Motivation der Lernenden.

Ausgangslage für die Umsetzung ist eine anspruchsvolle und authentische Problemstellung (Abb. 8.1, A), deren Lösung Anforderungen auf einem bestimmten Kompetenzniveau erfordert (folglich Kapitel 5.2 „Aufbau von Kompetenzen durch PBL“). Für einen erheblichen Anteil der Studierenden ist eine solche Problemaufgabe ohne Hilfestellung nicht lösbar, weil die zur Lösung notwendigen Wissensstrukturen noch nicht verfügbar sind. Weil sie den Anforderungen der Problemaufgabe nicht gewachsen sind, führt dies zu Unzufriedenheit und Motivationsproblemen (folglich Abb. 5.7 B, Kapitel 5.3 „Motivationale Faktoren beim PBL“). Ziel der Entwicklung der Lernumgebung ist es deshalb, mit Hilfe der Umsetzung der beiden didaktischen Modelle (4-Schritte-Modell und Anwendungsebenen-Modell) die notwendigen Lernprozess-unterstützenden Strukturen in Form von Lernmaterialien (Abb. 8.1, B) zu schaffen, die eine *selbständige* Lösung der Problemaufgabe (Abb. 8.1, A) für alle Studierenden ermöglichen.

Durch den wiederholten produktiven Einsatz der Lernumgebung im Unterricht können die Lernmaterialien von Einsatz zu Einsatz verbessert werden. Die Problemaufgabe wird auf diese Weise für jene Studierendengenerationen besser und selbständiger lösbar. Nach Abschluss der Entwicklung der kompetenzorientierten Lernumgebung werden die Lernenden durch die entwickelten Lernmaterialien der problembasierten Instruktion auf das Lösen einer anspruchsvolleren Problemaufgabe auf einem definierten Kompetenzniveau vorbereitet. Weil die Studierenden nun den hohen Anforderungen der Problemaufgabe gewachsen sind, führt dies zu Erfolgserlebnissen und zu einer Verstärkung der Motivation, was sich positiv auf den weiteren Lernprozess auswirkt (folglich Abb. 5.7 A).

8.2. Entwicklungsprozesse

Die Entwicklung von Unterrichts- oder Lerneinheiten für eine kompetenz- und problemorientierte Lernumgebung findet auf folgenden Ebenen statt:

- Kompetenz
- Teilkompetenz
- Unterrichts- oder Lerneinheit

Die Entwicklungsebenen entsprechen den Phasen des Aufbaus von Kompetenzen während verschiedener Zeiträume im Unterricht (siehe Kapitel 7 „Didaktische Modelle“, Abb. 7.1 und Tab. 7.1).

Abb. 8.2 und Tab. 8.1 geben einen Überblick über alle an der Entwicklung beteiligten Prozesse und Teilprozesse. Die drei Hauptprozesse bei der Entwicklung sind:

- die Konstruktion von anspruchsvollen, authentischen Problemaufgaben, die einen gewünschten Kompetenzlevel repräsentieren (Prozesse A und B)

- die Entwicklung der Lernmaterialien für die Unterrichts- oder Lerneinheit(en) (Prozess C)
- der produktive Einsatz der Lernumgebung im Unterricht und die Verbesserung der Lernmaterialien (Prozess D)

Weil der produktive Einsatz und die Qualitätsverbesserung der Lernumgebung Teil des Entwicklungsprozesses sind, sind meistens mehrere Durchläufe erforderlich, bis sich die Unterrichtseinheiten als stabil und fehlerfrei erweisen (folglich Abb. 8.1).

Zunächst soll ein Überblick über alle Hauptentwicklungsprozesse der Lernumgebung gegeben werden.

8.2.1 Überblick über die Entwicklungsprozesse

1. Ebene Kompetenz: Kompetenzorientierte Problemkonstruktion (Prozess A)

Startpunkt zur Entwicklung einer kompetenz- und problemorientierten Lernumgebung ist das Formulieren einer Problemstellung, die eine gewünschte Kompetenz repräsentiert (Abb. 8.2, START). Das Ziel der Entwicklung ist, dass die Kompetenz durch den Einsatz der Lernumgebung im Unterricht generiert wird (folglich Kapitel 5 „PBL“). Die Problemstellung sollte für die Zielgruppe relevant und interessant sein und in einem möglichst realistischen Kontext stattfinden. Für diesen Schritt ist eine ausführliche Kompetenz- und Zielgruppenanalyse durchzuführen und ein geeignetes Informatikmittel zur Lösung des Problems auszuwählen. Die Entwicklung der Problemstellung ist auch Endpunkt der Gesamtentwicklung der Lernumgebung, die abgeschlossen ist, sobald die Studierenden diese Problemstellung mit den bereitgestellten Lernmaterialien selbständig lösen können und motiviert und zufrieden sind (Abb. 8.2, DONE).

2. Ebene Teilkompetenz: Umsetzung des Anwendungsebenen-Modells (Prozess B)

Das Anwendungsebenen-Modell kommt dann zum Einsatz, wenn zum Erreichen der Kompetenz zum Lösen der Problemstellung aus Prozess A die Aneignung von Teilkompetenzen notwendig ist. In diesem Fall muss für jede Teilkompetenz eine Problemaufgabe konstruiert werden, die jeweils eine Unterrichtseinheit im 4-Schritte-Modell umfasst (siehe Kapitel 7 „Didaktische Modelle“ und Umsetzungsbeispiel in Abschnitt 8.4.1).

3. Ebene Lerneinheit: Umsetzung des 4-Schritte-Modells (Prozess C)

Die durch die obigen Entwicklungsprozesse konstruierten Problemaufgaben bilden die Ausgangslage zur Umsetzung einer Unterrichtseinheit nach dem 4-Schritte-Modell (siehe Kapitel 7 „Didaktische Modelle“). Die Formulierung der Aufgabenstellung (Phase DO) liefert den Lernenden alle Informationen und Anforderungen, damit die Aufgabe

gelöst werden kann. Der zur Lösung der Aufgabe erforderliche Kompetenzerwerb findet durch die Lernmaterialien der Phasen *SEE* und *TRY* statt. Die zur Lösung der Aufgabe anzueignenden Konzepte werden einerseits als Einführung in den Begleitunterlagen (Phase *SEE*) formuliert und dienen andererseits als Grundlage zur Formulierung des Instruktionstextes des *E.Tutorials*[®] (Phase *TRY*). Die im *E.Tutorial*[®] zu lösende Aufgabe wird aus den zu vermittelnden Konzepten konstruiert (Problembasierter Wissenserwerb, Kapitel 5.2.2). Die Formulierung der Anleitung des *E.Tutorials*[®] integriert Konzepte, Problemstellung und Handlungsschritte für die problembasierte Instruktion (folglich Kapitel 6 „Selbstgesteuerte, problembasierte Instruktion“).

4. Ebene Feedback: Produktiver Einsatz und Qualitätsfeedback (Prozess D)

Es hat sich bei der Entwicklung von Unterrichtseinheiten bewährt, die Lernmaterialien des 4-Schritte-Modells bereits während ihrer Entwicklung im Unterricht einzusetzen. Die Qualitätsverbesserung der Lerneinheiten geschieht auf mehreren Ebenen. Die Grundlage dazu bildet eine Reihe von Feedback-Informationen, die zur Verbesserung der Unterrichtseinheit auf verschiedenen Stufen führen. Dazu gehören:

Lokal:

- Feedback-Möglichkeiten im Anschluss an die *E.Tutorials*[®]
- die mündliche Abgabe der Phase *EXPLAIN* des 4-Schritte-Modells

Global:

- Daten der Leistungskontrolle
- Daten der Prozessanalyse während des Unterrichts

Auf der Stufe „Unterrichtseinheit“ dienen die Feedback-Informationen dazu, den Inhalt der Lernmaterialien des 4-Schritte-Modells (Konzepte, Anleitung, Problemstellung) anzupassen und zu verbessern. Auf den Stufen „Kompetenzen“ und „Teilkompetenzen“ fließt die Feedback-Information zurück in die Problemkonstruktion, die ebenfalls optimiert und angepasst werden kann.

Bez.	Entwicklungsprozesse	Hilfsmittel	Output
Ebene Kompetenz			
A	Problemkonstruktion	Webbasierte Umfrageapplikation, Lehrbetriebsapplikation für Dozierende	Problemstellung
A1	Festlegen des Zeitgefässes im Unterricht		
A2	Zielgruppen-Analyse		
A3	Kompetenz-Analyse		
A4	Auswahl Informatikmittel		
Ebene Teilkompetenz			
B	Problemkonstruktion		Problemstellungen
Ebene Unterrichtseinheit			
C	Erstellen der Lernmaterialien		Lernmaterialien
C1	Formulieren der Aufgabenstellung	<i>E.Template</i>	Aufgabenstellung
C2	Formulieren der Begleitunterlagen		Begleitunterlagen
C3	Problembasierte Instruktion		<i>E.Tutorial®</i>
C3.1	Problemkonstruktion		
C3.2	Formulieren des Instruktionstexts		
C3.3	Integration der Konzepte und Erläuterungen		
C3.4	Überprüfungsmöglichkeiten schaffen		
Ebene Qualitäts-Feedback			
D	Qualitäts-Feedback		Output-Kontrolle
D1	Verbesserung der Lernmaterialien	Online-Feedback-Möglichkeiten	Qualitätsverbesserung der Lernmaterialien
D2	Kompetenzkontrolle	Phase <i>EXPLAIN</i> , Resultate der Leistungskontrolle	Qualitätsverbesserung der Unterrichtseinheit
D3	Implementierung der Unterrichtseinheit in den Gesamtkurs	Prozessanalyse	Qualitätsverbesserung von Unterrichtseinheit und Kurs

Tab. 8.1: Liste der Entwicklungsprozesse einer kompetenz- und problemorientierten Lernumgebung.

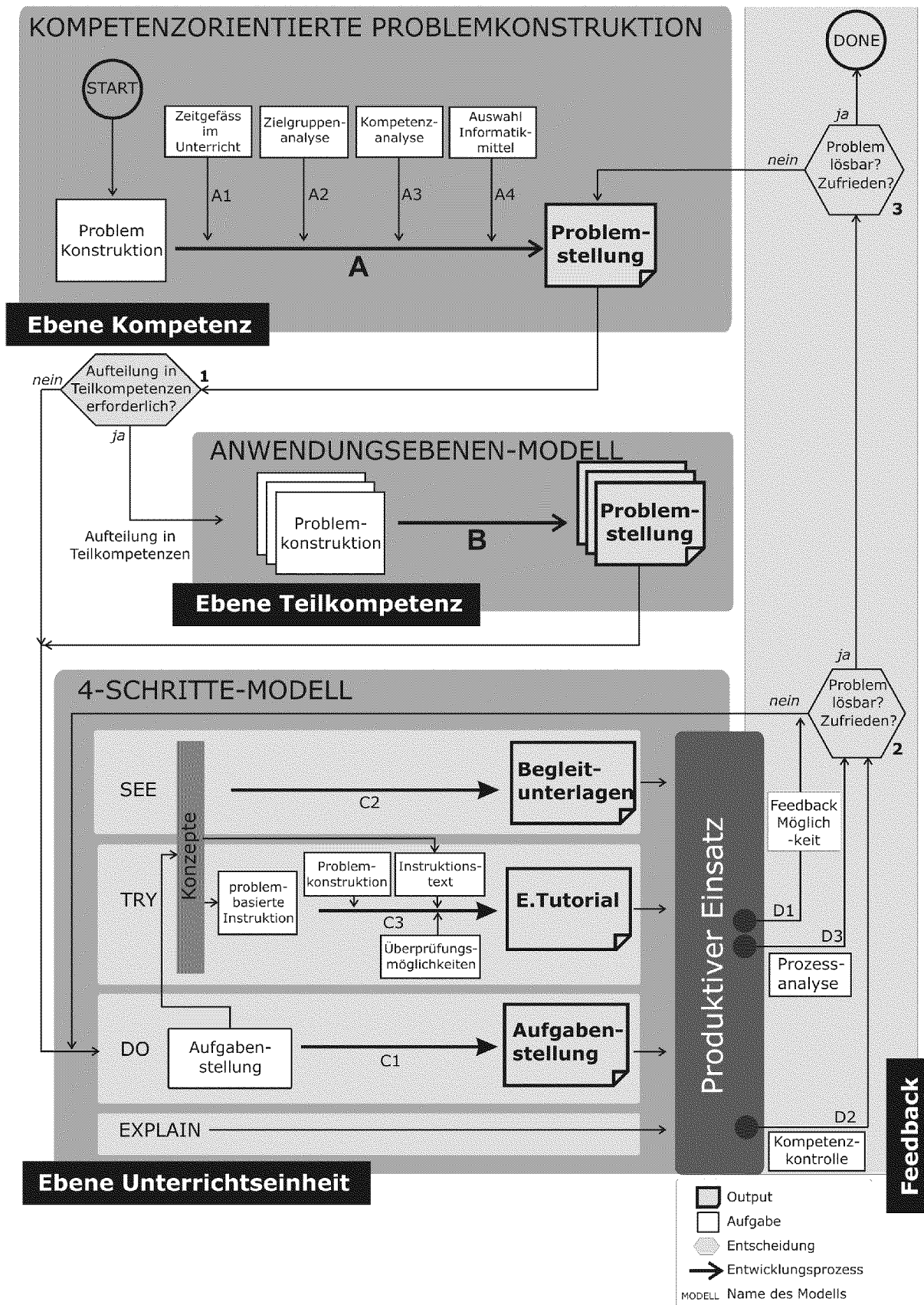


Abb. 8.2: Schematische Darstellung der Abläufe der Entwicklungsprozesse einer kompetenz- und problemorientierten Lernumgebung. Für die verschiedenen Bezeichnungen der einzelnen Prozesse (A, B, C etc.) siehe Tab. 8.1.

8.2.2 Übersicht über die eingesetzten Hilfsmittel

Zur Umsetzung der Entwicklungsprozesse kommen verschiedene Hilfsmittel zum Einsatz (siehe Tab. 8.1 und Tab. 8.2):

- Entwicklungsumgebung *E.Template*, bestehend aus:
 - Dokumentenvorlagen für die Begleitunterlagen der Phase *SEE*
 - programmierte Vorlage eines *E.Tutorials*[®]
 - *E.Tutorial*[®] *HowTo*
- Elektronisches Assessmentsystem
- Prozessanalyse
- Webbasierte Umfrageapplikation *Survey*
- Lehrbetriebsapplikation für Dozierende der ETH Zürich *eDoz*

Hilfsmittel	Name des Hilfsmittels	Quelle
Entwicklungsumgebung	<i>E.Template</i>	http://et.ethz.ch
Elektronisches Assessmentsystem		http://et.ethz.ch
Prozessanalyse		Teil des <i>E.Template</i>
Webbasierte Umfrageapplikation	<i>Survey</i>	http://elba.ethz.ch/services/survey_application
Lehrbetriebsapplikation für Dozierende der ETH Zürich	<i>eDoz</i>	http://edoz.ethz.ch

Tab. 8.2: Übersicht über die eingesetzten Hilfsmittel.

Weitere Details zum Einsatz der Hilfsmittel folgen an entsprechender Stelle bei der Umsetzung der Entwicklungsprozesse in den Abschnitten 8.3 bis 8.6.

8.2.3 Fallstudie: Umsetzung der didaktischen Modelle an einem Beispiel

In den folgenden Abschnitten dieses Kapitels soll die Umsetzung der didaktischen Modelle am Beispiel eines Programmierkurses für Naturwissenschaftlerinnen und Naturwissenschaftler gezeigt werden. Der Kurs trägt den Titel „Anwendungsnahes Programmieren“ (ETH Zürich, Kurs 251-0840-00L, Tab. 1.2). Die Entwicklung dieses Kurses begann im Jahre 2002, wurde 2005 erstmals als problembasierter Kurs im Blended Learning gehalten und im Jahre 2006 abgeschlossen (Tab. 8.3). Um die Durchführung

des Kurses auch während der Entwicklung zu gewährleisten, wurde der Fokus in jedem Jahr auf einen Entwicklungsschritt gerichtet. Im Jahre 2004 wurde kein Kurs durchgeführt, weil er im Rahmen einer Umstrukturierung im Studienplan vom 2. ins 4. Semester verschoben werden musste. Ab 2005 wurde das entwickelte Kursmaterial in einer weiteren Vorlesung mit dem Titel „Informatik 2 für Bauingenieure“ (ETH Zürich, Kurs 251-0846-00L, Tab. 1.2) eingesetzt.

Jahr	Kurseigenschaften	Form	Ziele / Fokus der Entwicklung
2002	1. Durchführung des Kurses 186 Studierende	traditionell (<i>Vorlesung, Übungen</i>)	Problemkonstruktion 1 Auswahl der Informatikmittel Kompetenz-Analyse
2003	2. Durchführung des Kurses 225 Studierende	traditionell (<i>Vorlesung, Übungen</i>)	Problemkonstruktion 2 Aufgabenstellung Einsatz eines ersten <i>E.Tutorials</i> [®] Zielgruppen-Analyse
2004	Keine Durchführung des Kurses	-	Umsetzung des Anwendungsebenen- und 4-Schritte-Modells
2005	3. Durchführung des Kurses 292 Studierende*	Blended Learning (<i>PBL, Vorlesung</i>)	Einsatz aller <i>E.Tutorials</i> [®] Qualitäts-Feedback Assessment
2006	4. Durchführung des Kurses 326 Studierende*	Blended Learning (<i>PBL, Vorlesung</i>)	Begleitunterlagen Verbesserung und Abschluss der Entwicklung

Tab. 8.3: Zeitlicher Verlauf der Entwicklung des Kurses „Anwendungsnahes Programmieren“ von 2002 (Beginn) bis 2006 (Ende). *inklusive Studierende der Vorlesung „Informatik 2 für Bauingenieure“.

8.3. Prozess A: Kompetenzorientierte Problemkonstruktion

Der Prozess der kompetenzorientierten Problemkonstruktion ist mit einer ganzen Reihe von Vorabklärungen verbunden, die sich aus mehreren Teilprozessen zusammensetzen (Prozesse A1 bis A4, Abb. 8.2, Tab. 8.1). Die damit verbundenen Analysen wurden über mehrere Jahre durchgeführt und ausgewertet. Die im Folgenden präsentierten Resultate sind über die gesamte Entwicklungszeit gesammelt worden. Weitere Details zur Konstruktion und Auswahl von Problemaufgaben sind dem Kapitel 5 „PBL“ zu entnehmen.

8.3.1 Prozess A1: Festlegen des Zeitgefässes im Unterricht

Den Studierenden werden für den Kurs 2 ECTS-Kreditpunkte gutgeschrieben. Das bedeutet, dass der totale Arbeitsaufwand in etwa 50-60 Stunden betragen sollte [Europäische Kommission 2004]. Gemäss Studienplan und Vorlesungsverzeichnis besteht der Kurs aus 2 Vorlesungsstunden und 2 Übungsstunden. Weil der problembasierte

Kurs für die Studierenden einen höheren Arbeitsaufwand bedeutet und die Vermittlung der Grundlagen nicht mehr in der Vorlesung stattfindet, wurde die Vorlesung auf die Hälfte reduziert und findet nur noch alle zwei Wochen statt.

8.3.2 Prozess A2: Zielgruppen-Analyse

Bei der Zielgruppen-Analyse steht die Abklärung folgender Fragen im Zentrum:

- 1. Die Zusammensetzung der Zielgruppe und Lernziele des Studiengangs
- 2. Die Vorpraxis und Gewohnheiten der Lernenden im Umgang mit dem Computer
- 3. Die subjektive Einschätzung zu Kompetenz, Motivation und Nützlichkeit des zu erwerbenden Wissens durch die Lernenden

Hilfsmittel:

Um diese Fragen zu klären, wurden Vorumfragen unter den Studierenden durchgeführt. Als Hilfsmittel für die Zielgruppen-Analyse wurde eine elektronische Umfrageapplikation und die offizielle elektronische Studierendenverwaltung für Dozierende der ETH Zürich eingesetzt (siehe Tab. 8.2).

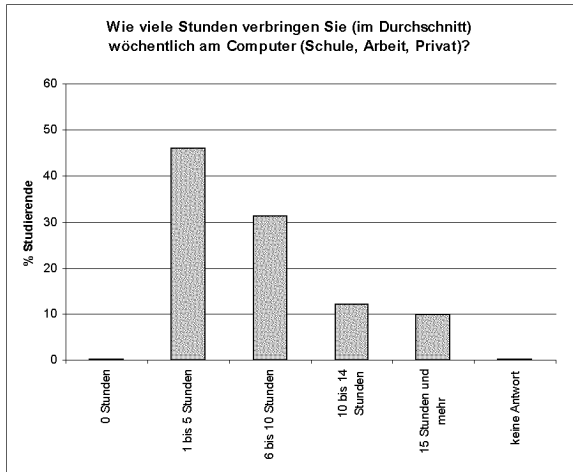
1. Zusammensetzung der Zielgruppe und Lernziele des Studiengangs

Der Kurs „Anwendungsnahes Programmieren“ kann von den Studierenden diverser naturwissenschaftlicher Fachrichtungen im Rahmen des Bachelor-Studiums als Grundlagenfach gewählt werden (siehe Tab. 1.2). Der Kurs wurde in den Jahren 2003 bis 2006 jährlich durchschnittlich von 158 Studierenden besucht. Die Studierenden kamen in den Jahren 2003 bis 2006 aus folgenden Fachrichtungen: Umweltnaturwissenschaften (58%), Agrar- und Lebensmittelwissenschaften (30%), Erdwissenschaften (10%) und diverse Fachrichtungen (2%). Der Frauenanteil lag von 2003 bis 2006 bei durchschnittlich 52%. Die Studierenden gaben folgende Sprachen als ihre Hauptsprache an: Deutsch (85%), Italienisch (7%), Französisch (3%), Rätoromanisch (2%), andere (3%).

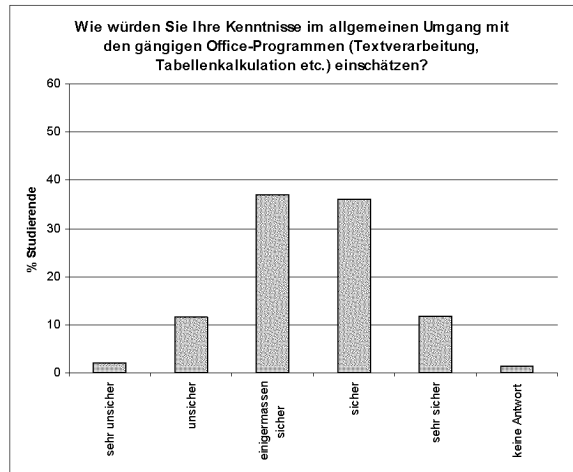
2. Vorpraxis und Gewohnheiten im Umgang mit dem Computer

Die Studierenden dieses Kurses besuchen bereits im 1. Semester einen Informatikkurs (ETH Zürich, Kurs 551-0432-00L, „Einsatz von Informatikmitteln“, Tab. 1.2). Der grundlegende Umgang mit einem Informationsarbeitsplatz kann somit vorausgesetzt werden (siehe Kapitel 3 „ICT-Kompetenzen“). Durchschnittlich haben praktisch alle Studierenden einen eigenen Computer. 70% haben einen eigenen Laptop zur Verfügung (siehe Kapitel 3 „ICT-Kompetenzen“, Abb. 3.6). 46% der Studierenden geben an, wöchentlich 1 bis 5 Stunden am Computer zu verbringen (Abb. 8.3, A). Bei 22% sind es mehr als 10 Stunden. 48% der Studierenden schätzen ihre Kenntnisse im allgemeinen

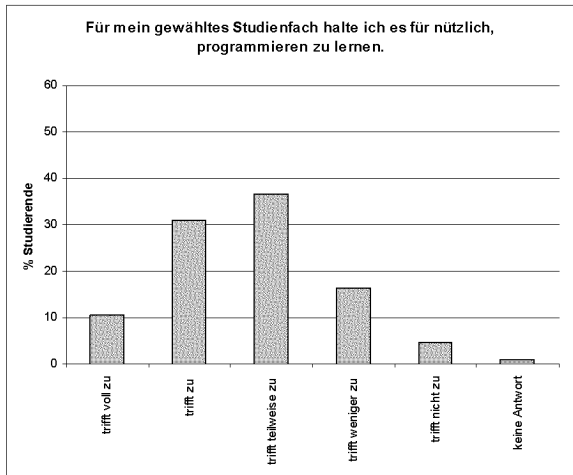
Umgang mit gängigen Office-Programmen als sicher bis sehr sicher ein, während 13% angeben, damit unsicher oder sehr unsicher zu sein (Abb. 8.3, B).



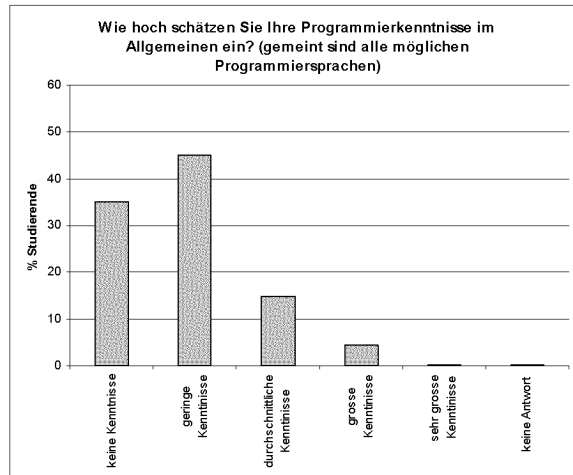
A



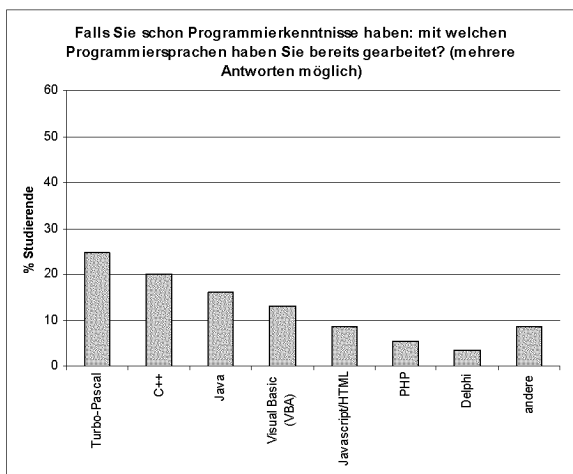
B



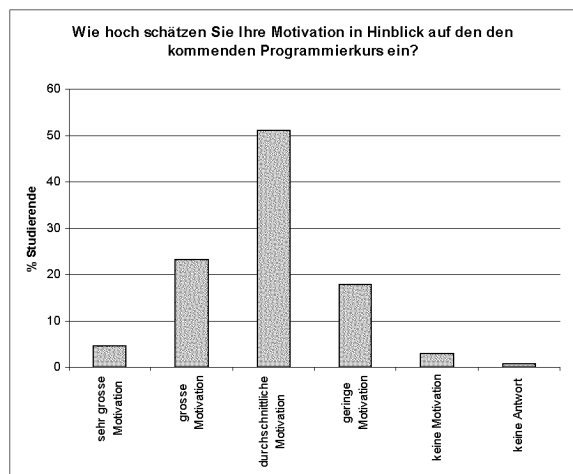
C



D



E



F

Abb. 8.3: Umfrage unter der Zielgruppe der Vorlesung „Anwendungsnahes Programmieren“ in den Jahren 2003 bis 2006 (N=304).

3. Subjektive Einschätzung zu Kompetenz, Motivation und Nützlichkeit des Wissens

In den Vorumfragen von 2003 bis 2006 gaben 41% der Studierenden an, dass sie es für ihr Studienfach für nützlich halten, programmieren zu lernen (Abb. 8.3, C). 21% hielten diese Kompetenz für ihren Studiengang für weniger nützlich. 35% der Befragten gaben an, keine Programmierkenntnisse zu haben. 45% der Studierenden schätzen ihre Programmierkenntnisse als gering und 4% als gross bis sehr gross ein (Abb. 8.3, D). Die Hälfte der Befragten (51%) schätzte die eigene Motivation für den Kurs als durchschnittlich ein, 28% überdurchschnittlich und 21% unterdurchschnittlich (Abb. 8.3, F).

Fazit der Zielgruppen-Analyse

Nach den Resultaten der Vorumfragen kann davon ausgegangen werden, dass praktisch alle Studierenden mit persönlichen Computern ausgerüstet sind. Sowohl eine Grundmotivation als auch eine generelle Sicherheit im Umgang mit dem Computer ist für den Programmierkurs vorhanden. Ein grosser Teil der Studierenden schätzt Programmierkompetenzen für ihren Studiengang als sinnvoll ein. Die Grundlage für diese Einstellung wurde einerseits durch einen Einführungskurs im 1. Semester geschaffen und kommt andererseits durch die Verschiebung des Kurses ins 4. Semester zu Stande. Da nur ein kleiner Teil der Studierenden über eigenständige Programmiererfahrungen verfügt (Abb. 8.3, D), kann nur bei wenigen Vorwissen vorausgesetzt werden.

8.3.3 Prozess A3: Kompetenz-Analyse

Für unsere Zielgruppe steht nicht das Programmieren neuer Software im Zentrum, sondern das Erweitern und Anpassen von bestehenden Programmen an ihre individuellen Bedürfnisse im Informationsarbeitsplatz (folglich Kapitel 2 „ICT-Kompetenzen“). Diese Kompetenz eignen sich die Lernenden am besten in einer möglichst authentischen Umgebung an. Das bedeutet, dass die Lernenden mit Beispielen und einer Programmiersprache arbeiten, die sie auch in der Realität antreffen. Da die meisten Studierenden nur wenig oder noch gar keine Programmierkenntnisse haben, steht zunächst die Aneignung von Programmiergrundlagen im Zentrum. Dazu gehört das Erlernen einer Auswahl von Grundkonzepten und Fertigkeiten einer Programmiersprache. Dies geschieht am besten dadurch, dass die Lernenden selber kleine lauffähige Programme schreiben. Die Grösse der Programme soll dann im Verlaufe des Kurses zunehmen, indem der Fokus schrittweise vom Verständnis der Syntax zur Entwicklung eines Algorithmus bis zum Umgang mit grösseren Programmen ausgedehnt wird (siehe Tab. 8.5).

8.3.4 Prozess A4: Auswahl der Informatikmittel

Die Wahl eines Informatikmittels für den Einsatz im Informationsarbeitsplatz ist von einer ganzen Reihe von Kriterien abhängig, die gegeneinander abgewogen werden müssen (Tab. 8.4). Obwohl die Wahl des Informatikmittels wichtig ist, darf dieser Punkt

nicht überschätzt werden. Im Zentrum steht die Lösung einer Problemstellung mittels Anwendung von Konzepten. Oft lohnt es sich, mit einem einfacheren Informatikmittel zu beginnen, es mit zunehmender Komplexität des Problems und der beteiligten Konzepte an seine Grenzen zu bringen, um dann auf ein umfangreicheres Informatikmittel umzusteigen.

Auswahlkriterien für das Informatikmittel	Erklärung
1. generelle Verbreitung	Ist das Informatikmittel weit verbreitet, so dass es auch in ein paar Jahren noch auf dem Markt sein wird?
2. Plattformunabhängigkeit	Kann das Informatikmittel auf allen gängigen Plattformen verwendet werden?
3. Installation in der Infrastruktur der Schule	Ist das Informatikmittel in den Computerräumen der Schule und auf den für Studierende verkauften Laptops installiert?
4. Kosten	Ist das Informatikmittel für Studierende günstig oder gratis erhältlich?
5. zielgruppenspezifische Verbreitung	Welches Informatikmittel ist bei dieser Zielgruppe in anderen Lehrveranstaltungen und im späteren Anwendungsbereich verbreitet?
6. Gewohnheiten	Wurde in der Vorbildung der Zielgruppe bereits ein Informatikmittel eingesetzt?
7. Eignung für Kompetenzlevel der Lernenden	Ist das Informatikmittel auch für Anfängerinnen und Anfänger geeignet, und/oder ist eventuell ein Einstieg im späteren Verlauf des Kurses denkbar?
8. Installationsaufwand	Steht der Installationsaufwand für das Informatikmittel in einem vernünftigen Verhältnis zu seiner Verwendung im Unterricht?

Tab. 8.4: Einige Kriterien für die Auswahl von Informatikmitteln.

Für einen problembasierten Einführungskurs ins Programmieren steht die Auswahl folgender Informatikmittel an:

- 1. Wahl der Programmiersprache
- 2. Wahl der Programmierumgebung

1. Wahl der Programmiersprache

Da die grundlegenden Programmierkonzepte bei vielen Programmiersprachen sehr ähnlich sind, ist die Wahl der Programmiersprache eher sekundär. Bei einer Programmierereinführung steht zunächst eine Reihe allgemeiner Programmierkonstrukte im Zentrum, die in einer Sprache ausprogrammiert werden. Letztendlich muss die Wahl

anhand der Kriterien aus Tab. 8.4 getroffen werden. In unserem Fall wurde sie vom Dozierenden bestimmt.

Die Programmiersprache *Java* (Software-Development-Kit SDK 1.4.X, <http://java.sun.com>) erfüllt die Entscheidungskriterien 1 bis 5. Kriterium 6 (Gewohnheit für ein Informatikmittel) ist aufgrund des hohen Anteils Studierender ohne Programmierkenntnisse weniger stark zu gewichten. Unter den Studierenden, die in der Vorumfrage angegeben haben, dass sie bereits Programmierkenntnisse haben, zeigt sich kein klarer Favorit (Abb. 8.3, E). *Java* liegt dabei hinter der Programmiersprache *Turbo Pascal* und *C++* an dritter Stelle. Punkt 7 (Eignung für Anfängerinnen und Anfänger) wird nur unter der Bedingung erfüllt, dass der Fokus zunächst auf die Aneignung der Grundlagen gerichtet wird. Eigenschaften der Programmiersprache *Java*, wie beispielsweise die Objektorientierung, soll zunächst bewusst ausgeblendet und auf später verschoben werden.

2. Wahl der Programmierwerkzeuge

Die Wahl der Programmierwerkzeuge soll der Aufgabengrösse angepasst werden. Sie sollen ab jenem Zeitpunkt eingesetzt werden, sobald es vom Umfang der Aufgabe her erforderlich ist. Die Idee stammt aus dem Programmierkurs „Programmieren und Problemlösen“ (ETH Zürich, Kurs 251-0842-00L, Tab. 1.2), bei dem die Grundlagen mit *Turbo Pascal* eingeübt werden und später auf eine professionelle Umgebung (Delphi) umgestiegen wird. Dadurch wird der Einstieg von weiteren Instruktionen, die eine professionelle Umgebung erfordert, entlastet. Die Wahl des Programmier-Editors wird so dem Kompetenzlevel des Kurses angepasst (Tab. 8.5).

Kompetenz-Level		Fokus auf Ebene	Werkzeug	Editor
1	Programmiergrundlagen	Syntax Kompilation	Texteditor Kompiler	<i>Notepad</i>
2	Entwicklung Algorithmus	kleine Programme (eine Klasse)	<i>Java Editor</i> mit Syntax-Highlighting oder Texteditor	<i>Notepad</i> oder anderer frei verfügbarer Programmiereditor
3	Umgang mit Programmen			
4	Umgang mit grösseren Programmen	grosse Programme (mehrere Klassen)	professionelle Programmierumgebung	<i>Eclipse</i>

Tab. 8.5: Im Kurs „Anwendungsnahes Programmieren“ eingesetzte Informatikmittel.

8.3.5 Output Prozess A: Problemstellungen

1. Problemstellung für die erste Teilkompetenz

Als Problemaufgabe für die erste selbständige (Teil-)Kompetenz zum Abschluss der Programmiergrundlagen (Tab. 8.5, Level 1) wurde die Programmierung des Galton Boards gewählt (Tab. 8.5, Level 2). Das Problem gilt als klassisches Beispiel, um ein stochastisches System zu demonstrieren (Abb. 8.4).

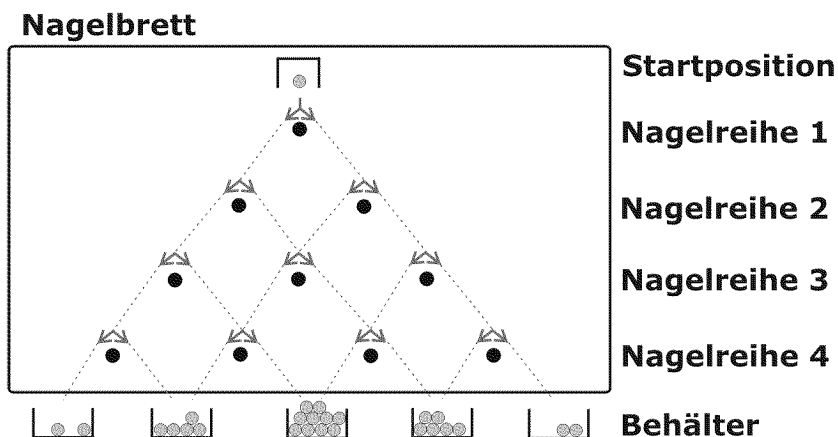


Abb. 8.4: Beispiel einer Problemstellung für ein erstes (Teil-)Kompetenzlevel: das Galton Board. Eine Kugel rollt von einer Startposition eine schiefe Ebene hinunter, trifft auf eine Reihe von Nägeln und landet schlussendlich in einem Auffangbehälter. Der Weg der Kugel wird jeweils mit gleicher Wahrscheinlichkeit rechts oder links vom Nagel fortgesetzt. Nach einer bestimmten Anzahl Entscheidungen, d.h. wenn die Kugel durch das ganze Nagelbrett gerollt ist, landet sie mit einer gewissen Wahrscheinlichkeit in einem der Behälter [Kursunterlagen „Anwendungsnahes Programmieren“, *E.Tutorial*[®] 4, <http://www.ap.ethz.ch>, Programmcode siehe Anhang B].

Zur Umsetzung des Problems sind wichtige Grundlagentheoriekonzepte für das Programmieren zur Anwendung zu bringen. Um eine einfache Version eines Galton Boards zu programmieren, müssen unterschiedliche Konzepte angewendet und umgesetzt werden können. Dazu gehören:

- Umgang mit der Programmierumgebung
- Daten in Variablen speichern und verarbeiten
- Bedingungsprüfungen
- Programmwiederholungen
- Generieren von Zufallszahlen
- Speichern und verarbeiten von Zahlenreihen (Arrays)

2. Problemstellung für die zweite Teilkompetenz

Als Problem für die zweite Teilkompetenz (Tab. 8.5, Level 3) wurde die Berechnung des Body-Mass-Index (BMI) in einem *Java-Applet* ausgewählt (Abb. 8.5).



Abb. 8.5: Beispiel einer Problemstellung für ein zweites (Teil-)Kompetenzlevel: Applet zur Berechnung des Body-Mass-Index (BMI). Durch zwei Eingabefelder können Körpergewicht und die Körpergröße eingegeben werden. Durch das Klicken der Schaltfläche „Berechnen“ wird neben dem berechneten BMI-Wert eine Meldung ausgegeben, ob die Person Normalgewicht hat, übergewichtig oder untergewichtig ist [Kursunterlagen Anwendungsnahes Programmieren, *E.Tutorial*® 6, <http://www.ap.ethz.ch>, Programmcode siehe Anhang B].

Um eine Version einer solchen BMI-Berechnung zu programmieren, sind folgende Konzepte zur Anwendung zu bringen:

- Generieren von Applets
- Umgang mit Fenster-Objekten (Buttons, Eingabefelder etc.)
- Behandlung von Ereignissen

Weitere Details zu den Problemstellungen sind den Kursunterlagen zu entnehmen [Kursunterlagen Anwendungsnahes Programmieren, <http://www.ap.ethz.ch>].

8.4. Prozess B: Anwendungsebenen-Modell

8.4.1 Entscheidung 1: Ist die Aufteilung in Teilkompetenzen erforderlich?

An dieser Stelle der Entwicklung muss die Frage beantwortet werden, ob für den Kompetenzerwerb in einer problembasierten Instruktion, die zur Lösung der Aufgabe aus Prozess A (siehe 3.5) erforderlich ist, eine Problemaufgabe ausreicht, oder ob sie durch das Anwendungsebenen-Modell über mehrere Probleme aufgebaut werden muss (Tab. 8.1, Entscheidung 1). Dies ist dann notwendig, wenn zur Lösung des Problems vorgängig eine ganze Reihe von isolierten Konzepten problembasiert eingeführt werden muss (siehe Kapitel 7 „Didaktische Modelle“).

Problemstellung 1: Galton Board-Problem

Die Galton Board-Aufgabe wird für Anfängerinnen und Anfänger erst dann selbständig lösbar, wenn die erforderlichen Konzepte (siehe Abschnitt 8.3.5) als anwendbares Wissen zur Verfügung stehen. Beim Anwendungsebenen-Modell soll eine Reihe von kleinen Problemen, die jeweils nur ein einzelnes Konzept bearbeiten, dieses Wissen erzeugen (siehe Kapitel 5 „PBL“; Abb. 5.3, problembasierter Wissenserwerb).

Problemstellung 2: BMI-Berechnung

Für die BMI-Berechnung kommt kein Anwendungsebenen-Modell zum Einsatz. Das Problem wird direkt ins 4-Schritte-Modell integriert.

8.4.2 Output Prozess B: Problemserie zum Aufbau einer (Teil-)Kompetenz

Für das Anwendungsebenen-Modell müssen zur Einführung der Konzepte weitere Probleme gefunden und deren Aufgabenstellung konstruiert werden (Tab. 8.6). Diese sollen weniger umfangreich sein und das Konzept möglichst gut repräsentieren (folglich Kapitel 5 „PBL“; Problembasierter Wissenserwerb). Jedes dieser Probleme wird dann ebenfalls über das 4-Schritte-Modell eingeübt.

	Problem	Konzepte	Fertigkeiten	Aufgaben
Problembasierter Wissenserwerb	Problem 1	Variablen Bedingungsprüfung	Umgang mit der Programmierumgebung, Daten speichern und verarbeiten, Bedingungsprüfungen	Elektronisches Schloss: Prüfen, ob 3 eingegebene Zahlen mit den 3 Zahlen eines Schlosses übereinstimmen
	Problem 2	Schleifen	Anwenden von Programm-wiederholungen	Berechnung von Standard-Papierformaten: z.B. Berechnung von DIN A4 aus dem Format DIN A0
	Problem 3	Zufallszahlen Arrays	Verwendung von Programm-bibliotheken, Speichern und verarbeiten von Datenreihen	Berechnen und Simulieren des Geburtstags-Paradoxons
Problembasiertes Anwenden	Problem 4	Simulation	Ein Programm so in mehrere Teile zerlegen, dass es unter Anwendung der Konzepte P1 bis P3 gelöst werden kann.	Programmieren des Galton Boards

Tab. 8.6: Aufbau einer Problemserie nach dem Anwendungsebenen-Modell zur Lösung der Galton Board-Aufgabe. Die Problemaufgaben 1 bis 3 dienen dem problembasierten Wissenserwerb, die Problemaufgabe 4 der problembasierten Anwendung (siehe Kapitel 7 „Didaktische Modelle“, Tab. 7.6). Die Lösungen der einzelnen Aufgaben sind im Anhang B zu finden [Homepage zum Kurs „Anwendungsnahes Programmieren“ siehe <http://ap.ethz.ch>].

8.5. Prozess C: Erstellen der Lernmaterialien für das 4-Schritte-Modell

Die auf der Ebene der Kompetenz und Teilkompetenz ausgewählten und konstruierten Probleme werden anschliessend ins 4-Schritte-Modell integriert. Die im 4-Schritte-Modell entstandenen Outputs bilden die Lernmaterialien für den kompetenzorientierten Unterricht mit dem gleichnamigen Modell (siehe Kapitel 7 „Didaktische Modelle“, Abb. 7.3). Im Folgenden soll die detaillierte Umsetzung für das 4-Schritte-Modell am Beispiel des Problems 3 des Programmierkurses (Zufallszahlen, Arrays, siehe Tab. 8.6) gezeigt werden.

Beispiel: Problem 3 (Konzepte: Zufallszahlen, Arrays)

Wie der Tab. 8.6 entnommen werden kann, wurden bis zu diesem 3. Problem der Umgang mit Variablen, Bedingungsprüfungen und Schleifen problembasiert eingeübt. Zwei wichtige Konzepte sind zur Programmierung des Galton Boards noch einzuführen: Das Generieren von Zufallszahlen und das Speichern und Verarbeiten von Datenreihen (Arrays).

Als zu lösendes Problem wurde die Berechnung und Simulation des Geburtstags-Paradoxons ausgewählt (Abb. 8.6). Die Problemstellung wird nun wie folgt ins 4-Schritte-Modell integriert:

- Das Formulieren der Aufgabenstellung (C1)
- Die zur Lösung der Aufgabe erforderlichen Konzepte kommen einerseits in die Begleitunterlagen (C2) und dienen andererseits als Ausgangslage der problem-basierten Instruktion (C3).

```

A
Geburtstag [0]=349
Geburtstag [1]=71
Geburtstag [2]=102
Geburtstag [3]=18
Geburtstag [4]=348
Geburtstag [5]=95
Geburtstag [6]=356
Geburtstag [7]=288
Geburtstag [8]=133
Geburtstag [9]=356
Geburtstag [10]=161
Geburtstag [11]=347
Geburtstag [12]=319
Geburtstag [13]=229
Geburtstag [14]=262
Geburtstag [15]=76
Geburtstag [16]=201
Geburtstag [17]=346
Geburtstag [18]=179
Geburtstag [19]=203
Geburtstag [20]=171
Geburtstag [21]=28
Geburtstag [22]=114
Geburtstag [23]=320
Geburtstag [24]=295
Geburtstag [25]=242
Geburtstag [26]=357
Geburtstag [27]=115
Geburtstag [28]=327
Geburtstag [29]=224
Gefundene Paare 1

B
Die Wahrscheinlichkeit, dass in einer Gruppe von 30 Personen mindestens zwei am
gleichen Tag Geburtstag haben, ist 71.0%

```

Abb. 8.6: Konsole-Output der Aufgabenstellung 3 zur Anwendung der Konzepte „Zufallszahlen“ und „Arrays“: Es muss ein Programm geschrieben werden, welches die Wahrscheinlichkeit berechnet, dass in einer Gruppe von 30 Personen mindestens zwei am gleichen Tag Geburtstag haben. Zur Lösung der Aufgabe ist ein Array mit 30 Elementen zu generieren. Jedem Element wird eine Zufallszahl zugewiesen und dann verglichen, wie viele gleiche Werte vorhanden sind (A). Anschliessend wird der Versuch 100 Mal wiederholt, um die Wahrscheinlichkeit zu erhalten (B). Programm-Code siehe Anhang B [Kursunterlagen Anwendungsnahes Programmieren, *E.Tutorial*® 4, <http://www.ap.ethz.ch>].

8.5.1 Prozess C1: Formulieren der Aufgabenstellung

In der Aufgabenstellung stehen alle zur Lösung der Aufgabe erforderlichen Informationen.

Dazu gehören:

- die genaue Aufgabenstellung, inklusive notwendiger Hintergrundinformationen zur Aufgabe
- alle Materialien (Daten, Dateien etc.), die zur Lösung der Aufgabe erforderlich sind
- die Abgabe-Bedingungen für die Phase *EXPLAIN*.

Nicht in diesen Teil gehören Begriffs- und Konzepterklärungen.

Hilfsmittel:

Zur Formulierung der Aufgabenstellung wird eine Dokumentenvorlage verwendet (Teil des *E.Template*, siehe Tab. 8.2).

8.5.2 Prozess C2: Formulieren der Begleitunterlagen

Die Begleitunterlagen enthalten folgende Elemente:

- Inhaltsverzeichnis für die ganze Unterrichtseinheit
- Alle in der Unterrichtseinheit behandelten Begriffe und Konzepte
- Kurze Beschreibung, was die Studierenden nach dieser Unterrichtseinheit können werden (entspricht den als Kompetenz formulierten Lernzielen)
- Kurze und kompakte Einführung in die Konzepte der Unterrichtseinheit
- Begründung, weshalb diese Konzepte für die Zielgruppe relevant sind

In diesem Teil werden die zu lernenden Konzepte einer Unterrichtseinheit dargestellt. Es soll ein kompaktes Gedankengerüst geschaffen werden, welches während dem späteren Anwenden in Phase *TRY* ausdifferenziert werden kann (folglich Abb. 7.3 und Tab. 7.3). Wichtig bei der Formulierung dieser Einführung ist die Kürze. Für diesen Teil hat sich die Verwendung von Bildern und Metaphern statt Definitionen bewährt (siehe Kapitel 6.3.2 „Konstruktive Begriffsbildung“).

Hilfsmittel:

Zur Formulierung dieser Begleitunterlagen wird eine Dokumentvorlage verwendet (Teil des *E.Template*, siehe Tab. 8.2).

8.5.3 Prozess C3: Problembasierte Instruktion

Ziel der problembasierten Instruktion ist es, eine starke Interaktion zwischen der Aufgabenstellung, der zu vermittelnden Konzepte sowie der Handlung der Lernenden herzustellen. Im Instruktionsfenster des *E.Tutorials*[®] werden diese drei Ebenen zusammengeführt (siehe Kapitel 6 „Problembasierte Instruktion“, Abb. 6.7).

Hilfsmittel:

Zur Erstellung von *E.Tutorials*[®] stehen folgende Hilfsmittel zur Verfügung (Teil des *E.Template*, siehe Tab. 8.2):

- Eine programmierte Vorlage eines *E.Tutorials*[®]
- Anleitung „*E.Tutorial*[®]*HowTo*“.

Für die Autorenanarbeit ist ferner ein HTML-Editor zu bedienen. Zusätzlich wird die Verwendung einer spezialisierten Software zur Erstellung von Print-Screens empfohlen. In Abb. 8.7 ist die Anordnung der verschiedenen Informatikmittel für die Autorenanarbeit dargestellt. Details zu den Autorenfertigkeiten und der Verwendung der dafür notwendigen Informatikmittel sind der Anleitung „*E.Tutorial*[®]*HowTo*“ zu entnehmen (Teil des *E.Template*, siehe Tab. 8.2).

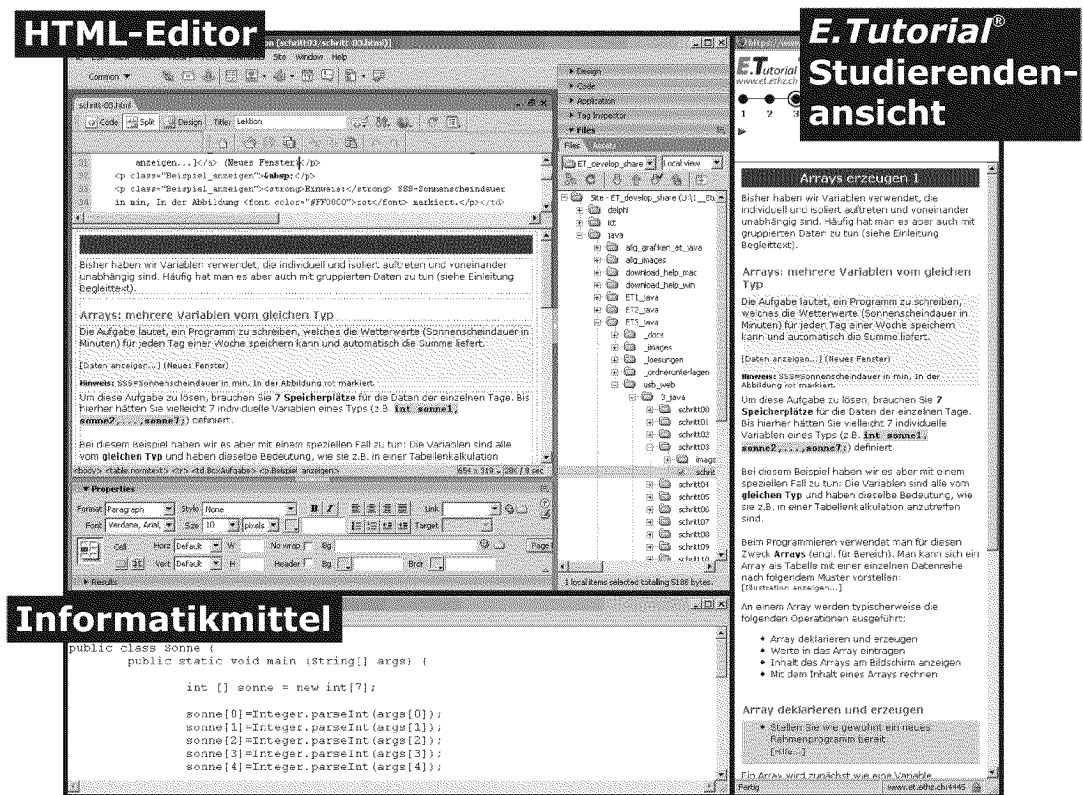


Abb. 8.7: Empfohlene Anordnung der Informatikmittel für die Autorenarbeit eines *E.Tutorials*[®]. Mit Hilfe eines HTML-Editors wird die programmierte Vorlage zur Erstellung eines *E.Tutorials*[®] (*E.Template*) bearbeitet (oben links). Das von den Studierenden zu bearbeitende Informatikmittel (hier ein Programmiereditor) wird ebenfalls auf dem Bildschirm positioniert (unten links). Auf diese Weise können die Arbeitsschritte entwickelt, ausgeführt und mit Hilfe einer Software zur Erstellung von Print-Screens festgehalten werden. Auf der rechten Seite wird das Instruktionfenster des fertigen *E.Tutorials*[®] in der Studierendenansicht angezeigt. Dies hilft dem Autor, sich in die Sicht der Studierenden beim Lösen des Problems zu versetzen.

E.Tutorial[®]: Einführung der Konzepte „Zufallszahlen“ und „Arrays“

Die Aufgabe des Geburtstags-Paradoxons erfordert die problembasierte Einführung der beiden Konzepte „Zufallszahlen“ und „Arrays“ (Tab. 8.6). Dazu sind folgende Aufgaben auszuführen:

- 1. Konstruktion von zwei weiteren Problemen, welche die zu vermittelnden Konzepte repräsentieren (Prozess C3.1)
- 2. Formulieren des Instruktionstextes zur Lösung dieser drei Aufgaben (Prozess C3.2)
- 3. Einbau von Konzepterklärungen in den Instruktionstext des *E.Tutorials*[®] (Prozess C3.3)
- 4. Einbau von Überprüfungsmöglichkeiten in das *E.Tutorial*[®] (Prozess C3.4)

Die während den einzelnen Prozessen entstehenden Outputs werden im *E.Tutorial*[®] zu einem problembasierten Instruktionsmedium zusammengefügt (siehe Kapitel 6 „Selbstgesteuerte, problembasierte Instruktion“).

1. Prozess C3.1: Problemkonstruktion

Bei den zwei weiteren kleineren Problemen für die beiden Konzepte „Zufallszahlen“ und „Arrays“ steht das Erlernen von Fertigkeiten im Zusammenhang mit diesen Konzepten im Zentrum (Tab. 8.7, Abb. 8.8 und Abb. 8.9). Die beiden Probleme dienen als Ausgangslage zur Formulierung von Instruktionstexten, Konzepterklärungen und Überprüfungsmöglichkeiten im *E.Tutorial*[®]. Die Probleme sollten nicht nur überschaubar, sondern auch auf das Vorwissen der Lernenden abgestimmt sein. Auf diese Weise wird dieses Wissen weiter vertieft und mit dem neuen verknüpft. In den ersten beiden *E.Tutorials*[®] haben die Studierenden gelernt, Variablen zu deklarieren, damit zu rechnen und deren Werte auf dem Bildschirm ausgeben.

Konzept	Problemstellung	Fertigkeiten
Zufallszahlen	Würfel (Abb. 8.8)	•Erzeugen von Zufallszahlen
Arrays	Sonnenscheindauer (Abb. 8.9)	•Werte in den Array eintragen •Inhalt des Arrays am Bildschirm anzeigen •Mit dem Inhalt eines Arrays rechnen

Tab. 8.7: Problembasierte Einführung der Konzepte „Zufallszahlen“ und „Arrays“ zusammen mit Fertigkeiten im *E.Tutorial*[®].

```

import java.io.*;
public class Wuerfel {

    public static void main (String[] args) {

        int wurf1, wurf2,i;

        for (i=0; i<10 ;i++) {

            wurf1 = (int)(Math.random() * 6)+1;
            wurf2 = (int)(Math.random() * 6)+1;

            if (wurf1==wurf2){System.out.println("*****YES*****");
            }

            else {

                System.out.println("no match");
            }

        }

    }
}
//main
//class

```

A

```

C:\Meine Java Programme>java Wuerfel
no match
no match
no match
*****YES*****
no match
*****YES*****
*****YES*****
no match
no match
no match

```

B

Abb. 8.8: Problemaufgabe 1 zur Einführung des Konzepts „Zufallszahlen“: Würfel-Programm, bei dem 10 Mal hintereinander mit zwei Würfeln gewürfelt wird und „YES“ ausgegeben wird, wenn die beiden Würfel die gleiche Augenzahl anzeigen. A: Java-Programmcode, an der eingerahmten Stelle werden zwei Zufallszahlen von 1 bis 6 generiert. B: Output des Programms nach der Ausführung in der Konsole [Kursunterlagen Anwendungsnahe Programmieren, *E.Tutorial*[®] 3, <http://www.ap.ethz.ch>].

```

import java.io.*;
public class Sonne {
    public static void main (String[] args) {

        int [] sonne = new int[7];

        for (int i=0; i<7; i++) {
            sonne[i]=Integer.parseInt(args[i]);
        }

        int summe, tag;
        summe=0;

        for (tag=0; tag<7; tag++) {
            summe = summe+sonne[tag];
        }

        System.out.println("Die Gesamt-Sonnenscheindauer ist: "+summe);
    }
}

```

A

```

D:\java Sonne 539 221 616 0 638 749 527
Die Gesamt-Sonnenscheindauer ist: 3290

```

B

Abb. 8.9: Aufgabenstellung 2 zur Einführung des Konzepts „Array“: Programm, welches Wetterwerte (Sonnenscheindauer in Minuten) für jeden Tag einer Woche speichert und die Summe berechnet. A: Java-Programmcode, an der eingerahmten Stelle wird ein Array mit 7 Elementen generiert und am Bildschirm ausgegeben. B: Output des Programms nach der Ausführung in der Konsole [Kursunterlagen Anwendungsnahe Programmieren, E.Tutorial® 3, <http://www.ap.ethz.ch>].

2. Prozess C3.2: Formulieren des Instruktionstexts

Für das Formulieren des Instruktionstextes ist eine genaue Analyse der einzelnen Handlungsschritte durchzuführen. Diese sind in einen logischen, aufbauenden Ablauf zu bringen, der auf der Ausgangs- oder Anfangskompetenz der letzten gelösten Aufgabe aufbaut und zum Ziel führt (z.B. ein funktionierendes Programm). Dieser Prozess ist mit dem Formulieren eines Drehbuches zur Umsetzung einer Filmsszene vergleichbar.

Für die beiden Problemaufgaben der Abb. 8.8 (Zufallszahlen) und Abb. 8.9 (Arrays) sind diese Handlungsschritte in den Tab. 8.8 und Tab. 8.9 aufgelistet.

Handlungsschritte zur Erzeugung von Zufallszahlen	
1.	Rahmenprogramm bereitstellen (bekannt aus <i>E.Tutorial</i> ® 1 und 2)
2.	Eine Ganzzahl-Variable deklarieren (bekannt aus <i>E.Tutorial</i> ® 1 und 2)
3.	Der Variable die Methode <code>math.Random()</code> zuweisen
4.	Die Zahl mit 6 multiplizieren
5.	Die Gleitkommazahl in eine Ganzzahl konvertieren
6.	Der Zahl 1 addieren
7.	Ausführen der restlichen Schritte (bekannt aus <i>E.Tutorial</i> ® 1 und 2)

Tab. 8.8: Handlungsschritte zur Generierung von „Zufallszahlen“ (Ganzzahl von 1 bis 6) [Kurs „Anwendungsnahe Programmieren“, *E.Tutorial*® 3, Lektion 2, <http://www.ap.ethz.ch>].

Handlungsschritte zur Erstellung von Arrays	
1.	Rahmenprogramm bereitstellen (bekannt aus <i>E.Tutorial</i> ® 1 und 2)
2.	Ein Array deklarieren
3.	Jedem Element eines Arrays eine Eingabezahl zuweisen
4.	Die Array-Werte am Bildschirm ausgeben
5.	Mit Hilfe einer Schleife die Array-Werte aufsummieren
6.	Ausführen der restlichen Schritte (bekannt aus <i>E.Tutorial</i> ® 1 und 2)

Tab. 8.9: Handlungsschritte zur Generierung von „Arrays“ [Kurs „Anwendungsnahe Programmieren“, *E.Tutorial*® 3, Lektionen 3 bis 5, <http://www.ap.ethz.ch>].

3. Prozess C3.3: Integration der Konzepte und Erläuterungen

Damit die Lernenden während dem Ausführen der einzelnen Handlungsschritte beim Lösen von Problemen die wesentlichen Punkte verallgemeinern, sind neben den Handlungsanweisungen Konzepterläuterungen und Zusammenfassungen in die Anleitung zu integrieren (siehe Kapitel 6.4 „Elemente einer selbstgesteuerten, problembasierten Anleitung“). Dies ist vergleichbar mit den Erklärungen, die Dozierende beim Vorlösen einer Aufgabe geben würden. Auf diese Weise soll erreicht werden, dass die Lernenden jederzeit wissen, was sie tun. Sie können ein Maximum aus der Arbeit mit einem Problem herausholen und die notwendige Flexibilität im Umgang mit einem Konzept aufbauen (Abb. 8.10).

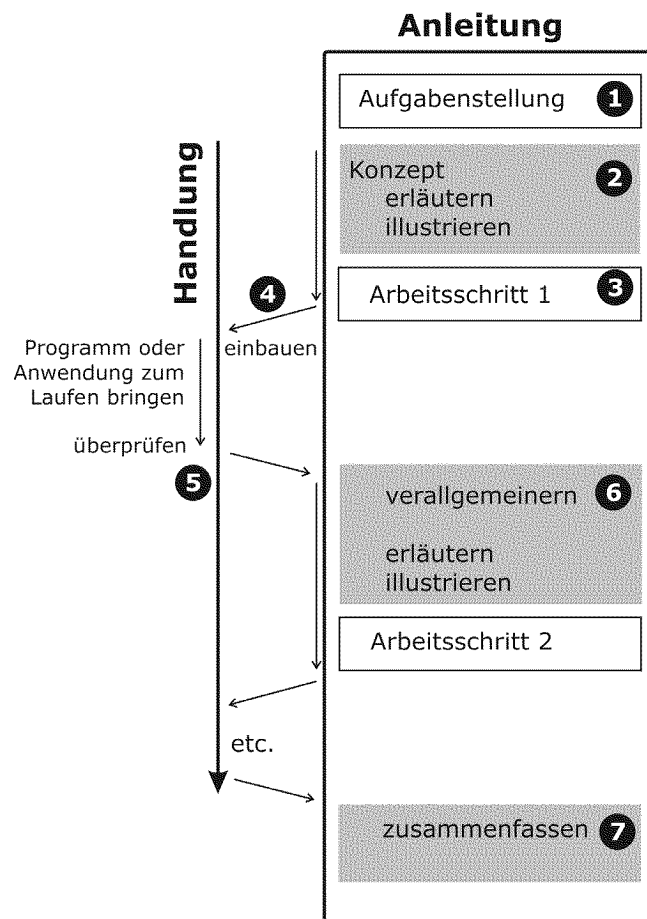


Abb. 8.10: Einbau von Konzepten in die handlungsbasierte Anleitung. Die Aufgabenstellung (1) macht das Beiziehen von Konzepten notwendig (2). Durch einen Arbeitsschritt (3) wird das Konzept zur Lösung eines Teilschritts der Aufgabe angewendet und direkt in eine Handlung umgesetzt (4). Die Überprüfungsmöglichkeit (5) führt die Lernenden zurück in die Anleitung. Das Konzept wird dort verallgemeinert (6) und es wird ein nächster Schritt vorbereitet und anschliessend ausgeführt. Am Ende einer Lektion werden alle wichtigen Punkte des Konzepts nochmals zusammengefasst (7).

Die Integration von Konzepten ins Instruktionsfenster des *E.Tutorials*[®] zur Erstellung von Zufallszahlen ist der Abb. 8.11 zu entnehmen.

4. Prozess C3.4: Überprüfungsmöglichkeiten schaffen

Durch das Einfügen von Hyperlinks ins Instruktionsfenster können im Überprüfungs-fenster Grafiken eingeblendet werden. Diese zeigen den Zustand der fertigen Umsetzung eines Arbeitsschrittes im Anwendungsfenster des *E.Tutorials*[®] (siehe Kapitel 7 „Didaktische Modelle“, Abb. 7.7). Die Integration von Überprüfungsmöglichkeiten ins Instruktionsfenster des *E.Tutorials*[®] zur Erstellung von Zufallszahlen ist der Abb. 8.11 zu entnehmen.

8.5.4 Output Prozess C3: *E.Tutorial*®

1. Struktur des *E.Tutorials*®

Im Instruktionsfenster des *E.Tutorials*® werden die Ebenen „Aufgabenstellung“, „Konzept“ und „auszuführende Handlungen“ sowie Hyperlinks zur Überprüfung der Handlungen als Teil des Designs zusammengeführt (siehe Abb. 8.10). Das Instruktionsfenster des *E.Tutorials*® wurde durch die Definition von CSS-Formaten (*Cascading Style Sheets*), ein Format zum Speichern von HTML-Formatierungen, folgendermassen strukturiert (Tab. 8.10):

Style-Name und Funktion	CSS-Format	Aussehen
Lektionstitel	Titel_Lektion	Hintergrund in der Farbe des <i>E.Tutorials</i> ®
Aufgabenstellung	BoxAufgabe	Gelber Hintergrund
Arbeitsschritte	BoxAnweisung	Grüner Hintergrund
Erläuterungen	Normtext	Weisser Hintergrund
Überprüfungsmöglichkeiten	Beispiel_anzeigen	Kleine Schrift (Hyperlinks)
Programm-Code	Code_Editor	Grauer Hintergrund
Ausgabe Konsole	Code_Konsole	Schwarzer Hintergrund

Tab. 8.10: Struktur des *E.Tutorials*®. Einige CSS-Formate des *E.Tutorial*®-Templates [Dahinden 2006].

2. *E.Tutorial*®-Lektionen

Ein *E.Tutorial*® wird so in Lektionen aufgeteilt, dass deren Bearbeitung ungefähr gleich lange dauert. Die Tab. 8.11 zeigt den Aufbau des ganzen *E.Tutorials*® dieser Lerneinheit. Es besteht aus 9 Lektionen. Für deren Bearbeitung ist insgesamt eine Zeit von 2 bis 4 Stunden vorgesehen.

Abb. 8.11 zeigt das Instruktionsfenster der Lektion 2 des *E.Tutorials*® zur Erstellung von „Zufallszahlen“ und Abb. 8.12 das Instruktionsfenster der Lektionen 3 und 4 mit dem Titel „Arrays erzeugen 1 und 2“. Wegen ihrer Länge wurde die Lektion zu den Arrays in zwei Teile aufgeteilt. Für die Elemente der Anleitung siehe Abb. 8.10 sowie Kapitel 6.4 „Elemente einer selbstgesteuerten, problembasierten Anleitung“.

	Lektion	Titel	Aufgabe	Inhalt
Problembasierte Instruktion	Lektion 1	Aufgabenstellung		Aufgabenstellung und Zielsetzung für das ganze <i>E.Tutorial</i> [®]
	Lektion 2	Zufallszahlen	Würfel	Konzept „Zufallszahlen“
	Lektion 3	Arrays erzeugen 1	Sonne	Konzept „Arrays“: Deklarieren und erzeugen, Werte ins Array eintragen, Inhalt des Arrays am Bildschirm anzeigen, mit dem Inhalt eines Arrays rechnen
	Lektion 4	Arrays erzeugen 2		Mit Hilfe von Schleifen (bekannt aus <i>E.Tutorial</i> [®] 2) das Array mit Werten füllen und die eingegebenen Werte am Bildschirm ausgeben
	Lektion 5	Mehrdimensionale Arrays		Mit Hilfe verschachtelter Schleifen (bekannt aus <i>E.Tutorial</i> [®] 2) mehrdimensionale Arrays ausgeben.
Problembasierte Anwendung	Lektion 6	Geburtstags-Paradoxon 1	Geburtstags-Paradoxon	Problem in sinnvolle Arbeitsschritte aufteilen. Schritt 1: Geburtstage zufällig generieren (Konzepte „Zufallszahlen“, „Arrays“ anwenden)
	Lektion 7	Geburtstags-Paradoxon 2		Schritt 2: Geburtstage paarweise vergleichen (Konzepte „geschachtelte Schleife“, „Bedingungsprüfung“ anwenden) Schritt 3: Gefundene Paare zusammenzählen
	Lektion 8	Geburtstags-Paradoxon 3		Schritt 4: 100 Versuche durchführen (Konzept „Schleifen“) Schritt 5: Summe der Versuche mit identischen Paaren berechnen.
	Lektion 9	Abschluss		Abschluss und Diskussion des Resultates

Tab. 8.11: Aufbau eines *E.Tutorials*[®] zum Thema „Zufallszahlen“ und „Arrays“ des Kurses „Anwendungsnahes Programmieren“. Die Ersten 5 Lektionen üben die Konzepte nach dem Anwendungsebenen-Modell problembasiert ein (problembasierte Instruktion), während die Lektionen 6 bis 9 dem Anwenden der Konzepte in einem umfangreicheren Problem (Geburtstags-Paradoxon) gewidmet sind (problembasierte Anwendung, siehe Tab. 8.6). Der Aufbau des Kurses und die Lösungen der Aufgaben sind dem Anhang zu entnehmen [Kurs „Anwendungsnahes Programmieren“, <http://www.ap.ethz.ch>].

A

Zufallszahlen

Zufallszahlen werden bei Simulationen und Spielprogrammen häufig gebraucht. Es wäre recht langweilig, wenn sich eine Simulation oder ein Spiel jedes Mal gleich verhalten würde. Zufälligkeit kann beim Programmieren Vielfalt erzeugen.

Aufgabe

Als erste Aufgabe schreiben wir ein einfaches Würfel-Programm.

Für dieses Programm muss eine Zufallszahl zwischen 1 und 6 erzeugt werden. Um in Java eine Zufallszahl zu erhalten, rufen wir die Methode **Random** in der Mathematik-Bibliothek auf:

- Deklarieren Sie eine Ganzzahl-Variablen (z.B. `wuerfel`).
- Weisen Sie der Variable die Methode `Math.random()` zu.

[Beispiel anzeigen...]

Diese Methode gibt eine Zufallszahl zurück und zwar eine Zahl vom Typ **double** im Bereich 0.0 bis 0.999... Wie erhalten wir nun daraus eine **Ganzzahl** zwischen 1 und 6?

- Wir multiplizieren die Zahl mit 6 und erhalten gleichmässig verteilte Zahlen im Bereich von 0.0 bis 5.999... (1)
- Als nächstes konvertieren wir die Gleitkommazahl in eine Ganzzahl durch das Voranstellen des Operators (`int`) und erhalten die Ganzzahlen 0 bis 5. Durch (`int`) werden die Dezimalstellen abgeschnitten: z.B. aus 3.7 wird die Zahl 3. (2)
- Schliesslich addieren wir 1 und erhalten die Zufallszahlen im Bereich von 1 bis 6. (3)

[Illustration anzeigen...]

B

• Führen Sie im Programm die obigen Schritte aus!

[Schritt 1 anzeigen...]
[Schritt 2 anzeigen...]
[Schritt 3 anzeigen...]

• Geben Sie die Zahl aus und testen Sie das Programm!

[Kurzform: Bildschirmausgabe]

Nun sollte eine Zahl von 1 bis 6 ausgegeben werden.

Als nächstes werden Sie Ihren Würfel zu einem kleinen Würfelspiel mit 2 Würfeln ausbauen.

Programm ergänzen

Ergänzen Sie das Programm, so dass...

- mit zwei Würfeln geworfen werden kann.
- 10 Mal hintereinander gewürfelt wird.
- `*****ybs*****` ausgegeben wird, wenn die beiden Würfel die **gleiche** Augenzahl haben
- und
- `Sorry` ausgegeben wird, wenn die beiden Würfel **nicht** die gleiche Augenzahl haben.

[Beispiel anzeigen...]

Hilfe

[Schritt 1...]
[Schritt 2...]
[Schritt 3...]

• Lassen Sie das Programm erneut laufen!

Die Zahlen, welche die Methode `random` benutzen, sind **genau** genommen nicht wirklich zufällig. Es werden lediglich Serien von Zahlen erzeugt, die wie Zufallszahlen aussehen, sogenannte Pseudo-Zufallszahlen.

Zusammenfassung:

- Zufallszahlen können mit der Methode

Abb. 8.11: Strukturierung der Lektion im Instruktionsfenster des *E.Tutorials*® zur Einführung des Konzepts „Zufallszahlen“. Die Elemente der handlungsbasierten Anleitung siehe Abb. 8.10 sowie Kapitel 6.4 „Elemente einer selbstgesteuerten, problembasierten Anleitung [Kurs „Anwendungsnahes Programmieren“, *E.Tutorial*® 3, Lektion 2 „Zufallszahlen“, <http://www.ap.ethz.ch>].

Arrays erzeugen 1

Bisher haben wir Variablen verwendet, die individuell und isoliert auftreten und voneinander unabhängig sind. Häufig hat man es aber auch mit gruppierten Daten zu tun (siehe Einleitung Begleittext).

Arrays: mehrere Variablen vom gleichen Typ
Die Aufgabe lautet, ein Programm zu schreiben, welches die Wetterwerte (Sonnenscheindauer in Minuten) für jeden Tag einer Woche speichern kann und automatisch die Summe liefert.

[Daten anzeigen...] (Neues Fenster)

Hinweis: 888=Sonnenscheindauer in min, in der Abbildung rot markiert.

Um diese Aufgabe zu lösen, brauchen Sie **7 Speicherplätze** für die Daten der einzelnen Tage. Bis hierher hatten Sie vielleicht 7 individuelle Variablen eines Typs (z.B. `int sonne1`, `sonne2`, ..., `sonne7`) definiert.

Bei diesem Beispiel haben wir es aber mit einem speziellen Fall zu tun: Die Variablen sind alle vom **gleichen Typ** und haben dieselbe Bedeutung, wie sie z.B. in einer Tabellenskalkulation anzutreffen sind.

Beim Programmieren verwendet man für diesen Zweck **Arrays** (engl. für Bereich). Man kann sich ein Array als Tabelle mit einer einzelnen Datenreihe nach folgendem Muster vorstellen:
[Illustration anzeigen...]

An einem Array werden typischerweise die folgenden Operationen ausgeführt:

- Array deklarieren und erzeugen
- Werte in das Array eintragen
- Inhalt des Arrays am Bildschirm anzeigen
- Mit dem Inhalt eines Arrays rechnen

Array deklarieren und erzeugen

- Stellen Sie wie gewohnt ein neues Rahmenprogramm bereit.
[Hilfe...]

Ein Array wird zunächst wie eine Variable deklariert, d.h. ein Name gegeben. Für ein Array vom Typ Integer mit dem Namen "sonne" sieht das z.B. so aus:

Beispiel: `int[] sonne;`

Ein Array wird mit dem Operator `new` gebildet, um eine Instanz der Klasse "sonne" zu erzeugen:
Beispiel: `sonne = new int [7];`

Dieser Array mit dem Namen "sonne" ist gross genug, um 7 Variablen vom gleichen Typ (Integer) abzuspeichern.

Die beiden Schritte werden meist in einer einzigen Anweisung geschrieben:

Beispiel: `int[] sonne = new int [7];`

- Deklarieren Sie ein Array, der die Sonnenscheindauer für jeden der sieben Wochentage abspeichern kann.

[Lösung anzeigen...]

Weiber zur nächsten Lektion...

© Institut für Computational Science, ETH Zürich, 2006

Arrays erzeugen 2

Das ganze Array wird mit seinem Namen angesprochen. Um ein einzelnes Element anzusprechen, benutzt man einen Index.

Ein Programm verweist mit einem Indexwert auf ein individuelles Element eines Arrays:

Beispiel: `sonne[2]` verweist auf die Zahl mit dem Index 2. Aber Vorsicht, diese befindet sich an **dritter Stelle** im Array.

Die erste Position in einem Array ist immer die "nullte" Position, dann folgt die erste, zweite und dritte Position usw. Der Index gibt die Position des aktuellen Elements im Array an.
[Illustration Anzeigen...]

Werte in den Array eintragen

Ihr zuvor erzeugtes Array hat zwar 7 Speicherplätze, noch sind den Speicherplätzen aber keine Werte zugewiesen worden (auch nicht 0!). Wie bei den Einzelvariablen können wir jedem Array-Element entweder eine fixe Zahl zuweisen oder die Programm-Anwenderin eine Zahl eingeben lassen (mit `Integer.parseInt(args[i])`).

- Weisen Sie **jedem** Element des Arrays eine Zahl aus der Eingabe zu (Typ Integer!)
[Beifehl für "Eingabezahl" anzeigen...]

[Beispiel anzeigen...]

Hinweis: Das bedeutet natürlich, dass später beim Ausführen des Programms 7 Zahlen mit eingegeben werden müssen.

Tipp: Versuchen Sie die Zuweisung der 7 Eingabezahlen mit Hilfe einer Schleife zu generieren! Die Zahlvariable der Schleife entspricht dann dem Array-Element.
[Kurzform: For-Schleife...]

Inhalt des Arrays am Bildschirm anzeigen

Von nun an geben Sie nur noch den Namen und Index des Arrays an, wenn Sie ihn am Bildschirm ausgeben oder damit rechnen wollen.

- Geben Sie die einzelnen Zahlen der einzelnen Wochentage am Bildschirm aus.

Hinweis: Die Zahlen sollen beim Ausführen des Programms mit der Tastatur eingegeben werden.
[Kurzform: Bildschirmausgabe...]
[Lösung Anzeigen...]

[Wie schreiben...?]

Die Mächtigkeit von Arrays kommt erst dann richtig zum Tragen, wenn man in einer Schleife alle Elemente des Arrays bearbeitet. Die Zahlvariable der Schleife ist dann gleichzeitig der Index im Array.

Mit dem Inhalt eines Arrays rechnen

Als nächstes wollen wir die Summe der Sonnenscheindauer in einer Woche ausrechnen. Die etwas umständliche Art, die Werte zu addieren, wäre:

`summe=sonne[0]+sonne[1]+...+sonne[n]`

Besser ist die Verwendung einer **for-Schleife**. Eine Variable wird benutzt, um sowohl den Schleifenzähler, als auch den Wert des Index festzuhalten. Eine zweite Variable speichert die Gesamtsumme, der bei jedem Schleifendurchgang ein weiterer Wert zugeordnet wird.

Betrachten wir folgendes Beispiel:

```
int index, summe;
summe=0;
for (index = 0; index < T; index++) {
  summe = summe + sonne[index];
}
```

Die Schleife wird 7 Mal durchlaufen. Bei jedem Durchlauf wird der Wert des Arrays zur Gesamtsumme hinzuaddiert.

- Führen Sie die Aufsummierung der Array-Werte mittels for-Schleife durch und geben Sie das Resultat am Bildschirm aus!

[Lösung anzeigen...]
[Kurzform: For-Schleife...]
[Kurzform: Bildschirmausgabe...]

Zusammenfassung:

- Ein Array ist eine Sammlung von Variablen. Der Name des Arrays wird vom Programmierer vergeben. Alle Elemente eines Arrays haben denselben Datentyp.
- Ein Array mit dem Namen "blabla" wird wie folgt deklariert:
`int[] blabla = new int [6];`
wobei 6 die Anzahl Elemente des Arrays ist.
- Auf eine individuelle Komponente eines Arrays wird mit einem Index verwiesen
`blabla [2] = 4;`
- Die Indizes beginnen bei Null und enden bei ein wenig weniger als der Anzahl Elemente des Arrays.
`blabla [0] bis blabla [6]`

an Vorwissen anknüpfen

Aufgabe stellen

Hinführung

Aufteilung in logische Schritte

Konzept holen

Konzept anwenden

Konzept im Problem weiterentwickeln

bereits eingeführte Konzepte als Kurzform

zunehmende Schrittgrösse und Selbständigkeit

Konzepte abstrahieren und zusammenfassen

A

B

Abb. 8.12: Strukturierung zweier Lektionen im Instruktionsfenster des E.Tutorials® zur Einführung des Konzepts „Arrays“. Die Elemente der handlungsbasierten Anleitung siehe Abb. 8.10 sowie Kapitel 6.4.

8.6. Prozess D: Qualitäts-Feedback

Das Ziel des Qualitäts-Feedbacks ist die qualitative Verbesserung einer Lerneinheit. Dies wird erst möglich, wenn sie produktiv im Unterricht eingesetzt wird (siehe Kapitel 4.3.1 „System-Engineering von Lernumgebungen“). Für die Verbesserung einer Lerneinheit stehen folgende Fragen im Zentrum (siehe Abb. 8.1, Entscheidungen 2 und 3):

- Ist die Problemaufgabe der Phase *DO* des 4-Schritte-Modells zusammen mit den Lernmaterialien durch die Lernenden selbständig lösbar?
- Sind die Lernenden nach dem Lösen des Problems zufrieden und motiviert?

Die Lerneinheit wird so lange überarbeitet und verbessert, bis beide Fragen in Abb. 8.1 mit „ja“ beantwortet werden können. Um dies zu beurteilen, werden auf verschiedenen Ebenen Feedback-Information erfasst (Tab. 8.12 und Abb. 8.2).

Feedback-Ebene	Feedback-Information
Verbesserung der Lernmaterialien (Prozess D1)	Feedback-Möglichkeit nach der Phase <i>TRY</i> des 4-Schritte-Modells
Kompetenzkontrolle (Prozess D2)	Die mündliche Abgabe der Probleme der Phase <i>EXPLAIN</i> des 4-Schritte-Modells
	Resultate der Leistungskontrolle
Implementierung der Lerneinheit in den Gesamtkurs (Prozess D3)	Daten der Prozessanalyse

Tab. 8.12: Feedback-Informationen zur Qualitätsverbesserung von Lerneinheiten.

Für die Qualitätsverbesserung einer Lerneinheit hat es sich bewährt, wenn kleinere Fehler im operativen Einsatz erkannt und möglichst direkt danach korrigiert werden, anstatt auf die Vorbereitungsphase eines nächsten Kurses verschoben zu werden (weitere Details siehe Kapitel 9.4.6 „Qualitätskontrolle eines Kurses durchführen“).

8.6.1 Prozess D1: Verbesserung der Lernmaterialien

Um Feedback-Daten zur Verbesserung der Lernmaterialien zu erhalten, haben die Lernenden nach jedem *E.Tutorial*[®] die Möglichkeit, Feedbacks abzugeben. Dies kann über ein direkt ins *E.Tutorial*[®] integriertes Formular abgewickelt und über Internet verschickt werden (Abb. 8.13). Die Lernenden werden aufgefordert, Auskunft über folgende Fragen zu geben:

- Hatten Sie Schwierigkeiten?
- Was fanden Sie gut?
- Was könnte verbessert werden?

- Wie haben Sie gearbeitet?

Durch dieses Freitext-Feedback erhält man für die Entwicklung und Verbesserung der Lernumgebung wichtige spezifische Informationen. Diese Feedback-Möglichkeit direkt nach der praktischen Arbeit mit dem *E.Tutorial*® wird stark genutzt (Tab. 8.13). Vor allem für einen Ersteinsatz einer Lerneinheit können durch die auf diese Weise gewonnene Information kleine Fehler und Unklarheiten schnell erkannt und direkt behoben werden.

	ET 1	ET 2	ET 3	ET 4	ET 5	ET 6	ET 7
Anzahl abgegebener Feedbacks	209	153	137	128	149	67*	39*

Tab. 8.13: Durchschnittliche Anzahl abgegebener Feedbacks via Online-Formular am Ende eines *E.Tutorials*® (ET) im Kurs „Anwendungsnahes Programmieren“ in den Jahren 2005 bis 2006. Anzahl Studierende 326. * Bei ET 6 und 7 hatten die Lernenden die Wahl, welches sie lösen wollten.

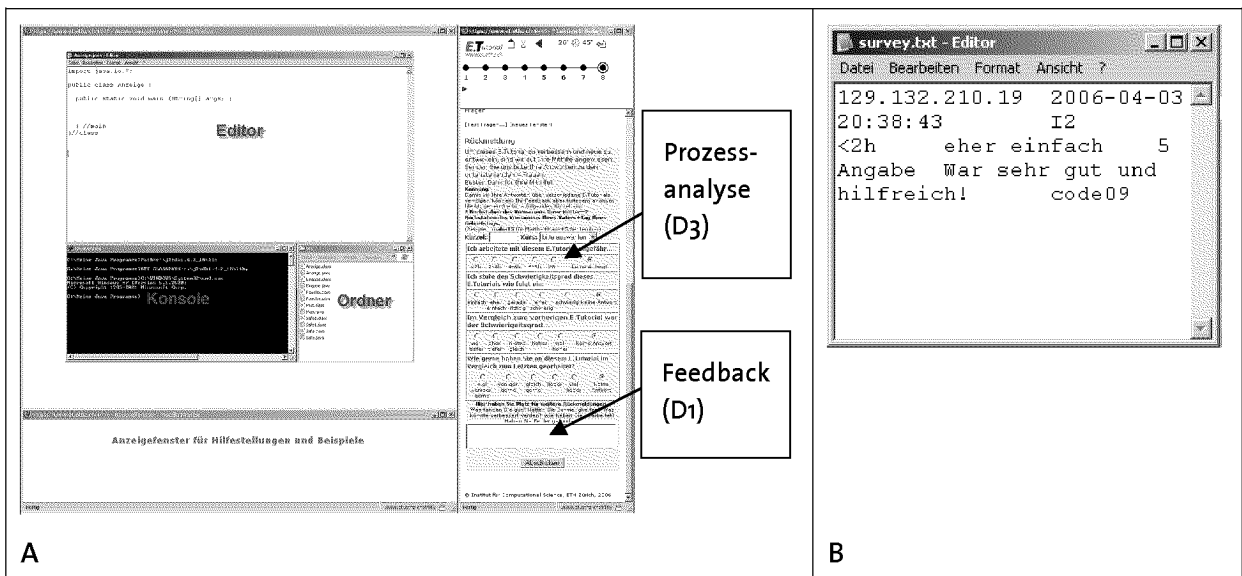


Abb. 8.13: Datenerhebung für Feedbacks (Prozess D1) und die Prozessanalyse (Prozess D3): Der Einbau eines Rückmeldeformulars direkt in das Instruktionfenster des *E.Tutorials*® ermöglicht ein kontinuierliches Sammeln von Feedback-Daten (A). Die Daten wurden in einer Log-Datei gesammelt (B) und anschließend mit einer Tabellenkalkulation ausgewertet.

8.6.2 Prozess D2: Kompetenzkontrolle

Zur Überprüfung, ob sich die Lernenden mit Hilfe der Lernumgebung die geforderten Kompetenzen angeeignet haben, werden Leistungsinformationen der Studierenden während verschiedenen Phasen des Unterrichts gesammelt (siehe Kapitel 9 „Einsatz der computergestützten Lernumgebung im Unterricht“ und 10 „Evaluation der Lernumgebungen“, Abb. 10.1). Daraus soll auf zu Grunde liegende Kompetenzen geschlossen werden.

Folgende Informationen spielen als Feedback-Information für die Kompetenzkontrolle eine Rolle:

- 1. Die mündliche Abgabe der Probleme der Phase *EXPLAIN* des 4-Schritte-Modells
- 2. Daten der Leistungskontrolle

1. Die mündliche Abgabe der Probleme der Phase *EXPLAIN* des 4-Schritte-Modells

Im 4-Schritte-Modell haben die Lernenden in der letzten Phase *EXPLAIN* einer Assistenzperson in einem persönlichen Gespräch zu erklären, wie sie das gestellte Problem einer Lerneinheit gelöst haben und wie die gelernten Konzepte darin zum Einsatz kommen (für Details siehe Kapitel 7 „Didaktische Modelle“). Diese Output-Information der Lerneinheit ist nicht nur für den Lernenden, sondern auch für die Entwicklerinnen und Entwickler von grossem Interesse. Weil die mündlichen Abgaben obligatorisch sind, kann man sich auf diese Weise ein Bild darüber verschaffen, welche „Wirkung“ eine Lerneinheit auf die ganze Gruppe und nicht bloss auf eine Auswahl davon (wie zum Beispiel bei den Feedbacks, Abschnitt 8.6.1) hat.

Folgende Fragen stehen für eine Entwicklerin oder einen Entwickler einer Lerneinheit im Zentrum:

- Haben die Lernenden durch die Arbeit mit der Lerneinheit die wesentlichen Konzepte verinnerlicht und ausdifferenziert?
- Wurden die Konzepte durch die Arbeit mit den Problemen richtig verstanden?

Die Informationen werden aus Beobachtungen und aus den Gesprächen mit den Studierenden gewonnen. Viele gute Ideen für eine Verbesserung einer Lernumgebung sind in unseren Kursen während mündlicher Abgaben entstanden.

Weitere Details zur mündlichen Abgabe während der Phase *EXPLAIN* siehe Kapitel 9.4.5 „Leistungskontrollen und –nachweise organisieren und durchführen“.

2. Daten der Leistungskontrolle

Am Ende eines Semesters findet jeweils ein unbenoteter Test statt (siehe Abb. 9.2). Die Lernenden müssen dabei anhand von anwendungsorientierten Fragen beweisen, dass sie in einem Konzept über anwendbare Wissensstrukturen verfügen, indem sie eine Reihe von Behauptungen zu einem gegebenen Problem als richtig oder falsch einstufen. Zur Durchführung dieses Tests steht ein elektronisches Assessmentsystem zum effizienten Prüfen grosser Studierendengruppen zur Verfügung [Bosia 2005] (Weitere Details zur Leistungskontrolle siehe Kapitel 10.3 „Anwendungsorientierte Leistungskontrolle als Mass für das Vorhandensein von anwendbaren Wissensstrukturen“).

Abb. 8.14 A zeigt eine solche anwendungsorientierte Testfrage zum Konzept „Array“. In Abb. 8.14 B ist der Output dieser Aufgabe zu sehen. Das Item B wurde am besten (89%) und Item E am schlechtesten (46%) gelöst. Mit Hilfe der Resultate der Leistungskontrolle kann man sich ein Bild darüber verschaffen, welche Flexibilität die Lernenden nach dem Unterricht im Umgang mit dem Konzept verfügen.

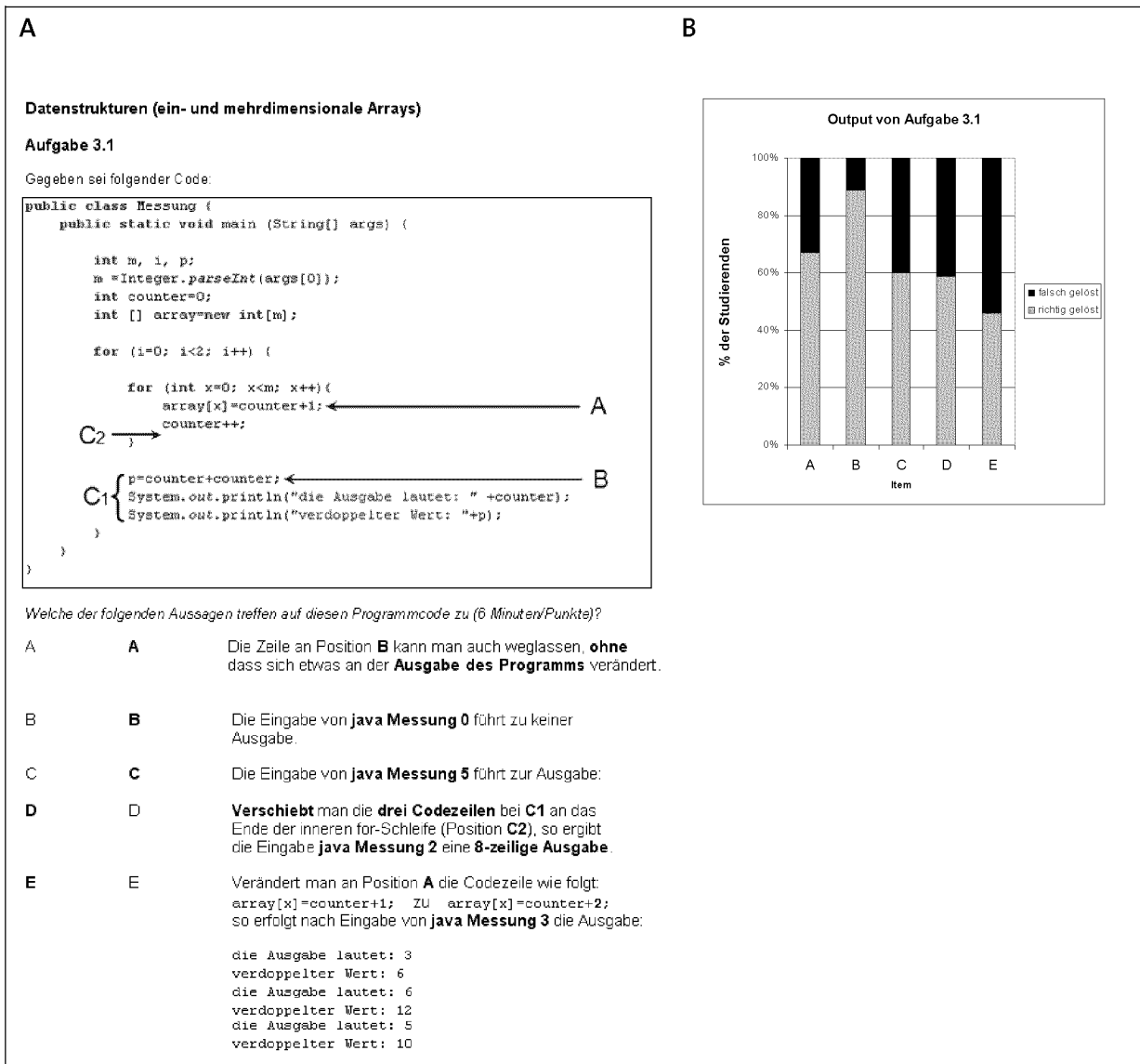


Abb. 8.14: A: Anwendungsorientierte Prüfungsfrage zum Konzept „Array“. Die korrekten Antworten sind fett geschrieben. B: Output der Prüfungsfrage (N=109). [Semesterendetest Kurs „Anwendungsnahe Programmieren“ Sommersemester 2007].

8.6.3 Prozess D3: Implementierung der Lerneinheit in den Gesamtkurs

Bei einem problembasierten Kurs führt eine neu erstellte Lerneinheit zu einer Teilkompetenz, die einen Aspekt der gesamten Kurskompetenz darstellt (folglich Kapitel 7 „Didaktische Modelle“, Abb. 7.1). Die einzelnen Lern- oder Unterrichtseinheiten müssen in einem problembasierten Kurs aufeinander abgestimmt sein. Die Qualität der Imple-

mentierung einer Lerneinheit in Bezug auf einen ganzen Kurs kann mit einer Prozessanalyse erfasst werden [Faessler 2006].

Bei einer Prozessanalyse werden bei einer Reihe von Variablen Änderungen erfasst (Details zur Prozessanalyse siehe Kapitel 10.2.2 „Datenerhebungen im Unterricht“ und 4.4 „Durchführung einer Prozessanalyse“). Für eine Lerneinheit wird zu folgenden Variablen je ein Wert erfasst:

- Zeitaufwand
- Schwierigkeitsgrad
- Motivation

Der Zeitaufwand wird als Absolutwert erfasst. Der eingeschätzte Schwierigkeitsgrad und die Motivation werden als Prozesswerte erfasst, indem die Studierenden ihre subjektiv eingeschätzte Veränderung der Werte in Bezug auf die jeweils vorangehende Lerneinheit angeben (Tab. 8.14).

Frage	Mögliche Antworten	Interpretation
1. Wie lange arbeiteten Sie mit diesem <i>E.Tutorial</i> ®?	<2h, 2-4h, 4-6h, 6-8h, >8h	Diese Grösse gibt den individuell investierten Aufwand der Studierenden wieder.
2. Wie stufen Sie den Schwierigkeitsgrad dieses <i>E.Tutorials</i> ® ein?*	einfach, eher einfach, gerade richtig, eher schwierig, schwierig	Die subjektiv empfundene Schwierigkeit ist eine wichtige Rückmeldung für die Anpassung des Anforderungsniveaus.
3. Wie stufen Sie den Schwierigkeitsgrad dieses <i>E.Tutorials</i> ® im Vergleich zum vorherigen ein?	viel tiefer, eher tiefer, in etwa gleich, höher, viel höher	Im Gegensatz zur Frage 2 zielt diese nicht auf absolute Werte, sondern auf die relativen Einschätzungen in Bezug auf das Vergangene <i>E.Tutorial</i> ®.
4. Wie gerne haben Sie an diesem <i>E.Tutorial</i> ® im Vergleich zum letzten gearbeitet?	viel weniger gerne, weniger gerne, gleich gerne, lieber, viel lieber	Kann als Veränderung der Motivation interpretiert werden.

Tab. 8.14: Fragen der Prozessanalyse, die nach jedem *E.Tutorial*® gestellt wurden [Bauer-Messmer 2005, Faessler 2006]. *Wird für das erste *E.Tutorial*® an Stelle von Frage 3 gefragt.

Die Prozessanalyse wurde mittels eines Online-Fragebogens am Ende des *E.Tutorials*® als HTML-Formular direkt ins Instruktionfenster eingebaut (siehe Abb. 8.13, A). Die Antworten wurden kontinuierlich gesammelt und in einer Log-Datei gespeichert (Abb. 8.13, B). Die Studierenden wurden aufgefordert, ihre Antworten mit einem mnemonischen Kürzel zu identifizieren, um die Anonymität der Umfrage zu gewährleisten. Die Antworten der Studierenden können so über mehrere Lerneinheiten miteinander in Bezie-

hung gebracht werden. Das detaillierte Erfassen und Interpretieren der Prozessanalyse ist Thema des Kapitels 10 „Evaluation der Lernumgebungen“. Hier sollen die Resultate zur Illustration des Vorgehens erwähnt werden.

Resultate der Prozessanalyse für das *E.Tutorial*[®] 3 („Arrays“ und „Zufallszahlen“)

In Abb. 8.15 sind die Resultate der Prozessanalyse für den Kurs „Anwendungsnahes Programmieren“ dargestellt. Die Markierung zeigt die Werte für *E.Tutorial*[®] 3 zu den Konzepten „Zufallszahlen“ und „Arrays“. Sowohl die aufgewendete Zeit, als auch der eingeschätzte Schwierigkeitsgrad nahm im Vergleich zum vorigen *E.Tutorial*[®] zu. Dies hat vermutlich damit zu tun, dass die Umsetzung dieser Konzepte generell als schwierig empfunden wird und die wenigsten Studierenden das Konzept vorher bereits gekannt haben. Die Lernenden arbeiten mit diesem *E.Tutorial*[®] aber auch lieber als mit dem vorangehenden. Diese Zunahme der Motivation ist wahrscheinlich darauf zurückzuführen, dass viele Grundlagen aus den ersten beiden *E.Tutorials*[®] in im *E.Tutorial*[®] 3 in einem umfangreicheren Problem (Geburtstags-Paradoxon, siehe Tab. 8.6) angewendet werden können, was generell deren Beliebtheit erhöht, vorausgesetzt, die Studierenden haben sich das Wissen der zu Grunde liegenden Konzepte zuvor angeeignet (Details zur Prozessanalyse siehe Kapitel 10.4 „Überprüfung der Implementierungsqualität in problembasierten Kursen“).

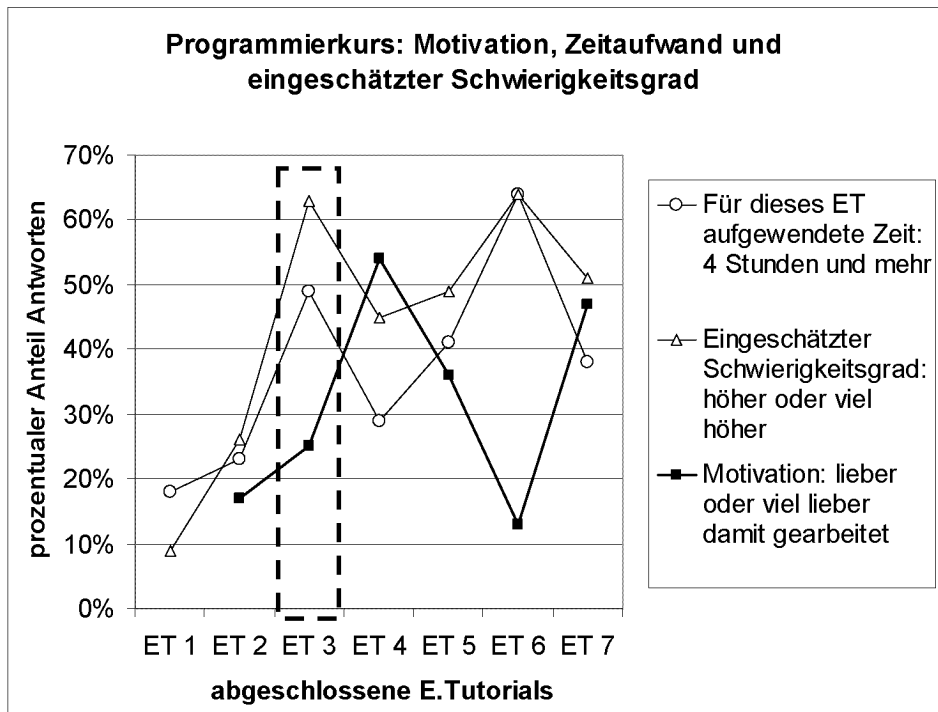


Abb. 8.15: Prozessanalyse des Programmierkurses (292 Studierende, mittlere Rücklaufquote: 75%). Die Markierung zeigt die Werte für das in diesem Kapitel besprochene *E.Tutorial*[®] 3. In den *E.Tutorials*[®] behandelte Konzepte: ET1: Variablen, Bedingungsprüfung, ET2: Iteration, ET3: Zufallszahlen, Arrays, ET4: Simulation (Galton Board), ET5: Objekte, Liniengrafik, ET6: GUI-Komponenten, ET7: Modellieren, Software-Entwicklung. Der Aufbau des Kurses und die Lösungen der Aufgaben sind dem Anhang zu entnehmen.

8.7. Fazit

In diesem Kapitel wurde ein Szenario für die Entwicklung von Lerneinheiten zum Aufbau einer kompetenz- und problemorientierten Lernumgebung am Beispiel eines Programmierkurses für Naturwissenschaftlerinnen und Naturwissenschaftler beschrieben. Die Umsetzung ist mit einem hohen Aufwand und den entsprechenden Entwicklungskosten verbunden [Hinterberger 2007]. Ist die Lernumgebung aber einmal erstellt, zeichnet sie sich durch Stabilität und einen hohen Automatisierungsgrad des Grundlagenunterrichts aus.

Durch die Entwicklung kompetenzorientierter Lerneinheiten haben wir aber nicht nur die Qualität des Unterrichts verbessert, sondern auch die Erwartungen der Lernenden an den Unterricht erhöht. Wir mussten dies in jenen Jahren feststellen, als der ICT-Kurs bereits kompetenzorientiert und selbstgesteuert, der Programmierkurs (dessen Entwicklung in diesem Kapitel beschrieben wurde) aber noch in traditioneller Form (mit Vorlesungen und Übungen im Präsenzunterricht) gehalten wurde. Die Lernenden hatten sich durch den einführenden ICT-Kurs an die Kompetenzorientierung und den hohen Grad an Selbststeuerung gewöhnt. Als dann im nachfolgenden Programmierkurs keine Lernmaterialien derselben Qualität mehr zur Verfügung standen und die

Lernenden wieder vermehrt einer Fremdsteuerung ausgesetzt waren, forderten sie eine qualitative Verbesserung der Lehrveranstaltung. In einer Nachumfrage zum Kurs gaben über 90% der Studierenden an, die Übungen lieber mit *E.Tutorials*® lösen zu wollen [Unterrichtsevaluation Kurs „Anwendungsnahes Programmieren 2002, <http://et.ethz.ch>]. Mehr zum Umstrukturierungsprozess von einer traditionellen zu einer kompetenzorientierten Blended Learning Umgebung ist dem Kapitel 10 „Evaluation der Lernumgebungen“ zu entnehmen.

Zusammenfassend halten wir fest, dass wir durch das in diesem Kapitel beschriebene Educational Engineering nicht nur die Kompetenz- und Problemorientierung umsetzen, sondern durch die unterrichtsnahe Entwicklung auch machbare und realistische Anforderungen generiert werden können, was die Effektivität und Effizienz der Lernmaterialien garantiert und schlussendlich die Motivation der Lernenden nachhaltig positiv beeinflusst.

Einsatz, Evaluation und Weiterentwicklung

Kapitel 9

Einsatz der computergestützten Lernumgebung im Unterricht

9.1. Einleitung

Im letzten Kapitel wurde besprochen, wie Lernmaterialien für den Unterricht im 4-Schritte-Modell entwickelt werden. Auf diese Weise ist ein Pool an Lernmaterialien entstanden (und wird laufend ausgebaut), der möglichst breit eingesetzt und zum Unterrichten von verschiedenen Kursen mit unterschiedlichen Zielgruppen zur Verfügung stehen soll.

Die entwickelten Lernmaterialien für das 4-Schritte-Modell zeichnen sich dadurch aus, dass sie dank der hohen Eigenaktivität der Lernenden und der qualitativ detaillierten Aufbereitung der Lerninhalte zu einem hohen Automatisierungsgrad und zu Stabilität der Lernumgebung führen. Es hat sich aber gezeigt, dass die entwickelten Lernmaterialien trotz der hohen Selbststeuerung der Lernenden nur zum gewünschten Lernerfolg führen, wenn sie in einen Unterricht eingebettet sind, bei dem *Ablaufstrukturen* und *Betreuungspersonen* den Unterricht im 4-Schritte-Modell optimal unterstützen (siehe durchgeführte Evaluation in Kapitel 10 „Evaluation der Lernumgebungen“). Vor allem bei grossen Klassen ist es für die Qualität des Unterrichts entscheidend, wie diese Elemente eines Lernsystems zur Durchführung von Kursen zum Einsatz kommen und zusammenspielen.

In diesem Kapitel soll beschrieben werden, wie Kurse, die nach dem 4-Schritte-Modell unterrichtet werden, idealerweise zu organisieren sind, welche Ablaufstrukturen dafür zur Verfügung gestellt und wie Betreuungspersonen der Situation angepasst einzusetzen sind. Im Abschnitt 9.3 wird ein Überblick über Aufgaben, Personen und Ablaufstrukturen des Unterrichts im 4-Schritte-Modell gegeben, in Abschnitt 9.4 folgt eine detaillierte Beschreibung der Aufgaben im zeitlichen Ablauf eines Kurses.

9.2. Kursimplementierung: von den Lernmaterialien zum operativen Kurs

Werden die in einem Pool gesammelten Lernmaterialien für den Unterricht im 4-Schritte-Modell Lernenden zur Verfügung gestellt, kann man noch nicht von Unterricht

oder einem Kurs sprechen. Die Durchführung eines Kurses im 4-Schritte-Modell erfordern

- die Unterstützung von Abläufen mit diversen Hilfsmitteln
- den Einsatz von Betreuungspersonen.

Das Ziel ist ein effizienter und möglichst individualisierter Ablauf des Unterrichts, auch bei grossen Klassen (Abb. 9.1).

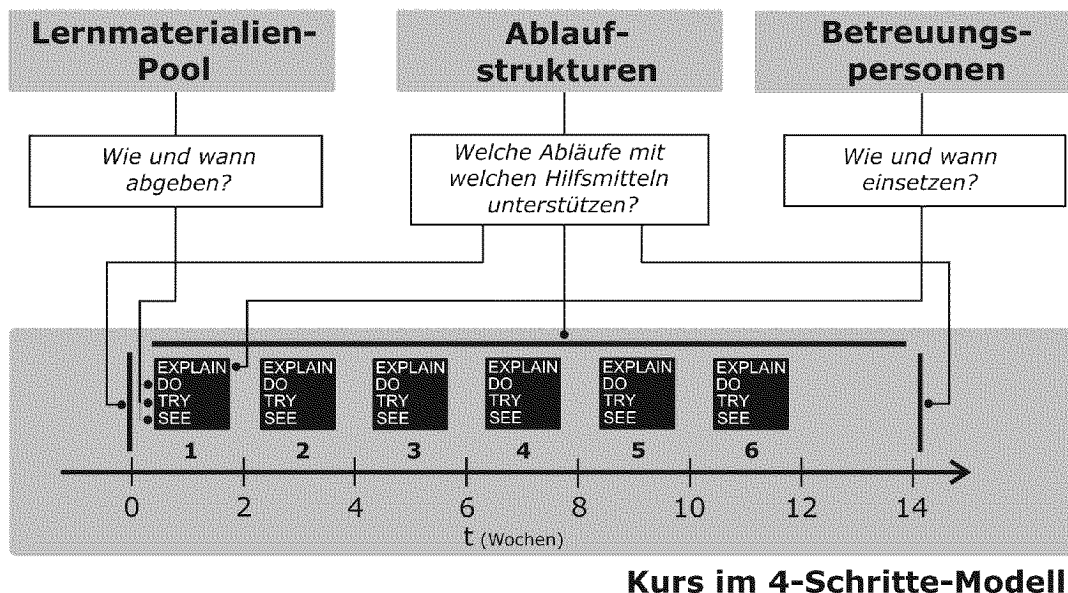


Abb. 9.1: Die effiziente Durchführung von Kursen nach dem 4-Schritte-Modell ist davon abhängig, wie und wann Lernmaterialien für die Phasen *SEE*, *TRY*, *DO* aus dem Pool abgegeben, Ablaufstrukturen durch Hilfsmittel unterstützt und Betreuungspersonen während der Phase *EXPLAIN* eingesetzt werden können.

9.2.1 Ablaufstrukturen

Damit die in einem Pool gesammelten Lernmaterialien möglichst flexibel und effizient zur Durchführung verschiedener Kurse eingesetzt werden können, sind sie bewusst so entwickelt worden, dass sie keine kursspezifischen Informationen enthalten. Um aber mit den Lernmaterialien einen Kurs im 4-Schritte-Modell durchzuführen, müssen sie mit Ablaufstrukturen in Verbindung gebracht werden, welche kursspezifische Daten und Informationen zur Verfügung stellen, wie Angaben über:

- Informationen über Inhalt und Ablauf eines spezifischen Kurses
- Daten der für einen Kurs immatrikulierten Personen
- Informationen zur Durchführung der Qualitätskontrolle eines spezifischen Kurses.

Getrennte Entwicklung von Lernmaterialien und Ablaufstrukturen garantieren einen flexiblen Einsatz

Das Ziel der Entwicklung einer Lernumgebung ist es, dass sich einzelne Lernmodule unterschiedlich kombinieren lassen und so Kursprogramme für verschiedene Bedürfnisse zusammengestellt werden können. Dies erfordert, dass die Lernmaterialien für das 4-Schritte-Modell so aufgebaut sind, dass sie möglichst vielfältig (Institutionen, Zielgruppen etc.) einsetzbar sind. Lernmaterialien und Ablaufstrukturen sind deshalb konsequent getrennt voneinander zu entwickeln, um die notwendige Flexibilität, die der Einsatz der Lernmaterialien für verschiedene Zielgruppen und Institutionen erfordert, zu gewährleisten.

Die Einbettung der Lernmaterialien in Unterrichtsstrukturen ist ein wichtiger Implementierungsprozess, der für jeden Kurs wieder neu und unabhängig von einer möglicherweise gleichzeitigen Entwicklung der Lernmaterialien durchgeführt werden muss. Die Organisation von Ablaufstrukturen eines durchgeführten Kurses ist mit Optimierungsprozessen verbunden, die sich nach den Eigenschaften eines Kurses richtet und nicht nach Lerninhalten.

Optimieren von Ablaufstrukturen für verschiedene Kurseigenschaften

Kurse können sich in folgenden Punkten unterscheiden: Zielgruppe, Dauer, Kreditpunkte, Klassengrösse und die Möglichkeiten, Betreuungspersonal zur Verfügung zu stellen. Die von uns durchgeführten Kurse an der ETH Zürich haben alle in etwa folgende Eigenschaften:

- Zielgruppe: Studierende der Natur- oder Bauingenieurwissenschaften im Grundstudium
- Dauer: 14 Semesterwochen
- Kreditpunkte: 2-3 ECTS (ca. 60-90 Arbeitsstunden)
- Klassengrösse: 140-280 Studierende
- Betreuungspersonal: 1 Assistenzperson pro 20 Studierende

Anhand der Kurseigenschaften ist bei der Implementierung darüber zu entscheiden, wie die Ablaufstrukturen in einem Kurs mit geeigneten Hilfsmitteln unterstützt werden können, damit der Unterricht im 4-Schritte-Modell optimal und effizient ablaufen kann. Ein für die Organisation von Ablaufstrukturen entscheidender Punkt ist sicherlich die Klassengrösse. Ein Kurs mit 280 Studierenden setzt ganz andere Anforderungen an die Organisation und die damit verbundene Auswahl der Hilfsmittel, als ein Kurs mit nur 25 Studierenden.

Die Auswahl der Hilfsmittel ist an spezifische Aufgaben gebunden, die zur Durchführung eines Kurses im 4-Schritte-Modell auszuführen sind. In Abschnitt 9.3 wird eine Übersicht über diese Aufgaben gegeben und in Abschnitt 9.4 folgt die Umsetzung der Aufgaben.

9.2.2 Rollen und Aufgaben der am Unterricht im 4-Schritte-Modell beteiligten Personen

Seit der Einführung des computergestützten Unterrichts im 4-Schritte-Modell haben sich die Rollen der am Unterricht beteiligten Personen grundlegend verändert. Traditionell ist die universitäre Wissensvermittlung an eine Vorlesung mit einer Dozentin oder einem Dozenten gebunden und in den Übungsstunden müssen die Assistierenden mit einer Gruppe von zugeteilten Studierenden die Lösung von gestellten Aufgaben besprechen. Die Lernenden sind bei der Wissensaneignung direkt von den eingesetzten Lehrpersonen und deren Instruktionstempo abhängig. Für die Studierenden ist es entscheidend, bei wem sie eine Vorlesung besuchen und bei welcher Assistenzperson sie eingeteilt sind. Im Gegensatz dazu gibt eine Individualisierung des Unterrichts durch die selbstgesteuerten, problembasierten Instruktionen allen Studierenden die gleichen Voraussetzungen und sie sind nicht mehr in gleicher Masse von den eingesetzten Lehrpersonen abhängig. Trotzdem werden auch nach der Entwicklung der Lernumgebung noch Präsenzveranstaltungen und Übungsstunden durchgeführt, diese erfüllen heute allerdings neue Aufgaben, die ebenfalls in diesem Kapitel diskutiert werden sollen.

Die Qualität des individualisierten Unterrichts im 4-Schritte-Modell ist direkt von der Art und Weise abhängig, mit der die Leistungsnachweise der Phase *EXPLAIN* in Form mündlicher Abgabegespräche durchgeführt werden (siehe Abb. 9.1). Wird diese anspruchsvolle Aufgabe von Assistenzpersonen übernommen, hat es sich gezeigt, dass durch ein Monitoring die Qualität des Unterrichts gesteigert und die Assistierenden in den Unterricht eingebunden werden können.

Für Durchführung von Kursen im 4-Schritte-Modell muss deshalb auch der Frage nachgegangen werden, wie in dieser hoch individualisierten Form des Unterrichts Betreuungspersonen eingebunden werden, die im Ablauf des Unterrichts eine wichtige Aufgabe übernehmen, an der Entwicklung der Lernumgebung aber nicht notwendigerweise beteiligt sind.

9.3. Übersicht über Aufgaben, Personen und Ablaufstrukturen für das Unterrichten eines Kurses im 4-Schritte-Modell

9.3.1 Aufgaben

In Tab. 9.1 sind alle Aufgaben aufgeführt, welche zur Durchführung eines Kurses im 4-Schritte-Modell geleistet werden müssen. Details zu den Aufgaben und deren zeitlichen Abläufen werden in Abschnitt 9.4 beschrieben.

Bez.	Aufgaben	Abschnitt
A	Kursspezifische Informationen bereitstellen	9.4.2
B	Lernmaterialien bereitstellen, verteilen und unterhalten	9.4.3
C	Assistenzpersonen-Monitoring durchführen	9.4.4
D	Leistungskontrolle und -nachweise organisieren und durchführen	9.4.5
E	Qualitätskontrolle eines Kurses durchführen	9.4.6
F	Präsenzveranstaltungen organisieren und halten	9.4.7

Tab. 9.1: Durchzuführende Aufgaben für den Unterricht im 4-Schritte-Modell. Zeitlicher Ablauf der Aufgaben siehe Abb. 9.2 und Tab. 9.4.

9.3.2 Personen und Verantwortungen

Um die Aufgaben aus Tab. 9.1 auszuführen, kommen zur Durchführung von Kursen im 4-Schritte-Modell abhängig von den Kurseigenschaften (Klassengrösse, Anzahl verfügbare Betreuungspersonen etc.) eine bestimmte Anzahl von Personen zum Einsatz, die im Unterricht verschiedene Verantwortungen wahrnehmen. Tab. 9.2 zeigt die Verteilung der Verantwortungen und Aufgaben in den von uns durchgeführten Kursen an der ETH Zürich.

Personen	Verantwortungen/Aufgaben
Dozierende/r	<i>Ist für den ganzen Kurs verantwortlich</i> -Definiert Kursinhalt, Lernziele und Leistungskontrolle. -Organisiert und hält Präsenzveranstaltungen.
Koordinator/in Übungsbetrieb	<i>Ist für die Organisation und Durchführung des Übungsbetriebes verantwortlich.</i> -Führt das Monitoring der Assistenzpersonen durch. -Stellt Lernmaterialien bereit und unterhält sie. Stellt Informationen zum Kurs bereit. -Organisiert und verwaltet die Leistungskontrollen.
Assistierende/r	<i>Ist als Coach für den Kontakt mit den Studierenden während der Phase EXPLAIN verantwortlich.</i> -Führt Leistungsnachweise durch und ist im Computerraum präsent.

Tab. 9.2: Verantwortungen und Aufgaben der am Unterricht im 4-Schritte-Modell beteiligten Personen in den von uns durchgeführten Kursen an der ETH Zürich.

9.3.3 Ablaufstrukturen

Geeignete Ablaufstrukturen helfen bei Kursen mit grossen Studierendenzahlen die Aufgaben aus Tab. 9.1 effizienter und einfacher zu lösen. Tab. 9.3 zeigt die eingesetzten Ablaufstrukturen zur Ausführung verschiedener Aufgaben aus Tab. 9.1 in den von uns durchgeführten Kursen an der ETH Zürich. Falls möglich werden Strukturen verwendet, welche in der Institution bereits zur Verfügung stehen. Andernfalls müssen neue Strukturen geschaffen werden.

	Aufgaben (Bez. aus Tab. 9.1)	Ablaufstruktur
Bereitstellen von kursspezifischen Informationen	Generelle Informationen zum Kurs (A) Zugang zu Lernmaterialien über das WWW (B) Einsicht in die Leistungskontrolle (D)	Kurs-Webseite (für Studierende und Assistierende)
	Eingabe der Leistungsnachweise (D) Musterlösungen der Problemaufgaben (C) Informationen und Materialien zur Assistenz (C)	Kurs-Webseite (für Assistierende)
Kommunikation	Kommunikation mit Assistierenden (C) Kommunikation mit Studierenden (A)	E-Mail*
Studierendenverwaltung	Verwaltung der Leistungsnachweise der Studierenden in einer Online-Datenbank (D)	Datenbank mit Anbindung an Lehrbetriebsapplikation*
Qualitätskontrolle	Qualitätskontrolle (Vor- und Nachumfragen) (E)	Umfrageapplikation*

Tab. 9.3: Eingesetzte Ablaufstrukturen zur Ausführung der Aufgaben aus Tab. 9.1 für den Unterricht im 4-Schritte-Modell in den von uns durchgeführten Kursen an der ETH Zürich. *Von der Institution ETH Zürich zur Verfügung gestellt.

9.4. Zeitlicher Ablauf und Umsetzung der Aufgaben von Kursen im 4-Schritte-Modell

9.4.1 Überblick

In diesem Abschnitt werden der Ablauf eines Kurses und die Aufgaben aus Tab. 9.1 näher beschrieben. Abb. 9.2 und Tab. 9.4 zeigen den zeitlichen Ablauf der Aufgaben für das Unterrichten im 4-Schritte-Modell. Zeitlich können bei der Durchführung eines Kurses folgende drei Phasen unterschieden werden: Vorbereitungsphase, Unterrichtsphase und Assessmentphase.

1. Vorbereitungsphase (Abb. 9.2, I)

Während der Vorbereitungsphase zu einem Kurs, die etwa einen Monat vor Unterrichtsbeginn einsetzt, müssen alle Informationen und Materialien zum Kurs bereitgestellt, die Computerraumpräsenz der Assistierenden der Phase *EXPLAIN* organisiert und die Studierendenverwaltung vorbereitet werden. Voraussetzung dafür ist, dass die Lernmaterialien für den Kurseinsatz bereitstehen (siehe Kapitel 8 „Entwicklung und Umsetzung kompetenzorientierter Lerneinheiten“). Während der Vorbereitungsphase wird in der Regel kein Unterrichtsinhalt mehr erstellt (siehe Qualitätskontrolle, Abschnitt 9.4.6).

2. Unterrichtsphase (Abb. 9.2, II)

Bei Unterrichtsbeginn (Woche 1) muss eine ganze Reihe von Aufgaben erledigt werden. Neben dem Verteilen von Materialien an die Studierenden werden die Assistierenden auf ihre Aufgabe während dem Kurs vorbereitet und begleitet. Die Dozierenden halten eine erste Präsenzveranstaltung, die anschliessend alle 14 Tage stattfindet. Während dem Semester übernehmen die Assistierenden die Hauptarbeit des Unterrichtsbetriebs in den Computerräumen und verwalten die Leistungsnachweise der Studierenden.

3. Assessmentphase (Abb. 9.2, III)

Am Ende des Semesters (vorletzte Semesterwoche) ist die Wissensvermittlung abgeschlossen, und es beginnt die Assessmentphase mit der Durchführung eines Semesterendetests. Dieser unbenotete Test auf Prüfungsniveau ist einerseits eine inhaltliche Kurszusammenfassung und dient andererseits der Prüfungsvorbereitung. Ein geprüfter Kurs wird mit einer benoteten Prüfung, die in der jeweiligen Prüfungssession stattfindet, abgeschlossen.

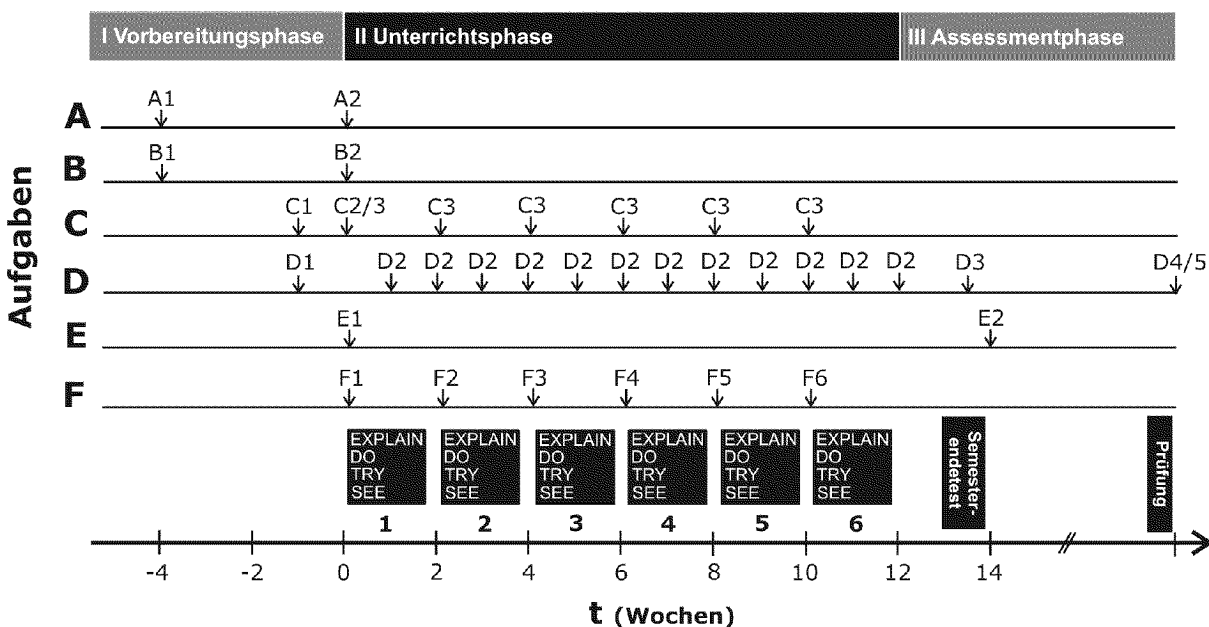


Abb. 9.2: Ablauf der verschiedenen Aufgaben zur Durchführung eines Kurses im 4-Schritte-Modell während eines Semesters. Details zu den Aufgaben siehe Abschnitte 9.4.2 bis 9.4.7 und Tab. 9.4. Abkürzungen: A: Kursspezifische Informationen bereitstellen, B: Lernmaterialien bereitstellen und verteilen, C: Assistenzpersonen-Monitoring durchführen, D: Leistungskontrolle und -nachweise organisieren und durchführen, E: Qualitätskontrolle eines Kurses durchführen, F: Präsenzveranstaltungen organisieren und halten.

Bez.	Aufgabe	Ablaufstruktur/Hilfsmittel/Form
I Vorbereitungsphase		
<i>Ca. 4 Wochen vor Unterrichtsbeginn</i>		
A1	Kursinformationen bereitstellen	Kurs-Webseite
B1	Lernmaterialien zum Kurs zusammenstellen	USB-Memory-Sticks, Druckmaterialien
C1	Einteilung der Assistierenden	
D1	Studierendenverwaltung	Datenbank, Webseite
II Unterrichtsphase		
<i>Semesterbeginn (Woche 1)</i>		
A2/E1	Vorumfrage bei den Studierenden	E-Mail, Umfrageapplikation
B2	Verteilung / Verkauf Lernmaterial	USB-Memory-Sticks, Druckmaterialien
C2	Einführung der Assistierenden	Meeting
<i>Laufendes Semester (Wochen 2-13)</i>		
C3	Monitoring der Assistierenden	E-Mail (alle 2 Wochen) und Betreuung im Computerraum
D2	Durchführung und Verwaltung von Leistungsnachweisen	Präsenz im Computerraum
F1-6	Durchführung von Präsenzveranstaltungen	Vorlesungssaal (alle 2 Wochen)
III Assessmentphase		
<i>Semesterabschluss (Woche 14)</i>		
D3	Semesterendetest (unbenotet)	Elektronisches Prüfungssystem
E2	Nachumfrage	Umfrageapplikation (<i>Survey</i>)
<i>Prüfungssession</i>		
D4	Prüfung (benotet)	Druckmaterialien
D5	Kreditpunkte und Noten erteilen	Lehrbetriebsapplikation für Dozierende (<i>eDoz</i>)

Tab. 9.4: Zusammenfassung des Ablaufs eines Kurses im 4-Schritte-Modell. Details zu den Aufgaben siehe Abschnitte 9.4.2 bis 9.4.7 und Abb. 9.2.

9.4.2 Kursspezifische Informationen bereitstellen (Aufgabe A)

1. Publizieren kursspezifischer Information

Für das Publizieren kursspezifischer Informationen wird für jeden Kurs eine eigene Webseite unterhalten. Dabei hat sich ein möglichst einfach zu aktualisierendes Layout bewährt, das sich effizient aus einer programmierten Vorlage erstellen lässt und fol-

gende Informationen und Zugänge enthält (für ein Beispiel einer Kurs-Webseite siehe <http://evim.ethz.ch>):

- Allgemeine Informationen und Aufbau des Kurses (Lernziele, Agenda, Kontakt, Bedingungen für Kreditpunkte, häufig gestellte Fragen, Prüfung)
- Materialien zur Vorlesung
- Zugang zu den Lernmaterialien (Link zu den *E.Tutorials*® im Lernmaterialien-Pool)
- Persönlicher Zugang über alle Leistungsnachweise (Testate)
- Passwortgeschützter Assistenzbereich mit allen Unterlagen zur Assistenz (siehe Abschnitt 9.4.5)

2. Kommunikation mit Studierenden und Assistierenden

Die Kommunikation mit den Studierenden und Assistierenden wird grösstenteils per E-Mail und während Präsenzveranstaltungen erledigt (siehe weiter unten). Die Verwendung zusätzlicher Kommunikationsplattformen ist nicht notwendig und auch nicht empfehlenswert, weil sie von den Studierenden kaum besucht und genutzt werden.

Die Studierenden erhalten nur bei Kursbeginn und -abschluss jeweils eine E-Mail mit einer Begrüssung, bzw. Verabschiedung und einem Link zur entsprechenden Online-Umfrage (siehe Abschnitt 9.4.6). Die Studierenden wenden sich bei Fragen und Problemen spontan per E-Mail bei den Assistierenden. Hierfür sind die E-Mailadressen der Assistierenden auf der Webseite zu publizieren.

Die Assistierenden erhalten zu Beginn jedes Lernmoduls per E-Mail Informationen zu Inhalt, Zielsetzungen, Abgabebedingungen und häufigen Fehlerquellen eines Lernmoduls (siehe Abschnitt 9.4.4).

9.4.3 Lernmaterialien bereitstellen, verteilen und unterhalten (Aufgabe B)

Die entwickelten Lernmaterialien werden in einem Pool zentral gespeichert, damit sie unterhalten und zur Durchführung verschiedener Kurse verwendet werden können (siehe Abschnitt 9.2.1). Um die Individualisierung nicht zu behindern, werden zu Beginn eines Kurses alle Lernmaterialien zusammen abgegeben. Auf diese Weise können die Lernenden die Lerneinheiten zeitlich flexibel bearbeiten. Für die Abgabe der Übungen während der Phase *EXPLAIN* hat sich zur Bewältigung von grossen Klassen hingegen die Vorgabe von Zeitfenstern bewährt (siehe Abschnitt 9.4.4).

Zur Vorbereitungsphase jedes Kurses gehört das Zusammenstellen von Lernmaterialien aus dem Lernmaterialien-Pool. Dies erfolgt durch ein Zusammenklicken der gewünschten Lernmodule aus dem Lernmodule-Pool (Abb. 9.3). Diese Aufgabe kann

aufgrund der getrennten Behandlung von Lernmaterialien und Ablaufstrukturen sehr schnell und effizient ausgeführt werden.

Zu den Lernmaterialien des 4-Schritte-Modells gehören:

- 1. Papierunterlagen der Phasen *SEE* und *DO* (Druckmaterialien in einem Kursordner)
- 2. *E.Tutorials*® der Phase *TRY* (elektronisch)

Details zu den Medien siehe Kapitel 7 „Didaktische Modelle“.

1. Druckmaterialien der Phasen *SEE* und *DO*

Die Unterrichtsunterlagen werden den Studierenden aus Kosten- und Umweltschutzgründen Anfang Semester gedruckt und zusammen mit einem Kursordner abgegeben. Es hat sich aus medienergonomischen Gründen nicht bewährt, die Papierunterlagen als elektronische Dokumente zu verteilen, beziehungsweise zur Verfügung zu stellen (siehe Kapitel 7.2.3 „Einsatz von Medien für das 4-Schritte-Modell“).

2. *E.Tutorials*® der Phase *TRY*

Die *E.Tutorials*® werden den Studierenden über zwei Wege angeboten (Abb. 9.3):

- Über das WWW, aufrufbar über die Kurs-Webseite
- Auf USB-Memory-Stick

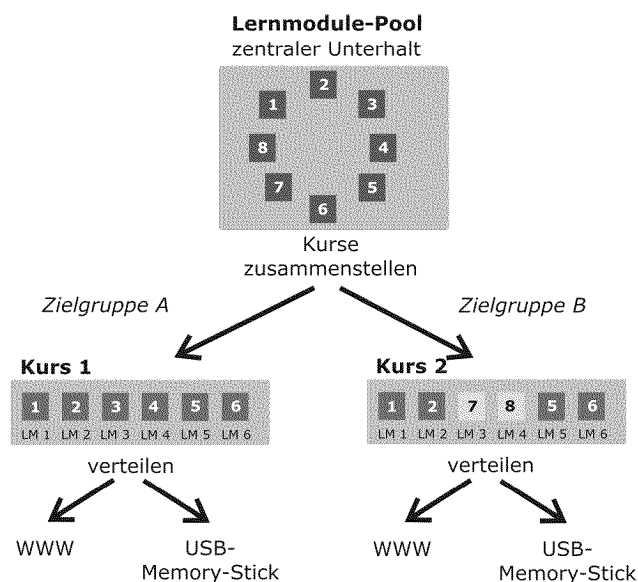


Abb. 9.3: Verschiedene Lernmodule werden in einem Pool zentral unterhalten und jeweils vor Kursbeginn zu einem Kurs für eine bestimmte Zielgruppe zusammengestellt. Die Verteilung erfolgt sowohl über das WWW als auch über die Abgabe von USB-Memory-Sticks.

Lernmaterialien auf USB-Memory-Stick

Die elektronischen Lernmaterialien auf USB-Memory-Stick abzugeben, hat gegenüber dem Verteilen über das WWW folgende Vorteile:

- Die Übungen können auch an Orten ohne Internetverbindung bearbeitet werden.
- Auf den USB-Memory-Sticks können nicht nur die Lernmaterialien abgespeichert werden, sie können auch als Speichermedien für die während der Phase *DO* zu erstellenden „Lernprodukte“ genutzt werden.

Bei einer durchgeführten Umfrage zum Gebrauch der USB-Memory-Sticks gab ein grosser Teil der Studierenden an, in einem nächsten Kurs nicht darauf verzichten zu wollen, auch wenn die Lernmaterialien gleichzeitig über das WWW verfügbar waren (Abb. 9.4). Ein ähnliches Phänomen konnten wir auch in den Computerräumen beobachten. Viele haben USB-Memory-Sticks oder früher CD-ROM benutzt, obwohl die Lernmaterialien im Computerraum über das WWW zugänglich gewesen wären.

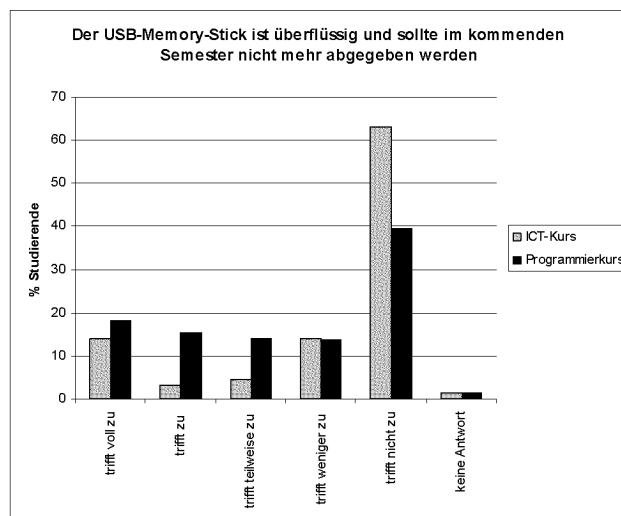


Abb. 9.4: Umfrage über die Verwendung von USB-Memory-Sticks als Verteil- und Speichermedium der Lernmaterialien in einem ICT- und einem Programmierkurs (ICT-Kurs: 2006, N=71, Programmierkurs: 2006, N=65). Die Unterschiede zwischen den beiden Kursen sind vermutlich darauf zurückzuführen, dass der ICT-Kurs im 1. und der Programmierkurs im 4. Semester durchgeführt wurde. Die Studierenden setzen im 4. Semester vermehrt eigene Laptops und nicht lokale Speichermedien ein (siehe Kapitel 3.4 „Ressourcenabklärung bei Informatik-Einführungskursen an der ETH Zürich“).

9.4.4 Assistenzpersonen-Monitoring durchführen (Aufgabe C)

Um den Unterricht im 4-Schritte-Modell auch mit grossen Klassen durchführen zu können, müssen Betreuungspersonen (Assistenzpersonen) eingesetzt werden. Weil sie die Personen sind, die bei dieser individualisierten Unterrichtsform am stärksten mit den Studierenden in Kontakt stehen, ist die Ausübung ihrer Aufgabe sowohl für den individuellen Lernerfolg der Lernenden als auch für die Qualität des gesamten Unter-

richts von grosser Bedeutung. Sie haben einen direkten Einfluss auf die Motivation der Lernenden und den Verlauf eines Kurses (folglich extrinsische Motivation, Kapitel 6.4.3 „Motivationale und emotionale Aspekte während der Instruktion“).

Neue und anspruchsvolle Aufgaben der Assistierenden erfordern ein Monitoring

Die Assistierenden haben beim Unterricht im 4-Schritte-Modell weder Einfluss auf die Auswahl des Lerninhalts, noch können sie den Lernenden beibringen, wie ein Problem aus ihrem Fach gelöst wird. Diese Aufgaben werden vollumfänglich von der Lernumgebung übernommen. Trotzdem übernehmen die Assistierenden als *Coach* oder *Guide on the side* eine wichtige und anspruchsvolle Aufgabe innerhalb des Lernsystems. Im Zentrum ihrer Tätigkeit steht die Kontrolle des Leistungsnachweises durch Abgabegespräche der Phase *EXPLAIN* des 4-Schritte-Modells (siehe nächster Abschnitt 9.4.5 sowie Kapitel 7 „Didaktische Modelle“). Es hat sich gezeigt, dass zur Ausübung dieser Aufgaben eine Informationsveranstaltung alleine nicht ausreicht. Damit die Assistierenden ihre Aufgabe als Teil des Lernsystems im Sinne des 4-Schritte-Modells ausführen, ist ein Monitoring notwendig.

Durch die Ausführung folgender Aufgaben werden die Assistierenden in den Unterricht eingebunden:

- Einführungsveranstaltung für alle Assistenzpersonen zu Beginn des Semesters (Aufgabe C2, Abb. 9.2)
- Monitoring der Assistierenden während des laufenden Semesters (via E-Mail und direkt in den Computerräumen) (Aufgabe C3, Abb. 9.2).

Die Einführungsveranstaltung regelt alles Organisatorische, erklärt die Aufgabengebiete und vermittelt eine erste Idee des Unterrichtskonzepts. Über die Inhalte der einzelnen Lernmodule werden die Assistierenden per E-Mail informiert. Als Teil des Monitorings werden die Assistierenden zu Beginn eines Kurses bei ihrer Arbeit im Computerraum direkt „on the job“ begleitet und ihre Aufgaben im Lernsystem in individuellen Gesprächen thematisiert.

In einem weiterführenden Projekt soll gezeigt werden, dass durch eine inhaltliche Beteiligung der Assistierenden am Unterricht helfen soll, sie besser ins Unterrichtssystem zu integrieren.

9.4.5 Leistungskontrollen und -nachweise organisieren und durchführen (Aufgabe D)

Während eines Kurses im 4-Schritte-Modell muss jeder einzelne Studierende folgende Leistungsnachweise und Leistungskontrollen erbringen, die von den Assistierenden durchgeführt werden:

- 1. Unterrichtsphase: Leistungsnachweise der einzelnen Lernmodule (individuelle Abgabegespräche Phase *EXPLAIN*) durchführen und die Leistungsnachweise verwalten (Aufgabe D2, Abb. 9.2)
- 2. Assessmentphase: Leistungskontrolle nach dem Unterricht (Aufgaben D3 und D4, Abb. 9.2).

1. Individuelle Abgabegespräche (Phase *EXPLAIN*)

Jedes bearbeitete Lernmodul ist mit einem individuellen Abgabegespräch von 5 bis 10 Minuten Dauer verbunden. Die Studierenden müssen hierfür einer Assistenzperson ihre während der Phase *DO* selbständig erarbeitete Aufgabe vorstellen (siehe Abb. 7.3, Kapitel 7 „Didaktische Modelle“). Ein solches Abgabegespräch hat nicht zum Ziel, die Studierenden abzufragen. Vielmehr sollen die Studierenden durch das Abgabegespräch zusammenfassen, was sie gemacht haben, und durch diese Reflexion aus den Detailfragen auf die allgemeine Konzeptebene zurückfinden. Ein Abgabegespräch besteht unter anderem aus folgenden Punkten:

- Was war das zu lösende Problem?
- Welche Konzepte waren für die Lösung des Problems beizuziehen und anzuwenden?
- Wie wurde das Problem gelöst?
- Welches waren im Problemlöseprozess die schwierigen Punkte und wie wurden sie bewältigt?

Details zu den Abgaben erfahren die Assistierenden aus den E-Mails zu den einzelnen Modulen (Aufgabe C3, Abb. 9.2) und stehen in den Abgabebedingungen zur Phase *DO* des 4-Schritte-Modells. Zur Bewältigung von grossen Klassen kann es notwendig werden, bestimmte Zeitfenster für eine Abgabe vorzugeben und zu erlauben, dass zwei Studierenden gleichzeitig bei derselben Assistenzperson die Arbeit abgeben. Bei den von uns durchgeführten Kursen von 14 Wochen Dauer dauert ein solches Abgabefenster für jedes Lernmodul jeweils 2 Wochen (siehe Abb. 9.1 und Abb. 9.2).

Übungsabgaben der Phase *EXPLAIN* des 4-Schritte-Modells per E-Mail oder über eine Dokumentenablage einer Lernplattform durchzuführen, hat sich nicht bewährt, weil dadurch wichtige Effekte der Abgabegespräche (z.B. die soziale Komponente) ausbleiben und dabei die Versuchung für die Lernenden gross ist, die Lösungen bei Mitstudierenden zu kopieren.

Verwalten der Leistungsnachweise grosser Klassen in einer Datenbank

Für die Verwaltung der Leistungsnachweise von individualisierten Kursen mit einer grossen Anzahl an Studierenden macht der Einsatz einer Online-Datenbank Sinn. Sie ermöglicht die elektronische Verwaltung der Leistungsnachweise durch mehrere Assistentende über das Internet (Abb. 9.5). Die Daten aller für den Kurs immatrikulierten Personen werden dabei direkt aus der Lehrbetriebsapplikation der ETH importiert [Lehrbetriebsapplikation ETH Zürich, <http://edoz.ethz.ch>]. Die Assistentenden geben die Leistungsnachweise über eine Webseite in die Datenbank ein. Zusätzlich können nach Kursende die Resultate der Leistungskontrollen hinzugefügt werden. Die Studierenden haben die Möglichkeit, über die Webseite Einsicht in ihre bereits erledigten Leistungsnachweise zu erhalten. Nach Abschluss des Kurses liefert die Datenbank die nötige Information zur Übermittlung von Kreditpunkten und Noten über die Lehrbetriebsapplikation.

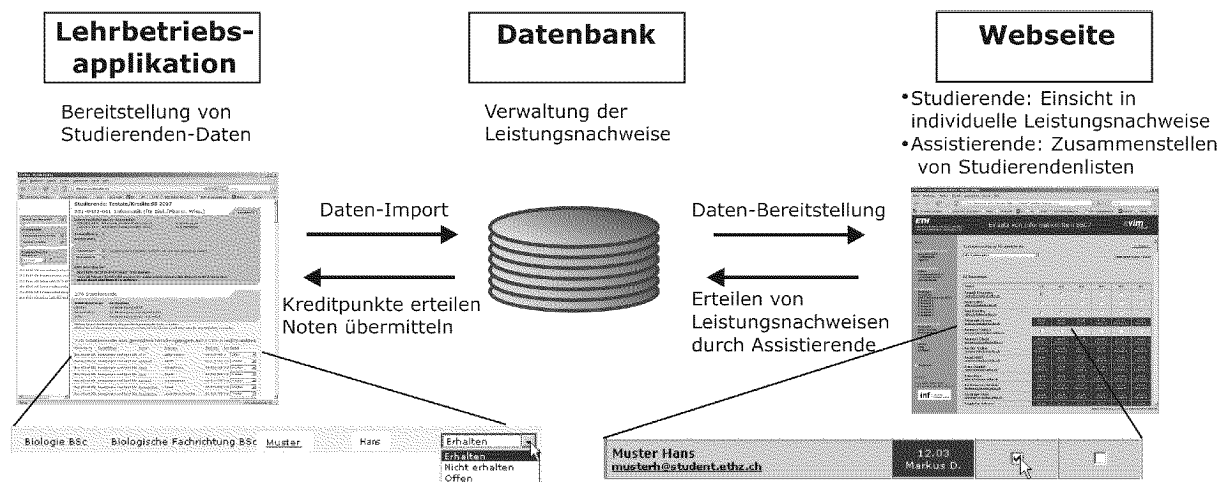


Abb. 9.5: Elektronisches Verwalten von Leistungsnachweisen grosser Klassen mit einer Datenbank. Die Daten der Studierenden werden direkt aus der Lehrbetriebsapplikation der ETH Zürich [eDoz, <http://edoz.ethz.ch>] in eine Datenbank importiert. In der Datenbank werden alle Leistungsnachweise verwaltet. Ausgewählte Daten lassen sich auf der Webseite anzeigen. Die Assistentenden speichern die Leistungsnachweise ebenfalls via Webseite in die Datenbank. Nach Abschluss des Kurses können Kreditpunkte und Noten der Studierenden wiederum über die Lehrbetriebsapplikation vergeben werden [für Details siehe „Installation und Konfiguration des ET-Portals“, Dahinden 2006].

2. Leistungskontrollen (Assessmentphase)

Während der Assessmentphase werden zwei Leistungskontrollen durchgeführt (siehe Abb. 9.2):

- Semesterendetest (unbenotet) (D3)
- Prüfung (benotet) (D4).

Für den unbenoteten Semesterendetest haben wir ein elektronisches Assessment-system zum effizienten Prüfen grosser Studierendengruppen entwickelt [Bosia 2005]. Weil der Semesterendetest im Gegensatz zur benoteten Prüfung ein Teil des Lernprozesses ist, sind die beim elektronischen Prüfen möglichen sofortigen Feedbacks sinnvoll. Sie zeigen den Lernenden ihren Wissensstand in den unterschiedlichen Lerninhalten auf. Die benotete Prüfung wird zurzeit aus Gründen der Rekursfähigkeit noch immer auf Papier durchgeführt.

Die Durchführung der Leistungskontrolle ist Inhalt eines weiterführenden Projekts mit dem Titel „K3+ basierte, elektronisch gestützte Leistungskontrolle“. Es hat zum Ziel, die während den Kursen angeeigneten Kompetenzen zu prüfen (siehe Kapitel 11.3.1).

9.4.6 Qualitätskontrolle eines Kurses durchführen (Aufgabe E)

Neben der Kontrolle und Verbesserung der Qualität der Lernmaterialien (siehe Kapitel 8.6 „Qualitäts-Feedback“) wird auch die Qualität der durchgeführten Kurse evaluiert. Der Grund für diese zusätzliche Evaluation ist, dass jeder Kurs schlussendlich wieder andere Bedingungen (Zielgruppe, Wochentage, Institution, Positionierung im Curriculum etc.) hat. Damit ein Kurs besser an die Bedürfnisse und Bedingungen einer spezifischen Zielgruppe angepasst werden kann, wird für jeden Kurs eine Vor- (E1) und eine Nachumfrage (E2) durchgeführt und ausgewertet (Beispiel eines solchen Fragebogens siehe Anhang C).

9.4.7 Präsenzveranstaltungen organisieren und halten (Aufgaben F1-6)

Es hat sich gezeigt, dass es trotz des Einsatzes einer individualisierenden, kompetenzorientierten Lernumgebung verschiedene Gründe gibt, an einer Präsenzveranstaltung (z.B. in Form einer Vorlesung im Frontalunterricht) festzuhalten. Der zeitliche Anteil der Präsenzveranstaltungen am Gesamtunterricht sowie deren Inhalt und Rolle haben sich durch die Entwicklung der computergestützten Lernumgebung allerdings stark verändert.

1. Reduktion des Präsenzanteils am Gesamtunterricht

In den von uns organisierten Kursen führen wir alle 14 Tage eine Präsenzveranstaltung in Form einer Vorlesung durch (siehe Abb. 9.2, F1-F6). Dies entspricht einer Reduktion des Präsenzanteils am Gesamtunterricht gegenüber früher um 50%. Der Grund dafür ist die Tatsache, dass die Vermittlung der Grundlagen durch die problembasierte Instruktion (Phase *TRY*) am Computer und nicht mehr während einer Präsenzveranstaltung stattfindet und ein weiterer Teil des Präsenzunterrichts als Abgabegespräche (Phase *EXPLAIN*) durchgeführt wird. Der Präsenzanteil wurde also zu Gunsten des begleiteten Selbststudiums gesenkt.

2. Neue Rolle von Präsenzveranstaltungen beim Unterricht im 4-Schritte-Modell

Durch den Unterricht im 4-Schritte-Modell erhalten die Lernenden die Möglichkeit, Grundlagen zusammen mit einem realen Kontext problemorientiert und selbstgesteuert einzuführen. Das Resultat ist aktives, einsetzbares Wissen, welches mit persönlichen Erfahrungen verknüpft ist. Diese Kompetenzorientierung ist durch keinen Frontalunterricht zu erreichen. Es stellt sich die Frage, welche Rolle dem Präsenzunterricht als Ergänzung zum Unterricht im 4-Schritte-Modell zukommt.

Einführen, zusammenfassen und weiterentwickeln der Themen. Die Hauptaufgabe einer Präsenzveranstaltung im Zusammenhang mit der Einführung von Grundlagen im 4-Schritte-Modell ist die Integration der gelernten Grundlagen in ein Gesamtbild. Die Lernenden sollen durch zusätzliche Informationen und Beispiele ihr angeeignetes Wissen zusammenfassen und weiterentwickeln können. Dazu gehört beispielsweise das Aufzeigen von weiteren Situationen, in denen das angeeignete Wissen eingesetzt werden kann. Mit der Anwendung des angeeigneten Wissens kommen die Lernenden bei bestimmten Problemen an ihre Grenzen. Eine weitere Rolle der Vorlesung ist es, diese Grenze aufzuzeigen und ergänzende Konzepte vorzustellen.

Möglichkeit eines Face-to-Face Kontaktes mit den für eine Unterrichtsveranstaltung verantwortlichen Personen. Es wird von vielen Studierenden geschätzt, wenn sie mit den Personen, die hinter der Entwicklung einer computergestützten Lernumgebung stehen und für eine Unterrichtsveranstaltung Verantwortung übernehmen, von Zeit zu Zeit in Kontakt treten können. Kommt dazu, dass Studierende einer Grundlagen-Lehrveranstaltung Lerninhalte häufig mit einem starken Bezug zu einer (Lehr-)Person lernen.

Eine Präsenzveranstaltung kann einem Fach jenes Gewicht verleihen, das dem tatsächlich zu leistenden Aufwand entspricht. Die Studierenden teilen ihre verfügbare Zeit nach den Anforderungen, die durch verschiedene Lehrveranstaltungen eines Semesters an sie gestellt werden. Lehrveranstaltungen ohne Präsenzanteil laufen Gefahr, als zusätzlicher, in der Freizeit zu leistender Aufwand eingestuft zu werden. Eine regelmässig stattfindende Präsenzveranstaltung kann helfen, einem Fach in einer Ausbildung jenes Gewicht zu verleihen, das dem tatsächlich zu leistenden Aufwand entspricht.

3. Problematik einer Konkurrenz zwischen Unterricht im 4-Schritte-Modell und dem Frontalunterricht

Die Vermittlung von Grundlagen mittels 4-Schritte-Modell hat einen Nachteil: sie ist relativ zeitintensiv. Demgegenüber könnten im Frontalunterricht die Grundlagen in einem Bruchteil der Zeit behandelt werden. Der wesentliche Unterschied liegt darin, dass auf diese Weise angeeignetes Wissen träge bleibt und nicht angewendet werden kann (folglich Kapitel 2.1 „Problem von trägem Wissen im Unterricht“). Aus diesem

Grund ist es für die Durchführung von Präsenzveranstaltungen wichtig, dass sie nicht als Konkurrenz sondern als Ergänzung zum kompetenzorientierten System stehen. Bei einer direkten Konkurrenz der beiden Systeme besteht die Gefahr, dass der Frontalunterricht als die effizientere Form der Wissensaneignung wahrgenommen wird. Die Lernenden sollten aber zur Einsicht gebracht werden, dass sich der zu leistende zusätzliche Aufwand lohnt. Der Mehrwert des zeitaufwändigeren Weges über die Kompetenzorientierung kommt erst dadurch zum Tragen, wenn die Anforderung der Leistungskontrolle nicht bloss die Wiedergabe von Theoriefakten verlangt, sondern Wissen prüft, das praktische Erfahrung in der Auseinandersetzung mit dem Inhalt erfordert (siehe dazu Kapitel 2.5.2 „Kognitive Prozessdimensionen einer Kompetenz“ sowie die durchgeführte Evaluation im nächsten Kapitel).

Zusammenfassend halten wir fest, dass es sich lohnt, auch im stark individualisierten Unterricht des 4-Schritte-Modells an einer Präsenzveranstaltung festzuhalten. Weil die Vermittlung der Grundlagen individualisiert am Computer stattfinden, erhalten die Dozierenden für die Durchführung des Präsenzunterrichts neue Freiheiten, die es zu nutzen gilt, damit eine Lehrveranstaltung als Ganzes interessanter und auch lehrreicher wird.

9.5. Fazit

Die Hauptarbeit beim Aufbau eines individualisierten und kompetenzorientierten Lernsystems steckt in der Aufbereitung der Lerninhalte. Für den produktiven Einsatz der Lernumgebung im Unterricht mit grossen Klassen müssen zusätzlich effiziente Ablaufstrukturen bereit gestellt werden, die ebenfalls mit dem Computer unterstützt werden können. Es hat sich bei der Entwicklung des Lernsystems bewährt, strikte zwischen Lernmaterialien und Ablaufstrukturen zu trennen. Die dadurch ermöglichte Flexibilität bildet die Grundlage, dass sich die Lernmaterialien zu Modulen organisieren lassen, die in einem Pool gesammelt, unterhalten und zu neuen Kursen für weitere Zielgruppen und Institutionen zusammengestellt werden können.

Wir haben bei der Entwicklung unseres Lernsystems bewusst auf den Einsatz einer Lernplattform oder eines Learning-Management-Systems verzichtet, um unabhängig zu bleiben und den Lernenden zusätzlichen Installations- bzw. Zugangsaufwand zu ersparen. Die entwickelten Lernmodule können, falls dies von einer Institution gewünscht wird, in eine Lernplattform eingebunden werden, um zum Beispiel gewisse Ablaufstrukturen (z.B. für die Verwaltung von Leistungsnachweisen) zu nutzen. Dies ist möglich, weil für die Entwicklung der Lernmodule eine einfache, weit verbreitete Technologie (HTML und *JavaScript*) verwendet wurde.

Unterrichten mit Hilfe einer computergestützten Lernumgebung kommt nicht ohne persönliche Betreuung aus. Der Präsenzunterricht steht bei dieser Art von Unterricht aber nicht mehr im Zentrum, sondern stellt eine Ergänzung zu den individuell an-

geeigneten Kompetenzen dar. Die Rollen der am Unterricht beteiligten Personen hat sich ebenfalls verändert: Assistierende werden zu Coaches und Dozierende stehen vor der Aufgabe, in der von ihnen durchgeführten Präsenzveranstaltung das durch die Lernumgebung generierte Wissen aufzunehmen, in einen Kontext zu stellen und die Breite dieses Kontexts aufzuzeigen.

Alle Teile einer solchen *Blended Learning* Umgebung sind Teil des selben Lernsystems und sollten auf die veränderten Bedingungen des individualisierten Wissenserwerbs abgestimmt sein. Dabei stellen sich für die Weiterentwicklung der Lernumgebung folgende Fragen:

- Wie lassen sich Assistierende besser ins Lernsystem integrieren?
- Wie kann Präsenzunterricht besser auf ein Lernsystem mit individuellem Wissenserwerb angepasst werden?

Kapitel 10

Evaluation der Lernumgebungen

10.1. Einführung

Im letzten Kapitel wurde gezeigt, wie eine kompetenzorientierte Lernumgebung durch die Umsetzung der beiden didaktischen Modelle *4-Schritte-Modell* und *Anwendungsebenen-Modell* entwickelt und im Unterricht eingesetzt wird. Dieses Kapitel behandelt die Evaluation dieser Lernumgebungen. Mit der Entwicklung computergestützter Lernumgebungen wurde das Ziel verfolgt, die Qualität, Flexibilität und Effizienz des Unterrichts nachhaltig zu verbessern. Durch den Ansatz des System-Engineerings wurden bereits existierende traditionelle Lernumgebungen mit Vorlesung, Übungen und Unterrichtsskript durch computergestützte Lernumgebungen im *Blended Learning* abgelöst, was mit einem mehrere Jahre dauernden Entwicklungs- und Implementierungsaufwand verbunden war (folglich Kapitel 4.3 „System-Engineering“, Abb. 4.4). Die durch diese Implementierung bewirkten Veränderungen in der Lernumgebung sollen im Zentrum der durchgeführten und in diesem Kapitel diskutierten Evaluationen stehen.

Die durch die Entwicklungsprozesse entstandenen Lernumgebungen sind hoch optimierte komplexe Systeme, deren Einzelfaktoren in vielfältiger Wechselwirkung zueinander stehen und nicht isoliert betrachtet werden können. Für die Evaluation eines solchen Systems bedeutet dies, dass neben der Beurteilung einzelner Systemelemente (wie z.B. die Usability eingesetzter Medien) auch Evaluationen durchgeführt werden, deren Fokus auf das *gesamte* Lernsystem gerichtet ist.

Während der Entwicklung und dem Einsatz der Lernumgebung im Unterricht sind über mehrere Jahre Daten gesammelt worden (folglich Kapitel 8.6 „Qualitäts-Feedback“). Die Analyse dieser Daten kann dazu genutzt werden, die Lernumgebungen zu evaluieren und deren Entwicklungsgeschichte zu illustrieren. Es entspricht der Eigenschaft eines durch den System-Engineering-Approach entstandenen Systems, dass sich viele allgemeine Strukturen nur im Zusammenhang mit ihrer Entstehungsgeschichte interpretieren und verallgemeinern lassen (folglich Kap. 4.3.1 „System-Engineering von Lernumgebungen“).

Weil die Einführung von Kontrollgruppen über die Zeitspanne ganzer Kurse nicht vertretbar und schwierig durchzuführen wären, können die bei der Evaluation ganzer Lernumgebungen erhobenen Veränderungen nicht auf einzelne Faktoren oder Inter-

ventionen zurückgeführt werden. Der Blick in ausgewählte Daten aus dem Unterricht kann uns aber helfen, während entscheidender Phasen des Entwicklungsprozesses Effekte zu visualisieren und zur Diskussion zu bringen.

10.2. In dieser Arbeit durchgeführte Evaluationen

10.2.1 Übersicht

Über einen Zeitraum von mehreren Jahren wurden zwei computergestützte Lernumgebungen (ICT-Kurs, Programmierkurs) in fünf durchgeführten Kursen mit jährlich insgesamt über 1100 Studierenden evaluiert (Stand 2006/07, Übersicht über alle Kurse siehe Kapitel 1, Tab. 1.2). Tab. 10.1 fasst die Art der Evaluationen und Datenerhebungen, sowie die betroffenen Lernumgebungen und Kurse zusammen.

Art der Evaluation	Art der Datenerhebung	Betroffene Lernumgebung und Kurse	Resultate in Kapitel/Abschnitt dieser Arbeit
Pre-Post-Analyse (ab 2001)	Statuserhebung	ICT-Kurs (IBPW/EvIM/I1) Programmierkurs (AP/I2)	Kapitel 3, 7 und 8
Outputanalyse (ab 2002)	Leistungserhebung	ICT-Kurs (IBPW)	Abschnitt 10.3
Prozessanalyse (ab 2005)	Prozesserhebung	ICT-Kurs (IBPW/EvIM) Programmierkurs (I2/AP)	Abschnitt 10.4

Tab. 10.1: In dieser Arbeit durchgeführte Evaluationen. Abkürzungen der Kurse: EvIM: Einsatz von Informatikmitteln, IBPW: Informatik für Biologie und Pharmazeutische Wissenschaften, AP: Anwendungsnahes Programmieren, I1: Informatik 1 für Bauingenieure, I2: Informatik 2 für Bauingenieure. Übersicht über alle Kurse siehe Tab. 1.2.

10.2.2 Datenerhebungen im Unterricht

Grundlage der in diesem Kapitel besprochenen Evaluationen bildet eine Reihe von Daten, die während der Entwicklung und dem Einsatz der Lernumgebung zur Qualitätsverbesserung der Lernmaterialien erhoben werden. Wichtiger Teil der Entwicklung ist der wiederholte Einsatz und die Optimierung der Lernmaterialien im Unterricht (siehe auch Kapitel 8.6 „Qualitäts-Feedback“). Zusätzlich wird zur Qualitätskontrolle durchgeführter Kurse bei jedem Einsatz einer Lernumgebung standardmässig eine Reihe von Daten erhoben (siehe Kapitel 9.4.6 „Qualitätskontrolle eines Kurses durchführen“). In Abb. 10.1 sind die verschiedenen Zeitpunkte der Datenerhebungen während eines Kurses dargestellt.

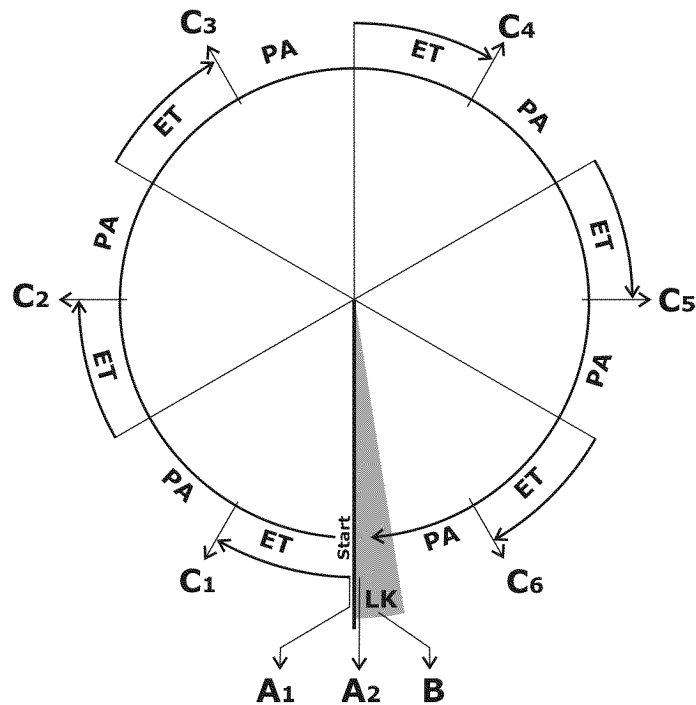


Abb. 10.1: Drei verschiedene Datenerhebungen während der Durchführung eines Kurses während eines Semesters. Abkürzungen: A: Staturerhebung, B: Leistungserhebung, C: Prozesserhebung, ET: *E.Tutorial*[®] (problembasierte Instruktion), LK: Leistungskontrolle, PA: Selbständige Problemaufgabe (problem-basierte Anwendung).

1. Staturerhebung (Abb. 10.1, A)

Das Erheben von Daten vor einer Veranstaltung hat zum Ziel, ein Ausgangsprofil der Teilnehmenden festzuhalten und ist Teil des Entwicklungsprozesses einer Lernumgebung (siehe Kapitel 8.3, Prozesse A2 und A3). Resultate der *Vorumfrage* sind in dieser Arbeit in folgenden Kapiteln zu finden:

- Kapitel 3.4: Vorumfrage zu den ICT-Kompetenzen
- Kapitel 8.3: Vorumfrage zu Vorpraxis, Gewohnheiten der Lernenden im Umgang mit dem Computer, subjektiver Einschätzung zu Kompetenz, Motivation und Nützlichkeit des zu erwerbenden Wissens durch die Lernenden.

Mit der Erhebung von Daten nach der Veranstaltung wird das Ziel verfolgt, sich ein Zustandsbild der Teilnehmenden zu verschaffen, nachdem sie den Kurs absolviert haben. Die Resultate der *Nachumfrage* sind in dieser Arbeit in folgenden Kapiteln zu finden:

- Kapitel 7.2.3: Nachumfragen zu den eingesetzten Medien im 4-Schritte-Modell
- Kapitel 7.4: Nachumfragen zum didaktischen Modell.

2. Leistungserhebung (Abb. 10.1, B)

Bei einer Leistungserhebung werden Daten durch eine Leistungskontrolle erhoben. Aus der Leistung einer Gruppe von Studierenden in der Leistungskontrolle kann auf zugrunde liegende Kompetenzen geschlossen werden. Dabei ist entscheidend, mit welcher Methode auf welchem Kompetenz-Level geprüft wurde [Fisseni 2004].

In Abschnitt 10.3 dieses Kapitels werden Leistungsdaten einer anwendungsorientierten Leistungskontrolle vorgestellt, die als Mass für anwendbare Wissensstrukturen auf Seiten der Lernenden dienen sollen.

3. Prozesserhebung (Abb. 10.1, C)

Im Gegensatz zu einem Leistungstest können mit Prozesserhebungen Veränderungen in einem laufenden Prozess festgestellt werden [Fisseni 2004]. Prozessanalysen sind vor allem aus der psychologischen Diagnostik bekannt. Veränderungen werden gemessen, indem zu unterschiedlichen Zeitpunkten eines Prozesses ein Zustandsbild erhoben und die Zustände in der Sequenz über die Zeit verglichen werden (Abb. 10.2). Diese Erfassung gibt Auskunft darüber, ob sich zwischen Zustand X und Zustand Y etwas verändert hat. Die Veränderungsmessung muss zu wenigstens zwei Zeitpunkten erfasst werden, sinnvoll sind allerdings drei Erhebungen und mehr. Die Prozessmessung erfasst, ob sich zwischen den Messzeitpunkten auf Grund einer Intervention etwas verändert hat. Die Schwierigkeit besteht darin, ein Mass zu finden, das über die Zeit vergleichbar ist.

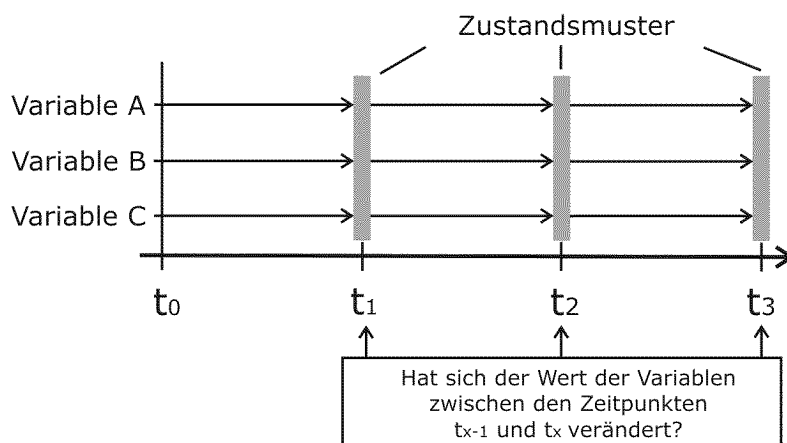


Abb. 10.2: Ablauf einer Prozessanalyse. In regelmässigen Abständen eines Prozesses werden Zustandsmuster erfasst, indem die Probanden für verschiedene Variablen befragt werden, ob sie gegenüber der letzten Erfassung Veränderungen festgestellt haben.

Sollen in einem System psychologische Variablen wie z.B. „Motivation“ oder „Zufriedenheit“ erfasst werden, dann ergibt sich das Problem, dass diese erst so definiert werden, dass Messungen möglich sind. Durch eine Prozessanalyse kann diese Art von Variablen als eingeschätzte Veränderung statt als Absolutwerte erfasst werden. Die Probanden

werden dabei selber als „Messinstrument“ eingesetzt, indem sie berichten, ob sie in einem Zeitabschnitt Veränderungen festgestellt haben.

In Abschnitt 10.4 werden Prozessanalysen eingesetzt, um die Qualität der Implementierung von Problemaufgaben in problembasierten Kursen zu prüfen.

10.3. Outputanalyse: Anwendungsorientierte Leistungskontrolle als Mass für das Vorhandensein von anwendbaren Wissensstrukturen

10.3.1 Übersicht

Ziel dieser Evaluation:		
Mit dieser Evaluation wurde untersucht, ob sich die Lernenden im Blended Learning-Unterricht aktives Wissen aneignen konnten. Hierfür wurden die Resultate der Leistungskontrollen (Prüfung) mehrerer Jahre miteinander verglichen.		
Evaluationskriterien	Messgrößen	Evaluationsinstrumente
Studierende beantworten anwendungsorientierte Prüfungsfragen besser, wenn Unterricht und Prüfung aufeinander abgestimmt sind.	Anteil korrekt beantworteter Anwendungsfragen an der Prüfung	Anwendungsorientierte Multiple-Choice (MC)-Fragen

Die Resultate dieser Evaluation wurden 2005 an der *World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA)* präsentiert [Faessler 2005].

10.3.2 Einführung

Eine zentrale Motivation für die Entwicklung und den Einsatz computergestützter Lernmaterialien ist die Durchführung eines Grundlagenunterrichts von kontrollierbarer Qualität. Die Aneignung von nachhaltigem aktiven Wissen soll durch die Vermittlung von definierten Kompetenzen im PBL erreicht werden (siehe Kapitel 2.4 „Erhöhen der Unterrichtsqualität durch die Kompetenzorientierung“ und Kapitel 5 „PBL“). Für den Informatikunterricht ist es nahe liegend, den Erwerb von ausgewählten Kompetenzen direkt am Computer durchzuführen, weil dies die Aneignung von zugehörigen Fertigkeiten und Fähigkeiten ermöglicht (folglich Kapitel 3 „ICT-Kompetenzen“). Dies führte zur Entwicklung von E-Learning-Materialien, mit deren Hilfe die Lernenden Informatik-Kompetenzen problembasiert direkt am Computer aneignen können. Die entwickelten Lernmaterialien kommen im Unterricht im *Blended Learning*, das heisst in Kombination mit traditionellen Unterrichtsstrukturen, zum Einsatz (folglich Kapitel 7 „Didaktische Modelle“ und Kapitel 9 „Einsatz der computergestützten Lernumgebung im Unterricht“).

Während verschiedener Entwicklungsphasen dieses Projekts hat sich die Frage gestellt, ob der Einsatz der entwickelten Lernmaterialien im Unterricht das Ziel erfüllt, dass die Lernenden ihre angeeigneten Wissensstrukturen zum Lösen von Problemen einsetzen können. Um dies zu klären, musste ein Mass definiert werden, welches Auskunft darüber gibt, inwiefern dieses Ziel erreicht worden ist.

10.3.3 Leistungskontrolle als Evaluationsinstrument

Leistungskontrollen (*Assessments*) können als Output des Unterrichts wichtige Daten über die Qualität des Unterrichts liefern. Aus der Leistung der Studierenden in der Leistungskontrolle kann auf zugrunde liegende Kompetenzen geschlossen werden [Fisseni 2004]. Dabei ist entscheidend, dass Unterricht und Leistungskontrolle aufeinander abgestimmt sind. Als Mass für die Vergleichbarkeit von Unterricht und Leistungskontrolle eignet sich Blooms Lernzieltaxonomie [Anderson 2001] (siehe Kapitel 2.5.2 „Kognitive Prozess-Dimensionen einer Kompetenz“). Das Kompetenzlevel K3 umfasst die Anwendung und K4 die Analyse von Konzepten in einer gegebenen Situation (siehe Tab. 2.5) und entspricht der Zielsetzung, die durch den kompetenzorientierten Grundlagenunterricht erreicht werden soll.

Die Umstrukturierung des Unterrichts Richtung Anwendungsorientierung hatte auch eine Umstrukturierung der Leistungskontrolle zur Folge. In den Leistungskontrollen der Jahre 2002 bis 2004 wurden vermehrt Prüfungsfragen gestellt, welche das Level K3 und K4 abfragen (Abb. 10.3).

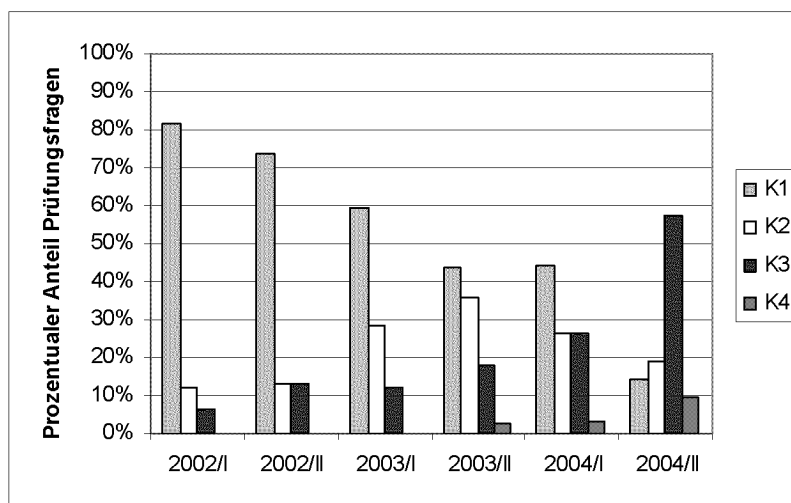


Abb. 10.3: Prozentualer Anteil an Prüfungsfragen eines bestimmten kognitiven Levels der Jahre 2002, 2003 und 2004. Jedes Jahr fanden zwei Prüfungen (I und II) statt. K1: erinnern, K2: verstehen, K3: anwenden, K4: analysieren. N: 33 bis 42 Fragen [Faessler 2005].

K3-basierte Prüfungsfragen

Eine effiziente Form, bei einer Leistungskontrolle die Kompetenzlevel K3 und K4 abzufragen, sind problembasierte Multiple-Choice Fragen [Faessler & Wyss 2005] (Abb. 10.4). Bei dieser Art von Prüfungsfragen erhalten die Studierenden eine Problemstellung (z.B. ein Programmcode), die im Unterricht behandelte Konzepte enthält (z.B. Programmwiederholungen, Bedingungsprüfungen und Objekte). Weiter enthält die Aufgabe fünf Behauptungen, die jeweils als richtig oder falsch eingestuft werden müssen. Zur richtigen Beantwortung der Frage müssen die Studierenden die Behauptungen in der gegebenen Problemsituation so weit nachvollziehen, dass sie sie auf ihre Richtigkeit überprüfen können. Dabei müssen die Studierenden auf das Wissen und die Erfahrungen, die sie während dem Unterricht erworben haben, zurückgreifen.

Gegeben sei folgende Visual Basic-Prozedur (A) und Excel-Tabelle (B):

```

Sub colour()
Range("B12").Formula = "=SUM(B2:B11)"/10
Range("B13").Value = Range("B12").Value / 10
Dim cell As Object
For Each cell In Range("B2:B11")
If cell.Value > Range("B13").Value Then
cell.Interior.ColorIndex = 3
ElseIf cell.Value <= Range("B13").Value Then
cell.Interior.ColorIndex = 24
End If
Next
End Sub

Private Sub CommandButton1_Click()
colour
End Sub

```

	A	B
1	food	fat [g]
2	Coke	0.00
3	Cheese, Emmental	32.00
4	Chocolate, milk with nuts	20.47
5	Egg, raw	10.50
6	Cornflakes	1.00
7	Beer, 4.8%	0.00
8	Fish, salmon	13.40
9	Wine, red, 12%	0.00
10	Pineapple, fresh	0.20
11	Apple, fresh	0.30
12		
13		
14		
15		
16		

Die Visual Basic-Prozedur wird auf die Excel-Tabelle angewendet.

Welche der folgenden Aussagen treffen nach dem Anklicken der Schaltfläche „click“ zu resp. *nicht* zu?

trifft zu	trifft nicht zu	
A <input type="radio"/>	A <input type="radio"/>	In Zelle B12 wird die Formel „=B2+B11“ stehen.
B <input type="radio"/>	B <input type="radio"/>	In Zelle B13 wird der Mittelwert der Zellen B2 bis B11 stehen.
C <input type="radio"/>	C <input type="radio"/>	Zellen mit dem gleichen Wert wie B13 werden mit der Farbe <code>colorIndex = 3</code> (rot) eingefärbt.
D <input type="radio"/>	D <input type="radio"/>	Zelle B13 wird eingefärbt.
E <input type="radio"/>	E <input type="radio"/>	Verändert man im obigen Code [Zeile 5] den Range („B2:B11“) neu zu Range („B2:B6“) , so würde der berechnete Mittelwert höher ausfallen.

Abb. 10.4: Beispiel einer problembasierten MC-Prüfungsfrage. Die Problemsituation enthält mehrere im Unterricht behandelte Konzepte. Zur Problemsituation werden fünf Behauptungen aufgestellt, die von den Studierenden als richtig oder falsch klassiert werden müssen [weitere Details siehe Faessler & Wyss 2005].

Mass für das Vorhandensein von transferfähigem Wissen

Die Resultate von problembasierten Prüfungsfragen sollen zeigen, inwiefern die Lernenden fähig sind, die gelernten Konzepte als aktives Wissen zur Lösung von Problemen zu mobilisieren. Sie können als Mass für das Vorhandensein von transferfähigem Wissen interpretiert werden, das in einer Lernumgebung angeeignet worden ist.

10.3.4 Entwicklungsverlauf von Unterricht und Leistungskontrolle

Tab. 10.2 und Abb. 10.5 zeigen die Entwicklung und Integration von Lernmaterialien in einen Kurs von 2001 bis 2005 (ETH-Kurs 252-0432-00L: „Informatik für Biologie und Pharmazeutische Wissenschaften“, Tab. 1.2). Der Kurs wurde bis 2001 in traditioneller Form ohne Unterstützung durch E-Learning gehalten. Die elektronischen Lernmaterialien (*E.Tutorials*®) wurden ab 2001 umgesetzt und waren ab 2002 im Unterricht im Einsatz. Durchgeführte Evaluationen im Jahre 2002 zeigten eine hohe Akzeptanz und Beliebtheit der Lernmaterialien bei den Studierenden [unpublizierte Evaluationsresultate 2002, <http://et.ethz.ch>].

Jahr	Anzahl Studierende	Unterrichtsform und eingesetzte Medien	Kompetenzlevel Leistungskontrolle
2001	281 Studierende	<i>Traditionell</i> : Vorlesung, Skript, Übungen	K1/2, erste K3/4
2002	276 Studierende	<i>Blended Learning</i> : Vorlesung, Skript Übungen mit <i>E.Tutorials</i> ®	K1/2, wenig K3/4*
2003	300 Studierende	<i>Blended Learning</i> : Vorlesung, Skript Übungen mit <i>E.Tutorials</i> ®	K1/2, wenig K3/4*
2004	321 Studierende	<i>Blended Learning</i> im 4-Schritte-Modell mit Vorlesung	K3/4, wenig K1/2*
2005	246 Studierende	<i>Blended Learning</i> im 4-Schritte-Modell mit Vorlesung	K3/4, keine K1/2

Tab. 10.2: Umstrukturierung eines traditionell gehaltenen Kurses zu einem problembasierten Blended Learning im 4-Schritte-Modell 2001 bis 2005. Parallel zur Entwicklung wurde das Kompetenzniveau der Leistungskontrolle angehoben. ETH-Kurs 252-0432-00L „Informatik für Biologie und Pharmazeutische Wissenschaften“ (siehe Tab. 1.2). *siehe Abb. 10.3.

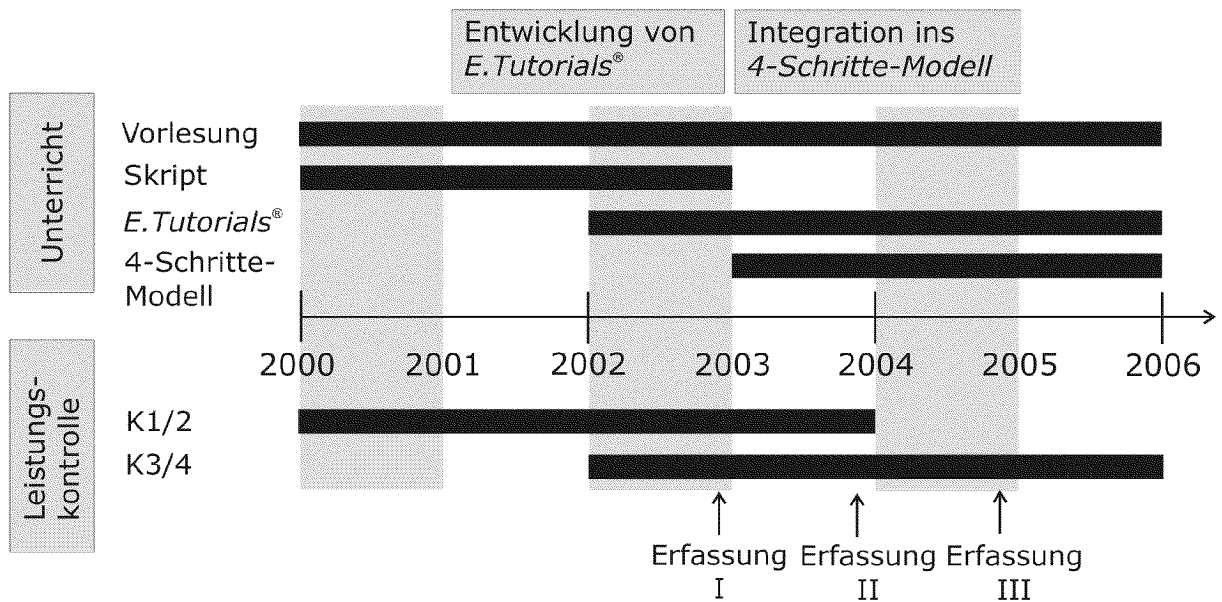


Abb. 10.5: Entwicklung von Unterricht und Leistungskontrolle eines Kurses von einem traditionellen zu einem problembasierten Blended Learning-Kurs nach dem 4-Schritte-Modell. Der Kurs wurde jährlich durchgeführt. Die *E.Tutorials®* kamen erstmals 2002 zum Einsatz, ab 2003 wurden sie im 4-Schritte-Modell eingesetzt. Die Erfassung der Leistungsdaten erfolgte nach den Kursen 2002 (I), 2003 (II) und 2004 (III). Details siehe Text.

Parallel zum Einsatz der E-Learning-Materialien wurde auch der kognitive Level der Leistungskontrolle angepasst. Die Studierenden mussten an der Prüfung vermehrt anwendungsorientierte Fragen beantworten (siehe Abb. 10.3). An den Prüfungen hat sich gezeigt, dass viele Studierende trotz der Umstellung zu Blended Learning und dem Einsatz von E-Learning-Materialien Mühe hatten, ihr angeeignetes Wissen zur Lösung von neuen Problemen anzuwenden [unpublizierte Resultate 2001 und 2002].

Was war das Problem? War die Umstrukturierung des Kurses zu einem Blended Learning-Kurs ungenügend? Trotz des neu geschaffenen Angebots eines anwendungsorientierten Zugangs zu den Lerninhalten, schienen sich die Lernenden nicht genügend aktives Wissen angeeignet zu haben, um damit an der Prüfung neue Probleme lösen zu können. Das theoriebasierte System aus Vorlesung und Skript stand vermutlich in Konkurrenz zum neuen anwendungs- und problembasierten System. Zudem stellten sich die Studierenden aus ihrer Schulerfahrung darauf ein, dass sie an der Prüfung den Inhalt aus Skript und Vorlesung in Form von K1/K2-Prüfungsfragen wiedergeben müssen und die Anwendung des Wissens an der Prüfung eher eine untergeordnete Rolle spielen wird.

Wir entschieden uns dann, das entwickelte E-Learning-Material nach einem einheitlichen didaktischen Modell umzusetzen, das die Verknüpfung verschiedener Wissensdimensionen und den Aufbau von individuellen Kompetenzen mit aktivem Wissen ermöglichen soll (siehe Kapitel 7 „Didaktische Modelle“ Abb. 7.2). Das eingeführte 4-Schritte-Modell stellt die problembasierte Aneignung und Anwendung von Wissen ins Zentrum des Unterrichts und bereitet die Studierenden auf eine anwen-

dungs- und problemorientierte Leistungskontrolle vor. Durch das 4-Schritte-Modell wurden zusätzlich zu den E-Learning-Materialien folgende Elemente entwickelt:

- Integration des Skripts in das 4-Schritte-Modell (Phase *SEE*)
- Ausbau der Phasen *DO* und *EXPLAIN* des 4-Schritte-Modells

Die Phase *SEE* bringt die Konzepte ins 4-Schritte-Modell und ermöglicht deren Verknüpfung mit Fertigkeiten während der Phase *TRY* (siehe Abb. 7.3). Die Phasen *DO* und *EXPLAIN* führen zusätzlich zur Aneignung von Fähigkeiten und metakognitivem Wissen, was die Problemlösefähigkeit verbessern soll. Im Jahre 2004 wurde der Kurs erstmals als Blended Learning-Kurs im problembasierten 4-Schritte-Modell durchgeführt und auch grösstenteils anwendungsorientiert geprüft (siehe Tab. 10.2, Abb. 10.3 und Abb. 10.5).

Zusammenfassend soll durch eine bessere Integration der E-Learning-Angebote in den Unterricht durch die Einführung eines einheitlichen didaktischen Modells die Problemlösefähigkeit der Lernenden verbessert werden und an der Prüfung messbar zum Ausdruck kommen. Gleichzeitig soll der kognitive Level der Leistungskontrolle den veränderten Bedingungen im Unterricht angepasst und ausschliesslich anwendungsorientiert durchgeführt werden.

10.3.5 Resultate der Outputanalyse

Abb. 10.6 zeigt die Verteilung von korrekt beantworteten K3/K4-Fragen in den Prüfungen 2002, 2003 und 2004. Studierende im Jahre 2004 waren signifikant besser beim Lösen der K3/4-Fragen als die Studierenden im Jahre 2002 ($p < 0.01$ Mann-Whitney-Test).

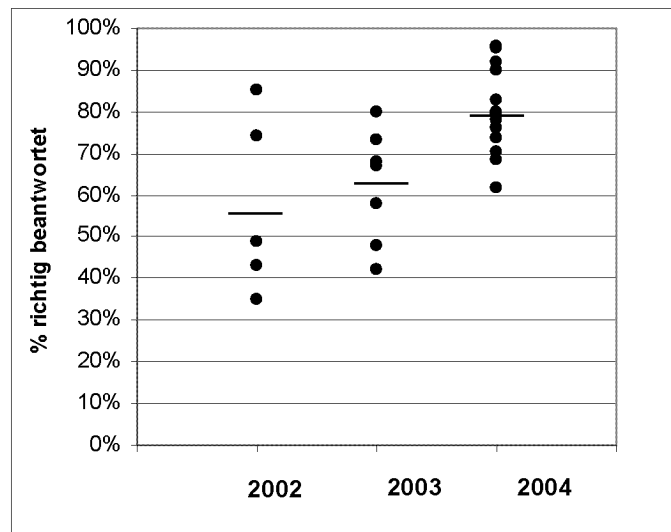


Abb. 10.6: Verteilung von korrekt beantworteten K3/4-Fragen in den Prüfungen von 2002 (Abb. 10.5, I), 2003 (II) und 2004 (III). Jeder Punkt repräsentiert eine anwendungsorientierte Prüfungsfrage. Im Jahre 2002 wurden 5 K3-Fragen gestellt, im Jahre 2003 8 und im Jahre 2004 14. Das Verhältnis der richtig beantworteten Fragen nahm von einem Jahr zum nächsten zu ($p < 0.01$ Mann-Whitney-Test). 2002: $N=49$, $\mu=57\%$. 2003: $N=94$, $\mu=62\%$. 2004: $N=77$, $\mu=79\%$.

10.3.6 Diskussion der Resultate der Outputanalyse

Dass die Studierenden im Jahre 2004 beim Beantworten von anwendungsorientierten Fragen signifikant besser waren als in den beiden Jahren zuvor, ist auf verschiedene Faktoren zurückzuführen. Ein wesentlicher Faktor ist vermutlich, dass die gesamte Umstellung des Unterrichts in ein *Blended Learning im 4-Schritte-Modell* erst zu diesem Zeitpunkt abgeschlossen war (Abb. 10.5). Die Tatsache, dass bereits in den Jahren 2002 und 2003 dieselben E-Learning-Materialien im Einsatz waren, als die Studierenden die anwendungsorientierten Fragen noch weniger gut meisterten, lässt den Schluss zu, dass nicht die E-Learning-Materialien allein, sondern deren Integration in den Unterricht durch die Umsetzung eines didaktischen Modells die entscheidenden Problemlösefähigkeiten hervorzubringen vermochten.

In dieser Diskussion sollen zwei Aspekte betrachtet werden:

- 1. Die Umstellung eines Kurses zu einem Blended Learning-Kurs mit 4-Schritte-Modell
- 2. Die Rolle der Leistungskontrolle für die Entwicklung einer Lernumgebung

1. Umstellung zu Blended Learning erfordert die Umstellung des gesamten Unterrichts

Diese Evaluation verstärkt die Überzeugung, dass es möglich ist, mit Hilfe von E-Learning-Materialien im Blended Learning eine deutliche Qualitätsverbesserung des Unterrichts herbeizuführen, die durch eine bessere Problemlösefähigkeit an der Prüfung zum

Ausdruck kommt. Es hat sich jedoch gezeigt, dass diese Wirkung erst dann eingetreten ist, als ein einheitliches, übergeordnetes didaktisches Modell eingeführt wurde, nach dem sich die einzelnen Elemente des Unterrichts organisieren. Es reicht nicht aus, wenn unter dem Argument des Blended Learning der traditionell theoriebasierte Unterricht mit E-Learning-Angeboten, die die Theorie anwenden, ergänzt wird. Auf diese Weise werden den Studierenden zwei Zugänge zum selben Inhalt geschaffen, obwohl es sich eigentlich um Wissen über das selbe Konzept handelt.

Das Entstehen von tragem Wissen in theoriebasierten Umgebungen zeigt, dass es für Novizen schwierig ist, die unterschiedlichen Wissensdimensionen der beiden Systeme miteinander in Verbindung zu bringen. Die Trennung zwischen Theorie und Praxis im traditionellen Unterricht erweist sich dabei als wenig hilfreich. Es muss deshalb *ein* einheitliches System geschaffen werden, welches erlaubt, dass die Lernenden die Theorie mit der Praxis verknüpfen und auf diese Weise ihr Wissen selber konstruieren. Das 4-Schritte-Modell löst dieses Problem auf effiziente Art und Weise, indem Konzepte durch das Lösen von Problemen direkt auf die praktische Ebene gebracht und mit Fertigkeiten verknüpft werden können. Die anschließende selbständige Anwendung in einer neuen Aufgabe und einer abschliessenden Reflexion darüber flexibilisiert das angeeignete Wissensnetz weiter und führt dazu, dass die Lernenden ihre Problemlösefähigkeiten weiter ausbauen. Das Entstehen einer Theorie-Praxis-Lücke wird dadurch gleich zu Beginn verhindert.

Für die Implementierung von E-Learning-Materialien im Blended Learning nach dem 4-Schritte-Modell bedeutet dies, dass die Bereitschaft aufgebracht werden muss, die traditionellen Lehrenden-zentrierten Unterrichtsstrukturen zu verändern und konstruktivistische Lernmethoden einzusetzen.

Eine weitere Voraussetzung für eine erfolgreiche Implementierung ist, dass die Leistungskontrolle den veränderten Bedingungen angepasst wird. Die Entwicklung von Unterricht und Prüfung zeigen, dass es nicht ausreicht, unter dem Argument des Wissens-Transfers E-Learning-Materialien zur Verfügung zu stellen, damit die Studierenden Theorie anwenden können, wenn an der Prüfung die Wiedergabe von Theoriefakten verlangt wird. Erst die Berücksichtigung der Anwendungsorientierung in der Leistungskontrolle bringt die Kompetenzorientierung ins System. Der zuvor stattfindende Unterricht soll direkt auf die Anforderungen der Leistungskontrolle vorbereiten.

2. Die Rolle der Leistungskontrolle für die Entwicklung der Lernumgebung

Leistungskontrollen spielen bei der Entwicklung von Lernumgebungen eine wichtige Rolle. Als Output des Unterrichts liefern sie wichtige Daten über die Wirksamkeit des Unterrichts. Evaluationsverfahren für den Unterricht erfassen typischerweise bloss Akzeptanz, Unterhaltungswert und kurzfristige Zufriedenheit der Studierenden im Unterricht. Die Effektivität einer ganzen Unterrichtseinheit, die etwas darüber aussagen sollte, *wie viel* und *was* die Studierenden gelernt haben, wird oft vernachlässigt. Dies ist

vor allem darauf zurückzuführen, dass Unterricht und Leistungskontrolle unabhängig voneinander betrachtet werden und die Evaluationsszenarien für den Unterricht zu einem Zeitpunkt stattfinden, wo der Lernprozess der Studierenden noch nicht abgeschlossen ist. Nicht selten steht er gar noch bevor. Um ein umfassenderes Bild zu erhalten, welche Kompetenzen eine Unterrichtseinheit hervorzubringen vermag, muss das Evaluationsverfahren den ganzen Lernprozess inklusive Leistungskontrolle umfassen. Für die Unterrichtsbeurteilung kommt dazu, dass die Lernenden erst nach einer Prüfung die Effektivität des Unterrichts beurteilen können.

Die Studierenden richten ihren Einsatz, den sie für eine Lehrveranstaltung investieren, nach den Anforderungen der Leistungskontrolle. Sind die Anforderungen anspruchsvoll, aber realistisch, und ist der gebotene Unterricht eine direkte Vorbereitung darauf, wird eine Lehrveranstaltung nicht nur motivierender, die Lernenden werden dafür auch mehr Zeit investieren und dadurch mehr lernen.

Kompetenzen müssen erst entwickelt werden, bevor sie geprüft werden können. Dieses Zusammenspiel zwischen Unterricht und Leistungskontrolle kann mit der Arbeit eines Dirigenten oder einer Dirigentin verglichen werden, der oder die sich zusammen mit Musikern auf ein Konzert vorbereitet. In der Probearbeit muss er oder sie die Musiker direkt auf eine definierte Leistung auf einem möglichst hohen Niveau vorbereiten.

Zusammenfassend halten wir fest, dass die Leistungskontrolle eine wichtige Rolle als Evaluationsinstrument von Unterricht einnehmen kann, um die Effektivität von Unterrichtseinheiten aufzuzeigen und zu verbessern.

10.4. Prozessanalyse: Überprüfung der Implementierungsqualität von Problemaufgaben in problembasierten Kursen

10.4.1 Übersicht

Ziel dieser Evaluation:		
Mit dieser Evaluation wurde die Qualität einer Implementierung von Problemaufgaben in problembasierten Kursen untersucht. Hierfür wurden durch eine Prozessanalyse Veränderungen in der eingeschätzten Effektivität von Problemaufgaben untersucht.		
Evaluationskriterien	Messgrößen	Evaluationsinstrumente
Der Effektivität von Problemaufgaben bleibt während eines Kurses stabil.	<ul style="list-style-type: none"> • Zeitaufwand • eingeschätzter Schwierigkeitsgrad • Zufriedenheit/Motivation 	Online-Prozessanalyse

Die Resultate dieser Evaluation wurden 2006 an der *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (ELEARN)* publiziert [Faessler 2006].

10.4.2 Einführung

Der Einsatz von problembasiertem Lernen (PBL) als Abfolge von problembasierter Instruktion und Anwendung mit dem 4-Schritte-Modell hat sich als sehr effektiv und motivierend herausgestellt (siehe Kapitel 5.4 „Effektives und motivierendes PBL“, Abb. 5.8). Die Umstellung eines Kurses von der traditionellen Form zu einem problembasierten Kurs mit problembasierter Instruktion und Anwendung ist mit einem umfangreichen Implementierungs- und Optimierungsprozess verbunden. Ein entscheidender Entwicklungsschritt ist dabei die Auswahl, respektive das Konstruieren von Problemaufgaben (siehe Tab. 8.1, Prozesse A und C3.1). Deren produktiver Einsatz in der Lernumgebung (Prozess D) führt zur Entwicklung von Lernmaterialien für die problembasierte Instruktion (Prozesse B und C). Die konstruierte oder ausgewählte Problemaufgabe legt den Kompetenzlevel einer Lerneinheit fest, die bereitgestellten Lernmaterialien helfen, die Effizienz des problembasierten Lernens zu erhöhen, indem die Lernenden auf das Lösen der Problemaufgabe vorbereitet werden. Umgekehrt müssen Problemaufgaben an das PBL angepasst werden.

Ein problembasierter Kurs setzt zum Aufbau einer Kompetenz eine Reihe von Problemaufgaben ein, die nach den beiden didaktischen Modellen „4-Schritte-Modell“ und „Anwendungsebenen-Modell“ unterrichtet werden (folglich Kapitel 7 „Didaktische Modelle“). Mit beiden didaktischen Modellen werden die Lernenden auf das Lösen eines umfangreichen, realistischen Problems vorbereitet. Das Anwendungsebenen-Modell kommt dann zum Einsatz, wenn die Komplexität eines zu vermittelnden Lerninhalts an-

spruchsvoll ist und über eine Reihe von Teilkompetenzen mit kleineren Problemen eingeübt werden muss (folglich Kapitel 7.3 „Das Anwendungsebenen-Modell“).

Egal nach welchem der beiden didaktischen Modelle Problemaufgaben entwickelt wurden, sie müssen letztendlich in einem Kurs so aufeinander abgestimmt sein, dass sie den individuellen Lernprozess über den gesamten Zeitraum unterstützen. Der Lerneffekt beim PBL ist dann am höchsten, wenn die Lernenden den Anforderungen der Problemaufgabe gewachsen sind und das Lösen zu Zufriedenheit und Selbstvertrauen führt (siehe Abb. 5.7). Eine positive Selbstwirksamkeitserwartung beeinflusst die Anfangsbedingungen, unter denen eine nächste Problemaufgabe gelöst wird (folglich Kapitel 5.3 „Motivationale und emotionale Faktoren des PBL“). Der Verlauf motivationaler Größen während eines Kurses sagt deshalb etwas über die Effektivität eines problembasierten Kurses aus.

10.4.3 Beispiel eines Implementierungsprozesses: Programmierkurs

In Kapitel 8 wurde als Beispiel einer Umsetzung der didaktischen Modelle als Implementation von PBL der Programmierkurs „Anwendungsnahes Programmieren“ (siehe Tab. 1.2) beschrieben. Als erste selbständige Aufgabe wurde die Programmierung des *Galton Boards* gewählt (siehe Abb. 8.4), dessen Umsetzung nach dem Anwendungsebenen-Modell erfolgte (Tab. 8.6, Problem 4). Die zur Umsetzung notwendigen Konzepte (Variablen, Bedingungsprüfung, Schleifen, Zufallszahlen und Array) werden durch kleinere Probleme nach dem 4-Schritte-Modell eingeübt (Abb. 10.7, siehe auch Tab. 8.6, Probleme 1 bis 3).

Diese Auswahl war eine Folge der Entwicklung der Jahre zuvor. Im Jahre 2003 wurde der Kurs noch traditionell mit Vorlesung und Theorieskript gehalten, während in einer begleitenden Übungsstunde Aufgaben gelöst wurden. Die Auswahl der Aufgaben wurde durch das Konzept des *Inverted Curriculums* beeinflusst, bei dem das Lösen von umfangreichen realen Programmieraufgaben eine Betrachtung von aussen nach innen ermöglichen soll [Meyer 1993]. Das Galton Board-Problem wurde als allererste Aufgabe gestellt (Abb. 10.7). Die zugrunde liegenden Konzepte wurden in Vorlesung und Skript eingeführt. Dieses Wissen war für das Lösen des Problems zu wenig aktiv und konnte von den Studierenden nicht mobilisiert werden. Die Folge war, dass die Studierenden beim Lösen des Problems überfordert waren, was nicht nur das Lernen am Problem verhinderte und die Überzeugung, ein nächstes Problem des Kurses lösen zu können, negativ beeinflusste (folglich Kapitel 5.3.2 „Selbstwirksamkeitserwartung“). Viele Studierende lösten die folgenden Aufgaben nicht mehr selber und gaben auf [unpublizierte Beobachtung 2003]. Der Lerneffekt, den man sich durch PBL verspricht, blieb in der Folge aus.

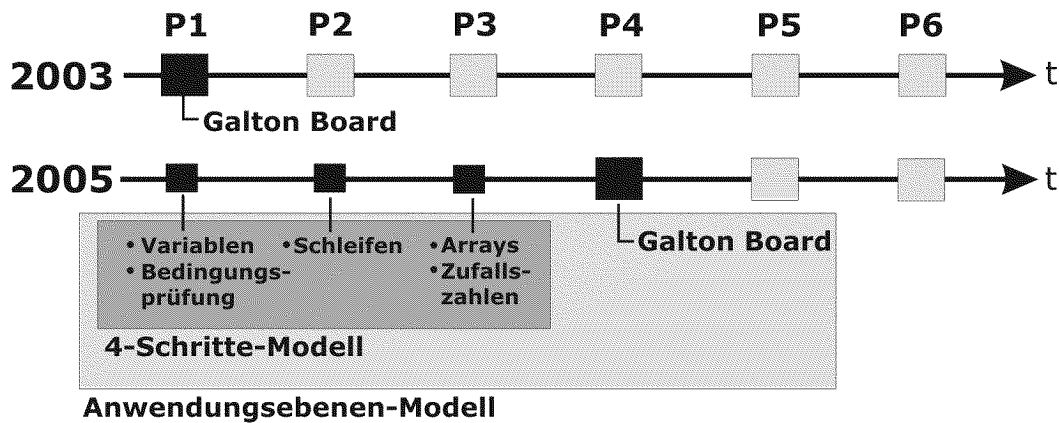


Abb. 10.7: Die Galton Board Aufgabe als erste Aufgabe konnte im Jahre 2003 von den meisten Lernenden nicht selbständig gelöst werden, was in der Folge zu Motivationsproblemen führte. 2004 wurden die Grundlage-Konzepte für die Galton Board-Aufgabe im Anwendungsebenen-Modell als Probleme P1 bis P3 eingeführt, die Galton Board-Aufgabe folgte als vierte Aufgabe (P4).

Die Einführung der Konzepte nach dem 4-Schritte-Modell (Tab. 7.3) soll zu aktivem und flexiblem Wissen und zur notwendigen Selbständigkeit führen, die Aufgabe selbständig lösen zu können (Abb. 10.7). Nach der Umstellung des Unterrichts nach den beiden didaktischen Modellen („4-Schritte-Modell“ und „Anwendungsebenen-Modell“) konnten wir im Jahre 2005 bei der Galton Board-Aufgabe feststellen, dass die Studierenden nicht nur in der Lage waren, die Aufgabe selbständig zu lösen, wir konnten auch einen starken Motivationsanstieg beobachten.

Um solche Motivationsverläufe zusammen mit anderen Größen über den Zeitraum ganzer Kurse zu quantifizieren, führten wir in unseren Kursen über mehrere Jahre Prozessanalysen durch.

10.4.4 Durchführung der Prozessanalyse

Bei einer Prozessanalyse werden zu verschiedenen Zeitpunkten eines Prozesses vergleichbare Werte erfasst, um Veränderungen in einem laufenden Prozess untersuchen zu können (siehe Kapitel 10.2.2, Abb. 10.2). Die Prozessanalyse wird so durchgeführt, dass die Lernenden für jedes gelöste Problem je einen Wert zu den Variablen „Zeitaufwand“, „eingeschätzter Schwierigkeitsgrad“ und „Zufriedenheit“, respektive „Motivation“ angeben können (erfasste Variablen und mögliche Antworten siehe Tab. 8.14). Um die Werte miteinander in Verbindung zu bringen und gleichzeitig die Anonymität der Umfrage zu gewährleisten, werden die Studierenden aufgefordert, ihre Antwort mit einem mnemonischen Kürzel zu identifizieren. Die Daten für diese Prozessanalyse werden durch einen Online-Fragebogen gesammelt, den die Lernenden jedes Mal ausfüllen und abschicken, wenn sie ein *E.Tutorial*[®] beendet haben (siehe Abb. 10.1). Der Online Fragebogen wurde am Ende eines *E.Tutorials*[®] als HTML-Formular direkt ins Instruktionfenster eingebaut. Dies hat den grossen Vorteil, dass keine weiteren Umfrageapplikationen eingesetzt werden müssen. Die Prozessdaten werden

kontinuierlich gesammelt, in einer Log-Datei abgespeichert und anschliessend ausgewertet (siehe Abb. 8.13).

10.4.5 Resultate der Prozessanalyse

Die Abb. 10.8 und Abb. 10.9 zeigen den Verlauf der Prozessdaten während des Programmierkurses (ETH Zürich, Kurse 251-0842-00L und 251-0846-00L, Tab. 1.2) und des ICT-Kurses (ETH Zürich, Kurs 252-0432-00L, Tab. 1.2) aus dem Jahre 2005 für folgende drei Variablen:

- Anteil der Studierenden, die mit einem *E.Tutorial*[®] *mehr als 4 Stunden* aufzuwenden hatten.
- Anteil der Studierenden, die den Schwierigkeitsgrad des *E.Tutorials*[®] als *schwieriger* einschätzten als im *E.Tutorial*[®] zuvor.
- Anteil der Studierenden, die mit einem *E.Tutorial*[®] *lieber oder viel lieber* gearbeitet haben als mit dem *E.Tutorial*[®] zuvor.

Folgende Zusammenhänge zwischen den erfassten Variablen konnten gefunden werden:

1. Vergleich der Motivation mit der eingeschätzten Schwierigkeit und der aufgewendeten Zeit

Eine numerische Analyse der Prozessdaten von allen *E.Tutorials*[®] zeigt eine hoch signifikante inverse Korrelation zwischen der Motivation und der für ein *E.Tutorial*[®] aufgewendeten Zeit ($r = -.17$; $p < 0.01$). Ebenfalls hoch signifikant ist der inverse Zusammenhang zwischen den Variablen Motivation und dem eingeschätzten Schwierigkeitsgrad ($r = -.30$; $p < 0.01$) (siehe Abb. 10.8 und Abb. 10.9).

2. Motivation und Einüben von grundlegenden Konzepten

Die Abnahme der Motivation während des einführenden ICT-Kurses nach ET₃ (Abb. 10.9, B) ist signifikant (Zweiseitiger t-Test, $t(108) = 5.70$; $p < 0.01$). In Abb. 10.8 kann beobachtet werden, dass während des Programmierkurses die Motivation ebenfalls signifikant abnahm, als die Studierenden nach dem Programmieren des Galton Boards in *E.Tutorial*[®] 4 wieder Grundlagen-Konzepte üben mussten (*E.Tutorial*[®] 5) (Zweiseitiger t-Test, $t(106) = 2.37$; $p < 0.05$).

3. Motivation beim Lernen von Grundlage-Konzepten, welche in eine komplexere Aufgabe eingebettet sind

Die Motivation nach dem *E.Tutorial*[®] 4 des Programmierkurses (Abb. 10.8, A) ist signifikant höher als nach *E.Tutorial*[®] 3 (Zweiseitiger t-Test, $t(100) = -3.06$; $p < 0.01$). Die Zu-

nahme der Motivation von *E.Tutorial*® 6 zu *E.Tutorial*® 7 ist ebenfalls signifikant (Zweiseitiger t-Test, $t(47) = -2.81$; $p < 0.01$).

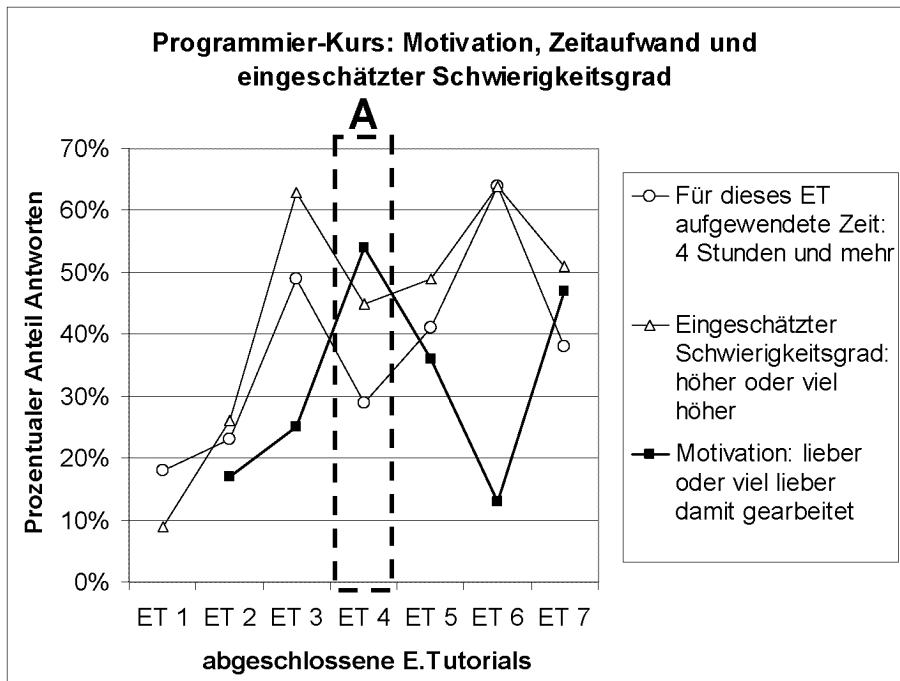


Abb. 10.8: Prozessanalyse des Programmierkurses (2005, 292 Studierende, mittlere Rücklaufquote: 75%): ET1: Variablen, Bedingungsprüfung, ET2: Iteration, ET3: Zufallszahlen, Arrays, ET4: Simulation (Galton Board), ET5: Objekte, Liniengrafik, ET6: GUI-Komponenten, ET7: Modellieren, Software-Entwicklung. Markierung A siehe Text.

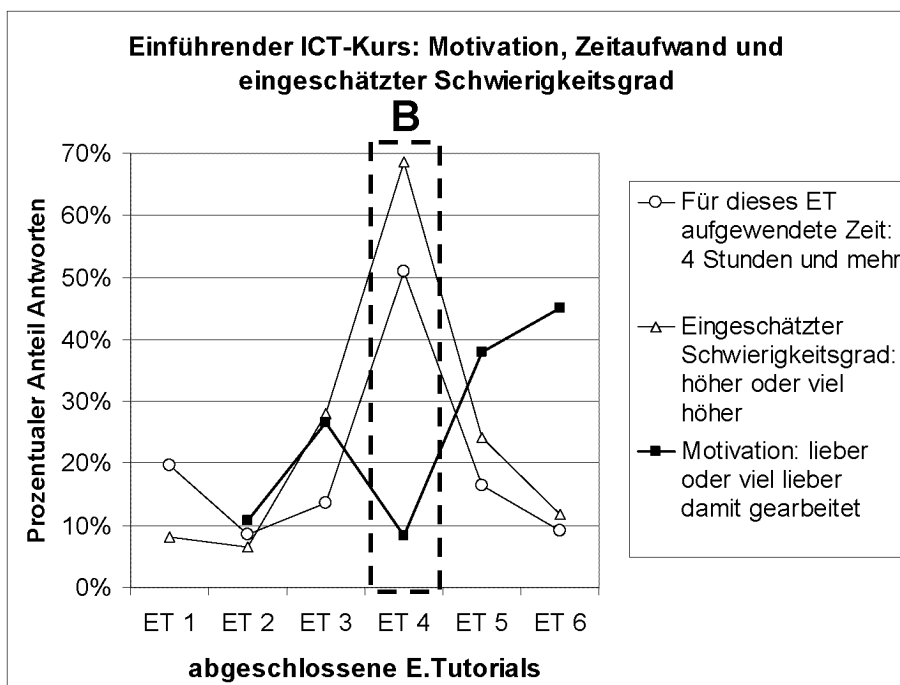


Abb. 10.9: Prozessanalyse des ICT-Kurses (2005, 246 Studierende, mittlere Rücklaufquote: 57%): ET1: Internet, WWW, ET2: Simulieren mit Excel, ET3: Visualisieren multivariater Daten, ET4: Daten verwalten mit Listen und Tabellen, ET5: Daten verwalten mit relationalen Datenbanken, ET6: Makroprogrammierung mit Excel. Markierung B siehe Text.

10.4.6 Diskussion der Resultate der Prozessanalyse

Folgende drei Punkte fassen unsere Befunde aus den Prozessanalysen der beiden Abb. 10.8 und Abb. 10.9 zusammen:

- 1. Die Motivation beim Lösen von Problemen kann vom eingeschätzten Schwierigkeitsgrad und von der aufgewendeten Zeit abhängen.
- 2. Das Einüben von Grundlagen kann die Motivation senken.
- 3. Die Motivation steigt wieder an, wenn danach eine umfangreichere Anwendungsaufgabe mit Praxisbezug gelöst wird.

1. Die Motivation beim Lösen von Problemen kann vom eingeschätzten Schwierigkeitsgrad und von der aufgewendeten Zeit abhängen

In den Abb. 10.8 und Abb. 10.9 ist immer dann eine Abnahme der Motivation zu sehen, wenn die Studierenden eine zeitaufwändige und anspruchsvolle Aufgabe zu lösen hatten (Abb. 10.8, ET 6 und Abb. 10.9, ET 4). Das gemeinsame der beiden Aufgaben ist, dass die Zeit, die von den Kursentwicklern für die Aufgaben eingerechnet wurden, unterschätzt wurde. In beiden Fällen stieg die Motivation auch in der nachfolgenden, weniger aufwändigen Aufgabe wieder an.

Es ist auffällig, wie in beiden Abbildungen der eingeschätzte Schwierigkeitsgrad und die aufgewendete Zeit eine ähnliche Kurvenform aufweisen und wie die Motivation fällt, sobald die beiden zu stark ansteigen. Diese Beobachtung stimmt mit dem Risiko-Wahl-Modell von Atkinson überein, das eine maximale Motivation voraussagt, wenn die zu lösende Aufgabe einen mittleren Schwierigkeitsgrad aufweist [Atkinson 1957].

Mit der Galton Board-Aufgabe des Programmierkurses konnten wir zeigen, dass der eingeschätzte und nicht der objektive Schwierigkeitsgrad die Motivation beeinflusst (Abb. 10.8). Im Jahre 2003, als das Galton Board die erste zu lösende Programmieraufgabe war, sank die Motivation stark ab. Diese Feststellung basiert auf persönlichen Beobachtungen. Es führte dazu, dass die Aufgabe nicht selbständig gelöst, sondern von Mitstudierenden kopiert wurde, um die Bedingungen der Abgabe zu erfüllen. Ein wichtiger Lerneffekt blieb aus. Im Jahre 2005 wies die Aufgabe immer noch den selben objektiven Schwierigkeitsgrad auf, sie wurde subjektiv aber als viel weniger schwierig eingeschätzt, weil drei *E.Tutorials*® zur Erarbeitung der Programmiergrundlagen vorangestellt wurden.

Ob eine Problemaufgabe motivierend ist, hängt also nicht von der Aufgabe selber ab, sondern von der Tatsache, ob die Lernenden das nötige Wissen zur Lösung einer Aufgabe mitbringen.

2. Das Einüben von Grundlagen kann die Motivation senken

Ein anderer Fall einer Motivationsabnahme tritt dann auf, wenn Studierende neue Konzepte einüben müssen, nachdem sie bereits auf das Anwendungsniveau gewechselt haben. Ein Beispiel dafür ist in Abb. 10.8 nach dem Galton Board (*E.Tutorial*[®] 5) zu beobachten. Dies hat höchst wahrscheinlich damit zu tun, dass die Lernenden in einem Inhalt bereits Sicherheit und Gewandtheit erfahren haben und danach wieder zum Einüben von neuen Inhalten zurückkehren mussten. Der Sinn der neuen Konzepte wird erst in der Anwendung erfahren, die nach dem *E.Tutorial*[®] und somit auch nach dem Erfassungszeitpunkt der Prozessanalyse stattfindet. Die Motivationskurve scheint daher immer auch ein Spiegel der Sicherheit oder Gewandtheit der Studierenden im Umgang mit einem Konzept zu sein.

3. Die Motivation steigt wieder an, wenn nach dem Einüben der Konzepte eine umfangreichere Anwendungsaufgabe mit Praxisbezug gelöst wird

Neu gelernte Konzepte in einem umfangreicheren realen Problem anzuwenden und dabei den eigenen Fortschritt zu realisieren, ist motivierender, als ein isoliertes Konzept an Hand eines Beispiels einzuüben. Die umfangreichere Aufgabe ist jedoch erst dann motivierend, wenn die Studierenden die zugrunde liegenden Konzepte und Fertigkeiten so weit beherrschen, dass ein effizientes und kompetentes Lösen der Aufgabe möglich ist. Die Prozessanalyse zeigt beim *E.Tutorial*[®] 4 des Programmierkurses (Abb. 10.8, Markierung A), dass die vorgängigen drei praktischen Einführungen der wichtigsten drei Konzepte dazu führten, dass die Studierenden jene Gewandtheit und das Verständnis der Konzepte erreichten, dass sie die Galton Board- Aufgabe nicht nur weniger schwierig einstufen als die vorgängigen Aufgaben, sondern auch lieber damit arbeiten. Der entscheidende Unterschied zwischen den Grundlagen- *E.Tutorials*[®] 1 bis 3 und dem *E.Tutorial*[®] 4 ist, dass die Konzepte nicht mehr isoliert auftreten, sondern bei der Umsetzung eines Algorithmus angewendet werden müssen und miteinander in Beziehung stehen (folglich Kapitel 5.2.2 „Typen von Problemaufgaben“). Um dies bewältigen zu können, brauchen die Lernenden ein differenziertes, flexibles und anwendbares Wissensnetz, das sie sich zuvor im 4-Schritte-Modell anzueignen haben. Das Design des *E.Tutorials*[®] mit der Galton Board-Aufgabe unterscheidet sich von den vorangehenden dadurch, dass die Unterstützung bei der Entwicklung des Algorithmus ansetzt und die Umsetzung weitgehend vom Lernenden selbst erledigt werden kann. Viele Studierende verzichteten auch auf den Gebrauch des *E.Tutorials*[®] und programmierten das Galton Board selbständig. Auf diese Weise sind zum Teil eigene kreative Lösungen entstanden, was ebenfalls auf das Vorhandensein eines aktiven und flexiblen Wissensnetzes zurückzuführen ist.

10.4.7 Folgerungen aus den Prozessanalysen

Folgerungen für die Auswahl und Konstruktion von Problemaufgaben für das PBL

Für die Vermittlung von Informatik-Kompetenzen ist es notwendig, zur Umsetzung des PBL grosse Probleme mit Realitätsbezug auszuwählen. Auf diese Weise können sich die Lernenden Problemlösefähigkeiten aneignen, die in der Komplexität den Anforderungen der realen Welt entsprechen. Grosse Problemaufgaben schaffen den zur Aneignung von Kompetenzen notwendigen Raum und eine definierte Anforderung, die es zu bewältigen gilt. Dies sind auch die Hauptargumente der Methode des *Inverted Curriculums*, bei der sich Studierende eine Kompetenz dadurch aneignen, indem sie komplexe und umfangreiche Probleme von aussen nach innen betrachten und schrittweise versuchen, diese zu verstehen und zu modifizieren [Meyer 1993]. Die Schwierigkeit beim Lernen mit grossen Problemaufgaben ist aber, dass viele Lernende durch die Komplexität und Anforderungen der Problemaufgabe überfordert sind und es in der Folge zu Motivationsproblemen kommen kann (siehe Kapitel 5.3 „Motivationale und emotionale Faktoren des PBL“). Dies kann dazu führen, dass Sie zwar den Umgang mit grossen Problemen lernen, das wichtige Einüben von Grundlagentheorien und -fertigkeiten aber zu kurz kommt.

Mit Hilfe von Prozessanalysen konnten wir zeigen, dass es möglich ist, Lernende durch problembasierte Instruktionen nach dem 4-Schritte-Modell so auf die Anforderungen grosser Anwendungsaufgaben vorzubereiten, dass ein Grossteil der Lernenden maximal von der Methode des PBL profitieren kann. Dies zeigt sich in der Prozessanalyse dadurch, dass die Motivation der Lernenden beim Lösen von grossen Problemen im Verlaufe eines Semesters nicht einbricht, sondern im Gegenteil, sogar zunimmt, weil sie den Anforderungen der Aufgabe genügen können.

Prozessanalyse als Instrument für die Qualitäts- und Ablaufkontrolle

Mit der Prozessanalyse haben wir ein Instrument geschaffen, um folgende Aufgaben auszuführen:

- 1. Die Qualität der Implementierung einzelner Problemaufgaben bei der Entwicklung eines problembasierten Kurses überprüfen.
- 2. Den Ablauf eines Kurses bei einer fertig entwickelten Lernumgebung kontrollieren.

1. Prozessanalyse als Instrument zur Qualitätsverbesserung problembasierter Kurse

PBL ist erst dann effektiv und motivierend, wenn es zu Zufriedenheit und Selbstvertrauen der Lernenden beim Lösen von Problemen führt. Bei der Durchführung von problembasierten Kursen mit grossen Studierendenzahlen, die unabhängig von Ort

und Zeit und mit einem hohem Anteil an Selbststeuerung stattfinden, sind wir darauf angewiesen, dass wir uns während verschiedener Zeitpunkte eines Kurses ein Bild darüber verschaffen können, wie sich die individuellen Lernprozesse der Studierenden im Verlaufe eines Semesters entwickeln. Mit Hilfe der Prozessanalyse haben wir die Möglichkeit, beim problembasierten Lernen von grossen Gruppen Einblicke in die individuellen Lernprozesse zu erhalten. Sie liefert uns eine Grundlage für die Verbesserung des Kurses, indem geeignete Massnahmen getroffen werden können, um die Lernumgebung von Kurs zu Kurs zu optimieren. Bei der Durchführung einer Prozessanalyse lässt sich mit der Erfassung der drei Variablen „aufgewendete Zeit“, „eingeschätzter Schwierigkeitsgrad“ und „Arbeitsmotivation“ die Effizienz und Effektivität einzelner Problemaufgaben des PBL visualisieren. Die Einfachheit der Analyse garantiert eine hohe Rücklaufquote, weil die Teilnahme für die Studierenden mit wenig zusätzlichem Aufwand verbunden ist. Die direkte Integration des Fragebogens in die E-Learning-Materialien hat den Vorteil, dass keine weiteren Umfrageapplikationen eingesetzt werden müssen, was die Effizienz der Analyse zusätzlich erhöht.

2. Prozessanalyse als Ablaufkontrolle und Fine-Tuning-Instrument einer Lernumgebung

Auch nach der Entwicklung einer Lernumgebung liefert die Prozessanalyse wichtige Daten für die Ablaufkontrolle und das Fine-Tuning einer Lernumgebung bei der Durchführung eines Kurses. Die Entwicklung der Lernumgebung zur Durchführung von ICT-Kursen war 2005 abgeschlossen (siehe Prozessanalyse in Abb. 10.9). Abb. 10.10 zeigt Prozessanalysen weiterer vier Durchführungen desselben ICT-Kurses für zwei verschiedene Zielgruppen. Ein Vergleich des Verlaufes der Kurvenformen zeigt, mit welcher Stabilität die Lernumgebung eingesetzt werden konnte. Die Studierenden arbeiteten mit den *E.Tutorials*® im Verlaufe eines Kurses immer lieber, der Motivationseinbruch bei *E.Tutorial*® 4, wie er 2005 noch aufgetreten war (Abb. 10.9, Markierung B), konnte durch einige Verbesserungen im *E.Tutorial*® und anderen Medien des 4-Schritte-Modells weitgehend behoben werden (Abb. 10.10).

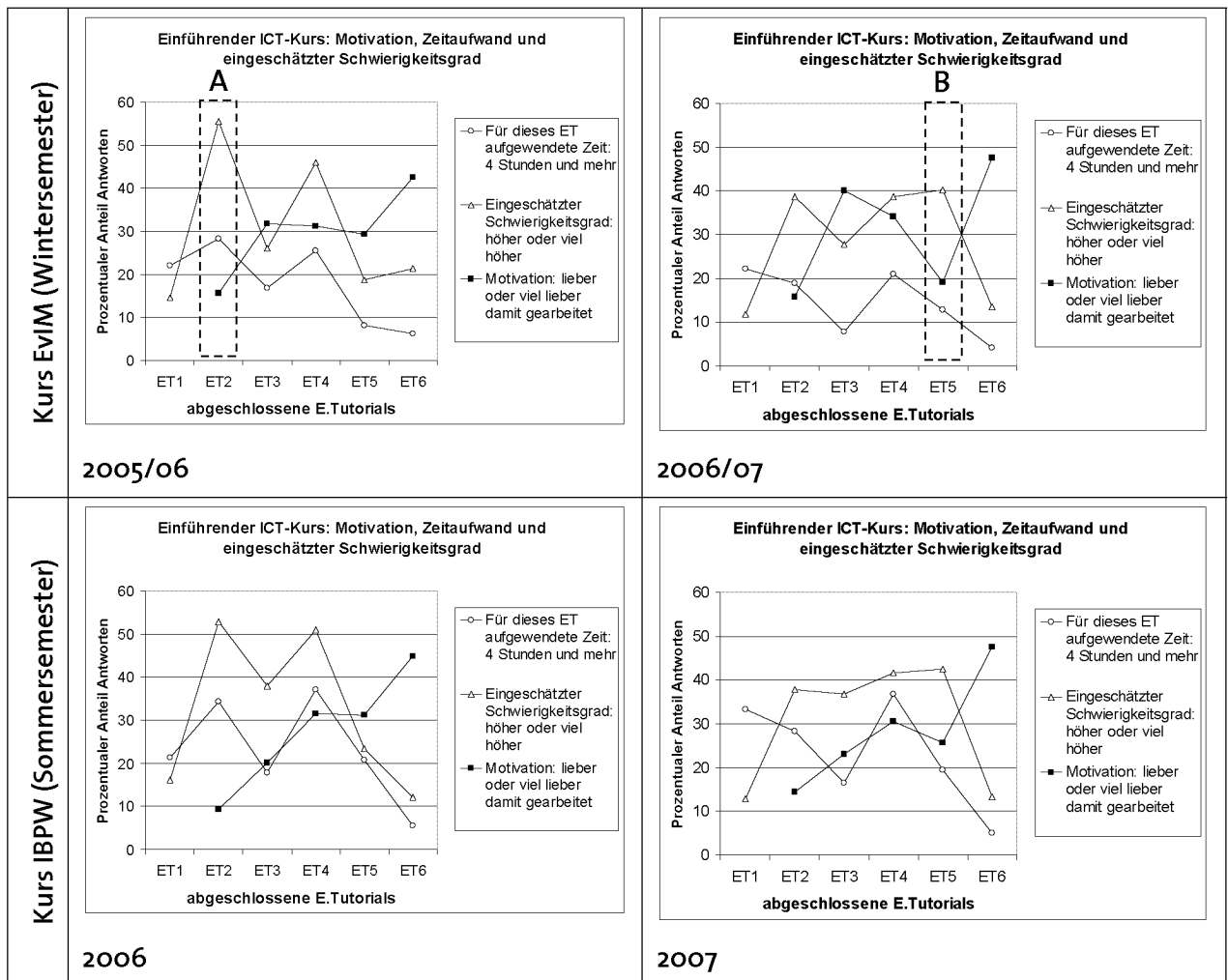


Abb. 10.10: Prozessanalysen durchgeführter ICT-Kurse in den Jahren 2005 bis 2007. Der Kurs IBPW (Informatik für Biologie und Pharmazeutische Wissenschaften, siehe Tab. 1.2) findet im Sommersemester statt und ist geprüft. Der Kurs EvIM (Einsatz von Informatikmitteln, siehe Tab. 1.2) findet im Wintersemester statt und ist ungeprüft. Anzahl Studierende (N) und durchschnittliche Rücklaufquoten (RQ): EvIM 2005/06: N=287, RQ= 54.6%, IBPW 2006: N=261, RQ= 46.9%, EvIM 2006/07: N=494, RQ=52.1%, IBPW 2007: N=278, RQ= 49.8%. Markierungen A und B siehe Text.

E.Tutorial[®] 2 wurde im Jahre 2005/06 durch ein neues, anspruchsvolleres *E.Tutorial*[®] ersetzt, welches einige zusätzliche Konzepte vermittelt (Abb. 10.10, Markierung A). Weil ein neues *E.Tutorial*[®] meist noch Fehler enthält und die einzelnen Elemente des 4-Schritte-Modells aufeinander abgestimmt werden müssen, dauert dessen vollständige Integration in eine Lernumgebung jeweils einige Semester.

Im Semester 06/07 wurde eine überarbeitete Version des *E.Tutorials*[®] 5 (Abb. 10.10, Markierung B) eingesetzt, die ein paar zusätzliche Konzepte behandelt. Die Lernenden hatten dafür nicht mehr Zeit aufzuwenden, sie stufte es aber als schwieriger ein als die Studierenden die Vorgängerversion. Dies war mit einem leichten Motivationsabfall verbunden, der im darauf folgenden Semester aber reduziert werden konnte.

Der Verlauf der Prozessdaten im Sommersemester 2007 zeigt annähernd den Idealfall eines problembasierten Kurses auf: Die aufzuwendende Zeit und der eingeschätzte Schwierigkeitsgrad bleiben stabil und die Lernenden lösen die Aufgaben immer lieber.

Lernumgebungen sind komplexe Systeme, deren Eigenschaften wir im operativen Einsatz mit Hilfe von Prozessanalysen erfassen können. Jede Lernumgebung hat dabei ihr charakteristisches Muster. Die Effektivität einer problembasierten Lernumgebung ist dann am höchsten, wenn die Zufriedenheit und Motivation der Lernenden im Verlaufe des Prozesses gesteigert werden kann. Soll am System etwas verändert werden (z.B. das Einfügen eines neuen *E.Tutorials*®), darf der Motivationsabfall nicht allzu stark ausfallen, sonst sinkt die Effektivität der Lernmethode.

10.5. Weiterführende Evaluationen

Eine interessante Weiterführung der Evaluation der entwickelten Lernumgebungen wäre eine Kombination von Prozess- und Leistungsdaten. Damit könnte das Zusammenspiel zwischen Unterricht und Leistungskontrolle genauer unter die Lupe genommen werden. Es könnten beispielsweise unterschiedliche Leistungsgruppen von Studierenden betrachtet und mit dem Verlauf der zugehörigen Prozessdaten des Unterrichts verglichen werden. Dabei wäre interessant zu sehen, ob im Unterricht Unterschiede zwischen den Studierenden, die bei der Leistungskontrolle gut oder weniger gut abgeschnitten haben, zu beobachten sind.

Kapitel 11

Schlussfolgerungen und Ausblick

In diesem Kapitel sollen die in dieser Arbeit gewonnenen Erkenntnisse zusammengefasst und die daraus sich aufdrängenden Möglichkeiten zur Weiterentwicklung der computergestützten Lernumgebungen aufgezeigt werden.

11.1. Schlussfolgerungen

Die Schlussfolgerungen dieser Arbeit konzentrieren sich auf folgende drei Bereiche:

- Wichtige Rolle der Informatik als Enabling Science
- Qualitätsverbesserung des Informatik-Grundlagenunterrichts durch kompetenzorientiertes E-Learning
- Entwickeln von Lernumgebungen als multidisziplinäre Disziplin

11.1.1 Wichtige Rolle der Informatik als Enabling Science

Die Informatik spielt heute bei der Ausübung von Tätigkeiten in vielen Bereichen (wie z.B. in der wissenschaftlichen Tätigkeit) eine Schlüsselrolle als *Enabling Science*. Für die Zielgruppe solcher Fachbereiche bedeutet dies, dass sie zur Ausübung solcher Tätigkeiten direkt davon abhängig sind, ob sie den Computer zusammen mit spezialisierten, professionellen Informatikmitteln effizient zur Lösung von Problemen einsetzen können. Diese Fähigkeit setzt das Vorhandensein von anwendbaren Wissensstrukturen aus dem Bereich der Informatik-Konzepte als aktiv einsetzbare „Brainware“ voraus. Der Informatikunterricht hat im Grundlagenunterricht einer Ausbildung für solche Fachbereiche den Auftrag, die Zielgruppe auf diese anspruchsvolle Aufgabe vorzubereiten. Weil die Trennung von Theorie und Anwendung, wie sie im Unterricht häufig vorgenommen wird, zu trägem statt zu aktivem Wissen führt, müssen neue Wege bei der Vermittlung von Informatik-Grundlagen gesucht werden. Die Lösung liegt im Ansatz, Informatik-Konzepte in Form von definierten Kompetenzen zu vermitteln, was die Aneignung von aktiven Wissensstrukturen ermöglicht. Mit den von uns entwickelten kompetenzorientierten Lernumgebungen unterrichten wir Informatik-Konzepte direkt mit konkreten Anwendungen, damit sich die Lernenden aktives Wissen aneignen.

Die Informatik nimmt als Enabling Science bei der interdisziplinären Vermittlung von Kompetenzen diverser Bereiche eine Sonderstellung ein, die es bei der Gestaltung des Unterrichts zu nutzen gilt. Andere Fachbereiche sollten dabei so unterstützt werden, dass sie selbständig ICT-Kompetenzen vermitteln können.

11.1.2 Qualitätsverbesserung des Informatik-Grundlagenunterrichts durch kompetenzorientiertes E-Learning

Der Einsatz von E-Learning bringt dann eine Qualitätsverbesserung des Unterrichts, wenn gegenüber traditionellen Unterrichtsmethoden und Medien ein Mehrwert geschaffen werden kann. Die von uns entwickelten Lernumgebungen generieren diesen Mehrwert, indem bereits im Grundlagenunterricht unter Einsatz eines Computers Kompetenzen vermittelt werden. Der Computer wird dabei sowohl als Instruktionsmedium als auch als Werkzeug zum Lösen von realitätsnahen Problemen eingesetzt. Diese Kombination ermöglicht eine selbstgesteuerte, problembasierte Aneignung von Kompetenzen, weil sie die Ausführung von realen Handlungen und das Sammeln von persönlichen Erfahrungen fördert.

Unsere Evaluationen haben gezeigt, dass der Aufbau von Kompetenzen nur gelingt, wenn alle Elemente einer Lernumgebung durch ein einheitliches didaktisches Modell organisiert sind und die Leistungskontrolle ebenfalls auf die veränderten Bedingungen einer Kompetenzorientierung angepasst wird.

E.Tutorial[®]-basierter Unterricht im 4-Schritte-Modell nimmt in der Welt des E-Learnings eine Sonderstellung ein, weil es den Computer in der Lehre da einsetzt, wo seine Stärken liegen, nämlich als Werkzeug zur Lösung von spezifischen Problemen.

11.1.3 Entwickeln von Lernumgebungen als multidisziplinäre Disziplin

So wie die Ingenieurwissenschaft Technologie umsetzt, braucht eine Wissensgesellschaft eine integrierende Wissenschaft, die Inhalte unter Berücksichtigung von didaktischen, pädagogischen und psychologischen Aspekten so umsetzt, dass funktionierende Lernumgebungen entstehen, mit deren Hilfe Wissen nachhaltig und effektiv vermittelt werden kann. Die Aufbereitung der Lerninhalte steht als „Rohmaterial“ der Lehre im Zentrum eines solchen *Educational Engineerings*. Da jeder Lerninhalt bei der Vermittlung seine Besonderheiten aufweist, ist diese Disziplin mit Vorteil der Fachdidaktik eines Fachbereichs anzugliedern.

In dieser Arbeit wurde ein Szenario für die Entwicklung und Umsetzung kompetenzorientierter Lernumgebungen beschrieben. Wir konnten zeigen, dass es möglich ist, Lernumgebungen zu entwickeln, die zum grossen Teil unabhängig von Ort und Zeit bearbeitet werden können und ein effektives und motivierendes Lernen ermöglichen. Wesentliche zu entwickelnde Elemente sind authentische Problemstellungen und Lernmaterialien, die sich auf die Problemstellungen beziehen. Für den Einsatz im Unter-

richt sind zusätzlich Betreuungspersonal und effiziente Ablaufstrukturen bereitzustellen. Alle Elemente der Lernumgebung müssen schlussendlich in einem System zusammenspielen und stehen im Dienst eines effizienten Transfers von Lerninhalten.

Die Aufbereitung von Lerninhalten ist nicht automatisierbar. Bei jedem neu zu vermittelnden Lerninhalt steht man wieder vor der selben Aufgabe: Wie lassen sich Konzepte auf effiziente Art und Weise vermitteln, damit sie von einer Zielgruppe individuell konstruiert, möglichst lange präsent bleibt, weiterentwickelt und vielfältig verwendet werden können.

11.2. Grundlagen für die Weiterentwicklung der Lernumgebungen

Im Zusammenhang mit dieser Arbeit sind folgende Outputs entstanden (siehe Abb. 1.2):

- Zwei kompetenzorientierte, computergestützte Lernumgebungen für den Informatikunterricht für Studierende der Natur- und Bauingenieurwissenschaften (für die Kompetenzen *Einsatz von ICT* und *Programmieren*)
- Zwei didaktische Modelle für das Unterrichten von Kompetenzen durch den Einsatz des Computers im Unterricht (*4-Schritte-Modell* und *Anwendungsebenen-Modell*)
- Ein Szenario für die Entwicklung neuer Lernumgebungen durch die Umsetzung der beiden didaktischen Modelle.

Die Weiterentwicklungsmöglichkeiten der in dieser Arbeit entstandenen Outputs sind vielfältig und können ganz unterschiedliche Einsatzgebiete betreffen. Zwei Schwerpunkte der Weiterentwicklung ergeben sich auf der Seite des Computers durch die unterschiedlichen Einsatzmöglichkeiten im Unterricht und auf der Seite der Lernenden oder der Lerninhalte durch den Bedarf nach kompetenzorientiertem Unterricht direkt an einem Computer.

1. Einsatzmöglichkeiten des Computers im Unterricht

Bei den im Rahmen dieser Arbeit entwickelten Lernumgebungen kommt der Computer gleichzeitig auf verschiedenen Ebenen zum Einsatz: Er ist erstens selber Lerninhalt und Gegenstand des Unterrichts (Einsatz des Computers im Informatikunterricht), er hilft zweitens beim Lösen von Problemen anderer Fachrichtungen (Computer als Werkzeug) und ist drittens Instruktionsmedium mit hohem Selbststeuerungsanteil (Einsatz des Computers als Medium beim E-Learning) (Abb. 11.1). Obwohl sich die Zielsetzungen der drei Einsatzebenen des Computers unterscheiden, können sie durch das Konzept des *Informationsarbeitsplatzes* integriert und auf eine Ebene gebracht werden, um Informatik-Konzepte durch das Lösen von Problemen direkt am Computer zu unterrichten (siehe Kapitel 3, „ICT-Kompetenzen für die wissenschaftliche Tätigkeit“). Die Kombina-

tion der in Abb. 11.1 skizzierten Ebenen im Informationsarbeitsplatz eröffnet viele Weiterentwicklungsmöglichkeiten (siehe Abschnitte 11.3 und 11.4).

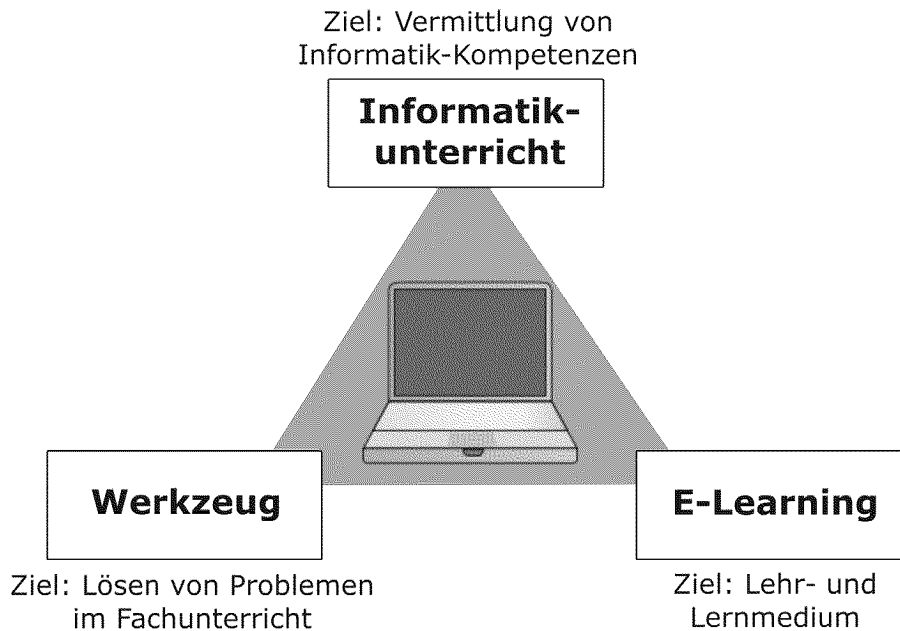


Abb. 11.1: Unterschiedliche Einsatzebenen und Zielsetzungen des Computereinsatzes im Unterricht. Als Informationsarbeitsplatz ist der Computer gleichzeitig Gegenstand des Informatikunterrichts, Werkzeug und Lernmedium.

2. Kompetenzorientierter Unterricht direkt am Computer

Für den professionellen Einsatz von Computern zum Lösen von Problemen aller Art ist aktives Wissen erforderlich, welches durch die verknüpfte Aneignung von Konzept- und Anwendungswissen zustande kommt (siehe Kapitel 2 „Kompetenzorientierter Unterricht“). Lernumgebungen, die den Computer als Informationsarbeitsplatz einsetzen, eröffnen die Möglichkeit, Konzepte zusammen mit einer konkreten Anwendung als Kompetenz zu unterrichten und durch den geschaffenen Kontext auf eine spezifische Zielgruppe auszurichten.

Die Weiterentwicklungsmöglichkeiten werden in diesem Kapitel wie folgt diskutiert:

- Weiterentwicklung der im Rahmen dieser Arbeit erstellten Lernumgebungen zu den Kompetenzen *Einsatz von ICT* und *Programmieren* (Abschnitt 11.3)
- Erstellen neuer computergestützter Lernumgebungen (Abschnitt 11.4).

11.3. Weiterentwicklung der Lernumgebungen „Einsatz von ICT“ und „Programmieren“

Im Rahmen dieser Arbeit sind zwei Lernumgebungen (*Einsatz von ICT* und *Programmieren*) für die Zielgruppen Naturwissenschaftler und Bauingenieure entwickelt worden, mit deren Hilfe Konzepte der Informatik interdisziplinär anhand von realen Problemen kompetenzorientiert unterrichtet werden können. Die Weiterentwicklung der beiden Lernumgebungen konzentriert sich auf folgende Punkte:

- Entwicklung eines Systems für die Leistungskontrolle (Abschnitt 11.3.1)
- Anpassen der Lernumgebungen auf weitere Zielgruppen (Abschnitt 11.3.2)
- Einfügen neuer Lerninhalte in die Lernumgebungen (Abschnitt 11.3.3)

11.3.1 Entwicklung eines Systems für die Leistungskontrolle

Die im Rahmen dieses Projekts entwickelten Lernumgebungen führen bei den Lernenden im Unterricht zu definierten Kompetenzen. Neben dem Aufbau von Kompetenzen ist das Prüfen, ob sich die Studierenden die erforderlichen Kompetenzen angeeignet haben, ein weiteres wichtiges Ziel des Unterrichts. Für die Weiterentwicklung der Lernumgebungen bedeutet dies, dass neben der Entwicklung von Lernmaterialien und Ablaufstrukturen zusätzlich ein System für die Leistungskontrolle entwickelt und in die Lernumgebung integriert werden muss.

Wie bereits in Kapitel 10 („Evaluation der Lernumgebungen“, Abschnitt 10.3) diskutiert, ist die Qualitätsverbesserung des gesamten Unterrichts von der Art und Weise abhängig, wie der Unterricht schlussendlich geprüft wird. Umgekehrt kann die Leistungskontrolle eine wichtige Rolle für die Qualitätskontrolle des Unterrichts spielen, indem sie die Effektivität des Unterrichts aufzeigt [Faessler 2004].

Für die Entwicklung eines Systems für die Leistungskontrolle innerhalb der Lernumgebung bedeutet dies, dass der Unterricht und die Leistungskontrolle aufeinander abgestimmt sein müssen. Dabei stellt sich die Frage nicht nur nach der optimalen Prüfungsform, sondern auch nach der Integration eines Prüfungssystems in eine Lernumgebung. Das System sollte so flexibel sein, dass es sich so effizient wie möglich in unterschiedliche Lernumgebungen integrieren lässt. Wie bei Unterrichtsmethoden ist eine solche Integration auch bei Prüfungsmethoden vor allem eine Frage der Aufbereitung des Inhalts. Die Integration einer Leistungskontrolle in eine Lernumgebung geht deshalb mit der Schaffung einer Organisationsstruktur einher.

Anforderungen an die Leistungskontrolle

Damit die Lernenden ihre im Unterricht gelernten Kompetenzen zeigen können, sollte das zu entwickelnde System ebenfalls kompetenzorientiert sein. Weiter sollte das System effizient und effektiv sein und die Lehrenden bei der Durchführung der Leistungskontrolle sinnvoll unterstützen.

Weitere Anforderungen an die Leistungskontrolle sind:

- Es sollen grosse Klassen geprüft werden können.
- Der Korrekturaufwand soll klein gehalten werden.
- Die Prüfung soll nicht länger als 1 bis 2 Stunden dauern.
- Beurteilungsfehler sollten möglichst ausgeschlossen werden.
- Die Prüfungen müssen rekursfähig sein.

Der Schwerpunkt der Weiterentwicklung unserer Lernumgebungen liegt im Aufbau und der Integration eines Systems für die computergestützte Leistungskontrolle, das oben genannte Anforderungen erfüllt.

11.3.2 Anpassen der Lernumgebungen auf weitere Zielgruppen

1. Einsatz der Lernumgebungen bei Institutionen mit ähnlichen Zielgruppen

Weil die Lernumgebungen durch indirekte Instruktion mit E-Learning weitgehend unabhängig von Ort, Zeit und Lehrperson bearbeitet werden können, sind sie auch ohne grossen Anpassungsaufwand in anderen Institutionen mit ähnlicher Zielgruppe einsetzbar (für Details zum Einsatz im Unterricht siehe Kapitel 9 „Einsatz der computergestützten Lernumgebung im Unterricht“). Verschiedene Hochschulen setzen bereits von uns entwickelte Lernumgebungen oder Teile davon in ihrem Unterricht ein. Für den Einsatz der Lernumgebungen interessieren uns dabei folgende Fragen:

- Welches Wissen brauchen Dozierende anderer Institutionen, um eine der Lernumgebungen in ihrem Unterricht einzusetzen?
- Welche Änderungen müssen an den Lernumgebungen vorgenommen werden, wenn sie unter anderen Bedingungen (Zeit, Altersgruppe, Integration ins Curriculum usw.) zum Einsatz kommen?
- Welche Kombinationen mit Präsenzunterricht (Blend) ergeben sich durch den Einsatz der Lernumgebungen an anderen Institutionen?

Weiterentwicklungs-Beispiele:

Die Zürcher Hochschule für Angewandte Wissenschaften (ZHAW) und die Hochschule für Technik Zürich (HSZ) verwenden die Lernumgebung „Einführung ins Programmieren“, um ihre Studierenden mit sehr heterogenem Vorwissen auf einen einheitlichen Stand zu bringen. Die Lernumgebung „Einsatz von ICT“ soll im Rahmen eines Entwicklungsprojekts an einer Universität in Sambia zum Einsatz kommen.

2. Einsatz der Lernumgebungen bei Institutionen mit anderen Zielgruppen

Mit einigen Einschränkungen sind die Lernmaterialien auch für ganz andere Zielgruppen einsetzbar. Die zu bearbeitenden Probleme sind für eine spezifische Zielgruppe ausgewählt worden, mit dem Ziel, die Anwendung der Informatik-Konzepte in ihrem Anwendungsfeld zu demonstrieren. So lernen beispielsweise Studierende der Naturwissenschaften das Verwalten von Daten mit der Schweizerischen Nährwertdatenbank. Der Realitätsbezug dieses Problems ist zwar für einen weiten Kreis von Studierenden diverser Fachrichtungen interessant, für Zielgruppen wie z.B. die Wirtschaftswissenschaften wäre dieses Beispiel aber zu weit von deren Anwendungsfeldern entfernt.

Eine Möglichkeit, um die Lernumgebung ohne grossen Aufwand trotzdem einer möglichst breiten Zielgruppe zugänglich zu machen, wäre, die Problemaufgabe der Phase *DO* durch eine andere Aufgabe aus dem Anwendungsfeld der Zielgruppe zu ersetzen.

Das Auswechseln der Problemaufgabe der Phase *TRY* durch ein anderes Beispiel ist zwar möglich, aber aufgrund der detaillierten Ausführungen im *E.Tutorial*[®] mit einem grossen Arbeitsaufwand verbunden. Ziel der Weiterentwicklung unserer Lernumgebungen ist der Aufbau eines Pools an Lernmaterialien, die bei einer möglichst breiten Zielgruppe die Aneignung von ICT-Kompetenz und Programmieren ermöglicht.

Weiterentwicklungs-Beispiel:

An der Hochschule für Wirtschaft (HSW) in Luzern werden die Lernmodule 4 und 5 des ICT-Kurses zur Vermittlung des Themas Daten verwalten eingesetzt. Die *E.Tutorials*[®] werden dafür unverändert übernommen, für die anschliessende Problemaufgabe der Phase *DO* wurde hingegen eine neue, auf die Zielgruppe abgestimmte Aufgabe formuliert.

11.3.3 Erweitern bereits bestehender Lernumgebungen durch neue Inhalte

Eine weitere Möglichkeit der Weiterentwicklung der Lernumgebungen ist ein Ausbau des Lehrangebots mit neuen Inhalten. Das Unterrichtsvolumen ist für unsere Zielgruppe auf die zwei Kurse mit je 2-3 ECTS Punkten beschränkt (siehe Abb. 3.9). Um das Angebot trotzdem auszubauen oder besser an die Wünsche einer Zielgruppe anzupas-

sen, können entweder Wahlmöglichkeiten geschaffen oder eine Lerneinheit durch eine andere ersetzt werden.

Weiterentwicklungs-Beispiel:

Der im Rahmen dieser Arbeit entwickelte ICT-Kurs wird an der ETH Zürich für verschiedene Zielgruppen angeboten (siehe Abb. 3.9). Für die Zielgruppe der Naturwissenschaften (Agrar- und Lebensmittelwissenschaften, Biologie, Bewegungswissenschaften und Sport, Chemie und angewandte Biowissenschaften, Erdwissenschaften, Umweltnaturwissenschaften) steht in Lernmodul 3 das Visualisieren mehrdimensionaler Daten auf dem Programm. Als die Lernmaterialien für die Durchführung eines Kurses für die Zielgruppe Bau-, Umwelt und Geomatik zum Einsatz kamen, wurde dieses Lernmodul auf Wunsch des Studiengangs durch ein anderes zum Thema „Matrizenrechnen mit Matlab“ ersetzt.

Es gibt auch Bestrebungen, das Angebot der Informatik für Naturwissenschaftlerinnen und Naturwissenschaftler mit weiteren Kursen auszubauen. Eine mögliche Weiterentwicklung wäre ein Vertiefungskurs zum Thema Datenverwaltung und Datenstrukturen, weil dieses Thema für diese Studienrichtungen von immer grösserer Bedeutung ist.

11.4. Entwicklung neuer computergestützter Lernumgebungen

Die Entwicklung neuer computergestützter Lernumgebungen konzentriert sich auf die Bereiche Informatikunterricht, Fachunterricht und industrielle Weiterbildung. Diesen Bereichen ist allen gemeinsam, dass sie an der Vermittlung von Kompetenzen im Zusammenhang mit dem Einsatz von Computern interessiert sind, obwohl sie sich bei den Zielsetzungen und den Unterrichtsgegenständen zum Teil stark unterscheiden.

11.4.1 Informatikunterricht

Der Informatikunterricht ist dafür prädestiniert, dass er direkt am Computer stattfindet, weil er den Computer selber zum Unterrichtsgegenstand hat. Beim Informatikunterricht steht primär das Vermitteln von Informatik-Konzepten im Zentrum. Die Aneignung von Konzeptwissen durch konkrete Anwendungen eröffnet die Möglichkeit, den Unterricht interdisziplinär zu gestalten und damit besser auf eine spezifische Zielgruppe auszurichten. Der Einsatz neuer computergestützter Lernumgebungen für den Informatikunterricht soll deshalb die Möglichkeit bieten, auf verschiedenen Stufen des Bildungswesens Informatik-Konzepte kompetenzorientiert direkt zusammen mit einer Anwendung zu lehren.

1. Stufe: Universitäre Hochschulen und Fachhochschulen

An Universitäten und Fachhochschulen wird Informatik meist für ganz unterschiedliche Zielgruppen unterrichtet. Neue computergestützte Lernumgebungen zusammen

mit dem 4-Schritte-Modell ermöglichen einen anwendungsorientierten Zugang zu den Informatik-Konzepten, indem die gewählten Beispiele aus dem Fachbereich der Zielgruppe gewählt werden können. Die Möglichkeit, den Informatikunterricht interdisziplinär zu gestalten, hat den zusätzlichen Vorteil, dass die Studierenden dadurch die Informatik als Teil ihrer Fachausbildung sehen (siehe Kapitel 3 „ICT-Kompetenzen für die wissenschaftliche Tätigkeit“).

Weiterentwicklungs-Beispiel:

Die Zürcher Hochschule für Angewandte Wissenschaften (ZHAW) hat eine Lernumgebung zum Unterrichten einer „Computerlizenz für Übersetzer“ nach dem Vorbild unserer Lernumgebungen erstellt. Bei diesem Kurs werden ICT-Kompetenzen unterrichtet, die für die Übersetzertätigkeit und das Übersetzerstudium notwendig sind.

2. Stufe: Mittelschule

In der Mittelschule steht das Vermitteln von Konzepten bei einem höheren Präsenzanteil und weniger Selbststudium der Lernenden im Vordergrund des Unterrichts. Die zu entwickelnden Lernumgebungen können als Lehrmittel den Unterricht zu ausgewählten Themen bei der Arbeit im Computerlabor unterstützen. Die Lerneinheiten sollten hierfür kürzere Sequenzen beinhalten als beim Einsatz an Hochschulen.

Weiterentwicklungs-Beispiele:

Lehrmittel zur Programmier Einführung

Eine computergestützte Lernumgebung für die Programmier Einführung in *Turbo Pascal* wird bereits an einigen Mittelschulen eingesetzt.

Lehrerfortbildung zum Erstellen computergestützter Lernmaterialien

Viele Lehrerinnen und Lehrer an Mittelschulen unterrichten bereits problembasiert und verfügen über eine Vielzahl an Lernmaterialien. Im Rahmen von Lehrerfortbildungen sollen sie zu Autoren zum Erstellen elektronischer Lernmaterialien (*E.Tutorials*[®]) und zum Unterrichten im 4-Schritte-Modell ausgebildet werden. Hierfür bietet ein für diesen Zweck entwickeltes *E.Tutorial*[®] mit dem Titel „Wie schreibe ich funktionierende *E.Tutorials*“ Unterstützung [*ET-HowTo*, <http://et.ethz.ch>].

11.4.2 Fachunterricht

Ein weiterer Bereich zum Erstellen neuer Lernumgebungen ist das Unterrichten anderer Fächer mit dem Einsatz von Computern als fachspezifisches Werkzeug. Bei diesen Lernumgebungen steht die Aneignung wichtiger fachspezifischer Kompetenzen des betroffenen Faches und nicht primär das Erlernen von Informatik-Konzepten im Zentrum. Häufig sind heute verwendete Werkzeuge aber so umfangreich, dass deren pro-

fessioneller Einsatz die Anwendung einer ganzen Reihe von Informatik-Konzepten (z.B. Programmierung, Verwaltung und Strukturierung von Daten usw.) erfordert. Das Potential der Entwicklung solcher Lernumgebungen liegt darin, dass die Bedienung eines fachspezifischen Werkzeugs im Zentrum des Unterrichts steht und durch den Realitätsbezug die Notwendigkeit zur Aneignung ausgewählter Informatik-Konzepte aufgezeigt und direkt in die Lernumgebung eingebaut werden könnte. Durch den Einsatz von Computern bietet sich auch für andere Fachrichtungen die Möglichkeit, bereits im Grundstudium handlungsorientiert Kompetenzen zu unterrichten.

Weiterentwicklungs-Beispiele:

Prädestinierte Fachbereiche für die Entwicklung solcher Lernumgebungen sind Mathematik (z.B. Computeralgebra), Statistik (Statistik-Pakete), Chemie (z.B. Software zur Molekülmodellierung), Geografie (GIS-Software) und viele mehr. Bisher gibt es noch keine umgesetzten Lernumgebungen in diesem Bereich.

11.4.3 Industrie

Das Potential zum Erstellen neuer computergestützter Lernumgebungen im industriellen Umfeld konzentriert sich vor allem auf folgende zwei Bereiche:

- 1. Firmeninterne Weiterbildung in Informatik für Mitarbeitende
- 2. Mitarbeiterschulung für spezialisierte Informatikmittel direkt am Computer

Für grosse Firmen ist zudem die Zeit- und Ortsunabhängigkeit solcher Lernumgebungen interessant, weil dadurch Reisekosten eingespart werden können.

1. Firmeninterne Weiterbildung in Informatik für Mitarbeitende

Für viele Berufe im industriellen Umfeld sind umfassende ICT-Kompetenzen entscheidend geworden. Es wird davon ausgegangen, dass ein ansehnlicher Teil der gesamten IT-Kosten (*TCO, Total Cost of Ownership*) in der Wirtschaft auf mangelhaft geschulte Anwenderinnen und Anwender zurückzuführen ist. Um die grossen Kompetenzunterschiede von Mitarbeiterinnen und Mitarbeitern auszugleichen, bieten viele Firmen interne Weiterbildungen in diesem Bereich an. Für viele Aufgaben reichen Software- und Produkteschulungen nicht mehr aus, es ist das Vermitteln von umfangreichen ICT-Kompetenzen erforderlich (siehe Abschnitt 10.4.1 Informatikunterricht).

2. Mitarbeiterschulung für spezialisierte Informatikmittel direkt am Computer

An vielen Arbeitsplätzen kommen hoch spezialisierte Informatikmittel zum Einsatz. Mit Hilfe neuer Lernumgebungen können die Mitarbeitenden in der Bedienung solcher Informatikmittel direkt am Arbeitsplatz in einem realen Umfeld geschult werden (siehe 4.2 Fachunterricht).

Weiterentwicklungs-Beispiel:

Im Rahmen einer Pilotstudie soll eine Lernumgebung für die SAP-Schulung für Controller eines grossen schweizerischen Unternehmens entwickelt werden.

Anhang

Anhang A

Kursprogramme

A1. ICT-Kurs

Programm des ICT-Grundlagenkurses. Einsatz der Lernmaterialien im Informatikunterricht der ETH Zürich siehe Tab. 1.2 und Abb. 3.9. Details zur ICT-Kompetenz siehe Kapitel 3 „ICT-Kompetenzen für die wissenschaftliche Tätigkeit“.

Modul 1: Rechnernetze und Kommunikation	
Aufgabe	Erstellen und publizieren einer Webseite
Konzepte	WWW, HTML, Übertragungsprotokolle, Grafikformate, File-Struktur, Webserver
Fertigkeiten	Ein Computer in Betrieb nehmen (Tab. 3.3, Punkt 1) Ein Computer mit einem Netzwerk verbinden (Tab. 3.3, Punkt 5) Mit Hilfe des Internets Informationen suchen (Tab. 3.3, Punkt 6) Den Computer zum Kommunizieren einsetzen (Tab. 3.3, Punkt 7)
Werkzeuge	<i>Mozilla Composer, Browser</i>
Situation	Eigene Webseite auf dem Webserver der ETH
Modul 2: Datenmodellierung	
Aufgabe	Modellieren und simulieren von Prozessen
Konzepte	Modell, Simulation, Funktionen, Absoluter- und relativer Bezug, Bedingte Anweisung, Zielwertsuche
Fertigkeiten	Eine Tabellenkalkulation einsetzen um Prozesse zu modellieren (Tab. 3.3, Punkt 8)
Werkzeuge	<i>MS Excel, Open Office</i>
Situation	Simulation Populationsdynamik und Medikamentkonzentration im Blut
Modul 3: Datenvisualisierung	
Aufgabe	Visualisieren mehrdimensionaler Daten
Konzepte	Visualisierungsmethode, Datensatz, Dimensionalität von Daten, Parallelkoordinaten, Punktediagramm-Matrix, Permutations-Matrix, Andrews' Kurven
Fertigkeiten	Mit Hilfe geeigneter Software aussagekräftige Diagramme und Illustrationen erstellen (Tab. 3.3, Punkt 4)
Werkzeuge	<i>Excel, VisuLab</i>
Situation	Mehrdimensionale Datensätze von Irisblüten und Bodenproben

Modul 4 und 5: Datenverwaltung	
Aufgabe	Verwalten von Nährwertdaten
Konzepte	Datenbank, Datenlisten, Attribut, Primärschlüssel, Datentyp, Pivot-Tabellen, Logische Operatoren, Relationenmodell, SQL
Fertigkeiten	Eine Datenbank einsetzen um Information zu verwalten (Tab. 3.3, Punkt 9)
Werkzeuge	<i>Excel, Access, Open Office</i>
Situation	Aktuelle Erdbebendaten und Schweizerische Nährwertdatenbank
Modul 6: Programmierung	
Aufgabe	Automatisieren von Abläufen mit Makros
Konzepte	Programm, Makro, Compiler, Objekt, Algorithmus, Fallunterscheidung, Schleife
Fertigkeiten	Ein vorhandenes Programm nach seinen Bedürfnissen anpassen (Tab. 3.3, Punkt 10)
Werkzeuge	<i>Excel, Visual Basic (VBA)</i>
Situation	Sozioökonomischer Datensatz, aktuelle Pollendaten

A2. Programmierkurs

Programm des Programmierkurses. Einsatz der Lernmaterialien im Informatikunterricht der ETH Zürich siehe Tab. 1.2 und Abb. 3.9. Details zum Programmierkurs ist dem Kapitel 8 zu entnehmen.

Modul 1: Grundlagenmodul I	
Aufgabe	Elektronisches Schloss
Konzepte	Variablen, Bedingungsprüfung, Falsche Eingaben abfangen
Werkzeuge	<i>Notepad</i>
Modul 2: Grundlagenmodul II	
Aufgabe	Berechnung von Standard Papier-Formaten
Konzepte	Schleifen
Werkzeuge	<i>Notepad</i>
Modul 3: Grundlagenmodul III	
Aufgabe	Berechnung und Simulieren des Geburtstags-Paradoxons
Konzepte	Zufallszahlen, Arrays
Werkzeuge	<i>Notepad</i>
Modul 4: Anwendungsmodul I	
Aufgabe	Galton-Board
Konzepte	Simulation
Werkzeuge	<i>Notepad</i> oder anderer frei verfügbarer Programmiereditor
Modul 5: Grundlagenmodul IV	
Aufgabe	Barcode-Programm
Konzepte	Liniengrafik, Objekte
Werkzeuge	<i>Notepad</i> oder anderer frei verfügbarer Programmiereditor
Modul 6: Anwendungsmodul II	
Aufgabe	Applet zur Berechnung des Body-Mass-Indexes
Konzepte	Applets, Fensterobjekte, Ereignisse
Werkzeuge	<i>Eclipse</i>
Modul 7: Anwendungsmodul III	
Aufgabe	Energie-Bilanz-Klimamodell
Konzepte	Umgang mit grossen Programmen, Modell-View-Controller-Schema
Werkzeuge	<i>Eclipse</i>

Anhang B

Programmcode Programmierkurs

B.1 E.Tutorial® 1: Elektronisches Schloss

```
import java.io.*;

public class Safe {

    public static void main (String[] args) {

        int a, b, c;
        a = 0;
        b = 0;
        c = 0;

        try{
            a = Integer.parseInt (args[0]);
            b = Integer.parseInt (args[1]);
            c = Integer.parseInt (args[2]);
            System.out.println ("Es wurden die drei Zahlen "+a+", "+b+" und "+c+" eingegeben.");
        }
        catch (Exception e){System.out.println ("Bitte geben Sie eine Ganzzahl ein.");
        }

        if (a==3) {
            if (b==2) {
                if (c==1) {System.out.println ("offen");}
                else{System.out.println ("geschlossen. Versuche es nochmals.");
                }
            }
            else{System.out.println ("geschlossen. Versuche es nochmals.");
            }
        }
        else{System.out.println ("geschlossen. Versuche es nochmals.");
        }

    } //main
} //class
```

B.2 E.Tutorial® 2: Berechnung von Standard-Papierformaten

```
import java.io.*;
public class Din_for {
    public static void main (String[] args) {
        double x, y, x_old, y_old;
        int i;

        x=1/Math.sqrt (Math.sqrt (2));
        y=1/x;

        x=x*100;
        y=y*100;

        for (i=1; i<=6; i++){

            System.out.println("Format A "+i);
            System.out.println(x + " cm");
            System.out.println(y + " cm");

            x_old=x;
            y_old=y;

            y=x_old;
            x=y_old/2;

        }
    } //main
} //class
```

B.3 E.Tutorial® 3: Geburtstags-Paradoxon

```
import java.io.*;
public class Geburtstag {
    public static void main (String[] args) {

        int anzPers;
        anzPers=Integer.parseInt (args[0]);

        int paar30[]=new int[100];

        for(int l=0; l<100; l++){

            int paar=0;
            int [] tag = new int[anzPers];

            for (int i=0; i<anzPers; i++){
                tag[i]=(int) (Math.random() * 365)+1;
            }

            for (int j=0; j<anzPers; j++){

                for (int k=j+1; k<anzPers; k++){
                    if (tag [j]==tag[k]){paar++;}
                }
            }

            System.out.println("Gefundene Paare "+paar);

            if (paar > 0) {paar30[l]=1;}

        }

        double y=0;

        for(int l=0; l<100; l++){
            y=y+paar30[l];
        }

        System.out.println("Die Wahrscheinlichkeit, dass in einer Gruppe von
        "+anzPers+" Personen mindestens zwei am gleichen Tag Geburtstag haben,
        ist "+y+"%");

    } //main
} //class
```

B.4 E.Tutorial® 4: Galton-Board

```
import java.io.*;
public class Nagelbrett {

    public static void main (String[] args) {

        int entscheidung, zufall, summe, hoehe, anzBehaelter, pos, kugeln;

        hoehe = Integer.parseInt(args[0]);
        kugeln = Integer.parseInt(args[1]);

        anzBehaelter = hoehe+1;
        int [] behaelter = new int[anzBehaelter];

        for(int j=0; j<kugeln; j++){
            summe=0;
            pos=0;

                for (int i=0; i<hoehe; i++){

                    entscheidung = (int)(Math.random() * 99 +1);

                    if (entscheidung < 50){zufall = -1; System.out.println(zufall);
                    }
                    else {zufall = 1; System.out.println(zufall);
                    }

                    summe = summe + zufall;

                } //Schleife 1 Kugel

            System.out.println("Die Summe ist: "+summe);
            pos = (summe + hoehe) / 2;
            System.out.println("Behaelter: "+pos);

            behaelter[pos] = behaelter[pos] + 1;

        } //Schleife viele Kugeln

        System.out.println("In den Behaeltern befinden sich folgende Anzahl Kugeln:");

        for (int i=0; i<anzBehaelter; i++){

            System.out.print(" "+behaeltes[i]);

        }

    } //main
} //class
```

B.5 E.Tutorial® 5: Barcode

```
import java.awt.*;
import java.applet.*;

public class Barcode
extends Frame
{
    public static void main(String[] args) {
        Barcode meinFenster1 = new Barcode();
    }

    public Barcode() {
        addWindowListener(new UnserFensterSchliesser());
        setSize (400, 300);
        setTitle("Einfaches Fenster");
        setVisible(true);
    }

    public void paint (Graphics g){
        int x = 80;

        while (x < 160){
            g.drawLine(x, 40, x, 100);
            int zufall = (int)(Math.random()*3+1);
            x=x+zufall;
        }

        int y1=(int)(Math.random()* 99999);
        g.drawString("76", 80, 120);
        g.drawString(""+y1, 100, 120);
        int y2=(int)(Math.random()* 99999);
        g.drawString(""+y2, 140, 120);
    }
} //class
```

B.6 E.Tutorial® 6: BMI

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class Bmi extends Applet implements ActionListener {
    Label beschriftung1 = new Label("Ihr Körpergewicht in kg:");
    TextField gewicht = new TextField (30);
    Label beschriftung2 = new Label("Ihre Grösse in cm:");
    TextField groesse = new TextField (30);
    Button berechnen = new Button ("Berechnen");
    Label resultat = new Label ("

    public void init() {
        add(beschriftung1);
        add(gewicht);
        add(beschriftung2);
        add(groesse);
        add(berechnen);
        add(resultat);
        berechnen.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {

    if (e.getSource() == berechnen){
        int size, weight;
        double body;
        size = Integer.parseInt(groesse.getText());
        weight = Integer.parseInt(gewicht.getText());

        body = weight/((size/100.0)*(size/100.0));

        if (body<18.5){resultat.setText("Sie haben Untergewicht. BMI:"+body);}
        else if (body>30.0){resultat.setText("Sie haben starkes Übergewicht. BMI:"+body);}
        else if (body>25.0){resultat.setText("Sie haben Übergewicht. BMI:"+body);}
        else {resultat.setText("Sie haben Normalgewicht. BMI:"+body);}

        repaint();
    }
}
}
```

Anhang C

Fragebogen

C.1 Vorumfrage (ICT-Kurs)

Vorumfrage zur Informatik-Veranstaltung

Zunächst bitten wir Sie um folgende Angaben zu Ihrer Person:

Fachrichtung

- Bewegungswissenschaften und Sport
- Pharmazeutische Wissenschaften
- Biologie
- andere:

Geschlecht

- weiblich
- männlich

Sie werden im kommenden Semester computerbasierte Tutorate durcharbeiten und am Schluss jedes bewerten können. Damit wir Ihre Antworten über verschiedene Tutorate verfolgen können, Ihr Feedback aber trotzdem anonym bleibt, bitten wir Sie zur Eingabe des folgenden Kürzels: 2 Buchstaben des Vornamens Ihrer Mutter + 2 Buchstaben des Vornamens Ihres Vaters + Tag Ihres Geburtstags. Beispiel: mahao5 für Martha + Hans + 5. September

Geben Sie hier bitte Ihr Kürzel ein:

Nun bitten wir Sie, die folgenden Fragen zu beantworten:

1. Wie viele Stunden verbringen Sie (im Durchschnitt) wöchentlich am Computer (Schule, Arbeit, Privat)?

- 0 Stunden
- 1 bis 5 Stunden
- 6 bis 10 Stunden
- 10 bis 14 Stunden
- 15 Stunden und mehr

2. Mein generelles Interesse an Informatik würde ich wie folgt bezeichnen:

- sehr gross
- gross
- mittel
- eher klein
- kein Interesse

3. Für mein gewähltes Studienfach halte ich es für notwendig, mich in Informatik weiterzubilden.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

4. Wie hoch schätzen Sie Ihre Motivation ein, um während dem kommenden Semester 3 bis 4 Stunden pro Woche für die Informatik zu investieren?

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

5. Was erwarten Sie vom Informatikunterricht im kommenden Semester?

6. Wie würden Sie Ihre Kenntnisse in folgenden Tätigkeitsgebieten einschätzen?

-ein neuer Computer/Laptop in Betrieb nehmen.

- sehr unsicher
- unsicher
- einigermaßen sicher
- sicher
- sehr sicher

-ein Computer an ein Netzwerk anschliessen.

- sehr unsicher
- unsicher
- einigermaßen sicher
- sicher
- sehr sicher

-mit einer Textverarbeitung (z.B. Word) ein Textdokument erstellen.

- sehr unsicher
- unsicher
- einigermaßen sicher
- sicher
- sehr sicher

-ein Illustrations-/oder Grafikprogramm einsetzen, um eine Illustration zu erstellen.

- sehr unsicher
- unsicher
- einigermaßen sicher
- sicher
- sehr sicher

-mit Hilfe des Internets Informationen beschaffen.

- sehr unsicher
- unsicher
- einigermaßen sicher
- sicher
- sehr sicher

-Information auf einer Webseite auf dem Internet verfügbar machen.

- sehr unsicher
- unsicher
- einigermaßen sicher
- sicher
- sehr sicher

-ein Computer zum Kommunizieren einsetzen.

- sehr unsicher
- unsicher

- einigermaßen sicher
- sicher
- sehr sicher

-in einer Tabellenkalkulation einfache Prozesse modellieren.

- sehr unsicher
- unsicher
- einigermaßen sicher
- sicher
- sehr sicher

-aus einer Reihe von Werten ein aussagekräftiges Diagramm erstellen.

- sehr unsicher
- unsicher
- einigermaßen sicher
- sicher
- sehr sicher

-ein Datenbanksystem einzusetzen, um Information zugänglich zu machen.

- sehr unsicher
- unsicher
- einigermaßen sicher
- sicher
- sehr sicher

-mit Hilfe von Makros Programmabläufe automatisieren.

- sehr unsicher
- unsicher
- einigermaßen sicher
- sicher
- sehr sicher

-ein Computer-Programm einer Programmiersprache den eigenen Bedürfnissen anpassen.

- sehr unsicher
- unsicher
- einigermaßen sicher
- sicher
- sehr sicher

Zum Abschluss bitten wir Sie noch um folgende Angaben:

7. Ich habe für den Kurs ein eigenes Laptop zur Verfügung

- ja
- nein

8. Welches Betriebssystem nutzen Sie auf Ihrem privaten Rechner?

- Windows
- MacOS
- Linux
- Dualboot: Windows und Linux
- Ich habe keinen Computer

9. Ich habe bereits eine eigene Internetseite (Homepage oder Blog).

- ja
- nein

10. Fragen, Anregungen?

C.2 Nachumfrage (ICT-Kurs)

Nachumfrage zur Informatik-Veranstaltung

Dieser Kurs wurde durch ein Filep-Projekt (Fonds zur Finanzierung lehrbezogener Projekte) unterstützt. Die Entwicklung solcher *E.Tutorials*[®] und elektronischer Prüfungen ist sehr zeit- und kostenintensiv. Da wir diese vor allem für Euch Studierende entwickeln, sind wir natürlich sehr an Ihrer Meinung interessiert. Dies soll uns helfen, die Qualität der Veranstaltung stetig zu verbessern. Wir haben deshalb einen kurzen Online-Fragebogen erstellt. Für das Ausfüllen brauchen Sie höchstens 5 Minuten.

Zunächst bitten wir Sie um folgende Angaben zu Ihrer Person:

Fachrichtung

- Bewegungswissenschaften und Sport
- Pharmazeutische Wissenschaften
- Biologie
- andere:

Geschlecht

- weiblich
- männlich

Für die Auswertung der Umfrage bitten wir Sie, Ihr Kürzel (welches Sie auch am Ende jedes *E.Tutorials*[®] angegeben haben) einzutragen. Das Kürzel ist ein Standardverfahren zur Anonymisierung von Umfragedaten.

Zur Erinnerung: 2 Buchstaben des Vornamens Ihrer Mutter + 2 Buchstaben des Vornamens Ihres Vaters + Tag Ihres Geburtstags. Beispiel: maha05 für Martha + Hans + 5.September

Geben Sie hier bitte Ihr Kürzel ein:

Nun bitten wir Sie, die folgenden Fragen zu beantworten:

1. Für mein gewähltes Studienfach halte ich es für notwendig, mich in Informatik weiterzubilden.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

2. Der Inhalt des Kurses war gut auf meine Vorkenntnisse abgestimmt.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

3. Ich konnte bei mir über das Semester hinweg Fortschritte in den Kompetenzen feststellen.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

4. Für diesen Kurs wurde die Lernmethode mit den 4 Phasen *SEE - TRY - DO - EXPLAIN* (sehen - probieren -

tun - erklären) entwickelt. Ich finde dies eine gute Lernmethode.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

5. Die *E.Tutorials*[®] haben dazu beigetragen, dass ich die abzugebenden Testaufgaben selbständig lösen konnte.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

6. Das Durcharbeiten der Module (*E.Tutorials*[®] und Testaufgabe) hat mehr Zeit in Anspruch genommen, als ich mir im Voraus vorgestellt habe.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

7. Die mündliche Abgabe der Übungen bei den Assistierenden hat mir nichts gebracht.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

8. Die Assistierenden kamen gut vorbereitet in die Übungsstunden.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

9. Die Assistierenden wählten eine Sprache, die meinem Level entsprach.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

10. Ich traue mir zu, dass ich das im Rahmen dieser Veranstaltung erarbeitete Wissen in Zukunft im Alltag (wissenschaftlich, beruflich, privat) einsetzen kann.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

11. Ich fand den Kurs insgesamt motivierend gestaltet.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

12. Der USB-Stick ist überflüssig und sollte im kommenden Semester nicht mehr abgegeben werden.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

13. An einem weiteren (auf diesem aufbauenden) Informatik-Kurs würde ich teilnehmen, falls ein solches anrechenbares Angebot bestehen würde.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

14. Rückblickend haben sich meine Erwartungen in Bezug auf die Veranstaltung erfüllt.

- trifft voll zu
- trifft zu
- trifft teilweise zu
- trifft weniger zu
- trifft nicht zu

15. Weitere Kommentare und Anregungen zur Veranstaltung:

Verzeichnisse

Verzeichnis A

Abbildungsverzeichnis

Abb. 1.1: Zielsetzungen dieser Arbeit.....	6
Abb. 1.2: Outputs dieser Arbeit.....	11
Abb. 2.1: Kompetenz.....	19
Abb. 2.2: Organisation von Wissen in Form von Konzepten	23
Abb. 2.3: Verknüpfung verschiedener Wissensarten	24
Abb. 2.4: Metakognitivem Wissen und metakognitiver Kontrolle	25
Abb. 2.5: Unterschiedliche Wege der Aneignung von deklarativem und prozeduralem Wissen.....	28
Abb. 2.6: Unterschied zwischen Kurs- und Fachkompetenz.....	34
Abb. 2.7: Taxonomie-Tabelle nach Anderson	36
Abb. 2.8: Modell für einen kompetenzorientierten Unterricht.....	37
Abb. 3.1: Informatik als Enabling-Science.....	39
Abb. 3.2: Fachunabhängige Kompetenzen beim Aufbau von Wissen in der wissenschaftlichen Tätigkeit.....	40
Abb. 3.3: Kompetenzen beim professionellen Umgang mit digitalen Daten in der wissenschaftlichen Tätigkeit.....	43
Abb. 3.4: Aufbau einer ICT-Kompetenz im Informationsarbeitsplatz	48
Abb. 3.5: Einsatz des Informationsarbeitsplatzes im Unterricht.....	52
Abb. 3.6: Verfügbarkeit von Laptops gemäss Umfragen unter den Studierenden.....	54
Abb. 3.7: Vorumfrage über das Vorwissen der Studierenden im Fach Informatik.....	56
Abb. 3.8: Vorumfrage über die Orte der Wissensaneignung im Fach Informatik.....	57

Abb. 3.9: Aufbau des Informatikunterrichts im Grundstudium einiger Studiengänge der ETH Zürich.....	58
Abb. 4.1: Das Didaktische Dreieck.....	59
Abb. 4.2: Indirekte Instruktion in einer kompetenzorientierten Lernumgebung.....	62
Abb. 4.3: Unterschiede zwischen einem Informationssystem und einem kompetenz-orientierten Lernsystem.....	63
Abb. 4.4: Methoden des System-Engineerings für die Entwicklung einer in der Praxis einsetzbaren Lernumgebung.....	67
Abb. 4.5: Beziehungen zwischen Elementen einer kompetenzorientierten Lernumgebung.....	69
Abb. 4.6: Interaktion zwischen den Lernenden und den Systemstrukturen in einer kompetenzorientierten Lernumgebung.....	76
Abb. 5.1: Kompetenzaneignung durch PBL.....	84
Abb. 5.2: Beziehung verschiedener Grössen beim Aufbau einer Kompetenz durch eine Problemaufgabe beim PBL.....	86
Abb. 5.3: Typen von Problemaufgaben beim PBL.....	87
Abb. 5.4: Aufgabengrösse, Komplexität und Wirkungsgrad von Problemaufgaben beim PBL.....	88
Abb. 5.5: Optimieren der Output-Kompetenz durch den kombinierter Einsatz zweier Aufgabentypen.....	89
Abb. 5.6: Abstimmen der Kompetenzlevels einer Reihe von Problemaufgaben zu einer Kurs-Kompetenz.....	90
Abb. 5.7: Virtuoser Kreislauf und Teufelskreis beim PBL.....	93
Abb. 5.8: Problembasierte Instruktion und Anwendung als effektive und motivierende Form des PBL.....	99
Abb. 6.1: Anleitung als Mittel zur Strukturierung bei der Vermittlung von Informationen für unterschiedliche Wissensarten.....	102
Abb. 6.2: Ablauf der Instruktion durch die Vermittlung von Konzepten und Fertigkeiten sowie deren praktische Anwendung.....	104
Abb. 6.3: Konstruktiver Aufbau von Begriffen in einer handlungs- und problemorientierten Instruktion.....	105

Abb. 6.4: Zusammenspiel zwischen Anleitung, Handlungsschritt und Überprüfung zur Lösung eines Problems einer selbstgesteuerten, problembasierten Instruktion.....	107
Abb. 6.5: Aufbau und Selbstüberprüfung einer (Teil-)Kompetenz einer problembasierten Instruktion mittels Anleitung.....	108
Abb. 6.6: Motivationsmodell beim PBL mit selbstgesteuerter, problembasierter Instruktion.....	111
Abb. 6.7: <i>E.Tutorial</i> [®] , eine Applikation für die elektronische, problembasierte Instruktion.....	113
Abb. 7.1: Gliederung des Aufbaus von Kompetenzen in Phasen von unterschiedlichen Zeiträumen	119
Abb. 7.2: Das 4-Schritte-Modell: In 4 Schritten zu aktivem und flexiblem Wissen unterschiedlicher Dimensionen	121
Abb. 7.3: Zeitlicher Ablauf des 4-Schritte-Modells und eingesetzte Medien.....	123
Abb. 7.4: Einsatz von Medien während der Instruktionsphase <i>TRY</i>	125
Abb. 7.5: Umfrage nach dem Einsatz zweier <i>E.Tutorials</i> [®] mit und ohne Video-Instruktionen im Überprüfungsfenster	126
Abb. 7.6: Umfrage über die Anordnung der Fenster im <i>E.Tutorial</i> [®]	128
Abb. 7.7: Einsatz von Hyperlinks im <i>E.Tutorial</i> [®]	129
Abb. 7.8: Umfrage über die Nutzung der Überprüfungsmöglichkeit.....	130
Abb. 7.9: Umfrage nach einem <i>E.Tutorial</i> [®] mit und ohne Verlinkung zu Theorie und Glossar	131
Abb. 7.10: Ablauf des Anwendungsebenen-Modells.....	133
Abb. 7.11: Evaluation des didaktischen Modells.....	134
Abb. 7.12: Didaktische Modelle für den kompetenzorientierten Unterricht.....	135
Abb. 8.1: Entwicklung einer kompetenz- und problemorientierten Lernumgebung.....	137
Abb. 8.2: Schematische Darstellung der Abläufe der Entwicklungsprozesse einer kompetenz- und problemorientierten Lernumgebung.....	142
Abb. 8.3: Umfrageresultate der Vorlesung „Anwendungsnahes Programmieren“	146
Abb. 8.4: Beispiel einer Problemstellung für ein erstes (Teil-)Kompetenzlevel: das Galton Board	150

Abb. 8.5: Beispiel einer Problemstellung für ein zweites (Teil-)Kompetenzlevel: Applet zur Berechnung des Body-Mass-Index (BMI).....	151
Abb. 8.6: Konsole-Output der Aufgabenstellung 3 zur Anwendung der Konzepte „Zufallszahlen“ und „Arrays“	154
Abb. 8.7: Empfohlene Anordnung der Informatikmittel für die Autorenarbeit eines <i>E.Tutorials</i> ®	157
Abb. 8.8: Problemaufgabe 1 zur Einführung des Konzepts „Zufallszahlen“	158
Abb. 8.9: Problemstellung 2 zur Einführung des Konzepts „Array“	159
Abb. 8.10: Einbau von Konzepten in die handlungsbasierte Anleitung.....	161
Abb. 8.11: Strukturierung der Lektion im Instruktionsfenster des <i>E.Tutorials</i> ® zur Einführung des Konzepts „Zufallszahlen“	164
Abb. 8.12: Strukturierung zweier Lektionen im Instruktionsfenster des <i>E.Tutorials</i> ® zur Einführung des Konzepts „Arrays“	165
Abb. 8.13: Datenerhebung für Feedbacks und die Prozessanalyse	167
Abb. 8.14: A: Anwendungsorientierte Prüfungsfrage zum Konzept „Array“	169
Abb. 8.15: Prozessanalyse des Programmierkurses	172
Abb. 9.1: Effizienter und individualisierter Ablauf des Unterrichts im 4-Schritte- Modell	178
Abb. 9.2: Ablauf der verschiedenen Aufgaben zur Durchführung eines Kurses im 4- Schritte-Modell während eines Semesters.....	184
Abb. 9.3: Zusammenstellen eines Kurses aus einem Lernmodule Pool	187
Abb. 9.4: Umfrage über die Verwendung von USB-Memory-Sticks als Verteil- und Speichermedium der Lernmaterialien	188
Abb. 9.5: Elektronisches Verwalten von Leistungsnachweisen grosser Klassen mit einer Datenbank	191
Abb. 10.1: Datenerhebungen während der Durchführung eines Kurses.....	199
Abb. 10.2: Ablauf einer Prozessanalyse	200
Abb. 10.3: Anteil Prüfungsfragen eines bestimmten kognitiven Levels	202
Abb. 10.4: Beispiel einer problembasierten MC-Prüfungsfrage.....	203

Abbildungsverzeichnis	255
Abb. 10.5: Entwicklung von Unterricht und Leistungskontrolle eines Kurses von einem traditionellen zu einem Kurs im 4-Schritte-Modell.....	205
Abb. 10.6: Verteilung von korrekt beantworteten K3/4-Fragen in den Prüfungen von 2002, 2003 und 2004.....	207
Abb. 10.7: Abfolge von Problemaufgaben des Programmierkurses 2003 und 2005.....	212
Abb. 10.8: Prozessanalyse des Programmierkurses 2005.....	214
Abb. 10.9: Prozessanalyse des ICT-Kurses 2005.....	214
Abb. 10.10: Prozessanalysen des ICT-Kurses 2005 bis 2007.....	219
Abb. 11.1: Unterschiedliche Einsatzebenen und Zielsetzungen des Computereinsatzes im Unterricht.....	224

Verzeichnis B

Tabellenverzeichnis

Tab. 1.1: Taxonomie zur Klassierung der Interaktivität von Lernsoftware	9
Tab. 1.2: Informatik-Einführungskurse der Natur- und Bauingenieurwissenschaften an der ETH Zürich	13
Tab. 2.1: Verschiedene Arten von Wissen	23
Tab. 2.2: Die unterschiedliche Verarbeitung von deklarativer und prozeduraler Information im Gehirn	27
Tab. 2.3: Prozessmerkmale des Konstruktivistischen Lernens	31
Tab. 2.4: Elemente eines transferfreundlichen Unterrichts	33
Tab. 2.5: Blooms kognitive Prozess-Dimensionen nach Anderson	35
Tab. 3.1: Komponenten der Information-Literacy	42
Tab. 3.2: Konzepte der Computer-Fluency nach Snyder	45
Tab. 3.3: Fertigkeiten der Computer-Fluency nach Snyder	45
Tab. 3.4: Fähigkeiten der Computer-Fluency nach Snyder	46
Tab. 3.5: Schrittweise Verknüpfung von Wissensdimensionen zum Aufbau einer Computer-Fluency beim Lösen eines realen Problems	47
Tab. 3.6: Durchgeführte Vorumfrage	55
Tab. 4.1: Wichtige motivationale Faktoren einer kompetenzorientierten Lernumgebung	72
Tab. 4.2: Dimensionen der Usability	74
Tab. 4.3: Elemente des Cognitive Apprenticeship	77
Tab. 4.4: Aktivitäten von Lehrenden und Lernenden während verschiedener Phasen des Cognitive Apprenticeship	77
Tab. 5.1: Grössen einer Problemaufgabe beim kompetenzorientierten PBL	86

Tab. 5.2: Motivationale Grössen beim PBL.....	94
Tab. 5.3: Formen der Arbeitszufriedenheit	95
Tab. 5.4: Dimensionen der Kontrollüberzeugung	96
Tab. 5.5: Grössen zur Erfassung der Implementierungsqualität einer Problemaufgabe beim PBL.....	98
Tab. 6.1: Art der Motivierung beim PBL mit problembasierter Instruktion	111
Tab. 7.1: Phasen beim Aufbau einer Kompetenz	119
Tab. 7.2: Zwei didaktische Modelle zur Organisation des Ebenenwechsels	120
Tab. 7.3: Die 4 Phasen des 4-Schritte-Modells.....	121
Tab. 7.4: Im 4-Schritte-Modell eingesetzte Medien	124
Tab. 7.5: Medien und Funktion der drei Fenster des <i>E.Tutorials</i> [®]	128
Tab. 7.6: Anwendungslevels A und B des Anwendungsebenen-Modells	133
Tab. 8.1: Liste der Entwicklungsprozesse einer kompetenz- und problemorientierten Lernumgebung.....	141
Tab. 8.2: Übersicht über die eingesetzten Hilfsmittel	143
Tab. 8.3: Zeitlicher Verlauf der Entwicklung des Kurses „Anwendungsnahe Programmieren“ von 2002 bis 2006	144
Tab. 8.4: Einige Kriterien für die Auswahl von Informatikmitteln	148
Tab. 8.5: Im Kurs „Anwendungsnahe Programmieren“ eingesetzte Informatikmittel	149
Tab. 8.6: Aufbau einer Problemserie nach dem Anwendungsebenen-Modell zur Lösung der Galton Board-Aufgabe.....	153
Tab. 8.7: Problembasierte Einführung der Konzepte „Zufallszahlen“ und „Arrays“ zusammen mit Fertigkeiten im <i>E.Tutorial</i> [®]	158
Tab. 8.8: Handlungsschritte zur Generierung von „Zufallszahlen“	159
Tab. 8.9: Handlungsschritte zur Generierung von „Arrays“	160
Tab. 8.10: Struktur des <i>E.Tutorials</i> [®]	162
Tab. 8.11: Aufbau eines <i>E.Tutorials</i> [®] zum Thema „Zufallszahlen“ und“ Arrays“ des Kurses „Anwendungsnahe Programmieren“.....	163

Tabellenverzeichnis	259
Tab. 8.12: Feedback-Informationen zur Qualitätsverbesserung von Lerneinheiten.....	166
Tab. 8.13: Durchschnittliche Anzahl abgegebener Feedbacks via Online-Formular am Ende eines <i>E.Tutorials</i> ®	167
Tab. 8.14: Fragen der Prozessanalyse	170
Tab. 9.1: Durchzuführende Aufgaben für den Unterricht im 4-Schritte-Modell.....	181
Tab. 9.2: Verantwortungen und Aufgaben der am Unterricht im 4-Schritte-Modell beteiligten Personen	182
Tab. 9.3: Eingesetzte Ablaufstrukturen für den Unterricht im 4-Schritte-Modell in den von uns durchgeführten Kursen an der ETH Zürich.....	183
Tab. 9.4: Zusammenfassung des Ablaufs eines Kurses im 4-Schritte-Modell.....	185
Tab. 10.1: In dieser Arbeit durchgeführte Evaluationen	198
Tab. 10.2: Umstrukturierung eines traditionell gehaltenen Kurses zu einem problembasierten Blended Learning im 4-Schritte-Modell 2001 bis 2005	204

Verzeichnis C

Literaturverzeichnis

- Aebli, H. (1983). *Zwölf Grundformen des Lehrens eine allgemeine Didaktik auf psychologischer Grundlage*. Stuttgart: Klett-Cotta.
- ALA, American Library Association (1989). *Presidential Committee on Information Literacy: Final report*. Chicago: ALA.
- Alexander, P., Schallert, D., and Hare, V. (1991). *Coming to terms: How researchers in learning and literacy talk about knowledge*. *Review of Educational Research*, 315-343.
- Anderson, L. W., Krathwohl, D. R., and Bloom, B. S. (2001). *A taxonomy for learning, teaching, and assessing : a revision of Bloom's taxonomy of educational objectives (Abridged ed.)*. New York: Longman.
- Atkinson, J. W. (1957). *Motivational determinants of risktaking behavior*. *Psychological Review*, 64, 359-372.
- Bandura, A. (1997). *Self-efficacy the exercise of control*. New York: Freeman.
- Bauer-Messmer, B., Faessler, L. und Wyss, M. (2005). *Einführungskurse ins Programmieren - eine didaktische Herausforderung Entwicklung, Einsatz und erste Erfahrungen mit E.Tutorials im Programmierunterricht*. Zürich: ETH Zürich, Department Informatik, Institut für Computational Science.
- Berner, H. (1999). *Didaktische Kompetenz Zugänge zu einer theoriegestützten bildungsorientierten Planung und Reflexion des Unterrichts*. Bern: Haupt.
- Bloom, B. S., Engelhart, M. D., Furst, E.J., Hill, W.H., and Krathwohl, D.R. (1956). *Taxonomy of educational objectives: Handbook I: Cognitive domain*. In: New York: David McKay.
- Bosia, L. (2005). *K3+ basierte, elektronisch gestützte Leistungskontrolle*. Zwischenbericht für das Filep-Projekt Nr. 206. Zürich: ETH Zürich.
- Bruggemann, A. (1974). *Zur Unterscheidung verschiedener Formen von "Arbeitszufriedenheit"*. *Arbeit und Leistung*, 28, 281-284.

- Butterworth, B. (1999). *What counts how every brain is hardwired for math*. New York: Free Press.
- Cohen, G. (1989). *Memory in the real world*. Hove etc.: Erlbaum.
- Collins, A., and Brown, J. (1988). *The Computer as a Tool for Learning through Reflection*. In: H. Mandl (Ed.), *Learning issues for intelligent tutoring systems* (pp. XIX, 307). New York etc.: Springer.
- Collins, A., Brown, J., and Newman, S. (1989). *Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics*. In: *Knowing, learning, and instruction: Essays in the honour of Robert Glaser* (pp. 453-494). Hillsdale, NJ: Erlbaum.
- Collins, A., Brown, J. S., and Holum, A. (1991). *Cognitive apprenticeship: Making thinking visible*. *American Educator*, 15(3), 6 - 11.
- Dahinden, M. (2006). *Installation und Konfiguration des ET-Portals*. Interner Bericht. Zürich: ETH Zürich, Institut für Computational Science.
- Dewey, J. (1991). *How we think*. Amherst, New York: Prometheus Books.
- Educational Testing Services ETS (2005). *Beyond Technical Competence: Literacy in Information and Communication Technology*. *Educational Technology Magazine*.
- Erpenbeck, J. (2003). *Handbuch Kompetenzmessung erkennen, verstehen und bewerten von Kompetenzen in der betrieblichen, pädagogischen und psychologischen Praxis*. Stuttgart: Schäffer-Poeschel.
- Europäische Kommission (2004). *Europäisches System zur Übertragung und Akkumulation von Studienleistungen (ECTS) - Kernpunkte*.
- Faessler, L., Hinterberger, H., and Bauer-Messmer, B. (2004). *The Application Guide: Each Student His Own Tutor*. Paper presented at the Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004, Lugano, Switzerland.
- Faessler, L., Hinterberger, H., Bosia, L., and Dahinden, M. (2005, June 27, 2005). *Assessment as an instrument to evaluate quality of instruction*. Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA) 2005, Montreal, Canada.
- Faessler, L. und Wyss, M. (2005). *Erstellen von anwendungsorientierten MC-Prüfungsfragen*. Eine kurze Einführung. Zürich: Institut für Computational Science, ETH Zürich.
- Faessler, L., Hinterberger, H., Dahinden, M., and Wyss, M. (2006, October 2006). *Evaluating student motivation in constructivistic, problem-based introductory computer science courses*. Paper presented at the World Conference on E-Learning in Corpo-

- rate, Government, Healthcare, and Higher Education (ELEARN) Honolulu, Hawaii, USA.
- Fisseni, H.-J. (2004). *Lehrbuch der psychologischen Diagnostik mit Hinweisen zur Intervention* (3. überarb. und erw. Aufl. ed.). Göttingen: Hogrefe.
- Flavell, J. (1979). *Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry*. *American Psychologist*, 34, 906-911.
- Gagné, R. M., and Briggs, L. J. (1974). *Principles of instructional design*. New York a.o.: Holt Rinehart & Winston.
- Gräsel, C., Prenzel, M. und Mandl, H. (1993). *Konstruktionsprozesse beim Bearbeiten eines fallbasierten Computerlernprogramms*. In: (pp. 55-66): Tarnai C.
- Gruber, H., Mandl, H. und Renkl, A. (2000). *Was lernen wir in Schule und Hochschule: Träges Wissen?* In: H. M. J. Gerstenmeier (Ed.), *Die Kluft zwischen Wissen und Handeln: Empirische und theoretische Lösungsansätze* (pp. S. 139-156). Göttingen: Hogrefe.
- Guilbert, J. J. (1977). *Educational handbook for health personnel*. Geneva: World Health Organization.
- Hey, J. (2004). *The Data, Information, Knowledge, Wisdom Chain: The Metaphorical link*. Intragovernmental Oceanographic Commission.
- Hinterberger, H., Faessler, L., and Bauer-Messmer, B. (2004). *From Hybrid Courses to Blended Learning: A Case Study Paper presented at the Proceedings of the 6th International Conference on New Educational Environments (ICNEE), University of Neuchatel, Switzerland*.
- Hinterberger, H. (2007). *E-Learning: Make it Simple but not Simpler*. *African Technology Development Forum (ATDF) Journal*, 4(2), 11-18.
- Hoff, E.-H. (1992). *Arbeit, Freizeit und Persönlichkeit wissenschaftliche und alltägliche Vorstellungsmuster* (2. überarbeitete und aktualisierte Aufl. ed.). Heidelberg: Asanger.
- ISO 9241-11. (1998). *Anforderungen an die Gebrauchstauglichkeit - Leitsätze* (Vol. 9241-11): DIN.
- ISO 9241. (2006). *Ergonomie der Mensch-System-Interaktion* (Vol. 9241): DIN.
- Kaiser, A. und Kaiser, R. (1999). *Metakognition Denken und Problemlösen optimieren*. Neuwied etc.: Luchterhand.
- Konrad, K. und Traub, S. (1999). *Selbstgesteuertes Lernen in Theorie und Praxis*. München: Oldenbourg.

- Kuhlen, R. (1991). *Hypertext ein nicht-lineares Medium zwischen Buch und Wissensbank*. Berlin etc.: Springer-Verlag.
- Mandl, H., Gruber, H., and Renkl, A. (1997). *Situiertes Lernen in multimedialen Lernumgebungen*. In: L. Issing, Klimsa, P. (Ed.), *Information und Lernen mit Multimedia*, 2. überarbeitete Auflage (pp. S. 167-178). Basel, Weinheim: Beltz Psychologie-Verlags-Union.
- Mandl, H. G. und Renjk, A. (1993). *Das träge Wissen*. *Psychologie heute*, 20, 64-69.
- Marini, A. G., and Genreux, R. (1995). *The challenge of teaching for transfer*. In: McKeough, A., Lupart, J.L. und Marini, A. (pp. 1-19).
- Mayer, R. E., and Wittrock, M.C. (1996). *Problem-solving transfer*. In: D. C. B. R.C.Calfee (Ed.), *Handbook of educational psychology* (pp. 47-62). NY: Macmillan.
- Merrill, M. D. (1994). *Instructional design theory*. NY.
- Meyer, B. (1993). *Towards an object-oriented curriculum*. *Journal of Object-Oriented Programming*, 76-81.
- Mietzel, G. (2003). *Pädagogische Psychologie des Lernens und Lehrens* (7. korr. Aufl. ed.). Göttingen: Hogrefe Verl. für Psychologie.
- Morkes, J., and Nielsen, J. (1997). *Concise, SCANNABLE, and Objective: How to Write for the Web*. <http://www.useit.com/papers/webwriting/writing.html>
- National Research Council (U.S.) Committee on Information Technology Literacy. (1999). *Being fluent with information technology*. Washington, DC: National Academy Press.
- Nielsen, J. und Lagrange, K. (1996). *Multimedia, Hypertext und Internet Grundlagen und Praxis des elektronischen Publizierens*. Braunschweig etc.: Vieweg.
- Polanyi, M. (1985). *Implizites Wissen*. Frankfurt am Main: Suhrkamp.
- Prpic, J. K., and Handgraft, R. G. (1999). *What is problem-based learning?*, <http://cleo.eng.monash.edu.au/teaching/learning/strategy/whatispbl.html>
- Reinmann-Rothmeier, G. M. und Mandl, H. (1997). *An- und Herausforderungen an die Erwachsenenbildung*. In: *Psychologie der Erwachsenenbildung* (pp. 366-371). Bern: Hogrefe.
- Renkl, A. (1996). *Träges Wissen: Wenn Erlerntes nicht genutzt wird*. *Psychologische Rundschau*, 47, 78-92.
- Reusser, K. (2005). *Problemorientiertes Lernen - Tiefenstruktur, Gestaltungsformen, Wirkung*. In: *Beiträge zur Lehrerbildung* (Vol. 23, pp. 159-181).

- Rheinberg, F. (2000). *Motivation* (3. überarb. und erw. Aufl. ed.). Stuttgart: Kohlhammer.
- Riedl, A. S., A. (2006). *Handlungsorientiertes Lernen. Aktuelle Entwicklungen aus der Lehr-Lern-Forschung und deren Anwendung im Unterricht*. München: Lehrstuhl für Pädagogik, Technische Universität München.
- Rosch, E. H., and Llojd, B.B. (1978). *Cognition and categorization*. Hillsdale, NJ: Erlbaum.
- Roth, G. (2003). *Fühlen, Denken, Handeln wie das Gehirn unser Verhalten*. Frankfurt am Main: Suhrkamp.
- Rotter, J. B. (1966). *Generalized expectancies for internal versus external control of reinforcement*. Psychological Monographs, 80(1), 1-28.
- Rotter, J. B. (1990). *Internal versus external locus of control of reinforcement*. American Psychologist, 45, 489-493.
- Schulmeister, R. (1996). *Grundlagen hypermedialer Lernsysteme Theorie - Didaktik - Design*. Bonn etc.: Addison-Wesley.
- Schulmeister, R. (2002). *Taxonomie der Interaktivität von Multimedia - ein Beitrag zur aktuellen Metadaten-Diskussion*. it+ti, 4, 193-199.
- Snyder, L. (2004). *Fluency with information technology: skills, concepts and capabilities*. Boston: Pearson Addison Wesley.
- Spitzer, M. (1996). *Geist im Netz Modelle für Lernen, Denken und Handeln*. Heidelberg etc.: Spektrum.
- Spitzer, M. (2002). *Lernen Gehirnforschung und die Schule des Lebens*. Heidelberg: Spektrum Akademischer Verlag.
- Steiner, E. (2004). *Erkenntnisentwicklung durch Arbeiten am Fall. Ein Beitrag zur Theorie fallbezogenen Lehrens und Lernens in Professionsausbildungen mit besonderer Berücksichtigung des Semiotischen Pragmatismus von Charles Sanders Peirce*. Universität Zürich, Zürich.
- Strittmatter, P., Niegemann und Helmut M. (2000). *Lehren und Lernen mit Medien eine Einführung*. Darmstadt: Wissenschaftliche Buchgesellschaft.
- Wahl, D. (2006). *Lernumgebungen erfolgreich gestalten vom trägen Wissen zum kompetenten Handeln* (2. erw. Aufl. ed.). Bad Heilbrunn: Klinkhardt.
- Wilson, K. G. und Daviss, B. (1994). *Redesigning education*. New York: Holt.

Lebenslauf

Personalien

Name: Fässler
Vorname: Lukas Emanuel
Geburtsdatum: 07. Mai 1972
Geburtsort: Luzern LU
Heimatort: Appenzell AI

Ausbildung

1979 - 1988 Primar- und Sekundarschule, Luzern
1988 - 1993 Lehrerseminar, Luzern, Abschluss: Primarlehrdiplom
1994 - 1999 Studium der Biologie an der Universität Bern, Abschluss: Dipl. phil. nat.
1999 Diplomarbeit in Neurobiologie bei Prof. Dr. D. Felix
Angiotensin II als Neurotransmitter im paraventriculären Nucleus des Hypothalamus

Berufliche Tätigkeiten

1999 Unterrichtsassistenz am Zoologischen Institut, Abteilung Neurobiologie, Universität Bern
2000 Assistent am Institut für Behavioral Neuroscience, ETH Zürich
2000-2001 Projektmitarbeiter in der Dokumentationsabteilung der Medizinaltechnikfirma Biotronik im Auftrag der Firma Sinteg, Weisslingen ZH
2001-2007 Assistent am Institut für Computational Science, ETH Zürich
Projektmitarbeiter der Filep-Projekte „Information Processing for Domain Scientists“ und „Computergestützter Programmierunterricht für Informatikerne Studiengänge“ am Institut für Computational Science, ETH Zürich