

Lehr-Erfahrung vermittelt durch Lehr-Muster: Ein Beitrag zur Didaktik der Informatik

Klaus Quibeldey-Cirkel
Universität Siegen, Technische Informatik, D-57068 Siegen
quibeldey@ti.et-inf.uni-siegen.de

Entwurfsmuster und Mustersprachen werden als neue „literarische Form“ in der Softwaretechnik gefeiert; sie machen Wissen, das auf Erfahrung beruht, wiederverwendbar. Von der Lernökonomie und Akzeptanz der *Musterform* profitieren immer mehr Entwurfsdisziplinen, so auch die Didaktik, verstanden als Entwurf und Gestaltung von Unterricht [2, 4, 10]. Hier werden derzeit Lehrmuster („pedagogical design patterns“) der objektorientierten Informatik in einem Internet-Projekt gesammelt.¹ Mit einem Beispiel führt der Beitrag in das Konzept der Lehrmuster ein. Er versteht sich zugleich als Aufruf an alle Lehrenden der Informatik, ihre Unterrichtserfahrung in der Musterform weiterzugeben.

Schlüsselwörter: Didaktik der Informatik, Lehrmuster, Erfahrungswissen

1 Wiederverwendbares Wissen

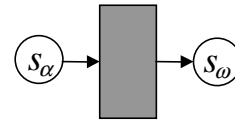
90 Prozent aller jemals forschenden und lehrenden Wissenschaftler leben heute, und so verdoppelt sich das verfügbare Wissen alle sechs Jahre [18]. Die Halbwertszeit des Informatikwissens wird auf wenige Jahre geschätzt, und Stimmen werden laut, die eine Revision der Lehrinhalte alle fünf bis sechs Jahre fordern [12]. Wie bewältigen wir die Wissensexplosion und Wissensflüchtigkeit *fachdidaktisch*? Fakt ist, daß es an Systematiken jenseits der Enzyklopädie und Fachinformationsdienste mangelt: didaktisches Wissen liegt diffus und kaum auffindbar im Literaturbestand unserer Fachgemeinschaft.

Lehr- und Lernformen der allgemeinen Didaktik werden hinlänglich diskutiert, siehe das „Handbuch Hochschullehre“ [8], was Lehrziele und -inhalte unserer Fachdidaktik betrifft, sei auf [3] verwiesen. Im folgenden geht es um die Vermittlung didaktischer *Erfahrungen (Best Practices)*, diese liegen bislang nur verstreut in den Tagungsbänden entsprechender Workshops vor [6 und Vorläufer]. Die Binsenweisheit, nach der sich nur wiederverwenden läßt, was man zuvor gefunden hat, macht uns in der Praxis zu schaffen. Hier profitiert der Experte von seinem erprobten Wissen, das er stets bei sich hat. Wünschenswert wäre ein Bestand an didaktischer Literatur, in dem praxisbewährte Lösungen organisiert vorlägen, dazu bedarf es aber einer „literarischen Form“, die Erfahrungswissen wiederverwendbar vermittelt.

¹ <http://www-lifia.info.unlp.edu.ar/ppp/index.html>

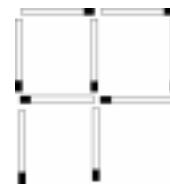
2 Was ist Erfahrungswissen?

Das Lösen eines Problems setzt Fachwissen voraus, das heißt Fakten- und Methodenwissen. Während sich Faktenwissen auf das Produkt bezieht (*Was-Information: Knowing that*), steht beim Methodenwissen der Prozeß im Vordergrund (*Wie-Information: Knowing how*). Für das Aufgabenlösen ist Fachwissen hinreichend, nicht aber für das Problemlösen; hier kommt eine andere Qualität ins Spiel. Bevor ich diese beschreibe, erläutere ich den Unterschied zwischen Aufgabe und Problem an einem Konzept aus der Denkpsychologie [5]. Nebenstehendes Bild zeigt die Komponenten eines Problems: einen unerwünschten Anfangszustand s_α , den erwünschten Zielzustand s_ω und die kognitive Barriere in der Überführung von s_α in s_ω .



Eine Aufgabe unterscheidet sich von einem Problem durch das Fehlen der Barriere; für die Bewältigung der Aufgabe ist die Methode bekannt, nur reproduktives Denken gefordert. Ob es sich um ein Problem oder um eine Aufgabe handelt, hängt also vom *Vorwissen* des Entwerfers ab. Für den erfahrenen CAD-Konstrukteur zum Beispiel ist der 3D-Entwurf kein Problem, sondern eine Aufgabe – für den unerfahrenen Laien dagegen ein Problem. An der Art der Barriere können wir die Probleme und ihre Überwindung klassifizieren:

- **Interpolationsprobleme:** Man weiß, was man will, und kennt die Mittel, um den erwünschten Zustand zu erreichen. Das Problem liegt in der geeigneten *Kombination* der Mittel, in der Interpolation zwischen Anfangs- und Zielzustand. Die Anzahl der möglichen und jeweils zu überprüfenden Transformationen ist meist sehr groß. Beispiele: Schachspiel und Stundenplanung.
- **Syntheseprobleme:** Man weiß zu Beginn, oder vermutet nach vergeblichen Anstrengungen, daß die bekannten Mittel nicht reichen. Problem des Alchimisten: Wie gewinne ich Gold aus Blei? Anfangs- und Zielzustand sind bekannt, unbekannt oder unbewußt sind wichtige Einzeloperationen, kombiniert mit den bekannten. Das Problem liegt in der Zusammenstellung, der Synthese, geeigneter Operatoren. Gelernte Einstellungen und Denkgewohnheiten können ursächlich für Synthesebarrrieren sein; Denksportaufgaben enthalten in der Regel derartige Barrieren. Rechts sehen Sie ein typisches Beispiel: Zwei Streichhölzer sollen so verschoben werden, daß drei gleich große Quadrate entstehen.²
- **Dialektische Probleme:** Man weiß, daß der Anfangszustand verändert werden muß, kennt aber den Zielzustand bloß vage oder gar nicht, allenfalls sind globale Zielkriterien bekannt, Komparative wie „besser“ oder „effizienter“. Die Lösung kann nur *dialektisch* erreicht werden, das heißt der Entwurf wird fortwährend auf



² Tip für Verzweifelte: Die Barriere liegt hier in der mentalen Voreinstellung, nur überlappungsfreie Quadrate bilden zu wollen.

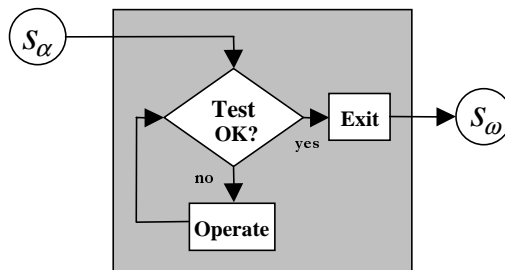
Widersprüche geprüft. Je offener das Problem hinsichtlich seines Zielzustandes, desto mehr muß versucht werden, durch zyklisches Erzeugen und Überprüfen von Alternativen die Zielkriterien zu präzisieren. In der Softwaretechnik überwinden wir dialektische Barrieren im *Dialog* mit dem Kunden (exploratorisches Prototyping).

Didaktische Probleme bergen vorwiegend Synthesebarrieren (Lehrpläne in Unterricht umsetzen) und dialektische (Lehre evaluieren). Für ihre Lösung benötigen wir Methoden- und Faktenwissen einerseits, Erfahrung im Überwinden der Barriere andererseits. Wie gesagt, ob es sich um ein Problem oder um eine Aufgabe handelt, hängt vom verfügbaren Vorwissen ab – Vorwissen, das auf Erfahrung beruht, heißt *Erfahrungswissen*.

Wie steht es mit der Kreativität? Das Lösen einer Aufgabe ist unschöpferisch, hier helfen Richtlinien, Regelwerke und Rezepte (normatives Methoden- und Faktenwissen). Dagegen ist das Lösen eines Problems a priori schöpferisch, hier helfen *Entwurfsmuster*, wie sie derzeit in der objektorientierten Softwaretechnik Furore machen [7, 14]. Entwurfsmuster sind praxisbewährte Lösungsstrukturen für stereotype Entwurfsprobleme in einem Kontext konkurrierender Kräfte. Sie sind keine dedizierte Einzellösungen, sondern bewährte Strategien und Taktiken, um eine Problembarriere zu überwinden.

In der Praxis bedeutet Entwerfen das exploratorische Suchen nach einer zufriedenstellenden Lösung, die gewisse Kriterien genügt, selbstbestimmte (Ratio, Intuition) und fremdbestimmte (Zeit- und Finanzvorgaben, technische Machbarkeit). Entwerfen folgt also einer *modalen* Logik: „erlaubt, verboten, geboten“. Entwurfsmuster beschreiben die *gebotene* Logik des Entwerfens; sie schränken die kombinatorische Vielfalt auf die praxiserprobten Wege ein. Indem sie Erfahrungswissen vermitteln, reduzieren sie Entwurfsprobleme zu Aufgaben.

Kern aller Entwurfsmuster sind *Heuristiken* (Findeverfahren: *heureka!*), deren Grundprinzip kennt viele Synonyme: „Kybernetisches Prinzip“ nach Norbert Wiener, „Vermutungen und Widerlegungen“ nach Karl Popper, „Schema und Korrektur“ nach Ernst Gombrich oder in den Ingenieurwissenschaften „Versuch und Irrtum“ [15]. Die Denkpsychologie bezeichnet das Handlungsmuster einer jeden Heuristik mit dem Akronym TOTE: TEST-OPERATE-TEST-EXIT [11]. Die simple Heuristik für das Einschlagen eines Nagels folgt dem TOTE-Zyklus: TEST: Nagel tief genug? Wenn nein, OPERATE: Hammerschlag und TEST. Wenn ja, EXIT: Nachbehandeln und Aufräumen. Erfahrungswissen gewinnen wir durch iteratives Handeln nach der TOTE-Leitlinie – und Kompetenz erlangen wir, wenn wir auf diese Weise einen Vorrat an *erfolgreichen* Mustern erwerben. Ich komme zur zentralen Frage:



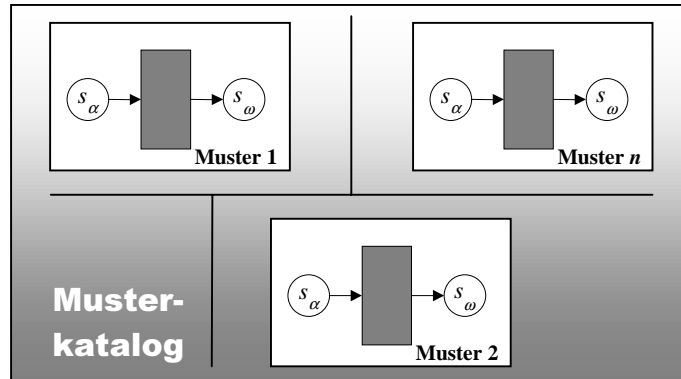
3 Wie ordnen wir Erfahrungswissen?

Entwurfsmuster sind sowohl generisch, die Lösung erzeugend, als auch deskriptiv, die Lösung beschreibend – die *prozessuale* Beschreibung eines Problem-Lösungs-Paars. Durch Lösen der Kräfte in einem Problemkontext s_α entsteht der Lösungskontext s_ω .

Ein Muster i beschreibt die notwendigen Überführungsschritte und die daraus resultierenden Strukturänderungen:

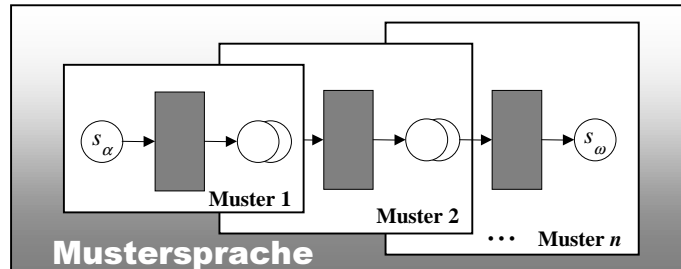
$$s_\alpha^i \xrightarrow{\text{Muster } i} s_\omega^i.$$

Lose gekoppelte Muster werden in *Katalogen* gesammelt, siehe rechts. Gibt es



überlappende Kontexte: $s_\omega^{i-1} \cong s_\alpha^i$, und bilden n Muster eine geschlossene Kontextkette: $s_\alpha^1 \xrightarrow{\text{Muster 1}} s_\omega^1 \cong s_\alpha^2 \xrightarrow{\text{Muster 2}} s_\omega^2 \cong \dots \cong s_\alpha^n \xrightarrow{\text{Muster n}} s_\omega^n$,

so spricht man von einer *Mustersprache*, siehe folgendes Bild.



In der Softwaretechnik ist die *alexandrinische* [1] Musterform – Problem, Kontext, Kräfte, Lösung –

als Erfahrungsvehikel akzeptiert und verbreitet; hier ist der transferierte Lehrinhalt Entwurfserfahrung aus den Bereichen Systemanalyse, Entwurf, Programmierung und Projektorganisation.³ Analog sucht man in der Didaktik Lehrmuster („pedagogical design patterns“); hier ist der transferierte Inhalt didaktische Erfahrung, also Problem-Lösungs-Paare der Unterrichtsgestaltung [2, 4, 10]: Wie *entwerfe* ich eine Lehrveranstaltung und ihre Komponenten? Wie Vorträge, Übungen, Labore und Studienprojekte?

³ Online-Archiv für Software-Musterkataloge und -Mustersprachen:
<http://st-www.cs.uiuc.edu/users/patterns/EgPatterns.html>

4 Ein Didaktik-Projekt im Internet

Derzeit favorisieren Dozenten und Trainer der objektorientierten Informatik eine spezielle Ausprägung der alexandrinischen Musterform: Name, Zweck, Motivation, Anwendbarkeit, Struktur, Vor- und Nachteile, Praktische Umsetzung, Bekannte Anwendungen, Verwandte Lehrmuster, Ressourcen und Referenzen. Im Gegensatz zu den didaktischen „Prinzipien“ und „Leitbildern“ der etablierten Literatur [3, 8] ist die literarische Form der Lehrmuster *pragmatisch*, das heißt handlungsorientiert.

Mehrere Dutzend Lehrmuster wurden bereits in einem Internet-Katalog gesammelt: „Pedagogical Patterns: Successes in Teaching Object-Oriented Technologies“ (siehe Fußnote 1). Projektziel ist die Publikation der Lehrmuster als Mustersprache. Die folgende Tabelle nennt die Namen einiger Lehrmuster, die sich auf die Strukturplanung einer Veranstaltung beziehen. (Ein vollständiges Beispiel dieser Kategorie gebe ich im Anhang.) Die Musternamen sollten selbstredend sein: Gemeinsame Motivation ist in diesem Fall die Förderung des aktiven Lernprozesses durch Laboraufgaben und Gruppenarbeit.

Lehrmuster zur Strukturplanung einer OOx-Veranstaltung
• Reading, Critique, Lecture, Activity, Presentation with discussion (RCLAP) Pattern (Mary Gorman, Susan Burk, American Management Systems Training Services, USA)
• Lecture-Examples-Activity-Student Presentation-Evaluation (LEASE) Pattern (Martin L. Barrett, East Tennessee State University)
• Lab-Discussion-Lecture-Lab (LDLL) Pattern (Mary Lynn Manns, University of North Carolina at Asheville)
• Lecture-Activity-Student Presentation-Discussion (LASD) Pattern (Helen Sharp, The Open University, England)
• Explore-Present-Interact-Critique (EPIC) Pattern (Jorgen Lindskov Knudsen, Ole Lehrmann, Aarhus University, Denmark)
• Preparation, Industrial Presentation and Roundtable (PIPR) Pattern (Fernando Brito e Abreu, INESC and Lisbon Technical University (ISEG), Portugal)
• Discussion-Activity-Review-Lab-Review (PIPR) Pattern (Gary L. Craig, Superlative Software Solutions, Inc., North Carolina)
• Concept, Glossary, Problem, Analyze, Discuss, Design (CoG-PADD) Pattern (Donald J. Bagert, Texas Tech University)

Aufruf zum Mitmachen

Der Beitrag führt in das Konzept der Lehrmuster ein und versteht sich zugleich als Aufruf zur Diskussion der Musterform als Vehikel für Lehr-Erfahrung. Die Siegener Fachgruppe Technische Informatik unterhält eine Mailing-Liste als Diskussionsforum für Lehrmuster aus dem deutschsprachigen Raum.⁴ Diese sollen schließlich mit dem Internet-Projekt abgestimmt und in einem globalen Lehrmuster-Katalog zusammengeführt werden.

⁴ Interessenten können sich mit dem Text *SUBSCRIBE lehrmuster* unter folgender Adresse in die Mailing-Liste eintragen: lehrmuster-request@ti.et-inf.uni-siegen.de

Literatur

- [1] Alexander, Christopher; Angel, Shlomo; Fiksdahl-King, Ingrid; Ishikawa, Sara; Jacobson, Max; Silverstein, Murray: A Pattern Language: Towns, Buildings, Construction. New York: Oxford University Press, 1977.
- [2] Anthony, Dana L. G.: Patterns for Classroom Education. In: Pattern Languages of Program Design 2. Reading u. a.: Addison-Wesley 1996.
- [3] Baumann, Rüdiger: Didaktik der Informatik. Stuttgart u. a.: Klett, 2. vollständig neu bearbeitete Auflage 1997.
- [4] Brito e Abreu, Fernando: Pedagogical Patterns: Picking Up the Design Patterns Approach. In: Object Expert, März-April 1997, S. 37-41.
- [5] Dörner, Dietrich: Problemlösen als Informationsverarbeitung. Stuttgart: Kohlhammer 1976.
- [6] Forbrig, Peter; Riedewald, Günter (Hrsg.): Software Engineering im Unterricht der Hochschulen SEUH '97. Stuttgart: Teubner 1997.
- [7] Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John: Design Patterns: Elements of Reusable Object-Oriented Software. Reading u. a.: Addison-Wesley, 1995.
- [8] Handbuch Hochschullehre: Informationen und Handreichungen aus der Praxis für die Hochschullehre, Bonn: Raabe 1994.
- [9] Kuhn, Thomas S.: The Structure of Scientific Revolutions. University of Chicago 1970.
- [10] Lilly, Susan: Patterns for Pedagogy. In: Object Magazine, Januar 1996, S. 93-96.
- [11] Miller, George A.; Galanter, Eugene; Pribram, Karl H.: Plans and the Structure of Behavior. London u. a.: Holt, Rinehart & Winston 1960.
- [12] N. N.: Interview: Professor Peter Lockemann und Dr. Wilhelm Denz zur Situation der Informatik-Ausbildung: Den stabilen Arbeitsplatz wird es nicht mehr geben. In: Informatik-Spektrum, Jg. 18, H. 2, 1995, S. 111-113.
- [13] Osterloh, Jürgen: Was tue ich eigentlich, wenn ich lehre – und was kann ich ändern? In: [8, S. A 1.3 ff.].
- [14] Quibeldey-Cirkel, Klaus: Das aktuelle Schlagwort: Entwurfsmuster. In: Informatik-Spektrum 19 (1996), H. 6, S. 326-327.
- [15] dito: Das Objekt-Paradigma in der Informatik. Stuttgart: Teubner 1994.
- [16] dito: Quo vadis, Informatik? Aspekte einer objektorientierten Entwurfslehre. In: Objekt-Spektrum 2 (1995), Heft 1, S. 30-36.
- [17] Thiele, Albert: Überzeugend Präsentieren: Präsentationstechnik für Fach- und Führungskräfte. Düsseldorf: VDI 1991.
- [18] Warnecke, Hans-Jürgen: Revolution der Unternehmenskultur: Das Fraktale Unternehmen. Berlin u. a.: Springer 1993.

Anhang: Musterbeispiel

Die klassische Musterform „Problem – Kontext – Kräfte – Lösung“ hat viele Ausprägungen erfahren, abhängig von der jeweiligen Entwurfsdisziplin. Der Ent-

wurf von Lehrveranstaltungen bedient sich der folgenden Form (Überschriften römisch I bis XI); ich beschreibe mit ihrer Hilfe eine Erfahrung, die ich aus der Strukturplanung meiner Vorlesung „Objektorientierter Systementwurf“ gewon-

nen habe. Wegen seiner besonderen Eignung für softwaretechnische Lehrveranstaltungen wurde das Muster in den Internet-Katalog „Pedagogical Patterns: Successes in Teaching Object Technology“ aufgenommen (siehe Fußnote 1).

I ETHOS

Kurzwort für eine fünfgliedrige Struktur und prägnante Gedächtnisstütze: ETHOS steht für ein universelles Muster zur Vermittlung ingenieurwissenschaftlicher Themen; es hilft, die verschiedenen Facetten eines Themas im Blick zu halten. So hat eine ingenieurwissenschaftliche Problemlösung im allgemeinen ökonomische (economic), technische, menschliche (human), organisatorische und soziale Aspekte [17].

II Zweck

Das ETHOS-Muster strukturiert einen Vortrag, eine Vortragsreihe (Vorlesung) oder ein Manuskript in übersichtlicher Form. Ziel ist die *facettenreiche*, aber *ganzheitliche* Vermittlung eines Themas.

III Motivation

Sie wollen eine Lehrveranstaltung konzipieren, zum Beispiel eine Einführungsvorlesung oder ein Industrieseminar über ein ingenieurwissenschaftliches Thema, etwa „Objektorientierte Konzepte und Methoden“. Der Lehrstoff soll in seiner *Bandbreite* vermittelt werden; der Versuch aber, auf jede Spektrallinie im thematischen Spektrum einzugehen, führt unweigerlich zu pedantisch und langweilig wirkenden Vorträgen. Entlang eines „Grundgedankengangs“ der Lehrveranstaltung müssen Gebiete ausgewählt und festgelegt werden, die Sie in längeren

Phasen detailliert behandeln wollen: Sie suchen nach didaktischen „Scheinwerfern“, die Sie ausrichten können, um die verschiedenen Wissensgebiete als Bestandteil des Grundgedankengangs auszuleuchten [13]. Ihr Anspruch läßt sich wie folgt skizzieren:

Sie wollen

- ! Abwechslung in Ihre Vorträge bringen;
- ! keinen wichtigen Aspekt vergessen;
- ! einem *roten Faden* folgen, auf dem Sie Ihre Vorträge reihen können, um ein *vernetztes* Lernen zu fördern;
- ! die Perspektive der Teilnehmer erweitern auf die interdisziplinären Bezüge.

IV Anwendbarkeit

Der Gegenstand sollte *ingenieurwissenschaftlich* sein: zum Beispiel eine durchgängige Methode für Analyse, Design und Programmierung komplexer Systeme. Weniger geeignet sind primär geisteswissenschaftliche und künstlerische Themen, bei denen wirtschaftliche oder technische Aspekte eine untergeordnete Rolle spielen.

Prüfen Sie, ob die folgenden Merkmale auf Ihr Problem zutreffen:

- ? das Lehrthema hat *paradigmatische* Breite; Paradigma bedeutet zweierlei: erstens, „Denkmuster“, ein übergeordnetes Prinzip, das für eine Teildisziplin typisch, aber nicht klar formulierbar ist, sondern sich in Beispielen manifestiert; und zweitens, „disziplinäre Matrix“, die Konstellation von Meinungen und Werten, die eine Fachgemeinschaft (*scientific community*) verbindet [9]; Beispiele für paradigmatische Breitenthemen sind: Entwurfs- und Projektplanungsme-

thoden, Multimedia, Internet und Informationstechnik;

- ? das Thema umfaßt zahlreiche und vielfältige Aspekte, die Sie in mindestens fünf Kurseinheiten (Vorträgen einer Lehrveranstaltung) vermitteln wollen;
- ? es handelt sich um eine Einführungsvorlesung oder um ein Einführungsseminar, das heißt die Vermittlung von *Orientierungswissen* im Gegensatz zum Detailwissen steht im Vordergrund.

ETHOS eignet sich weiterhin für die inhaltliche Strukturierung von:

- ? Einzelvorträgen und Fachbüchern über ein technisches Querschnittsthema;
- ? universitären Seminaren (Pro- und Hauptseminaren), hier besonders für das Ordnen der Seminarliteratur in Form eines *Readers* (Fachaufsätze, zusammengestellt in einem Ordner als Diskussionsgrundlage für die Referate) oder *Semester-Apparats* (Lehrbücher, zusammengestellt auf einem Regal, reserviert für die Veranstaltung).

V Struktur

Für die Strukturplanung einer Lehrveranstaltung ist das Akronym ETHOS streng *sequentiell* (somit auch ein Test, ob alle Aspekte beleuchtet werden), als Lese-Anweisung ist es bedingt sequentiell: ETHOS-Kapitel können auch *sporadisch*, das heißt in beliebiger Reihenfolge gelesen werden.

Grundstruktur: Als Initialwort spricht die ETHOS-Struktur für sich selbst:

- E:** wirtschaftliche (*economic*),
- T:** technische (*technical*),

H: menschliche (*human*), vor allem lernpsychologische und ergonomische,

O: organisatorische (*organizational*), vor allem projektorganisatorische,

S: gemeinschaftliche (*social*), vor allem soziologische, aber auch ökologische Aspekte.

Feinstruktur: Untergliedern und gewichten Sie eventuell die Hauptaspekte durch Indizieren, z. B.: T₁, T₂, ..., T_n.

VI Vor- und Nachteile

☞ ETHOS erlaubt *Kaleidoskop*-Vorträge, das heißt eine bunte Folge in sich abgeschlossener Einzelvorträge in einem facettenreichen Sachgebiet. Die Vorgehensweise ähnelt dem Stil der anglo-amerikanischen *Lecture*: innerhalb eines thematischen Rahmens werden interessante Einzelvorträge angeboten [13];

☞ favorisiert Breite vor Tiefe oder Generalwissen vor Spezialwissen;

☞ fördert vernetztes Lernen, vermittelt Kontextwissen;

☞ erlaubt kontinuierliches Lernen: auch wenn einzelne Vorträge versäumt wurden, verliert der Teilnehmer nicht den Anschluß; mit anderen Worten: ETHOS unterstützt die Kapselung der Vorträge als Lerneinheiten;

☞ ist das Manuskript zur Vorlesung oder zum Seminar gleich strukturiert (in ETHOS-Kapiteln), so bleibt die *Kongruenz* zwischen Vortrag und Manuskript gewahrt, das bedeutet, zeitlich bedingte Schnitte oder willkürliche Übergänge entfallen;

☞ ETHOS bietet ein organisatorisches Schema, in das Sie neue Themen (aktuelle Entwicklungen) leicht einordnen können, ohne die Grundstruktur Ihrer Veranstaltung zu ändern.

- ☞ ETHOS setzt eine umfangreiche sachlogische und didaktische Analyse des Lehrinhalts voraus;
- ☞ die Vorplanung ist zeitaufwendig, denn Bedeutung und Zusammenhang der einzelnen ETHOS-Aspekte müssen herausgearbeitet werden.

VII Praktische Umsetzung

Betten Sie die ETHOS-Aspekte in die Gesamtstruktur Ihrer Lehrveranstaltung ein, fallen Sie nicht mit der Tür ins Haus. Als Beispiel folgen die Themenübersicht und das Inhaltsverzeichnis zur Vorlesung „Objektorientierter Systementwurf“, wie sie im Studiengang Technische Informatik an der Universität Siegen seit dem Wintersemester 1994/95 gehalten wird.

ETHOS angewandt auf eine Themenübersicht:

OOS		Objektorientierter Systementwurf
		Thema
1.		Überblick: Studienmodell „Informatik-Systemtechnik“
2.		Entwurfparadigmen in der Informatik
3.		Entwurfskomplexität
4.		Komplexitätsbewältigung
5.	E	Software-Industrialisierung
6.	T₁	OOx: Abstrahieren, Teilen, Kommunizieren
7.	T₂	OOAD: Grundlagen der Analyse- und Designmethoden
8.	T₃	OOP: Definition und Einordnung der Objekt-Sprachen
9.	H₁	Kognitive Aspekte: Entwerfen als Problemlösen
10.	H₂	Vom „Tripel des Objekts“: Struktur & Verhalten & Beschränkung
11.	O	Managementaspekte: Techniktransfer und Projektorganisation
12.	S	Von einer Wissenschaft des Entwerfens
13.		Rückblick: Fragenkatalog und Lehr-Evaluation

ETHOS angewandt auf ein Inhaltsverzeichnis:

1 Paradigmenwechsel in der Informatik 1

Paradigmenwechsel im großen 2

Kuhns These 3

Paradigmenwechsel im kleinen 5

Die methodenlose Zeit 6

Programmierkunst vs.

Software-Engineering 7

Der Faktor „Mensch“ 8

SA/SD vs. OOx 10

Objektorientierte Weltmodelle 15

Ausführbare Weltmodelle 17

Szenario des objektorientierten

Entwurfs 18

2 Das Problem:

Komplexität und ihre Bewältigung 20

„No Silver Bullet“ 20

Die deskriptive Natur der Komplexität 21

Komplexität und ihre Dimensionen 24

Entwurfskomplexität 35

„Hopes for the Silver“ 42

Die „Magische Zahl Sieben“ 43

Die „Architektur des Komplexen“ 46

Dekomposition: „Divide et Impera“ 50

ETHOS-Aspekte der Objektorientierung

3 ETHOS: E wie „economic“ 61

Auf dem Weg zur

Software-Industrialisierung 62

Prinzip „Lokalisierung“ 63

Software-Wiederverwendung 66

Standard-Klassenbibliotheken 71

Wettbewerbsfaktoren 74

Produktive Software-Entwicklung 75

Qualitative Software 78

4 ETHOS: T wie „technical“ 80

Objektorientierte Konzepte 80

Abstrahieren 81

Teilen 93

Kommunizieren 96

Objektorientierte Anwendungen 104

Analyse und Design 104

Programmiersprachen 116

5 ETHOS: H wie „human“ 123

Zur Psychologie der Objektorientierung 124

Entwerfen als Problemlösen 124

Kognitive Strukturen 130

Schema & Korrektur 136

Der Beitrag des Objekt-Paradigmas 139

	Zur Philosophie der Objektorientierung	152
	Die Begriffswelt der Ontologie	153
	Ein ontologisches Objektmodell	155
6	ETHOS: O wie „organizational“	162
	Techniktransfer	162
	Fragen zur neuen Technik	163
	Fragen zur Schnittstelle	
	„Kunde-Entwerfer“	164
	Fragen zum Entwurfsprozeß	165
	Fragen zur Projektkontrolle	
	und Mitarbeiterführung	168
	Managementaspekte	170
	Homomorphie und Lean Management	171
	Objektmanagement	174
7	ETHOS: S wie „social“	183
	„The Science of Design“	183
	Das Künstliche erschaffen	185
	Curriculum einer	
	Wissenschaft des Entwerfens	186
	Der Beitrag des Objekt-Paradigmas	193
	Der architektonische Entwurf	195
	Ideale einer	
	verallgemeinerten Entwurfslehre	196
	Der gute Entwurf	
	aus architektonischer Sicht	197
	Der Beitrag des Objekt-Paradigmas	200
	Die Ontologie des Entwerfens	202
	Tiefenstruktur:	
	Zustände, Ereignisse, Gesetze	202
	Der gute Entwurf	
	aus ontologischer Sicht	204
A	Exkurse	208
	Metaphorik	208
	Klassifikation	212
	Vererbung kontra Kapselung	213
	„The Treaty of Orlando“	215
	OMG-Terminologie	218
B	Die objektorientierte Methode	
	am Beispiel	228
	Analyse	228
	Design	229
	Programmierung	231
C	Verzeichnisse	235
	Literatur	235
	Personen	254
	Glossar	258
	Stichwörter	269

VIII Bekannte Anwendungen

- ⇒ OOS-Vorlesung seit dem Wintersemester 1994/95 [16]
- ⇒ OOS-Manuskript [15]
- ⇒ Präsentationsfibel [17]

IX Verwandte Muster

Prinzipiell sind fachdidaktische Ausprägungen der klassischen Musterform „Problem – Kontext – Kräfte – Lösung“ allesamt Kandidaten für die Strukturplanung einer Lehrveranstaltung. Erfahrungen mit diesen Formen liegen als Lehrmuster bislang nicht vor.

X Ressourcen

keine.

XI Literatur

[9, 13, 15, 16, 17]