



Fachbereich 12
Betriebssysteme
und verteilte Systeme



Computergestütztes Adaptives Testsystem in Ruby on Rails

Diplomarbeit

Stefan Schöttelndreyer
Matrikelnummer 652351
Studiengang: Angewandte Informatik
mit Anwendungsfach Elektrotechnik

Lehrstuhl Betriebssysteme und verteilte Systeme,
Fachbereich 12 – Elektrotechnik und Informatik, Universität Siegen
Hölderlinstraße 3, 57068 Siegen

1. Gutachter: Prof. Dr. Roland Wismüller
2. Gutachter: Dipl.-Ing. Andreas Hoffmann

Inhaltsverzeichnis

| | |
|--|-----|
| Inhaltsverzeichnis..... | I |
| Abbildungsverzeichnis..... | III |
| Abkürzungsverzeichnis..... | IV |
| | |
| 1 Einleitung und Motivation..... | 1 |
| | |
| 2 Grundlagen des adaptiven Testens..... | 3 |
| 2.1 Definition..... | 3 |
| 2.2 Die Geschichte des computergestützten adaptiven Testens..... | 6 |
| 2.3 Testtheorien..... | 6 |
| 2.3.1 Die Klassische Testtheorie (KTT)..... | 7 |
| 2.3.2 Die Item-Response-Theorie (IRT)..... | 8 |
| 2.3.2.1 Das 1-PL-Modell..... | 9 |
| 2.3.2.2 Das 2-PL-Modell..... | 10 |
| 2.3.2.3 Das 3-PL-Modell..... | 11 |
| 2.3.2.4 Bewertung der Modelle..... | 12 |
| 2.4 Adaptive Teststrategien..... | 13 |
| 2.4.1 Testbeginn..... | 13 |
| 2.4.2 Itemauswahl..... | 14 |
| 2.4.3 Fähigkeitsschätzung..... | 15 |
| 2.4.4 Abbruchstrategien..... | 18 |
| 2.5 Adaptives Testen an Hochschulen..... | 20 |
| | |
| 3 Bestehende adaptive Testsysteme..... | 21 |
| 3.1 CAT-ASVAB..... | 21 |
| 3.2 PersonFit..... | 22 |
| 3.3 IRT-CAT..... | 23 |
| 3.4 MATHCAT..... | 24 |
| 3.5 Weitere adaptive Testsysteme..... | 25 |
| 3.6 Bewertung der Systeme..... | 25 |
| | |
| 4 Konzept und Architektur..... | 27 |
| 4.1 Das Framework Ruby on Rails..... | 27 |
| 4.2 Das Konzept von CAT on Rails..... | 29 |
| | |
| 5 Projektstudie CAT on Rails..... | 33 |
| 5.1 Benutzerschnittstelle..... | 33 |
| 5.2 Datenbankstruktur..... | 35 |
| 5.3 Programmstruktur..... | 36 |
| 5.4 Algorithmen..... | 37 |
| 5.4.1 Latenzberechnung..... | 38 |
| 5.4.2 Korrektheit der Antwort..... | 38 |
| 5.4.3 Itemauswahl..... | 39 |
| 5.4.4 Fähigkeitsschätzung..... | 41 |
| 5.4.5 Beendigung des Tests..... | 42 |

| | |
|--|----|
| 5.5 Integration..... | 43 |
| 5.5.1 Das computergestützte Klausursystem KLAUSIE..... | 43 |
| 5.5.2 Vor- und Nachteile einer Integration..... | 45 |
| 5.5.3 Voraussetzungen für eine Integration in KLAUSIE..... | 45 |
| 6 Fazit und Ausblick..... | 47 |
| Literaturverzeichnis..... | 49 |
| Anhang A..... | 51 |
| Anhang B..... | 51 |

Abbildungsverzeichnis

| | |
|---|----|
| Abbildung 2-1: Adaptiver Testablauf..... | 4 |
| Abbildung 2-2: Verwendung der Items bei P&P bzw. CAT..... | 5 |
| Abbildung 2-3: IC-Funktion im 1-PL-Modell..... | 10 |
| Abbildung 2-4: IC-Funktion im 2-PL-Modell..... | 11 |
| Abbildung 2-5: IC-Funktion im 3-PL-Modell..... | 12 |
| Abbildung 2-6: CAT-Testverlauf..... | 14 |
| Abbildung 2-7: Likelihood-Funktion für einen 3-Item-Test..... | 16 |
| Abbildung 2-8: Log-Likelihood-Funktion für einen 3-Item-Test..... | 17 |
| Abbildung 2-9: Dichtefunktion der Standardnormalverteilung..... | 18 |
| Abbildung 3-1: Vereinfachtes Flussdiagramm des PersonFit-Algorithmus..... | 23 |
| Abbildung 3-2: Testdurchführung in MATHCAT..... | 25 |
| Abbildung 3-3: Prinzipieller Ablauf eines CAT..... | 26 |
| Abbildung 4-1: Model-View-Controller..... | 28 |
| Abbildung 4-2: Auszug der Verzeichnisstruktur in Rails..... | 28 |
| Abbildung 5-1: Benutzerschnittstelle aus Sicht des Teststellers..... | 33 |
| Abbildung 5-2: Datenbankstruktur von CAT on Rails..... | 35 |
| Abbildung 5-3: Klassen des CAToR-Modells und deren Beziehungen..... | 36 |
| Abbildung 5-4: Controller-Klassen in CAT on Rails..... | 37 |
| Abbildung 5-5: Datenbankstruktur von KLAUSIE..... | 44 |
| Abbildung A-1: Ordnerstruktur /..... | 51 |
| Abbildung B-1: Ordnerstruktur /src..... | 52 |

Abkürzungsverzeichnis

| | |
|-------------|---|
| 1-PL-Modell | Einparametrisches logistisches Modell |
| 2-PL-Modell | Zweiparametrisches logistisches Modell |
| 3-PL-Modell | Dreiparametrisches logistisches Modell |
| CAST | Computerized Adaptive Screening Test |
| CAT | Computerized Adaptive Test |
| CAToR | Computerized Adaptive Test on Rails |
| CAT-ASVAB | Computerized Adaptive Testing Version of the Armed Services Vocational Aptitude Battery |
| DRY | Don't Repeat Yourself |
| ETS | Educational Testing Service |
| GRE | Graduate Record Examinations |
| IBT | Internetbasierter Test |
| IRT | Item-Response-Theory |
| KTT | Klassische Testtheorie |
| MAP | Maximum-A-Posteriori-Methode |
| MLE | Maximum Likelihood Estimation |
| MVC | Model-View-Controller |
| P&P Test | Paper & Pencil Test |
| SEM | Standard Error of Measurement |
| TOEFL | Test of English as a Foreign Language |

1 Einleitung und Motivation

Durch die Entwicklung der letzten Jahrzehnte bekommen Computer eine immer größere Bedeutung. Viele Aufgaben des täglichen Lebens wie beispielsweise die Zubereitung des allmorgendlichen Espresso oder das Kaufen einer Bahnfahrkarte werden an Automaten erledigt. Dieser Prozess der Automatisierung wird auch in Zukunft weiter voranschreiten und auf immer mehr Bereiche ausgedehnt werden, die Wahrnehmung hierfür wird jedoch sinken [Lan03]. Die aus der Automatisierung resultierenden Vorteile liegen auf der Hand. Zum einen ist es kosteneffizienter und zum anderen können Maschinen bzw. Programme eine Präzision erreichen, die außerhalb der Fähigkeiten von Menschen liegt. Auch im Hochschulbereich werden zunehmend Programme eingesetzt, um den Lernprozess der Studierenden zu unterstützen und deren Wissensstand zu prüfen, denn auch in diesem Umfeld lassen sich die Abläufe gegenüber herkömmlichen Verfahren effizienter gestalten. Über die reine Durchführung hinaus kann eine computerbasierte Klausur beispielsweise zumindest teilweise automatisch korrigiert werden, was den Korrektoren mehr Zeit verschafft, um individueller auf die Studierenden einzugehen und sich intensiver auf ihre Lehrveranstaltungen vorzubereiten. Auf diese Weise kann die Automatisierung zur Verbesserung der Lehre beitragen.

Betrachtet man die aktuelle Prüfungssituation an Hochschulen, so können die meisten Prüfungen zwei Gruppen zugeordnet werden: die oben schon erwähnte Gruppe der Klausuren sowie mündliche Prüfungen. Im Bereich der Klausuren werden automatisierte Verfahren heute schon produktiv eingesetzt, als Beispiel sei hier das Klausursystem KLAUSIE (vgl. Kapitel 5.5.1) der Universität Siegen genannt. Die Umsetzung einer schriftlichen Prüfung mit Papier und Stift (englisch „Paper & Pencil“, kurz P&P) in ein computerbasiertes System ist vergleichsweise einfach, da im Vorfeld alle Aufgaben und evtl. sogar die Reihenfolge der Aufgaben feststehen.

Mündliche Prüfungen hingegen zeichnen sich vor allem dadurch aus, dass der Prüfer sehr individuell auf den Prüfling eingehen kann und so seine Fähigkeiten mehr oder weniger schnell und genau einzuschätzen weiß. Hierbei werden, bewusst oder unbewusst, adaptive Strategien eingesetzt, d. h. der Prüfer passt seine Fragen an das Antwortverhalten des Prüflings an und wählt sie so, dass er mit möglichst wenigen Fragen eine präzise Einschätzung der Fähigkeit des Studenten bekommt. Aufgrund dieser Eigenschaft zählen mündliche Prüfungen zur Gruppe der adaptiven Tests. Im Gegensatz zu einer Klausur haben mündliche Prüfungen zudem den Vorteil, dass das Ergebnis dem Studenten unmittelbar nach der Prüfung mitgeteilt werden kann.

Die Strukturen einer mündlichen Prüfung zu automatisieren und in einem computer-gestützten Testsystem zur Anwendung zu bringen, um die Vorteile dieser Prüfungsform nutzen zu können, ist ein sehr komplexes Problem, mit dem sich Forscher seit den 1970er Jahren bis heute beschäftigen. Diese Systeme werden unter dem Begriff „computerized adaptive testing“ oder kurz CAT zusammengefasst. Im Gegensatz zu einer computer-gestützten Klausur müssen in einem CAT-System die Mechanismen einer mündlichen Prüfung über komplexe Algorithmen abgebildet werden. Dieser Umstand stellt eine besondere Herausforderung an den Teststeller dar, denn die Algorithmen können nie alle vollkommen generisch genutzt werden. Sie sollten achtsam gewählt und an den jeweiligen Test angepasst sein.

Es existieren zwar schon Programme, die der Durchführung adaptiver Tests dienen (siehe Kapitel 3), jedoch kann aus den dort angeführten Gründen zum jetzigen Zeitpunkt keines der an jener Stelle beschriebenen Systeme für die Umsetzung eines computerbasierten adaptiven Tests als Ergänzung zu mündlichen Prüfungen an Hochschulen herangezogen werden. Ziel dieser Arbeit ist es, diese Lücke zu schließen, indem eine Anwendungssoftware entwickelt wird, mit der die Machbarkeit eines solchen adaptiven Testsystems überprüft werden kann.

Im Verlauf dieser Arbeit werden dafür in Kapitel 2 zunächst die Grundlagen des adaptiven Testens erläutert. Hierzu gehören eine Definition, eine Beschreibung des Ablaufs eines adaptiven Tests, eine Diskussion über die Vor- und Nachteile von adaptiven Tests im Vergleich zu klassischen Tests, das Darlegen der historischen Entwicklung von CAT-Systemen sowie eine Erläuterung der Theorien und Modelle, auf denen Tests basieren. Des Weiteren werden im Grundlagenteil die sogenannten „adaptiven Teststrategien“ dargestellt. Darunter versteht man die unterschiedlichen Module, aus denen ein adaptiver Test besteht.

Im Anschluss an die Grundlagen werden in Kapitel 3 bereits vorhandene CAT-Systeme und deren Vorgehensweise geschildert. Ziel dieser Betrachtung ist die schon vorweggenommene Erkenntnis, ob ein bestehendes System für den Einsatz in Hochschulen verwendet werden kann. Da aufgezeigt wird, dass dies nicht der Fall ist, muss ein solches System entwickelt werden.

Für die Umsetzung der in den Grundlagen dargelegten Theorien in ein CAT-System werden in Kapitel 4 zunächst die Architektur und die Prinzipien des Frameworks „Ruby on Rails“ beschrieben, in dem die Software entwickelt wird. Anschließend erfolgt eine Darstellung des Konzepts von dem im Rahmen dieser Arbeit entwickelten System.

Das folgende Kapitel befasst sich mit der Umsetzung der in den Grundlagen theoretisch und im Konzept abstrakt beschriebenen Algorithmen in ein konkretes adaptives Testprogramm. Darüber hinaus wird dem Leser ein Überblick über die Benutzerschnittstelle, die Datenbank- sowie die Programmstruktur des entwickelten CAT-Systems gegeben. Abschließend wird exemplarisch eine Eingliederung des erstellten CAT-Systems in das schon bestehende Programm KLAUSIE diskutiert.

2 Grundlagen des adaptiven Testens

Dieses Kapitel erläutert die Grundlagen, auf denen Tests, vor allem aber computergestützte adaptive Tests beruhen. Es wird erklärt, was ein adaptiver Test ist, welche Fähigkeiten solch ein Test bietet und welche Vor- bzw. Nachteile er gegenüber klassischen Tests aufweist. Im Anschluss wird ein historischer Überblick von den frühen schriftlichen adaptiven Tests bis hin zu modernen, computergestützten adaptiven Tests gegeben. Darauf folgend werden die zugrunde liegenden Testtheorien und ggf. ihre Modelle dargelegt und im Hinblick auf das zu erstellende Testsystem bewertet. Darüber hinaus werden die für computergestützte adaptive Tests wesentlichen Strategien wie beispielsweise die Auswahl der Aufgaben oder die Fähigkeitsschätzung des Probanden aufgezeigt.

2.1 Definition

Dies Dokument ist nicht vollständig. Es soll lediglich einen Überblick über die behandelte Thematik geben. Wenn Sie an einer ungekürzten Fassung interessiert sein sollten wenden Sie sich per Mail an stefan (at) schoettelndreyer (dot) org.

2.2 Die Geschichte des computergestützten adaptiven Testens

2.3 Testtheorien

2.3.1 Die Klassische Testtheorie (KTT)

2.3.2 Die Item-Response-Theorie (IRT)

2.3.2.1 Das 1-PL-Modell

2.3.2.2 Das 2-PL-Modell

2.3.2.3 Das 3-PL-Modell

2.3.2.4 Bewertung der Modelle

2.4 *Adaptive Teststrategien*

2.4.1 Testbeginn

2.4.2 Itemauswahl

2.4.3 Fähigkeitsschätzung

2.4.4 Abbruchstrategien

2.5 *Adaptives Testen an Hochschulen*

3 Bestehende adaptive Testsysteme

3.1 *CAT-ASVAB*

3.2 *PersonFit*

3.3 *IRT-CAT*

3.4 *MATHCAT*

3.5 *Weitere adaptive Testsysteme*

3.6 *Bewertung der Systeme*

4 Konzept und Architektur

4.1 Das Framework Ruby on Rails

4.2 Das Konzept von CAT on Rails

5 Projektstudie CAT on Rails

5.1 *Benutzerschnittstelle*

5.2 *Datenbankstruktur*

5.3 *Programmstruktur*

5.4 *Algorithmen*

5.4.1 Latenzberechnung

5.4.2 Korrektheit der Antwort

5.4.3 Itemauswahl

5.4.4 Fähigkeitsschätzung

5.4.5 Beendigung des Tests

5.5 *Integration*

5.5.1 Das computergestützte Klausursystem KLAUSIE

5.5.2 Vor- und Nachteile einer Integration

5.5.3 Voraussetzungen für eine Integration in KLAUSIE

6 Fazit und Ausblick

Die Zielsetzung dieser Arbeit war es, ein System zu entwickeln, mit dem computergestützt adaptive Tests zur Verwendung als Prüfungsform im Hochschulbereich durchgeführt werden können. Zu diesem Zweck wurden zunächst die Grundlagen erläutert, auf denen computergestützte adaptive Tests basieren. Im Speziellen wurde adaptives Testen definiert und charakteristische Eigenschaften wurden aufgezeigt. Nach einer kurzen Darstellung der historischen Entwicklung computergestützter adaptiver Tests folgte eine Einführung in psychometrische Testtheorien, welche die Basis wissenschaftlich begründbarer Tests bilden. Im Anschluss daran wurden die Modelle der Item-Response-Theorie und die wichtigsten Strategien zur Durchführung von Tests, basierend auf der IRT, vorgestellt. Kapitel 3 stellte die Funktionsweise und Umsetzung der Strategien einiger in Anwendung befindlicher CAT-Systeme dar.

Da keines der vorgestellten Systeme für die in dieser Arbeit zu lösende Problemstellung herangezogen werden konnte, wurde in Kapitel 4 ein eigenes Konzept für ein CAT-System mit dem Namen CAT on Rails entwickelt. Hierzu wurde zunächst ein kurzer Überblick über das verwendete Framework Ruby on Rails gegeben. Im Anschluss daran wurden die Strategien aufgezeigt, die in CAT on Rails Verwendung finden und schon zuvor in Kapitel 2.4 eingehend erläutert wurden.

Kapitel 5 befasste sich mit der Umsetzung von CAToR. Hier wurde die Benutzerschnittstelle und die zugrunde liegende Datenbankstruktur dargestellt, ein Überblick über die Programmstruktur mit den verwendeten Klassen des Modells und der Programmsteuerung gegeben und die Umsetzung der zuvor theoretisch beschriebenen Strategien in konkrete Algorithmen erklärt. Abschließend wurde eine mögliche Integration in das bestehende Klausursystem KLAUSIE mit ihren Vor- und Nachteilen diskutiert und es wurden einige Überlegungen aufgezeigt, die der Planung einer solchen Eingliederung dienen.

Am Beispiel des im Rahmen dieser Arbeit entwickelten Systems CAToR wurde dargelegt, auf welche Weise ein computergestütztes adaptives Testsystem erstellt werden kann. Die Entwicklung von CAToR zeigte jedoch gleichsam die Grenzen dieser Arbeit auf. Die Idee, das KLAUSIE-System um eine adaptive Komponente zu erweitern, um mittelfristig mündliche Prüfungen durch einen computergestützten adaptiven Test zu ersetzen, muss durch die Erkenntnisse dieser Arbeit etwas gedämpft werden. Vor allem zwei Gründe führen zu dieser Einschätzung. Zum einen müssen die im Testsystem verwendeten Algorithmen unter realen Bedingungen auf ihre Eignung hin überprüft und gegebenenfalls angepasst oder ersetzt werden. Sollte sich beispielsweise bei einem Test unter realen Bedingungen herausstellen, dass dem Testkandidaten zu viele oder zu wenige Aufgaben gestellt werden, so muss man den Schwellwert des Standardfehlers (siehe Listing 7) nach oben bzw. unten korrigieren. Eine solche Anpassung kann jedoch zur Folge haben, dass die Fähigkeit des Kandidaten nicht mehr zufriedenstellend berechnet wird. Dieser Missstand wiederum könnte durch die Wahl eines anderen Algorithmus zur Fähigkeitsschätzung des Probanden (siehe Listing 6) eliminiert werden. Eine umfassende Beschreibung der Algorithmen bietet [Bak04]. Eine Anpassung der Algorithmen ist für jeden Test, d. h. im konkreten Fall für jedes zu prüfende Fach neu vorzunehmen, was den Verwaltungsaufwand erhöht und weitreichende Kenntnis beim Teststeller voraussetzt.

Ein zweiter Grund ist der kalibrierte Itempool. Die modellkonforme Kalibrierung der Items basiert wie alle CAT-Strategien auf komplexen statistischen Modellen. Eine genaue Evaluierung der Itemparameter ist die Grundvoraussetzung für die Verwendung eines CAT im Produktionsbetrieb. Für den Betrieb eines CAT-Systems zur Überprüfung der Fähigkeit von Studierenden im Hochschulbereich als Ergänzung oder Ersatz von mündlichen Prüfungen müssten zunächst modellkonforme Itempools aller zu prüfender Fächer erstellt werden, was einen sehr hohen Anfangsaufwand bedeutet. Wie schon im letzten Kapitel erwähnt, sollte die Itembank, welche in diesem Fall für jedes zu prüfende Fach erstellt werden müsste, nach [Ham79] und [Gre84] mindestens 1000 Items enthalten und zur Kalibrierung von jeweils mindestens 300 Testpersonen beantwortet werden. Wegen dieses hohen Aufwandes bei der Erstellung eines CAT ist eine Kosten-Nutzen-Analyse vor dem Erstellen äußerst wichtig. Der initiale Aufwand zur Erstellung eines modellkonformen Itempools und zur Justierung der benutzten Algorithmen kann auch ein Grund dafür sein, dass die meisten bestehenden CAT-Systeme Tests abbilden, die für eine große Anzahl von Probanden ausgelegt sind.

Eine weiterführende Arbeit könnte den Bereich der Erstellung einer modellkonformen, kalibrierten Itembank thematisieren. Wie schon angesprochen, bildet eine kalibrierte Itembank die Grundlage eines produktiv eingesetzten CAT und ist somit auch eine Voraussetzung für die im letzten Kapitel thematisierte Integration von CAToR in KLAUSIE. Da die Erstellung eines Itempools thematisch in den Bereich der Psychologie einzuordnen ist, wäre eine interdisziplinäre Zusammenarbeit mit Studierenden der Psychologie wünschenswert. Die Eingliederung der in dieser Arbeit entwickelten Programmfunktion in das KLAUSIE-System könnte Gegenstand einer weiteren Arbeit sein.

Literaturverzeichnis

- [Bak01] Baker, Frank: *The Basics of Item Response Theory*, Second Edition, ERIC Clearinghouse on Assessment and Evaluation, Madison, 2001
- [Bak04] Baker, Frank; Kim, Seock-Ho: *Item response theory – Parameter estimation techniques*. Second Edition, Dekker Verlag, New York, 2004
- [Che98] Chen, S.-K.: *A comparison of maximum likelihood estimation and expected a posteriori estimation in computerized adaptive testing using the generalized partial credit model*. Educational and Psychological Measurement, Vol. 58, Nr. 4, 569-595, 1998
- [Emb00] Embretson, Susan E.; Reise, Steven P.: *Item Response Theory for Psychologists*. Lawrence Erlbaum Associates, Mahwah, New Jersey, 2000
- [Gut90] Guthke, J.; Böttcher, H. R.; Sprung, L.: *Psychodiagnostik*. Thieme Verlag, Stuttgart, 1990
- [Gre84] Green, Bert F.; Bock, R. Darrell; Humphreys, Lloyd G.; Linn, Robert L.; Reckase, Mark D.: *Technical Guidelines for Assessing Computerized Adaptive Tests*, Journal of Educational Measurement, Volume 21 Nr. 4, Seite 347 - 360, 1984
- [Ham79] Hambleton, R. K.; Cook, L. L.: *The robustness of latent trait models and effects of test length and sample size on the precision of ability estimates*. Proceeding of the 1979 Computerized Adaptive Testing Conference. Department of Psychology, University of Minnesota, 1979
- [Hof08] Hoffmann, A.; Wismüller, R.: *Sicherheitskonzept für elektronische Prüfungen an Hochschulen auf Basis eines ticketbasierten, virtuellen Dateisystems*. In Seehusen, S.; Lucke, U.; Fischer, S. (Hrsg.): Die 6. e-Learning Fachtagung Informatik - DeLFI 2008, Lübeck, S.197-208, 2008
- [Hor76] Hornke, Lutz F.: *Grundlagen und Probleme antworabhängiger Testverfahren*. Haag + Herchen, Frankfurt am Main, 1976
- [Kin89] Kingsbury, G. Gage; Zara, Anthony R.: *Procedures for Selecting Items for Computerized Adaptive Tests*, Applied Measurement in Education, 2, Seite 359-375, 1998
- [Lan03] Langmann, Reinhard: *Taschenbuch der Automatisierung*, Hanser Fachbuchverlag, 2003
- [Lee06] Lee, Yoon-Hwan; Park, Jung-Ho; Park, In-Yong: *Estimation of an Examinee's Ability in the Web-Based Computerized Adaptive Testing Program IRT-CAT*. Journal of Educational Evaluation for Health Professions, 3. Jg., Artikel Nr. 4, Dezember 2006
- [Len07] Lenz, Patrick: *Build Your Own Ruby On Rails Web Applications*. SitePoint Pty. Ltd., Collingwood. 2007
- [Lin06] Linacre, John Michael: *Computer Adaptive Tests (CAT), Standard Errors and Stopping Rules*, Rasch Measurement Transactions, Institute for Objective Measurement, Inc., Chicago, 2006
- [Lie98] Lienert, Gustav A.; Raatz, Ulrich: *Testaufbau und Testanalyse*. 6. Auflage, Studienausgabe, Psychologie-Verl.-Union, 1998
- [Lin00] Linden van der, Wim J.; Glas, Cees A.W.: *Computerized Adaptive Testing – Theory and Practice*. Kluwer Academic Press, Dordrecht, 2000
- [Lor68] Lord, Frederic M., Novick, Melvin R.: *Statistical Theories of Mental Test Scores*, Addison-Wesley Pub. Co., 1968

- [Lor71] Lord, Frederic M.: *A Theoretical Study of Two-Stage Testing*. Psychometrika, 36 Jg., Nr. 3, S. 227-242, 1971
- [Lor80] Lord, Frederic M.: *Applications of item response theory to practical testing problems*, Lawrence Erlbaum Associates Inc., Hillsdale, 1980
- [McB75] McBride, James R.: Scoring adaptive tests. In David J. Weiss (Ed.): *Computerized adaptive trait measurement – problems and prospects*, Psychometric Methods Program, Department of Psychology, University of Minnesota, Minneapolis, 1975
- [Mei99] Meijer, R. R. & Nering, M. L.: *Computerized adaptive testing. Overview and introduction*. Applied Psychological Measurement, 23, 187-194, 1999
- [Moo07] Moosbrugger, H.; Kelava, A.: *Testtheorie und Fragebogenkonstruktion*. Springer Verlag, Heidelberg, 2007
- [Ort00] Ortega, Manuel; Bravo, José: *Computers and education in the 21st century*. Kluwer Academic Publisher, Dordrecht, 2000
- [Owe75] Owen, Roger J.: *A Bayesian Sequential Procedure for Quantal Response in the Context of Adaptive Mental Testing*. Journal of the American Statistical Association, Vol. 70, Nr. 350, Seite 351-356, 1975
- [Ros96] Rost, J.: *Lehrbuch Testtheorie und Testkonstruktion*. Huber Verlag, Bern, 1996
- [Ros99] Rost, J., Carstensen, C. H.; von Davier, M.: *Sind die Big Five Raschskalierbar? Eine Reanalyse der NEO-FFI-Normierungsdaten*. Diagnostica, 45, 119-127, 1999
- [San97] Sands, William A.; Waters, Brian K.; McBride, James R.: *Computerized Adaptive Testing – From Inquiry to Operation*. American Psychological Association, Washington, 1997
- [Sch99] Schmidt, Jens U.; Gutschow, Katrin: *Vom Papier zum Bildschirm – Computergestützte Prüfungsformen*. W. Bertelsmann Verlag, Bielefeld, 1999
- [Sod07] Sodoké, Komi; Raïche, Gilles; Nkambou, Roger: The adaptive and intelligent testing framework: PersonFit, Seventh IEEE International Conference on Advanced Learning Technologies, 2007
- [Spe07] Spearman, C.: *Demonstration of formulae for true measurement of correlation*. American Journal of Psychology, University of Illinois Press, 1907
- [Tho05] Thomas, Dave; Heinemeier Hansson, David: *Agile Web Development with Rails*, 4. Auflage, The Pragmatic Bookshelf, Dallas, 2005
- [Wai00] Wainer, Howard: *Computerized Adaptive Testing – A Primer*. Second Edition, Lawrence Erlbaum Associates, Mahwah, New Jersey, 2000
- [Wei82] Weiss, David J.: *Improving Measurement Quality and Efficiency with Adaptive Testing*, 1982
- [Wei83] Weiss, David J.: *New Horizons in Testing – Latent Trait Test Theory and Computerized Adaptive Testing*. Academic Press, New York, 1983
- [Wei84] Weiss, David J.; Kingsbury, G. Gage: *Application of Computerized Adaptive Testing to Educational Problems*, Journal of Educational Measurement, Volume 21 Nr. 4, Seite 361 - 375, 1984
- [Wis90] Wise, L. L.; McHenry, J. J.; Chia, W. J.; Szenas, P. L., McBride J. R.: *Refinement of the computerized adaptive screening test*. U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, 1990
- [Wot80] Wottawa, H.: *Grundriß der Testtheorie*. Juventa Verlag, München, 1980

Anhang A

Die Anhänge zu dieser Arbeit sind auf dem beiliegenden Datenträger gespeichert. Einen Teil der Ordnerstruktur stellt Abbildung A-1 dar.








Abbildung A-1: Ordnerstruktur /

Im Unterverzeichnis `doc` des Verzeichnisses `Diplomarbeit` befindet sich die Datei „Computergestuetztes Adaptives Testsystem in Ruby on Rails.pdf“, welche diese Arbeit im Portable Document Format darstellt. Der Ordner `src` beinhaltet das im Rahmen dieser Arbeit entstandene Programm CAT on Rails (siehe Anhang B).

Anhang B

Im Verzeichnis `src`, welches sich auf dem beiliegenden Datenträger unter dem Ordner `Diplomarbeit` befindet, liegt der Quellcode des Programms CAT on Rails. Er ist wiederum in einem Ordner mit dem Namen `CATOR` zusammengefasst und beinhaltet weitere Unterordner. Ein Auszug der Ordnerstruktur veranschaulicht Abbildung B-1. Die wichtigsten Ordner und die sich darin befindlichen Dateien werden im Folgenden erläutert.

-  `app`: In diesem Ordner befindet sich der Programmcode in verschiedenen Unterverzeichnissen. Der genaue Inhalt dieser Unterverzeichnisse wird im nächsten Absatz erläutert.
-  `controllers`: Die Controller-Klassen des Programms sind hier in separaten Dateien gespeichert.
-  `models`: Jede Datei in diesem Ordner beinhaltet eine Modell-Klasse der Programmarchitektur.
-  `views`: Für jedes Modell der Programmarchitektur existiert in diesem Ordner ein korrespondierender Unterordner, in dem die Vorlagen für das Rendering im Browser liegen. Zusätzlich beinhaltet der Ordner einen Unterordner namens `layouts`, in dem globale Vorlagen gespeichert sind.
-  `config`: Dieser Ordner enthält als erwähnenswerte Dateien `database.yml`, in welcher der Datenbankadapter (in diesem Fall SQLite 3) festgelegt wird und `routes.rb`, welche die Konfiguration des Routings, d. h. die Umleitung der Benutzeranfragen an den richtigen Controller, beinhaltet.

- 📁 db: Der Ordner db enthält die SQLite-Datenbank in der Datei `development.sqlite3`, das aktuelle Datenbankschema in der Datei `schema.rb` und den Unterordner `migrate`, in dem die Dateien der Migrations¹ liegen.
- 📁 public: Dieser Ordner enthält aus dem Web zugängliche Dateien und Ordner wie Bilder, Stylesheets, JavaScript, et cetera.
- 📁 script: In diesem Ordner liegen Scripte u. a. zum Ausführen des Webservers, der Generatoren oder der Entwicklungskonsole.
- 📁 tmp: Hier werden temporäre Daten abgelegt.
- 📁 vendor: In diesem Ordner wird Code von Drittanbietern gespeichert. Installierte Plugins befinden sich z. B. im Unterverzeichnis `plugins`, die für diese Arbeit genutzte Version 2.2.2 des Rails-Frameworks ist im Unterverzeichnis `rails` abgelegt.

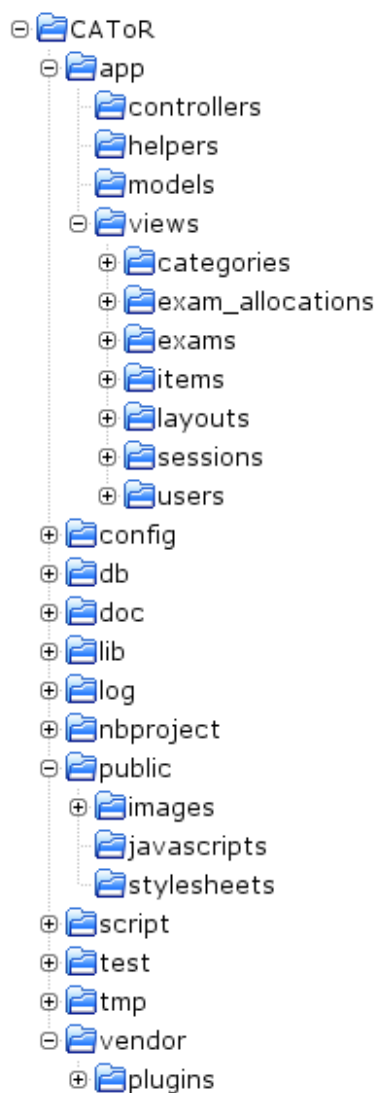


Abbildung B-1: Ordnerstruktur /src

¹ Weitergehende Informationen über Migrations in [Tho05] und [Len07].

Der Unterordner `app` beinhaltet den eigentlichen Programmcode. Aus diesem Grund wird der Inhalt seiner schon erwähnten Unterordner hier noch einmal näher erläutert. Jede Klasse ist in einer eigenen Datei abgelegt. Der Unterordner `controllers` enthält die Controller-Klassen (siehe Abbildung 5-4) in folgenden Dateien:

- `application.rb`: Hier werden die Methoden definiert, die auch in verschiedenen anderen Controllern Anwendung finden. Alle anderen Controller erben von dieser Klasse (siehe „ApplicationController“ in Abbildung 5-4).
- `exams_controller.rb`: Diese Datei enthält die Klasse `ExamsController`. Herauszustellen ist die Methode `proceed`, in der die Durchführung des Tests mit einem Großteil der in dieser Arbeit beschriebenen Algorithmen kodiert ist.
- `categories_controller.rb`, `users_controller.rb`, `items_controller.rb`, `exam_allocations_controller.rb`: Diese Dateien enthalten die Klassen `CategoriesController`, `UsersController`, `ItemsController` und `ExamAllocationsController`.
- `sessions_controller.rb`: In der Klasse `SessionsController` werden die Methoden für den Login- bzw. Logout-Vorgang definiert.

Im Unterordner `models` sind die Dateien `category.rb`, `exam.rb`, `exam_allocation.rb`, `item.rb`, `level.rb`, `possible_solution.rb`, `solution.rb` und `user.rb` vorhanden. Sie enthalten die Klassen `Category`, `Exam`, `ExamAllocation`, `Item`, `Level`, `PossibleSolution`, `Solution` sowie `User` und repräsentieren das Modell der MVC-Architektur.