



***7363 - Web-basierte Anwendungen***  
***4750 – Web-Engineering***

Eine Vertiefungsveranstaltung



# *Organisatorisches*

Platzvergabe, Teambildung  
Regeln zur Teilnahme und Scheinvergabe  
Gliederung  
Literatur



- Besonderheit im WS 2008/2009
  - Die Bachelor-Vertiefung Web-Engineering wird erstmals angeboten
  - Sie sollte zwar auf dem Bachelor-Pflichtkurs „WBA“ des 5. Fachsemesters aufbauen, aber
    - WBA läuft in diesem Semester parallel
    - Aus höheren Semestern gab es keinen großen Bedarf mehr an dieser Vertiefung
  - Die Folge:
    - WBA wurde nicht zur Voraussetzung erhoben
    - Offenbar gibt es keine WBA-Absolvent(inn)en unter den Kursteilnehmern.
  - Die LV wird erstmals gemeinsam mit der Diplom-Vertiefung WBA gehalten, in Übereinstimmung mit der Äquivalenzliste
    - In diesem Semester gibt es deutlich mehr Diplom- als Bachelor-Studierende im Kurs



- Besonderheit im WS 2008/2009
  - Die LV wird erstmals gemeinsam mit der Bachelor-Vertiefung Web-Engineering gehalten, in Übereinstimmung mit der Äquivalenzliste
    - In diesem Semester gibt es deutlich mehr Diplom- als Bachelor-Studierende im Kurs
  - Der Inhalt der LV hat sich im Vergleich zu früheren WBA-Kursen erheblich geändert:
    - XML als Voraussetzung ist entfallen
    - Web Services, insb. XML-RPC, SOAP & WSDL spielen keine Rolle mehr
    - Es besteht keine Wahlfreiheit mehr bei der Implementierungssprache für das Projekt
    - Im Zentrum steht nicht mehr Interoperabilität von Web Services, sondern strukturiertes Vorgehen bei der Erzeugung i.w. interaktiver Web-basierter Anwendungen
    - Datenbankunterstützung und Sicherheitsaspekte spielen diesmal eine wesentliche Rolle
    - Die Projektaufgabe ist zwar im Kern erhalten geblieben, wurde aber erheblich an die neuen Ziele angepasst



- Ein heterogenes Teilnehmerfeld
  - Im Diplomstudiengang gibt es keinen WBA-Grundkurs
  - Im Bachelor-Studiengang belegen nicht alle den WBA-Grundkurs
  - WBA-Vorkenntnisse sind manchmal gar nicht, manchmal in erheblichem Umfang vorhanden
- Problem
  - Welche Grundlagen sollen wie gründlich behandelt werden?
- Lösungsweg
  - **Fragebogen zur Selbsteinstufung**, anonym
  - Anpassung des Vorlesungsmaterials an die Verteilung der Ergebnisse
  - Grenzen:
    - Gesamtzeit der LV
    - Unter- und Überforderungen bei sehr stark abweichenden Vorkenntnissen
    - Überschneidungen mit dem zeitgleichen WBA-Grundkurs



- Regeln zur Teilnahme und Scheinvergabe
  - Siehe Modulhandbuch: 50% Projekt/Praktikum, 50% mündliche Prüfung
- Platzvergabe, Teambildung
  - Praktikumsplätze wurden zugeteilt. Probleme bitte melden.
  - Das Praktikumsprojekt erfordert (stabile) Zweier-Teams. Bitte in den nächsten zwei Wochen bilden!
- Zeitplan (vorläufig)
  - Siehe Tabelle auf der nächsten Folie



# Termine im WS 2008

(Stand: 25.11.08)



Datum (Mi)	Vorlesung	Praktikum
15.10.08	Einführung, Regeln, RoR-Demo	Platzvergabe, Frageb., Ruby
22.10.08	SMTP, HTTP	Ruby
29.10.08	HTTP, WebDAV	Web-Server installieren
5.11.08	Sessions, Cookies, TLS	HTML-Formulare, CSS
12.11.08	HTML-Formulare, CSS, CGI & Co.	HTML-Formulare
19.11.08	Modellieren / ORM mit ActiveRecord	Rails-Vorübung
26.11.08	ActiveRecord, Migrations	Rails-Vorübung
3.12.08	Projektvorstellung Views mit ActionPack; partials, helpers	Selbständiges Arbeiten, Insb. Projektorganisation
10.12.08	Views mit Ajax. REST, ActiveResource	Projekt
17.12.08	Controller, Routing	Projekt
22.12.08 - 2.1.09	Nein (Weihnachtspause)	
7.01.09	Testing / Filter, Auth.	Projekt
14.01.09	Optimierungen: Caching, ...	Projekt
21.01.09	Sicherheit: CSS, XSS, SQL Inj. ...	Projekt
28.01.09	Rollout, Performance tuning	Projekt
2.02. - 13.02.09	Prüfungswochen / Projektabnahmen!	



# Inhalte des Kurses

Eine Stoffsammlung, weder erschöpfend noch garantiert zu schaffen...



- Kleine Rails-Demo
  - Ein „*appetizer*“, heute ☺
- Vorbereitungen
  - Ruby-“Crashkurs“
- Grundlagen Web-basierter Anwendungen (je nach Vorkenntnissen)
  - Schicht-7-Protokolle: SMTP, HTTP
  - HTML und XHTML (Grundlagen)
  - CSS, CSS2
  - HTTP-Server, CGI-Programmierung, Formulare, Sessions
  - Template-Technik
  - AJAX: Grundlagen
  - REST
  - Mashups am Beispiel des Google API (Maps) – sofern die Zeit gestattet



- Rails-Grundlagen
  - Prinzipien: *Conv. over Conf.*, DRY, MVC; *agile development*, *test-driven*
  - Rails-Versionen
  - Basics 1: Projekteinrichtung, Verzeichnisstruktur; ActionPack & Ruby
  - Basics 2: DB-Einrichtung, *dev/test/prod*, *environment*, *Rakefile*, *shells*, *routes*, *logging*
  - DB: Abstraktion via Adapter, *migrations*, *dbconsole*
  - Model: ORM mit ActiveRecord, CRUD, Validierung, (Fehler-) Meldungen
  - View: Stylesheets, Layouts, Templates, Partials, Form helpers
  - Controller 1: Sessions, Parameters, Rendering, Filter
  - Controller 2: REST, XML u.a. Formate
  - Routing: Konfiguration, REST-Variante, URL-Generierung, helper
  - Tests: *Fixtures*, *unit/functional/integration/performance tests*, *Mock obj.*
  - RESTful *applications* mit ActiveResource



- Rails im Einsatz
  - SSL, Authentifizierung und Rechteverwaltung
  - *Deployment*, Skalierbarkeit, Performance, Webserver-Wahl, Session-Konsistenz; Capistrano/SVN
  - Sicherheitsaspekte: Abwehr/Vorbeugung von SQL injection, XSS etc.
- Rails-Addons und -Plugins
  - E-Mail-Komponente ActionMailer
  - Bitte umblättern: Paginierungstechnik
  - Web 2.0: AJAX, JavaScript-Bibliotheken und ihre Kapselung
  - Mashups am Beispiel des Google-APIs



- Bücher
  - **Programming Ruby (2nd ed.)**  
Dave Thomas. The Pragmatic Bookshelf, Raleigh, NC, 2005.  
ISBN 0-9745140-5-5  
*ACHTUNG: 3. Auflage (zu Ruby 1.9/2.0) für Ende 2008 angekündigt*
  - **Agile Web Development with Rails (2nd ed.)**  
Dave Thomas, David H. Hansson. The Pragmatic Bookshelf, Raleigh, NC, 2006. ISBN 0-9776166-3-0, GTIN 9780977616633  
*ACHTUNG: 3. Auflage (zu Rails 2.x) für Ende 2008 angekündigt*
  - **Ruby on Rails 2**  
Hussein Morsy, Tanja Otto. Galileo Computing, Bonn, 2008.  
ISBN13 9783898427791 (auch on-line einsehbar!)
  - **Rapid Web Development mit Ruby on Rails**  
Ralf Wirdemann, Thomas Baustert. Hanser 2006, ISBN 3-446-40932-7



- Bücher (Forts.)
  - **Deploying Rails Applications**  
E. Zygmontowicz, B. Tate, C. Begin. The Pragmatic Bookshelf, Raleigh, NC, 2008.  
ISBN 0-9787392-0-5
  - **Rails Recipes**  
Chad Fowler. The Pragmatic Bookshelf, Raleigh, NC, 2006.  
ISBN 0-9776166-0-6, GTIN 9780977616602



- Bücher (Forts.)
  - **Apache. The Definitive Guide**  
Ben Laurie & Peter Laurie. 2nd ed. O'Reilly, Sebastopol, CA, 1999.  
ISBN 1-56592-528-9
  - **HTTP kurz & gut**  
Clinton Wong. 1. Aufl. O'Reilly, Köln, 2000. ISBN 3-89721-230-7
  - **CGI Programmierung mit Perl**  
Scott Geulich, Shishir Gundavaram, Gunther Birznieks. 2. Aufl. O'Reilly,  
Köln, 2001. ISBN 3-89721-167-X
- Links
  - Siehe *website* des Kurses, u.a. die dortige Linksammlung:  
<http://www.informatik.fh-wiesbaden.de/~werntges/lv/wba01.html>



# Kurze Einführung in Rails

Kern: Eine kleine Rails-Demo

- Konvention über Konfiguration
- DRY (*Don't Repeat Yourself*)
- ORM
- MVC-Muster
- REST (seit Rails 2.0)
- *Test-driven development, agile*



- PHP
  - Framework „CakePHP“ nimmt Anleihen bei Rails
- Python
  - Framework „Django“ übernimmt einige Rails-Ideen
- Groovy/Java
  - Framework „Grails“ orientiert sich sogar am Namen von Rails...
- Stürmische Entwicklung von RoR, nach wie vor im Gang
  - 13.12.2005: V 1.0, entstanden aus Projekt „Basecamp“
  - 28.03.2006: V 1.1, RJS
  - 19.01.2007: V 1.2, REST, UTF-8 *handling*
  - 07.12.2007: V 2.0, Viele Neuerungen, REST ersetzt SOAP
  - 01.06.2008: V 2.1, Viele Verbesserungen, u.a. neue *migrations*-Verwaltung
  - **21.11.2008**: V 2.2, „I18N“ *support, thread safety*



# AKTUELL – Rails 2.2 erschienen!



- Neu

- *118N*
- *Thread safety, connection pool*
- *Improved HTTP validators: etag, last-modified*
- *Ruby 1.9, JRuby*
- *Better API docs & great guides*
- Quelle: <http://weblog.rubyonrails.org/> (Posted by David November 21, 2008 @ 05:22 PM)

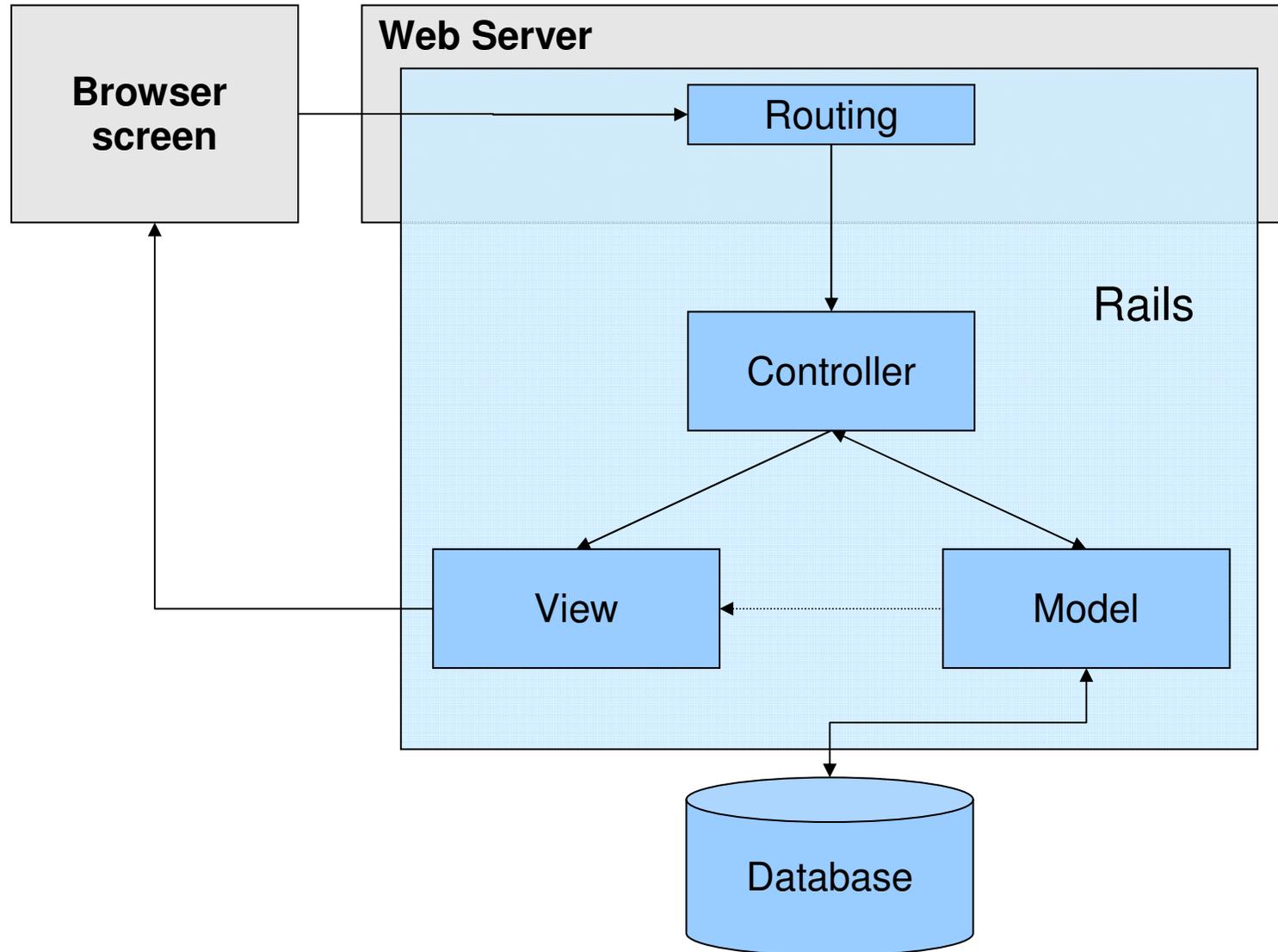
**Entscheidung:**  
Wechseln wir auf  
Rails 2.2 für unser  
Projekt?

- Umstieg:

```
$ sudo gem update --system           # gem 1.3.1 erforderlich!  
$ sudo gem install rails             # Update von Rails selbst  
$ cd /path/to/my/rails/project  
$ rake rails:update                  # Umstellung des Projekts  
#  
# VORSICHT - manche Plug-ins und Gems sind  
# vielleicht noch nicht auf Rails 2.2 angepasst!
```



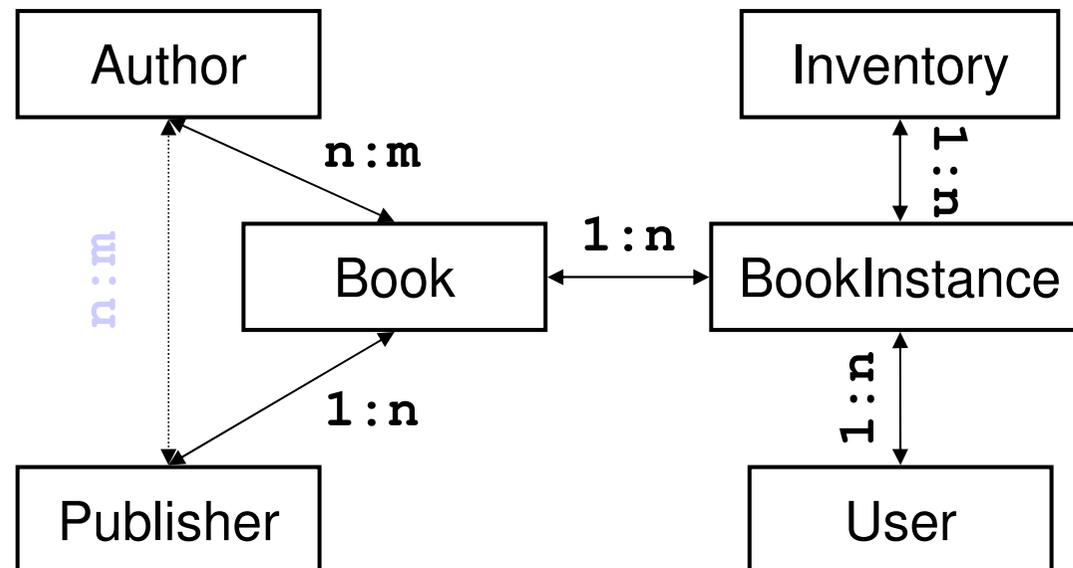
# Grobstruktur einer RoR-Anwendung



- Klassen

- Author
- **Book**
- BookInstance
- Inventory
- Publisher
- User

← unser erstes Ziel, vereinfacht



- Spalte
  - title
  - (authors)<sup>|</sup>
  - publisher\_id
  - pub\_year
  - edition
  - isbn
  - gtin
- Datentyp
  - string
  - (n.a.)<sup>|</sup>
  - integer
  - integer
  - integer
  - string
  - integer

- Spalte
  - title
  - authors
  - publisher
  - pub\_year
  - edition
  - isbn
  - gtin
- Datentyp
  - string
  - string
  - string
  - integer
  - integer
  - string
  - integer

- Spalte
  - book\_id
  - instance\_id
  - borrowed\_at
  - borrower\_id
  - due\_at
  - returned\_at
- Datentyp
  - integer
  - integer
  - timestamp
  - integer
  - date
  - timestamp

- Spalte
  - last\_name
  - first\_name
  - middle\_initials
- Datentyp
  - string
  - string
  - string

- Spalte
  - name
  - address
  - city
  - zip
  - country
- Datentyp
  - string
  - text
  - string
  - integer
  - string

- Spalte
  - last\_name
  - first\_name
  - middle\_initials
  - member\_id
  - member\_since
  - blocked
- Datentyp
  - string
  - string
  - string
  - integer
  - date
  - boolean

- Author
  - books
- Book
  - authors
- BookInstance
  - book
  - borrower
- Inventory
  - borrowed\_items
  - overdue\_items
- Publisher
  - books
  - authors
- User
  - borrowed\_items
  - overdue\_items

- Projekt erstellen

```
$ rails lib
... (viele Verz.& Dateien)
$ cd lib
```
- Gerüst bauen lassen

```
$ script/generate \
  scaffold book
  title:string
  authors:string
  publisher:string
  pub_year:integer
  edition:integer
  isbn:string
  gtin:integer

... (viele Verz.& Dateien)
```
- Erster Blick in die DB

```
$ script/dbconsole
.headers ON
.databases
.tables
```
- Tabelle generieren (Migr.)

```
$ rake db:migrate
```
- DB-Check

```
$ script/dbconsole
.headers ON
.tables
select * from books
```

- Web Server starten
  - \$ `script/server &`
- Browser starten
  - `http://localhost:3000/`
  - `http://localhost:3000/books`
- CRUD-Zyklus
  - Mehrere Bücher anlegen
    - Ein DB-Check pro Schritt
  - Änderungen, Löschungen
    - Jeweils per DB-Check verfolgen

- Modelle
  - Weitere anlegen
  - Relationen herstellen
- *Join table*
  - Migration generieren
  - Einrichten
- Migrationen
  - Ggf. ergänzen
  - Ausführen
- Tests
  - Erster Durchlauf
- Datenbank
  - Opt.: *Initial load*
- Webserver starten!
- Views
  - Eindeutschen
  - Relationen!



# Ruby-Crashkurs...

Stichwort- und Materialsammlung

- Online-Quellen
  - Ruby für Umsteiger, Ruby in 20 Minuten
    - <http://www.ruby-lang.org/de/documentation/ruby-from-other-languages/>
    - <http://www.ruby-lang.org/de/documentation/quickstart/>
  - Das Pickaxe-Buch, erste Auflage
    - Siehe Ruby-LV, Begleitmaterial und Links (deutsch & englisch verfügbar)
  - Ruby-Grundlagen. Kapitel des Rails-Buchs von Wirdemann & Baustert
    - <http://www.rapidwebdevelopment.de/download/ruby.pdf>
  - Einführung in Ruby. Kapitel 4 des Rails-Buchs von Morsy & Otto
    - [http://openbook.galileocomputing.de/ruby\\_on\\_rails/ruby\\_on\\_rails\\_04\\_001.htm](http://openbook.galileocomputing.de/ruby_on_rails/ruby_on_rails_04_001.htm)
  - Try Ruby! Ein 15-Minuten-Tutorial via Browser
    - <http://tryruby.hobix.com/>

- Vorstellen
  - Kommentare, # vs. =begin ... =end
  - **Syntaxregeln**, Stilfragen; Variablen / Referenzen / Objekte / OO-Folgen
  - Strings und expandierte Zeichen wie \t, \n
  - String-Methoden, insb. der Indexoperator und seine Möglichkeiten
  - **Symbole**; RegExp
  - Arrays, Hashes, Ranges
  - Bedingte Verarbeitung: if, while, until, for, case/when
  - Code-Blöcke, **Iteratoren** (insb. „each“)
  - Exception handling
  - Klassen und Vererbung, Attribute & Klassenattribute, Klassenmethoden
  - Reflection, Code-Generatoren (attr\_reader etc.)
  - **Allgemeine Form eines Methodenaufrufs**
    - Variable Argumentliste, impliziter Hash, Defaultbelegungen, Codeblock

- Verweise auf die Ruby-LV (WS 2006)
  - Quelle: PDF-Dateien unter <http://www.informatik.fh-wiesbaden.de/~werntges/lv/ruby/ruby06.html>
  - Bedingte Verarbeitung: if, while, until, for, case/when
    - Teil 01, „Verzweigungen und Schleifen“, S. 71ff
  - Code-Blöcke, **Iteratoren** (insb. „each“)
    - Teil 01, S. 79ff
  - Exception handling
    - Teil 01, S. 112ff
  - **Allgemeine Form eines Methodenaufrufs**
    - Variable Argumentliste, impliziter Hash, Defaultbelegungen, Codeblock
      - Teil 02, Kap. „Aufruf von Methoden“ (S. 2-75ff)