Projekt: swp11-1 erstellt am 23.05.2011
Leiter: Martin Walther
Bearbeiter: Sebastian Schüller , Sebastian Dorn, Matthias Zigan, Daniel Gerighausen,
Stan Altmann, Johann Berger, Martin Walther

Entwurfsbeschreibung für den Update des OLAT-PA Moduls

Inhaltsverzeichnis

1. Allgemeines

Das OLAT-PA Modul ist schon seit einiger Zeit im laufendem Prüfungsbetrieb der Universität Leipzig integriert, es bedarf jedoch eines Updates auf die neue OLAT-Version 7.2. Da es nicht möglich ist das bestehende OLAT-PA einfach zu integrieren sind einige Änderungen des alten Systems nötig, die hier entworfen werden.

1.1 Was ist das OLAT-PA Modul?

Das Prüfungsmodul (OLAT-PA) implementiert die Prüfungsverwaltung der Universität Leipzig am Institut für Informatik in das OLAT-LMS. Mitglieder der Universität können sich an diesem mit ihren Benutzerdaten (studserv-Daten/LDAP) anmelden. Jedes Mitglied hat eine klar definierte und vom System vergebene Rolle z.B. Student, Prüfer oder Prüfungsamt. Für jede Rolle sind verschiedene Sichten, Funktionalitäten und Rechte definiert.
Als Beispiel kann man folgendes Szenario anbringen: ein Prüfer erstellt ein neues Prüfungsangebot PrX, nachdem das Prüfungsamt PrX genehmigt hat, ist es freigeschaltet und für andere zugänglich. Studenten können sich nun an PrX an- und abmelden.
Zusätzlich ist zu sagen, dass das Prüfungsamt die meisten Zugriffsrechte hat: es verwaltet neben dem Administrator des Systems. Eine Trennung zwischen schriftlichen und mündlichen Prüfungen ist momentan nicht zu 100 Prozent gewährleistet.

1.2 Systemvoraussetzungen

Da sich die Systemvoraussetzungen vom eigentlichen OLAT-LMS nicht geändert haben, werden sich diese auch nicht für das OLAT-PA Modul ändern. Grundsätzlich wird ein Server mit entsprechender Breitband-Anbindung an das Internet benötigt, welches Betriebssystem auf diesem zum Einsatz kommt ist nicht relevant. Es müssen folgende Pakete bzw. Software auf dem Serversystem installiert sein:

- MySQL
- Tomcat und
- JRE

Der OLAT-PA Nutzer hingegen benötigt lediglich einen Browser (Firefox, Internet Explorer, Crome usw.) und einen Internetzugang.

1.3 Allgemeines zum Rollenkonzept

Da nicht alle Rollen des OLAT-LMS benötigt werden und ein Sonderfall als neue Rolle auftritt, muss noch definiert werden welche Rolle wie zu verstehen ist. Für die Nutzung des OLAT-PA Moduls benötigen man nur 3 Rollen(siehe 1.1). Die Rolle Student wird dem *DefaultUser* zugewiesen, diese Rolle hat nur lese Rechte. Somit kann sich ein Student an eine freigeschaltete Prüfung an- und wieder abmelden. Für die Rollen Prüfer und Prüfungsamt benötigen wir nur eine Erweiterung des *DefaultUsers*. Die Funktionalität des Prüfers bildet man auf die Rolle des Autors ab, somit kann dieser Prüfungen anlegen und verwalten. Das Prüfungsamt hingegen, sollte keine administrative Rolle bekommen, somit ist es erforderlich für das Prüfungsamt eine neue Rolle in das Rollenkonzept einzufügen, die ebenfalls den *DefaultUser* erweitert.

2. Grundsätzliches Design

Am grundsätzlichen Design sollte sich während des Update-Prozesses nicht viel ändern. Eine Erklärung zum verwendeten Design findet man im Dokument (Aufgabenblatt5_Lsg.pdf, Punkt 3).

3. Paket und Klassenstruktur

3.1 Erweiterungskonzept und dessen Realisierung

Das OLAT-PA Modul wurde als eigenständige Erweiterung zum OLAT-LMS entwickelt. OLAT bietet für solche Erweiterungen (Extensions) verschiedene Ansatzpunkte um eigenen Entwicklungen an das OLAT-LMS anzubinden. Für die Aktualisierung des PA-Moduls ist zu beachten, dass in der neuen OLAT-Version das Spring-Framework vollständig integriert wurde; dadurch muss beispielsweise die Ordnerstruktur um einen Punkt _spring erweitert werden.

Zusätzlich soll auch der Pfad des PA Moduls geändert werden:

Alte Ordnerstruktur	Neue Ordnerstruktur
*de.xman.**	*de.unileipzig.xman.**
**.controller*	**._spring*
**._content*	**.controller*
**._i18n*	**._content*
**._static*	**._i18n*
**.PacketExtension.java*	**._static*
	**.PacketExtension.java*

Pakete, die lediglich von anderen Paketen im PA Modul aufgerufen werden, benötigen diese Orderstruktur jedoch nicht.
Der neu hinzugekommen Ordner *_spring* muss eine *PaketName**Context.xml*** Datei enthalten. Diese Datei registriert einen Erweiterungspunkt am System.

Hier ist zu beachten, dass die Grundstruktur der XML-Datei immer wie folgt aussehen muss.

```
<?xml version="1.0" encoding="UTF-8"?>
    <beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:context="http://www.springframework.org/schema/context"
        xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.0.xsd">
    <bean id="packageID" class="packagePath">
    </bean>
</beans>
```

Da diese Struktur im alten OLAT noch nicht vorhanden war, hat sich unser Team auf folgende Konvention geeinigt:

```
<?xml version="1.0" encoding="UTF-8"?>
    <beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:context="http://www.springframework.org/schema/context"
        xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.0.xsd">
    <bean id="xman.packegName" class="de.unileipzig.xman.packageName">
    </bean>
</beans>
```

3.2 Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem

Die grundsätzliche Struktur des Gesamtsystems hat sich in einigen kleinen Bereichen verändert. Im Grunde nutzt das PA Modul den *org.olat.core* um die Funktionalität zu erreichen. Da der Core jetzt wieder fest zum OLAT gehört (OLAT 6.1 Core eigenständiges Projekt, nur als core.jar vorhanden) sind einige Änderungen aufgetreten. Beispielsweise haben sich einige Bezeichnungen für Klassen geändert oder Pakete wurden verschoben.

Die wichtigsten Ansatzpunkte sind hier dargestellt:

org.olat.core.gui. //allgemein*
org.olat.core.gui.control. //speziell*
*org.olat.core.util.**
*org.olat.core.id.**
*org.olat.core.commons.persistence.**

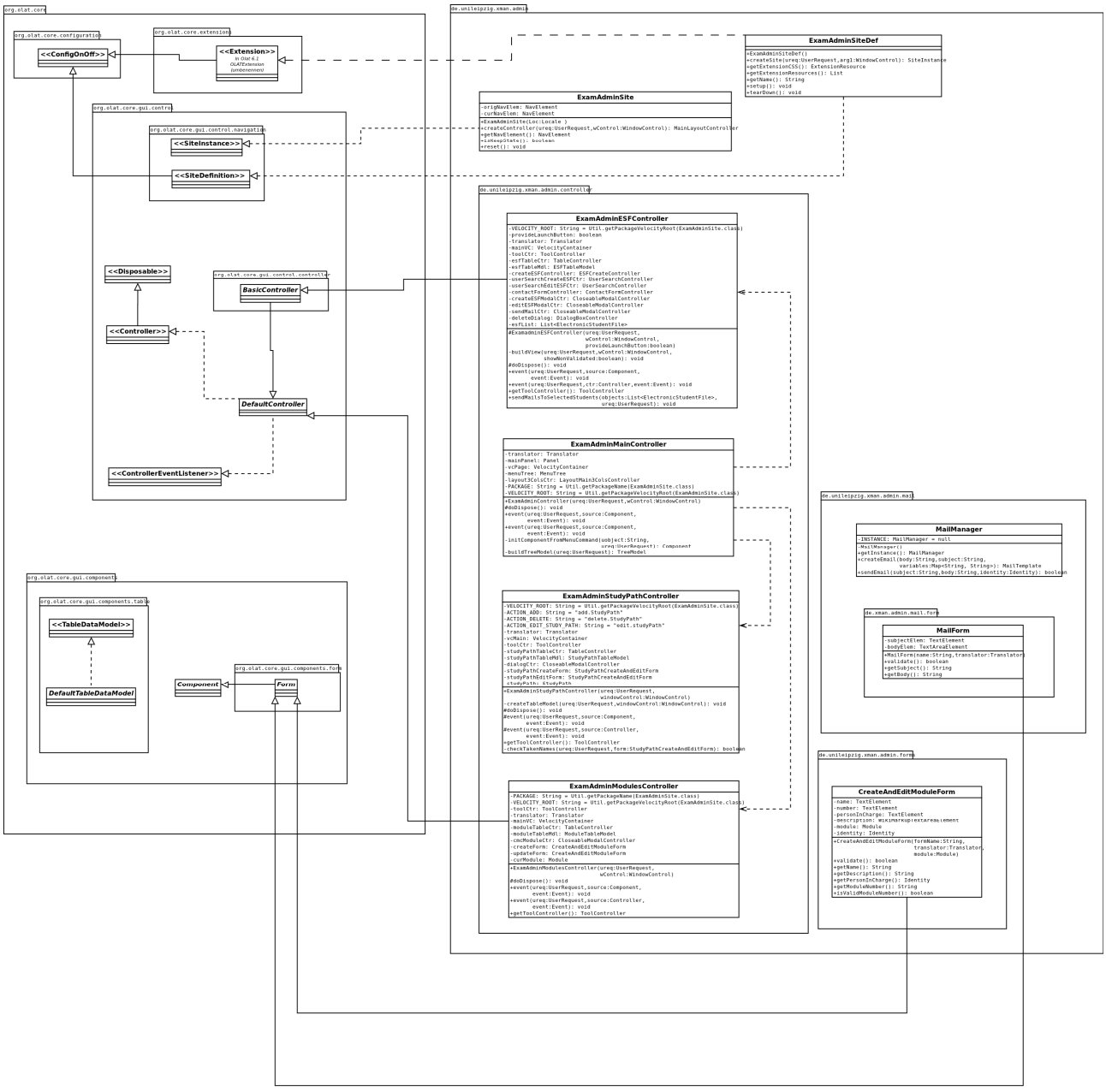
3.3 Grundsätzliche Struktur- und Entwurfsprinzipien einzelner Pakete

Da die Struktur der Pakete im einzelnen erhalten bleiben soll, wird an dieser Stelle nur auf Änderungen eingegangen, die sich im Bezug auf das Gesamtsystem ergeben haben.

de.unileipzig.xman.admin:

Als erstes betrachtet man das Paket *de.unileipzig.xman.admin*, dabei sind folge Änderungen auf dem ersten Blick ersichtlich:
In der Klasse *de.unileipzig.xman.admin.controller.ExamAdminMainController* wird der *LayoutBasicController* verwendetet, dieser wurde jedoch umbenannt und heißt jetzt *MainLayoutController.* Weiter wird im gesamten Paket mehrfach der *TableController* verwendet, wobei dieser keinen Controller als Übergabe mehr benötigt

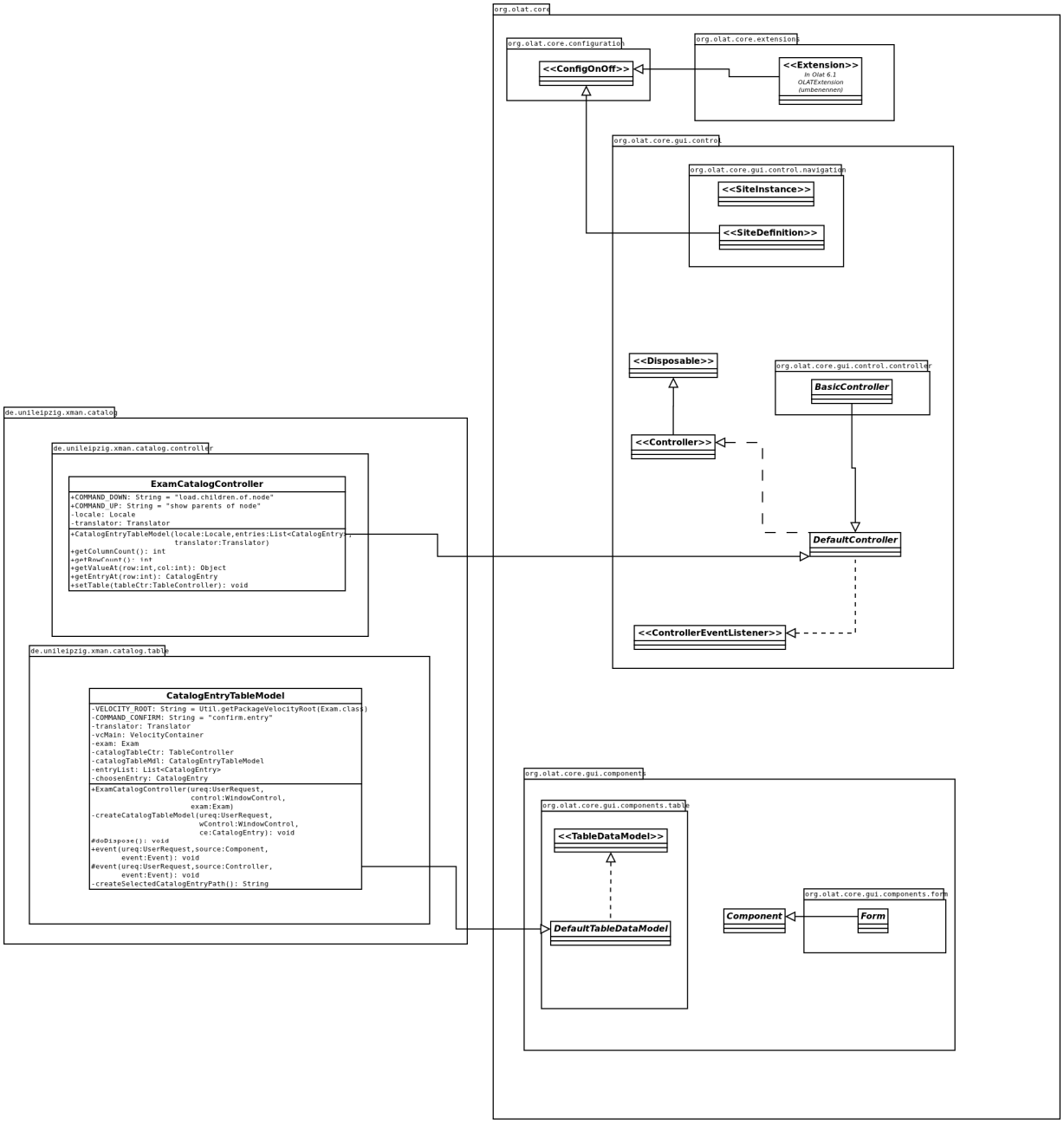
```
new TableController(esfTableConfig, ureq, wControl, translator /* ,this */);
```



de.unileipzig.xman.catalog:

Beim *ExamCatalogControlle* fällt als erstes auf, dass die Klasse *org.olat.basesecurity.ManagerFactory* nicht mehr existiert, deren Funktionalität wird jedoch von dem *BaseSecurityManager* übernommen. Der *TableController* ist auch hier zu finden (siehe *de.unileipzig.xman.admin*). Aufgrund der strickten Einhaltung des Singleton-Prinzips muss bei einigen Methodenaufrufen der Zusatz *getInstance()* eingefügt werden.
Zum Beispiel:

CatalogManager.getChildrenOfExceptRepoEntries(entryList.get(i)).size()
CatalogManager.getInstance().getChildrenOfExceptRepoEntries(entryList.get(i)).size()
[de.unileipzig.xman.catalog.controller.ExamCatalogController Z:203]



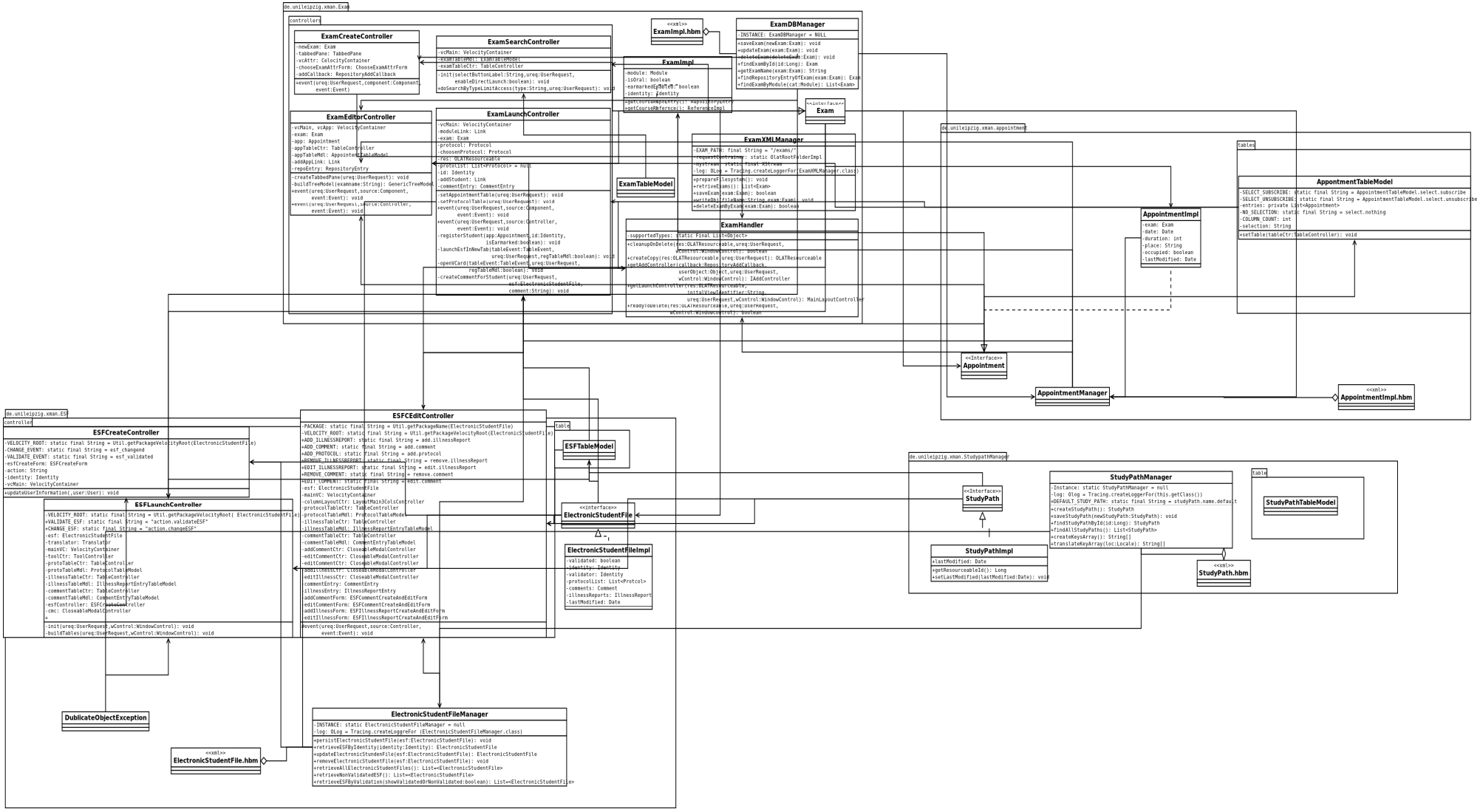
de.unileipzig.xman.portal und de.unileipzig.xman.protocol:

Hier wurden keine besonderen Änderungen festgestellt. Lediglich der *TableController* ist auch hier zu finden (siehe *de.unileipzig.xman.admin*).

Da sich bei der ersten Analyse des Entwurfs und der Struktur lediglich die Controller der einzelnen Pakete ändern (wie in den Beispielen oben) setzten wir nun das Augenmerk in einem Beispiel auf die Zusammenarbeit einzelner Paket des PA Moduls.

Projekt: swp11-1 erstellt am 23.05.2011
Leiter: Martin Walther
Bearbeiter: Sebastian Schüller , Sebastian Dorn, Matthias Zigan, Daniel Gerighausen, Stan Altmann, Johann Berger, Martin Walther

de.unileipzig.xman.appointment, de.unileipzig.xman.exam und de.unileipzig.xman.esf:



In dem UML-Klassendiagramm ist das Zusammenspiel der Pakete *appointment*, *exam* und *esf* dargestellt. Appointment dient zum Einfügen eines Exams in den persönlichen Kalender, hier ist zu prüfen ob es eine bessere Möglichkeit gibt. Die Studentenakte (esf) nutzt selbst viele Pakete innerhalb des PA Moduls, dabei sollten bei Änderungen darauf geachtet werden, dass sich keine Namen ändern.

Datenbankanbindung:

Das PA Modul verwendet einige neue Datenbanktabellen, diese müssen in die aktuelle Datenbank eingefügt werden. Für den Import kann die alte PA Datenbankkonfigurationsdatei verwendet werden. Folgende Tabellen werden in die bestehende Datenbank eingebunden:

o_xman_illnessReport
o_xman_commentEntry
o_xman_illnessReportEntry
o_xman_esf
o_xman_studyPath
o_xman_exam
o_xman_module
o_xman_appointment
o_xman_protocol
o_xman_comment
o_xman_commentEntry
o_xman_illnessReport
o_xman_illnessReportEntry
o_xman_esf type
o_xman_studyPath type

Da die Datenbankverwaltung in der neuen Version des OLAT durch das Spring-Framework organisiert wird, kann es zu Datentyp-Kontroversen kommen.