



# Migration eines Praktikums auf eine eLearning-Plattform

Studienarbeit

Lehrstuhl für Rechnernetze und Internet  
Wilhelm-Schickard-Institut für Informatik  
Fakultät für Informations- und Kognitionswissenschaften  
Universität Tübingen

von

**Marc-Oliver Pahl**

Betreuer:

Prof. Dr.-Ing. Georg Carle  
Dipl.-Inform. Heiko Niedermayer

Tag der Abgabe: 26. Juli 2007

---



---

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Tübingen, den 26. Juli 2007



### **Kurzfassung:**

Viele (universitäre) Praktika finden noch in „traditioneller“ Weise „mit Papier und Bleistift“ statt. Diese Arbeit zeigt Möglichkeiten auf, die sich durch die Umstellung des kompletten Arbeitsablaufes auf eine eLearning-Plattform ergeben.

Neben grundsätzlichen Erwägungen wird als Beispiel das Internetpraktikum des Lehrstuhls für Rechnernetze und Internet an der Universität Tübingen angeführt. Hier brachte die Umstellung einen deutlichen Rückgang des Betreuungs- und Korrekturaufwandes. Gleichzeitig wurde der Lerneffekt für die Teilnehmer vor allem bei der theoretischen Vorbereitung stark erhöht und die Aufgabenstellung im Praxisteil klarer.

Die Arbeit führt Kriterien zur Auswahl einer geeigneten eLearning-Plattform an. Verdeutlicht werden diese durch Überlegungen, die zur Erstellung der Plattform hinter dem Internetpraktikum – dem Labsystem – geführt haben. Diese Überlegungen können auch für andere PHP/ MySQL-basierte Systeme von Nutzen sein.

Der neue integrierte Arbeitsablauf durch das Systems wird aufgezeigt und die dabei entstehenden Veränderungen für Autoren, Teilnehmer, Betreuer und Korrektoren erörtert.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Zielsetzung der Arbeit . . . . .	2
1.2	Gliederung der Arbeit . . . . .	3
<b>2</b>	<b>Möglichkeiten durch und Anforderungen an ein Lernsystem</b>	<b>5</b>
2.1	In welche Teile gliedert sich ein Praktikum? . . . . .	5
2.2	Der Theorieteil . . . . .	5
2.2.1	Multiple-Choice-Fragen . . . . .	5
2.2.2	Information aufteilen . . . . .	7
2.2.3	Das Medium nutzen . . . . .	8
2.3	Der Praxisteil . . . . .	8
2.3.1	Information aufteilen . . . . .	8
2.3.2	Keine zu detaillierte Instruktionen . . . . .	9
2.3.3	Verbesserungsvorschläge erfragen . . . . .	9
2.3.4	Musterlösungen formulieren/ Hinweise für Betreuer . . . . .	9
2.3.5	Wichtigkeit der Punkte . . . . .	9
2.3.6	Gemeinsames anstelle parallelen Arbeitens . . . . .	10
2.4	Die Korrektur . . . . .	10
<b>3</b>	<b>Konzeption des Labsystems</b>	<b>11</b>
3.1	Allgemeine Grundlagen . . . . .	11
3.1.1	Was muss ein eLearning-System leisten? . . . . .	11
3.1.2	Wahl der Plattform . . . . .	12
3.1.3	Konfigurierbarkeit . . . . .	12
3.1.4	Aufteilung der Daten . . . . .	13
3.1.5	Das Benutzerinterface . . . . .	13
3.2	Spezielle Umsetzung . . . . .	14
3.2.1	Rahmenwerk . . . . .	14

---

3.2.2	Klassen und Interfaces: Abstraktion . . . . .	15
3.2.3	Modulstruktur . . . . .	15
3.2.3.1	Kommunikation mit der Datenbank . . . . .	15
3.2.3.2	Kommunikation mit der Außenwelt . . . . .	17
3.2.4	Die Lernobjekte . . . . .	17
3.2.4.1	Die Element-Basisklasse . . . . .	17
3.2.4.2	Das Seitenelement . . . . .	18
3.2.4.3	Das Multiple-Choice-Element . . . . .	18
3.2.4.4	Problem Interaktion in HTML . . . . .	18
3.2.4.5	Die Freitextfrage . . . . .	19
3.2.4.6	Das Sammelement . . . . .	19
3.2.4.7	Die Lerneinheit . . . . .	20
3.2.4.8	Der Eintrag im Zeitplan . . . . .	20
3.2.5	Fazit . . . . .	20
<b>4</b>	<b>Das Labsystem in der Praxis</b>	<b>21</b>
4.1	Das Authoring . . . . .	21
4.2	Praktikumsablauf im Wintersemester 2005/ 2006 . . . . .	23
4.3	Die Teilnehmersicht . . . . .	25
4.4	Die Betreuung . . . . .	28
4.5	Die Korrektur . . . . .	28
4.6	Punkteverwaltung . . . . .	29
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>31</b>
	<b>Literatur</b>	<b>33</b>



# 1. Einleitung

Im Fach Informatik gibt es an der Universität Tübingen diverse „Praktika“. In der Regel läuft solch ein Praktikum derart ab, dass man sich im ersten Teil theoretische Hintergründe aneignet und diese dann im zweiten Teil in praktischen Übungen anwendet. Das Lesen der Grundlagen erfolgt oft alleine. Der Praxisteil wird zumeist im Team bearbeitet. Das übt zum einen das kooperative Arbeiten ein und spart zum anderen Ressourcen: Im konkreten Fall des „Internetpraktikums“ [colm], das ich im Folgenden als Beispiel wähle, schaffen wir es, bis zu zwölf Teams pro Woche an zwei Testaufbauten mit jeweils sechs Rechnern arbeiten zu lassen. Wir können damit bis zu 24 Studenten pro Semester ausbilden.

Das Internetpraktikum ist das Praktikum des Lehrstuhls für Rechnernetze und Internet [riwe]. Es beschäftigt sich mit grundlegenden Mechanismen und Diensten in Rechnernetzen wie dem Internet<sup>1</sup>.

Dieses Praktikum fand zum ersten Mal im Wintersemester 2003/ 2004 statt; damals noch auf Basis des Buches „Mastering Networks: An Internet Lab Manual“ von Jörg Liebeherr und Magda El Zarki [LiZa04]. Ich selbst war Teilnehmer der Lehrveranstaltung. Die Aufgaben des Buches sind für die Lehre an amerikanischen Universitäten konzipiert worden. Die Instruktionen sind sehr detailliert gegeben<sup>2</sup>. Als Antwort auf die Fragen müssen des Öfteren Programmausgaben abgegeben werden. Insgesamt sind die Aufgaben interessant aber viel zu umfangreich und langwierig für unser Internetpraktikum, das mit vier bzw. drei Semesterwochenstunden in einer Prüfung angerechnet werden kann. Daher wurde ich zusammen mit Uwe Bilger für das darauf folgende Sommersemester angestellt, die Inhalte zu überarbeiten und neue Aufgaben, die in angemessenerer Zeit lösbar sind, zu stellen. Der buchbasierte Ablauf des Praktikums fand so, wie aktuell die meisten Praktika in Tübingen, statt: Man bekommt – in unserem Fall kopierte – Aufgaben und Fragen, die man dann in Form eines „Protokolls“ beantworten muss.

Ein Vorteil dieser Abgabep Praxis ist, dass man lernt, „Protokolle“ zu schreiben. Ein großer *Nachteil* solch eines Praxisteils für einen *Teilnehmer* ist, dass *viel Zeit für die Form*, also das Kenntlichmachen, auf welche Frage man sich bezieht (z.B. durch Abtippen derselben) verbraucht wird. Aus Sicht eines *Korrektors* ist die dadurch entstehende *Verschiedenheit der Form der Abgaben* ein *Nachteil*: Die Korrektur dauert deutlich länger, da man erst suchen muss, worauf sich die Antworten beziehen. Selbst wenn diese ordentlich gekenn-

---

<sup>1</sup>Wir behandeln unter anderem die Themen Funktionsweise des Ethernet, statisches Routing, dynamisches Routing, TCP/ UDP, DNS, NAT, DHCP, Firewalling mit iptables, Einbruchserkennung mit Snort, Aufsetzen eines Webservers, SSL, IPSEC, VPN, Multicast und WLAN.

<sup>2</sup>Meistens werden alle für die Durchführung des Versuches relevanten Befehle als Listing abgedruckt.

zeichnet sind, *fehlen* auf der Abgabe in der Regel *Aufgabenstellung und Instruktionen*. Man muss die Abfolge der Aufgaben im Kopf haben, um korrigieren zu können.

Ein weiteres Problem betrifft den *Theorieteil*: Dort muss eine *große Menge an Informationen* gelesen werden – ohne Kennzeichnung, was besonders relevant ist und was eher unwichtig ist. Für den Rezipienten ist es dabei schwer, zu gewichten. Wahrscheinlich fehlen ihm bei der späteren praktischen Übung einige Grundlagen, die er beim Theroiestudium für unwichtiger erachtet hat, die sich dann aber als besonders relevant herausstellen.

Bei der Konzeption der Aufgaben und Auswahl der Texte hat man als Konzipierender vor Augen, was einem wichtig ist und was nicht. Auf den ersten Blick *fehlt nur ein geeigneter Weg, dies den Lernenden anzuzeigen...*

## 1.1 Zielsetzung der Arbeit

Diese Arbeit soll aufzeigen, wie sich ein Praktikum durch den Einsatz eines computerbasierten Systems deutlich effizienter durchführen lässt. Zur Anschauung führe ich meine Umsetzung anhand des Labsystems [labs] für das Intertnetpraktikum [colm] an.

Um die in der Einleitung schon angedeuteten Verbesserungen anzugehen, hätte ich auf ein bestehendes System [BaHMH02] zurückgreifen und es adaptieren können. Die Hauptgründe, warum ich mich entschied, ein neues System „from scratch“ zu entwickeln, waren hohe Lizenzkosten [KIWa04], mangelnde Erfahrung mit existierenden Systemen (Frühjahr 2004) und vor allem die Zeit. Da ich schon mehrere auf PHP/ MySQL basierende Webportale programmiert hatte, erschien mir der Aufwand, ein komplett neues System zu erstellen, vertretbar. Im Nachhinein betrachtet, habe ich ihn erheblich unterschätzt. Es hätte wahrscheinlich nicht so lange gedauert, eine bestehende Plattform auf unsere Bedürfnisse umzukonfigurieren. Dennoch hätte es vieler Anpassungen bedurft und am Ende wäre nicht genau das entstanden, was mir vorschwebte.

Insgesamt hat sich das Projekt für alle Beteiligten gelohnt: Der Lehrstuhl hat ein passgenaues System und ich habe Erfahrung in der Umsetzung eines umfangreicheren Projekts gesammelt. Vielleicht findet die Arbeit darüber hinaus auch Einsatz bei anderen Lehrveranstaltungen oder an anderen Lehrinrichtungen.

Eine erste Version des PHP/ MySQL-basierten Websystems entstand zum Sommersemester 2004. Es kam umgehend mit unseren neuen Inhalten, die wir schon innerhalb der Software erstellt hatten, zum Einsatz. Mit der ersten Version wurde im Sommersemester 2004 und im Wintersemester 2004/ 2005 erfolgreich gearbeitet. Nur durch den Einsatz des Systems waren wir in der Lage, zu zweit die Studenten zu betreuen und gleichzeitig noch Aufgaben zu erstellen<sup>3</sup>. Von Seiten der Studierenden gab es durchweg positives Echo auf die neue Erfahrung des rechnergestützten Abgebens der Praktikumsaufgaben bis hin, dass dies das bestorganisierte Praktikum sei, an dem sie teilgenommen hätten!

Da das System in extrem kurzer Zeit entstand<sup>4</sup>, war der Code unsauber und faktisch nicht wartbar<sup>5</sup>. Das machte es nahezu unmöglich, Verbesserungsvorschläge, die den teilnehmenden Studenten<sup>6</sup> und mir einfielen sinnvoll umzusetzen. Der Konzeptionsteil 3 der Arbeit kann daher auch als Leitfaden für eine sinnvolle vor der Implementierung stehende Planungsphase dienen.

---

<sup>3</sup>Ab der Hälfte des Semesters hat uns Johannes Riedel noch tatkräftig unterstützt. Die späteren Prelabs hat großteils Heiko Niedermayer erstellt.

<sup>4</sup>Da ich inkrementell nach unserem Bedarf entwickelte, standen kurz vor Beginn des Sommersemesters 2004 nur die Komponenten zum Erstellen von Inhalten, da wir das im Rahmen unseres Hiwi-Jobs machten. Die Funktionalität für Benutzer musste ich dann innerhalb von drei Wochen komplett schreiben, damit unsere Inhalte auch genutzt werden konnten.

<sup>5</sup>Ein weiteres Beispiel für den unschätzbaren Wert einer Planungsphase *vor* Entwicklungsbeginn.

<sup>6</sup>Allen Studenten sei hier für ihre konstruktiven Hinweise gedankt!

Motiviert durch die positiven Erfahrungen und nicht zuletzt, da auch von anderen Hochschulen Interesse bestand, entschied ich mich im Dezember 2004, mit dem Entwurf einer zweiten Version zu beginnen.

## 1.2 Gliederung der Arbeit

Im anschließenden zweiten Teil der Arbeit werden Möglichkeiten aufgezeigt, die sich durch den Einsatz eines eLearning-Systems für den Theorie- und Praxisteil ergeben können. Diese können als Entscheidungshilfe für ein eLearning-System herangezogen werden.

Der dritte Teil zeigt Schlüsselaspekte bei der Konzeption eines solchen Systems auf.

Im vierten Teil wird über die Erfahrungen bei der Durchführung des Internetlabs mit dem Labsystem berichtet.

Den Abschluss der Arbeit bilden Fazit und Ausblick.



## 2. Möglichkeiten durch und Anforderungen an ein Lernsystem

Dieser Abschnitt beinhaltet Verbesserungsmöglichkeiten, die sich durch die Verwendung eines eLearning-Systems ergeben können.

Dazu werden – wie schon in der Einleitung begonnen – ausgehend von einem „traditionellen“ Praktikum, am Beispiel des Internetpraktikums vor der Umstellung, Verbesserungsvorschläge angeführt und motiviert. Dadurch werden zugleich Anforderungen an ein Lernsystem gestellt.

Das Kapitel hilft somit bei der Auswahl einer geeigneten Lernplattform als Grundlage für ein Praktikum.

### 2.1 In welche Teile gliedert sich ein Praktikum?

Universitäre Praktika – mindestens im Bereich Informatik – sind aus Sicht der Teilnehmer im Wesentlichen zweigeteilt: Ein erster Teil, in welchem die theoretischen Grundlagen (zumeist individuell) erworben werden sollen: der *Theorieteil*. Und ein zweiter Teil, in dem dieses Wissen (zumeist im Team) praktisch angewandt werden soll: der *Praxisteil*.

Aus der Gesamtsicht eines Praktikums kommt vor Phase eins die Erstellung der Aufgaben und der theoretischen Hintergründe. Nach Phase zwei müssen die Lösungen der Teilnehmer korrigiert und die Ergebnisse verwaltet werden (siehe auch 4).

### 2.2 Der Theorieteil

#### 2.2.1 Multiple-Choice-Fragen

Wie schon einleitend geschrieben, besteht ein Theorieteil im Wesentlichen aus Dokumenten, die (zumeist alleine) durchzuarbeiten sind. Ein großes *Problem* dabei ist die *Fülle an Information* und die *fehlende Lenkung* bei deren Bearbeitung<sup>1</sup>.

Dabei ist den Autoren bzw. denjenigen, die die Information recherchieren in der Regel bekannt, welche Dinge besondere Relevanz haben. Sie kennen sowohl die spätere praktische Übung als auch den Nutzen in der „realen“ Welt. Einen Lösungsansatz, nämlich die Information auf ein Mindestmaß zu beschränken und dadurch zu fokussieren halte ich –

---

<sup>1</sup>Neben eigenen Texten versuchen wir beim Internetpraktikum auf frei im Netz verfügbare verlässliche Dokumente zurückzugreifen, die damit natürlich nicht 100%ig auf unsere Bedürfnisse zugeschnitten sind. Bei ihrer späteren Tätigkeit werden die Teilnehmer genau diese Situation vorfinden.

zumindest an der Universität – für wenig sinnvoll<sup>2</sup>. Vielmehr muss und kann ein Mittel gefunden werden, den Lernenden zu motivieren und zu lenken.

Der letzte Versuch des ersten Internetpraktikums war eine Kooperation mit dem schweizer Vitels-Projekt<sup>3</sup> [vite]. Neben den Texten gibt es im Theorieteil der Vitels-Versuche *Multiple-Choice-Fragen*, die man innerhalb eines Websystems [webc] beantwortet.

Solche Fragen kann man sehr gut nutzen, um die Studenten zu lenken. Neben der Leitfunktion schaffen sie auch einen Anreiz, den Theorieteil (aufmerksam) durchzuarbeiten. Dies führt zu einer besseren Vorbereitung der Lernenden.

Beim Vitels-Projekt werden diese Möglichkeiten meines Erachtens nicht genutzt. Die Multiple-Choice-Fragen werden nämlich erst mit dem Praxisteil bewertet; also dann, wenn der Korrektor Zeit hat. Nach dieser Spanne nimmt sich der Teilnehmer aber nicht mehr die Zeit, die Korrektur aufmerksam durchzugehen. Vor allem hat er schon wieder vieles der hinter der Antwort stehenden Theorie vergessen, da er diese noch vor dem Versuch in der Vorbereitung gelesen hat. Gerade die Verwendung einer computerbasierten Lösung erlaubt mehr:

Multiple-Choice-Fragen können *sofort korrigiert* werden<sup>4</sup>.

Damit lassen sich gleich mehrere Effekte erzielen:

- Die Teilnehmer haben *zeitnah Feedback* und *erkennen dadurch ihre Defizite sofort* bei der theoretischen Vorbereitung noch vor dem Praxisteil. Im Vitels-Fall kann es schließlich sein, dass ein Lernender alle Fragen falsch beantwortet, selbst aber davon ausgeht, alles verstanden zu haben.
- Eine sofortige Korrektur erlaubt *mehrere Antwortversuche*. Der Teilnehmer kann somit seine Antwort korrigieren und *beschäftigt sich dabei eingehender* mit der Theorie<sup>5</sup>. Ein „Lernen durch falsch machen“ ist nur dann möglich, wenn es keine Punkte auf die Fragen gibt<sup>6</sup>. Deshalb sollte man hier meines Erachtens auf eine notenrelevante Beurteilung verzichten. Die Korrektoren sollten lediglich darauf achten, ob der Teilnehmer ernsthaft versucht hat, die Fragen zu lösen und diese nicht einfach „weggeklickt“ hat.
- Durch sofortige Korrektur kann auch die *Teilnahme am Praxisteil davon abhängig* gemacht werden, ob der Theorieteil erfolgreich gelöst wurde: Die Teilnehmer erhalten erst dann die Möglichkeit, die Fragen des Praxisteils zu beantworten, wenn sie den Theorieteil abgeschlossen haben. Unvorbereitetes Erscheinen wird so bis zu einem gewissen Grad ausgeschlossen<sup>7</sup>.

Auch ohne Bepunktung ist bei Ankreuzaufgaben die Gefahr des „Abschreibens“ besonders gegeben. Die Lösung lässt sich nämlich sehr einfach kommunizieren: „Antwortmöglichkeit

<sup>2</sup>Im beruflichen Leben steht man ebenfalls vor einer Fülle an mehr oder weniger relevanter Information.

<sup>3</sup>Das Vitels-Projekt ist ein Verbund mehrerer schweizerischer Universitäten. Es beschäftigt sich mit ähnlichen Themen wie unser Internetpraktikum, findet aber vollkommen virtuell statt: Man kann als Kursteilnehmer ein Zeitfenster für den Versuchsaufbau (an einer schweizer Universität) buchen und dann per Remote-Zugriff Experimente durchführen.

<sup>4</sup>Es müssen nur Antwort- und Lösungsvektor verglichen werden.

<sup>5</sup>Dies erfordert geeignetes Feedback, aus dem der Teilnehmer entnehmen kann, wo seine Fehler liegen. Wenn er beispielsweise für eine ganze Seite mit Fragen ein Feedback in der Form „Falsch“ im Sinne von „Irgendwo auf der Seite ist mindestens ein Fehler“ bekommt, hilft ihm dies wenig und motiviert auch nicht, die richtige Lösung zu suchen.

<sup>6</sup>Sobald es punkterelevant wird, werden die Teilnehmer wahrscheinlich nicht mehr ausprobieren. . .

<sup>7</sup>Näheres dazu bei der Beschreibung des Multiple-Choice-Elements.



**Abb. 2.1:** Eine Seite im Theorieteil des Internetlabs mit Multiple-Choice-Frage. Der Teilnehmer hat die Aufgabe zuerst falsch (rot hinterlegt), dann richtig gelöst und bekommt jetzt die Erklärung der Lösung (ganz unten) angezeigt.

eins und drei sind richtig.“ Auch hier kann wieder ausgenutzt werden, dass die Fragen digital präsentiert werden, indem für jeden Teilnehmer die Antwortmöglichkeiten permutiert werden<sup>8</sup>. Der Austausch der Lösung bleibt weiterhin möglich, muss aber in Form von „Der Himmel ist blau, weil das blaue Licht bei Tag am meisten von der Atmosphäre gestreut wird!“ erfolgen. Wenn die Teilnehmer sich so gegenseitig sogar noch die Lösung sagen, prägt sie sich mit großer Wahrscheinlichkeit ein. Somit ist der Lernerfolg auch durch „Abschreiben“ sichergestellt.

Ein *Problem* der automatischen Korrektur in Bezug auf den Lerneffekt ergibt sich, wenn die *Lösung nicht sofort einsichtig* ist. Dies wird oft der Fall sein. Der Teilnehmer bekommt zwar die Lösung, sieht sie aber nicht ein. Aus diesem Grund erscheint es mir wichtig, neben der Frage und den Antwortmöglichkeiten als dritten Bestandteil auch noch eine *kurze Erklärung* abzuspeichern, die die Lösung verständlich machen soll und mit ihr angezeigt wird.

### 2.2.2 Information aufteilen

Bei der traditionellen Aufbereitung von Information auf Papier gibt es natürliche Informationsgrenzen, in Form eines DIN-A4-Blattes beispielsweise. Beim digitalen Authoring gibt es diese in der Regel nicht<sup>9</sup>. Dadurch läuft man Gefahr, zu viele Information in eine

<sup>8</sup>Dies sollte freilich nur beim ersten Öffnen der Frage geschehen. Eine Änderung des Erscheinungsbildes der Lösung bei jedem Neuladen würde sehr irritieren.

<sup>9</sup>Eine Webseite ist beispielsweise theoretisch unbegrenzt.

virtuelle „Seite“ zu packen. Die meisten Lernenden werden dann aber von der Fülle an Information (Scrollbalken!) „erschlagen“.

Gerade beim Einarbeiten von Information in digitale Informationssysteme sollte man daher auf eine *sinnvolle Aufteilung der Inhalte in Unterkapitel* achten. Persönlich habe ich die Erfahrung gemacht, dass es angenehmer ist, kurze Informationseinheiten nacheinander durchzulesen, da man so den Eindruck hat, vorwärts zu kommen. Auch ist es angenehm, wenn man nicht scrollen muss.

### 2.2.3 Das Medium nutzen

Die Umstellung auf ein *neues Medium schafft andere Möglichkeiten*. Beispielsweise auf einer Webseite können *viele unterschiedliche Präsentationsformen* wie Animationen oder Filme eingebunden werden, um ein neues „Lernerlebnis“ zu schaffen. Davon sollte Gebrauch gemacht werden, wo dies als Unterstützung sinnvoll erscheint und den Lern- und Lehrerfolg erhöht.

Auch die *Rezeptionsform* kann angepasst werden, beispielsweise vom linearen hin zu einer stärker nichtlinear verlinkten Darstellung. In unserem Fall greift dies allerdings nicht, da hier alle Informationseinheiten durchgearbeitet werden müssen. Das sequentielle Vorgehen stellt implizit sicher, dass dies geschieht.

## 2.3 Der Praxisteil

Der Praxisteil besteht aus Instruktionen und zugehörigen Aufgaben. Beim „klassischen“ Praktikum liegen die Aufgaben in Papierform vor. Wieso nicht diese Aufgaben auf Papier weiterhin verwenden und den Teilnehmern online die zugehörigen Antwortfelder zur Verfügung stellen?

Bei meiner ersten Realisierung eines Online-Systems waren nur die Aufgaben und ihre Antwortfelder online. Die Instruktionen befanden sich in einem auszudruckenden PDF-Dokument mit Verweisen auf die Fragen („Q10“) an den entsprechenden Stellen. Der *Medienbruch* stellt ein nicht zu unterschätzendes Problem dar: Die Teilnehmer versuchten immer wieder, die Aufgaben zu beantworten, ohne den (Offline-) Instruktionstext zu Rate zu ziehen. Dadurch entstand unnötiger Betreuungsaufwand, da dies zu Fragen führt. Man sollte also darauf achten, *in einem Medium zu bleiben*<sup>10</sup>.

Noch etwas anderes zeigt diese Erfahrung: Unsere Fragen enthielten so viel Instruktion, dass man vermeintlich auf die Aufgaben schließen konnte. Als Autor sollte man darauf achten, *Fragen und Instruktionen strikt zu trennen* und in der digitalen Einheit „Frage“ nur die eigentliche Frage zu stellen – die Instruktionen in einen Textabschnitt davor.

Das eingangs beschriebene Problem, dass die Korrektoren den Versuchsablauf auswendig wissen müssen, war auch entstanden, weil diesen im Ursystem nur die Fragen zu den Antworten angezeigt wurden<sup>11</sup>. Die Korrektur erfordert häufig das erneute Nachlesen der genauen Aufgabenstellung. Bei der Auswahl einer Lernplattform sollte daher darauf geachtet werden, dass den Korrektoren zur Orientierung auch die Instruktionen angezeigt werden.

### 2.3.1 Information aufteilen

Beim Praxisteil gilt es noch stärker, wie im Abschnitt 2.2.2 beschrieben, die *Instruktionen aufzuteilen*. Zumeist bestehen Praktikumsaufgaben aus mehreren Teilen. Diese wiederum setzen sich oft aus kleinen Schritten zusammen.

Ich habe die Erfahrung gemacht, dass man auf dem Papier eher dazu neigt, mehrere Aufgaben zusammenzufassen („Wie sieht XY aus? Was bedeutet das für YZ?“). Dies verführt

<sup>10</sup>Durch die Umstellung auf Onlineinstruktionen ist der Betreuungsaufwand signifikant zurückgegangen.

<sup>11</sup>Die Instruktionen waren ja nicht online vorhanden.



die Teilnehmer, die Aufgaben nicht getrennt zu beantworten. Die Korrektoren müssen sich die Antworten auf die einzelnen Teile aus der Antwort herausuchen. Beim Online-Medium ist eine derartige Aufgabenzusammenfassung unnötig. Es gilt keinen Platz zu sparen. Eine echte *Aufteilung der Aufgaben* führt vielmehr dazu, dass die Teilnehmer diese auch alle bearbeiten und nicht aus Versehen Teile weglassen oder eine Mischung der Aufgaben beantworten.

Vierzig Aufgaben auf Papier wirken eher abschreckend – besonders, wenn sie sich dazu noch auf dreißig Seiten befinden. Online stellt dies kein Problem dar, da die Wahrnehmung durch den Leser eine andere ist<sup>12</sup>.

### 2.3.2 Keine zu detaillierte Instruktionen

Wie in der Einleitung beschrieben, sind die Instruktionen in [LiZa04] sehr detailliert. Wir haben bewusst darauf verzichtet. Anleitungen zu einer Problemstellung, die schon in einem vorhergehenden Versuch gelöst wurde, sind deutlich abstrakter gehalten. So geben wir im ersten Versuch den Befehl zum Setzen der Netzwerkadresse und ähnlichem explizit an. In den späteren Versuchen findet sich nur noch der Hinweis, auf welche Adressen die Karten zu setzen sind.

Dies führt dazu, dass die Teilnehmer automatisch das erworbene Wissen memorieren und einsetzen müssen. Aus eigener Erfahrung kann ich sagen, dass es so auch mehr Spaß macht, da man nicht seitenweise eigentlich irrelevante Information vorfindet, sondern (heraus-)gefordert wird.

*Weniger ist* also bei den Instruktionen *mitunter mehr*.

### 2.3.3 Verbesserungsvorschläge erfragen

Beim Durcharbeiten eines Versuches hat man als Teilnehmer oft Verbesserungsvorschläge. Im Zuge der Umstellung der Aufgaben habe ich zu jedem Versuch eine zusätzliche Frage für *Verbesserungsvorschläge und Hinweise* eingebaut. Diese wird sowohl im Theorie- als auch im Praxisteil angezeigt. Die Erfahrung hat gezeigt, dass die Lernenden beim Durcharbeiten oft wertvolle Anmerkungen und Verbesserungsvorschläge haben, diese aber bis zur nächsten Besprechung vergessen und somit nicht kommunizieren. Hier hilft das Feld<sup>13</sup>.

### 2.3.4 Musterlösungen formulieren/ Hinweise für Betreuer

Auch Praktikumsaufgaben haben einen Erwartungshorizont bzw. eine Musterlösung. Diese ist sowohl *für die Betreuer als auch für die Korrektoren sehr wichtig*. Das eLearning-System sollte daher eine Möglichkeit bieten, die Musterlösung einzublenden<sup>14</sup>. Wenn Probleme bei bestimmten Aufgabenteilen bekannt sind, können auch zusätzliche Hinweise, die nur für die Betreuer erscheinen, sehr hilfreich sein<sup>15</sup>.

### 2.3.5 Wichtigkeit der Punkte

Die Bedeutung der erreichbaren Punkte habe ich anfangs stark unterschätzt. Es hat sich gezeigt, dass sich die Teilnehmer beim *Umfang und der Qualität ihrer Lösung sehr daran orientieren*. Die Punkte sollten also mit Bedacht vergeben werden.

Wir haben es so gestaltet, dass alle Versuche bei uns jeweils gleich viele Punkte ergeben. Im Zuge der Konzipierung ist dies eine praktikable Lösung, da man nur schwer abschätzen

---

<sup>12</sup>Der Monitor wird nicht „dicker“.

<sup>13</sup>Insgesamt hat sich dieser einfache Mechanismus als Feedbackkanal sehr bewährt. Der Teilnehmer kann einfach etwas notieren und der Adressat hat nach Abschluss des Versuchs eine komplette Seite mit Verbesserungsvorschlägen und Anmerkungen.

<sup>14</sup>Zur Korrektur kann die Musterlösung zusammen mit der Teilnehmerlösung eingeblendet werden.

<sup>15</sup>Vorausgesetzt natürlich, diese Probleme sind für die Teilnehmer von didaktischem Nutzen und machen die Aufgaben nicht nur unnötig langwierig – in solch einem Fall sollte man die Aufgabe anpassen.

kann, wie schwierig die einzelnen Aufgaben für die Teilnehmer sein werden. Wenn man ihren Schwierigkeitsgrad kennt – wie dies nach mehreren Jahrgängen der Fall ist –, bietet es sich an, die Gesamtpunktzahl dem Versuch anzupassen. Wir haben zum Beispiel einige eher kurze und einfache Versuche und andere, die über mehrere Präsenztermine gehen und daher auch mit mehr Punkten in die Gesamtbewertung einfließen sollten.

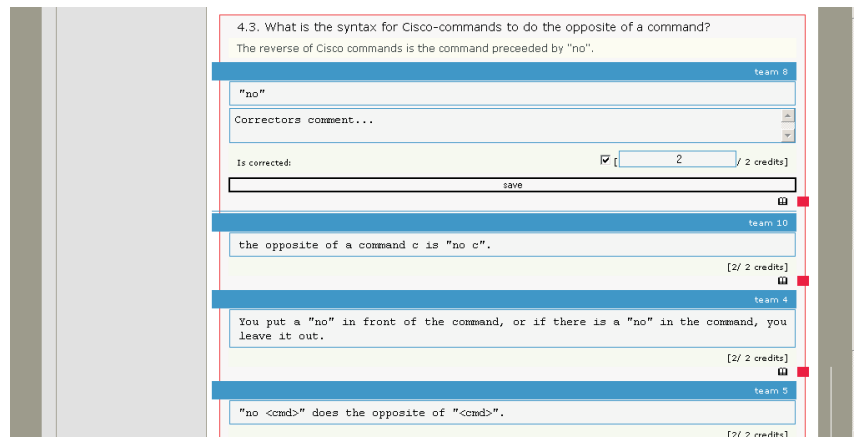
### 2.3.6 Gemeinsames anstelle parallelen Arbeitens

Bei Teamarbeit kann es – vor allem bei guten Teilnehmern – leicht geschehen, dass diese versuchen, möglichst schnell durch das Praktikum zu kommen und dafür die Aufgaben aufteilen und parallel arbeiten. Damit sind sie zwar schneller, haben jeder aber auch nur den Teil, den sie selbst gemacht haben, „erlebt“. Um dies zu verhindern, habe ich in meinem System einen Mechanismus eingebaut, der es dem *gesamten Team* immer nur erlaubt, *eine Frage zugleich* zu beantworten<sup>16</sup>.

## 2.4 Die Korrektur

Noch stärker als die Teilnehmer werden die Korrektoren durch ein computerbasiertes System entlastet. Aus Gründen der Gerechtigkeit *korrigieren sie quer*<sup>17</sup>. Bei Papierabgaben ginge dies nur, wenn sie sich genau zeitlich koordinierten und die Abgaben weiterreichten. Daneben müssten sie sich auf jeder Abgabe erst zurechtfinden.

In einer geeigneten eLearningumgebung kann der *Korrektor* sich die *Antworten aller Teams zusammen mit der Frage und der Aufgabenstellung* anzeigen lassen und in viel weniger Zeit korrigieren.



**Abb. 2.2:** Kreuzkorrektur einer Aufgabe des Praxisteils. Die zweite Zeile enthält die Musterlösung. Team 8 ist gerade zur Korrektur geöffnet.

<sup>16</sup>Als Seiteneffekt wird damit auch zu einem gewissen Grad verhindert, dass die Teammitglieder aus Versehen gegenseitig ihre Lösungen überschreiben.

<sup>17</sup>Also Korrektor A Aufgabe 1 aller Teams, Korrektor B Aufgabe 2 usw.

## 3. Konzeption des Labsystems

Um die in Kapitel 2 beschriebenen Anforderungen umzusetzen, bedarf es einer geeigneten Plattform – eines eLearning-Systems. Auf dem Markt gibt es viele solcher Systeme. Übersichten finden heute vor allem sich im Web (z.B. [bild]). Einige aktuelle Open-Source-Projekte sind [mood], [ilia], [olat].

Bei Beginn unserer Umstellung auf eine Online-Plattform Anfang 2004 war mir lediglich WebCT [webc] bekannt. Vor allem wegen der hohen Kosten und dem nicht optimalen Funktionsumfang<sup>1</sup> schied dieses aus. [BaHMH02] zeigt, dass kostengünstige brauchbare Systeme damals nicht verbreitet waren. Ich habe mich daher entschieden, ein neues eLearningssystem auf Basis von PHP und MySQL zu entwickeln [labs]. Die konzeptuellen Aspekte speziell dieses Systems möchte ich im Folgenden erläutern. Sie lassen sich auf andere (PHP/ MySQL-) Softwareprojekte übertragen.

Auch mit Kenntnis der neuen Systeme hat meine Entwicklung ihre Daseinsberechtigung. Im Gegensatz zu diesen beschränkt sich das Labsystem auf die (beschriebene) benötigte Funktionalität und ist auf schnelles effizientes Lernen optimiert<sup>2</sup>.

### 3.1 Allgemeine Grundlagen

#### 3.1.1 Was muss ein eLearning-System leisten?

Der Funktionsumfang, der für ein vollständig rechnergestütztes Praktikum benötigt wird, gliedert sich im Wesentlichen in die zwei Punkte *Content-Management* und *Course-Management*.

Der *Content-Management*-Teil ist für die Verwaltung der Inhalte zuständig. Anders als ein reines Content-Management-System muss ein eLearning-System auch mit Benutzerantworten umgehen können (speichern, korrigieren, ...).

Zum *Course-Management* zählen Komponenten, wie die Benutzerverwaltung, die Punkte der Benutzer speichern und auswerten zu können, aber z.B. auch, den Teilnehmern Versuche nur zu bestimmten Zeiten freizuschalten (Scheduling).

---

<sup>1</sup>(in Bezug auf die von mir angeführten Punkte und die komplexe Bedienung)

<sup>2</sup>Der riesige Funktionsumfang und die dadurch leicht überladene Oberfläche vieler eLearning-Systeme ist dazu hinderlich. Ich habe versucht, das System hinter dem Inhalt so weit wie möglich zurücktreten zu lassen. Dazu gehört auch eine möglichst unauffällige und gleichzeitig intuitive und übersichtliche Benutzerführung.

### 3.1.2 Wahl der Plattform

Die Wahl der Plattform ist grundlegend und hängt wesentlich von der vorhandenen Infrastruktur ab.

Eine wichtige Zielsetzung an mein System ist dessen *Erreichbarkeit* von Seite der Teilnehmer: Der Zugriff sollte möglichst „von Überall“ erfolgen können und nicht beispielsweise nur innerhalb des Praktikumsraums<sup>3</sup>. Vor allem im universitären Umfeld, welches eher heterogen bezüglich der Betriebssysteme und installierten Software ist, scheidet damit eine Lösung, die auf zu installierender Software basiert, aus. Diese ist auf Rechnern in Computer-Pools zumeist nicht vorhanden und für normale studentische Nutzer nicht installierbar, da diese keine entsprechenden Rechte besitzen.

Eine gute Lösung des Problems ist ein *webbasiertes Portal*. Webbrowser existieren für fast jedes Betriebssystem und sind innerhalb der Universität auch standardmäßig zusammen mit der physikalischen Anbindung an das Internet auf allen Rechnern vorhanden. Schon die Präsenz von Browser-Plugins wie Flash oder auch Java kann zum jetzigen Zeitpunkt nicht vorausgesetzt werden<sup>4</sup>. Deshalb setzt das System clientseitig nur einen HTML-Interpreter (Webbrowser) voraus.

Da sich auch die Browser unterscheiden, muss ein Beurteilungskriterium für „sauberen“ HTML-Code gefunden werden. Dieses existiert und zwar in Form des Validators des WorldWideWeb-Konsortiums [[valib](#)]. Sowohl der generierte HTML-Code, als auch die Stylesheets [[valia](#)] wurden auf Standardkonformität überprüft und sollten damit auf allen HTML-konformen Browsern zu akzeptabler Anzeige kommen.

Neben der breiten Installationsbasis an Browsern für HTML bietet dessen Verwendung in Kombination mit CascadedStyleSheets (CSS) einen weiteren großen Vorteil: Die schon eingebaute weitgehende Trennung von Inhalt und Aussehen. Die durchgängige Verwendung von CSS erlaubt die zentrale Veränderung des Aussehens, ohne den Code, der die Webseiten generiert, anpassen zu müssen<sup>5</sup>.

### 3.1.3 Konfigurierbarkeit

Eine weitere Grundfragestellung ist, auf welche Weise das System konfigurierbar sein soll. Anpassungen, die Änderungen im Quelltext erfordern, sind für einen breiten Benutzerkreis nicht durchführbar. In meinem System gibt es eine *zentrale Konfigurationsdatei*, aus der alle für den Kurs relevanten Daten gelesen werden. Dies ist grundlegend, da alle im Programmcode verwendeten Parameter damit stets dynamisch aus der Konfiguration gelesen werden müssen.

Nur durch Verwendung von separaten Konfigurationsdateien kann *eine* Installation des Systems für mehrere Kurse gleichzeitig genutzt werden<sup>6</sup>.

Neben den „normalen“ Konfigurationsparametern habe ich auch die Interaktion des Systems mit dem Bediener flexibel gehalten: Alle verwendeten Strings werden aus einer *Sprachdatei* eingelesen. So ist es einfach möglich, das Programm auf eine andere Sprache zu portieren.

Schließlich lassen sich auch die in 3.1.2 angesprochenen Stylesheets und somit das Aussehen des Systems für jeden Kurs ändern.

---

<sup>3</sup>Um beispielsweise den Theorieteil von zuhause bearbeiten zu können.

<sup>4</sup>Für die Inhalte können diese selbstverständlich genutzt werden. Nur das eLearning-System selbst sollte wenig Voraussetzungen auf Clientseite haben, um flexibel eingesetzt werden zu können.

<sup>5</sup>So lässt sich beispielsweise im konkreten Beispiel des Labsystems die Farbe aller zentralen Komponenten mit elf Einträgen in einen Theme-Stylesheet verändern.

<sup>6</sup>Das System ist also nur einmal auf dem Server installiert, kann aber mehrere völlig verschiedene Kurse (und damit Konfigurationen) gleichzeitig bedienen.

### 3.1.4 Aufteilung der Daten

Das Backend des Systems bildet die Datenbank. In ihr werden alle dynamischen Objekte gespeichert. Um eine sinnvolle Verteilung der Daten vorzunehmen, habe ich mir folgende Fragen gestellt: Wer generiert die Daten? Was ist die Lebensdauer der Daten? Wie häufig werden sie gebraucht?

Ich bin zu dem Schluss gekommen, das System auf drei bzw. vier Datenbanken aufzubauen, die sich auf völlig verschiedenen Systemen befinden können.

- Die erste Datenbank ist für die *langlebigen Daten*: die Texte der Theorieteile, die Instruktionen, die Fragen. . .
- Die zweite Datenbank ist für die *kurzlebigen Daten*: die Benutzerantworten, die Korrekturen, die Punkte, die Benutzerrechte. . .
- Die dritte Datenbank ist für *temporäre Systemdaten*<sup>7</sup>.
- Die vierte Datenbank beherbergt die *Benutzerstammdaten*<sup>8</sup>.

### 3.1.5 Das Benutzerinterface

Das System interagiert mit dem Bediener über seine Benutzerschnittstelle. Auf deren sinnvolle Gestaltung ist daher besonderes Augenmerk zu legen [Hani03]!

Die Bedienung des Systems muss *einheitlich* erfolgen. Hauptsächlich bedeutet dies in meinem Fall, dass für alle Elemente dieselbe Menuleiste mit denselben Symbolen erscheint und die Anordnung der Felder bei Bearbeitung der Elemente immer gleich ist. Nur so kann sich der Benutzer auf die Bedienung einstellen und schnell arbeiten<sup>9</sup>.

Die *Bedienelemente* sollten so *wenig physischen Platz* auf dem Bildschirm verbrauchen wie möglich. Alle wichtigen Funktionen sollten dabei direkt erreichbar sein: Wichtig ist der Inhalt, die dazugehörigen Bearbeitungsmöglichkeiten sollten nicht störend auffallen. Ich habe daher eine Reihe von aussagekräftigen *Symbolen* kreiert, mit deren Hilfe die Bedienung erfolgt.

*Nicht benötigte Bedienelemente* sollten *ausgeblendet* werden. Das Menusystem ist so implementiert, dass es kontextbezogen Untermenüs automatisch ein- und ausblendet. Dadurch bleibt die dargebotene Information überschaubar. Bedienelemente, die gerade nicht zur Verfügung stehen, werden ausgeblendet<sup>10</sup>.

Ein anderes Beispiel für die *Kontextsensitivität der Oberfläche* sind die Freitextfragen: Ist eine Frage aktiv<sup>11</sup>, werden die Schaltflächen zum Beantworten aller<sup>12</sup> anderen Fragen solange ausgeblendet, bis die Frage gespeichert wurde<sup>13</sup>.

An *kritischen Stellen*, also beispielsweise beim Löschen von Elementen, sollte das *System zur Sicherheit nachfragen*, um ungewollter Fehlbedienung vorzubeugen. Dem Benutzer wird so die Angst vor unvorhersehbaren schlimmen Folgen genommen, da er weiß, dass bei

<sup>7</sup>Im wesentlichen sind dies die Session-Daten.

<sup>8</sup>Sie wird nur benötigt, wenn keine anderweitige Benutzerverwaltung eingebunden wird.

<sup>9</sup>– gegebenenfalls nach kurzer Einarbeitung

<sup>10</sup>Dazu zählt beispielsweise auch, dass man temporär die Editorfunktionalität mit ihren zahlreichen Schaltflächen ausblenden kann.

<sup>11</sup>, wird also gerade beantwortet

<sup>12</sup>(Innerhalb der aktuellen Lerneinheit)

<sup>13</sup>Einem versehentlichen Nicht-Speichern (Antwort muss aktiv zum Server übertragen werden) wird dadurch ebenfalls vorgebeugt.

solchen Vorgängen eine Nachfrage erscheint<sup>14</sup>. Insbesondere, da ich keine Undo-Funktion eingebaut habe<sup>15</sup>, ist dies wichtig!

Es bietet sich an, *Farben zur Lenkung* einzusetzen. Innerhalb des Systems gibt es ein *Farbleitsystem*: Jedes Element hat seine Farbe. Der Nutzen ist nicht zu unterschätzen. Auf einer langen Seite lassen sich so zum Beispiel schnell die Multiple-Choice-Fragen identifizieren.

Die *Korrektur* der Antworten auf die Multiple-Choice-Fragen wird *farblich gekennzeichnet* (vgl. Abb. 2.1) sowie der Korrekturstatus und der *Punktebalken*<sup>16</sup> farbig hinterlegt. Der Lernende erkennt sofort an der Farbe, ob eine Antwort richtig war und ob er sich noch im „grünen Bereich“ befindet.

Bei den vorhandenen Themes ist es so, dass der *Hintergrund am dunkelsten* und der Bereich mit den eigentlichen *Inhalten am hellsten* ist. Dadurch wird die größte *Aufmerksamkeit* visuell auf den Inhalt *gelenkt*.

Die bisherigen Punkte zielen auf einfaches und schnelles Arbeiten mit dem System ab. In Bezug auf die Arbeitsgeschwindigkeit sind auch *Scrollwege zu berücksichtigen*: Ist es möglich, bei einer langen Seite (oder geringen Auflösung) an die Bedienelemente zu kommen, ohne lange auf der Seite hin- und herscrollen zu müssen? Zu diesem Zweck habe ich die Navigationselemente stets doppelt eingebaut: Oben und unten auf jeder Seite (vgl. Abb. 2.1). Nach dem Lesen muss also nicht extra nach oben gescrollt werden, um über die dort vorhandenen Navigationspfeile oder das Menu umzublättern.

Ein wichtiger Punkt bei einem Lernsystem ist auch die Buchführung über Änderungen: Die so genannte *History*. Jeder Veränderungsvorgang an einem dynamischen Element generiert einen History-Eintrag, auf den alle Benutzer direkt zugreifen können (Buchsymbole in Abb. 2.1). So lassen sich Änderungen von Teilnehmern, Korrektoren und Autoren rückverfolgen. Für die Teilnehmer ist dies vor allem relevant, um zu sehen, ob eine Frage im Nachhinein geändert wurde und um den Verlauf der Korrektur und den Korrektor zu ermitteln. Die Korrektoren sehen, wann wer auf eine Frage geantwortet hat. Die Autoren sehen, wer zuletzt an einem Element gearbeitet hat.

## 3.2 Spezielle Umsetzung

Der erste Teil dieses Kapitels hat grundlegende Überlegungen dargestellt. Dieser Teil widmet sich jetzt den konkreteren Implementierungsaspekten speziell des Labsystems [labs]. Das System teilt sich in den Rahmen und die eigentlichen Lernobjekte.

### 3.2.1 Rahmenwerk

Der Rahmen übernimmt Funktionen, die um die Lernobjekten „herum liegen“ – sowohl logisch als auch visuell.

Ich will diese nur kurz andeuten: Er stellt die Funktionalität zur *Benutzerauthentifizierung* zur Verfügung<sup>17</sup> und verwaltet auch die Nutzerrechte<sup>18</sup>. Er stellt das *Menu* bereit und

<sup>14</sup>Exploratives Erlernen der Oberfläche wird möglich.

<sup>15</sup>Implementierungsansätze wären eine RollBack-Funktion der Datenbank oder die Implementierung einer solchen „per Hand“.

<sup>16</sup>Dieser wird ab einer voreinstellbaren Prozentzahl an Punkten rot und zeigt damit an, dass die Punktzahl zu gering ist.

<sup>17</sup>Dazu existiert ein eigenes Interface. Eine authentifizierung gegen eine beliebige Quelle (Kerberos/LDAP an der Informatik in Tübingen) ist somit leicht realisierbar.

<sup>18</sup>Die Spezifikation der benötigten Rollen innerhalb eines Lernsystems ist ebenfalls sehr wichtig und grundlegend. Die wesentlichen Rollen in meinem System sind Benutzer (eingeloggt), Autor, Korrektor, Musterlösungsbetrachter, Rechteadministrator.

bringt die *Lernobjekte zur Anzeige*<sup>19</sup>. Schließlich gibt es im Rahmen auch noch ein *Mailinterface* – eine einfache und effektive Art asynchroner Kommunikation mit den Betreuern. Die Lernenden können, ohne das Portal zu verlassen, ihre Fragen mailen<sup>20</sup>.

Ebenfalls zum Rahmen würde ich die Stylesheets zählen. Wie schon am Ende von Kapitel 3.1.2 beschrieben, lassen sich Inhalt und Asehen durch die Verwendung von CSS trennen<sup>21</sup>. Zusätzlich kann man dies nutzen, um die *Präsentation dem Ausgabemedium anzupassen*. Durch die Verwendung eines Print-Stylesheets lassen sich die Inhalte sauber medienadäquat ausdrucken<sup>22</sup>.

Da die Kommunikation zwischen Rahmen und Lernobjekten einheitlich erfolgt (vgl. 3.2.3.2), kann der Rahmen auch die Generierung der *Standardverwaltungsseiten* für Objekte und ähnliches übernehmen.

### 3.2.2 Klassen und Interfaces: Abstraktion

In der Programmiersprache PHP gibt es seit Version 4.3 Unterstützung für objektorientierte Programmierung<sup>23</sup>. Gerade beim Erstellen einer Lernumgebung bietet sich die Verwendung von Klassen und Interfaces an, da diese die dem Inhalt zugrunde liegende Struktur gut in ein Programm abbilden: Wenige grundlegende Lernobjektklassen existieren, die in verschiedensten Instanzen (Inhalte) zusammen eine Lerneinheit bilden.

In 3.1.5 habe ich von einheitlicher Schnittstelle zum Benutzer gesprochen. Ebenso ist dringlich darauf zu achten, dass die Schnittstellen der Objekte zueinander vereinheitlicht sind (Interface/ *Vererbung*). Das Abstraktionsprinzip [SpK101] ist einzuhalten. Die Klassen sind also voneinander abzukapseln und Sichtbarkeitsgrenzen nicht zu brechen.

### 3.2.3 Modulstruktur

Der vielleicht wichtigste Teil neben dem Abstecken des gewünschten Funktionaltätsumfangs ist der Entwurf einer geeigneten Modulstruktur. Von mir selbst auch oft unterschätzt, hat sich bei diesem Projekt wieder gezeigt, dass jede in die Vorplanung investierte Stunde bei der Implementierung mehrfach gewonnen wird<sup>24</sup>.

Die wichtigsten Fragen für mich bei dieser Arbeit waren: Welche Funktionalität ist vielen Objekten gemeinsam? Welche Komponenten müssen miteinander kommunizieren?

Meine Aufteilung findet sich im Diagramm 3.2.3.

#### 3.2.3.1 Kommunikation mit der Datenbank

Wie schon in 3.1 dargestellt, ist eine wichtige Funktionalität der Austausch mit der Datenbank. Daher habe ich zuerst eine (Wrapper-)Klasse zur Kommunikation mit der Datenbank erstellt. Alle Anfragen gehen über sie<sup>25</sup>. Die einzelnen Lernobjekte haben darüber liegend jeweils eine (einheitliche) Schnittstelle, die sie mit den Daten aus der Datenbank versorgt (vergleiche Diagramm 3.2.3 mitte). Desweiteren haben sie Schnittstellen zu den zusätzlich benötigten Datenbanken mit den Benutzerantworten und ähnlichem (vergleiche Diagramm 3.2.3 unten).

<sup>19</sup>Dazu werden jeweils Funktionen in den entsprechenden Lernobjektinstanzen aufgerufen.

<sup>20</sup>In diesem Zusammenhang könnte die Frage nach einem Chat, einem Forum oder ähnlichem auftreten. Deren Implementierung halte ich für unnötig, da für dieses Aufgabengebiet geeignete Tools zur Verfügung stehen, die sich einfach einbinden lassen sollten.

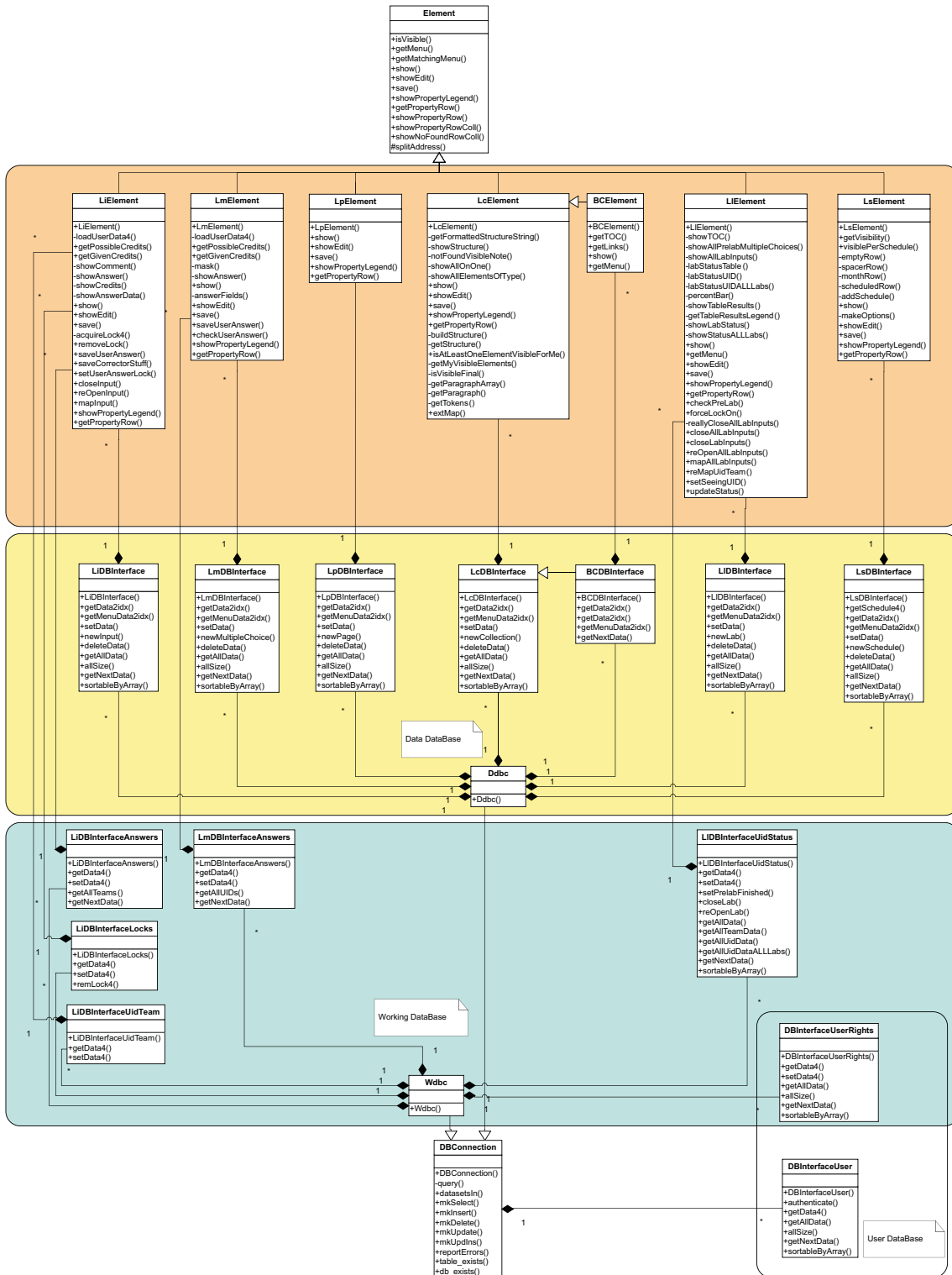
<sup>21</sup>Auch Autoren können für ihre Inhalte eigene Styles innerhalb des System erstellen. Dazu stellt der Rahmen – ebenso wie für das Menu – einen integrierten Editor für einen Benutzerstylesheet bereit. Die Definition des Layouts kann also ebenfalls innerhalb des Systems erfolgen.

<sup>22</sup>Mein Print-Stylesheet blendet das Menu aus, fügt der Schriftart wieder Serifen hinzu etc.

<sup>23</sup>In Version 4.3 fehlt noch die Unterstützung von Sichtbarkeitsbereichen. In Version 5 ist diese vorhanden.

<sup>24</sup>Insbesondere Änderungen, wie sie bei jedem Projekt unerwartet erforderlich werden, lassen sich mit einer guten Modulstruktur zumeist mit wenigen Handgriffen durchführen.

<sup>25</sup>Dadurch sollte es sogar möglich sein, die Anfragesprache (MySQL) ohne größeren Aufwand zu wechseln und das System beispielsweise mit einer Postgres-Datenbank zu betreiben.



**Abb. 3.1:** Die für die Lernobjekte relevanten Module<sup>a</sup>. Die Kommunikation zwischen Rahmen und Lernobjekten (zweite Ebene von oben) sowie den Lernobjekten untereinander erfolgt über die Interfaces oben. Die einzelnen Lernelemente erben von Element. Die mittlere Schicht bilden die Interfaces der Objekte zu der Datenbank mit den langlebigen Daten (siehe 3.1.4)<sup>b</sup>. Die dritte Schicht diejenigen zu der Datenbank mit den kurzlebigen Daten<sup>c</sup>. Die eigentliche Datenbankkommunikation erfolgt über die Klasse DBConnection. Rechts unten sind die für die Benutzerverwaltung relevanten Klassen aufgeführt.

<sup>a</sup>Das UML-Diagramm ist unvollständig; insbesondere fehlen die Variablen in der Signatur der Methoden.

<sup>b</sup>- gekapselt in Ddbc

<sup>c</sup>- gekapselt in Wdbc



### 3.2.3.2 Kommunikation mit der Außenwelt

Die Kommunikation zwischen den Lernobjekten und dem Rahmen<sup>26</sup> erfolgt ebenfalls über eine genormte Schnittstelle (vgl. 3.2.1) bzw. eine Elementbasisklasse (vgl. Abb. 3.2.3). Dies bietet sich an, da ein Großteil der Funktionalität der Lernobjekte immer gleich ist und somit sowohl Methoden als auch Variablen nur in der Mutterklasse implementiert bzw. deklariert sein müssen<sup>27</sup>. Durch die Verwendung einer genormten Schnittstelle zum Rahmen kann das System einfach um weitere Grundlernobjekte erweitert werden. Die Basisklasse liefert dazu schon einen Großteil der notwendigen Implementierung und beschleunigt das Entwickeln damit sehr.

### 3.2.4 Die Lernobjekte

Die Lernobjekte stellen das Herzstück des eLearning-Systems dar. Sie bestimmen die Möglichkeiten. Als Grundelemente gibt es das Seitenelement **Page**, die Multiple-Choice-Frage und die Freitextfrage **input**. Diese lassen sich beliebig in einem Sammelement **collection** zu einer Einheit kombinieren. Daneben gibt es dann noch die Lerneinheit und Einträge im Zeitplan **schedule**. Die Elemente werden innerhalb des Systems durch ihre Kürzel (p, m, i, c, l, s) angesprochen. Ich will nachfolgend kurz auf einige Details der Implementierung dieser Objekte eingehen.

#### 3.2.4.1 Die Element-Basisklasse

Element
<pre> +\$elementId : char +\$idx : int +\$title : string +\$matchingMenu : string +\$visibleOnlyInCollection : bool +\$history : string +\$iamVisible : bool </pre>
<pre> +isVisible() : bool +getMenu(in \$fullAddress : string, in &amp;\$menu2Populate, in \$paragraph : string = "", in \$isActive : bool = false) : string +getMatchingMenu() : string +show(in \$fullAddress : string, in \$paragraph : string = "", in \$hideNavigation : bool = false) : string +showEdit(in \$fullAddress : string) : string +save() +showPropertyLegend() : string +getPropertyRow(in \$prefix : string, in \$disabled : bool = false) : string +showPropertyRow(in \$prefix : string, in \$disabled : bool = false) : string +showPropertyRowColl(in \$address : string, in \$disabled : bool = true, in \$prefix : string) : string +showNoFoundRowColl(in \$address : string) : string #splitAddress(inout &amp;\$parentAddr : string, out &amp;\$myAddrArray : string) </pre>

**Abb. 3.2:** Element-Basisklasse.

Alle Lernelemente erben von der Element-Basisklasse. Ich möchte kurz auf ihre Eigenschaften (vgl. Abb. 3.2) eingehen.

Jedes Element kann über den Selektor „isVisible()“ gefragt werden, ob es unter den gegebenen Bedingungen<sup>28</sup> sichtbar ist<sup>29</sup>.

Die getMenu()-Methode muss das Menu für die Instanz zurückliefern. Bei einfachen Elementen ist dies lediglich eine Zeile. Bei der Collection kann das Menu lang sein, da die Methode hier rekursiv die gerade sichtbaren integrierten Elemente pollt.

Die beiden show()-Methoden liefern die Präsentation des Objektes zurück.

<sup>26</sup> – auf gewisse Weise die Darstellungsschicht und damit Schnittstelle zum Benutzer –

<sup>27</sup> Code möglichst zentral halten!

<sup>28</sup> Welche Rechte hat der Benutzer? Welche Schedules laufen gerade? Was ist der interne Zustand (bspw. Antwort geschlossen)?

<sup>29</sup> Dies dient nur als Information (bspw. für das Menu). Bei der Darstellung muss diese Eigenschaft nicht extra abgeprüft werden. Nicht sichtbare Elemente liefern auch nur eine Fehlermeldung in ihrer show()-Methode.

Die restlichen Methoden dienen im Wesentlichen der Generierung der Element-Standardverwaltungsseiten durch den Rahmen (vgl. 3.2.1).

#### 3.2.4.2 Das Seitenelement

Dieses Element ist für alle Arten von Texten gedacht. Also insbesondere für die Theorieteiltexte und die Instruktionen des Praxisteils. Es besteht im Wesentlichen aus einem Textteil. Da sich sowohl direkt Texte als auch sämtliche HTML-Tags einbauen lassen, ist dieses Element sehr flexibel. Flash-Objekte oder Java-Applets können problemlos integriert werden.

Das Seitenelement stellt keine systemgestützte Interaktion mit dem Teilnehmer zur Verfügung. Es ist daher nicht sehr komplex und eignet sich gut als Beispielimplementierung für eigene Lernobjektclassen.

#### 3.2.4.3 Das Multiple-Choice-Element

Dieses Element ist die Repräsentation der, vor allem für den Theorieteil zur Lernunterstützung vorgesehenen, Multiple-Choice-Fragen. Es besteht aus einer Frage, beliebig vielen Antwortmöglichkeiten mit Lösungsvektor und dem Feld für die Erklärung der Lösung (vgl. 2.2.1 Ende).

Das System interagiert mit dem Benutzer insofern, als dieser Antworten ankreuzen kann (vgl. 3.2.4.4). Wie in 2.2.1 besprochen, ist das Besondere meiner Implementierung, dass die Antworten für jeden Teilnehmer permutiert werden. Das System speichert dazu zu jedem Teilnehmer bei jeder Frage einen Permutationsvektor<sup>30</sup>. Die Antworten sind also für den Benutzer auch beim nächsten Betrachten der Frage in derselben Reihenfolge.

Als interaktives Element kann die Multiple-Choice-Frage verschiedene Zustände haben: Sie kann offen, also noch beantwortbar, sein und geschlossen<sup>31</sup>.

#### 3.2.4.4 Problem Interaktion in HTML

Wie in 3.1.2 dargelegt, bietet der Client-/ Serveransatz mit HTML viele Vorteile. In Bezug auf Benutzereingaben ergeben sich jedoch einige Schwierigkeiten: Daten werden nur per Push vom Client zum Server übertragen. Zustandsänderungen, wie die Beantwortung einer Frage, sind also explizit an den Server zu übertragen<sup>32</sup>. Der in HTML dazu vorgesehene Weg ist ein Formular.

Problematisch wird es, wenn mehrere interaktive Elemente auf einer Seite zusammenkommen (vgl. 3.2.4.6). Formulare kann man in HTML nicht schachteln. Dies wäre auch im Sinne der Abkapselung der Elemente zueinander nicht realisierbar, da die Antworten an die einzelnen Elemente zurückgeschickt werden. Eine kombinierte Behandlung würde die Serverzeit deutlich erhöhen, da jedes Mal potenziell alle Antworten geändert werden könnten und gespeichert werden müssten.

Zur Lösung habe ich einen Mechanismus implementiert, der dafür sorgt, dass immer nur eine Frage zugleich beantwortet werden kann. Für die Multiple-Choice-Elemente ist dies durch ein Flag, welches der Client sendet<sup>33</sup> realisiert. Graphisch werden die Bedienelemente der anderen Fragen auf der Seite solange ausgeblendet, bis die aktuell geöffnete Frage gespeichert ist (vgl. Abb. 3.3). Dadurch wird dem Benutzer auch visuell klargemacht, dass er seine Antwort zum Server senden muss, damit es weitergehen kann.

Bei den Freitextfragen (vgl. 3.2.4.5) ist der Sperrmechanismus serverseitig implementiert, da die Fragen für alle Teammitglieder und nicht nur den aktuellen Client gesperrt werden müssen (vgl. 2.3.6).

<sup>30</sup>Würde der Vektor nicht gespeichert, sprängen die Antworten bei jedem Betreten oder Reload (Speichern einer Antwort) um.

<sup>31</sup>Der Teilnehmer hat sie dann entweder richtig beantwortet oder zu viele Fehlversuche gehabt und bekommt jetzt die Lösung angezeigt.

<sup>32</sup>Sie sind insbesondere nicht synchron zur Eingabe. Eingaben beim Client sind dem Server im Moment der Eingabe unbekannt.

<sup>33</sup>„Ich will Frage drei beantworten.“

**Abb. 3.3:** Zwei Freitexteingaben. Die obere Frage ist gerade in Beantwortung. Die untere hat keine Schaltflächen und ist gesperrt (symbolisch durch das Schloss angedeutet).

### 3.2.4.5 Die Freitextfrage

Die Repräsentation der mit Freitext zu beantwortenden Frage besteht im Wesentlichen aus Frage, Musterlösung und erreichbaren Punkten. Während der Bearbeitung kommen noch die der Benutzerantwort, der Korrekturkommentar und die erreichten Punkte hinzu. Da die Freitextfrage sich nicht automatisch korrigiert, kann sie neben offen und geschlossen auch noch in Korrektur<sup>34</sup> und korrigiert sein.

### 3.2.4.6 Das Sammelement

Das Sammelement dient zur Aggregation der Grundelemente. Bei der Implementierung kommt die Verwendung der Vererbung und die damit vorhandene gemeinsame Außenschnittstelle zum Tragen: Alle aggregierten Elemente können gleich angesprochen werden. Ich habe zwei Darstellungsmodi implementiert. Einen, der alle aggregierten Elemente auf einer Seite anzeigt und einen, der jedes Element auf einer eigenen Seite ausgibt und dazu noch eine Übersichtsseite generiert. Durch Kombination beider Ansichten<sup>35</sup> lässt sich der Inhalt didaktisch sinnvoll gliedern (vgl. 2.2.2, 2.3.1).

Während sich die Sichtbarkeiten der Grundelemente an Benutzergruppen binden lässt, kann man die Sammelemente an Zeitpläne (vgl. 3.2.4.8) binden<sup>36</sup>. Sie sind dann nur während des Zeitplanes sichtbar. Da die Sammlung auch nur aus einem Element bestehen kann, lässt sich somit jedes Element an einen Zeitplan binden.

<sup>34</sup>Der Zustand „in Korrektur“ erlaubt es den Korrektoren zwar Punkte zu vergeben und diese auch zusammen mit einem Kommentar zu speichern, dies dem Teilnehmer aber noch nicht anzuzeigen, um sich beispielsweise mit anderen Korrektoren abzustimmen.

<sup>35</sup>Jedes Sammelement lässt sich je nach Aufruf auf die eine oder andere Art betrachten.

<sup>36</sup>Dadurch, dass man jedes Element in ein Sammelement packen kann, ist es möglich jedes Element an Zeitpläne zu binden. Umgekehrt ist ein Sammelement auch nur sichtbar, wenn mindestens ein enthaltenes Element sichtbar ist.

#### 3.2.4.7 Die Lerneinheit

Die Lerneinheit kapselt alle für die Durchführung einer Lerneinheit benötigte Funktionalität (vgl. 2). Sie ist das komplexeste Element. Sie stößt die Korrektur der Fragen an und generiert Übersichtsseiten mit allen Fragen sowie die Punkteübersichten.

Eine Lerneinheit besteht aus zwei Sammlungen (vgl. 3.2.4.6): Einer für den Theorieteil und einer für den Praxisteil.

Auch die Lerneinheit lässt sich an einen Zeitplan (vgl. 3.2.4.8) binden.

#### 3.2.4.8 Der Eintrag im Zeitplan

Ein Eintrag im Zeitplan besteht aus einem Start-, einem Endzeitpunkt und einem Link auf das Element, das er bindet.

#### 3.2.5 Fazit

Die vorgestellten Lernobjekte ermöglichen es, ein Praktikum auf die in den vorherigen Kapiteln vorgestellte Weise durchzuführen. Das System ist sicher weniger flexibel als andere Lernsysteme. Dafür ist es aber auch nicht so komplex und damit zum einen leicht handhabbar und zum anderen auch leicht erweiterbar<sup>37</sup>.

---

<sup>37</sup>Für Chats, Foren und andere möglicherweise erwünschte Funktionalität existieren genügend Standardimplementierungen, die leicht eingebunden werden können.

## 4. Das Labsystem in der Praxis

Wie in 1.1 schon angeführt, kam die erste Version des Systems bereits im Sommersemester 2004 zum Einsatz. Die im vorigen Kapitel beschriebene vollständige Überarbeitung ist seit Sommersemester 2005 am Lehrstuhl für Rechnernetze und Internet [riwe] der Universität Tübingen zur Durchführung des Internetlabs [colm] im Einsatz. Seit dem Wintersemester 2005/ 2006 ist ebenfalls eine Installation an der Fachhochschule Albstadt-Sigmaringen im Einsatz.

Da ich das Projekt von Anfang an begleitet habe, will ich in diesem Abschnitt von den Erfahrungen, die sich mit der Umstellung ergaben, berichten.

### 4.1 Das Authoring

Das Erstellen von Inhalten, die in einer Webumgebung präsentiert werden, bietet andere Möglichkeiten als ein Blatt Papier als Zielmedium: Es kann zwischen viel mehr *verschiedenen Darstellungsformen* ausgewählt werden (vgl. 2.2.3).

Wir haben dies beim Erstellen von neuen Inhalten für das Internetlab nur zum Teil genutzt. Anstelle Informationen in unserem System bereitzustellen, verweisen wir des Öfteren direkt auf frei verfügbare Webseiten mit relevanten Informationen<sup>1</sup>. Dem Teilnehmer wird so nebenbei gezeigt, wo sich die relevanten Informationen im Netz (auch für eigene Recherchen in Bezug auf nicht direkt behandelte Themen) finden lassen.

Ein Defizit sehe ich bei unseren Inhalten im Bereich der interaktiven Elemente. Wir haben nur wenige Animationen und Applets, in denen die Teilnehmer direkt Sachverhalte ausprobieren können. Die praktischen Erfahrungen der Lernenden im Praxisteil gleichen dies zu einem gewissen Teil aus. Viele Dinge lassen sich aber in einer Animation besser veranschaulichen (Beispielsweise die Verteilung von Paketen in einem Netzwerk) und lernen sich im Theorieteil damit einfacher. Grund für das Fehlen ist, dass wir keine Zeit hatten, solche Animationen zu suchen oder zu erstellen. Hieran kann in Zukunft noch gearbeitet werden.

In 2.2.2 und 2.3.1 habe ich die nicht mehr vorhandene „natürliche“ Segmentierung der Inhalte durch DIN-A4-Seiten angeführt. Es besteht die Gefahr, den Leser durch ein zu viel an Information abzuschrecken (sehr lange Webseiten). Sinnvoll genutzt bewirkt die

---

<sup>1</sup>Ein Problem, dass dadurch entsteht ist, dass Links im Internet zerbrechen und (schlimmer noch) Inhalte der verwiesenen Seiten sich einfach ändern können (Wikipedia). Als Lösung bietet es sich an, auf solche Seiten nicht zu linken (Microsoft-Links waren beispielsweise nach einem halben Jahr tot) oder diese lokal zu spiegeln (Urheberrecht beachten).

The screenshot shows a web browser window displaying the 'internetlab' portal. The browser's address bar shows 'uni-tuebingen/informatik/rechnernetze und internet/internetpraktikum'. The page title is 'root@PC1> internetlab'. The date and time are 'Tue, 3 Jan 2006 23:58:58 +0100'. The page is divided into a left sidebar menu and a main content area.

**Left Sidebar Menu:**

- mop
  - log out
  - myRights
- lab
  - index
  - thursday slides
  - FAQ
  - send mail
  - forum
- labss05
- labws0405
- labss04
- labs
  - labindex
  - The demonstration lab
    - prelab - demonstr...
    - 1. prelab - de...
    - 2. Suggestions...
    - all questions
    - prelab status
    - lab - demonstrati...
    - lab status
  - The basics
    - lab - demonstration lab
  - Static routing
  - Dynamic routing
  - TCP/UDP
  - DNS
  - NAT/DHCP
  - Security I
  - Security II
  - IP Multicast
  - Wireless Lan
- schedule
  - schedule
  - roomschedule
  - interviews

**Main Content Area:**

## The demonstration lab

2005: Marc-Oliver Pahl

### 1. prelab - demonstration content

#### Prelabtext for the demo lab

Hello Marc-Oliver Pahl (mop), this lab is just for getting familiar with the lab system!

You find all elements here you'll find in a "real lab" to get familiar with them.

All labs have two parts:

- The prelab
- The lab

The **prelab** should give you the **theoretical background** of what you will do later on in the lab session. The more carefully you read the texts, the easier the lab will be for you. There are multiple choice questions after most of the prelab sections. These questions should help you to recover and memorize the most important aspects of the topics you have just read about.

As pointed out in our first meeting, you find a button "**check prelab**" on the overview page of all prelab questions (last menu entry of the prelab). Clicking on it makes the system check your answers. At the moment you have three (3) attempts to get the correct answer. Then the System will give you the answer and sometimes an explanation to it.

To be able to perform the lab, **each groupmember** has to have answered **all prelab-questions**. When all your team mates are finished with the prelab session, the lab session will become visible to you.

In the **lab** you will use the techniques you read about in the prelab. Here, you solve some exercises together as a team. The available lab text gives you some instructions. During the lab you'll have to **answer some questions**. You do this as a team (as you will see both of you have the same answering fields). You should switch formulating the answers so that each groupmember writes down some of the answers! It is a big challenge and exercise to formulate and write down ones's thoughts clearly...

1.2. This is a multiple choice question. Click on "give answers" to set the check marks. Don't forget to "save"...

answer1 (correct)

answer2 (wrong)

answer3 (correct)

Here you might find some remarks why the solution is correct.

1.3. This is another multiple choice question. Now I don't give you the answers ;)

When was the first webbased internet lab held in Tübingen?

winter semester 2003/ 2004

summer semester 2004

winter semester 2004/ 2005

summer semester 2005

winter semester 2005/ 2006

Well the lab started in winter 2003/ 2004 based on a book of Jörg Liebeherr. It was maintained by Uwe Bilger then. In summer 2004 it got totally renewed by Marc-Oliver Pahl and Uwe Bilger. The later prelabs were edited by Heiko Niedermayer. All was put on the predecessor of the system you are just using. The current system was introduced in summer 2005.

© Lehrstuhl für Rechnernetze und Internet © IHSYSTEM mop 2005

Abb. 4.1: Das Webportal <http://colmar.informatik.uni-tuebingen.de>

freie *aktive Seiteneinteilung* allerdings genau das Gegenteil. Die Praxis hat bestätigt, dass kurze Texteinheiten dem Leser das Gefühl geben, voranzukommen. Es macht dabei einen Unterschied, ob die Inhalte auf *einer* Seite in kurzen Abschnitten oder immer nur wenige Abschnitte auf *vielen* Seiten präsentiert werden. Im letzteren Fall wird die gleiche Menge an Text als deutlich weniger wahrgenommen<sup>2</sup>. Da auch der Autor dieser Wahrnehmung unterliegt, obliegt es diesem umso mehr, auf einen angemessenen Umfang zu achten.

Anders als eine gedruckte Versuchsanleitung verleiten Webseiten nicht so sehr dazu, nach vorne zu „spicken“. Bei einer sinnvollen Aufteilung werden die Teilnehmer nicht durch spätere Aufgabenstellungen und Schaubilder abgelenkt. Die Versuchsteilnehmer sind so *konzentrierter* auf den aktuellen Aufgabenteil. Die Gesamtheit des Versuches geht dabei nicht verloren, da sie sich in der Dokumentstruktur, die auf der ersten Webseite des Praxis-teils präsentiert wird, widerspiegelt. Durch eine sinnvolle Aufteilung wird Fehlerpotenzial und damit der Betreuungsaufwand vermindert.

Wie schon in 2.3.1 beschrieben, hat es sich sehr bewährt, in ein Freitextfragenelement (3.2.4.5) jeweils *nur eine (Teil-)Frage* einzubauen. Die Antworten der Teilnehmer sind dadurch präziser auf die Frage gerichtet und die Korrektur wird schneller. Einzuwenden ist hier allerdings, dass die Teilnehmer dadurch nicht mehr lernen, genau zu lesen und Aufgabenstellungen auch einzeln zu erfassen – ein wichtiger Bestandteil des Informatikstudiums. Der Vorteil einer beschleunigten Korrektur und weniger Rückfragen überwiegt hier meines Erachtens.

Ein letzter Punkt, den ich hier anführen möchte, ist die sofortige *Änderungsmöglichkeit* von Online-Inhalten<sup>3</sup>. Gerade beim ersten Durchlauf unserer neuen Versuche stellten sich Instruktionen und Aufgaben oft als verbesserungsbedürftig heraus. Da wir als Betreuer anwesend waren, konnten wir unsere Erkenntnisse aus den Teilnehmerrückfragen umgehend für nachfolgende Teams in das System einfließen lassen. Dies ist natürlich problematisch, da frühere Teams dann eventuell gewisse Hilfestellungen nicht hatten. Hier greift die *History*, anhand derer jeder Änderungen nachvollziehen kann (vgl. 3.1.5). Mögliche Ungerechtigkeiten werden so auch für die Teilnehmer transparent. Bei der Korrektur konnten wir dann ebenfalls anhand der History nachvollziehen, ob die Antwort vor oder nach der Änderung erfolgte und dies bei der Punktevergabe berücksichtigen<sup>4</sup>.

## 4.2 Praktikumsablauf im Wintersemester 2005/ 2006

Dieser Abschnitt ist ein Einschub, in dem ich kurz anhand eines Versuches beschreiben möchte, wie der Ablauf des Praktikums bei uns im Wintersemester 2005/ 2006 aussieht.

Unser Versuch beginnt Donnerstag morgens mit einem kurzen Vortrag zu einigen der vorkommenden Inhalte. Der Theorieteil ist zwar schon seit Montag verfügbar, aber unser Teilnehmer hat bisher erst kurz reingeschaut. Wie die Meisten erledigt er die theoretische Vorbereitung am Wochenende. Dazu loggt sich der Teilnehmer auf der Portalseite [colm] mit seinem persönlichen Account ein, arbeitet die Inhalte durch und beantwortet die Multiple-Choice-Fragen. Wenn er alle Fragen beantwortet hat, klickt er auf eine Schaltfläche im Portal und das System überprüft seine Antworten. Er bekommt umgehend Feedback, welche Aufgaben richtig gelöst wurden und welche falsch (vgl. Abb. 2.1). Eine Aufgabe wird als falsch angezeigt. Bei dieser Aufgabe war unser Student auch unsicher und entscheidet sich jetzt für die andere Lösung. Eine erneute Überprüfung zeigt ihm an,

---

<sup>2</sup>Im Gegensatz zu allen anderen Labs ist in unserem TCP/ UDP-Lab genau der Fall vorhanden, dass sehr viele Instruktionen und Aufgaben auf einer Seite stehen. Regelmäßig ist dieses Lab dasjenige, bei dem die Teilnehmer über den Umfang stöhnen.

<sup>3</sup>Die Änderung wird dabei sofort propagiert, da ab dem Änderungszeitpunkt nur noch die neue Formulierung angezeigt wird.

<sup>4</sup>Die Hinweise der Teilnehmer in den ersten Durchgängen waren von enormer Hilfe bei der Optimierung der Aufgaben. Ihnen sei an dieser Stelle noch einmal ausdrücklich dafür gedankt.

The screenshot shows the 'internetlab' interface for a user 'root@PC1'. On the left is a navigation menu with categories like 'mop', 'lab', 'labs', 'schedule', 'admin', and 'improvements'. The main content area is titled 'DNS' and shows a table of team progress:

#	team	m-credits	i-credits
1	i9	16/18	46/60

Below the table, the interface displays 'Answers of ALL USERS' for several questions:

- 1. General stuff**
  - 1.2. List three of the Internet's root DNS servers! How did you find them?
 

One possibility: the local bind's/ named's db.cache file... usually in /var/named/db.cache

```
host -v -t ns .
```

a.root-servers.net//198.41.0.4//ns.internic.net//Dulles, VA, USA  
 b.root-servers.net//128.9.0.107//ns1.isi.edu//Marina Del Rey, CA, USA  
 c.root-servers.net//192.33.4.12//c.psi.net//Herndon, VA and Los Angeles, CA, USA

We found a list at:  
[http://www.topipguide.com/free/t\\_DNSRootNameServers-3.htm](http://www.topipguide.com/free/t_DNSRootNameServers-3.htm)

[3/ 3 credits]

correct
  - 1.3. How does your PC know where these root nodes are? Does it know? Does it have to know?
 

It doesn't have to know.  
 It is sufficient that it knows the node on top of itself (e.g. the nameserver of your provider).  
 This node will know the root nodes and query recursively around...

No it doesn't know them. My PC only needs to know one DNS-Server as an entry-point to the DNS. When it utters a recursive request to the DNS my PC does not have to worry about the structure of the DNS and whether there are any root-servers.

[3/ 3 credits]

correct
- 3. Configuring DNS**
  - 3.1.1. Configuring the root-servers
  - 3.1.2. Describe the meaning of all fields of the SOA record! (top part of db.root above)

**Abb. 4.2:** Ansicht des Korrektors beim Korrigieren eines Teilnehmers (Teams). Der Teilnehmer sieht dieselbe Seite nur ohne die Korrekturschaltflächen. Dies ist die Übersichtsseite mit allen Fragen. Die Instruktionen fehlen hier. Eine andere Ansicht zeigt auch die Instruktionen zu den Aufgaben, so wie der Teilnehmer sie zur Verfügung hat.

dass diese stimmt. Den möglichen dritten Versuch hat er also – wie die meisten Teilnehmer – nicht gebraucht.

Erst wenn beide Teammitglieder mit ihrem Account die Multiple-Choice-Fragen fertig beantwortet haben, bekommen sie Zugriff auf den Praxisteil.

Für die praktischen Aufgaben hat jedes Team einen reservierten Zeitslot<sup>5</sup>. Das Team trifft sich an diesem Tag im Praktikumslabor und arbeitet den Praxisteil gemeinsam durch. Je nachdem, wie viele Schwierigkeiten es mit dem Versuch hat, braucht es mal kürzer und mal deutlich länger für einen kompletten Versuch. Deshalb habe ich dieses Jahr zum ersten Mal ganze Tage vergeben. Ein Team hat also einen vollen Tag das gesamte Testbett für sich<sup>6</sup>. Dies ist ein Fortschritt zum letzten Semester, in dem wir Vier-Stunden-Blöcke verteilt hatten, da die Teilnehmer oft nicht innerhalb ihrer vier Stunden fertig wurden und dann an einem anderen Tag noch einmal komplett aufbauen und konfigurieren mussten. Die Teams können sich ihre Zeit damit auch flexibler einteilen und längere Pausen machen oder als Unterbrechung andere Veranstaltungen besuchen. Zu einer festgelegten Zeit des Tages<sup>7</sup> sind die Betreuer anwesend, die bei Problemen helfen können.

Unser Team trifft sich am Dienstag morgen im Labor und beginnt mit dem Setup. Nachdem die Rechner verkabelt und konfiguriert sind, setzen die beiden Teilnehmer die Instruktionen um und beantworten die eingestreuten Fragen direkt an der entsprechenden Stelle auf der Webseite. Das Team hat noch eine Frage, die es mit dem Betreuer klärt und ist dann nach fünf Stunden mit dem Praxisteil fertig. Das System schließt einen Versuch automatisch, wenn dessen Zeit – in der Regel zwei Wochen – abgelaufen ist. Da unser Team aber schon fertig ist, kann es den Versuch gleich schließen. Da die beiden noch etwas nachschauen wollen, lassen sie den Versuch noch offen, korrigieren am übernächsten Tag noch eine

<sup>5</sup>Der Theorieteil sollte entsprechend vor diesem Tag durchgearbeitet sein, was in der Regel auch klappt.

<sup>6</sup>Da wir zwei komplette Testbetten haben, können zwei Teams pro Tag arbeiten.

<sup>7</sup>– normalerweise zwei Stunden lang –



Aufgabe und schließen den Versuch dann.

Die Betreuer und Korrektoren haben schon die ganze Zeit die Möglichkeit, die Lösungen des Teams einzusehen. Erst nach dem Schließen können sie aber korrigieren (vgl. Abb. 4.2), da unser Team erst jetzt keine Änderungen mehr vornehmen kann. Falls unser Team doch noch etwas ändern will können sie ihm den Versuch auch nochmal freischalten<sup>8</sup>. Da die Korrektoren quer korrigieren, warten sie, bis alle Teams geschlossen haben, um dann auf einmal zu korrigieren. Die drei Betreuer Joachim, Andreas und Edgaras haben sich die Aufgaben aufgeteilt und korrigieren jetzt jeder ihren Teil. Unser Teilnehmer kann dabei direkt den Korrekturstatus und auch schon die bereits korrigierten Aufgaben und deren Kommentare einsehen (vgl. Abb. 4.2, Abb. 4.3). Manchmal ist dem Korrektor eine Lösung nicht ganz klar und muss erst besprochen werden. Dazu kann die Aufgabe im System korrigiert, aber noch nicht freigegeben werden. Somit sehen die Teilnehmer die Korrektur noch nicht, sie ist aber schon gespeichert und kann von allen Korrigierenden eingesehen werden.

Die drei Korrektoren werfen noch einen kurzen Blick auf die Theorieteilpunkte, um zu sehen, ob unser Teilnehmer auch die Theorieteilfragen gewissenhaft beantwortet hat:

Da dies der Fall ist, ist die Bearbeitung des Versuchs damit abgeschlossen. Wäre die Theorieteilpunktzahl zu gering, müsste der Korrektor sich die Lösungen unseres Teilnehmers anschauen und ihn dazu befragen, falls beispielsweise bei allen drei Versuchen das Gleiche oder gar Nichts angekreuzt ist.

Entsprechend der Punkteübersicht für den einen Versuch (vgl. Abb. 4.3) gibt es auch eine aggregierte für alle Versuche, in der die Gesamtpunktzahlen dargestellt werden.

Nach diesem Einschub komme ich zurück zu den Änderungen die sich für die verschiedenen Benutzer des Systems ergeben.

### 4.3 Die Teilnehmersicht

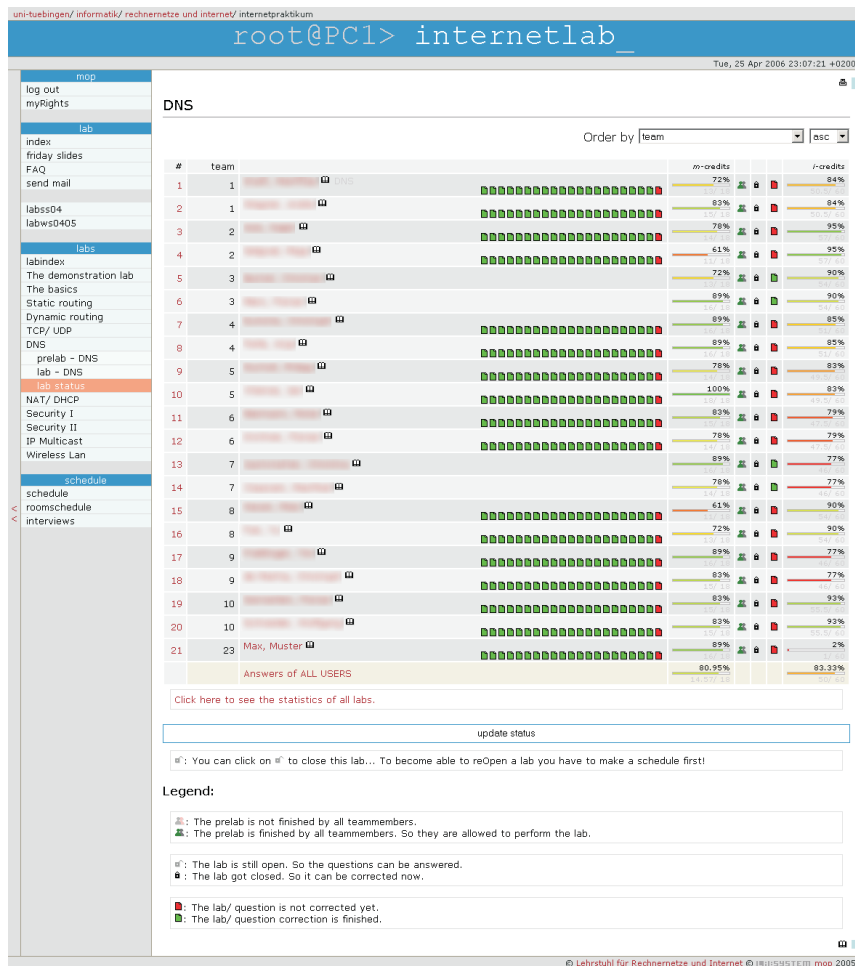
Hauptgrund für die Erstellung des Systems und die Umstellung des Praktikums auf dieses war der Wunsch, das Praktikum für die Teilnehmer effektiver zu gestalten. Deshalb sind diese Hauptangelpunkt.

Da ich selbst das Praktikum vor der Umstellung als Teilnehmer und nach der Umstellung als Betreuer erlebt habe, kann ich nicht vom selben Standpunkt vergleichen. Dies ist allerdings vielleicht sogar ein Vorteil, da ich so weiß, wie es mir selbst erging und wie mir dieselbe Situation jetzt erscheint, wenn ich meine Teilnehmer beobachte.

Wie schon in 1 angeführt, sehe ich beim Theorieteil ein großes Problem im Umfang des Materials und der fehlenden Lenkung innerhalb dieses. Außerdem gibt es keinen Ansporn, das Material aufmerksam durchzuarbeiten, da sich die späteren Aufgaben meist auch mit einem Teil des Wissens oder nachträglichem Nachschauen bearbeiten lassen und damit weniger Zeit gebraucht wird. Soweit die Sicht des Teilnehmers. Aus Sicht des Konzipierenden des Praktikums ist diese Einstellung schlecht. Es können immer nur wenige Teilaspekte innerhalb des Praxisteils berührt werden. Wichtig zum Verständnis und Lernziel sind aber die gesamten Informationen des Theorieteils oder zumindest ein deutlich größerer Teil als derjenige, der später für die praktischen Versuche benötigt wird. Der vorgeschlagene Lösungsansatz mit den *Multiple-Choice-Fragen* (2.2.1) war ein Erfolg. Die Fragen zeigen den Teilnehmern an, worauf sie besonderes Augenmerk zu legen haben. Die Hürde, dass jeder Einzelne die Fragen beantwortet haben muss, um am Praxisteil teilnehmen zu können bewirkt einen Ansporn, den gesamten Theorieteil durchzugehen<sup>9</sup>. Ein unvorbereitetes

<sup>8</sup>Bereits vorgenommene Korrekturen gehen dabei nicht verloren, werden dem Teilnehmer aber verborgen und müssen nach dem erneuten Schließen durch den Korrektor wieder freigeschaltet bzw. entsprechend der neuen Lösung angepasst werden.

<sup>9</sup>Die für alle Teilnehmer einsichtige Liste, wer wann mit dem Theorieteil fertig wurde, spornt dazu an, bei Zeiten fertig zu werden.



**Abb. 4.3:** Punkteübersicht während der Korrektur. Die bunten Seiten links geben den Korrekturstatus der einzelnen Aufgaben wieder. Die Punktebalken spiegeln die Punktzahl des Theorie- und Praxisteils grafisch wieder. Die Symbole dazwischen zeigen an, ob das gesamte Team den Theorieteil abgeschlossen hat, der Praxistil geschlossen und die Korrektur abgeschlossen wurden.

Erscheinen wird durch das System praktisch unmöglich gemacht. Natürlich könnten die Teilnehmer einfach alle Fragen wegeklicken, dies fällt dann aber den Korrektoren auf. Im Gegensatz zu vielen anderen Praktika, sind die Teilnehmer bei uns in der Regel ausreichend vorbereitet für den praktischen Teil und zwar jedes einzelne Teammitglied. Dies ist sicher auch ein Grund, warum die Teilnehmer dieses Praktikum als besonders positiv erleben: Alle haben ausreichend Vorwissen, um gemeinsam durch den Versuch zu kommen und spätestens hinterher hat jeder Teilnehmer verstanden, was es mit der Lösung auf sich hat. Ohne diesen Vorbereitungszwang<sup>10</sup> wäre dies nicht so, weil die meisten Studenten einfach zu faul wären sich ordentlich vorzubereiten.

Die Verwendung des Internet als zugrunde liegende Plattform hat dazu geführt, dass die Teilnehmer dieses deutlich verstärkter zur Recherche nutzen. Sie finden damit beiläufig sowohl weiterführende Informationen als auch heraus, welche Quellen es gibt<sup>11</sup> und welche davon wie verlässlich sind. Da das Internet zur Zeit eine der wichtigsten, wenn nicht die wichtigste Informationsquelle – mindestens im Bereich Informatik – ist, ist das *Wissen um verlässliche Informationsquellen* wertvoll.

Der Praxisteil hat durch die Umstellung auf das neue System deutlich an *Klarheit* gewonnen. Wenn ich mit unserem Betreuungsaufwand von früher vergleiche, so ist dieser heute nahe bei Null. Die oben (4.1) schon angesprochene Aufteilung des Versuchs auf mehrere Seiten und Sinneinheiten kombiniert mit separierten Aufgabenteilen<sup>12</sup> versetzt die Teilnehmer in die Lage, *autonom arbeiten* zu können. Fragen, die sich ergeben, beziehen sich nicht mehr auf das, *was* zu tun ist, sondern auf unerwartete Probleme<sup>13</sup>. Diese treten auf und sind auch von den Betreuern nicht immer zu lösen. Dazu braucht es einfach Erfahrung, die man nach mehreren Jahren in der Materie zu einem gewissen Maße hat. Gerade solche Probleme haben manchmal den größten Lerneffekt: Die Durchführung des Versuches wird zwar mehr oder weniger kurz unterbrochen, aber gerade die Fehlerlokalisierung ist eine der Aufgaben, vor die man im Beruf immer wieder gestellt werden wird.

Die Bearbeitungszeit von vor der Umstellung auf die Online-Plattform und danach lässt sich nicht direkt vergleichen, da wir die Aufgaben grundlegend erneuert haben. Aus meinem eigenen Erleben kann ich sagen, dass das Ziel, *auf die Inhalte zu fokussieren* voll erreicht wurde. Die Teilnehmer haben nach ihrer Versuchsdurchführung keinen Aufwand mehr, aus den Notizen ein Protokoll zu formen, da die Antworten schon das Protokoll sind<sup>14</sup>. Damit lässt sich auch eine etwas längere Arbeitszeit im Praktikumlabor, wie sie bei uns der Fall ist, mindestens teilweise rechtfertigen. Eine Nachbearbeitungsphase zur Auswertung von Messergebnissen, wie beispielsweise in der Physik, gibt es bei uns nicht. Dennoch arbeiten die Teilnehmer auch nach dem Versuch zuhause noch an den Aufgaben. Auch bei vorhandener Nachbearbeitungsphase lässt sich ein Online-System also sinnvoll nutzen.

Dadurch, dass die Aufgaben innerhalb des Systems korrigiert werden, können die Teilnehmer die Korrekturen von einzelnen Aufgaben schon einsehen, bevor der gesamte Versuch

---

<sup>10</sup>, der aufgrund der technischen Vorkehrung durch das System und damit nicht durch eine Person (sozialdruck) ausgeübt wird,

<sup>11</sup>Meinen ersten RFC habe ich beispielsweise im Rahmen des Internetlabs gelesen.

<sup>12</sup>Natürlich ist auch eine klare Aufgabenstellung hier anzuführen. Bis zu dieser hat es bei uns ungefähr drei Iterationen/ Semester gedauert. Die meisten Änderungen ergaben sich allerdings beim ersten Durchlauf.

<sup>13</sup>Man hat zwar alles richtig gemacht, doch es tut nicht, weil irgendein Faktor Einfluss nimmt, der normalerweise vielleicht nicht auftritt.

<sup>14</sup>Ich möchte auch hier nochmal erwähnen, dass dadurch das Wissen, wie ein Versuchsprotokoll auszu-sehen hat nicht unbedingt erworben wird. In unserem Fall waren die Abgaben aber von Anfang an keine Protokolle in dem Sinne, wie sie beispielsweise bei chemischen Versuchen abzugeben sind. Falls es darauf ankommt, zu lernen, wie ein ordentliches Protokoll aussieht, kann man auch die Felder entsprechend anordnen oder diese auf einer abschließenden Übersichtsseite entsprechend präsentieren. Die neue Präsentationsform steht dem also nicht im Wege.

korrigiert ist<sup>15</sup>. Dies ist insofern interessant, dass die *Korrektur* lerntechnisch am meisten bewirkt, wenn die *zeitnah* zum Versuch erfolgt, da die Teilnehmer dann noch in Erinnerung haben, was sie überhaupt gemacht haben.

## 4.4 Die Betreuung

Durch die sich aus dem neuen System ergebenden Möglichkeiten zur klaren Strukturierung und Aufteilung der Versuche ist der Betreuungsaufwand deutlich zurückgegangen. Mittlerweile<sup>16</sup> ist es zum ersten Mal so, dass viele Teams *gänzlich ohne Rückfragen* die Versuche bearbeiten. Dies liegt natürlich auch daran, dass wir die Aufgaben im Laufe der letzten Semester optimiert haben. Bei der Umstellung auf das neue System zum Sommersemester 2005 war der Rückgang aber signifikant.

Einen großen Nutzen für die Betreuer bieten die *Musterlösungen*. Mit ihrer Hilfe kann sich schnell in den Sachverhalt eingearbeitet werden: Der Betreuer öffnet einfach die Stelle, an der es Probleme gibt in seinem Account und bekommt die Lösung angezeigt<sup>17</sup>. Eine Möglichkeit, die wir bisher noch wenig nutzen, sind *Hinweise* innerhalb der Versuche, die *nur den Betreuern* angezeigt werden. Diese bieten sich an Stellen an, die sich für die Teilnehmer als Problemstellen herausgestellt haben<sup>18</sup>.

## 4.5 Die Korrektur

Der Korrekturaufwand ist sehr zurückgegangen. In dem Jahr, in dem ich Teilnehmer war, gab es individuelle Abgaben pro Team ohne festgelegte Form. Die Korrektur einer solchen Abgabe dauert sehr lange und ist schwierig, da man als Korrektor ständig vor Augen haben muss, wo im Versuch sich der Antwortende gerade befindet. Verbessern lässt sich dies, indem man wenigstens die Form vorgibt. Noch besser ist es, wenn man dem Teilnehmer gleich einen Instruktionstext mit vorgefertigten Freiräumen für die Antworten gibt. Dann ist man schon fast bei der Situation des Labsystems.

Im jetzigen System ist die *Ansicht des Versuchs für Korrektoren und Teilnehmer dieselbe*. Die Korrektoren können sich zusätzlich die *Musterlösung* einblenden (vgl. Abb. 4.3). Sie sehen so zu jeder Zeit genau das, was auch die Teilnehmer zur Bearbeitung der Aufgabe vor sich hatten – ein großer Vorteil, müssen sie jetzt nicht mehr nachschauen, worum es bei der Frage eigentlich geht<sup>19</sup>.

Die Online-Abgabe ist *getippt*. Als langjähriger Korrektor von handschriftlichen Abgaben kann ich beurteilen, dass es mitunter sehr zeitaufwändig ist, die Handschrift eines Menschen zu lesen.

Da wir mehrere Korrektoren haben, korrigieren diese aus Gründen der Gerechtigkeit<sup>20</sup> und nicht zuletzt der Effizienz<sup>21</sup> quer. Das heißt Korrektor A Aufgaben 1-5 aller Teams, Korrektor B 6-10 etc. Das System bietet dazu die Möglichkeit, *die Antworten aller Teams auf einmal* anzuzeigen. Der Korrektor kann so bequem alle Teams zu einer Aufgabe korrigieren (vgl. Abb. 2.2).

Vor allem die letzten beiden Punkte haben die Korrektur sehr beschleunigt.

<sup>15</sup>Unsere drei Betreuer korrigieren quer.

<sup>16</sup>Wintersemester 2005/ 2006

<sup>17</sup>In der Praxis haben wir im Betreuerraum einen Ausdruck des Versuches mit Musterlösung, da man diesen mit zu den Rechnern der Teilnehmer nehmen kann, ohne diesen die Musterlösung auf dem Bildschirm zeigen zu müssen.

<sup>18</sup>Wenn solche Problemstellen keinen Lerneffekt haben, sollte die Aufgabe verändert werden!

<sup>19</sup>Bei Bedarf können Teilnehmer wie Korrektoren eine Seite aufrufen, die lediglich die Fragen und Antworten ohne Instruktionstext enthält.

<sup>20</sup>Eine andere Möglichkeit wäre, die Teams unter den Betreuern jedes Mal zu tauschen.

<sup>21</sup>Es geht deutlich schneller, eine Aufgabe bei allen Teams vergleichend zu korrigieren, als jedes Mal bei einem Team von vorne mit dem Versuch anzufangen.

## 4.6 Punkteverwaltung

Bei der „manuellen Korrektur“ müssen nach der Korrektur noch die Punkte übertragen werden, da das Protokoll zumeist zurückgegeben wird. Sinnvoller Weise sind die Punkte aller Aufgaben zu übertragen. Im Idealfall hat der Teilnehmer Einsicht in seine Punkte, um zu sehen, wie er steht.

Bei uns entfällt dieser Schritt natürlich. Das System stellt Teilnehmern und Korrektoren *diverse Übersichtseiten* mit ihren Punkten zur Verfügung (vgl. Abb. 4.3). Hier hat sich besonders das in 3.1.5 angesprochene *Farbsystem* bewährt. Unsere Teilnehmer müssen momentan 75% der möglichen Punkte erreichen, um zu bestehen. Das System zeigt durch die Färbung des Punktebalkens an, ob dieses Kriterium in Gefahr ist. Ein Blick auf die Statistik zeigt, welche Teams im kritischen roten Bereich sind (vgl. Abb. 4.3).



## 5. Zusammenfassung und Ausblick

Die Arbeit hat gezeigt, dass die *Umstellung eines Praktikums* von konventioneller Form auf ein eLearningsystem *viele Vorteile* bringt. Sie hat dazu zuerst *allgemeine Veränderungen* angeführt, die zu einer Verbesserung des Praktikums vor allem (aber nicht nur) für die Lernenden beitragen können. Diese können als *Kriterien zur Auswahl einer geeigneten Lernplattform* dienen. Sie waren die *Anforderungen an das Labsystem. Schlüsselentscheidungen* bei der *Konzeption* dieses neuen eLearning-Systems wurden im nächsten Teil aufgezeigt. Kapitel vier stellte die *Vorteile*, die sich für das *Internetpraktikum* durch die Umstellung ergeben haben, heraus.

Vielen Praktika steht der Wandel und damit einhergehende Wechsel in ein neues Medium noch bevor. Die Arbeit hat *einen möglichen Weg* gezeigt und soll auch dazu ermutigen, den Schritt zu wagen. Der Aufwand, der durch die Umstellung entsteht, wird in den Folgejahren wieder ausgeglichen<sup>1</sup> und die *Lehre stark verbessert*:

THIS COURSE WAS EXCELLENT...<sup>2</sup>

---

<sup>1</sup>Der Betreuungs- und Korrekturaufwand geht deutlich zurück. Durch Einsparung von entsprechenden Hiwis in den Folgejahren lassen sich auch die Umstellungskosten amortisieren.

<sup>2</sup>Abschlusskommentar eines Teams im Sommersemester 2005





# Literatur

- [BaHMH02] Peter Baumgartner, Hartmut Häfele und Kornelia Maier-Häfele. *E-Learning Praxishandbuch; Auswahl von Lernplattformen; Marktübersicht - Funktionen - Fachbegriffe*. StudienVerlag. 2002.
- [bild] <http://www.bildungserver.de/zeigen.html?seite=1571>, Vergleich von Web-systemen.
- [colm] <http://colmar.informatik.uni-tuebingen.de>, Webportal des Internetpraktikums.
- [Hani03] Frank Hanisch. *Evolution der neuen Medien in der Lehre*, Kapitel 7, S. 27–43. WSI. Editor: Marc-Oliver Pahl, 2003.
- [ilia] <http://www.ilias.de>, Das Ilias-Projekt.
- [KlWa04] Bernd Kleimann und Klaus Wannemacher. *E-Learning an deutschen Hochschulen, Von der Projektentwicklung zur nachhaltigen Implementierung*. HIS GmbH. 2004.
- [labs] <http://labsystem.m-o-p.de>, Das Labsystem.
- [LiZa04] Jörg Liebeherr und Magda El Zarki. *Mastering Networks: An Internet Lab Manual*. Press. 2004.
- [Münz05] Stefan Münz. *SelfHTML 8.1*. Kiel. 2005.
- [mood] <http://moodle.org>, Das Moodle-Projekt.
- [olat] <http://www.olat.org>, Das Olat-Projekt.
- [riwe] <http://net.informatik.uni-tuebingen.de>, Webseite des Lehrstuhls für Rechner-netze und Internet.
- [Schm04] Egon Schmid. *PHP Handbuch*. PHP Documentation Group. 2004.
- [SpKl01] Michael Sperber und Herbert Klaeren. *Vom Problem zum Programm. Architektur und Bedeutung von Computerprogrammen., 3. Auflage*. Teubner. 2001.
- [valia] <http://jigsaw.w3.org>, CSS-Validator des W3C.
- [valib] <http://validator.w3.org>, HTML-Validator des W3C.
- [vite] <http://www.vitels.ch>, Webseite des Vitels-Verbundes.
- [webc] <http://www.webct.com>, Webseite des kommerziellen WebCT-Programms.