

# Lernzielgraphen und Lernzielerfolgsanalyse

Markus Steinert<sup>1</sup>

Didaktik der Informatik  
TU-München  
Boltzmannstr. 3  
85748 Garching  
markus.steinert@in.tum.de

**Abstract:** Die vorliegende Arbeit beschreibt eine Methode, wie eine konzeptuell-didaktische Gliederung der Informatik empirisch ermittelt werden kann. Die durch diese Methode gewonnene Klassifizierung unterscheidet sich von bekannten derartigen Strukturierungen dadurch, dass Lernziele als zentrale Elemente verwendet werden. Gegenüber abstrakten Taxonomien hat dies den Vorteil größerer Praxisnähe. Sofern die Lernziele darüber hinaus mit konkreten Aufgabenstellungen verknüpft sind und für diese Aufgaben Leistungserhebungen vorliegen, lässt sich analysieren, inwieweit die Lernziele von den Lernenden erreicht wurden. Da umfangreiche Leistungserhebungen bisher nur aus dem Hochschulbereich vorliegen, werden die Erstellung des Lernzielgraphen und die Lernzielerfolgsanalyse im Hochschulumfeld erläutert. Die Übertragung auf den schulischen Bereich ist jedoch bei entsprechender Datenlage problemlos möglich.

## 1 Einführung

Nach wie vor ist eine konzeptuell-didaktische Klassifizierung der Inhalte der Informatik ein Gegenstand der didaktischen Forschung. Mit zu den bedeutendsten Publikationen in diesem Bereich zählen die Arbeiten von A. Schwill [Sc93], sowie von P. Denning [De99] [De03]. Während die Taxonomie von A. Schwill auf den fundamentalen Ideen von J.S. Bruner [Br60] basiert, definiert P. Denning ohne Angabe einer detaillierten Methode sogenannte „Great Principles“ als bestimmend für die Informatik. Dennoch sind die Klassifikationen von A. Schwill und insbesondere die neueste Klassifikation von P. Denning rein konzeptuell sehr ähnlich. Obwohl die Notwendigkeit einer konzeptuellen Klassifizierung der Informatik außer Frage steht, werden vor allem die fundamentalen Ideen von einigen Autoren (z.B. [Ba98]) durchaus kritisch gesehen. Insbesondere die Frage, ob alle fundamentalen Ideen wirklich fundamental im Sinne von J.S. Bruner sind, ist durchaus nicht in allen Fällen empirisch gesichert (siehe dazu etwa [Sc06a], [Sc06b]).

Darüber hinaus hat sich auch gezeigt [Sc06a] [Sc06b], dass die Verwendung von Konzepten ein prinzipielles Problem nach sich zieht: Die Konzepte, seien es die fundamenta-

---

<sup>1</sup> geb. Schneider

len Ideen oder die „Great Principles“, sind Abstraktionen verschiedener konkreter Inhalte. Welche Inhalte das aber konkret sind, bleibt in vielen Fällen offen. Durch diese fehlende Spezifizierung verlieren die Konzepte, trotz ihres intuitiv „fundamentalen“ Charakters, vieles von ihrer Wertigkeit, da es in der Praxis oft schwer ist einen Inhalt einem bestimmten Konzept zu zuordnen.

Man könnte vermuten, dass von der Seite der Hochschulinformatik eine Klärung der Frage, was denn die zentralen Inhalte der Informatik seien, gegeben werden könnte. Insbesondere mit der Einführung von Bachelor / Master-Studiengängen müssten die Vorlesungen zur Einführung in die Informatik vergleichbare zentrale Konzepte vermitteln und diese Konzepte könnten dann die Grundlage einer detaillierten Modellierung eines konzeptuellen Netzwerkes bilden. Bereits ein kurzer Blick auf die Webseiten namhafter deutscher Universitäten [Tu07] [Tb07] [Ra07] zeigt, dass bereits in den Einführungsvorlesungen die Ansichten über die wesentlichen Inhalte der Informatik weit auseinandergehen.

Angesichts der geschilderten Probleme scheint ein praxisnäherer Ansatz zur inhaltlichen Strukturierung der Informatik empfehlenswert. Deshalb wird in der vorliegenden Arbeit ein lernzielorientierter Ansatz gewählt. Lernziele sind konkreter als Konzepte und lassen sich unter Verwendung geeigneter psychologischer Taxonomien nicht nur inhaltlich, sondern auch lernpsychologisch klassifizieren. Konkrete Problemstellungen, wie etwa Klausuraufgaben, werden auf die zugrundeliegenden Lernziele analysiert. Diese Lernziele werden anschließend in einem Lernzielgraphen unter Verwendung inhaltlicher und didaktischer Relationen vernetzt. Da für Klausuraufgaben im Hochschulsumfeld umfangreiches Datenmaterial in Form von Prüfungsleistungen vorliegt, lassen sich durch geschickte Kombination der Leistungen in verschiedenen Aufgaben Rückschlüsse auf die Leistungsfähigkeit der Lernenden in einzelnen Lernzielen ziehen

## **2 Lernzielgraph und revidierte Taxonomie von Bloom**

Die revidierte Bloomsche Lernzieltaxonomie [AK01] erweitert die bekannte Bloomsche Klassifikation [B156] kognitiver Lernziele um eine Wissensdimension. Neben den kognitiven Dimensionen (Wissen, Verständnis, Anwendung, Analyse, Synthese, Beurteilung) werden orthogonal dazu folgende Wissensstufen eingeführt: Faktenwissen, konzeptuelles Wissen, prozedurales Wissen und metakognitives Wissen. Letzteres beschreibt domänenübergreifendes Transferwissen und ist für die vorliegende Arbeit nicht relevant.

### **2.1 Die revidierte Lernzieltaxonomie und der Lernzielgraph im Beispiel**

Das folgende Beispiel dient der Verdeutlichung der revidierten Bloomschen Taxonomie. Außerdem wird daran die Methode der Konstruktion des Lernzielgraphen vorgestellt. Im Modulplan der RWTH-Aachen [Ra07] ist für die Veranstaltung „Praktische Informatik“ unter anderem folgendes „Lernziel“ angegeben:

„Beherrschung der wesentlichen Konzepte imperativer und objektorientierter Sprachen, sowie wichtiger Programmier Techniken in diesen Sprachen“

Hinter dieser Formulierung verbergen sich insgesamt vier Lernziele:

1. Wesentliche Konzepte imperativer Sprachen beherrschen (C/3)
2. Wesentliche Konzepte objektorientierter Sprachen beherrschen (C/3)
3. Wichtige Programmiertechniken imperativer Sprachen beherrschen (P/3)
4. Wichtige Programmiertechniken objektorientierter Sprachen beherrschen (P/3)

In dieser Form lassen sich die Lernziele in die revidierte Bloomsche Taxonomie einordnen. Wie bei Anderson & Krathwol [AK01] beschrieben, erfolgt die Zuordnung eines Lernziels sehr systematisch: Aus den Verben resultiert die Zuordnung zu der kognitiven Dimension, aus den Substantiven die Zuordnung zu der Wissensdimension. Da die oben genannten Lernziele 1 – 4 verlangen, dass die entsprechenden Inhalte „beherrscht“ werden, sind alle in die dritte kognitive Dimension einzuordnen (Anwendung). Bei den Lernzielen 1 und 2 handelt es sich offensichtlich um konzeptuelles Wissen. Im Gegensatz dazu handelt es sich bei den Lernzielen 3 und 4 um die Beherrschung von „Techniken“; ein eindeutiger Hinweis, dass man es mit prozeduralem Wissen zu tun hat. Die

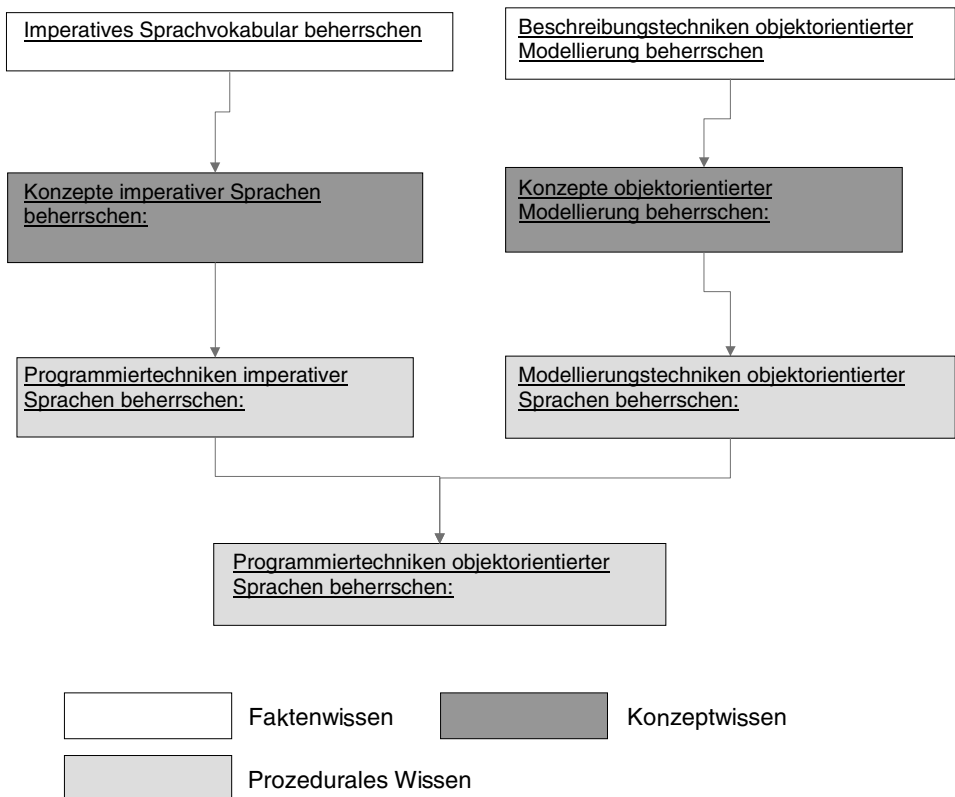


Abbildung 1: Lernzielgraph

soeben erläuterte Zuordnung ist bei den Lernzielen 1 – 4 am rechten Rand angegeben. Die Zeichen F, C und P symbolisieren Faktenwissen, kognitives oder prozedurales Wissen; die Zahlen das jeweilige kognitive Niveau: 1 → Verständnis, ... 6 → Beurteilung. In den Lernzielgraphen fließen nun die Relationen zwischen den einzelnen Lernzielen ein: In obigem Beispiel ist das konzeptuelle Lernziel 1 (imperative Konzepte) sicherlich eine Voraussetzung für das Erreichen des prozeduralen Lernziels 3 (imperative Programmieretechniken). Das konzeptuelle Lernziel 1 hat jedoch andererseits auch Faktenwissen als Voraussetzung, nämlich die Beherrschung des imperativen Sprachvokabulars. Analog benötigt man auch für die Beherrschung der Konzepte objektorientierter Sprachen entsprechendes Faktenwissen: Etwa die Beherrschung der Modellierungssprache UML. Die Beherrschung der Programmieretechniken objekt-orientierter Sprachen setzt jedoch nicht nur die Beherrschung objektorientierter Konzepte voraus; sie setzt voraus, dass objektorientierte Modellierungstechniken, -ein prozedurales Lernziel-, und die imperativen Programmieretechniken beherrscht werden. Fasst man diese Überlegungen in einem geeigneten Graphen zusammen, so ergibt sich der in Abbildung 1 dargestellte Lernzielgraph. Der Pfeil zwischen zwei Lernzielen symbolisiert dabei eine Vorrangrelation zwischen zwei Lernzielen, d.h. ein Lernziel von dem eine Kante ausgeht, ist Voraussetzung des Lernzieles, an dem diese Kante endet. Die in Abbildung 1 verwendete Vorrangrelation für Lernziele wurde erstmals von A. Staller [St06] eingesetzt.

### 3 Lernzielgraphen von Klausuraufgaben

Das soeben angegebene Beispiel zeigt die grundsätzliche Vorgehensweise bei der Erstellung des Lernzielgraphen. Im weiteren werden nun konkrete Problemstellungen, insbesondere Klausuraufgaben, aus Einführungsveranstaltungen in die Informatik analysiert und der zugehörige Lernzielgraph erstellt.

#### 3.1 Aufgabe 1: Objektorientierte Modellierung

Die erste Aufgabe, deren Lernzielgraph hier ermittelt wird, wurde im WS 2000/2001 an der TU-München als Klausuraufgabe zur Vorlesung „Einführung in die Informatik“ gestellt. Sie thematisiert einen Sachverhalt, der sich durch das Composite – Pattern beschreiben lässt:

- Aufgabe 1: Eine Prüfung kann entweder mündlich oder schriftlich abgelegt werden oder als mehrteilige Prüfung stattfinden. Jede Prüfung besitzt außerdem eine Note.
- a) Modellieren Sie diesen Sachverhalt in graphischer Darstellung und verwenden Sie dabei Ihre Kenntnisse über Entwurfsmuster.
  - b) Geben Sie graphisch ein Instanzdiagramm der Prüfung „Einführung in die Informatik I“ an, die Sie gerade ablegen.

Eine mögliche Lösung ist in Abbildung 2 angegeben.

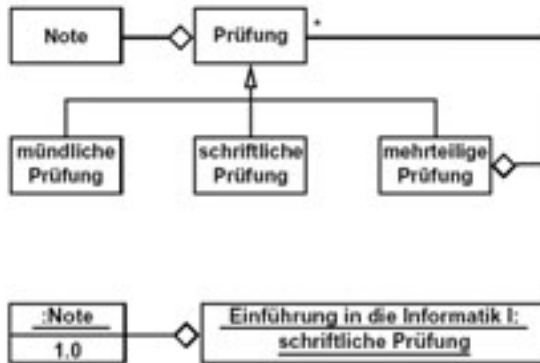


Abbildung 2: Lösung zu Aufgabe 1a und 1b

Die Formulierung der Aufgabenstellung und der Lösung bilden die Grundlage der Ermittlung des Lernzielgraphen. Dabei lassen sich zwei grundsätzlich unterschiedliche Kategorien von Lernzielen definieren: Sogenannte explizite Lernziele, deren Inhalt speziell durch die Aufgabenstellung bedingt ist und Lernziele, die implizit in der Aufgabe enthalten sind. Explizite Lernziele schlagen sich direkt in der Formulierung des Aufgabentextes nieder und bestimmen wesentlich die Formulierung der einzelnen Teilaufgaben. In Aufgabe 1 wären dies etwa folgende Lernziele:

- Rekursive Strukturen mithilfe des Composite Pattern beschreiben
- Das Composite Pattern instanzieren

Daneben gibt es implizite Lernziele, ohne die das Erreichen der expliziten Lernziele nicht möglich ist. Sie lassen sich etwa aus der Musterlösung herausarbeiten, indem man untersucht, welche Inhalte der Informatik auf welchem kognitiven Niveau benötigt werden, um die Lösung zu finden. In obigem Beispiel wäre dies alle Konzepte objektorientierter Klassenmodellierung: Klasse, Instanz, Aggregation, Vererbung, aber auch die Konzepte aus dem Bereich rekursiver Datenstrukturen. Diese Konzepte müssen jedoch in der vorliegenden Aufgabe in einem neuartigen Kontext identifiziert werden. Wir haben es deshalb mit Lernzielen der kognitiven Stufe 4 zu tun. Daneben ist jedoch auch Faktenwissen im Bereich der UML notwendig; allerdings lediglich auf Anwendungsniveau. Setzt man diese Überlegungen in einem Lernzielgraph um, so ergibt sich ein Graph wie in Abbildung 3. Die einzelnen Wissensstufen in Abbildung 3 sind dabei durch die Zahlen 1, ..., 6 bei dem jeweiligen Lernziel codiert.

### 3.2 Aufgabe 2: Referenzgeflecht und Instanzdiagramm

Für die im nächsten Abschnitt erläuterte Methode der Lernzielerfolgsanalyse wird der Lernzielgraph einer weiteren -thematisch verwandten- Aufgabe benötigt. Es handelt sich ebenfalls um eine Klausuraufgabe, bei der für ein gegebenes objektorientiertes Programm das Instanzdiagramm oder Referenzgeflecht angegeben werden soll. Auf die Präsentation der Aufgabedetails wird hier aus Platzgründen verzichtet.

### 3.3 Kombination der Lernzielgraphen und lernpsychologische Auswertung

Da es sich bei den Aufgabenstellungen in den Aufgaben 1 und 2 um Problemstellungen mit dem Themenschwerpunkt „Objektorientiertes Modellieren“ handelt, haben die beiden Lernzielgraphen gemeinsame und unterschiedliche implizite Lernziele. Diese verschiedenen Mengen von Lernzielen werden im Weiteren von Interesse sein. Den kombinierten Lernzielgraphen, der die Lernziele beider Problemstellungen umfasst, zeigt Abbildung 4: Hierbei grenzt die gestrichelte Linie die Lernziele ein, die beiden Aufgaben gemeinsam sind; die durchgezogene Linie begrenzt Lernziele, die ausschließlich der Aufgabe 1 zu zuordnen sind und die gepunktete Linie diejenigen, die nur für Aufgabe 2 relevant sind.

Aus den Vorrangrelationen sowie den Zuordnungen der Lernziele zu der revidierten Bloomschen Taxonomie lassen sich Rückschlüsse hinsichtlich des Schwierigkeitsgrades einer Aufgabe ziehen. Sinnvoll erscheint es dabei für jede Wissensstufe, also für Faktenwissen, konzeptuelles Wissen und prozedurales Wissen getrennt, jeweils in entspre-

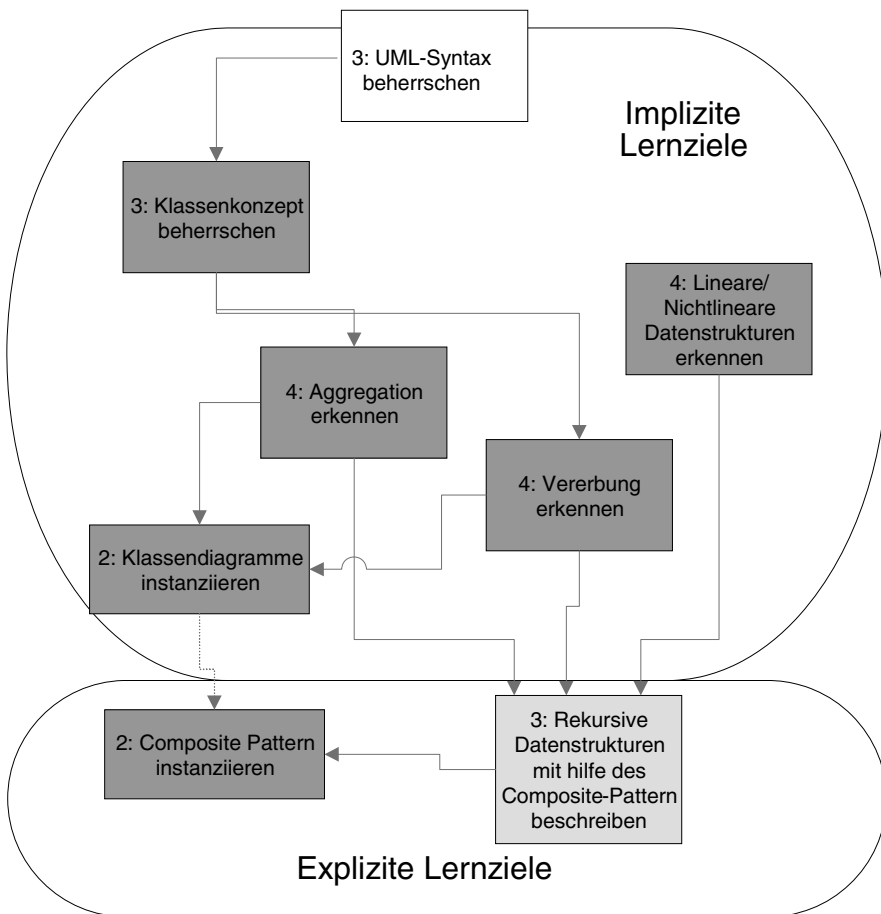


Abbildung 3: Lernzielgraph der Aufgabe 1

chende Schwierigkeitsbewertung anzugeben. Folgende Vorgehensweise bietet sich an: Die Komplexität einer Teilaufgabe und damit eines expliziten Lernziels auf einer bestimmten Wissensstufe wird bestimmt durch das Maximum der kognitiven Stufen der Lernziele derselben Wissensstufe, die von diesem impliziert werden und der kognitiven Stufe des Lernziels selbst; die Schwierigkeitsstufe wird somit nach einer Art Flaschenhalsprinzip ermittelt.

Beispielsweise hat das Lernziel der Aufgabe 1b (Composite Pattern instanziiieren) unter Berücksichtigung von Transitivitäten 5 konzeptuelle Lernziele als Voraussetzung; von diesen weisen zwei die Wissensstufe 4 auf. Somit ergibt sich für dieses Lernziel die Schwierigkeitsbewertung 4. Tabelle 1 zeigt die Schwierigkeitsstufen von Aufgabe 1 und 2 im Überblick.

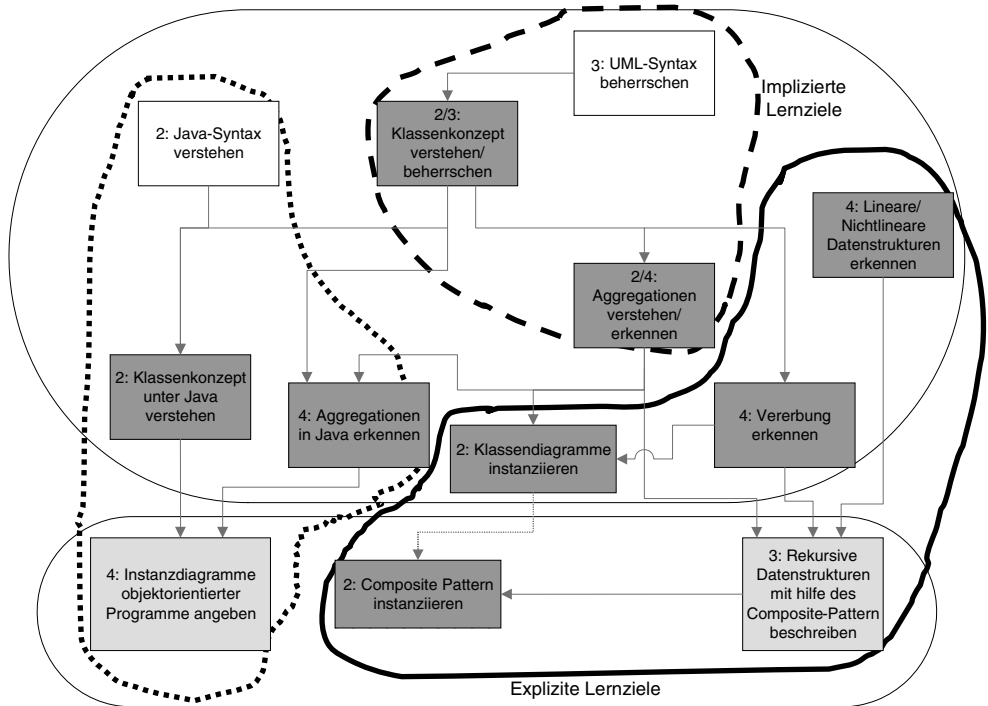


Abbildung 4: Kombiniertes Lernzielgraph der gemeinsamen und exklusiven Lernziele

Wissenstufe	Aufgabe 1 / Kognitive Stufe	Aufgabe 2 / Kognitive Stufe
<b>Faktenwissen</b>	Stufe 3	Stufe 3
<b>Konzeptuelles Wissen</b>	Stufe 4	Stufe 4
<b>Prozedurales Wissen</b>	Stufe 3	Stufe 4

Tabelle 1: Schwierigkeitsstufen gemäß dem Lernzielgraph in Abbildung 4

Eine grobe Einschätzung der in Tabelle 1 angegebenen Schwierigkeitsstufen macht deutlich, dass beide Beispiele auf den verschiedenen Wissensstufen durchaus ähnliche Schwierigkeitsstufen aufweisen. Der objektive Schwierigkeitsgrad dürfte deshalb ver-

gleichbar sein und die Studenten sollten in den beiden Aufgaben ähnliches Leistungsvermögen zeigen.

## 4 Die Lernzielerfolgsanalyse

### 4.1 Grobanalyse

In der Lernzielerfolgsanalyse soll untersucht werden, inwieweit die Studierenden die expliziten und impliziten Lernziele der oben erläuterten Problemstellungen erreicht haben. Da beide Aufgaben lernpsychologisch ähnliche Schwierigkeitsanforderungen aufweisen, sollten sie von den Studenten mit ähnlichem Erfolg gelöst worden sein. Eine einfache Mittelwertanalyse der Prüfungsleistungen dieser Klausuraufgaben, --siehe Tabelle 2--, zeigt jedoch, dass die Studierenden insgesamt, aber auch einzelne Gruppen, ein sehr unterschiedliches Leistungsvermögen in den beiden Aufgaben haben.

Auffällig ist, dass die Studierenden allgemein die Aufgabe zum Referenzgeflecht deutlich schlechter bewältigt haben als die Aufgabe zur objektorientierten Modellierung. Dies deutet darauf hin, dass die Studierenden mit den Lernzielen, die ausschließlich der Aufgabe zum Referenzgeflecht zu zuordnen sind (Java-Syntax, Umsetzung von Klassenmodellen in Java, Interpretation von OO-Programmen), mehr Schwierigkeiten haben als mit denjenigen Lernzielen, die nur für die Modellierung relevant sind.

	Modellierung	Referenzgeflecht
Alle (703)	50,85 %	33,91 %
Frauen (132)	47,92 %	23,96 %
Männer (567)	51,57 %	36,20 %
Mathematik Diplom (21)	59,52 %	54,27 %
Informatik Diplom (570)	52,68 %	35,46 %
Informatik Bachelor (66)	35,61 %	21,59 %

Tabelle 2: Mittlere relative Punktezahl der Studierenden bei den untersuchten Klausuraufgaben

Von Interesse ist auch der Unterschied zwischen weiblichen und männlichen Studierenden: Zwar erreichen die weibliche Studierenden in beiden Aufgaben eine niedrigere durchschnittliche Punktezahl als männliche, jedoch sind die Unterschiede in den Aufgaben zum Referenzgeflecht deutlich größer. Weibliche Studierende dürften somit bei den Lernzielen, die ausschließlich der Aufgabe zum Referenzgeflecht zu zuordnen sind, größere Lücken aufweisen als männliche.

Auffällig ist weiterhin die mittlere Punktezahl der Studierenden der Mathematik, die Informatik nur als Nebenfach studieren; sie bilden die Gruppe mit der höchsten mittleren Punktezahl und haben in beiden Aufgaben ein ähnliches Leistungsvermögen. Sie bilden somit eine Gruppe, die der oben angeführten lernpsychologischen Schwierigkeitsbewertung am Nächsten kommt.



## 4.2 Differenzanalyse

Anstatt wie in der Grobanalyse eine bestimmte Standardgruppe zu wählen und deren Leistungsfähigkeit in den beiden Aufgaben zu bestimmen, lässt sich die Vorgehensweise auch invertieren. Man stellt die Frage, wie viele Studierende haben beispielsweise beiden Aufgaben gut gelöst, oder welche haben die Modellierungsaufgabe gut, jedoch die Aufgabe zum Referenzgeflecht schlecht gelöst. Hintergrund dieser Strategie ist wiederum der kombinierte Lernzielgraph: Studenten, die etwa genau eine der beiden Aufgaben mangelhaft gelöst haben, haben die Lernziele, die ausschließlich der mangelhaft gelösten Aufgabe zu ordnen sind, nicht erreicht; es ist jedoch sehr wahrscheinlich, dass sie die Lernziele, die beiden Aufgaben gemeinsam sind, erreicht haben. (Vereinfachend gehen wir im weiteren davon aus, dass eine Aufgabe als mangelhaft gelöst gilt, wenn weniger als 40% der Punkte erreicht werden).

Im Detail ergeben sich folgende Fragen:

1. Welche Studenten haben beide Aufgaben mangelhaft gelöst und damit in erster Linie die beiden Aufgaben gemeinsamen Lernziele (UML-Syntax beherrschen, Klassen- und Aggregationskonzept beherrschen) nicht erreicht?
2. Welche Studenten haben genau eine der beiden Aufgaben mangelhaft gelöst und somit gerade die Lernziele, die exklusiv der mangelhaft gelösten Aufgaben zu zuordnen sind, nicht erreicht?
3. Welche Studenten haben beiden Aufgaben zufriedenstellend gelöst und damit alle Lernziele im wesentlichen erreicht?

Die Auswertung dieser Fragen liefert folgende Ergebnisse:

1. Etwa 30% der Studenten haben beide Aufgaben mangelhaft gelöst. Es ist somit sehr wahrscheinlich, dass diese Gruppe große Lücken bei den gemeinsamen Lernzielen aufweist. Die Studierenden dieser Gruppe beherrschen somit weder das Klassen- und das Aggregationskonzept noch können sie die UML-Syntax anwenden
2. Genau eine der beiden Aufgaben mangelhaft:
  - (a) Etwa 33% der Studenten haben die Aufgabe zum Referenzgeflecht mangelhaft gelöst, jedoch die Aufgabe zur Modellierung zufriedenstellend. Man kann somit davon ausgehen, dass diese die gemeinsamen Lernziele erreicht haben, nicht jedoch die Lernziele, die exklusiv der Aufgabe zum Referenzgeflecht zu zuordnen sind: Somit verstehen etwa 33% der Studenten ein Java-Programm nicht oder können das Klassenkonzept nicht in Java umsetzen, obwohl sie das Klassen- und Aggregationskonzept verstanden haben.
  - (b) Etwa 4% der Studenten haben die Aufgabe zur Modellierung mangelhaft gelöst, jedoch die Aufgabe zum Referenzgeflecht zufriedenstellend. Somit können etwa 4% der Studenten rekursive Strukturen nicht mit Hilfe des Composite-Pattern darstellen, obwohl sie das Klassen- und Aggregati-

onskonzept verstanden haben, die UML-Syntax beherrschen und das Klassenkonzept in Java umsetzen können.

3. Etwa 32% der Studenten haben beide Aufgaben zufriedenstellend gelöst und somit die vorausgesetzten Lernziele erreicht

Die Auswertung zeigt nochmals deutlich die Hauptschwierigkeiten der Studierenden: Bei etwa 30% der Studierenden liegen absolute Grundlagenprobleme vor: Basisbegriffe der Objektorientierung (Klasse, Objekt, Aggregation) wurden nicht verstanden. Weitere 30% haben große Schwierigkeiten mit dem Verständnis von Java-Programmen. Nur ca. 30% haben die Lernziele der Aufgaben wirklich erreicht. Diese Zahlen sind natürlich besorgniserregend angesichts der Tatsache, dass die Vorlesung, deren Klausuraufgaben hier analysiert werden, einen rein objektorientierten Ansatz wählte. Sie müssen jedoch auch vor dem Hintergrund der Studiensituation im WS 2000/2001 gesehen werden: Es war die Zeit als die TU München etwa 1200 Studienanfänger mit Hauptfach Informatik hatte. Unter diesen dürfte der Anteil der Studierenden, die ihre Studienentscheidung mit einem geringen Maß an Reflektion trafen, relativ hoch gewesen sein.

Sehr aufschlussreich ist es, für die soeben präsentierten Gruppen (1, 2a, 2b, 3) die Prüfungsleistungen in der gesamten Klausur zu ermitteln. Tabelle 3 zeigt die relativen Punktezahlen aller Aufgaben der Abschlussklausur im WS 2000/2001 für diese Studentengruppen (die Titel der Aufgaben sollen einen groben Anhaltspunkt hinsichtlich des Inhalts der jeweiligen Aufgabe geben)

<b>Aufgabentitel</b>	1	2 a	2 b	3
Signaturen und Terme	63,78%	81,62%	87,50%	90,17%
Boolesche Funktionen	35,74%	50,43%	72,99%	76,74%
Markov-Algorithmus	31,51%	58,17%	68,53%	79,56%
Rekursive Programmierung	28,37%	51,01%	78,45%	87,56%
Sortieren von Sequenzen von Zahlen	21,72%	37,87%	60,34%	72,50%
Reihungen	5,76%	14,37%	33,19%	55,50%
Binärbäume	8,42%	22,39%	50,69%	62,84%
Modellierung	17,15%	61,22%	21,98%	76,00%
Referenzgeflecht/Instanzdiagramm	6,10%	9,83%	70,69%	80,78%

Tabelle 3: Prüfungsleistungen in der Abschlussklausur zu „Einführung in die Informatik 1“ im WS 2000/2001

Die Studentengruppe 1 (Modellierung und Referenzgeflecht mangelhaft) weist in allen Aufgaben, die eng mit dem Klassenkonzept oder objektorientierter Programmierung verknüpft sind, also den Aufgaben zu Binärbäumen und Reihungen, im Durchschnitt eine sehr geringe Punktezahl auf. In konzeptuell anderen Aufgaben, wie beispielsweise den Aufgaben zu Markov-Algorithmen oder Booleschen Funktionen, hat diese Studentengruppe deutlich bessere Leistungen. Offensichtlich kann durch das beschriebene Ver-

fahren also der Lernzielerfolg in einer bestimmten Aufgabenklasse ermittelt werden. Entsprechende Beobachtungen lassen sich auch bei den Studentengruppen 2a und 2b machen, wo genau eine der beiden untersuchten Aufgaben mangelhaft gelöst wurde: So weist etwa die Gruppe 2a große Lücken im Bereich der objektorientierten Programmierung nicht jedoch im Bereich der Modellierung auf. Demzufolge ist das Leistungsvermögen dieser Gruppe auch in den Aufgaben zu Binärbäumen und Reihungen niedrig. Umgekehrt zeigt die Gruppe 2b, die genau die Aufgabe zum Referenzgeflecht zufriedenstellend gelöst hat nicht jedoch die Modellierungsaufgabe, in den Aufgaben zu Binärbäumen und Reihungen höhere Punktezahlen. Die Leistungen der Gruppe 3 schließlich liegen bei beiden Aufgaben im selben Bereich; dies unterstützt die oben erläuterte lernpsychologische Schwierigkeitsbewertung, wonach beide Aufgaben in allen drei Wissensdimensionen ähnliche Ansprüche aufweisen.

## 5 Ausblick

Die hier vorgestellte Technik der Lernzielerfolgsanalyse stellt eine erste Iteration des Verfahrens dar. Es wird in Zukunft verfeinert werden. Insbesondere sollte der Prozess der Erstellung des Lernzielgraphen so weit wie möglich formalisiert werden. Darüber hinaus ist die Methode der Lernzielerfolgsanalyse im Hinblick auf statistische Gründlichkeit zu verbessern: Insbesondere bedarf die Verwendung simpler Mittelwerte oder die hier willkürlich gesetzte Grenze von 40% der Punkte für eine mangelhafte Leistung einer Modifizierung. Ebenso dürften sich durch Kombination der Lernzielgraphen von mehr als zwei Klausuraufgaben Aussagen über einzelne Lernziele gewinnen lassen.

Grundsätzlich ist das Verfahren jedoch geeignet eine Vielzahl dringender didaktischer Fragestellungen zu bearbeiten. Naheliegend ist es, die hier auf zwei Aufgaben begrenzte Diskussion auf ganze Klausuren zu erweitern und somit den Lernzielgraph der betreffenden Vorlesung zu ermitteln. Dadurch lässt sich der Lernerfolg der Studierenden in den Lernzielen der Vorlesung quantitativ angeben. Die damit einhergehende Identifizierung von Lernproblemen spezifischer Gruppen von Studierenden, könnte zur Optimierung der Vorlesungsstrategie beitragen.

Eine sehr viel weiter reichende Untersuchung betrifft die konzeptuelle Strukturierung der Informatik: Durch Erstellung der Lernzielgraphen von Vorlesungen verschiedener repräsentativer Lehrstrategien können die Lernziele dieser Vorlesungen sowie deren gegenseitige Abhängigkeiten dargestellt werden. Mithilfe geeigneter Schnittmengenbildung lassen sich die zentralen Lernziele und somit letztendlich auch die zentralen Konzepte der Informatik empirisch ermitteln. Das Eingangs geschilderte Problem der mangelnden Praxisnähe konzeptueller Taxonomien ist bei der beschriebenen Methode naturgemäß behoben.

Nicht zuletzt im schulischen Bereich ist der Einsatz denkbar. Zwar fehlen im Moment die hierzu notwendigen umfangreichen Datenerhebungen, jedoch wird dieses Problem in Zukunft, wenn beispielsweise an bayerischen Gymnasien flächendeckend der Unterricht im Pflichtfach Informatik (Jahrgangsstufe 9 und 10) einsetzt, lösbar sein. Es ließe sich damit nicht nur der Lernzielerfolg in einer bestimmten Jahrgangsstufe, sondern im Rahmen einer Langzeitstudie die Entwicklung des Lernzielerfolgs in Abhängigkeit von der Lehrstrategie untersuchen.

## Literaturverzeichnis

- [AK01] Anderson, L.; Krathwol, D.: A taxonomy of Learning, Teaching, and Assessing, Addison Wesley Longman 2001
- [Bl56] Bloom, B.S. (Ed.), Engelhardt, M.D., Furst, E.J., Hill, W.H., & Krathwol, D.R. (1956). Taxonomy of educational objectives: Handbook I: Cognitive domain. New York: David McKay.
- [Ba98] Baumann, R.: Fundamentale Ideen der Informatik – gibt es das?“, in „Informatische Bildung in Deutschland, Perspektiven für das 21. Jahrhundert“, Koerber, B. und Peters I.
- [Br60] Bruner, J.S.: "The process of education", Cambridge Mass. 1960 (dt. Übers.: "Der Prozeß der Erziehung", Berlin 1970)
- [De99] Denning, P.: "Computer Science: The Discipline," Encyclopedia of Computer Science, A. Ralston and D. Hemmendinger, eds., Nature Publishing Group, 2000, pp. 405-419.
- [De03] Denning, P.: "Great Principles of Computing," Comm. ACM, Nov. 2003, pp. 15-20
- [Ra07] RWTH Aachen; Modulhandbuch als Anlage zur Prüfungsordnung des Bachelorstudiengangs Informatik; (p. 6); geprüft am 24.1.2007; <http://www.informatik.rwth-aachen.de/Studierende/Bachelor+Master/Studium/Bachelor-Master-Informatik-Modulhandbuch2006-10-17.pdf>
- [Sc93] Schwill, A.: Fundamentale Ideen der Informatik. In: Zeitschrift für Didaktik der Mathematik, 1993/1
- [Sc06a] Schneider, M.: Begriffliche Strukturen der Informatik: Ein empirischer Zugang, 3. Workshop der GI-Fachgruppe "Didaktik der Informatik", Juni 2006, Potsdam
- [Sc06b] Schneider, M.: Functional Modelling, Fundamental Ideas and Threads in the Subject Informatics. In (Dagiene, V.; Mittermeir, R.): Proceedings of the Second International Conference „Informatics in Secondary Schools: Evolution and Perspectives“; Vilnius 2006; pp. 413 – 423
- [St06] Staller, A.: Merging domain knowledge and task analysis in an ontology, in A. Méndez-Vilas, A. Solano Martín, J. Mesa González & J.A. Mesa González (Eds.), Current Developments in Technology-Assisted Education (Vol. 3), pp. 1585-1589, 2006
- [Tm07] Technische Universität München; Modulbeschreibung der Vorlesung zur „Einführung in die Informatik 1“, geprüft am 24.1.2007: <http://www.in.tum.de/studium/cm.html?id=IN0001>
- [Tb07] Technische Universität Braunschweig; Beschreibung des konsekutiven Bachelor-/Masterstudiengangs Informatik (p. 60), geprüft am 24.1.2007: <http://wiki.cs.tu-bs.de/images/e/eb/MHB-3voll.pdf>