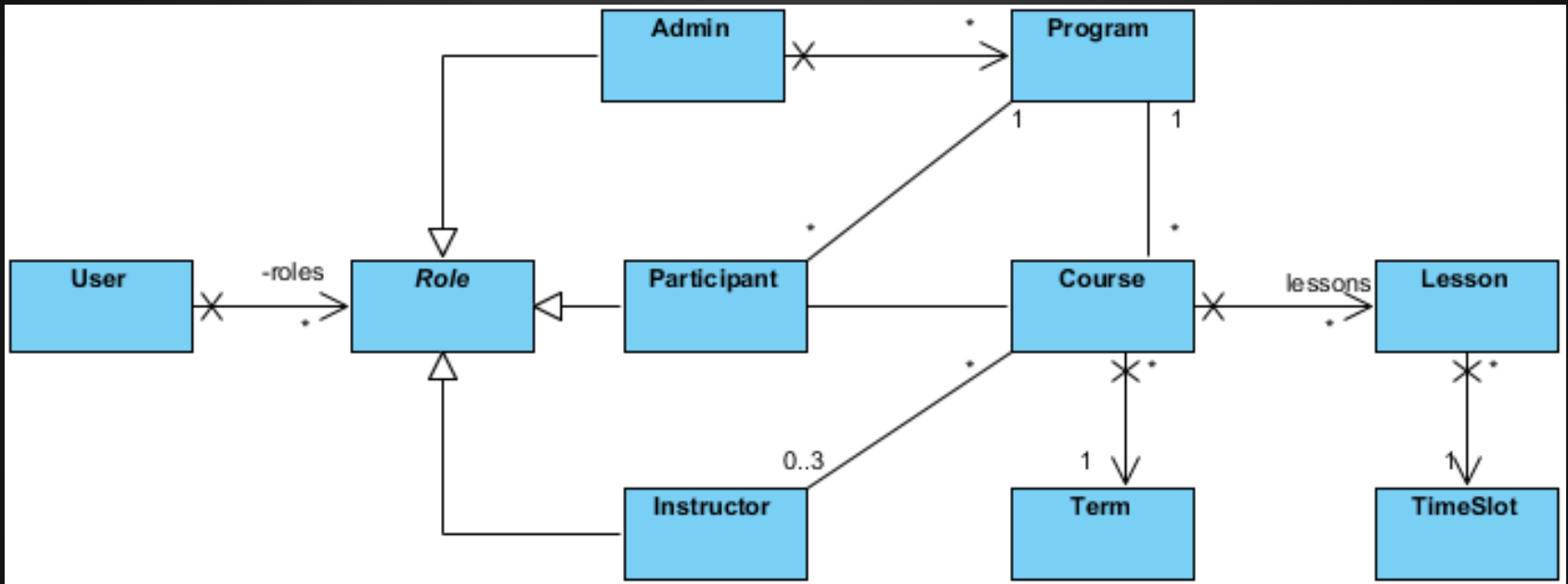


Belegsystem 2.0

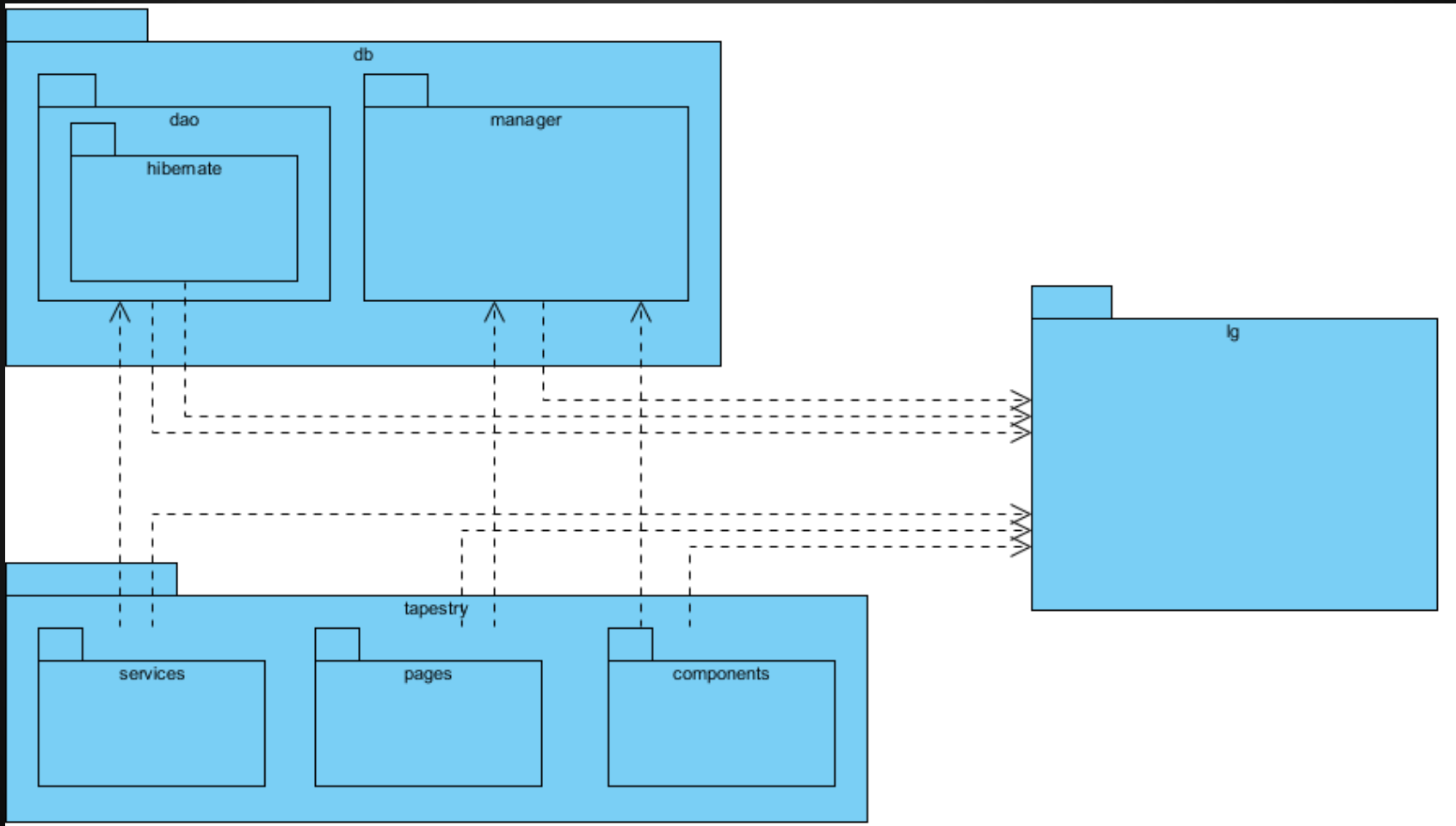


Fabian Dannenberger // Minh Hue Vu // Dominik Hübner
Steven Ulrich // Sascha Reimann

Überblick: Model



Überblick: Architektur



Verwendete Technologien

- Tapestry
 - komponentenorientiertes Open-Source Java Web-Framework
 - inklusive Tapestry-Hibernate
- Hibernate als OR-Mapper
 - MySQL als Datenbank
- MulTex für Exception Reporting
 - erweitert um AnnotatedExc

```
@ExcMessage("Die Unterrichtseinheit {0} steht in zeitlichem Konflikt mit der Unterrichtseinheit {1}")  
public static final class LessonTimeConflictExc extends AnnotatedExc {}
```

Tapestry: Zentrale Eigenschaften

- MVC-Pattern
 - Model: lg-Package, kennt View & Controller nicht
 - Controller: Java-Klasse für Seite im Packages pages
 - View: HTML-basierte Template-Datei zur Java-Klasse
- Live-Class-Reloading
- Komponentenorientierung / POJO-Klassen als Basis
- Geschwindigkeit

Testanwendung* Produktliste plus Kategorien

Framework	16 User / 1.000 Produkte	1 User / 10.000 Produkte
Wicket	499ms	1885ms
Lift	850ms	633ms
Tapestry 5.2.5	192ms	100ms

*<http://www.jtict.com/blog/rails-wicket-grails-play-lift-jsp/>

Tapestry: IoC-Container

- Kern des Frameworks
- keine XML-Konfiguration wie bei Spring nötig
- Über (App)Module-Klasse konfigurierbar
- Meta-Programming bzw. (limitierte) aspektorientierte Programmierung

```
@Advice
@Match("**Manager")
public static void adviseSecurityCheck(final MethodAdviceReceiver receiver, final ApplicationStateManager applicationStateManager) {
    MethodAdvice advice = new MethodAdvice() {
        public void advise(MethodInvocation invocation) {
            User user = applicationStateManager.getIfExists(User.class);
            if (!AccessFilter.checkAccess(user, receiver.getMethodAnnotation(invocation.getMethod(), Restricted.class))) {
                invocation.setCheckedException(MultexUtil.create(InsufficientRightsExc.class));
                invocation.rethrow();
            } else {
                invocation.proceed();
            }
        }
    };
    for (Method method : receiver.getInterface().getMethods()) {
        if (method.isAnnotationPresent(Restricted.class)) {
            receiver.adviseMethod(method, advice);
        }
    }
}
```


Vorführung!

Belegen

localhost:8080/belegsystem/course/book

CSS ABP

Angemeldet: s757997 [Abmelden](#)



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Belegen

Student-Kurse

Profil

Auswahlkriterien zur Belegung

Studiengang: Semester: Zug: Dozenten: Semester (Zeit):

Suche: [Suchen](#) [Zurücksetzen](#)

Kurs	Titel	Semester	Zug	Dozenten	Termine	Auslastung	Aktion
SWP2	Softwareprojekt 2	4	1	Christoph Knabe	Mittwoch 10:00 - 11:30 Mittwoch 08:00 - 09:30 Mittwoch 12:15 - 13:45	1 von 30	stornieren

Stundenplan

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag
08:00			SWP2 (Softwareprojekt 2) Knabe D E16a stornieren			
10:00			SWP2 (Softwareprojekt 2) Knabe D E16a stornieren			