Spezifikation und Implementation eines sicheren Lernerfolgskontrollmoduls für CSCL - Werkzeuge

Diplomarbeit

im Fach Naturwissenschaftliche Informatik

vorgelegt von

Andre Döring[1]

an der Technischen Fakultät der Universität Bielefeld

Betreuung durch

Prof. Peter B. Ladkin PhD FBCS

15. Oktober 2003

[1]andre@rvs.uni-bielefeld.de

Inhaltsverzeichnis

Zusammenfassung

Ziel der Arbeit ist die Implementation und Integration eines Softwaremoduls zur Lernerfolgskontrolle in ein existierendes eLearning-Werkzeug. Für dieses Modul habe ich die Form des Multiple-Choice-Tests gewählt. Die Entwicklung eines solchen Moduls wirft jedoch im Vorfeld einige Fragen auf.

Zunächst ist zu klären, ob und wie ein solcher Multiple-Choice-Test aus didaktischer Sicht in ein eLearning-System integrierbar ist. Ebenso muss betrachtet werden, welche Anforderungen potentielle Benutzer von eLearning-Systemen an ein solches Modul stellen.

eLearning-Systeme sind netzwerkbasierte Softwaresysteme, und daher ist aus meiner Sicht zusätzlich zur Anforderungsanalyse zu überprüfen, welche Sicherheitsmechanismen in mein Modul integriert werden müssen, damit es innerhalb eines eLearning-Werkzeugs nicht zum Sicherheitsrisiko wird.

Da in der Literatur keine Analysen über Sicherheitprobleme von eLearning-Systemen zu finden waren, nehme ich dieses zum Anlass vorab eine generelle Analyse zu Sicherheitproblemen von eLearning-Systemen durchzuführen. Die Ergebnisse dieser Sicherheitsanalyse nutze ich dann als Basis für die spezielle Analyse der Sicherheitsprobleme meines Softwaretools.

Die Arbeit gliedert sich in zwei Teilbereiche. In Teil I. stelle ich zunächst das in der Arbeit betrachtete eLearning-Konzept vor: das kooperative computergestützte Lernen (CSCL). Dann werde ich untersuchen, in welcher Form Multiple-Choice-Tests didaktisch sinnvoll in CSCL-Systeme zur Lern- bzw. Leistungskontrolle integriert werden können. Als letztes betrachte ich, welche Sicherheitsprobleme für CSCL-Systeme existieren und welche Sicherheitsmechanismen zum eindämmen dieser Probleme notwendig sind.

Teil II. setzt die im ersten Teil gewonnenen Erkenntnisse in die Praxis um. Beschrieben wird die Spezifikation des Multiple-Choice-Test-Moduls zur Lernerfolgskontrolle innerhalb des CSCL-Systems. Diese Spezifikation wird ebenfalls einer Sicherheitsanalyse unterzogen. Diese basiert auf den Ergebnissen der allgemeinen Sicherheitsanalyse für CSCL-Systeme und der Spezifikation des Moduls. Die Implementation vereint die Analyseergebnisse und die Spezifikation des Moduls zu einem Softwaresystem.

Somit ist das Ziel der Arbeit erreicht: Die Spezifikation und Entwicklung eines sicheren Lernerfolgskontrollmoduls für CSCL - Werkzeuge.

Teil I

CSCL: Bestandsaufnahme und Sicherheitsanalyse

Kapitel 1

Einleitung

Die Präsentation von Lehrinhalten in der universitären Lehre beschränkt sich heutzutage nicht mehr nur auf das durchführen von Vorlesungen und Seminaren. Vielmehr wird versucht Konzepte zu entwickeln, Lehrinhalte auf computergestützte Lernumgebungen, sogenannte eLearning-Systeme zu übertragen. Hierzu werden Lehrinhalte nach didaktischen Maßstäben multimedial aufbereitet und durch Multimediaanwendungen präsentiert.

Das in dieser Arbeit behandelte spezielle eLearning-Konzept ist das Konzept des computergestützten kooperativen Lernens[1]. Beim CSCL-Ansatz arbeitet der Student nicht ausschließlich alleine mit dem Computer. Vielmehr steht der aktive Diskurs mit Kommilitonen und Dozenten im Zentrum. Gruppenarbeit wird durch Computertools unterstützt.

Die Rolle des Computers liegt daher neben der Bereitstellung von Lehrinhalten hauptsächlich in der Unterstützung der Organisation des Lernprozesses und der Gruppenarbeit.

CSCL ist ein interdisziplinäres Forschungsgebiet bei dem aktive Zusammenarbeit zwischen Pädagogik und Informatik stattfinden muss. Die Pädagogik liefert die Lehrmethoden und die Informatik setzt diese mit aktuellen Multimediatechniken auf dem Bildschirm um.

Allerdings werden bei der Begeisterung für diesen relativ neuen Zweig des Computerlernens einige negative Aspekte vernachlässigt. Neben fehlender einheitlicher Standards für die Umsetzung von CSCL-Projekten nimmt auch der Aspekt Computer- bzw. Datensicherheit eine eher sekundäre Rolle ein. Für CSCL-Systeme gibt es bisher kaum Forderungen für Sicherheitsstandards bzw. wurden bisher Sicherheitsstandards umgesetzt. Offensichtlich wird bei der Entwicklung übersehen, dass auch CSCL-Anwendungen sicherheitskritische sensible Daten verwalten, die zu Angriffszielen von Crackern werden können.

Deshalb und im Hinblick auf die Implementation des Testmoduls ist die Computersicherheit im Allgemeinen und speziell die Computersicherheit in CSCL-Werkzeugen der eine zentrale Aspekt des ersten Teils meiner Arbeit. Die Andere zentrale Frage dieses Teils ist, inwiefern Lernerfolgs- oder Leistungskontrolltests

[1] engl. Computer Supported Cooperative/Collaborative Learning (CSCL)

mit didaktischem Mehrwert in CSCL-Systeme integriert werden können.

Gliederung Teil I.

In Kapitel 2 auf der nächsten Seite werde ich allgemein erläutern, was unter einem CSCL-Werkzeug zu verstehen ist, die aktuellen Sicherheitsstandards von CSCL-Werkzeugen kritisch betrachten, und weiterhin beschreiben, welche Form von Lern- und Leistungskontrolle in einem CSCL-Werzeug sinnvoll erscheint. Kapitel 3 auf Seite 14 dient als kurze Einführung in wichtige Termini der Computersicherheit und als Beschreibung der von mir zur Analyse benutzen Methoden. In Kapitel 4 auf Seite 24 werde ich dann mit diesen Methoden analysieren, welche Sicherheitsgefahren durch Crackerangriffe auf CSCL-Werkzeuge zu erwarten sind und welche Maßnahmen zur Steigerung der Sicherheit sinnvoll erscheinen.

Kapitel 2

Kooperatives Lernen und Lernerfolgskontrolle

2.1 Computer Supported Cooperative/Collaborative Learning (CSCL)

Bisherige Erfahrungen im Einsatz von computergestützter Lehr/-Lernsoftware, wie z.B. programmierte Lehrbücher, intelligente tutorielle Systeme, haben gezeigt, dass herkömmliches computergestütztes Lehren und Lernen nur geringe Effektivität im Lernerfolg aufweist [Schulmeister 2001].

Deshalb geht die aktuelle Forschung im Bereich des computergestützten Lernens in Richtung computergestützter kooperativer Lehr-/Lernsysteme. Dieser Ansatz wird als CSCL (Computer Supported Cooperative/Collaborative Learning) bezeichnet. Als Grundannahme dieser Forschungsrichtung wird verstanden, die Motivation und Effektivität des Lernprozesses durch die Einbeziehung von Studenten und Dozenten in den Lernprozess zu steigern.

Diese Erkenntnis geht konform mit den Ergebnissen der pädagogischen Forschung, wonach kooperative Lernformen individuellem Lernen oft überlegen sind. Passives verfolgen einer Vorlesung oder isoliertes lesen eines Fachbuches ziehen (jedenfalls für die meisten Lerner) einen geringeren Lernerfolg nach sich, als aktives diskutieren und erarbeiten von Lerninhalten in betreuten oder unbetreuten Lerngruppen [Schulmeister 2001].

2.1.1 Verteilte Wissenskommunikation statt isoliertem Lernen

Die theoretische Basis des CSCL-Ansatzes bildet eine konstruktivistische Sichtweise, sowie Situiertheit beim Lernenprozess und lernen auf Basis verteilter Kognition [Schulmeister 2001].

Der Kontruktivismus beschreibt Wissen nicht als Referenzen auf Weltobjekte, sondern Wissen wird als Ergebnis eines kognitiven Konstruktionsprozesses gesehen [Schulmeister 2001]. Es wird dynamisch generiert und nicht fest gespeichert

[Schulmeister 2002]. Demnach unterstützt gerade der aktive Diskurs in Lerngruppen die Bildung und Festigung von Wissen. Das "eintrichtern" von Wissen (vgl. Nürnberger Trichtermodell) ist daher, entsprechend dieser Sichtweise, nicht mehr aktuell.

Menschliches Handeln ist grundsätzlich eingebettet in soziale Kontexte und nicht das Resultat isolierter Entscheidungs- und Verarbeitungsprozesse eines Individuums [Kerres 2001]. Ebenso ist Wissenserwerb gerade ein aktiver und handlungsorientierter Prozess der beeinflusst wird durch den sozialen Kontext. Demnach ist es sinnvoll, Lernen in einen *situierten* Kontext einzubetten. Lernaufgaben sollen authentisch und realitätsnah sein. In dieser Form gelerntes kann daraufhin besser und leichter angewendet und auf neue Situationen übertragen werden [Schulmeister 2001].

Wie der Begriff "kooperativ" sagt, und wie bisher herausgearbeitet, ist das lernen in Gruppen ein zentraler Aspekt in CSCL. Allerdings ist beim Gruppenlernen das Wissen in der Gruppe unterschiedlich verteilt. Nicht jeder Gruppenteilnehmer hat das gleiche Wissen über das Thema wie ein Anderer. Um eine Kommunikation innerhalb der Gruppe über einen Themenbereich zu ermöglichen, müssen die Lernenden einen geeigneten Repräsenationmechanismus etablieren (z.B. Diagramme, Tabellen, Texte, etc.), der dann in das CSCL-System eingefügt werden kann. Der hierfür notwendige Entscheidungsprozess erfordert Team- und Diskussionfähigkeit innerhalb der Gruppe. CSCL-Systeme können somit die Bildung und das Management von verteiltem Wissen [Schulmeister 2001] und die Fähigkeit zu Teamarbeit fördern.

2.1.2 CSCL – Interdisziplinäre Anforderungen

Die Realisierung von CSCL-Systemen stellt unterschiedliche Anforderungen an die beteiligten Teildisziplinen. Maßgeblich sind Erkenntnisse der Informatik und der pädagogischen Didaktik.

Informatische Sichtweise

Die Informatik ermöglicht den Teil, welcher im allgemeinen als *delivery system* oder *Bildungsmittel* bezeichnet wird [Kerres 2001]. Es handelt sich dabei um jene Systemteile, die für den technischen Ablauf des Lehr-/Lernsystems notwendig sind. Mit Hilfe dieser Technologie können die technischen Anforderungen der Systemdesigner und Benutzer an ein CSCL realisiert werden. Denkbare Anforderungen sind nach [Schulmeister 2001]:

- gemeinsame multimediale Arbeitsbereiche (shared workspaces) für gemeinsame Objekte einer Gruppe
- Programme zur synchronen und asynchronen Kommunikation (z.B. Chat, Foren, etc.) und Bearbeitung von gemeinsamen Objekten innerhalb von Arbeitssitzungen (Sessions)

- verschiedene Sichten (Views) auf Arbeitbereiche, welche das Maß der Kooperation bestimmen: von individuellen bis zu eng gekoppelten Sichten
- verschiedene Kooperationsmodi, welche Sichten und Zugriffsrechte auf Daten kombinieren
- Arbeitsprozessunterstützung
- ein gut benutzbares Mensch-Maschine-Interface

Realisiert werden solche Bildungsmittel mit den sogenannten *Neuen Medien*. Diese beinhalten Technologien wie Rechnernetze und verteilte Systeme sowie alle Arten von Multimediatechnologien[1] (Internettechnolgien, Audio/-Videotechnologien, Bildverarbeitung etc.).

Pädagogische Didaktik

Wie beschrieben spielt das Lernen in Gruppen bei CSCL eine wichtige Rolle. Der Lernprozess soll konstruktivistisch und situiert sein. Die Pädagogik untersucht und liefert Methoden zur besseren Durchführung von kollektivem Lernen. Ebenso untersucht sie, wie Computer das Problem der eingeschränkten sozialen und größtenteils nonverbalen Kommunikation im Rahmen von CSCL verbessern können [Schulmeister 2001].

Desweiteren ist aus pädagogischer Sicht interessant, in welcher Form Lehrende Einfluss auf den Lernprozess nehmen können und in welcher Weise dieser von ihnen mit Hilfe didaktisch aufbereiteter Computerlösungen steuerbar ist [Kerres 2001]; z.B. in Form von webbasierter Lernumgebungen oder Online-Tutorien.

Ein weiterer wichtiger Forschungsschwerpunkt ist das didaktische Design der *Bildungsmedien*, also der Inhalte, die über die Bildungsmittel zugänglich sind. Die Vergangenheit zeigte, dass sich gerade der didaktische Wert von Bildungsmedien auf simple und einfallslose didaktische Konstruktionen beschränkte, deren Nutzen für den Lernerfolg in Frage zu stellen ist [Schulmeister 2001, Kerres 2001].

2.1.3 Lernerfolgskontrolle und Leistungskontrolle

Neben dem aktiven erarbeiten von Lerninhalten in der Gruppe, wird der Lernprozess ebenso von Phasen des individuellen Erarbeitens von Lerninhalten geprägt. In diesen Phasen, beispielsweise der Bearbeitung einer Vorlesung, sollte dem Lerner die Möglichkeit gegeben werden sein Wissen zu überprüfen.

Wie [Kerres 2001] anfügt, nehmen Lernerfolgskontrollen immer weniger Stellenwert ein, da sie mit hohen Kosten und didaktischem Aufwand verbunden sind. Als komfortable Realisierung von Lernerfolgskontrolle sieht er jedoch den Multiple-Choice-Test (siehe Kapitel 2.1.4 auf Seite 9).

[1] Definitionsversuche von Multimedia vgl. [Schulmeister 2002]

Der Nutzen von Lernerfolgskontrollen für den Lernenden und deren Aufbau erklärt sich aus motivationspsychologischer Sicht [Kerres 2001]:

- Lernerfolgskontrolle gilt als Bekräftigung für den Lerner weiterzuüben
- Lernerfolgskontrolle testet Verständnis von Sachverhalten und fördert die Anwendung abstrakter Konzepte an konkreten Beispielaufgaben
- Voraussetzung für diese positiven Effekte ist eine fundierte didaktische Planung der Lernerfolgskontrolle

Innerhalb situierter Ansätze, wie CSCL, war vormals die Nützlichkeit von Lernerfolgskontrollen in Frage gestellt. Mittlerweile hat sich die Meinung dahingehend geändert, dass Lernerfolgskontrollen als Unterstützung für den Lerner im Lernprozess bereitstehen sollten.

> "Sie sollen verhindern, dass Lerner *meinen* etwas verstanden zu haben, was sie tatsächlich nur flüchtig gelesen haben; sie sollen dazu beitragen das Lerner sich Gedanken machen über Lerninhalte, und dass die Inhalte aktiv wiedergegeben werden." ([Kerres 2001], S.206)

Andererseits soll als Kritik angemerkt werden, dass ausschließlich Drill&Practice orientierte Frage-und-Antwort-Schemata als einengende Kontrolle im Lernprozess angesehen werden können [Kerres 2001, Schulmeister 2002]. Daraus ergeben sich folgende Paradigmen [Kerres 2001]:

- Lernerfolgskontrollen sind ein Mittel zur Selbstkontrolle und demnach nicht als verpflichtendes Lernmodul in CSCL-Systeme integriert
- Lernerfolgskontrollen enthalten nicht nur reine Ergebnisinformationen, sondern bieten dem Lerner eine zusätzliche Hilfestellung an
- Fehler werden unmittelbar korrigiert bzw. eine Wiederholung falscher Fragen wird angeboten

Für ein Lernkontrollmodul innerhalb eines CSCL-Werkzeugs sehe ich zwei Hauptanwendungsfälle. Zum einen als (kritisch zu sehende) Leistungsüberprüfung innerhalb von Tutorien oder schriftlicher Tests. Zum anderen als freiwillige Überprüfung des Wissens und als Ergänzung zur Gruppenarbeit.

Es ergeben sich demnach verschiedene grundsätzliche Anforderungen an ein Lernerfolgskontrollmodul innerhalb von CSCL-Werkzeugen:

- es ist ein autonomes Modul innerhalb des CSCL-Werkzeugs; deshalb ist es jederzeit flexibel einsetzbar, aber es besteht kein Zwang zu dessen Nutzung
- es besitzt didaktischen Nutzen, beispielsweise in Form erweiterter Hilfestellung nach inkorrekter Beantwortung einer Frage

- es enthält ein einfaches Management für die Bereitstellung von Fragen und Tests seitens der Dozenten, um die Benutzung des Moduls zu vereinfachen und seine Akzeptanz zu erhöhen
- wenn das Modul zu studienrelevanter Leistungsüberprüfung genutzt wird, weist es Sicherheitsmerkmale auf, um Testdaten, Ergebnisse und sonstige sensible Daten vor unberechtigtem Zugriff zu schützen

2.1.4 Computer Multiple-Choice-Tests

Multiple-Choice-Tests werden benutzt um verschiedene Formen allgemeiner Tests wie z.B. Intelligenztests, Persönlichkeitstests und Wissenstests anzulegen. Sie basieren auf dem Drill&Practice-Prinzip des Behaviorismus [Schulmeister 2001]. Auf eine richtig oder falsch beantwortete Frage folgt eine entsprechende Reaktion in Form von beispielsweise Lob oder Restriktion.

Aufbau

Ein Multiple-Choice-Test am Computer besteht aus einer variablen Anzahl von Fragen mit n-Antwortalternativen. Die Anzahl der Alternativen variiert in der Regel zwischen drei und fünf, wobei drei optimal sind [Bortz & Döring 2002]. Die Anzahl der Antwortalternativen sollte sich im Laufe des Tests nicht ändern.

Es existiert in der Regel eine richtige Antwortalternative (1-aus-n-Format). Um den Schwierigkeitsgrad zu steigern, können auch mehrere verknüpfte Antwortalternativen richtig sein (m-aus-n-Format). Diese müssen dann exakt bestimmt werden.

Es ist nicht zwingend, Antwortalternativen als Sätze zu formulieren. Denkbar sind auch grafische Lösungen, beispielweise durch die Vorgabe einer Grafik in der per Drag&Drop etwas manipuliert werden kann. Oder eines Schiebereglers, der in die korrekte Position gebracht werden muss [Kerres 2001]. Auch Umordnungs- oder Zuordnugsaufgaben sind möglich.

Nach einer Antwort folgt die Reaktion des Systems. Ist sie richtig, folgt die nächste Frage und z.B. ein Lob oder weiterführende Information zur gestellten Frage; ist sie falsch bestehen mehrere Möglichkeiten der Reaktion:

- dem Benutzer wird mitgeteilt, dass die Antwort falsch war und die nächste Frage wird gestellt
- es wird zusätzlich erklärt, warum die Antwort falsch war und die richtige Lösung erläutert
- der Benutzer erhält die Möglichkeit zur Wiederholung der Frage
- dem Benutzer werden zusätzlich Tipps anzeigt die eine Wiederholung der Frage erleichtern, z.B. Literaturhinweise etc.
- am Ende des Tests erhält der Benutzer nochmals die Möglichkeit, alle falsch beantworteten Fragen zu wiederholen

Testfairness

Bedingt durch die starre Form sind Multiple-Choice-Tests stark abhängig von der Tagesform der Benutzer, da der Schwierigkeitsgrad des Tests sich nicht an z.B. schlechte Konzentrationsfähigkeit anpassen kann, wie das innerhalb eines mündlichen Gespräches der Fall wäre. Unter schlechten Voraussetzungen können ebenso schlecht bzw. unklar formulierte Fragen und Antwortalternativen zu einer Verzerrung des Ergebnisses führen.

Deshalb sollen beim Testdesign folgende Regeln beachtet werden: Die Fragen, sogenannte Testitems, sollen so formuliert werden, dass sie für einen uniformierten Benutzer eindeutig verständlich sind und mit gleicher Wahrscheinlichkeit für richtig gehalten werden. Erfüllen Antwortalternativen diese Voraussetzungen heißen sie *gute Distraktoren* [Bortz & Döring 2002].

Um einen ausgewogenen Test zu designen, sollen die Testitems in verschiedene Schwierigkeitstufen zwischen 0.0 (Schwierigkeit 0%) und 1.0 (Schwierigkeit 100%) klassifiziert werden. Ein durchschnittlicher Test hat eine Schwierigkeit von ca. 0.6 also 60% [Bortz & Döring 2002].

Um einen Überblick über Testtheorie und der Gewinnung von guten Distraktoren und deren Schwierigkeitsgrad zu gewinnen, verweise ich auf die einschlägige Literatur (z.B. [Bortz & Döring 2002, Uni Minnesota 2003, Parks]).

Kritik

Aufgrund ihres schematischen und starren Aufbaus, sind Multiple-Choice-Tests sehr komfortabel und ökonomisch am Computer auswertbar [Kerres 2001, Bortz & Döring 2002].

Demgegenüber stehen einige Nachteile: Multiple-Choice-Tests fordern von Benutzern schlichte Wiedererkennungsleistungen. Ebenso ist es möglich gute Ergebnisse durch raten zu erlangen [Bortz & Döring 2002].

Daher sind Multiple-Choice-Tests in der Pädagogik oftmals kritisiert worden. Sie gelten als veraltetes Konzept mit fraglichem didaktischen Nutzen [Schulmeister 2001]. [Kerres 2001] allerdings hält eine grundsätzliche Ablehnung für unangemessen. Neben der schlichten Überprüfung von Wissen lassen sich auch Tests zur Überprüfung von Fertigkeiten konstruieren. Hierbei ist allerdings der Konstruktionsaufwand erheblich höher.

Ich denke, dass Multiple-Choice-Tests als freiwillige Überprüfung von gelerntem Wissen durchaus ihre Berechtigung haben. Dieses spiegelt sich beispielsweise in vielen Lehrbüchern wieder, die an Kapitelenden eine solche Möglichkeit anbieten. Bei studienrelevanten Leistungstests muss genau abgeschätzt werden, inwiefern Multiple-Choice-Tests herkömmlichen Klausuren und Prüfungsgesprächen überlegen sind. Es ist zu überprüfen, ob das designen von Tests zur Leistungskontrolle nicht zeit- und kostenintensiver ist, als der didaktische Nutzen rechtfertigt. Um Tests wiederzuverwerten und Chancengleicheit für alle Studierenden zu gewährleisten, ist zu bedenken, dass der Inhalt der Lehre auf solche Standardtests abgestimmt sein muss.

2.1.5 Aktuelle Plattformen

Vergleicht man nun aktuelle CSCL-Plattformen, stellt man fest, dass verschiedene Plattformen verschiedene konzeptuelle Ansätze verfolgen.

VITAL (Virtual Teaching and Learning)

Die VITAL[2]-Plattform stellt den Aspekt der Kommunikation (synchron oder asynchron) zwischen Lernenden in den Vordergrund. Sie ist gegliedert in Verschiedene *virtuelle Räume*, die konzeptuell an reale Räume angelehnt sind. Innerhalb der privaten sowie öffentlichen Räume kann kommuniziert und diskutiert werden, können Dateien getauscht werden usw. Ebenso besteht die Möglichkeit kooperativ Hypermediadokumente anzufertigen oder durch vorhandene Hypermediadokumente (z.B. Lerneinheiten etc.) zu navigieren. Bereitgestellt werden verschiedene Module wie: Worldbrowser, Chatfenster, Audiokonferenz, Frage-Antwortbrett und Bibliothek.

Blackboard

Einen anderen Ansatz verfolgt die kommerzielle Plattform Blackboard[3]. Hier steht neben den Möglichkeiten zur synchronen und asynchronen Kommunikation mehr die technische und organisationale Integration des Systems in die Lernumwelt im Vordergrund. Lehrmaterialien lassen sich zwischen Dokumentformaten im- und exportieren. Es existieren administrative Werzeuge wie z.B. eine Kurs- und Benutzerverwaltung und ein mächtiges Tutorsystem.

Worksphere

Die als Basis für mein Lernkontrollmodul eingesetzte Plattform ist die Open-Source-Lösung Worksphere[4]. Programmiert wurde sie von Heiko Holtkamp[5], 2003 Student der Naturwissenschaftlichen Informatik an der Universität Bielefeld.

Eingesetzt wird die Worksphere zur Zeit an der Fakultät für Gesundheitwissenschaften[6] der Uni-Bielefeld als CSCL-Werzeug für den Fernstudiengang *Angewandte Gesundheitswissenschaften*[7]. In Vorbereitung ist der Einsatz für die Lehre der Gruppe Rechnernetze und verteilte Systeme (RVS[8]) an der technischen Fakultät der Universität Bielefeld.

Die Worksphere beinhaltet ein Administrationstool (Gruppen und Benutzerverwaltung), schwarze Bretter, Arbeitsforen und Chatrooms, sowie eine Mediathek zum Download von Lehrmaterialien. Desweiteren bekommt der Benutzer Unterstützung durch verschiedene Hilfe- und Infoseiten.

[2]VITAL: http://www.ipsi.gmd.de/concert/projects/clear
[3]Blackboard: http://www.blackboard.com
[4]Worksphere: http://www.worksphere.de
[5]eMail: heiko@worksphere.de
[6]GW-Uni-Bielefeld: http://www.uni-bielefeld.de/gesundhw/index.html
[7]Fernstudium: http://plattform.gesund.uni-bielefeld.de
[8]RVS: http://www.rvs.uni-bielefeld.de

Die konzeptuellen Unterschiede dieser drei Plattformen sind offensichtlich. Die einen legen mehr Wert auf das bereitstellen von Kommunikationmöglichkeiten der Lernenden untereinander (z.B. VITAL), andere betonen mehr technische Features zur Lösung bestimmter allgemeiner Probleme der Integration des Werkzeugs in den Lehrbetrieb (z.B. Blackboard).

Solche oder ähnliche Lösungen existieren zudem in vielfältiger Form und es fällt schwer einen detaillierten Überblick zu gewinnen.

Mindestanforderungen

Allerdings gibt es Versuche zu klassifizieren, welche Funktionsmerkmale Learning Management Systeme[9] (LMS) ausmachen sollten. In einer Studie über LMS in [Schulmeister 2003] werden fünf mindestens notwendige Funktionsmerkmale genannt. Diese zitiere ich im folgenden ([Schulmeister 2003], S.55):

- Eine Benutzerverwaltung
- Eine Kursverwaltung
- Ein Rollen und Rechtevergabe mit differenzierten Rechten
- Kommunikationsmethoden und Werkzeuge für das Lernen
- Die Darstellung der Kursinhalte, Lernobjekte und Medien in einem netzwerkfähigen Browser.

Liest man diese Studie, wird allerdings auch nicht hundert Prozentig klar, was die einzelnen Funktionsmerkmale im Detail beinhalten. Es scheint Probleme zu bereiten, ein LMS und seine Funktionsmerkmale aufgrund der vielfältig existierenden Lösungen und verschiedenen Anforderungen als System klar abzugrenzen.

2.2 Zusammenfassung und Kritikpunkte

CSCL-Werzeuge zeichnen sich durch eine orts- und zeitunabhängige Benutzbarkeit der Lerninhalte und Module aus. Gearbeiet wird beispielsweise in virtuellen Räumen. Die Arbeitsweise ist meistens selbstorganisiert. Allerdings ist das vordergründige Ziel nicht, die Aufgaben oder Themen alleine zu bearbeiten, sondern der aktive Diskurs über Inhalte in Gruppen, z.B. zwischen Dozenten und Studenten, steht im Vordergrund. Diese Kommunikation findet synchron oder asynchron statt (z.B. über Foren und Chats). Dozenten haben Kontrolle über den Lernprozess, da praktisch eine permanente aktive sowie passive Verbindung mit den Studenten besteht und gute Feedback-Möglichkeiten gegeben sind.

Als Kritik ist anzumerken, dass viele CSCL-Werkzeuge Einzellösungen bestimmter Bildungseinrichtungen darstellen. Ihre Konzepte sind nur schwer auf andere Anwendungsumgebungen übertragbar. Vielfach fehlt das Geld und/oder die didaktische Kompetenz bei den Entwicklern um eine didaktisch hochwertige Lösung zu schaffen. Stattdessen werden sie quick&dirty implementiert

[9] CSCL-Werkzeuge gehören in diese Klasse von Software-Systemen

und verfehlen das Ziel einer effektiv nutzbaren Lern- und Übungsplattform [Schulmeister 2001].

Insgesamt folgt die didaktische Konzeption sowie die Implementierung von eLearning-Systemen keinem einheitlichen Standard. Es gibt versuche Standards für Lerntechnologien zu etablieren, was aber bei der Vielzahl der bereits existierenden CSCL-Werzeuge schwierig erscheint. Es existieren verschiedene Ansätze von verschiedenen Projektgruppen die sich mit der Standardisierung beschäftigen, ohne bisher eine gültige Übereinkunft festgelegt zu haben. Zwei Vorschläge sind der IMS Global Learning Standard[10] und die IEEE Learning Technologie Standards[11]. Mit dem IMS Standard wird mittlerweile für Produkte als *"IMS compliant"* geworben.

Der Integration eines Lernerfolgskontrollmoduls für CSCL-Systeme in Form eines Multiple-Choice-Tests steht nichts im Wege, solange die Nutzung des Testmoduls für Studenten freiwillig ist. Als Modul zur studienrelevanten Leistungskontrolle muss der didaktische Nutzen jedoch genau überprüft werden, muss Chancengleichheit für Studierende gewahrt bleiben und deshalb die Lehre entsprechend abgestimmt sein.

Der Aspekt (Computer-)Sicherheit findet in Design und Implementierung von CSCL-Werkzeugen wenig Aufmerksamkeit. Denn Sicherheit ist kostspielig, zeit- und personalaufwendig. In aktuellen Studien zu eLearning-Plattformen werden nur Authentifizierung und Datenverschlüsselung als Sicherheitsmechanismen gefordert, wobei Authentifizierung von den meisten und Datenverschlüsselung von den wenigsten Lernplattformen unterstützt wurde [Schulmeister 2003].

Diese sind nicht ausreichend um ein System als *sicher* im Sinne von *secure* zu bezeichen. Es muss neben Authentifizierung und Vertraulichkeit auch Integrität und Verfügbarkeit gewährleistet sein (vgl. [Schneier 2001, Anderson 2001, Schneider & Werner 2001, Stallings 1995].

[10] IMS: http://www.imsproject.org
[11] IEEE-LTSC: http://ltsc.ieee.org

Kapitel 3

Sicherheitsanalyse von Computersystemen

Um einen Einblick in die Begriffe der Sicherheitstechnik zu geben, erläutere ich hier kurz den von mir im folgenden benutzten Begriffsapparat. Danach beschreibe ich die zur Sicherheitsanalyse verwendete Methode der Angriffsbäume (engl. Attack-Trees) nach [Schneier 2001].

Ich versuche die Darstellung intuitiv verständlich zu gestalten, damit auch - nicht Sicherheitsexperten - die durchgeführten Sicherheitsanalysen verstehen und sich ihrer Grenzen bewusst sind.

3.1 Begriffe der Computersicherheit

In der Analyse ist es wichtig, das zu analysierende System klar abzugrenzen. Deshalb sollte verstanden werden, wie ein System aufgebaut ist und welche Eigenschaften es hat.

Ich werde erläutern, welche Bedeutung der Begriff Sicherheit in der Computersicherheit besitzt. Weiterhin werde ich skizzieren, wie Fehlerursachen und Ausfall verstanden werden und welche Rolle der menschliche Benutzer im Kontakt mit dem System inne hat.

3.1.1 System

Ein System besteht aus einer Anzahl von Objekten, welche selber wieder Systeme sein können. Diese Objekte stehen in bestimmten Beziehungen zueinander. Sie können sich gegenseitig beeinflussen und miteinander interagieren. Dieses nennt man Verhalten.

Ein System hat eine Umwelt bestehend aus Objekten und anderen Systemen. Mit dieser kann das System ebenso interagieren. Es kann Einflüssen aus der Umwelt ausgesetzt sein und selbst die Umwelt beeinflussen.

Außerhalb der Umwelt der Systeme befinden sich weitere Objekte und Systeme, welche als Welt bezeichnet werden. Die Welt hat keine direkte Beziehung

zum System bzw. Einfluss auf das Verhalten des Systems.

Betrachtet man eine Momentaufnahme des Systems, bezeichnet man diese als Zustand des Systems. Nach [Laprie 1992] setzt sich der Zustand eines Systems aus der Beschaffenheit der betrachteten Einheit und einer Menge von Randbedingungen zusammen. Dieses bezieht sich sowohl auf das Verhalten als auch die Struktur der betrachteten Einheit. Aus dieser Definition folgt, dass ein Systemzustand immer relativ vom Verhalten ("der Prozess x verarbeitet Programmstack y"), als auch von den Randbedingungen (z.B. der Zustand Z betrachtet in Bezug auf Benutzereingaben $i_1, ..., i_n$) abhängt und demnach *der* Systemzustand an sich nicht existiert. (weiteres vgl. [Laprie 1992, Ladkin 2001])

In der Computersicherheit sind Systeme beispielsweise Software, einzelne Computer, Server oder Netzwerke. Jeder Systemteil hat seine eigene Funktion. Die Umwelt ist das "Umfeld" des Systems. Denkbar ist ein Serverraum, ein Netzwerk oder die Benutzer der Systeme.

Betrachtet man ein Computersystem, verarbeitet der Prozessor Daten. Dieses ist eine Interaktion innerhalb eines Systems, also ein Verhalten des Computersystems. Dieses Verhalten der Interaktion zwischen Speichersystem und Prozessorsystem innerhalb des Computersystems beschreibt dann ein Verhalten, welches man als "Datenverarbeitung" bezeichnet.

Wenn ein Benutzer an einem Computersystem arbeitet, Dateien verändert und Programme ausführt, dann ist dieses ein Beispiel für die Interaktion zwischen Umwelt und System. In diesem Fall liegt es im Auge des Betrachters wer das System und wer die Umwelt ist.

3.1.2 Zuverlässigkeit

Ein System soll vertrauenswürdig Leistung erbringen. Leistung erklärt sich als das beobachtbare Verhalten des Systems [Laprie 1992]. Hiernach definiert sich ein zentraler Begriff im Hinblick auf Sicherheitsbetrachtungen: die Zuverlässigkeit eines Systems. Die Zuverlässigkeit eines Systems ist definiert als "die Vertrauenswürdigkeit eines Rechensystems, so dass Vertrauen in die Leistung, die es erbringt gesetzt werden kann"([Laprie 1992],S.101).

Im Detail beinhaltet der Begriff Zuverlässigkeit verschiedene Attribute, die in der Sicherheitstechnik ein zentrale Rolle spielen. Ein zuverlässiges Datenverarbeitungssystem ist sicher, d.h. verfügbar, funktionsfähig und geschützt [Laprie 1992]. Diese Aussage ist auf alle Arten von Datenverarbeitungsystemen anwendbar. Im folgenden betrachte ich allerdings ausschließlich computergestützte und netzwerkbasierte Datenverarbeitungssysteme.

3.1.3 Sicherheit - Security und Safety

Der deutsche Begriff Sicherheit spaltet sich im englischen in die Begriffe Security und Safety. Security bezieht sich allgemein auf Schutz gegen beabsichtigte Angriffe und Safety auf Schutz gegen unbeabsichtigte Ereignisse [Schneider & Werner 2001].

Security

Spricht man in der Datenschutz- und Datensicherheitstechnik von Security, dann ist das Ziel, Angriffe gegen die vier Dienste Authentizität, Vertraulichkeit, Integrität und Verfügbarkeit von Systemen abzuwehren bzw. zu erschweren [Schneier 2001, Anderson 2001, Schneider & Werner 2001, Stallings 1995].

Vertraulichkeit: Stellt sicher, dass Informationen eines Computersystems nur bestimmten Personen zugänglich sind. Dieses bezieht sich auf Lese-, sowie Ausführrechte von Programmen und Daten. Merkmale sind Anonymität, Unbeobachtbarkeit, Unverkettbarkeit und Abhörsicherheit der Daten und Verbindungen [Schneier 2001, Schneider & Werner 2001]. Ein Beispiel für die Realisierung von Vertraulichkeit ist die Benutzerverwaltung mit entsprechenden Rechten wie sie in heutigen Computersystemen verwendet wird.

Authentifizierung: Beinhaltet Mechanismen, um ein System vor unbefugtem Zugriff zu schützen. Viele heutige Systeme benutzen Authentifizierung, z.B. Handys oder Banken. Authentifizierung funtioniert durch Login-Protokolle, wie z.B. per Texteingabe, Kartenidentifikation oder biometrische Erfassung. Es wird sichergestellt, dass der Authentifizierte auch wirklich die entsprechend passende Person ist und nicht jemand der sich dafür ausgibt. Allerdings existieren verschiedene Probleme. Benutzeridentifikationen und deren Passwörter sind oftmals nicht sicher und ausspürbar [Schneier 2001]. Biometrische Systeme können sehr sicher sein [Anderson 2001], sind aber entsprechend teuer und gesellschaftlich kritisch diskutiert. Alle diese Mechanismen verlieren ihren Nutzen, wenn man z.B. einfach die Festplatte eines Computers ausbauen kann, auf der unverschlüsselte Daten gespeichert sind; oder Computer mit einer laufenden Sitzung unbeaufsichtigt zurückgelassen werden.

Integrität: Stellt sicher: wer, wann und ob Daten seit einem bestimmten Zeitpunkt geändert bzw. gelöscht wurden oder nicht; bezieht sich also auf Schreibrechte von Daten und Programmen [Schneier 2001]. Merkmale sind Zurechenbarkeit, Übertragungsintegrität, Abrechnungssicherheit [Schneider & Werner 2001] der Daten und Verbindungen. Um Integrität sicherzustellen, setzt man z.B. Verschlüsselung oder Sicherheitszertifiakte beim Datenaustausch ein. Populäres Beispiel ist das signieren von eMails mit Hilfe von Pretty-Good-Privacy (PGP[1]).

Verfügbarkeit: Stellt sicher, dass überhaupt eine Kommunikation zwischen Systemen stattfindet. Server stürzen ab (Denial-of-Service-Attacke) oder fallen Viren zum Opfer [Schneier 2001, Anderson 2001, Anonymus 1999]. Die Daten auf diesem Server sind dann erstens nicht mehr verfügbar und zweitens

[1] PGP-International: http://www.pgpi.org

womöglich geändert oder zerstört. Gegenmaßnahmen sind u.a. Virenscanner und Firewalls zur Abwehr von Angriffen gegen Verfügbarkeit.

Safety

Safety stellt die technische Funktionsfähigkeit des Systems in den Vordergrund. Einige Systeme müssen jederzeit verfügbar sein. Dieses beinhaltet einen Schutz gegen technischen Ausfall (Funktionssicherheit) und technische Sicherheit gegen ungewollte Gefahren für die Systemumwelt [Schneider & Werner 2001]. Ein Beispiel: Server-Hardware soll jederzeit und zuverlässig ihren Dienst erfüllen (Funktionssicherheit). Trotz Dauerbetriebes darf sie nicht überhitzen und anfangen zu brennen, womit angrenzende Systeme ebenfalls betroffen wären (technische Sicherheit).

3.1.4 Fehlerursachen und Ausfall

Ein im Betrieb befindliches System funtioniert nicht ausschließlich korrekt. Es ist immer mit einem zeitweisen Auftreten von Fehlerzuständen (engl. error) zu rechnen. Diese Fehlerzustände sind dann verantwortlich für den eventuell folgenden Ausfall (engl. failure) des Systems. Fehler[2] und deren Quellen werden in [Laprie 1992] nach drei Hauptkriterien klassifiziert: nach Art, Ursprung und Dauer des Fehlers.

Unterscheidet man Fehler nach ihrer Art beobachtet man Zufallsfehler und Absichtsfehler. Die ersten entstehen rein zufällig und unbeabsichtigt, die zweiten werden absichtlich erzeugt, vielleicht sogar mit böswilligen Hintergrund [Laprie 1992].

Der Ursprung von Fehlern kann vielfältiger Natur sein. Er kann einem physikalischen Problem (physikalischer Fehler) zugrunde liegen oder durch die Unzulänglichkeit des Menschen (menschliche Fehler) entstehen. Die Ursache kann innerhalb des Systems liegen (interner Fehler), oder externer Beeinflussung (externer Fehler) sein. Weitere Quellen von Ursachen liegen in fehlerhafter Systementwicklung, oder fehlerhaftem Betrieb und Wartung (Entwurfsfehler) oder falscher Benutzung des Systems (Benutzungsfehler) [Laprie 1992].

Entscheidend für die Betrachtung und Auswirkung von Fehlern ist auch deren Dauer. Fehler können von punktuellen Bedingungen abhängen (temporärer Fehler) und sind daher nur für eine bestimmtes Zeitintervall anwesend. Oder sie sind *nicht* abhängig von punktuellen Bedingungen (permanenter Fehler) und solange anwesend, bis die meistens interne oder externen Fehlerursache beseitigt ist [Laprie 1992].

Ist das System ausgefallen, ist die Art des Ausfalls interessant. Aus Sicht der folgenden Sicherheitsanalyse ist vor allem der Schweregrad des Ausfalls relevant. Dieser wird in der Regel auf Basis der vom ihm verursachten Kosten gemessen. Der Schweregrad des Ausfalls hängt von der Art der Fehlerursache, des Fehlerzustandes und den Randbedingungen der Umwelt ab. Handelt es sich um einen

[2]Fehler werden hier in der Regel betrachtet im Sinne von Fehlerursache (engl. fault)

kritischen Ausfall, so sind die Kosten des Ausfalls erheblich höher als der Nutzen der Leistung des betriebsbereiten Systems. Ist der Ausfall unkritisch, so liegen die Kosten des Ausfalls in der selben Größenordnung wie der Nutzen des betriebsbereiten Systems [Laprie 1992]. Systeme deren Ausfälle ausschließlich unkritisch oder zu einem akzeptablen Grad unkritisch sind heißen Fail-Safe System. [Laprie 1992]

Ein einfaches Beispiel: Das betrachtete System ist das Netzwerk einer Softwarefirma. Die Daten, persönliche und firmenspezifische, der Mitarbeiter sind auf einem Fileserver gespeichert und werden über NFS lokal am Arbeitsplatz zur Verfügung gestellt. Eine lokale Kopie ist somit prinzipiell obsolet. Es existiert allerdings *kein* Backupsystem, falls der Fileserver ausfällt. Die Systembetreuer sind im Rahmen einer Rufbereitschaft 24 Stunden täglich erreichbar und für die Betriebsfähigkeit des Systems zuständig.

Nun geschieht aber genau dieses. Die Fehlerursache (fault) könnte z.B. ein physikalischer, permanenter, interner Fehler sein: eine Platine des Servers schmort durch. Die Datenbits werden nicht mehr korrekt transportiert und verarbeitet, womit der Fehlerzustand (error) klassifiziert ist. Der Schweregrad des folgenden Ausfalls hängt nun von den Randbedingungen ab.

Geschieht der Ausfall an einem Sonntagmorgen, an dem keiner oder nur wenige Arbeiten, sind die Kosten (Arbeitsausfall) im Verhältnis zur Leistung bei korrektem Betrieb (Bereitstellung der Daten) eher gering. Der Sonntag ist aber kein normaler Arbeitstag. D.h., die Systembetreuer beheben am Sonntag den Fehler und die Mitarbeiter arbeiten nach Behebung des Ausfalls am Montagmorgen normal weiter.

Geschieht der Ausfall allerdings an einem Werktag kurz vor einer wichtigen Termindeadline, so ist der Ausfall als kritisch einzustufen. Die Kosten (z.B. Verlust eines Kunden und weiterer Aufträge) sind unter diesen Umständen sehr wahrscheinlich höher als die Leistung und Reparaturkosten des Servers.

Als Fazit sollten die Systemadministratoren über ein Backupsystem nachdenken, um diese Art kritischer Ausfälle zu vermeiden bzw. zu überbrücken und das System Fail-Safe zu machen.

3.1.5 Benutzer und Angreifer

In der Computersicherheit wird unterschieden zwischen Menschen, die das System Benutzen (Benutzer) und Menschen die das System korrumpieren wollen (Angreifer). Benutzer und Angreifer sind sogenannte Akteure, die in für sie nützlicher Form und meistens zielgerichtet mit dem System agieren. Akteuren werden Rollen zugewiesen, welche den Typ des Akteurs näher spezifiziert.

Benutzer werden innerhalb des Systems durch ihnen zugewiesene Rollen repräsentiert. Jede dieser Rollen ist mit bestimmten Rechten im System verknüpft Daten zu lesen, zu schreiben oder Programme auszuführen. Ein Administrator hat zum Beispiel umfangereichere Rechte im System als Joe-User. Das Rechtesystem wird durch Authentifizierung der Benutzer am System realisiert.

Angreifer sind diejenigen, die böswillig ein System oder Teile eines Systems stören bzw. zerstören wollen. In der Literatur werden sie meistens Hacker genan-

nt (z.B. bei [Schneier 2001]). Ich werde mich aber an die Unterscheidung von [Anonymus 1999] halten, der diese als Cracker bezeichnet.

Es ist zu beachten, dass das beste technische Sicherheitssystem nicht sicher ist, solange die Menschen die es benutzen sich nicht an die Sicherheitsrichtlinien zur Systembenutzung halten [Schneier 2001, Anonymus 1999].

3.2 Sicherheitsanalyse mit Angriffsbäumen (Attack-Trees)

Der Begriff Angriffsbäume (Attack-Trees) wurde von [Schneier 2001] geprägt. Allerdings existieren Angriffsbäume in ähnlicher Form als Konzept sogenannter Fehler-Bäume (engl. Fault-Trees) schon lange in der Analyse von Sicherheitsproblemen.

3.2.1 Aufbau und Semantik

Angriffsbäume stellen verschiedene Angriffe gegen ein System in einer Baumstruktur dar. Angriffsbäume haben immer die Form von UND/ODER-Bäumen. Der Wurzelknoten beschreibt das Ziel, welches der Angreifer erreichen möchte. Die Unterknoten beschreiben Unterziele oder Teilziele (engl. subgoals) die auf dem Weg dorthin je nach Verknüpfungsart nur teilweise oder vollständig erfüllt sein müssen. Bei UND-Verknüpfungen müssen alle Unterziele erfüllt sein, bei ODER-Verknüpfungen muss jeweils nur mindestens ein Unterziel erfüllt sein.

Angriffspfade und Angriffszenarien

Durch die UND/ODER Semantik der Angriffsbäume ist eine Darstellung dieser auch in Form von booleschen Gleichungen möglich. Durch die Logik wird eine systematische Beschreibung von Angriffspfaden und Angriffsszenarien ermöglicht.

Ein Angriffspfad ist der Weg, den der Angreifer von einem Angriffszenario des Baumes, entlang der Unterziele, bis zum Angriffsziel zurücklegen und erfüllen muss. Als Angriffsszenarien bezeichnet man die booleschen Verknüpfungen der Blätter des Angriffsbaumes [Moore & Ellison & Linger 2001].

Alle Angriffsszenarien können, disjunktiv verknüpft, zu einem globalen Angriffsszenario zusammengefügt werden. Dieses globale Angriffsszenario beinhaltet dann alle innerhalb der Betrachtungsebene klassifizierten Möglichkeiten, einen Angriff auf das Ziel zu beginnen. Jedes einzelne Angriffsszenario bezeichnet somit die Voraussetzung zur Durchführung eines Angriffs entlang des dann folgenden Angriffspfades.

Visualisierung

Angriffsbäume können textuell (s. Tabelle 3.1 auf der nächsten Seite) oder grafisch (s. Abbildung 3.1 auf der nächsten Seite) repräsentiert werden. An-

griffsbäume beinhalten allgemein beliebig viele Kombinationen von UND- und ODER-Verknüpfungen.

Die einzelnen Angriffsziele sowie die zu erfüllenden Teilziele werden in eckigen Kästchen dargestellt. Diese sind beliebig nummerierbar, wobei die Nummerierung eindeutig sein muss. Kästchen die grau unterlegt sind, repräsentieren die oben beschriebenen Angriffszenarien. Alle grau unterlegten Kästchen eines Angiffsbaumes zusammen genommen bilden somit das globale Angriffsszenario des Angriffsbaums.

Goal	G_0	Goal	G_0
AND	G_1	OR	G_1
	G_2		G_2
	...		...
	G_n		G_n

Tabelle 3.1: Textuelle Repräsentation eines UND- und ODER-Baumes

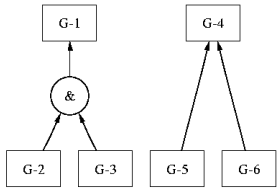


Abbildung 3.1: Grafische Repräsentation eines UND- und ODER-Baumes

Ein Beispiel: Der UND/ODER-Baum in Abbildung 3.2 auf der nächsten Seite enthält die in Tabelle 3.2 auf der nächsten Seite dargestellten Angriffsszenarien. Dass heißt, um das Ziel G-0 zu erreichen, muss der Angriffspfad über

$$(G_3 \wedge G_4) \vee (G_5) \vee (G_6) \tag{3.1}$$

gewählt werden. Diese Formel wäre dann die boolesche Repräsentation des globalen Angriffsszenarios für den Angriffsbaum in Abbildung 3.2 auf der nächsten Seite.

Erweiterungen

[Schneier 2001] fügt zusätzlich an die Knoten Gewichte an. Diese beschreiben die notwendigen Kosten zum erreichen des entsprechenden Unterzieles. Möglich sind z.B. Kosten in Geldeinheiten, oder Attribute wie "dieser Knoten ist nicht lösbar" bzw. "dieser Knoten ist immer lösbar". Diese Zusätze können helfen, einen besseren Überblick über wahrscheinlichere oder teurere Angriffswege zu bekommen, sind aber optional zu gebrauchen.

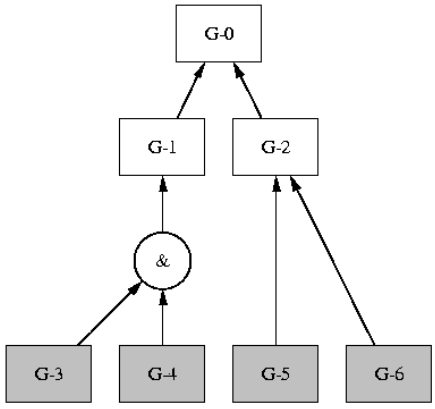


Abbildung 3.2: Beispiel für einen einfachen UND/ODER-Baum

⟨ G-3 , G-4 ⟩

⟨ G-5 ⟩

⟨ G-6 ⟩

Tabelle 3.2: Die drei möglichen Angriffsszenarien des einfachen UND/ODER-Baums

Vollständigkeit und Korrektheit

Die notwendige Bedrohungsanalyse zur Erstellung eines Angriffsbaumes bezeichnet [Schneier 2001] als Ad-Hoc Angelegenheit. Der Analyst lässt soviele Bedrohungen für das entsprechende Angriffsziel in die Analyse einfließen, bis ihm keine Möglichkeit mehr einfällt. Demnach ist die Analyse stark abhängig von der Expertise des Analysten.

Daraus ist zu schließen, dass die Sicherheitsanalyse mit Angriffbäumen *keine* beweisbar vollständige Analyse des betrachteten Sicherheitsproblems darstellt, sondern Lücken aufweisen wird. Vielmehr ist sie eine relativ simple und intuitive Möglichkeit, Bedrohungsanalysen schematisch darzustellen.

Auch existiert bei [Schneier 2001] keine Angabe darüber, wie rigoros die einzelnen Teilziele voneinander abhängen müssen. Es wird kein strenger kausaler Zusammenhang gefordert, wie es zum Beispiel in Form des *counterfactual-tests* der *Why-Because-Analyse* in [Ladkin 2001] der Fall ist.

Unter diesen Voraussetzungen ist es demnach nicht möglich, die Korrektheit einer Analyse mit Angriffsbäumen durch einen formalen Beweis aufzuzeigen, da kein Formalismus hierzu existiert. Es ist zwar möglich, Angriffsszenarien und Angriffspfade formal syntaktisch mit Hilfe von Aussagenlogik zu beschreiben. Die kausalen Zusammenhänge der Teilziele können aber nur auf unformalem Wege begründet werden.

3.2.2 Motivation und Kritik

Vorteile einer Analyse mit Angriffsbäumen ist, dass die Ebene der Betrachtung frei gewählt werden kann. Es ist möglich, Angriffsszenarien auf allgemeiner Ebene zu analysieren, ebenso können einzelne Unterziele im Detail in gleichen oder separaten Angriffsbäumen betrachtet werden. Angriffswege werden intuitiv und systematisch dargelegt. Die boolesche Semantik ermöglicht eine formal syntaktische Beschreibung der Analyse.

Angriffsbäume können in Softwareentwicklungsprozesse eingebunden werden. Sie ermöglichen eine vorab und/oder begleitende Analyse von Sicherheitsproblemen in Softwareentwicklungsprozessen und können jederzeit aktuellen Veränderungen angepasst werden. Beschrieben wird dieses als Spiral-Attack-Tree-Analyse (SATA) in [Ellermann 2002].

Anzumerken ist, dass ein Angriffsbaum nicht zwangsläufig alle Bedrohungsmöglichkeiten abdeckt und somit keine bedenkenlose Sicherheit garantieren kann. Die Vollständigkeit eines Angriffsbaumes ist in jedem Fall abhängig von der Expertise des Erstellers und seiner Kreativität beim erdenken möglicher Angriffszenarien. Es wird immer wieder Cracker geben, die mit neuen und nicht bedachten Ideen angreifen.

Auch ist nach der Aufstellung eines Angriffsbaumes nicht bewiesen, dass dieser korrekt ist. Deshalb muss bei der Analyse mit Angriffsbäumen darauf geachtet werden, dass das betrachtete System und dessen Bedrohungen möglichst klar abgrenzt sind, damit dadurch die unformale Begründung der Analyse in sich schlüssig und für Dritte nachvollziehbar bleibt.

3.3 Nutzen von Sicherheitsanalysen

Die Sicherheitsanalyse allgemein gibt Aufschluss über Sicherheitsprobleme in Computersystemen. Sie bietet eine Möglichkeit, systematisch Sicherheitslücken aufzudecken und deren Ursache und Wirkung zu beschreiben.

Aber eine Analyse hat ihre Grenzen und eine durchgeführte Analyse garantiert nicht automatisch ein sicheres System. Probleme werden übersehen, Sicherheitlücken müssen auch tatsächlich beseitigt werden.

Menschen bedienen Computersysteme. Halten diese sich nicht an die Sicherheitsrichtlinien, helfen auch die besten technischen Lösungen nicht weiter.

Eine weitere Problemklasse liegt bei der Programmierung der Systeme selbst. Eine Sicherheitsanalyse des Designs ist nicht in der Lage, Programmierfehler zu erkennen. In sicherheitskritischen Systemen können Programmierfehler dann letztendlich fatale Auswirkungen herbeiführen.

Um ein System demnach möglichst sicher zu machen, darf nicht nur eine detaillierte Analyse stattfinden. Die Benutzer und Entwickler des Systems müssen für Sicherheitsschwächen sensibilisiert werden. Sicherheitsanforderungen und Sicherheitsrichtlinien müssen bei der Entwicklung des Systems, der Systemintegration und der Systembenutzung rigoros umgesetzt werden. Regelmäßige Sicher-

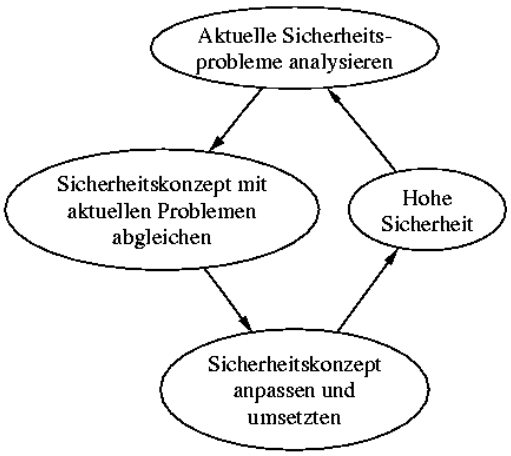


Abbildung 3.3: “Magischer” Kreislauf der Sicherheitstechnik

heitsanalysen tragen dazu bei, Sicherheitsbedürfnisse an aktuelle Sicherheitsprobleme anzupassen (siehe Abbildung 3.3).

Kapitel 4

Sicherheit in CSCL-Werkzeugen

CSCL-Werkzeuge sind netzwerkbasierte Software Applikationen. Aus diesem Grunde sind sie grundsätzlich anfällig für alle Arten bekannter Sicherheitsprobleme in Netzwerken, z.B. Angriffe gegen Integrität oder Vertraulichkeit. Ebenso habe ich in Kapitel 2.2 auf Seite 12 bereits festgestellt, dass die bisher in CSCL-Werkzeugen geforderten bzw. implementierten Sicherheitsmechanismen nicht ausreichend sind um diese Systeme als *sicher* zu bezeichnen.

In der Analyse ist demnach herauszuarbeiten,

- welche Eigenschaften ein CSCL-System ausmachen
- welche Angriffsszenarien auf CSCL-Systeme aufgrund dieser Eigenschaften denkbar sind
- welche Konsequenzen die Durchführung eines Angriffs auf das System hat
- wie diese Konsequenzen zu bewerten sind und welche sicherheitstechnischen Schutzmechanismen notwendig werden

Bei der Analyse ist jedoch zu beachten, dass es sich bei CSCL-Werkzeugen nicht um Hochsicherheitssysteme handelt. Vielmehr sind sie didaktische Lernwerkzeuge, bei denen intuitive und leichte Benutzbarkeit im Vordergrund steht. Bei der Auswertung wird daher betrachtet, welche Systemteile mehr und welche weniger sicherheitskritisch sind. Bei der Bewertung der Analyse wird zwischen Sicherheitsbedürfnissen und Benutzbarkeit des Systems abgewogen.

4.1 CSCL-Werkzeuge als System

In diesem Abschnitt beschreibe ich CSCL-Werkzeuge im Sinne des Systembegriffs aus Kapitel 3.1.1 auf Seite 14. Dazu betrachte ich die Eigenschaften von CSCL-Werkzeugen und der mit dem System interagierenden Umwelt.

Ziel ist eine klare Abgrenzung eines CSCL-Systems zu erlangen. Dieses ist für die Analyse notwendig, da sie innerhalb eines klar definierten Problembereichs stattfinden muss um eine klare Struktur und Begründungsgrundlage zu ermöglichen (s. Kapitel 3 auf Seite 14).

Als Basis für die systemische Klassifikation gilt die Beschreibung von CSCL-Werkzeugen im gesamten Kapitel 2 auf Seite 5. Die folgenden Beschreibungen sind als Interpretation der oben beschriebenen Eigenschaften von CSCL-Werkzeugen im Sinne der Systemontologie zu verstehen.

Zum Verständnis der Analyse setze ich grundlegende Kenntnisse über den Aufbau und Funktion von Client-Server-Applikationen voraus. Details hierzu finden sich z.B. bei [Peterson & Davie 2000].

4.1.1 Beschreibung des CSCL-Systems

Allgemeine Spezifikation

Ein CSCL-Werkzeug ist eine in ein Computernetzwerk integrierte Applikation. Das heißt die Teilsysteme des CSCL-Systems, welcher Art auch immer, werden auf einem (Netzwerk-)Server ausgeführt und die Umwelt kommuniziert über einen Kommunikationskanal (z.B. einer Datenleitung) mit den entsprechenden Teilsystemen.

Die Server, auf denen die Ausführung des Systems beruht, haben genau zwei Aufgaben:

1. Ermöglichen der Kommunikation zwischen System und Umwelt.
2. Speichern und bereitstellen von zur Ausführung des Systems benötigter Daten.

Konkret bedeutet dieses, dass ein CSCL-System einen Dienst besitzt, der die Kommunikation mit der Außenwelt regelt. Dieser heißt im folgenden Webserver. Ebenso besitzt das System einen Dienst, der das bereitstellen und speichern von Daten erledigt. Dieses wird teilweise durch den Webserver geschehen. Zusätzlich wird ein spezieller Dienst benötigt, der ausschließlich für die Verwaltung und Bereitstellung von Daten zuständig ist. Dieser erhält die Bezeichnung Datenbankserver.

Zustände

Das CSCL-System hat drei hier relevante und klassifizierbare Zustände:

1. Betriebsbereit: das System ist abgeschaltet, kann aber jederzeit in Betrieb genommen werden.
2. In Betrieb: das System befindet sich im regelgerechten Betrieb entsprechend seiner Spezifikation.
3. Ausgefallen: das System ist aufgrund einer Fehlerursache ausgefallen.

Diese Zustände können problemlos auf die oben erwähnten Server übertragen werden. Im regelgerechten Betrieb gibt es deshalb nur zwei im Sinne der Spezifikation gültige Zustandübergänge g_1 und g_2:

1. Betriebsbereit $\rightarrow_{g_1}$ In Betrieb
2. In Betrieb $\rightarrow_{g_2}$ Betriebsbereit

Das Auftreten einer Fehlerursache (fault) kann zum Ausfall (failure) des Systems führen. Dieses entspricht folgendem Zustandsübergang f_1:

3. In Betrieb $\rightarrow_{f_1}$ Ausfall

Dieser Zustandsübergang ist zwar möglich, entspricht aber nicht den Spezifikationen und ist damit in diesem Sinne nicht als gültiger Zustandsübergang, sondern als Zustandsübergang von einem gültigen Zustand in einen Fehlerzustand (error) zu klassifizieren[1].

Klassifikation der verwalteten Daten

Ein CSCL-System verwaltet verschiedene Arten von Daten die Sicherheitstechnisch unterschiedlich kritisch einzustufen sind[2].

Persönliche Daten: Daten die persönliche Informationen enthalten (Namen, Adressen, Noten, etc.) und Daten die zum persönlichen Gebrauch erstellt wurden (Texte, Präsentationen, etc.). Sie sind sehr sensibel und damit sicherheitskritisch.

Allgemeine Daten: Daten, die allgemein zugänglich sind (z.B. Bekanntmachungen, Hilfeseiten, etc.); sind weniger sicherheitskritisch da in der Regel öffentlich.

Programmdaten: Daten, die zur Ausführung des CSCL-Systems notwendig sind (z.B. Skripte, Konfigurationsdaten, etc.); Sie sind auf jeden Fall dann sicherheitskritisch, wenn eine Veränderung dieser Daten zum Ausfall des CSCL-Systems führen würde.

Auf Daten kann man drei grundlegende Funktionen anwenden: man kann sie *lesen* und man kann sie *schreiben*, wobei mit schreiben auch das nachträgliche Verändern (manipulieren) von Daten gemeint ist. Programmdaten[3] kann man *ausführen*.

4.1.2 Die Umwelt des Systems

Allgemeine Spezifikation

In der Umwelt des CSCL-Systems sind alle Systeme enthalten die mit dem CSCL-System interagieren. Diese Interaktion ist konkret auf eine bereitgestellte Kom-

[1] Ein System kann natürlich auch den Zustand *Defekt* besitzen, weil innerhalb des Systems ein Teilsystem ausgefallen ist. Allerdings spielt dieser hier keine Rolle, da nur der Zustandübergang betrachtet wird, der zu einem *gezielt* herbeigeführten Fehlerzustand führt, dessen Ergebnis ein Ausfall des Systems ist

[2] Zu den Einstufungen vergleiche z.B. [Schneier 2001, Anonymus 1999, Anderson 2001]

[3] Programmdaten werden auch als Programme bezeichnet

munikationsleitung beschränkt[4]. Konkret sind das:

1. Andere Rechnersysteme, sogenannte *Clients*, die als Service die Kommunikationsverbindung zu den CSCL-System-Servern erstellen und aufrecht halten.

2. Menschen oder Computersysteme, die mit Hilfe der Services der Clients mit dem CSCL-System interagieren wollen.

Menschen, sogenannte Akteure, oder Computer verbinden sich mit Hilfe der Clients z.B. über ein Internetwork mit dem CSCL-System und benutzen dann die bereitgestellten Funktionen des CSCL-Systems um ein Ziel (z.B. das Schreiben einer Mitteilung) zu erreichen.

Zustände

Für die Analyse des CSCL-Systems ist nun wichtig, die Zustände des Kommunikationskanals zu klassifizieren, da dieser die direkte Verbindung des CSCL-Systems mit der Umwelt darstellt. Fällt er aus, ist keine Interaktion mit Umwelt möglich. Da die Umwelt das System zielgerichtet benutzt, kann man unter diesen Umständen das System für die Umwelt als unbenutzbar bezeichen. Der Kommunikationskanal besitzt folgende hier relevante Zustände:

1. Betriebsbereit: Verbindung zwischen System und Umwelt und demnach der Datenfluss ist möglich; es fließen aber keine Daten

2. In Betrieb: Verbindung zwischen System und Umwelt besteht; es fließen Daten.

3. Unterbrochen: die Verbindung zwischen System und Umwelt ist unterbrochen; es können keinen Daten fließen.

Demnach ergeben sich im Sinne der Spezifikation die gültigen Zustandsübergänge g_3 und g_4:

4. Betriebsbereit $\rightarrow_{g_3}$ In Betrieb
5. In Betrieb $\rightarrow_{g_4}$ Betriebsbereit

Desweiteren ist auch folgender Zustandübergang f_2 möglich:

6. In Betrieb $\rightarrow_{f_2}$ Unterbrochen (Ausfall)

Dieser Zustandsübergang entspricht nicht den Spezifikationen und überführt das System von einem gültigen Zustand in einen Fehlerzustand.

[4] Auch wenn man lokal an einem Rechner arbeitet, kommuniziert man über die Kommunikationsleitung des Webservers in Form eines Loopback-Device mit dem CSCL-System. Deshalb mache ich hier an dieser Stelle keine Unterscheidung diesbezüglich.

Akteure und Rollen

Die Akteure die mit einem CSCL-System interagieren lassen sich durch drei Rollen klassifizieren. Jede Rolle ist mit bestimmten Rechten Daten innerhalb des CSCL-Systems zu lesen, zu schreiben oder Programmdaten auszuführen verknüpft. Diese Rollen sind:

Administratoren: Obliegt die Aufgabe, das System betriebsbereit oder im Betrieb zu halten. Sie verwalten das CSCL-System und sorgen für die Integration des CSCL-Systems in die Lernumwelt. Sie haben vollständigen Zugriff auf alle Teilsysteme.

Studenten: Benutzen das System in dem sie die bereitgestellten Teilsysteme wie z.B. Messageboards nutzen. Sie haben Zugriff auf ihre persönlichen Daten und Programme oder auf Daten und Programme für die sie explizit eine Berechtigung erhalten haben.

Sonstige: Alle anderen Akteure die mit dem System interagieren. Dies sind z.B. Dozenten, Tutoren oder Cracker. Sie haben je nach Rolle unterschiedliche bis keine Rechte im System.

4.1.3 Sicherheitsmechanismen

Die oben beschriebenen Systeme unterliegen nach Kapitel 2.2 auf Seite 12 teilweise verschiedener Sicherheitsmechanismen.

1. Ein Zugriff auf einen Server verlangt eine Authentifizierung. Die Authentifizierung erfolgt von extern durch einen Client oder innerhalb des CSCL-Systems selbst, z.B. bei einem Zugriff auf den Datenbankserver.

2. Über die Datenleitungen erfolgt, je nach Sicherheitstandard, die verschlüsselte Übertragung der Daten, um abhören oder verfälschen der Daten durch Dritte zu erschweren.

4.1.4 Zusammenfassung

Das CSCL-System ist auf der zur Analyse verwendeten Betrachtungsebene durch folgende Teilsysteme mit entsprechenden Aufgaben beschrieben:

Webserver: Ermöglicht die Kommunikation zwischen System und Umwelt; verwaltet Programmdaten (z.B. bei webbasierten Applikationen); besitzt Authentifizierungsmechanismen

Datenbankserver: Stellt die zur Ausführung des Systems benötigten Daten zur Verfügung; hierbei handelt es sich meistens um inhaltsbezogene Daten (sensible Daten und allgemeine Daten) und weniger um programmbezogene Daten (Programmdaten); besitzt Authentifizierungsmechanismen.

Netzwerk: Stellt die Kommunikationskanäle in Form von Datenleitungen zur Verfügung. Diese sind für die Kommunikation zwischen CSCL-System und Umwelt notwendig; kann verschlüsselte Daten übertragen.

Stört nun jemand gezielt den regelgerechten Betrieb des CSCL-Systems bzw. seiner Teilsysteme, oder verschafft sich Zugang zu Daten oder Programmdaten, für die er keine Authentifizierung besitzt, kann man dieses als Angriff auf das CSCL-System bezeichen.

4.2 Angriffsziel CSCL-System

Nach den Betrachtungen im vorigen Abschnitt, lassen sich folgende allgemeine Angriffsziele für einen Angriff auf ein CSCL-System definieren:

1. Gezieltes stören eines Teilsystems bishin zur Überführung des Teilsystems in den Fehlerzustand Ausfall.

2. Erlangung von unautorisiertem Zugriff auf ein Teilsystem ohne passende[5] Authentifizierung, mit dem Ziel Daten zu lesen (ausspionieren), Daten zu schreiben (verändern), oder Programmdaten auszuführen und das Teilsystem dadurch in den Fehlerzustand Ausfall zu überführen.

Anmerkungen zur Analyse

Im folgenden konktretisiere ich die Analyse durch anwenden der Angriffsziele auf die einzelnen Teilsysteme Webserver, Datenbankserver und Datenleitung. Die folgenden unformalen Darstellungen der Angriffe erläutern kurz die grundsätzliche Vorgehensweise bei Angriffen auf die entsprechenden Ziele. Im Detail stelle ich jeden Angriff am Ende des Kapitels als Angriffsbaum dar. Diese sind dann in Verbindung mit den folgenden Beschreibungen intuitiv klar.

Da die Betriebsfähigkeit der CSCL-Systeme von den oben beschriebenen Systemen abhängt, ist es ausreichend, die Analyse auf diese zur Beschränken, da ein Ausfall dieser einen Ausfall des gesamten CSCL-Systems nach sich zieht. Außerdem ist jetzt nicht bekannt, wie das CSCL-System konkret beschaffen ist. Auch diese Tatsache erfordert eine allgemeine Betrachtung.

Da auf jedes Teilsystem über Authentifizierungmechanismen (bzw. überwundene Authentifizierungsmechanismen) zugegriffen werden kann, stelle ich als erstes den "Angriff zur Erlangung von Administratorzugriff" vor, da dieser in jeder weiteren Analyse enthalten ist. Dieses ist immer dann der Fall, wenn als Voraussetzung für ein Angriffsszenario Adminstratorzugriff notwendig sind.

Für etwaige Fachbegriffe und Details innerhalb der Analyse verweise ich auf die einschlägige Literatur, z.B. [Anderson 2001, Schneier 2001, Anonymus 1999, Stallings 1995, Schneider & Werner 2001] oder das Glossar.

[5] *Passend* meint hier, dass der Angreifer zwar gültige Zugangsdaten besitzen kann, diese aber nicht zu seiner Person passen, weil sie nicht seine eigenen sind

4.2.1 Angriff zur Erlangung von Administratorzugriff

Ziel dieses Angriffs ist, Administratorzugriff auf dem Zielsystem zu erlangen. Hat der Cracker dieses Ziel erreicht, kann er im System auf alle Dienste und Dateien zugreifen und sie nach eigenen Wünschen gebrauchen bzw. missbrauchen. Demnach ist es dann leicht, das System in den Fehlerzustand Ausfall zu überführen oder sensible Daten zu lesen oder zu verändern oder Programmdaten auszuführen. Um Administratorzugriff zu erlangen gibt es mehrere Möglichkeiten:

Zugangsdaten nutzen: Zugangsdaten werden durch lexikalische Angriffe, Social Engeneering oder abhören der Kommunikation autorisierter Benutzer ausspioniert oder gecrackt. Weiterhin kann der Cracker die offene Sitzung eines autorisierten Benutzers übernehmen. Hat der Cracker lediglich Userzugriff auf das System, kann er in Kombination mit dem Teilangriff "Programmfehler nutzen" Administratorzugriff erlangen.

Programmfehler nutzen: Der Cracker hat Kenntnisse von Sicherheitslücken im Zielsystem und nutzt diese mit Hilfe von Programmiertechniken aus.

Admin sein: Denkbar ist auch, dass er selber Administrator ist und z.B. im Auftrag Dritter Manipulationen am System durchführt; dieses ist allerdings kein klassischer Angriff auf ein System und nur der Vollständigkeit wegen erwähnt

4.2.2 Angriff auf Kommunikationsstrukturen

Ziel eines Angriffs auf die Kommunikationsstrukturen, d.h. auf die Datenleitungen bzw. Kommunikationskanäle ist:

Lesen der Kommunikation: Der Angreifer zapft den Kommunikationskanal an um Daten mitzulesen. Dieses geschieht indem er direkten physikalischen Zugriff auf das Netzwerk besitzt oder Admininstratorzugriff hat oder sich auf das System über Hintertüren, sogenannte Trojaner[6], Zugriff verschafft. Findet die Kommunikation über die Datenleitungen unverschlüsselt statt, ist es natürlich noch einfacher, etwas mit den gelesenen Daten anzufangen.

Schreiben der Kommunikation: Notwendig ist auch hier, wie beim Lesen der Kommunikation, ein Zugriff auf die Datenleitung. Über eine sogenannte Spoofing-Attacke schreibt (manipuliert) der Angreifer die Kommunikation. D.h. er leitet den Datenstrom über seinen eigenen Netzwerkserver um, indem er sich als (fälschlicher Weise) korrekter Kommunikationspartner ausgibt, verändert die ankommen Daten des Senders und schickt sie an den ursprünglichen Empfänger weiter.

[6]Der Angreifer schleust ein Programm in das fremde Netzwerk ein, welches Daten in diesem Netzwerk liest oder und diese dem Angreifer übermittelt. Vgl. dazu die Geschichte über das Trojanische Pferd...

Ausfall der Kommunikation erzielen: Der einfachste Weg ist die Trennung der physikalschen Netzwerkverbindung durch einen Angreifer mit physikalischem Zugriff auf das Netzwerk. Besitzt der Angreifer Administratorzugriff auf den Webserver, kann er die entsprechenden Services die die Kommunikation mit der Außenwelt regeln, abschalten. Die dritte Möglichkeit ist ein sogenannter Denial-Of-Service-Angriff. Der Angreifer stellt in kurzer Zeit soviele Anfragen an den Webserver, bis dieser überlastet ist. Die Folge ist, dass der Webserver keine weiteren Anfragen anderer Benutzer beantworten kann. Dieses kommt einem Ausfall der Kommunikation gleich. Weiterhin kann der Webserver unter der Überlast abstürzen, fällt somit aus, was eine Kommunikation mit der Welt ebenso unmöglich macht.

4.2.3 Angriff auf den Datenbankserver

Ziel eines Angriffs auf den Datenbankserver ist erstens sensible Daten zu lesen oder zu schreiben (verändern). Und zweitens (Administrator-)Zugriff auf den Datenbankserver zu erlangen, um diesen in den Fehlerzustand Ausfall zu überführen. Erreichbar sind diese Ziele über folgende Wege:

Datenbank lesen/schreiben: Der Angreifer benötigt Datenbankserverzugriff. Dieses erreicht er indem er bekannte Sicherheitslücken (z.B. URL-Hacking) ausnutzt, Administratorzugriff erlangt oder auf einen Datenbankserver ohne Authentifizierungsmechanismen trifft. Hat er nun zusätzlich SQL-Kenntnisse, kann er direkt die Tabellen der Datenbank lesen oder schreiben.

Ausfall der Servers herbeiführen: Er attackiert den Datenbankserver selbst, wie beim Angriff auf die Kommunikationsstrukturen (Ziel Ausfall). Oder er überschreibt die Datenbank und macht sie dadurch unbrauchbar. Dieses würde einem Ausfall des Servers gleichkommen, da nicht mehr vorhandene Daten nicht mehr übermittelt werden können.

4.2.4 Angriff auf den Webserver

Dieser Angriff hat das Ziel, den Webserver zum Ausfall zu bringen und somit die Kommunikation des Systems mit der Außenwelt zu unterbrechen. Erreichbar ist dieses durch:

Zugang besitzen Der Angreifer besitzt physikalischen Zugang, und trennt den Webserver vom Netz. Oder er erlangt Administratorzugriff und fährt den Server runter, oder verändert die Konfigurationsdaten des Servers dahingehend, dass dieser sofort oder zu einem späteren Zeitpunkt ausfällt.

Denial-Of-Service Wurde schon unter Angriff auf die Kommunikationsstrukturen (Ziel Ausfall) beschrieben.

4.3 Die Angriffsbäume

Auf den nächsten Seiten werden die oben unformal beschriebenen Angriffe durch die Darstellung als Angriffsbäume ergänzt:

1. Angriff zur Erlangung von Administratorrechten: Abbildung 4.1 auf der nächsten Seite.
2. Angriff auf die Kommunikationsstrukturen: Abbildung 4.2 auf Seite 34.
3. Angriff auf den Datenbankserver: Abbildung 4.3 auf Seite 35.
4. Angriff auf den Webserver: Abbildung 4.4 auf Seite 36.

Zur Semantik und Notation der Angriffsbäume verweise ich an dieser Stelle nochmal auf Kapitel 3.2 auf Seite 19.

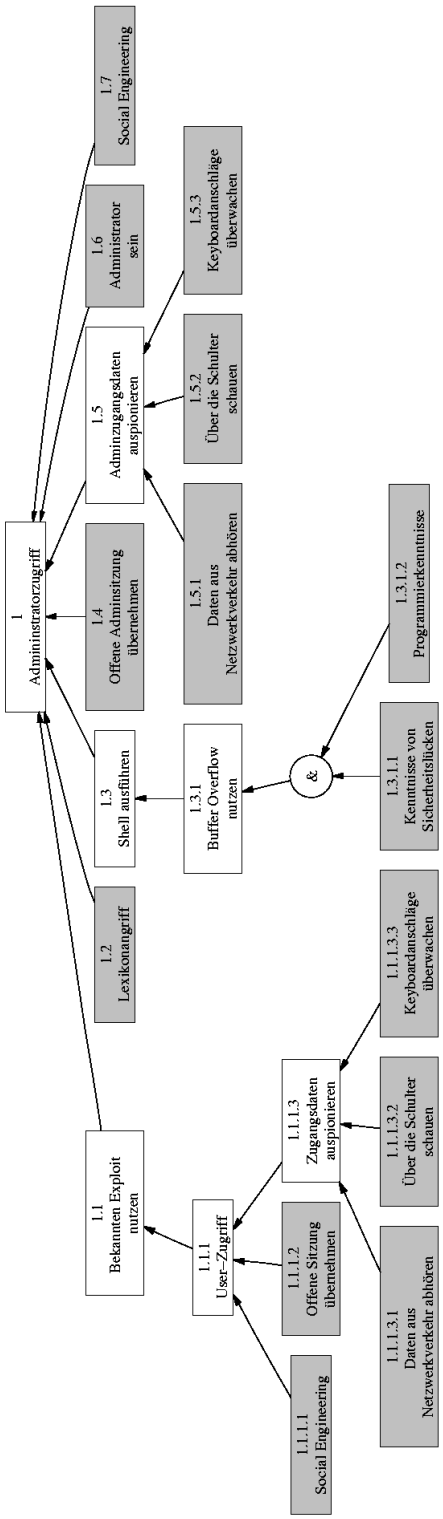


Abbildung 4.1: Angiffsbaum zur Erlangung von Administratorzugriff

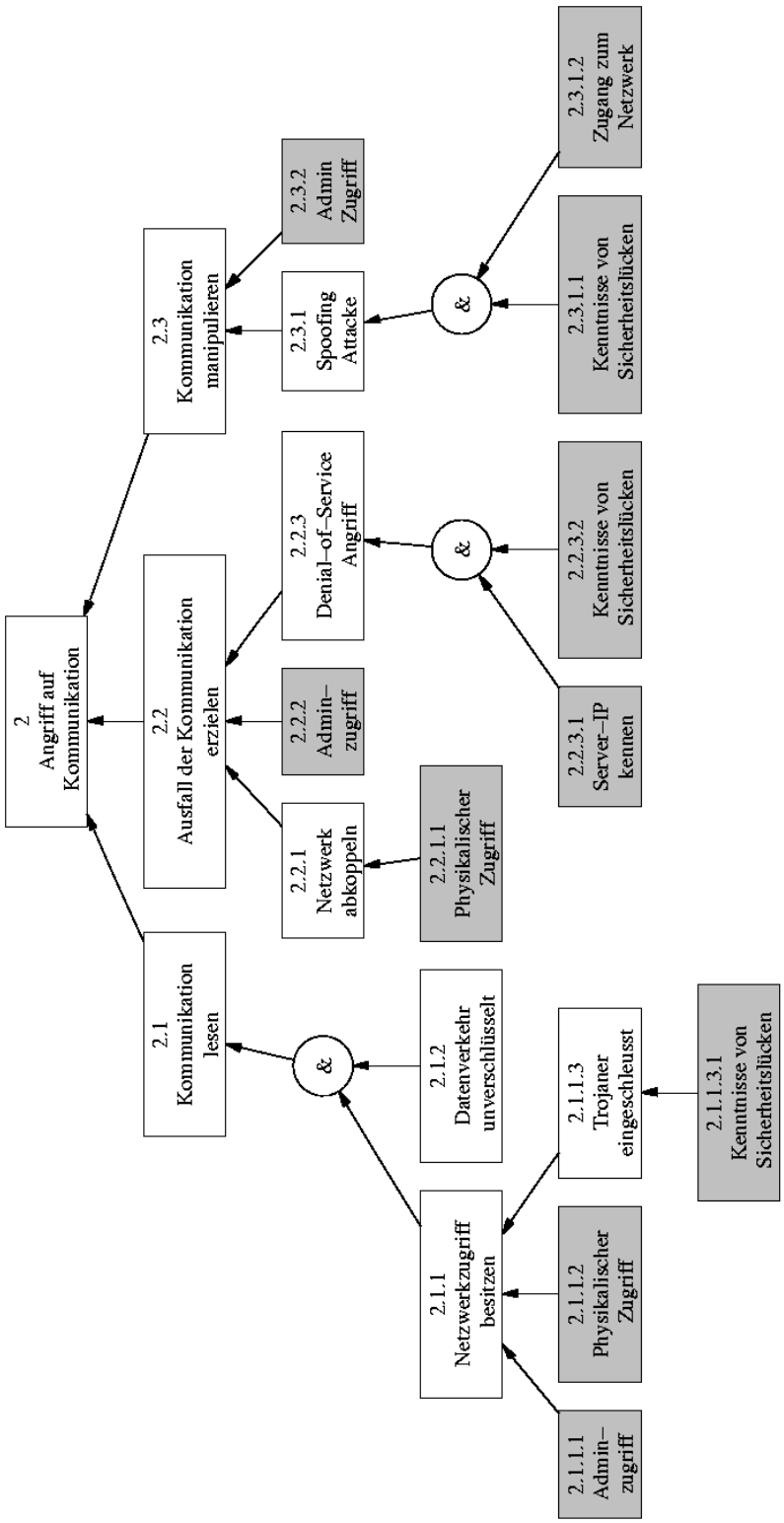


Abbildung 4.2: Angiffsbaum zum Angriff auf Kommunikationsstrukturen

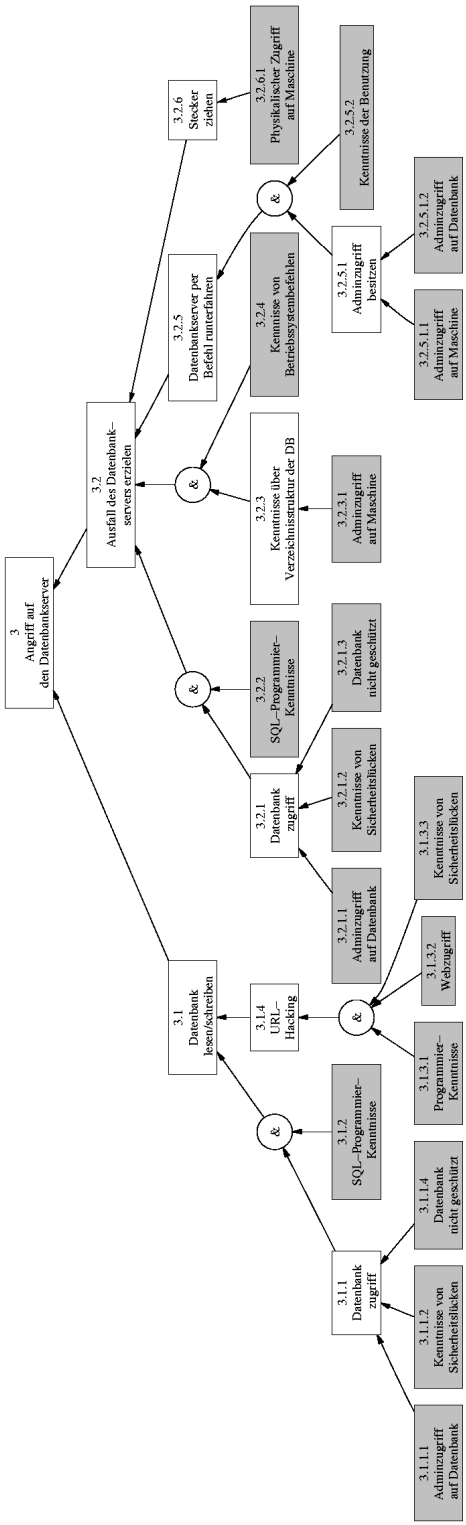


Abbildung 4.3: Angiffsbaum zum Angriff auf die Datenbasis

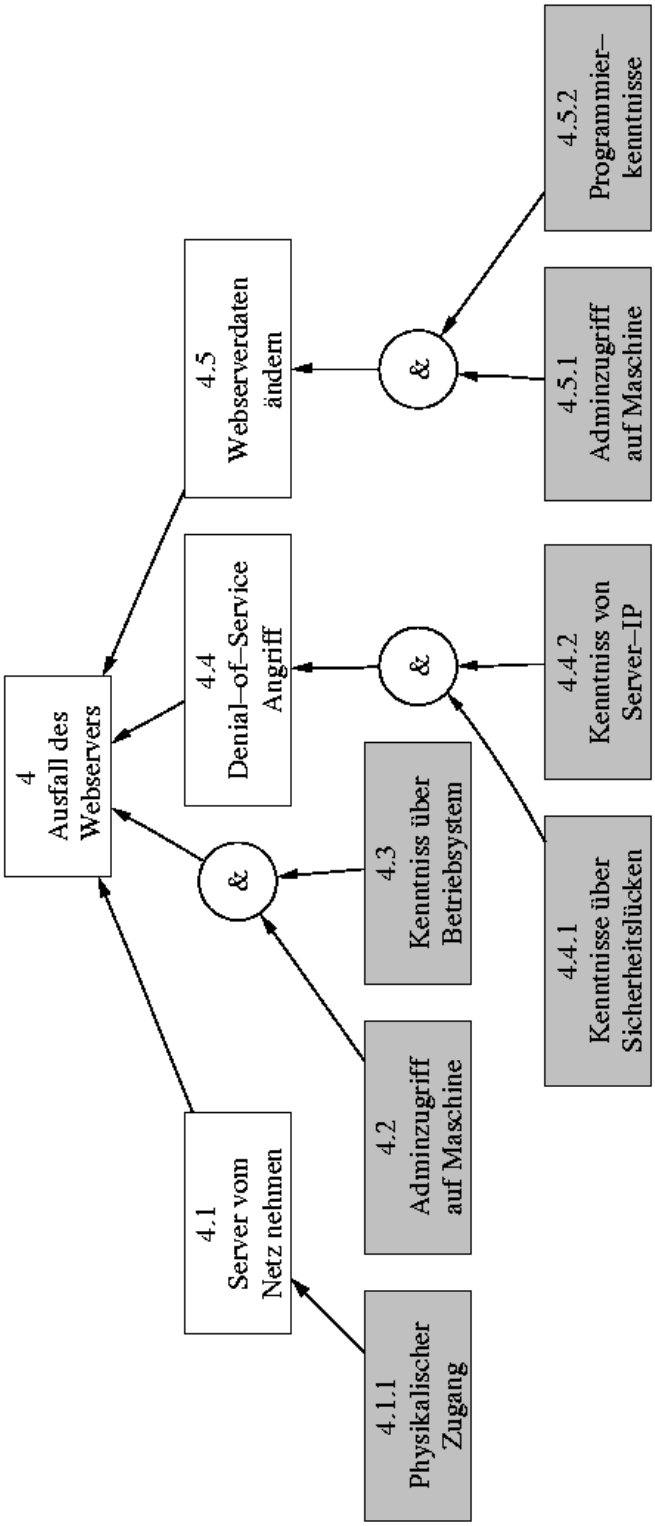


Abbildung 4.4: Angiffsbaum zum Angriff auf den Webserver

4.4 Bewertung der Analyse

Um die Analyse abzuschließen beurteilte ich, welche Angriffe in Bezug auf die Sicherheitsbedürfnisse eines CSCL-Systems sicherheitskritisch sind. Es soll ein Überblick über sinnvolle und notwendige Sicherheitsmechanismen in CSCL-Systemen gewonnen werden.

4.4.1 Sicherheitsbedürfnisse von CSCL-Systemen

CSCL-Systeme sind netzwerkbasierte Systeme. Sie speichern und verwalten persönliche/sensible sowie sonstige Datenobjekte in Datenbanken. Skripte laden die Daten vom Server und Webbrowser stellen sie durch die Anweisungen der Skripte dar. Dieses sollte jederzeit möglich sein. Demnach ergeben sich drei grundlegende Sicherheitsbedürfnisse für CSCL-Systeme:

1. Vertraulichkeit, vorallem bzgl. der persönlichen Daten und Kommunikation.
2. Integrität der Daten vor dem schreiben unautorisierter/unauthentifizierter Dritter.
3. Verfügbarkeit des Systems zu jeder Zeit.

Tabelle 4.1 auf Seite 39 stellt in einer Übersicht dar, welche Angriffe auf welche Sicherheitsdienste verletzen.

4.4.2 Bewertung der Angriffsszenarien

Fast alle Angriffsszenarien zielen auf Verletzung von Vertraulichkeit und Authentifizierung und Integrität (vgl. Tabelle 4.1 auf Seite 39). D.h., Angreifer lesen oder schrieben Daten oder führen Programmdaten aus. Mögliche Angriffsziele sind dann:

- persönliche Daten
- sensible Daten, im Besonderen Prüfungsdaten und gespeicherte Studienleistungen
- Daten, durch dessen Veränderung weiterer Schaden angerichtet wird; z.B. Webserverskripte, SQL-Skripte

Da hier alle sensiblen Daten innerhalb des CSCL-Systems direkt oder indirekt betroffen sind, ist die Art solcher Angriffe als besonders kritisch zu bewerten. Weitere Angriffe auf Vertraulichkeit und Authentizität betreffen die Netzwerkkommunikation:

- lesen (auspionieren) von Zugangsdaten
- lesen der allgemeinen Kommunikation über Datenleitungen

- schreiben (manipulieren) der Netzwerkkommunikation

Das Lesen und Schreiben von Netzwerkkommunikation kann u.U. aufwendig sein. Auch ist von Fall zu Fall zu unterscheiden, ob die Daten die über das Netzwerk verteilt werden, wirklich sicherheitsbedürftig sind. Zugangsdaten jedoch müssen immer sicher übertragen werden, um keine gefährliche Schwachstelle im System zu schaffen (vgl. dazu [Schneier 2001, Anderson 2001]).

Die restlichen Angriffe richten sich gegen die Verfügbarkeit (vgl. Tabelle 4.1 auf der nächsten Seite) des Systems. Das System soll prinzipiell jederzeit verfügbar sein. Online stattfindende Prüfungen sind besonders kritische Situationen. Hier *muss* das System ausfallsicher sein. Im sonstigen Betrieb ist die Zeit entscheidend, in der das System wieder gestartet werden kann. Einige Minuten sind bei einem CSCL-System tolerierbar, da die Daten während des normalen Lehrbetriebs nicht zeitkritisch abgerufen werden müssen.

4.4.3 Notwendige Sicherheitsmaßnahmen und Lehrbetrieb

Aufgrund heute existierender Sicherheitsmechanismen in der Computertechnik können viele Sicherheitsprobleme effizient und ohne den Lehrbetrieb zu stören eingedämmt werden. Allerdings reichen die in [Schulmeister 2003] geforderten Mechanismen Authentifizierung und verschlüsselte Übertragung für CSCL-Systeme nicht aus, um *Security* zu gewährleisten, zumal verschlüsselte Übertragung bei den meisten CSCL-Systemen nicht implementiert wurde [Schulmeister 2003].

Da laut [Anonymus 1999] ein großer Teil aller Angriffe auf Computersysteme auf Social-Engineering beruht, gilt es die Benutzer für Sicherheitsprobleme zu sensibilisieren. Wichtigster Punkt ist die Sicherheit und Geheimhaltung von Zugangsdaten, da diese eine Basis für jedes Angriffsszenario darstellen.

Es empfielt sich Daten stets verschlüsselt zu übertragen, damit nicht ungewollt sensible Daten im Klartext an unbefugte Dritte gelangen. Ergo sollte das CSCL-System auf einem Webserver aufbauen, der Verschlüsselung unterstützt.

Die Systemadministration bildet ein Computersicherheitsteam (CST). Das CST ist zuständig für die Aufstellung von systemweit gültigen Sicherheitsrichtlinien. Diese geben Auskunft über die Verhaltensweise bei Angriffen oder Ausfällen der Server und erläutern den Benutzern die sichere Benutzung des Systems.

Ebenso installiert das CST ein Backup-System zur regelmäßigen Sicherung der Daten der Datenbank und ein Sekundärsystem das innerhalb kurzer Zeit den Ausfall des Primärsystems überbrückt. Alle Server sind mit einer Firewall geschützt, die auf die entsprechenden Bedürfnisse zugeschnitten und korrekt konfiguriert ist.

Die oben genannten Mechanismen beeinträchtigen den Lehrbetrieb kaum, aber helfen die kritischsten Sicherheitsprobleme einzudämmen. Allerdings gewähren auch sie keine absolute Sicherheit. Es gibt immer wieder neue Sicherheitsprobleme, basierend auf Programmierfehlern oder Ideen findiger Cracker. Deshalb

müssen die Sicherheitsmechanismen up-to-date gehalten werden, um hohe Sicherheit zu gewährleisten (siehe Abbildung 3.3 auf Seite 23).

Angriff	VER	AUT	INT	VFK
1. Angriff auf den Webserver				
a) Server zerstören	√	√	√	√
b) Server deaktivieren				√
c) Server manipulieren	√	√	√	√
2. Angriffe auf die Datenbasis				
a) Daten Zerstören	√	√	√	√
b) Daten manipulieren	√	√	√	
c) Daten lesen	√	√		
d) Datenbankserver deaktivieren				√
3. Angriffe auf die Kommunikation				
a) Abhören	√	√		
b) Deaktivieren				√
c) Manipulieren	√	√	√	

Tabelle 4.1: Angriffszenarios und entsprechend verletzte Sicherheitsdienste Vertraulichkeit (VER), Authentifizierung (AUT), Integrität (INT), Verfügbarkeit (VFK)

4.5 Kritik der Analysemethode

Wie schon in Kapitel 3.2 auf Seite 19 erwähnt liegt der Schwachpunkt der Analyse darin, dass sie von der Expertise des Analysten, in diesem Fall von mir, abhängt. D.h., die Vollständigkeit der Analyse kann ich nicht beweisen. Die Begründung des Ergebnisses der Analyse ist auf unformalem Wege dadurch gewährleistet, dass sie in einem klar definierten System stattfand, jeder Angriffsschritt erläutert wurde und die Analyse damit durch Dritte nachvollziehbar ist.

Der Vorteil der von mir verwandten Methode der Angriffsbäume ist jedoch, dass etwaige neue oder mir unbekannte Angriffsszenarios problemlos in den Angriffsbaum eingefügt werden können.

Kapitel 5

Einleitung

Die Varianten möglicher Lernerfolgskontrolltests innerhalb von CSCL-Werkzeugen sind vielfältig. Die Palette reicht von Multiple-Choice-Tests, über Lückentests, bis hin zu Tests mit natürlichsprachlicher Eingabe von Antworten.

Von all diesen Tests stellt der Multiple-Choice-Test (MCT) eine einfache und am Computer gut auszuwertende Alternative dar. In Kapitel 2.1.3 auf Seite 7 und Kapitel 2.1.4 auf Seite 9 habe ich festgestellt, dass ein in der Lernerfolgskontrolle eingesetzter MCT bestimmten Anforderungen genügen muss, um einen didaktischen Mehrwert und gute Bedienbarkeit zu gewährleisten.

Dazu gehört nicht nur eine strukturierte Darstellung am Bildschirm und leichte Bedienbarkeit des Moduls. Sondern das Modul ist echt modular in die Anwendung integriert und ermöglicht eine flexible Nutzung. Tests sind einfach und schnell erstellbar. Die Testdurchführung ist flexibel gestaltet, indem Fragen in verschiedenen Formen wiederholt werden können. Zusätzlich werden weiterführende Erklärungen und Referenzen zu der gestellten Frage bei Bedarf angeboten.

Da die Implementation des Testmoduls Sicherheitsmerkmale aufweisen soll, muss festgellt werden, welchen Sicherheitsstandards ein Modul zur Lernerfolgskontrolle innerhalb eines CSCL-Werkzeugs genügen muss.

Gliederung Teil II.

In Kapitel 6 auf der nächsten Seite spezifiziere ich das MCT-Modul. Die Spezifikation beinhaltet die Anforderungsanalyse, sowie mögliche Anwendungsfälle für das Modul. In Kapitel 7 auf Seite 57 analysiere ich das Modul auf Basis der Spezifiaktion im Hinblick auf mögliche Sicherheitsprobleme und erläutere deren Konsequenzen für die Implementierung des Moduls. Kapitel 8 auf Seite 63 beschreibt die wichtigsten Implementationsdetails des Moduls.

Kapitel 6

Spezifikation des Multiple-Choice-Testers zur Lernerfolgskontrolle

Dieses Kapitel beschreibt die benutzerspezifischen Anforderungen der Akteure an das Multiple-Choice-Testmodul. Dieses wird den Namen *tQuest* erhalten und im folgenden auch so genannt.

Die folgende Spezifikation des Moduls bzw. Softwaresystems[1] ist allerdings keine vollständige Beschreibung eines Softwareentwicklungsprozesses. Sie stellt nur die Informationen dar, die für das Verständnis der Form der Implementation vonnöten sind.

6.1 Anforderungen der Akteure

Bevor das Design und die Implementation des Systems beginnen kann, ist es wichtig, die genauen Anforderungen der Akteure an das Systems zu kennen. Diese gilt es im Rahmen der sogenannten Anforderungsanalyse festzustellen [Koreimann 1992, Schneider & Werner 2001].

Fehler, die in der Anforderungsanalyse nicht entdeckt und beseitigt werden, pflanzen sich im weiteren Entwicklungsprozess vielfach gravierender fort und können deshalb spätestens bei der Implementation erhebliche Probleme produzieren.

6.1.1 Akteure

Das Modul wird von verschiedenen Akteuren verwendet. Jedem Akteur wird eine Rolle innerhalb des Systems zugewiesen.

Die Rolle des Akteurs im System beschreibt das Wesen und die Art der Interaktion des Akteurs mit dem System. Die Anforderungen des Akteurs an das

[1] Im folgenden nur System genannt

System orientieren sich somit an seiner Rolle im System und den damit verbundenen Art der Interaktion mit dem System.

In der Spezifikation des Systems werden drei Rollen unterschieden – Administratoren, Studenten und Sonstige – deren Interaktion mit dem System und die daraus resultierenden Anforderungen an tQuest im weiteren beschrieben werden[2].

Systemadministratoren

1. Allgemeine Rolle im System
   - verwalten das allgemeine CSCL-Werkzeug
   - verwalten und warten alle Module des CSCL-Systems und sorgen für die Integration neuer Module in das CSCL-System
   - haben Zugriff auf das gesamte System
   - kommunizieren mit den Benutzern des Systems
2. Daraus implizierte spezielle Anforderungen an tQuest
   - einfache Integration von tQuest in das CSCL-Werkzeug
   - einfache Administration tQuest innerhalb des CSCL-Werkzeugs

Studenten

1. Allgemeine Rolle im System
   - bearbeiten Lerninhalte alleine oder in Gruppen
   - bearbeiten Testfragen: freiwillig zur persönlichen Kontrolle oder in Prüfungen
   - erwarten Feedback zu ihrer Leistung
   - erwarten faire Testfragen
   - kommunizieren mit anderen Benutzern des Systems
2. Daraus implizierte spezielle Anforderungen an tQuest
   - leichte Benutzbarkeit von tQuest bei Tests
   - Steigerung des Lerneffekts durch zusätzliche Informationen und Wiederholungsmöglichkeiten der Fragen
   - Auswertung der Leistung am Ende des Tests
   - tQuest soll jederzeit freiwillig und unabhängig von direkten Lerneinheiten benutzt werden können, d.h. es soll wie z.B. bei einem Messageboard einen direkten Zugang zu verschiedenen Tests zur Lernkontrolle geben

[2]Nähere Beschreibungen zu den Rollen finden sich auch in Kapitel 4.2 auf Seite 29

Sonstige

... z.B. Dozenten oder Tutoren.

1. Allgemeine Rolle im System

   - bereiten Online-Lerninhalte auf und stellen diese für die Studenten in der Lernumgebung bereit
   - erstellen Testfragen, Testantworten und Hilfen zu Testfragen um damit Online-Tests für die Studenten zur Lern- oder Leistungkontrolle bereitzustellen
   - verwalten alle für die Onlinelehre notwendigen Daten mit dem CSCL-Werkzeug
   - kommunizieren mit anderen Benutzern des Systems
   - geben Studenten Hilfestellungen

2. Daraus implizierte spezielle Anforderungen an tQuest

   - leichtes einfügen von Testfragen, Testantworten und Hilfen für Testfragen
   - leichtes erstellen von Tests: automatisch sowie manuell
   - einfache Mechanismen um die Tests entsprechenden Studentengruppen bereitzustellen
   - einfache Benutzbarkeit von tQuest

6.1.2 Allgemeine Anforderungen an tQuest

Die eben festgestellten Anforderungen der Akteure an tQuest und der in Kapitel 2.1.4 auf Seite 9 beschriebene Aufbau eines Multiple-Choice-Tests sind maßgeblich für die weitere Realisation des Moduls. Der Querschnitt dieser Anforderungen an tQuest fasse ich nun in einer Übersicht zusammen.

Ergänzt wird diese Liste durch technische Details für die Implementation meines Projekts im Rahmen meiner Diplomarbeit. Die allgemeinen Anforderungen an tQuest lassen sich in folgende Kategorien gliedern:

1. Benutzung

   - Testfragen, Testantworten und Hilfestellungen sind einfach erstellbar
   - Tests sind einfach zu erstellen: die Erstellung kann automatisch sowie manuell geschehen, Hilfestellungen können eingefügt werden
   - der Testablauf ist je nach Anforderung variabel gestaltbar:
     - falsche Fragen können direkt oder am Ende des Tests oder garnicht wiederholt werden

  - Rückmeldungen (Antwort richtig oder falsch) nach Testfragen können optional eingeschaltet werden
  - Hilfestellungen zu Fragen können jederzeit oder garnicht eingesehen werden
- am Testende folgt eine Auswertung der Leistung

2. Zugriffsrechte

- tQuest besitzt eine eigene Rechteverwaltung, die die Umsetzung der oben beschrieben Rollen über Zugriffsrechte ermöglicht
- es können Zugriffsrechte für Teilmodule und Tests vergeben werden
- die Rechteverwaltung orientiert sich an den globalen Rechten des Basis-CSCL-Werkzeugs und bietet eine dementsprechende Schnittstelle

3. Integration

- tQuest ist leicht in das als Basis benutzte CSCL-Werkzeug integrierbar
- fertige Tests sind einfach innerhalb des CSCL-Werzeugs bereitstellbar und direkt einfach abrufbar
- tQuest ist weitestgehend autonom von den restlichen Modulen des CSCL-Werkzeugs und direkt zugänglich; bisher ist allerdings mindestens die Rechteverwaltung mit dem Basis CSCL-Werkzeug verbunden (Details hierzu in Kapitel 6.3.5 auf Seite 50 und Kapitel 8.2.3 auf Seite 82)

4. Technik und weiter Details

- das zu benutzende und als Basis dienende CSCL-Werkzeug ist die webbasierte Worksphere[3]
- die Rechteverwaltung von tQuest basiert auf den Benutzergruppen der Worksphere
- Programmiersprachen sind PHP4, HTML und Javascript
- die Test- und Programmdaten werden in einer MySQL-Datenbank und in Textdateien abgelegt
- tQuest ist ersteinmal nur zur Lernerfolgskontrolle vorgesehen

6.2 Anwendungsfälle

Aus den im vorigen Kapitel aufgestellten allgemeinen Anforderungen lassen sich im Detail vier zentrale Anwendungsfälle für tQuest ableiten:

[3]Worksphere: http://www.worksphere.de

1. Erstellen und Verwalten von Testfragen und Themen
   (a) Testinhalte einfügen.
   (b) Testinhalte ändern oder löschen.
   (c) Themen einfügen und löschen.
2. Erstellen und Verwalten von Tests
   (a) Tests automatisch erstellen lassen.
   (b) Tests manuell erstellen.
3. Durchführung von Tests.
4. Verwalten von Zugriffsrechten.

Diese werden im weiteren näher erläutert und durch schematischen Abbildungen in ihren Ablaufsvarianten dargestellt.

Die Abbildungen

Die Symbole in den Abbildungen besitzen folgende Semantik:

- Kanten
  - der am Startzustand beginnende Pfad mit durchgezogenen Kanten beschreibt den optimalen Pfad durch den Anwendungsfall
  - gestrichelte Kanten beschreiben jeweils optionale Möglichkeiten für den Benutzer an entsprechender Stelle im Anwendungsfall
  - gestrichelte Kanten mit einem Punkt am Ende statt einer Pfeilspitze verweisen auf einen weiteren Anwendungsfall, den der Benutzer von dieser Stelle aus durchführen kann. Diese stelle ich aus Gründen der Übersichtlichkeit in einer anderen Abbildung dar
- Knoten
  - rechteckige grau unterlegte Knoten sind Startzustände und bezeichnen die Akteure die das Recht haben diesen Anwendungsfall durchzuführen
  - elipsenförmige Knoten beschreiben die möglichen Schritte des Anwendungsfalls die der Benutzer vollziehen kann
  - sechseckige Knoten beschreiben
    * grau unterlegt: bidirektionale Verzweigungen die eine Entscheidung im Programmablauf erfordern
    * weiß unterlegt: die getroffene Entscheidung
  - rautenförmige grau unterlegte Knoten beschreiben den definitiven Abschluss des Anwendungfalls

6.2.1 Erstellen und Verwalten von Testfragen und Themen

Dieser Anwendungsfall beschreibt, wie Testfragen eingegeben werden können. Es besteht die Möglichkeit Fragen einzugeben, zu ändern oder zu löschen.

Die Eingabe von Daten und das Löschen und Ändern von Daten stellen aufgrund ihrer Ablaufstruktur jeweils einen eigenen Anwendungsfall dar. Mögliche Akteure der Anwendungsfälle sind Administratoren und Sonstige. Für die schematischen Abbildungen dieser Anwendungfälle siehe Abbildung 6.1 auf Seite 52 und Abbildung 6.2 auf Seite 53.

Um Fragen einzugeben müssen zuerst entsprechende Themen in die Datenbank eingepflegt werden. Dieses geschieht in der Themenverwaltung. Wegen der Einfachheit dieses Anwendungfalles wird er nicht als Abbildung dargestellt.

6.2.2 Erstellen und Verwalten von Tests

Testfragen können vom Benutzer (Administratoren oder Sonstige) manuell oder vom Computer automatisch und nach bestimmten Kriterien ausgewählt werden. Nach Beendigung der Testfragenauswahl wird der Name und Ablauf des Tests vom Benutzer festgelegt. Daraufhin kann der Test in der Datenbank gespeichert werden und steht zur Durchführung zur Verfügung.

Um bei der automatischen Erzeugung der Testfragenauswahl den Benutzerwünschen möglichst entgegen zu kommen, benötigt der Testgenerator einige Angaben über:

- den Rahmen des *Schwierigkeitsgrades* der Fragen in Prozent (0% - 100%)
- die gewünschte *Anzahl* der zu stellenden Fragen
- die gewünschte *Struktur* des Tests als Kombination folgender Auswahlmöglichkeiten:
  - mit Rückmeldung (richtige Antwort, falsche Antwort) oder ohne Rückmeldung nach Beantwortung einer Frage
  - mit Hilfestellung oder ohne Hilfestellung zu Fragen
  - mit Fragenwiederholung oder ohne Fragenwiederholung nach falsch beantworteter Frage
  - mit Wiederholung oder ohne Wiederholung aller falscher Fragen am Testende

Bei der manuellen Erstellung von Tests sind nur Angaben über die *Struktur* des Tests notwendig. Die schematische Darstellung dieses Anwendungsfalls ist zu finden in Abbildung 6.3 auf Seite 54.

Anmerkung dazu: die Abbildung enthält zwei Anwendungsfälle und demnach auch zwei optimale Pfade. Die Anwendungfälle beginnen jeweils nach dem Knoten “Testverwaltungsoberfläche öffnen” durch die jeweiligen Kanten zu den Knoten “Testgenerator starten” und “Manuellen Testgenerator starten”.

6.2.3 Durchführung von Tests

Tests werden zwar in der Regel ausschließlich von Studenten durchgeführt, können aber theoretisch von allen Benutzern durchgeführt werden. Das Zugriffsrecht auf einen Test wird in der Rechteverwaltung für Tests eingestellt.

Testabläufe variieren entsprechend der Intention des Tests. Bei Tests zur Lernerfolgskontrolle besteht nach einer falsch beantworten Frage die Möglichkeit Rückmeldungen zu erhalten, die Frage ggf. zu wiederholen und bei Bedarf Hilfestellungen zur Frage in Anspruch zu nehmen.

Bei Leistungskontrolltests wird dieses entfallen und nach einer falsch beantworteten Frage direkt die nächste Frage folgen. Verschiedene Ablaufvarianten sind vorgesehen und können per Option bei der Testerstellung angegeben werden (siehe Kapitel 6.2.2 auf der vorherigen Seite).

Die schematische Darstellung dieses Anwendungsfalls ist zu finden in Abbildung 6.4 auf Seite 55. Die Details über den Anwendungsschritt “Test wird durchgeführt” sind in der Abbildung 6.5 auf Seite 56 zum allgemeinen Testablauf zu finden.

6.2.4 Zugriffsrechte verwalten

Der Administrator hat die Möglichkeit jedem Benutzer ein Zugriffsrecht auf Teilmodule und Tests zuzuweisen. Dieses geschieht durch öffnen der entsprechenden Tabellen und auswählen der Rechte. Wegen der Einfachheit dieses Anwendungfalles stelle ich ihn nicht als Abbildung dar.

6.3 Teilmodule von tQuest

Bei der Beschreibung der Anwendungsfälle sind die Bezeichnungen mehrerer Teilmodule von tQuest genannt worden, die ich nun näher spezifiziere.

6.3.1 Testverwaltungsoberfläche

Ist die zentrale Oberfläche von tQuest. Von hier aus werden alle weiteren Teilmodule gestartet. Die Anzahl der angezeigten Teilmodule hängt von den in der Rechteverwaltung (s.u.) zugewiesenen Zugriffsrechten des Benutzers auf tQuest-Module ab. Insgesamt stehen folgende Teilmodule zur Auswahl:

- Testfragen verwalten (neu, bearbeiten, löschen)
- Themenliste bearbeiten
- Test manuell erstellen
- Testgenerator ausführen
- Tests bearbeiten

- einen freigeschalteten Test auswählen und durchführen
- Zugriffsrechte für tQuest-Teilmodule verwalten
- Zugriffsrechte für Tests verwalten
- Online-Hilfe

6.3.2 Die Testfragenverwaltung

Verwaltet alle Testfragen und ermöglicht das eingeben, löschen und bearbeiten von Testfragen und Themen.

Testfrageneingabemaske

Hier können in ein Formular die notwendigen Daten für eine Testfrage eingegeben werden. Es existieren folgende Felder:

- Thema (Auswahlbox mit Themen der Themenliste)
- Frage (Textfeld)
- Antworten zur Frage, variabel (Textfeld; Antworten per Knopfdruck hinzufügen oder löschen)
- Nummer der richtigen Antwort (Auswahlbox mit Einträgen: 1 bis Anzahl-Antworten)
- Schwierigkeitsgrad (Auswahlbox mit Einträgen 1 - 100 in 5er Schritten)
- Hilfestellungen (Textfeld)

Jede Frage hat genau eine richtige Antwort.

Testfragentabelle

Zeigt in einer Tabelle alle Testfragen und zugehörige Daten im Überblick an. Von hier aus kann der Benutzer Fragen neu eingeben, löschen oder bearbeiten. Die Fragen in der Tabelle sind nach dem Tabellenfeld “Thema” sortiert.

Zum Bearbeiten oder Löschen der Fragen öffnet sich die Testfrageneingabemaske. Zum Bearbeiten der Frage ist sie editierbar, zum Löschen der Frage ist sie nicht editierbar.

Themenliste

Jeder Frage wird ein Thema zugewiesen. Themen müssen vorab in die Themenliste eingepflegt werden und stehen dann bei der Eingabe der Fragen zur Verfügung. Die Themenliste bietet folgende Funktionalität:

- neue Themen eingeben (Textfeld mit Knopf)
- Themenliste nach einem Ausdruck filtern (Textfeld mit Knopf)
- mehrere Themen auf einmal aus der Themenliste löschen (markieren über Checkboxen; per Knopfdruck löschen)

6.3.3 Testverwaltung

Dieses Teilmodul wird zum Erstellen und Bearbeiten von Tests benötigt. Der Benutzer kann entscheiden, ob der Computer automatisch oder er selber manuell die Fragen für den Test auswählen möchte. Ausgewählte Fragen können wieder aus der Auswahl entfernt werden.

Die Reihenfolge der Fragen im Test entspricht der Reihenfolge der Auswahl der Fragen. Sie kann aber vom Benutzer nach Abschluss der Auswahl verändert werden.

Beim speichern ist es notwendig den Testnamen sowie den Testablauf (s. Kapitel 6.2.2 auf Seite 46) festzulegen. Zugriffsrechte für Tests können, müssen aber nicht, an dieser Stelle zugewiesen werden. Dieses kann auch in der Testrechteverwaltung geschehen.

Manueller Testgenerator

Ermöglicht die manuelle Auswahl von Testfragen. Die Fragen können aus der Testfragentabelle auswählt werden. Ausgewählte Fragen werden farblich markiert. Alle ausgewählten Fragen können in einer Übersicht angezeigt werden.

Automatischer Testgenerator

Erstellt automatisch und zufällig eine Liste von Testfragen aus den existierenden Fragen auf Basis von Angaben zur Teststruktur (s. Kapitel 6.2.2 auf Seite 46). Nach der Fragenauswahl liefert der Generator eine Auswertung. Ab dann wird verfahren wie bei der manuellen Testerstellung.

Tests bearbeiten

Bereits gespeicherte Tests können im nachhinein bearbeitet werden. Da jeder Test einzigartig bleiben soll, muss ein bearbeiteter Test unter einem anderen Namen gespeichert werden.

Jeder Test soll deshalb einzigartig belieben, damit ein erstellter Test nicht im nachhinein von Dritten manipuliert werden kann (z.B. der Schwierigkeitsgrad)

und demnach der Testdesigner sicher sein kann, dass es sich bei dem erstellten um seinen Test handelt.

6.3.4 Testauswahlmenu

Ermöglicht auswählen und durchführen von zuvor erzeugten und gespeicherten Tests. Die Zugriffsmöglichkeit der auswählbaren Tests hängt von den Zugriffsrechten des Benutzers ab. Der Testablauf richtet sich nach der bei Speicherung des Tests vorgegebenen Ablaufstruktur.

Während der Durchführung des Tests müssen Antworten über einen Knopf ausgewählt werden. Die ausgewählten Antworten werde farblich markiert. Ist eine Auswahl vorhanden, kann die Frage/Antwort über einen Knopfdruck abgeschlossen werden.

Ist der Test beendet erhält der Benutzer eine Auswertung seines Testergebnisses. Diese enthält:

- Name des Tests
- durchschnittlicher Schwierigkeitsgrad des Tests in %
- Gesamtanzahl der Fragen
- Anzahl falscher und richtiger Antworten in %
- alle falschen und richtigen Fragen mit den jeweils richtigen Antworten in zwei separaten Tabellen

6.3.5 Die Rechteverwaltung

Die einzelnen Teilmodule und Tests werden durch Zugriffsrechte vor unbefugtem Zugriff geschützt.

Zugriffsrechte auf Teilmodule

Jeder Benutzergruppe muss ein Recht für den Zugriff auf einzelne Teilmodule von tQuest zugewiesen werden. Die Rechteverwaltung von tQuest ist gekoppelt an die Rechteverwaltung des übergeordneten CSCL-Werkzeugs, hier der Worksphere. D.h., dort vorhandene Benutzergruppen können in die Rechteverwaltung von tQuest übernommen werden.

Es werden zwei Tabellen angezeigt:

1. Tabelle zum Hinzufügen neuer Gruppen von Extern in tQuest. Es werden ausschließlich die hinzufügbaren Gruppen (Auswahlbox) angezeigt.
2. Tabelle mit den zugewiesenen Rechten der bereits in tQuest eingefügten Benutzergruppen. Diese können dann entsprechend geändert werden.

Jeder in tQuest eingefügten Benutzergruppe können folgende Rechte (per Checkbox) zugewiesen werden:

Tests erstellen: Dürfen auf das Teilmodul "Testverwaltung" zugreifen.

Tests bearbeiten: Dürfen auf das Teilmodul "Testverwaltung/Tests bearbeiten" zugreifen.

Tests ausführen: Dürfen auf das Teilmodul "Testdatenbank" zugreifen; welchen Test sie speziell ausführen dürfen wird in der Rechteverwaltung für Tests festgelegt.

Fragen verwalten: Dürfen auf das Teilmodul "Testfragenverwaltung" zugreifen.

Die Zugriffsbeschränkung bezieht sich speziell auch auf das Ausführen konkreter Skripte.

Zugriffsrechte für Tests

Jedem Test kann ein Zugriffrecht zugewiesen werden. Diese Zugriffrechte basieren auf die in tQuest eingefügten Benutzergruppen des Basis-CSCL-Werkzeugs. Diese Zuweisung kann in diesem Teilmodul vorgenommen werden. Es gibt nur die beiden Rechte:

- Benutzergruppe *darf* den Test ausführen
- Benutzergruppe *darf* den Test *nicht* ausführen

Schnittstelle

Die Spezifikation der Schnittstelle zwischen der tQuest-Rechteverwaltung und der Rechteverwaltung eines Basis-CSCL-Werkzeugs wird in Kapitel 8.2.3 auf Seite 82 gegeben.

6.3.6 Online-Hilfe

tQuest verfügt über ein umfangreiche online Hilfe zu allen Teilmodulen. Diese ist von der Testverwaltungsoberfläche erreichbar und erscheint bei Aufruf in einem externen Fenster.

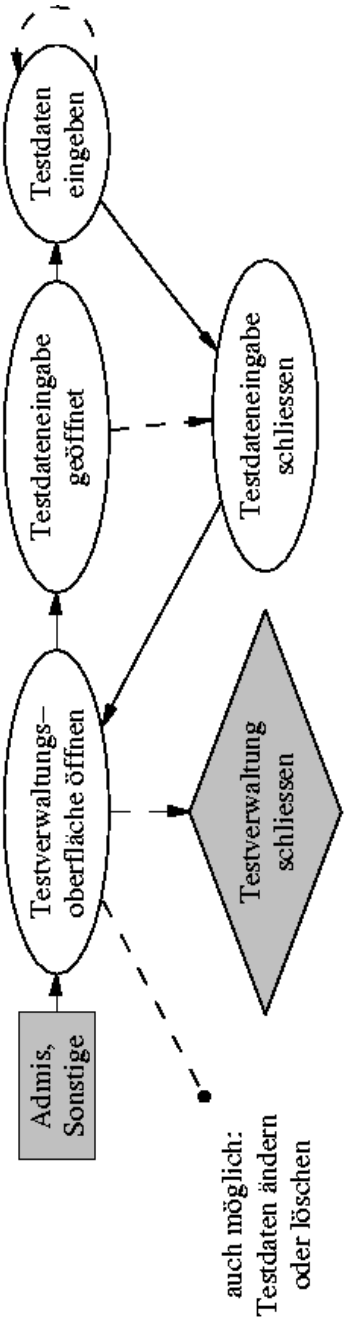


Abbildung 6.1: Schema des Anwendungsfalls “Testinhalte einfügen”

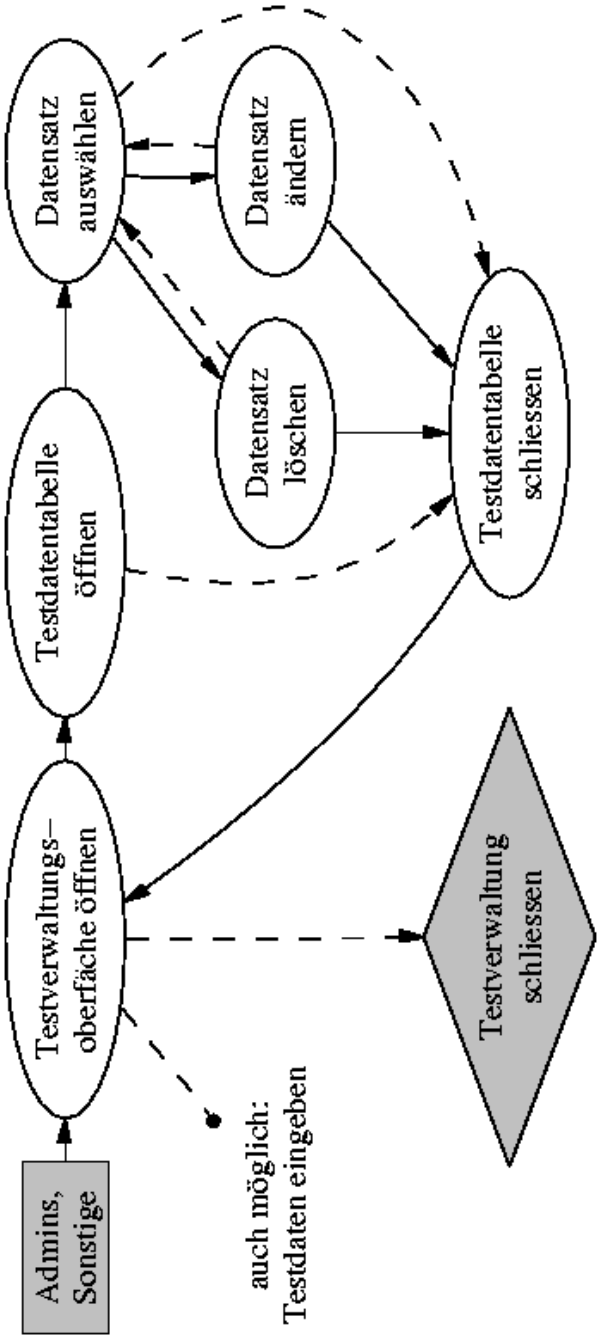


Abbildung 6.2: Schema des Anwendungsfalls “Testinhalte löschen/ändern”

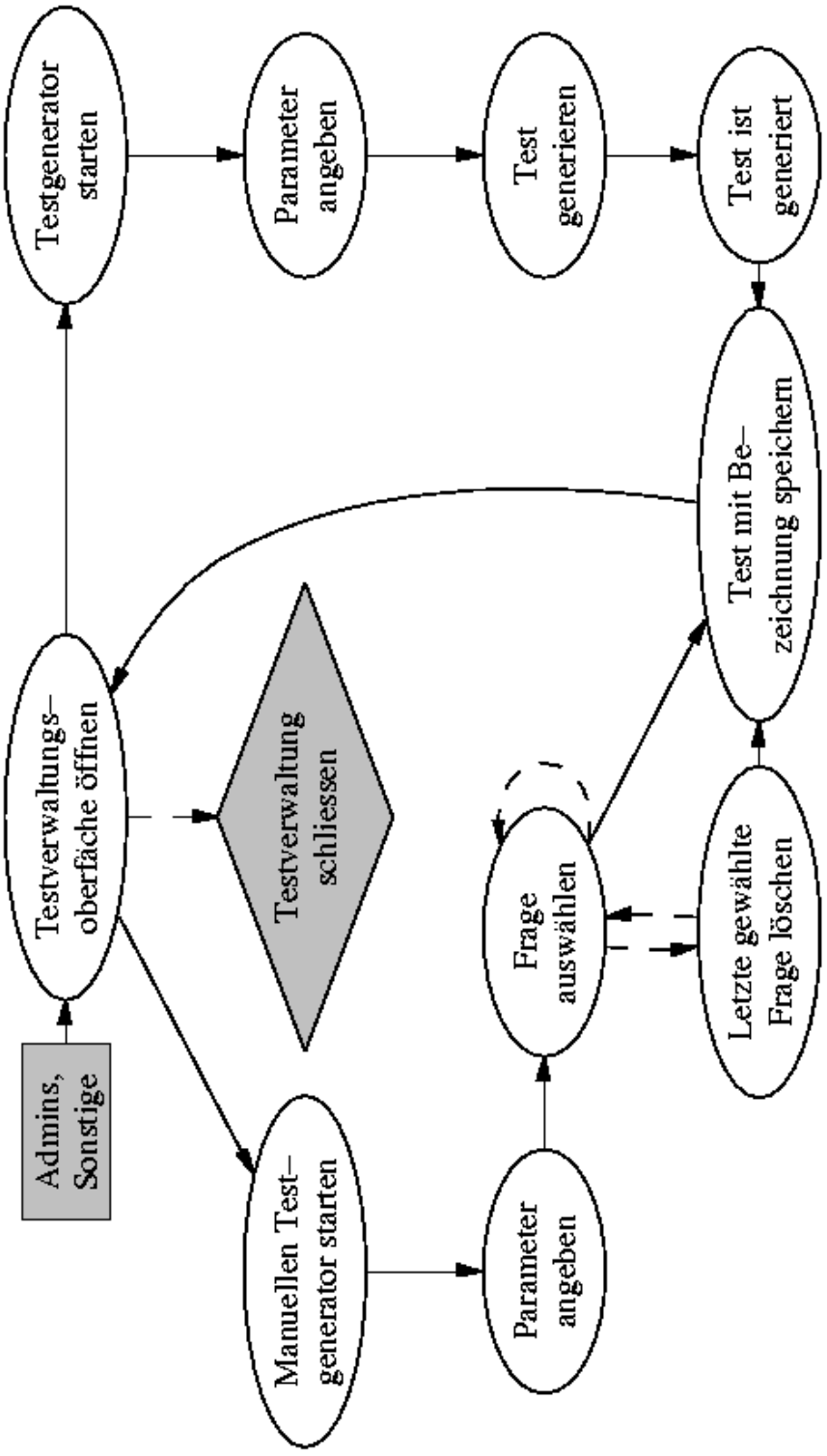


Abbildung 6.3: Schema der Anwendungsfälle “Automatische Testerzeugung” und “Manuelle Testerzeugung”

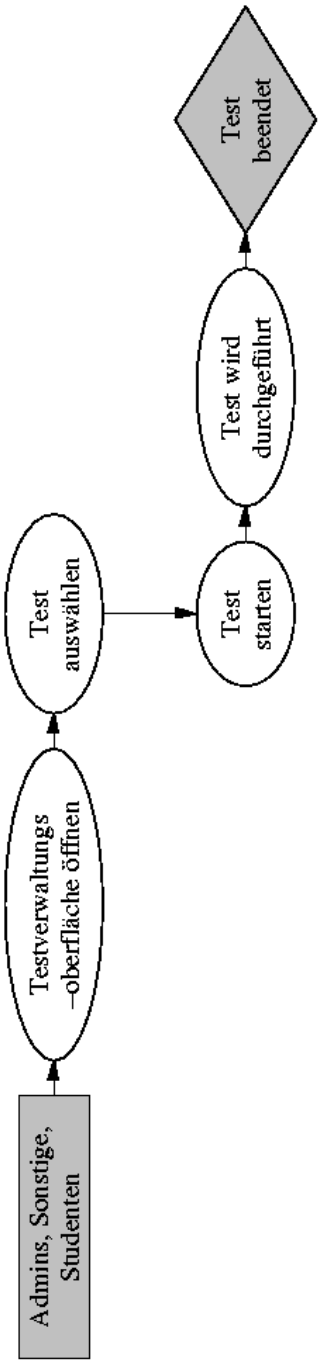


Abbildung 6.4: Schema des Anwendungsfalls “Test durchführen”

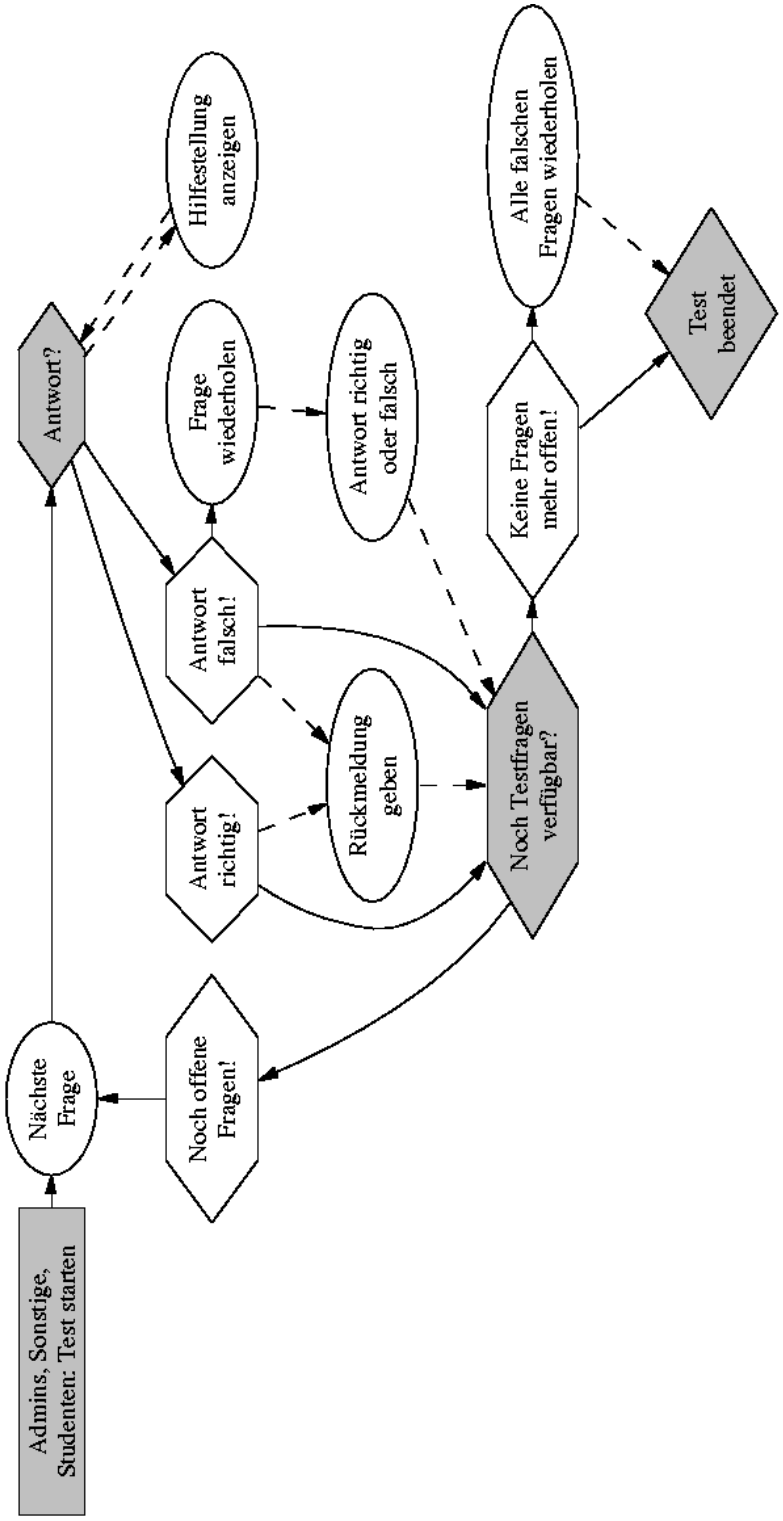


Abbildung 6.5: Schematische Darstellung eines allgemeinen Testablaufs

Kapitel 7

Sicherheitsanalyse des Multiple-Choice-Test-Moduls

Bei der Betrachtung der notwendigen Sicherheitsmechanismen für das Multiple-Choice-Test-Moduls (MCTM) müssen zwei Fälle unterschieden werden: wird das Modul nur zur Lernerfolgskontrolle benutzt, oder im Rahmen von Leistungskontrolltests für studienrelevante Prüfungsleistungen. Beide Fälle erfordern aufgrund unterschiedlicher Voraussetzungen unterschiedlich starke Sicherheitsmechanismen.

7.1 Sicherheit und Lernerfolgskontrolle

Lernerfolgskontrolle zeichnet sich dadurch aus, dass sie freiwillig ist. Sie dient der Verbesserung des persönlichen Lernerfolgs. Aus diesem Grund scheint es unnötig, vom persönlichen Antrieb zur Zerstörung abgesehen, Angriffe auf die Daten des Moduls durchzuführen, da dadurch kein persönlicher Vorteil geschaffen würde.

Selbst wenn ein solcher Angriff durchgeführt wird, ist ein Erfolg nicht kritisch, da ein Testmodul zur Lernerfolgskontrolle keine sensiblen Daten speichert. Wie ein solcher Angriff ablaufen könnte ist in Kapitel 4 auf Seite 24 über allgemeine Sicherheitsanforderungen in CSCL-Systeme und den dort enthaltenen Abbildungen nachzulesen.

Interessanter wird es, wenn das Testmodul zur Leistungskontrolle verwendet wird. Dann werden nämlich sehrwohl sensible Daten verwaltet die ein Ziel von Crackerangriffen sein könnten. tQuest ist zwar nicht als Modul zur Leistungskontrolle geplant, aber es kann dazu erweitert werden. Dehalb führe ich dazu erforderliche Sicherheitsanalyse jetzt schon durch, um die Ergebnisse für die jetzige Implementation bereits im Hinterkopf zu haben.

7.2 Sicherheit und Leistungskontrolle

Wird also das MCTM zur Leistungskontrolle verwendet ergeben sich einige Sicherheitsprobleme, die bei der Implementierung und dem Betrieb des Moduls bedacht

werden müssen.

7.2.1 Angriffsszenarien für das MCTM

Computerbasierte Leistungskontrolle innerhalb von CSCL-Systemen wird aufgrund der Spezifikation zwangläufig online abgehalten. Das heisst die Test- und Ergebnisdaten werden durch Netzwerkkommunikation entweder zum Benutzer oder zum Datenbankserver übertragen. Da eindeutig bekannt sein muss, wer einen Test durchführt, müssen zwangsläufig Authentifizierungsmechanismen eingesetzt werden. Diese Konstellation führt zu folgenden Angriffsszenarien:

1. Angriff auf gespeicherte Test- und Ergebnisdaten
2. Angriff auf die Authentizität der Prüflinge
3. Angriff auf allgemeine Sicherheitsprobleme netzwerkbasierter Anwendungen

Diese werde ich nun näher erläutern.

Angriff auf gespeicherte Testdaten

Gespeicherte Daten sind ein Hauptangriffsziel der meisten Crackerangriffe (vgl. z.B. [Schneier 2001, Anonymus 1999]). Wird das Modul zur Leistungskontrolle verwendet sind besonders gespeicherte Testdaten für Angreifer von Interesse, da diese sowohl die Fragen und Antworten der Prüfungen enthalten, als auch die Ergebnisse der Prüfungen. Diese Daten sind Teil offizieller Studienleistungen.

Ziel ist das Auspionieren von Fragen, deren richtigen Antworten sowie ganzer Tests. Testergebnisse können im Nachhinein "frisiert" werden, um schlechte Leistungen im Ergebnis zu "verbesseren". Ebenso ist vorstellbar, dass die Leistungen von anderen Studierenden "verschlechtert" werden.

Angriff auf die Authentizität des Prüflinge

Wird ein Test Online abgehalten während die Prüflinge sich nicht unter Sichtkontrolle des Testleiters befinden, entsteht das Problem der Authentifizierung. Es ist schwer nachvollziehbar, ob der Prüfling, der die Fragen beantwortet auch wirklich die passende Person ist oder nicht. Beispielsweise könnte ein Student höheren Semesters die Fragen beantworten.

Die Authentizität des Prüflings ist auch durch einen Netzwerk- oder Datenbankangriff gefährdet. Bei einer Online-Prüfung kann z.B. ein "Man-in-the-Middle" (vgl. [Anonymus 1999]), die Daten des Probanden unter dessen Namen verfälschen oder die persönlichen Daten des Probanden durch Manipulation der Datenbasis ändern oder dessen Identität annehmen.

Angriff auf allgemeine Sicherheitsprobleme netzwerkbasierter Anwendungen

Netzwerksysteme sind anfällig für verschieden Arten von Sicherheitsangriffen. Die spezifischen Probleme die sich für CSCL-Werkzeuge und deren Module ergeben wurde in Kapitel 4.2 auf Seite 29 eingehend behandelt.

Es ist also denkbar, dass ein Angreifer z-B. die Prüfung über das Netz abhört, Daten verfälscht, URL-Hacking betreibt oder durch Denial-of-Service-Angriffe auf den Webserver oder Datenbankserver die Prüfung sabotiert.

7.2.2 Angreifer

Als Angreifer kommen potentiell zwei Personengruppen in Frage. Auf der einen Seite die Studenten, die versuchen ihre Ergebnisse durch Cracken der Datenbank zu verbessern oder aus persönlichen Gründen die Daten anderer Studenten zu verschlechtern.

Auf der anderen Seite die "Sonstigen". Damit sind Dozenten, Administratoren, Tutoren und Cracker gemeint, die aus Freundschaft, Habgier oder anderer persönlicher Gründe die Datenbank, den Webserver oder das Netzwerk cracken oder legal betreten, um gespeicherte Daten zu erlangen oder diese zu manipulieren.

7.2.3 Klassifikation der Angriffe

Betrachtet man die oben beschrieben Angriffe, so ist festzustellen, dass sie als Spezialisierungen der in Kapitel 4.2 auf Seite 29 beschriebenen (Basis-)Angriffsszenarien auf ein CSCL-System einzuordnen sind. Aus diesem Grund verzichte ich an dieser Stelle auf eine erneute Wiederholung der Analyse und Darstellung der Angriffszenarien als Angriffbäume, da lediglich das Angriffziel geändert wird. Die allgemeinen Angriffziele der aufgezeigten Basisangriffszenarien werden nur durch spezielle Angriffsziele der Angriffsszenarien des MCTM erweitert.

Aus "Angriff auf die Kommunikation" wird z.B. "Angriff auf die Kommunikation zur Manipulation des Testablaufs". An dieser Stelle zeigt sich der Vorteil einer allgemeinen Sicherheitsanalyse des Systems und der Wiederverwendbarkeit der Analyseergebnisse. Ohne viel Aufwand lassen sich die bereits erlangten Ergebnisse auf neue Softwaremodule mit ihren speziellen Sicherheitsproblemen übertragen.

Aus Gründen der Vollständigkeit liste ich allerdings in Tabelle 7.1 auf der nächsten Seite auf, welche Basisangriffe bei den jeweiligen Angriffsszenarien des MCTM erfüllt sein müssen, um eine Angriff erfolgreich durchzuführen. Die Basisangriffe erhalten folgende Kürzel:

- Angriff auf die Kommunikation (AKO)
- Angriff auf die Datenbasis (ADB)
- Angriff auf den Webserver (AWS)

Angriff	AKO	ADB	AWS
1. Angriff auf gespeicherte Testdaten		√	
2. Angriff auf die Authentizität des Probanden	√	√	
3. Angriff auf allgemeine Sicherheitsprobleme netzwerkbasierter Anwendungen	√	√	√

Tabelle 7.1: Zeigt den Zusammenhang der Basisangriffe auf das CSCL-System und der speziellen Angriffe auf das MCTM

Gleicht man diese Tabelle mit der Tabelle 4.1 auf Seite 39 ab, so erhält man die verletzten Sicherheitsdienste durch die entsprechenden Angriffe auf das MCTM.

7.3 Bewertung der Analyse

In der Analyse habe ich festgestellt, dass der Betrieb des MCTM zur Leistungskontrolle verschiedene Sicherheitsprobleme aufweist. Demnach muss bereits bei der Implementation darauf geachtet werden, dass gewisse Risiken von vornherein vermieden werden. Nun werde jeden Angriff bewerten und mögliche Lösungen aufzeigen.

7.3.1 Angriff auf gespeicherte Testdaten

Ist als kritisch einzustufen, da die Objektivität der Bewertung und Chancengleichheit gefährdet ist. Die Testdaten zur Leistungskontrolle bedürfen besonderen Schutzes.

Es empfielt sich darüber nachzudenken, die Daten ausschließlich auf einem nicht permanent vernetzen oder sehr sicheren Computer zu speichern und entweder per Diskette oder kurzer Übertragung der Daten auf den Prüfungsserver für Prüfungen zugänglich zu machen. Die Kommunikation über ein Netzwerk muss in jedem Fall über einen sicheren Kanal erfolgen.

Ebenso ist es möglich, die Fragen und Antworten zunächst von den korrekten Ergebnissen getrennt zu speichern. Ein Auswertung wird dann nicht direkt während des Tests sondern später separat geschehen und verhindern, dass ein Angreifer die Ergebnisse während des Tests ausspioniert oder verändert.

7.3.2 Angriff auf die Authentizität des Prüflings

Dieses Problem existiert für Prüfungen jeglicher Art. Bei Face-To-Face-Prüfungen löst man es einfach durch das Zeigen eines Personalausweises, was hohe Sicherheit gewährleistet.

Findet die Authentifizierung digital und ohne Sichtkontakt statt, ist es schwieriger. Login-Daten können ohne Probleme zeitweise auf andere Personen übertragen werden. Selbst eine sehr sichere biometrische Authentifizierung

[Anderson 2001] würde das Problem nicht lösen, da nach der Authentifizierung die Fragen von einer dritten Person beantwortet werden können.

Ergo kann das Problem der Authentifizierung bei Prüfungen am Computer nur *sicher* dadurch gelöst werden, dass die Prüflinge sich per Sicht und Ausweis beim Prüfer authentifizieren und während der Prüfung unter Aufsicht stehen.

7.3.3 Angriff auf allgemeine Sicherheitsprobleme netzwerkbasierter Anwendungen

Angriffe dieser Art stellen ein generelles Problem dar. Schutz bieten hohe Sicherheitsstandards, wie z.B. verbindliche Sicherheitsrichtlinien, aktuelle Sicherheitssoftware, sichere Übertragung und sinnvolle Authentifizierung.

Weiteres dazu wird z.B. in [Anderson 2001, Schneier 2001, Anonymus 1999, Stallings 1995] vertieft.

7.4 Implikationen der Analyse auf die Spezifikation und Implementation von tQuest

tQuest ist innerhalb dieser Arbeit als Modul zur Lernerfolgskontrolle angelegt. Das bedeutet, dass innerhalb von tQuest zunächst keine sensiblen Daten gespeichert werden (vgl. Kapitel 7.1 auf Seite 57). Demnach sind auch keine besonderen Sicherheitsvorkehrungen vonnöten.

Da tQuest allerdings in eine webbasierte Applikation (der Worksphere) integriert wird, sollen gewisse Sicherheitsmassnahmen zum Schutz vor allgemeinen Angriffen integriert sein. Ebenso soll sich tQuest in die Sicherheitsmechanismen der Worksphere eingliedern.

Diese geschieht durch die in Kapitel 6.3.5 auf Seite 50 beschriebene Rechteverwaltung. Diese basiert auf der Rechteverwaltung und den Authentifizierungsmechanismen der Worksphere und gliedert sich somit gut in die bestehende Applikation ein. Aus Sicht der Worksphere stellt die Integration von tQuest somit kein Sicherheitsproblem dar.

In der Implementation von tQuest sind folgende Sicherheitsmechanismen umgesetzt:

- Verwendung der Rechte der Worksphere
- schützen jedes einzelnen Teilmoduls durch diese Rechte, um unautorisierten Zugriff zu unterbinden
- zusätzliches schützen jedes einzelnen Skripts, um URL-Hacking zu unterbinden
- schützen jedes einzelnen Tests um Authentizität der Tests zu gewährleisten und Tests nur für bestimmte Benutzer freizuschalten

Sollte tQuest später zu einem Modul zur Leistungskontrolle erweitert werden, stellen diese Mechanismen meiner Meinung nach eine gute Basis zur Weiterentwicklung dar. Konkrete Vorschläge für dann erforderliche Mechanismen sind in der Analyse oben zu finden.

Kapitel 8

Implementationsdetails zu tQuest

In diesem Kapitel erläutere ich die Details zum Aufbau und Integration von tQuest innerhalb der Worksphere. Mit Aufbau meine ich die Struktur der Teilmodule und des Programmcodes von tQuest, sowie die Semantik der verwendeten Datenbanktabellen und deren Felder.

8.1 Allgemeines zur Implementation

Die Implementation von tQuest ist als Modul der Worksphere gedacht. Das Design von tQuest ist weitestgehend dem Design der Worksphere angepasst.

Aus diesem Grunde schien es bei der Implementation logisch, für tQuest die Technologien der Worksphere mit zu nutzen. Somit war auch die Integration von tQuest in die Worksphere einfach zu realisieren.

Zur Implementation von tQuest wurden diese Technologien verwendet:

- Betriebssystem war Debian/Linux mit Kernel Version 2.4.21
- die Beschreibungsprache HTML nach der Spezifikationsversion 4.0[1]; es ist größtenteils der XHTML-Standard[2] umgesetzt
- die Scriptsprache PHP Version 4.3.3[3]; der PHP-Interpreter war als Modul in den Apache-Webserver V1.3.28 geladen
- die PHPLIB[4] als Bibliothek die Funktionen zur Sessionverwaltung und Datenbankzugriff bereit stellt
- die Datenbank ist implementiert im SQL-Dialekt MySQL und war während der Implementation auf einem MySQL-Datenbankserver der Version 4.0.14 bereitgestellt

[1] HTML beim W3C: http://www.w3.org/TR/1998/REC-html40-19980424/
[2] XHTML beim W3C: http://www.w3.org/TR/xhtml1/
[3] PHP: http://www.php.net
[4] PHPLIB: http://www.sanisoft.com/phplib/manual/ von Kristian und Boris Erdmann

- einige Codefragmente in der aplettbasierten Programmiersprache JavaScript; diese sind für die Funktionsfähigkeit von tQuest nicht relevant

Die Implementation von tQuest benutzt Open-Source-Software[5] und ist ebenso ein Open-Source-Projekt.

Um tQuest zu benutzen, muss sich der Benutzer über den Authentifizierungsmechanismus der Worksphere am System anmelden. Innerhalb von tQuest werden die Informationen der Worksphere-Benutzerverwaltung und Sessionverwaltung genutzt, um die in Kapitel 6.3.5 auf Seite 50 spezifizierte Rechteverwaltung für tQuest umzusetzen.

Diese Kopplung der Worksphere mit tQuest erfolgt hart-codiert sowie über Schnittstellen, die in Kapitel 8.2.3 auf Seite 82 näher spezifiziert werden. Zweck dieser Schnittstellen ist, die Verzahnung der Worksphere und zentraler Module von tQuest möglichst flexibel zu gestalten, um eine spätere Entkopplung von tQuest als eigenständig integrierbares Modul in verschiedene eLearning-Plattformen zu erleichtern.

tQuest ist manuell in die Worksphere integriert, da die Worksphere bisher kein zentrales Administrationstool zur Integration weiterer Module bereitstellt.

Die Implementation von tQuest ist als "Abbild" der in Kapitel 6 auf Seite 41 beschriebenen Spezifikation und der in Kapitel 7.4 auf Seite 61 beschriebenen Implikationen der Sicherheitsanalyse für das MCTM in Programmcode zu sehen.

8.2 Details der Implementation von tQuest

Im Folgenden beschreibe ich die Details der Implementation von tQuest. Dieses beinhaltet:

- Beschreibung der Gesamtstruktur von tQuest: wie ist Implementation aufgebaut; welche Scripte der Worksphere wurden benutzt
- Beschreibung der Semantik und Kontrollfluss der einzelnen PHP-Scripte: welche Funktion erfüllen die einzelnen Skripte;
- Beschreibung der Semantik und Relationen der einzelnen Datenbanktabellen: welche Funktion erfüllen die Tabellen; welche Relationen existieren zwischen den Tabellen
- Spezifikation und Erklärung der externen Programmschnittstellen
- Beschreibung des Konfigurationskonzeptes von tQuest

8.2.1 Gesamtstruktur von tQuest

Das Hauptverzeichnis von tQuest wurde dem Hauptverzeichnis der Worksphere hinzugefügt. Erreichbar ist es über den Pfad

[5] Open-Source: http://www.opensource.org/

```
<webserver-Pfad>/worksphere/tQuest/
```

Im Hauptverzeichnis befindet sich die Startseite von tQuest. Das Hauptverzeichnis ist untergliedert in verschiedene Unterverzeichnisse mit entsprechenden Skripten.

Alle Skripte mit der Endung .php sind direkt vom PHP-Interpreter ausführbar. Alle Skripte mit der Endung .inc werden beim Start der Worksphere eingebunden. Sie sind nicht direkt ausführbar, sondern enthalten Klassen- und Funktionsbeschreibungen oder Konstanten und Variablen die von tQuest benötigt werden.

8.2.1.1 Übersicht aller neuen Verzeichnisse und Skripte

Diese Verzeichnisse und Skripte wurden der Worksphere hinzugefügt. Die Liste zeigt die Bezeichnung des Verzeichnisses und die Bezeichnung der darin enthaltenen Skripte mit einer kurzen Angabe zu Sinn und Funktion:

./tQuest:	Hauptverzeichnis von tQuest
test_verwaltung.php	Startseite von tQuest
./admin:	Adminstrationsskripte
mctm_rechte.php	Zugriffsrechte auf Module
mctm_test_rechte.php	Zugriffsrechte auf Tests
./hilfe:	Hilfeseiten von tQuest
admin_hilfe.php	Hilfe zur Administration
index.php	Startseite der Hilfe
stammdaten_hilfe.php	Stammdaten Hilfe
testdatenbank_hilfe.php	Testdatenbank Hilfe
testverwaltung_hilfe.php	Testverwaltung Hilfe
./layout:	Layoutdateien für tQuest
./layout/img:	Benötigte Bilder in tQuest
./lib:	Verz. für Bibliotheksdateien
Config.inc	Konfiguartionsdatei
Constants.inc	Systemweite Konstanten
FormCreator.inc	Erzeugen von HTML-Formularelementen
Permissions.inc	Konstanten zur Auswertung von Zugriffsrechten
./testdatenverwaltung:	Testfragenverwaltung
td_bearbeiten.php	Testfrage bearbeiten
td_loeschen.php	Testfrage löschen
td_neu.php	Testfrage neu eingeben
td_thema_neu.php	Themenliste verwalten
td_uebersicht.php	Testfragenübersicht

./testgenerator:	Testverwaltung
tg_auswahl_speichern.php	Test in Datenbank speichern
tg_details.php	Detailsansicht einer Frage in Übersicht
tg_details_speichern.php	Detailsansicht einer Frage nach Auswahl
tg_speichern.php	Fragenauswahl als Liste anzeigen
tg_testgenerator.php	Testfragenauswahl per Computer
tg_tests_bearbeiten.php	Test laden zum bearbeiten
tg_uebersicht.php	Übersicht der auswählbaren Fragen

./testkontrolle:	Testablauf
test_auswertung.php	Testauswertung anzeigen
test_hilfe.php	Hilftext für Frage anzeigen
test_kontrolle.php	Testablaufkontrolle
test_uebersicht.php	Übersicht aller auswählbarer Tests

8.2.1.2 Angepasste und benutzte Dateien der Worksphere

Folgende Dateien der Worksphere wurden zur Integration von tQuest angepasst:

./worksphere:	im Hauptverz. der Worksphere
header.html	tQuest-Logo hinzugefügt
navigation.html	Link zur tQuest Startseite hinzu

./worksphere/lib:	in der Bibliothek
prepend.php	einbinden von Config.inc

./worksphere/styles:	in Stylesheets der HTML-Dokumente
formate.css	eigene Vorlagen hinzugefügt

In jedem Script von tQuest wird die Bibliotheksdatei

`./worksphere/lib/prepend.php`

eingebunden. Diese lädt alle zur Sessionverwaltung und zum Datenbankzugriff der Worksphere notwendigen Skripte aus der PHPLIB. Dieses bewirkt, dass tQuest an die Sessionverwaltung der Worksphere gekoppelt wird. Die Funktionen der PHP-Lib stehen dadurch auch tQuest zur Verfügung.

8.2.1.3 Datenbanktabellen

Für die Implementation von tQuest wurden der Worksphere-Datenbank weitere Tabellen eingefügt:

(1) mc_antworten	Antworten der Fragen
(2) mc_fragen	Fragentabelle
(3) mc_rechte	Zugriffsrechte für Module

```
(4) mc_test_rechte              Zugriffsrechte für Tests
(5) mc_testfragen               Fragen der Tests
(6) mc_tests                    Tabelle der Tests
(7) mc_themen                   Thementabelle
```

Eine Worksphere-Tabelle wird von tQuest mitbenutzt:

```
(8) permissions                 Rechteverwaltung der Worksphere
```

Die Nummerierung wird bei der Beschreibung der Skripte als Referenz auf DB-Tabellen verwendet. Die nächsten Seiten listen die Felder der Tabellen und ihre Datentypen auf.

Feldname	Typ	Beschreibung
frage_id	bigint	ID der Fragen, Relation mit mc_fragen
antwort_text	text	Antworttext
korrekt	tinyint, defaultwert 0	1: Frage korrekt, 0 sonst
id	bigint	Primärschlüssel

Tabelle 8.1: mc_antworten

Feldname	Typ	Beschreibung
frage_id	bigint, auto-increment	Primärschlüssel
frage_text	text	Text der Frage
thema_id	int	ID des Themas aus mc_themen
schwierigkeit	int	Schwierigkeit der Frage
hilfe_text	longtext	Text der als Hilfe angezeigt werden kann
zeitstempel	varchar(40)	Datum der Erstellung
geloescht	tinyint	1: Frage gelöscht, 0 sonst

Tabelle 8.2: mc_fragen

Feldname	Typ	Beschreibung
gruppen_id	bigint	Primärschlüssel
tests_erstellen	tinyint	1: Benutzergr. darf Teste erstellen, 0 sonst
tests_bearbeiten	tinyint	1: Benutzergr. darf Teste bearbeiten, 0 sonst
fragen_verwalten	tinyint	1: Benutzergr. darf auf die Stammdaten zugreifen, 0 sonst
gruppen_name	varchar(250)	Name der Benutzergruppe
tests_ausfuehren	tinyint	1: Benutzergr. darf grundsätzlich Tests durchführen, 0 sonst

Tabelle 8.3: mc_rechte

Feldname	Typ	Beschreibung
id	bigint, auto-increment	Primärschlüssel
test_id	bigint	Test dessen Rechte geregelt werden
gruppe_id	bigint	Benutzergruppe, welcher das Recht erteilt wird
ausfuehren	tinyint	1: Benutzergr. darf Test durchführen, 0 sonst
bearbeiten	tinyint	1: Benutzer darf Test bearbeiten, 0 sonst

Tabelle 8.4: mc_test_rechte

Feldname	Typ	Beschreibung
test_id	bigint	ID des Tests aus mc_tests
frage_id	bigint	ID der Frage aus mc_fragen
frage_pos	int	Position der Frage im Test
id	bigint,auto-increment	Primärschlüssel

Tabelle 8.5: mc_testfragen

Feldname	Typ	Beschreibung
test_id	bigint, auto-increment	Primärschlüssel
test_name	varchar(80)	Name des Tests
hilfetext	tinyint	1: Hilfetexte zulassen, 0 sonst
wiederholen	tinyint	1: falsche Fragen direkt wiederholen, 0 sonst
testende	tinyint	1: falsche Fragen am Testende wied., 0 sonst
zeitstempel	varchar(40)	Datum der Erstellung
test_beschreibung	mediumtext	Beschreibung des Tests
test_geloescht	tinyint	1: Test gelöscht, 0 sonst
rueckmeldungen	tinyint	1: Rückmeldungen anzeigen, 0 sonst
auswertung	tinyint	1: Auswertung anzeigen, 0 sonst

Tabelle 8.6: mc_tests

Feldname	Typ	Beschreibung
thema_id	int, auto-increment	Primärschlüssel
thema_name	varchar(80)	Bezeichnung des Themas
geloescht	tinyint	1: Thema gelöscht, 0 sonst

Tabelle 8.7: mc_themen

8.2.2 Module, Verzeichnisse und Skripte

Die Verzeichnisstruktur bildet die einzelnen Teilmodule und Bibliotheken gemäß der Spezifikation von tQuest ab. Nun wird im Detail erläutert, welche Teilmodule oder Bibliotheken die einzelnen Verzeichnisse abbilden und welche Funktionalität die einzelnen Skripte besitzen.

8.2.2.1 Startseite

Die Startseite

`./tQuest/test_verwaltung.php`

liegt im Hauptverzeichnis von tQuest. Diese Seite ist durch Zugriffskontrollen geschützt. Das heißt, je nach Zugriffsrecht des Benutzers werden verschiedene Bereiche der Seite ein- oder ausgeblendet. Dazu wertet tQuest die Zugriffsrechtetabellen (6) der Rechteverwaltung aus.

8.2.2.2 Administration und Rechteverwaltung

Die Skripte zur Administration von tQuest werden im Verzeichnis

`./tQuest/admin`

abgelegt. Neben den Skripten zur Rechteverwaltung können hier in Zukunft weitere Skripte zur Konfiguration und Administration von tQuest gespeichert werden. Es ist vorgesehen, dass *grundsätzlich* alle Skripte dieses Verzeichnisses nur von registrierten Administratoren ausgeführt werden dürfen.

Bisher beinhaltet diese Verzeichnis die Skripte zur Rechteverwaltung von tQuest:

./tQuest/admin/mctm_rechte.php
Dieses Skript verwaltet die Zugriffsrechte auf Teilmodule von tQuest. Beim starten wird die tQuest-Rechtdatenbank (3,4) mit der Rechtedatenbank der Worksphere (8) abgeglichen:

1. Wurde eine Benutzergruppe der Worksphere gelöscht, so wird sie auch aus tQuest gelöscht. Diese beinhaltet das Löschen aller rechtespezifischen Daten aus der Testdatenbank (5,6) und der Rechtedatenbank (3,4).

2. Wurde die Bezeichnung einer Benutzergruppe der Worksphere geändert, wird sie auch in tQuest geändert. Die neue Bezeichnung wird in die Tabellen (3) und (4) von tQuest übernommen.

Nach der Initialisierung erscheint ein Statusbericht über vorgenommene Änderungen in der Rechtedatenbank. Danach folgt der Hauptbildschirm der Zugriffsverwaltung. Dieser enthält zwei Tabellen (s. Abbildung 8.1 auf der nächsten Seite).

1. Neue Gruppe in die tQuest Rechtverwaltung einfügen:
   Ist dann möglich, wenn noch nicht alle Benutzergruppen der Worksphere in tQuest eingefügt wurden. Dieses wird durch vergleichen der beiden Datenbanktabellen festgestellt. Falls das hinzufügen von Benutzergruppen möglich ist und vom Benutzer durchgeführt wird, übernimmt tQuest die Gruppe in die eigene Rechtedatenbank. In Tabelle (3) werden die Rechte für Module angelegt. In Tabelle (6) wird für jeden existierenden Test ein Eintrag mit keinen Rechten für diese Benutzergruppe geschrieben. Dieses ist für die Initialisierung und Integrität der Rechteverwaltung notwendig.

2. Rechte von bereits in die tQuest Rechtedatenbank existierender Gruppen ändern: die Änderungen der Benutzers werden in die tQuest Rechtedatenbank geschrieben.

Die Optionen möglicher Rechte sind in der Spezifikation der Rechteverwaltung in Kapitel 6.3.5 auf Seite 50 nachzulesen.

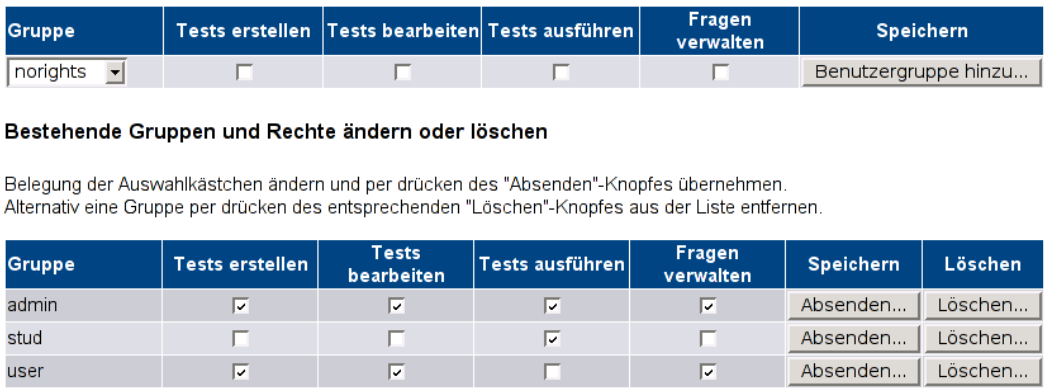


Abbildung 8.1: Übersicht der Zugriffsrechte für Module. Oben: es kann eine weitere Gruppe in die tQuest Rechtetabelle hinzugefügt werden. Unten: die bereits eingefügten Gruppen und ihre Rechte. Sie können an dieser Stelle verändert oder gelöscht werden

Zu beachten ist, dass jedes Skript von tQuest an die Authentifizierung durch die Zugriffskontrolle der hier festgelegten Rechte gebunden ist. Das bedeutet, dass in jedem Skript vor der Ausführung des Programmcodes eine Abfrage bezüglich der Zugriffsberechtigung des Benutzers auf dieses Skript erfolgt. Damit wird der unautorisierte Zugriff auf Skripte von tQuest verhindert.

./tQuest/admin/mctm_rechte.php

Der Administrator kann jedem Test einzeln Zugriffsrechte für in tQuest registrierte Benutzergruppen zuweisen (s. Abbildung 8.3 auf der nächsten Seite). Die Auswahl erfolgt in der Testübersicht der Rechteverwaltung (s. Abbildung 8.2 auf der nächsten Seite). Diese werden in der Rechtetabelle für Tests (4) gespeichert.

Für jeden Test wird für jede Benutzergruppe ein Eintrag erstellt. Dieses ist notwendig für die Integrität der Rechtedatenbank.

Testname	Beschreibung	Rechte vergeben	Erstellt	Rechte
Allgemeinwissentest	für alle die wollen	Ja	10.10.2003, 22:54	Ändern...
Digitaltechnik I.	Test für 1. Semester	Ja	10.10.2003, 22:56	Ändern...
Umfassender Wissenstest	Für die Zwischenprüfung	Nein	10.10.2003, 22:57	Ändern...

Abbildung 8.2: Übersicht der Tests denen Rechte zugewiesen werden können. Angezeigt wird u.a. ob überhaupt bereits Rechte vergeben wurden.

Rechte für Test zuweisen

Parameter	Wert
Testname	Allgemeinwissentest
Testbeschreibung	für alle die wollen
Anzahl Fragen	2
Durchschn. Schw.	45
Recht: Test bearbeiten	☐ admin ☐ stud ☑ user
Recht: Test ausführen	☑ admin ☑ stud ☑ user

Rechte setzen...

Abbildung 8.3: Zugriffsrechte für Tests zuweisen

8.2.2.3 Die Online-Hilfe

Befindet sich im Verzeichnis:

`./tQuest/hilfe/`

Hier liegen alle wichtigen Seiten zur tQuest Online-Hilfe. Die Startseite der Hilfe

./tQuest/hilfe/index.php
Enthält einen Admin-Bereich, der durch die Rechteverwaltung geschützt ist und nur für Administratoren angezeigt wird. Absehen von der Startseite sind alle weiteren Seiten

```
./tQuest/hilfe/amin_hilfe.php
./tQuest/hilfe/stammdaten_hilfe.php
./tQuest/hilfe/testdatenbank_hilfe.php
./tQuest/hilfe/testverwaltung_hilfe.php
```

“starre” HTML-Seiten.

8.2.2.4 Testfragenverwaltung

Verwaltet alle Fragen, Antworten und Themen die im Test zur Verfügung stehen. Sie liegt im Verzeichnis

`./tQuest/testdatenverwaltung`

von tQuest.

./tQuest/testdatenverwaltung/td_neu.php
Das Skript ermöglicht die Eingabe neuer Testfragen (s. Abbildung 8.4). Um die Zahl der Antworten variabel zu gestalten, werden diese in einem einfachen PHP-Array verwaltet, dem für jede weitere Antwort ein neues Element hinzugefügt wird.

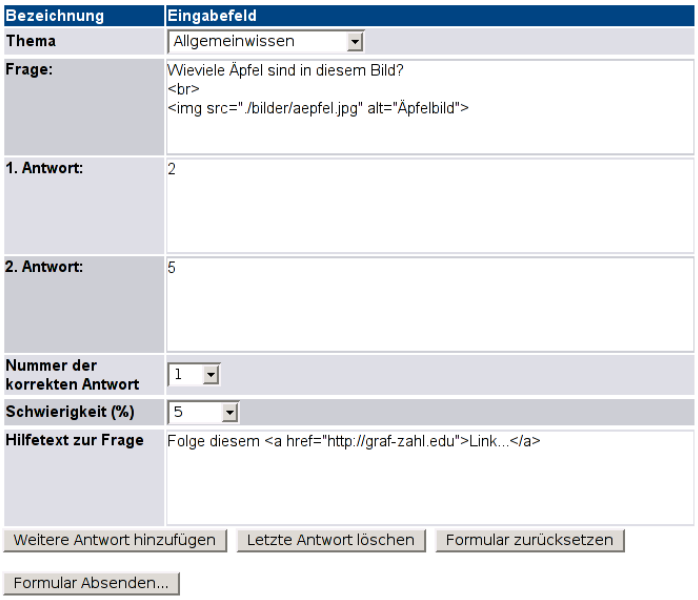


Abbildung 8.4: Eine neue Frage eingeben. Die Anzahl der Antworten kann über den Knopf “Weitere Antworten hinzufügen” erhöht oder über den Knopf “Letzte Antwort löschen” verringert werden.

Die Daten der Fragen und Antworten können auch HTML-Tags enthalten. Diese werden in den Übersichtstabellen entsprechend ausgewertet und dargestellt. Das ermöglicht z.B das Einbinden von Bildern in die Testfragen.

Die HTML-Formulardaten werden über ein POST abschickt und durch erneuten aufruf des Skriptes ausgewertet. Die allgemeinen Daten zur Frage werden in der Fragentabelle (2) gespeichert. Die Antworten zur Frage in der Antwortentabelle (1), wobei eine Relation zwischen den beiden Tabellen über die ID der Fragen besteht. Es erfolgt eine Rückmeldung über das Einfügen für den Benutzer.

./tQuest/testdatenverwaltung/td_bearbeiten.php
Ermöglicht das Bearbeiten von Testfragen. Die Maske hat die gleiche Struktur wie die der Testfrageneingabe (s. Abbildung 8.4). Allerdings ist es nicht möglich, die Anzahl der Antworten zu verändern.

Dieses scheint für mich sinnvoll, da nach einer Änderung der Antwortanzahl die Frage anders gewichtet werden müsste und sich dadurch Teststrukturen verändern würden. Das Bearbeiten von Fragen ist im Prinzip nur für das korrigieren von inhaltlichen Fehlern und Tippfehlern vorgesehen.

Die geänderten HTML-Formulardaten werden über ein POST abschickt und durch erneuten aufruf des Skriptes ausgewertet. Die allgemeinen Daten zur Frage werden in der Fragentabelle (2) geändert. Die Antworten zur Frage in der Antwortentabelle (1). Die Identifikation der zu ändernden Tabellenzeilen erfolgt über die ID der Frage. Es erfolgt eine Rückmeldung über erfolgreiche das Ändern der Daten an den Benutzer.

./tQuest/testdatenverwaltung/td_loeschen.php
Zeigt einen Fragendatensatze im gewohnten Format (s. Abbildung 8.4 auf der vorherigen Seite) an. Der Benutzer muss per Knopfdruck entscheiden, ob er den Datensatz löschen will oder nicht. Es erfolgt dazu eine Rückmeldung.

Wird die Frage gelöscht, wird in Tabelle (2) der Wert der Spalte *geloescht* der zu löschenden Frage auf 1 gesetzt. Die Fragen werden also nicht aus der Datenbank entfernt. Dadurch ist die Integrität Testdatentabelle sicher gestellt. Über das Flag *geloescht* ist es nun per Datenbankabfrage möglich, nur ungelöschte Fragen in der Übersicht anzeigen zu lassen.

./tQuest/testdatenverwaltung/td_uebersicht.php
Zeigt in einer Tabelle alle nicht gelöschten Fragen der Testfragentabelle (2) sortiert nach Thema an (s. Abbildung 8.5).

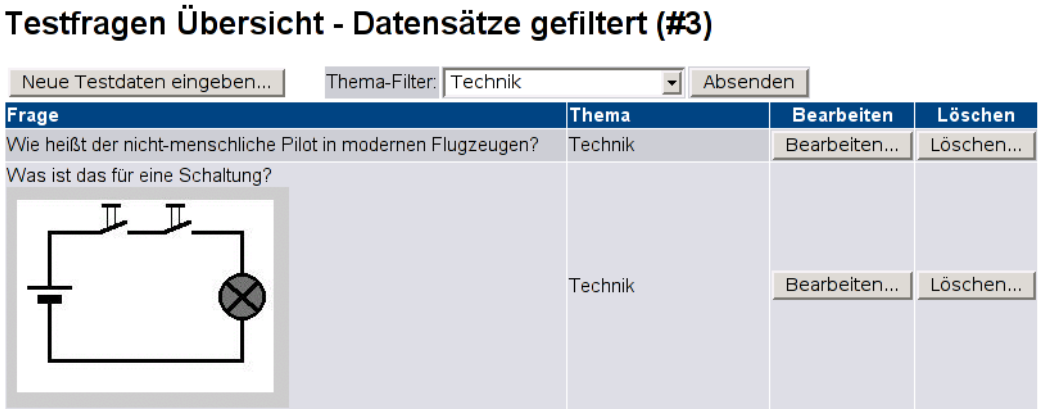


Abbildung 8.5: Die Tesfragenübersicht. Alle nicht gelöschten Fragen werden angezeigt. Es können Fragen eingegeben, gelöscht oder bearbeitet werden. Die Datensätze sind über den Filter nach Themen filterbar.

./tQuest/testdatenverwaltung/td_thema_neu.php
Zeigt in einer Übersicht alle Themen an. Es können Themen einzeln hinzugefügt

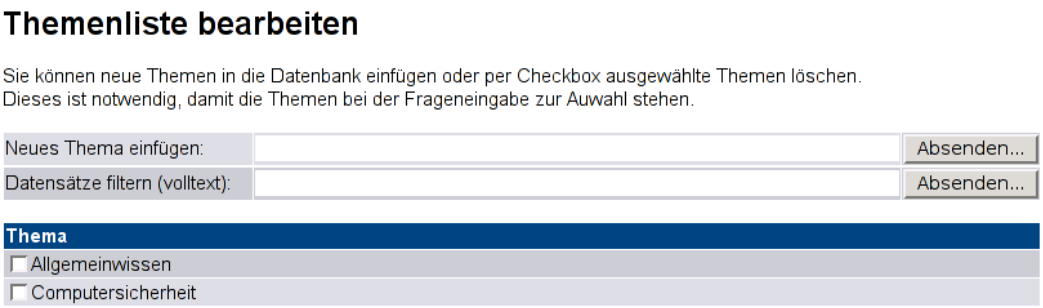


Abbildung 8.6: Die Bearbeitungsmaske für Themen. Möglich ist eingeben, filtern und löschen per Checkbox selektierter Themen.

oder mehrere auf einmal gelöscht werden (s. Abbildung 8.6). Werden Datensätze hinzugefügt, werden sie über eine POST-Aktion des Formulars der Datenbanktabelle für Themen (7) gespeichert.

Zum Löschen werden die selektierten Checkboxen ausgewertet. Die Thementabelle (7) enthält für jeden Datensatz das Flag *geloescht*. Beim Löschen eines Themas wird lediglich dieses Flag auf 1 gesetzt, um die Integrität der Fragentabelle bzgl. der Themen zu erhalten. Die Fragentabelle (2) und die Thementabelle (7) stehen über die Themen-ID miteinander in Relation. Nach dem Löschen erfolgt eine Rückmeldung.

Zum Filtern der Datensätze wird die Thementabelle (7) bzgl. des angegeben Filters durchsucht, die entsprechenden Datensätze selektiert und angezeigt.

8.2.2.5 Testverwaltung und Generator

Liegt im Verzeichnis

`./tQuest/testgenerator`

Es beinhaltet die Skripte für die automatische Fragenauswahl, die manuelle Fragenauswahl, die Übersicht der ausgewählten Fragen, Detailansichten der Fragen und die Speicherroutine für den Test.

./tQuest/testgenerator/tg_uebersicht
Dieses ist die Startseite der manuellen Fragenauswahl. Es zeigt die Übersicht aller in den Test einfügbarer Fragen an (s. Abbildung 8.7 auf der nächsten Seite). Markierte Fragen werden farbig unterlegt. Die Fragen werden aus der Fragentabelle (2) entnommen und nach Thema sortiert.

Die IDs der ausgewählten Fragen werden in einem PHP-Array als Elemente abgelegt. Dieses Array wird beim Aufruf der Seite in der Session der Worksphere registriert. Das hat den Effekt, dass die ausgewählten Fragen durch das Array solange und auf jeder Seite von tQuest verfügbar bleiben, bis das Array wieder freigegeben wird. Dieses kann manuell geschehen oder beim Beenden der Worksphere-Session durch abmelden des Benutzers. Beim manuellen löschen per

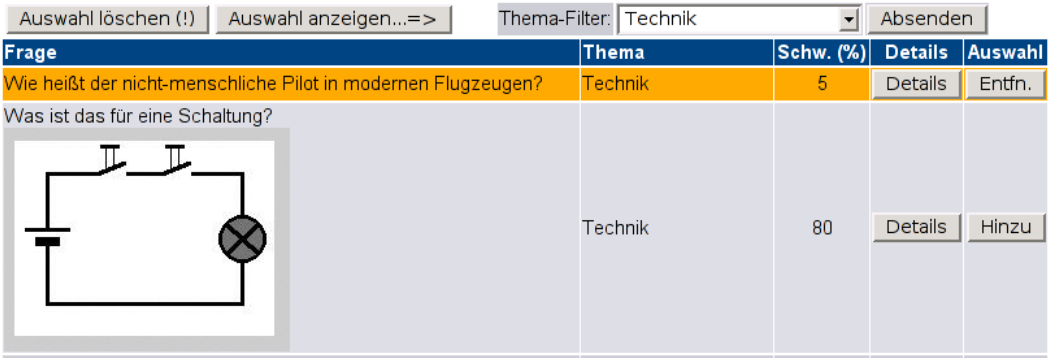


Abbildung 8.7: Die Startseite der manuellen Testfragenauswahl. Mit dem "Hinzu"-Knopf oder "Entf."-Knopf kann eine weitere Frage in die Auswahl übernommen bzw. entfernt werden. Ausgewählte Fragen werde farblich unterlegt.

Druck auf den Knopf "Auswahl löschen" wird das Array ohne weiter Rückfragen neu initialisiert und dadurch geleert.

Die Reihenfolge der Elemente innerhalb des Array bestimmt die Reihenfolge der Fragen im Test. Neue Elemente, sprich neue Fragen, werden immer an das Ende des Arrays durch drücken des Knopfes "Hinzu" angehängt.

./tQuest/testgenerator/testgenerator.php
Ermöglicht die automatische Auswahl der Fragen nach in der Spezifikation definierten Kriterien (s. Abbildung 8.8 auf der nächsten Seite). Um die Auswahl der Fragen zu mischen, werden die Fragen zunächst aus der Fragentabelle (2) nach den Kriterien selektiert und in einem PHP-Array gespeichert.

Danach wird das Array per Zufallsgenerator von PHP neu strukturiert und die Anzahl der gewünschten Fragen in das Fragenauswahl-Array durch heraustrennen der vorderen Elemente des Ergebnisarrays des Zufallsgenerator übernommen.

Sind weniger Fragen bzgl. der angegebene Kriterien in der Fragentabelle (2) vorhanden als die Anzahl der gewünschten Fragen vorgibt, wird automatisch das Maximum der möglichen Fragen aus der Datenbank gewählt. Wurden keine Kriterien festgelegt, erfolgt die automatische Selektion alle Fragen der Fragentabelle (2). Es wird ein Statusbericht nach Abschluss der Generation angezeigt.

./tQuest/testgenerator/tg_speichern.php
Zeigt die ausgewählten Datensätze an (s. Abbildung 8.9 auf Seite 78). Das Skript selektiert dazu alle Datensätze aus der Fragentabelle (2) entsprechend der Elemente im Fragenarray. Der Benutzer hat die Möglichkeit die Position der Fragen innerhalb des Fragearrays zu ändern. Dazu wird der entsprechenden Frage eine neue Position innerhalb des Fragenarrays zugewiesen. Das Fragenarray wird an der entsprechenden Stelle zerlegt, das zu verschiebende Element aus dem Fragenarray entfernt, in die Trennstelle eingefügt und das Array wieder zusammengesetzt. Danach wird die Seite neu geladen um die Änderungen anzuzeigen.

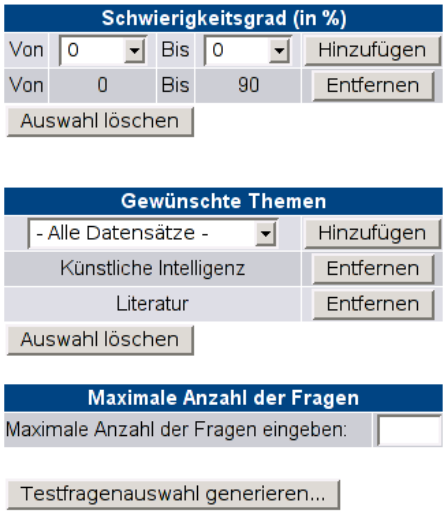


Abbildung 8.8: Die Kriterien der automatischen Testfragenauswahl. Es können mehrere Schwierigkeitsgrade und Themen gewählt werden, die vom Generator berücksichtigt werden. Überscheidungen der Werte sind möglich, ohne als Ergebnis doppelte Fragen zu erhalten.

Von hier kann der Benutzer zur Testspeicherung gelangen. Die entscheidenden Daten für den Test sind ja bereits im Fragenarray enthalten und dadurch auch in dem Skript zur Speicherung des Tests verfügbar.

./tQuest/testgenerator/tg_speichern.php (A) und

./tQuest/testgenerator/tg_details_speichern.php (B)
Zeigt jeweils die Detailansicht einer Frage in gewohnter Form. Skript (A) wird als Detailansicht der Testfragenübersicht der manuellen Auswahl ausgeführt. Hier ist es zusätzlich möglich, die Frage per Knopfdruck der Auswahl der Testfragen hinzuzufügen.

Das Skript (B) wird als Detailansicht der Fragen in der Übersicht aller ausgewählter Fragen angezeigt. Es bestehen keine weiter Optionen für den Benutzer.

./tQuest/testgenerator/tg_auswahl_speichern.php
Das Skript zeigt alle gewählten Testfragen und die möglichen Optionen zum Testablauf (s. Abbildung 8.10 auf Seite 79) in zwei Tabellen an. Es speichert ebenso den Test in der Datenbank . Der Benutzer *muss* hier den Namen und Ablauf des Test festlegen. Die Einträge zur Testbezeichnung und Zugriffsrechte für diesen Test sind optional. Die Eingabe des Benutzers erfolgt in ein HTML-Formular und wird nach dem Absenden von diesem Skript ausgewertet.

1. Fehlen zum speichern des Test verpflichtende Angaben (Testname und Ablauf) wird der Benutzer darauf hingeweisen und die Eingabemaske wird erneut geöffnet.

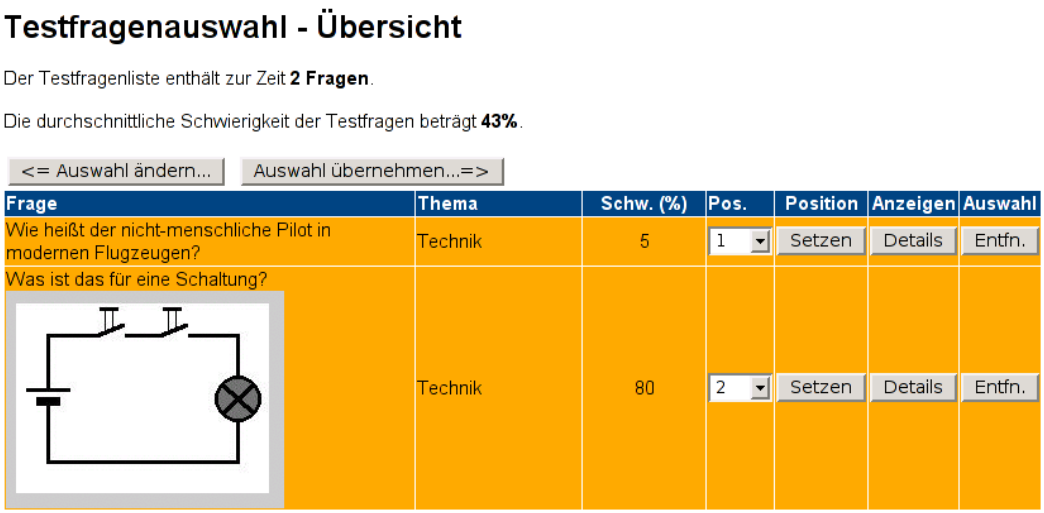


Abbildung 8.9: Die ausgewählten Fragen in der Auswahlliste. Der Benutzer kann den Fragen über die Auswahlboxen und Druck auf "Setzen" eine neue Position innerhalb der Liste zuweisen. werden ignoriert.

2. Sind alle Daten vollständig werden sie in in die entsprechende Datenbanktabellen geschrieben:
   - allgemeine Testdaten in die Tabelle der Tests (6)
   - die Testfragen und ihre Position innerhalb des Tests in die Testfragentabelle (5)
   - in Tabelle für Testrechte (4) wird ein Eintrag für jeden Benutzer mit der ID-des Tests erzeugt und die Rechtewerte gesetzt

Es erfolgt eine Rückmeldung über den Erfolg der Speicherung.

./tQuest/testgenerator/tg_bearbeiten.php
Zeigt eine Übersicht aller Tests dessen Bearbeitung möglich ist (s. Abbildung 8.11 auf der nächsten Seite). Für den Benutzer sind aber nur die Tests auswählbar, die für ihn zur Bearbeitung durch die Rechteverwaltung für Tests (4) freigeschaltet sind. Wählt der Benutzer einen Test zur Bearbeitung, werden lediglich die Fragen des Test aus Tabelle (5) in das Testfragenarray geladen und in der Übersichtliste (Skript tg_speichern.php) angezeigt.

Der Benutzer erhält eine Rückmeldung über das erfolgreiche Laden der Fragen.

8.2.2.6 Testablaufkontrolle

Im Verzeichnis

`./tQuest/testkontrolle`

Paramter	Meine Wahl
Testname:	Soziale Systeme I.
Beschreibung (optional):	Freiwilliger Test für alle Studenten der Soziologie WS2003/2004
Ansehen von Hilfetexten:	◉ Ermöglichen ○ Sperren
Rückmeldungen nach Fragen:	◉ Zulassen ○ Sperren
Falsch beantwortete Fragen:	◉ Direkt einmal wiederholen ○ Nie wiederholen
Am Testende:	○ Alle falsch beantworteten Fragen erneut anzeigen ◉ Direkt zur Testauswertung
Recht, diesen Test zu bearbeiten:	**Gruppen:** ☐ stud ☑ user ☑ admin
Recht, diesen Test auszuführen:	**Gruppen:** ☑ stud ☐ user ☐ admin

Auswahl speichern...

Abbildung 8.10: Die Eingabemaske für die Daten zur Speicherung von Tests.

Testname	Beschreibung	#Fragen	DS (%)	Erstellt	Wählen	Löschen
Allgemeinwissentest	für alle die wollen	2	45	10.10.2003, 22:54		
Digitaltechnik I.	Test für 1. Semester	2	80	10.10.2003, 22:56		
Umfassender Wissenstest	Für die Zwischenprüfung	15	53	10.10.2003, 22:57	Bearbeiten	Löschen
Wissenstest 3	Jahrgang 5	3	53	11.10.2003, 14:27		

Abbildung 8.11: Die Übersicht zur Testbearbeitung. Für diesen Benutzer ist nur ein Test zur Bearbeitung freigeschaltet.

liegen alle Skripte die für die Testdurchführung benötigt werden.

./tQuest/testkontrolle/test_uebersicht.php
Zeigt die Übersicht (s. Abbildung 8.12) aller Tests der Datenbank. Es sind nur die Tests ausführbar, die für den Benutzer durch die Testrechteverwaltung (4) freigeschaltet wurden.

Testname	Beschreibung	#Fragen	DS (%)	Erstellt	Wählen
Allgemeinwissentest	für alle die wollen	2	45	10.10.2003, 22:54	
Digitaltechnik I.	Test für 1. Semester	2	80	10.10.2003, 22:56	
Umfassender Wissenstest	Für die Zwischenprüfung	15	53	10.10.2003, 22:57	Los...
Wissenstest 3	Jahrgang 5	3	53	11.10.2003, 14:27	Los...

Abbildung 8.12: Die Übersicht durchführbarer Tests. Für diesen Benutzer sind zwei Tests zur Durchführung freigeschaltet.

./tQuest/testkontrolle/test_kontrolle.php
Diese Skript kontrolliert den festgelegten Ablauf der Tests. Beim ersten laden des Skripts werden die Ablaufdaten des Test aus der Testtabelle (6) ausgewertet und das Testkontrollmodul damit initialisiert. Die Ablaufkontrolle erfolgt über Entscheidungsbäume realisiert durch verschiedene IF... THEN... ELSE-Blöcke.

Jede Frage wird auf einer eigenen HTML-Seite angezeigt (s. Abbildung 8.13). Selektierte Antworten werden farbig unterlegt. Es findet keine Abfrage statt, ob eine Antwort vor dem abschicken ausgewählt wurde oder nicht. Fragen ohne ausgewählte Antwort werden als falsch bewertet. Das gibt die Möglichkeit für den Benutzer, Fragen zu überspringen. Nach dem Absenden wird auf Basis der

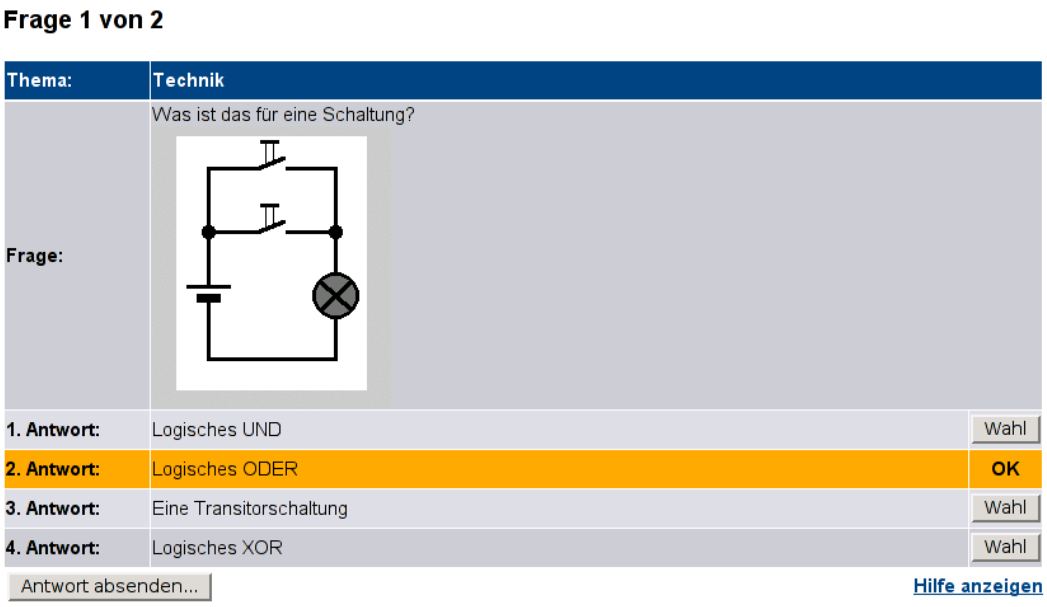


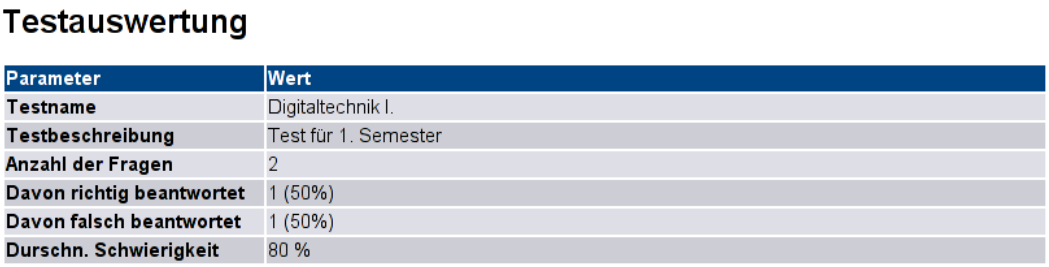
Abbildung 8.13: Ansicht ein Frage bei der Testdurchführung. Eine Antwort wurde ausgewählt. Der Hyperlink "Hilfe anzeigen" wird nur eingeblendet, wenn Hilfetexte im Testablauf zugelassen sind.

Antwortentabelle (1) ausgewertet, ob die Antwort richtig oder falsch war. Richtig sowie falsch beantwortete Fragen werden jeweils in einem eigenen Array in Form der Fragen-ID gespeichert. Diese beiden Arrays werden beim Start des Tests in der Session der Worksphere registriert, damit sie während des gesamten Tests zur Verfügung stehen.

Rückmeldungen werden als HTML-Seiten angezeigt.

./tQuest/testkontrolle/test_hilfe.php
Zeigt eine Hilfeseite mit dem entsprechende Hilfetext der aktuelle Testfrage in einem externen Fenster an. Dazu wird der Hilfetext aus der Fragentabelle (2) geholt.

./tQuest/testkontrolle/test_auswertung.php
Zeigt in einer Tabelle die Testauswertung an (s. Abbildung 8.14). Berechnet wird die Anzahl der richtig und falsch beantworteten Fragen in Prozent. Desweiteren wird eine Detailansicht der Testfragen mit jeweils korrekter Antworten angezeigt. Die benötigten Daten stammen aus den Arrays, die beim Teststart



Testauswertung

Parameter	Wert
Testname	Digitaltechnik I.
Testbeschreibung	Test für 1. Semester
Anzahl der Fragen	2
Davon richtig beantwortet	1 (50%)
Davon falsch beantwortet	1 (50%)
Durschn. Schwierigkeit	80 %

Abbildung 8.14: Ansicht der Testauswertung am Testende.

in der Worksphere-Session registriert wurden.

8.2.2.7 tQuest Bibliothek

Die Programmbibliothek mit Include- und Konfigurationdateien für tQuest befindet sich im Verzeichnis

`./tQuest/lib`

Diese Dateien werden zur Ausführung von tQuest benötigt.

./tQuest/Config.inc
Die Konfigurationsdatei von tQuest. Sie wird beim Start der Worksphere eingebunden und lädt wiederum alle benötigten Include-Dateien für die Ausführung von tQuest. Näheres dazu in Kapitel 8.2.4 auf Seite 83.

./tQuest/Constants.inc
Systemweite Konstanten. Zweck ist, oftmals in der Applikation benötigte Werte zentral zu verwalten.

./tQuest/FormCreator.inc
Eine Klasse, die Funktionen bereitstellt, welche auf Basis entsprechender Parameter HTML-Formular-Elemente zurückgeben.

Häufig verwendete HTML-Formular-Elemente, wie z.B. Label und Eingabefeld, sollen mit einem einzigen Funktionsaufruf zu generiert werden können.

Bisher sind das diese Elemente:

```
createTextfieldCol()    Label+Textfeld in Tabellenspalte
createTextareaRow()     Label+Textarea in Tabelenzeile
createSelectRow()       Label+Auswahlbox in Tabellenzeile
createSelect()          Label+Auswahlbox
createInputType()       HTML-Input-Type (z.B. Button)
createStyleInputType()  HTML-Input-Type mit Hintergrundfarbe
createSpace()           x-Platzhalter ( )
```

./tQuest/FormCreator.inc
Eine Klasse die Methoden zur Abfrage von Zugriffsrechten für Benutzer bereitstellt. Diese sind:

```
havePerm()              Recht auf tQuest-Module zuzugreifen
haveTestPerm()          Recht auf Tests zuzugreifen
```

Die Methoden liefern einen booleschen Wert zurück, je nachdem ob das Recht in der Datenbank gesetzt ist (TRUE) oder nicht (FALSE).

Die Klasse benötigt die Konstanten der Include-Datei *Constants.inc*. Die Abfrage der Rechte in den einzelnen Funktionen geschieht über eine SWITCH...CASE-Block, je nachdem welches Recht in der einzelnen Funktion abgefragt werden soll (z.B. darf der Benutzer-Test-X *bearbeiten*?).

8.2.3 Integration und Schnittstellen

Um tQuest einfach in andere eLearning-Systeme integrieren zu können, ist es sinnvoll, generische Schnittstellen bereitzustellen. Da besonders die Rechteverwaltung von tQuest zunächst inkompatibel zu der Rechteverwaltung anderer eLearning-Plattformen sein kann, ist eine Schnittstelle für die Rechteverwaltung bereits in tQuest realisiert.

Um diese Schnittstelle zu Nutzen, muss die eLearning-Plattform über ein konzeptuell ähnliches Rechtesystem wie die Worksphere verfügen. D.h., es wird eine Tabelle erwartet, in der Benutzergruppen und deren IDs gespeichert werden. Denn diese müssen innerhalb von tQuest bekannt sein, damit Benutzergruppen in die Rechtverwaltung von tQuest übernommen werden können.

Die notwendigen Einstellungen dazu werden in der Konfigurationsdatei *Config.inc* (s. Kapitel 8.2.4 auf der nächsten Seite) von tQuest vorgenommen. Dort muss der Administrator den Namen der fremden Datenbank, der entsprechenden Rechteverwaltungstabelle und den Namen der Felder der Benutzergruppen-ID und Benutzergruppen-Name zuweisen. Die Werte der Zuweisungen werden in Konstanten gespeichert. Diese Konstanten sind dann in tQuest an Stelle der konkreten Werte in der Programmcode eingefügt.

Anzumerken ist, dass tQuest in der vorliegenden Version noch nicht unabhängig von der Worksphere als Modul anderer eLearning-Systeme betrieben werden kann. Denn bisher werden neben der Rechteverwaltung auch Stylesheets und andere Funktionen, wie z.B. dass Sessionmanagement, der Worksphere in

der tQuest-Implementation genutzt. Es müssen weitere Schnittstellen definiert werden, sowie die Bibliothek von tQuest um entsprechende Funktionen erweitert werden, um die vollständige Integrationsfähigkeit von tQuest zu gewährleisten.

Da die Worksphere auf Open-Source-Bibliotheken beruht und selber ein Open-Source-Projekt ist, besteht zumindest kein rechtliches Problem bei der Erweiterung der tQuest-Bibliothek um Worksphere bzw. PHP-LIB Funktionen.

8.2.4 Konfigurationskonzept

Die Konfiguration von tQuest wird zentral in der Datei *Config.inc* im *lib*-Verzeichnis von tQuest verwaltet. In dieser Datei können vom Administrator alle notwendigen Einstellungen zum Betrieb von tQuest vorgenommen werden.

Ebenso werden in dieser Datei alle von tQuest benötigten Include-Dateien geladen. Neue Include-Dateien müssen daher dieser Datei hinzugefügt werden.

Alle einstellbaren Parameter werden in dieser Datei verschiedenen Konstanten zugewiesen, die dann Anstelle des konkreten Wertes im Programmcode eingefügt wurden.

Bisher sind folgende Konfigurationsoptionen implementiert:

- Parameter zur Schnittstelle der Rechtedatenbank
- zu ladende Include-Dateien
- Anzahl der Datensätze, die in der Testübersicht und der Testfragenübersicht angezeigt werden sollen

8.2.5 Benutzerfreundlichkeit und Design

Bei der Implementation von tQuest habe ich darauf geachtet, dass Eingabemasken und Tabellen mit ähnlicher Funktionalität und Semantik ein einheitliches *Look & Feel* besitzen. Dieses erhöht den Wiedererkennungswert, steigert den Gewöhungseffekt und soll die Benutzung von tQuest nachhaltig erleichtern.

Die Masken und Tabellen habe ich versucht so zu gestalteten, dass sie selbsterklärend sind. Allerdings wird in einigen Modulen zusätzlich die Struktur der Maske erläutert und die Wirkung einzelner Knöpfe erklärt.

Viele Interaktionen des Benutzers mit tQuest werden durch Rückmeldungen an den Benutzer bestätigt und deren Auswirkungen auf das System beschrieben.

8.2.6 Test und Betriebsfähigkeit

Die Implementation von tQuest wurde von mir eingehend getestet. Dieser Test bezog sich sowohl auf die Funktionalität als auch die Darstellbarkeit in verschiedenen Web-Browsern. Beim Test war tQuest als Modul der Worksphere Version 1.0-beta auf einem LAMP-System (Linux, Apache, MySQL, PHP) mit Debian/linux installiert.

Folgende Browser habe ich getestet:

- KDE Konqueror 3.1.2 (GNU/Linux)
- Microsoft Internet Explorer 6.0 (Windows XP/2000)
- Mozilla 1.4 (GNU/Linux)
- Netscape Communicator 4.78 (Sun/Solaris)
- Opera 7.11 (Debian/Linux, Windows 2000)
- Opera 7.20pre (Debian/Linux, Windows XP)

tQuest konnte auf allen Browsern problemlos ausgeführt werden.

8.3 Zusammenfassung und Ausblick

In der vorliegenden Version V1.0beta ist tQuest als voll funktionsfähiger Multiple-Choice-Tester für die Worksphere implementiert. Die in Kapitel 6 auf Seite 41 aufgestellte Spezifikation wurde vollständig in der Implementation umgesetzt. Die in Kapitel 7.4 auf Seite 61 herausgearbeiteten Sicherheitsempfehlungen sind ebenso in die Implementation eingeflossen.

Der nächste Schritt der Weiterentwicklung von tQuest wäre die Entkopplung aus der Worksphere. tQuest stünde dann als eingenständiges Modul für verschiedene eLearning-Plattform bereit.

Um tQuest letztendlich zu einer Standalone-Applikation zu erweitern oder mit tQuest Tests zu Leistungskontrolle online durchzuführen sind umfangreichere Erweiterungen nötig. Als Standalone-Applikation müsste tQuest mindestens über ein eigenes Session-Management und eine vollwertige Benutzerverwaltung verfügen.

Um tQuest zusätzlich als Applikation zur Leistungskontrolle aufzuwerten, müssten die in Kapitel 7.2 auf Seite 57 beschriebenen Sicherheitsmechanismen in tQuest eingefügt werden.

Ich stelle hier kurz einen möglichen Plan zur weiteren Entwicklung von tQuest vor:

1. Testphase und Revision
   (a) Test von tQuest durch Benutzer
   (b) allgemeine Code-Revision
   (c) striktes trennen von Funktionalität und Inhalt in den PHP-Skripten
   (d) umsetzten von Benutzerkritik durch Anpassungen in der Implementation
2. Entkopplung von der Worksphere
   (a) eigene Stylesheets

   (b) Schnittstellen zur Sessionverwaltung

3. tQuest zur Leistungskontrolle

   (a) erweitern des Sicherheitskonzepts von tQuest um die Ergebnisse der Analyse von Kapitel 7.2 auf Seite 57

4. tQuest als eigenständige Applikation

   (a) hinzufügen einer eigenen Sessionverwaltung

   (b) hinzufügen einer eigenen Benutzerverwaltung

8.4 tQuest Quickstart

Um tQuest nach der Installation zu benutzen, muss zunächst die Rechtedatenbank der Worksphere mit der Rechtedatenbank von tQuest abgeglichen werden. Dieses geschieht im Administrationstool *"Zugriffsrechte für tQuest-Module verwalten"*.

Es ist von der tQuest-Startseite aus erreichbar. Die gewünschten Benutzergruppen der Worksphere werden hier in tQuest übernommen und mit Zugriffsrechten versehen. Erst *danach* können die bereits in der Worksphere registrierten Benutzer dort freigeschaltete Module von tQuest benutzen. Um diese Einstellungen vorzunehmen ist ein Administratorzugang zur Worksphere notwendig.

Vor der Eingabe von Testfragen, sollten zunächst die entsprechenden Themen in die Themendatenbank eingepflegt werden. Diese stehen dann in einer Auswahlbox bei der Testfrageneingabe zur Verfügung. Für Unklarheiten bei der Benutzung von tQuest steht die Online-Hilfe für jeden Benutzer zur Verfügung. Diese ist von der tQuest-Startseite zu erreichen.

Die Installation von tQuest ist in der Datei *Install_tQuest.pdf* auf der beigefügten CD dokumentiert.

Kapitel 9

Zusammenfassung und Perspektiven

Die Zielsetzung der Arbeit war die Entwicklung eines leistungsfähigen Tools zur Lernerfolgskontrolle innerhalb von CSCL-Werkzeugen. Die Literaturecherche ergab, dass der Multiple-Choice-Test für diese Art der Lernerfolgskontrolle geeignet war und durch seinen starren Aufbau gut softwaretechnisch umzusetzen ist.

Allerdings fiel mir bei der Recherche ebenso ins Auge, dass Aspekte wie Software- oder Datensicherheit in der Entwicklung von CSCL-Systemen eine sehr untergeordnete Rolle spielen. Da aber CSCL-Systeme persönliche oder sensible Daten verwalten, schien es sinnvoll, zunächst zu betrachten, welche Sicherheitsprobleme allgemein innerhalb von CSCL-Systemen bestehen und wie diese sich auf die Entwicklung und Implementation meines speziellen Moduls auswirken.

Das Ergebnis der Analyse zeigte, dass CSCL-Systeme, genau wie viele andere netzwerkbasierte Kommunikationssysteme, Ziel verschiedener Angriffszenarien sein können. Für die Entwicklung meines Moduls bedeutete das zu analysieren, wie sicherheitskritisch die verwalteten Daten sind und welche Angriffsszenarien denkbar sind, um demnach Empfehlungen über notwendige Sicherheitsmechanismen zu formulieren.

Da mein Testmodul zunächst nur zur Lernerfolgskontrolle benutzt werden soll, habe ich festgestellt, dass keine harten Sicherheitsvorkehrungen getroffen werden mussten. Es werden keine sensiblen oder persönliche Daten vom Testmodul verwaltet. Vielmehr war dann das Ziel, die Sicherheitsmechanismen des Testmoduls denen des CSCL-Systems anzugleichen, damit das Testmodul kein Sicherheitsrisiko innerhalb des CSCL-Moduls darstellt. Letztendlich ist das MCTM auch nur so sicher wie die CSCL-System selbst, da es ein Teilsystem des CSCL-Systems ist.

Die Ergebnisse der Sicherheitsanalyse sind ebenso zusätzlich für die weitere Entwicklung von CSCL-Systemen und meines Testmoduls interessant. Es ist zu schliessen, dass Sicherheitsmechanismen in Zukunft bei der Forschung und Entwicklung von CSCL-Systemen eine größere Rollen spielen sollten. Ich denke, die Analyse hat gezeigt, auf welche verschiedenen Arten und Weisen Cracker die Authentizität von Benutzern, Vertraulichkeit und Integrität von Daten und die

Verfügbarkeit des Systems verletzen können.

Der zweite Teil meiner Analyse betrifft den Einsatz meines Moduls für studienrelevante Leistungskontrolle. Aus der Analyse geht hervor, dass verschiedene Sicherheitsprobleme bei der Weiterentwicklung von tQuest als Leistungskontrollmodul zu berücksichtigen sind, um vorallem Authentizität der Benutzer und Verfügbarkeit des Systems zu jeder Zeit zu gewährleisten.

Die konkrete Implementation meines Moduls wurde deshalb so offen gestaltet, dass das hinzufügen weiterer Features und die Erweiterung als eigenständiges Lernerfolgs- bzw. Leistungskontrollmoduls für CSCL-Werkzeuge gut zu realisieren ist. Wird die Weiterentwicklung des MCTM zu einem Leistungskontrollmodul angestrebt, stellt die entsprechende Analyse eine gute Basis zur Realisierung von Sicherheitsmechanismen dar.

Entstanden ist tQuest, ein intuitiv zu benutzendes Multiple-Choice-Testmodul für CSCL-Werkzeuge. Es adaptiert die Sicherheitsmechanismen des übergeordneten CSCL-Werkzeugs, hier der Worksphere, und fügt sich daher sicherheitstechnisch unkritisch darin ein.

Literaturverzeichnis

[Anderson 2001] Ross Anderson, 2001, *Security Engeneering*, Wiley.

[Anonymus 1999] Anonymus, 1999, *Hackers Guide - Sicherheit im Internet und im lokalen Netz*, Markt und Technik Verlag.

[Booch & Rumbaugh, Jacobson 1999] Grady Booch, James Rumbaugh, Ivar Jacobson, 1999, *Das UML-Benutzerhandbuch*, Addison-Wesley.

[Bortz & Döring 2002] Jürgen Bortz, Nicola Döring, 2002, *Forschungsmethoden und Evaluation*, 3.Aufl., Springer-Verlag Berlin Heidelberg.

[Ellermann 2002] Martin Ellermann, 2002, *Design und Entwicklung einer vorlesungsbegeleitenden Übungsplattform*, Diplomarbeit im Fach Naturwissenschaftliche Informatik, Universität Bielefeld.

[Kerres 2001] Michael Kerres, 2001, *Multimediale und telemediale Lernumgebungen*, 2.Aufl., Oldenburg Verlag München Wien.

[Koreimann 1992] Dieter S. Koreimann, 1992, *Grundlageen der Software-Enwicklung*, Oldenburg Verlag München Wien.

[Krause 2001] Jörg Krause, 2001, *PHP4 - Grundlagen und Profiwissen*, Hanser Verlag.

[Krause 2001] Jörg Krause, 2001, *PHP4 - Die Referenz*, Hanser Verlag.

[Ladkin 2001] Peter B.Ladkin, 2001, *Causal System Analysis*, Draft-Version 2.0.

[Laprie 1992] Jean-Claude Laprie, 1992, *Dependability: Basic Concepts and Terminology*, Hrsg., Springer Verlag Wien.

[Moore & Ellison & Linger 2001] Andrew P. Moore, Robert J. Ellison, Richard C. Linger, *Attack Modelling for Information Security and Survivability*, Carnegie Mellon University Pittsburgh, Report Number: CMU/SEI 2001 TN 001.

[Parks] Jay Parks, *Multiple-Choice-Test*, Internetquelle University of Mexico,

`http://www.flaguide.org/cat/mutiplechoicetest/multiple_choice_test1.html`

[Peterson & Davie 2000] Larry L. Peterson, Bruce S. Davie, 2000, *Computernetze. Ein modernes Lehrbuch*, dpunkt.verlag.

[Raymans 2001] Heinz-Gerd Raymany, 2001, *MySQL im Einsatz*, Addison-Wesley.

[Schneier 2001] Bruce Schneier, 2001, *Secrets & Lies*, dpunkt.verlag/Wiley.

[Schneider & Werner 2001] Uwe Schneider, Dieter Werner 2001, Hrsg., *Taschenbuch der Informatik*, 4. Aufl., Fachbuchverlag Leibzig.

[Schulmeister 2002] Rolf Schulmeister, 2002, *Grundlagen hypermedialer Lernsysteme - Theorie, Didaktik, Design*, Oldenburg Verlag München Wien.

[Schulmeister 2001] Rolf Schulmeister, 2001, *Virtuelle Universität. Virtuelles Lernen*, Oldenburg Verlag München Wien.

[Schulmeister 2003] Rolf Schulmeister, 2003, *Lernplattformen für das virtuelle Lernen*, Oldenburg Verlag München Wien.

[Stallings 1995] Wiliam Stallings, 1995, *Sicherheit im Datennetz*, Prentice Hall.

[Uni Minnesota 2003] University of Minnesota, 2003, *Writing Multiple-Choice Items*, Internetquelle University of Minnesota,

`http://www.ucs.umn.edu/oms/multchoice.htmlx`

Kapitel 10

Glossar

Eine kurze Erläuterung einiger verwendeter Begriffe. Nähere Details finden sich z.B. in [Laprie 1992, Anonymus 1999, Peterson & Davie 2000, Schneier 2001, Anderson 2001].

Absturz, abstürzen Wenn eine Computer (plötzlich) ausfällt und neu gebootet werden muss.

Ausfall (engl. failure) "Abweichung der erbrachten Leistung von der in der System-Spezifikation geforderten Leistung. Übergang von korrekter Leistungserbringung zu fehlerhafter Leistungserbringung." ([Laprie 1992], S.140)

Authentifizieren Überprüfen der Identität und damit der Berechtigung eines bestimmten Benutzers oder Clients.

Benutzer Jeder, der ein Computersystem oder Systemressourcen nutzt.

Buffer-Overflow-Angriff Es wird versucht mit einer geschickt zusammengesetzten Texteingabe in ein unsicheres System eine Shell (Befehlseingabeoberfläche) zu starten. Diese Shell sollte möglichst Administratorrechte besitzen.

Crack, cracken Mit einer Software oder anderer Technik Sicherheitsvorkehrungen eines Computersystems zu umgehen.

Cracker Person, die mit böswilliger Absicht und unter Missachtung von Gesetzen die Sicherheit eines Computers verletzt.

CSCL Abk. für "Computer Supported Cooperative/Collaborative Learning". Computerbasierte Lernform, die auf Computerunterstützung aber gemeinschaftliches lernen aufbaut.

Fehlerursache (engl. fault) "Anerkannte oder hypothetische Ursache für einen Fehler. Fehlerursache, die vermieden oder toleriert werden sollte. Auswirkung auf das betrachtete System durch Ausfall durch den Ausfall eines anderen Systems, das mit dem System zusammengewirkt hat oder zusammenwirkt." ([Laprie 1992], S. 141)

Fehlerzustand (engl. error) “Teil eines Systemzustandes, der dafür verantwortlich ist, dass ein Ausfall auftritt. Offenbarung einer Fehlerursache im System.” ([Laprie 1992], S. 142)

Firewall Eine Einrichtung oder Technik (Software oder Hardware), die unbefugten Zugriff auf ein Computersystem verhindern soll.

Hacker Person, die sich für Betriebsysteme, Software , Sicherheit und allgemein das Internet interessiert. Oder ein Programmierer.

Lexikon Angriff Knacken von Passwörtern mit Hilfe eines Wörterbuches. Kennt der Angreifer den gültigen Loginnamen, kann er alle Wörter der Lexikons als Passwort ausprobieren. Funktioniert gut bei einfachen Passwörtern. Einfachste Form eines Lexikon-Angriffs wäre das durchgehen des Wörterbuchs von vorne bis hinten, bis der Treffer gefunden wurde (Brute-Force).

Man-In-The-Middle-Angriff Eine dritte Partei fängt die Netzwerkkommunikation zwischen zwei Parteien ab, indem sie sich dem Sender als Empfänger ausgibt und danach die Daten an den eigentlich regulären Empfänger weiter sendet. Diese Dritte Partei heißt dann Man-In-The-Middle.

MCTM Abk. für Multiple-Choice-Test-Modul.

Shell Ein Befehlsinterpreter oder jedes Programm, das Standardeingaben (z.b. über Tastatur) annimmt und diese Befehle an das System weitergibt.

Spoofing-Angriff Technik der Fälschung von Daten auf einem Netzwerk durch Vortäuschung einer falschen Absenderadresse, oft mit der Absicht, durch die gefälschte Absenderadresse authentifiziert zu werden. Bsp. hierfür wäre der Man-In-The-Middle-Angriff.

Trojaner Ein Programm, das ohne Wissen des Benutzers heimlich und unautorisiert Aufgaben durchführt. Diese können die Sicherheit des Systems verletzen.

Zugriffskontrolle Jedes Mittel, Gerät oder Technik die es einem Administrator erlaubt, bestimmten Benutzern den Zugriff auf bestimmte Ressourcen zu verweigern oder zu gewähren. Z.B. auf Dateien, Verzeichnisse, Programme, Server oder Netzwerke.

Kapitel 11

Danksagung und Erklärung

11.1 Danksagung

Ich bedanke mich bei meinen beiden Betreuern Prof. Peter B. Ladkin und I Made Wiryana für Ihre Unterstützung bei meiner Diplomarbeit.

Ferner bedanke ich mich bei Heiko Holtkamp und Martin Ellermann, die mir bei verschiedenen Problemen die Worksphere betreffend helfen konnten.

11.2 Erklärung zu meiner Diplomarbeit

Name: Andre Döring Matrikel Nr.: 1340768

Ich erkläre, daß ich die Diplomarbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Bielefeld, den 16.10.2003

Andre Döring