Universität Leipzig
Softwaretechnikpraktikum 2008

Anwendungs- und Entwicklerhandbuch "RepositoryExtension"

Dynamische Einbindung von Lernressourcen in OLAT

Gruppe SWP08-7
Gruppenleiter: Oliver Perseke

17. Juli 2008

Inhaltsverzeichnis

1 Einleitung

Dieses Dokument soll ein Leitfaden für Entwickler sein, die den Extensionspoint RepositoryExtension bedienen möchten. Es wird auf die notwendigen Änderungen an OLAT eingegangen werden, die den neuen ExtensionPoint hinzufügen, danach wird die Struktur einer RepositoryExtension beschrieben um abschließend Hinweise zur Integration zu liefern.

Bei der RepositoryExtension handelt es sich um eine Erweiterung des Extension-Modells von OLAT, der es ermöglicht neue Lernressourcen zu einem bestehenden OLAT-System hinzuzufügen. Dazu wurde ein weiterer ExtensionPoint erstellt, welcher von der schon vorhandenen ActionExtension abgeleitet wird. Somit funktioniert die Einbindung einer solchen Extension analog zur Vorgehensweise bei einer ActionExtension über die Klasse "Extension".

Gleich zu Beginn soll noch angemerkt welche Formen von Lernressourcen als RepositoryExtension möglich sind. Es wird nur eine Schnittstelle für die Klassen CourseNode (welche vom Bildungsportal Sachsen (BPS) erstellt wurde, nähere Informationen können von dort bezogen werden) und RepositoryEntry angeboten. Diese bieten die Möglichkeit an, die neue Lernressource innerhalb des Tabs "Lernressourcen" zu integrieren und einem Kurs hinzuzufügen. Die Funktionsfähigkeit anderer Interaktionen innerhalb von OLAT können nicht gewährleistet werden und müssten gegebenenfalls selbst untersucht werden. Beispielsweise ist es nicht möglich "Kollaborations-Tools", also Lernressourcen die in Gruppen verwendet werden können, zu erstellen, da diese durch eine anderen Schnittstelle bedient werden, welche nicht durch die RepositoryExtension angeboten wird.

2 Installation

2.1 OLAT Vorbereiten

Vorraussetzungen

Java SDK 1.5.x oder SDK 1.6.x `http://java.sun.com`

Eclipse 3.2 oder neuer `http://www.eclipse.org` (Optional)

Apache Ant `http://ant.apache.org/`

MySQL `http://www.mysql.org/`

Apache Tomcat 5.x oder neuer `http://tomcat.apache.org`

(*vgl. hierzu auch die technische Dokumentation von OLAT* `http://www.olat.org/docu/dev/`)

OLAT beziehen und installieren

Da sich das OLAT-Projekt stetig weiterentwickelt und es täglich neue Versionen gibt, können wir die Kompabilität zu neuen Versionen nicht garantieren. Um das Projekt in die neuste Version from HEAD zu integrieren, müssen Sie die aktuellen Dateien aus dem OLAT-CVS und die Quellcodedateien aus diesem Projekt mergen.

Unsere Version basiert auf dem Build 1631 vom 18.06.2008 08:49:36. Diesen können Sie über den CVS vom OLAT-Projekt (`cvs.olat.org`) beziehen. In der Entwicklungsumgebung Eclipse ist dieses über die CVS-Importfunktion mit Angabe des Datums möglich. Für kommandozeilenbasiertes CVS konsultieren Sie bitte die CVS-Anleitung.

Zur weiteren Installation und Konfiguration von OLAT folgen Sie bitte der OLAT Developer Documentation `http://www.olat.org/docu/dev/` Kapitel 1.

OLAT-Erweiterung einbinden

Laden Sie von unserer Projekt-Homepage `http://pcai042.informatik.uni-leipzig.de/~swp08-7/swp/` den Prototyp V1.1 (unter Downloads → Ergebnisse) herunter. In dem Zip-Archiv finden Sie:

Schnittstelle.zip OLAT-Erweiterung zur dynamischen Einbindung von Lernressourcen

SchwarzesBrett.zip Beispiel-Lernressource "SchwarzesBrett"

Glossary2.zip Beispiel-Lernressource "Glossary2" (Nicht funktional, lediglich demonstrative Zwecke)

Zur Installation der OLAT-Erweiterung entpacken Sie das Archiv Schnittstelle.zip in den Ordner `olat\webapp\WEB-INF\`. Anschließend kompilieren Sie das OLAT-Projekt über die Befehle `ant build` und `ant deploy` neu.

2.2 Einbinden von Lernressourcen-Erweiterungen

Das Einbinden von Lernressourcen-Erweiterungen folgt im wesentlichen der OLAT-vorgegebene Vorgehensweise zur Einbindung von Erweiterungen
`http://www.olat.org/docu/dev/EasilyExtendingOLAT.html` sowie der Struktur von Lernressourcen `http://www.olat.org/docu/dev/resources_repository.html`.

Kopieren der Dateien

Kopieren Sie das JAR-Archiv der RepositoryExtension in den Ordner `olat\webapp\WEB-INF\lib`
Wenn für die Lernressource ein ICON Eingebunden werden soll, so kopieren Sie das PNG in den Ordner `olat\webapp\static\themes\[THEMA]\images\olat`

Konfigurationsdateien anpassen

Damit OLAT die neue Erweiterung kennt, muss diese in die `olat_extensions.xml` im Ordner `olat\webapp\WEB-INF` eingetragen werden:

```
1  <beans>
2
3      <!-- generic OLAT extensions -->
4      <bean id="extmanager" class="org.olat.core.extensions.ExtManager" singleton="
           true">
5          <property name="extensions">
6              <list>
7  [1]             <bean id="schwarzesBrett" class="de.unileipzig.swp08.gruppe7.
       schwarzesbrett.SchwarzesBrettExtension" singleton="true" />
8              </list>
9          </property>
10     </bean>
11     ...
12     <import resource="olat_buildingblocks.xml"/>
13     <bean id="bbfactory" class="org.olat.course.nodes.CourseNodeFactory" singleton=
           "true">
14         <property name="nodeConfigurationList">
15             <list>
16             <!-- building blocks are defined in the imported "buildingblocks.xml"
                   file. -->
17                 <ref bean="st" />
18                 <ref bean="sp" />
19                 ...
20 [2]            <ref bean="schwarzesbrett" />
21             </list>
22         </property>
23     </bean>
24     ...
25 </beans>
```

Listing 1: olat_extensions.xml

Der erste Eintrag [1] dient dazu, die Extension an sich zu definieren. Angegeben wird in dem XML-Element “bean” jeweils als Attribut

'id' Beliebig wählbare, eindeutige Identifikation

'class' Klasse die “Extension” implementiert, z.B. “`de.myname.project.MyProjectExtension`”

'singleton' true (Es gibt nur eine Instanz der Klasse)

Der zweite Eintrag [2] gibt eine Referenz auf die XML-Datei `olat/webapp/WEB-INF/olat_buildingblocks.xml`, in der die CourseNodes definiert werden. Anzugeben ist eine ID als Parameter für das Attribut "bean" in dem Element `<ref/>`. In der `olat_buildingblocks.xml` ist die CourseNode-Configuration der Erweiterung mit der selben ID einzutragen:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/
      dtd/spring-beans.dtd">
3 <beans>
4  ...
5  <bean id="schwarzesbrett" class="de.unileipzig.swp08.gruppe7.schwarzesbrett.nodes.
       SchwarzesBrettCourseNodeConfiguration" singleton="false" />
6  ...
7 </beans>
```

Listing 2: olat_buildingblocks.xml

Anpassung der CSS-Datei

In der CSS-Datei `olat/webapp/static/themes/default/all/olat.css` sind zwei Einträge anzulegen, um dem "Datei-Typ" der neuen Lernressource das kopierte Icon (*vgl. Kopieren der Dateien*) zuzuweisen:

```
1 #b_nav_main li.b_resource_FileResource-SCHWARZESBRETT div, .o_FileResource-
      SCHWARZESBRETT_icon { background-image: url(../images/olat/schwarzesbrett.png)
      }
2 div.b_toolbox li a.o_schwarzesbrett {background-image: url(../images/olat/
      schwarzesbrett.png); }
```

Listing 3: olat.css

Die CSS-Klassennamen, in diesem Beispiel `b_resource_FileResource-SCHWARZESBRETT` sowie `o_schwarzesbrett` wird Ihnen von dem Entwickler der Lernressource mitgeteilt. Wenn Sie selbst Entwickler einer Lernressource sind, so finden Sie in dem Kapitel *Entwicklung von Lernressourcen* Hinweise zur Festlegung. Der Pfad des PNG-Icons entspricht dem Icon, dass Sie bereits kopiert haben (*vgl. Kopieren der Dateien*).

Neustarten des Java-Webapplication Servers

Damit die neu hinzugefügte Lernressource von OLAT geladen wird, starten Sie bitte Ihren Applicationserver entsprechend der Vorgaben des Entwicklers neu.

Beispiel für Tomcat unter Linux:

```
1 cd /usr/share/apache-tomcat-6.0.16/bin/
2 ./shutdown.sh
3 ./startup.sh
```

Listing 4: Restart Apache Tomcat

3 Entwicklung von Lernressourcen

3.1 Allgemeiner Aufbau einer Lernressource

Eine Erweiterung die die RepositoryExtension Schnittstelle bedient muss eine definierte Struktur besitzen, die vorwiegend durch Vorgaben aus OLAT heraus festgelegt ist (vgl. hierzu `http://www.olat.org/docu/dev/ElementsExplained.html` bzw. `http://www.olat.org/docu/dev/ArchitecturalOverview.html`). Eine Darstellung und Erklärung bis ins Detail kann an dieser Stelle nicht gegeben werden, hierfür sollten genannte Dokumente vom OLAT Projekt herangezogen werden.

Im Folgenden wird sich nun dieser Struktur gewidmet werden und so eine Basis gegeben werden, für die eigene Erstellung einer Lernressource. An diese Richtlinien muss sich weitgehend gehalten werden, da anderenfalls ein korrektes Zusammenspiel mit OLAT nicht gewährleistet werden kann.

Die Klassendefinitionen in diesen Abschnitt beinhalten dabei den Platzhalter "xxx", welcher durch einen beliebigen selbstgewählten Namen ersetzt werden kann.

xxxExtension.java

beinhaltet den Konstruktor für die jeweilige Extension und wird vom ExtensionManager geladen

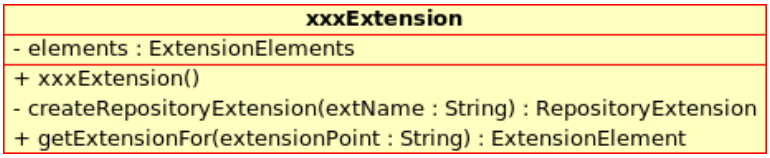

Abbildung 1: xxxExtension.java

xxxRepositoryExtension.java

implementiert die Methoden der RepositoryExtensions-Schnittstelle und stellt insbesondere die notwendige Funktionalität und Informationen für den Tab "Lernressourcen" zur Verfügen. Die Beschreibung dieser Schnittstelle folgt im nächsten Kapitel.

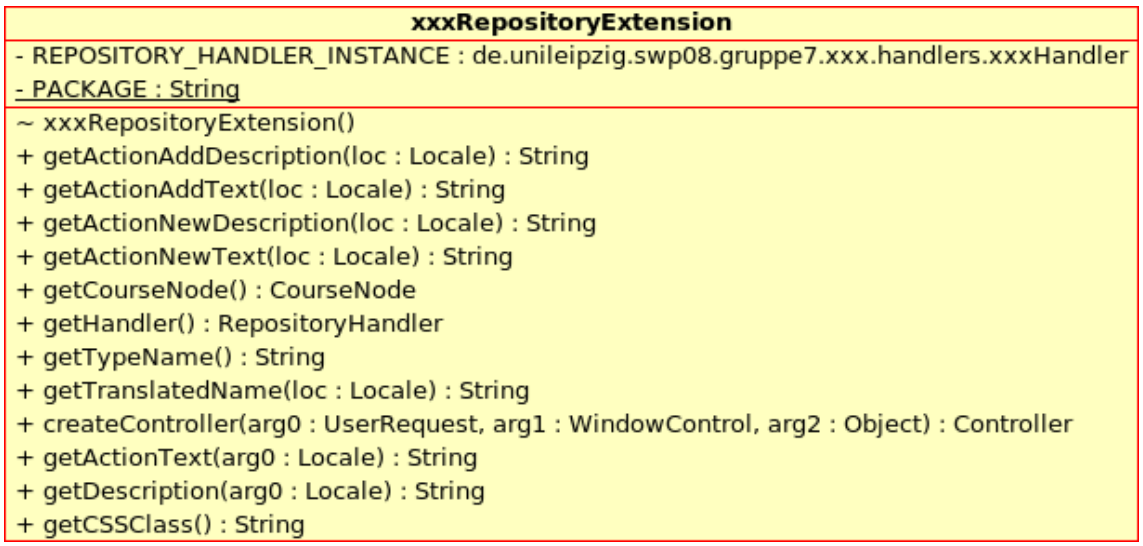

Abbildung 2: xxxRepositoryExtension.java

xxxManager.java

koodiniert die verschiedenen Funktionen wie Erstellen/Löschen/Editieren.

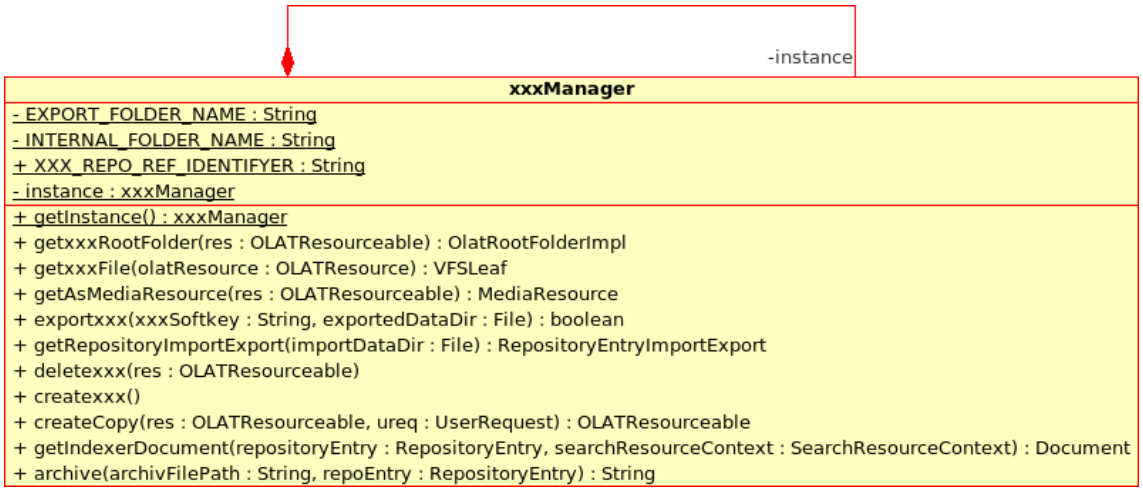

Abbildung 3: xxxManager.java

xxxHandler.java

verwaltet die Aufrufe der unterschiedlichen Funktionalitäten und ruft dann den zugehörigen Controller auf.

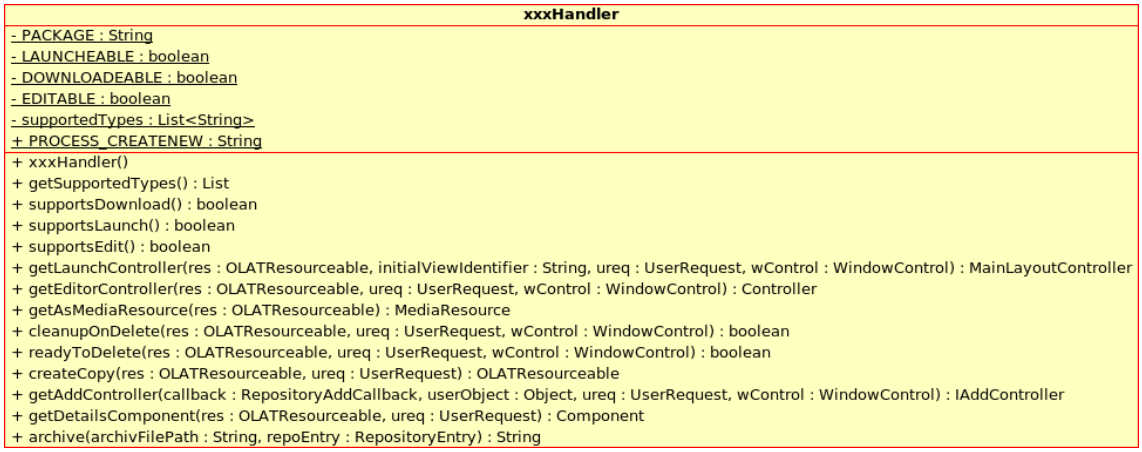

Abbildung 4: xxxHandler.java

xxxFileresource.java

definiert den Namen und den Typ der Lernressource und prüft ob ein entsprechendes Verzeichnis zum Speichern besteht.

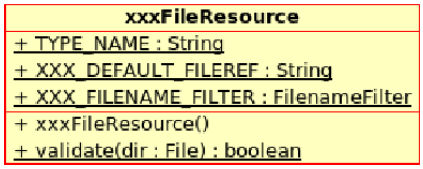

Abbildung 5: xxxFileresource.java

xxxCourseNodeConfiguration.java

ist die Schnittstelle hin zur Integration in einen Kurs, zum Verständniss der genauen Funktionsweise wird eine Recherche in den Materialien vom BPS empfohlen.

xxxCourseNode.java

ist verantwortlich für das Laden der Controller um die Funktionalität für die Lernressourcen-Instanz im Zusammenspiel mit einem Kurs bereitzustellen.

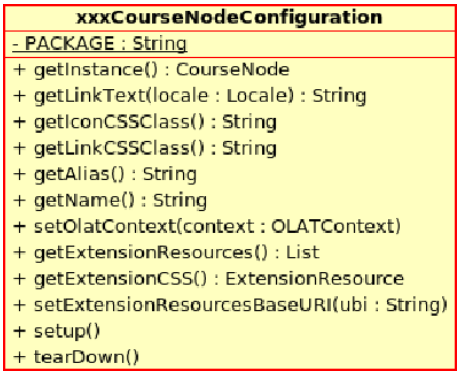

Abbildung 6: xxxCourseNodeConfiguration.java

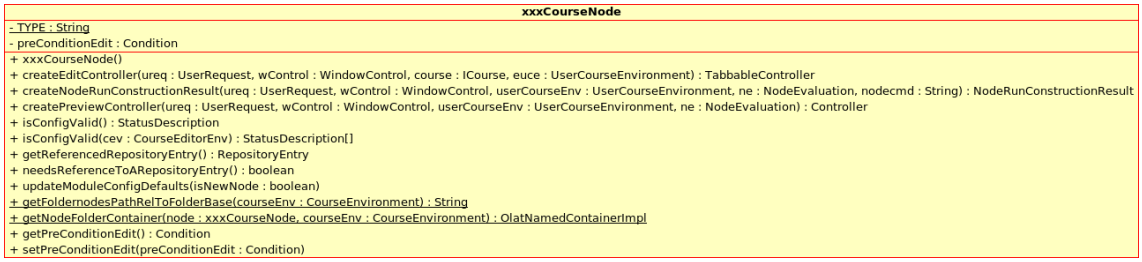

Abbildung 7: xxxCourseNode.java

xxxEditController.java

gibt die Möglichkeit eine Lernressourcen-Instanz innerhalb eines Kurses zu bearbeiten.

xxxRunController.java

Stellt die Funktionalität einer Lernressourcen-Instanz innerhalb des Kurses zur Verfügung.

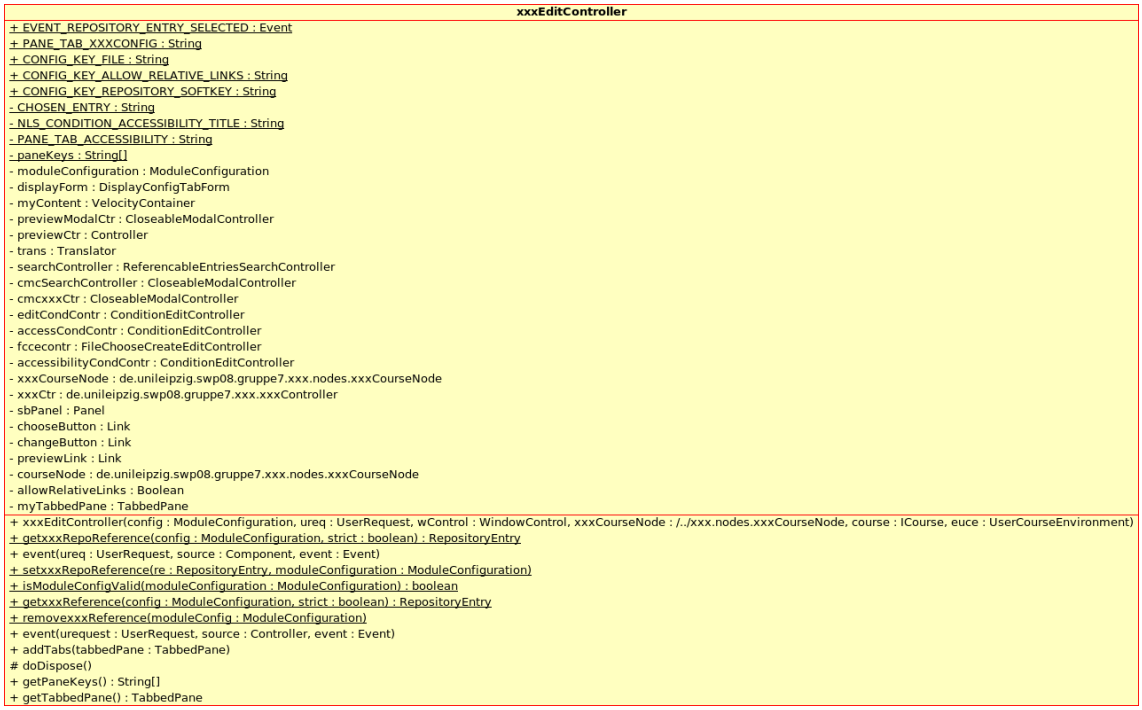

Abbildung 8: xxxEditController.java

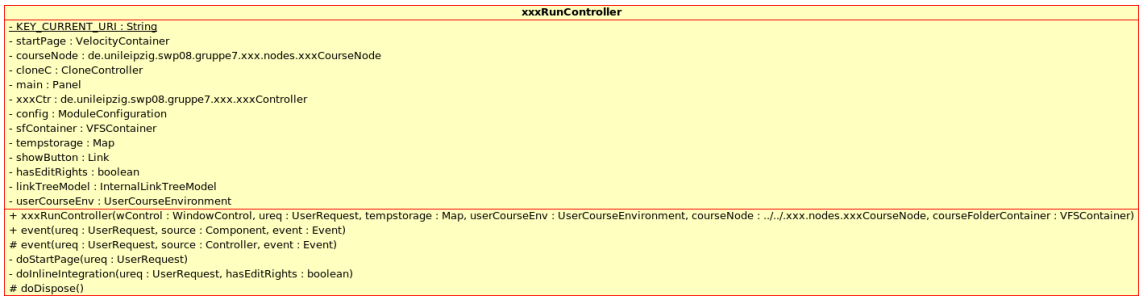

Abbildung 9: xxxRunController.java

3.2 Spezieller Aufbau der Lernressource Schwarzes Brett

Bei der Entwicklung des Lernressource SchwarzesBrett wurden die oben beschriebenen Klassen verwendet. Desweiteren wurden folgenden Klassen erstellt um die Funktionen des Schwarzenbrett bereitzustellen

Die folgenden 3 Klassen stellen anlegen/entfernen/ändern von Themen des Schwarzen Brettes zu Verfügung.Es wird weiterhin der gespeicherte Text aus der Datei ausgelesen und für die Anzeige verarbeitet.
Für jedes angelegte Schwarzes Brett wird eine xml-Datei erstellt, inwelcher Thema, Beschreibung, Erstelldatum und Autor gespeichert wird.

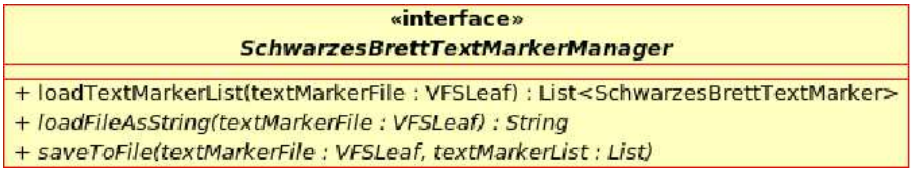

Abbildung 10: SchwarzesBrettTextMarkerManager.java

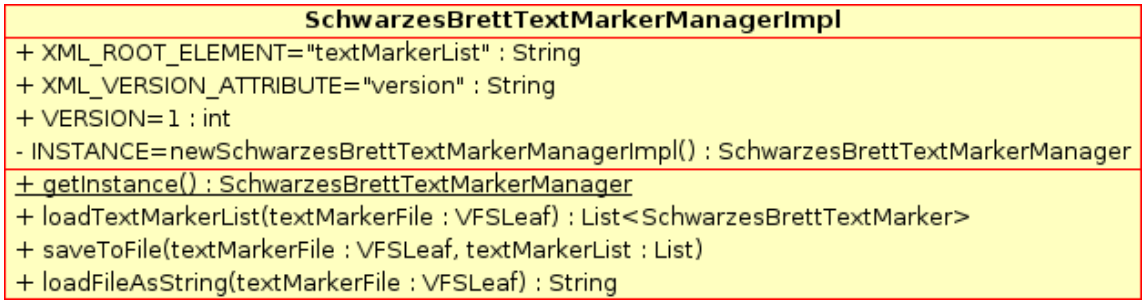

Abbildung 11: SchwarzesBrettTextMarkerManagerImpl.java

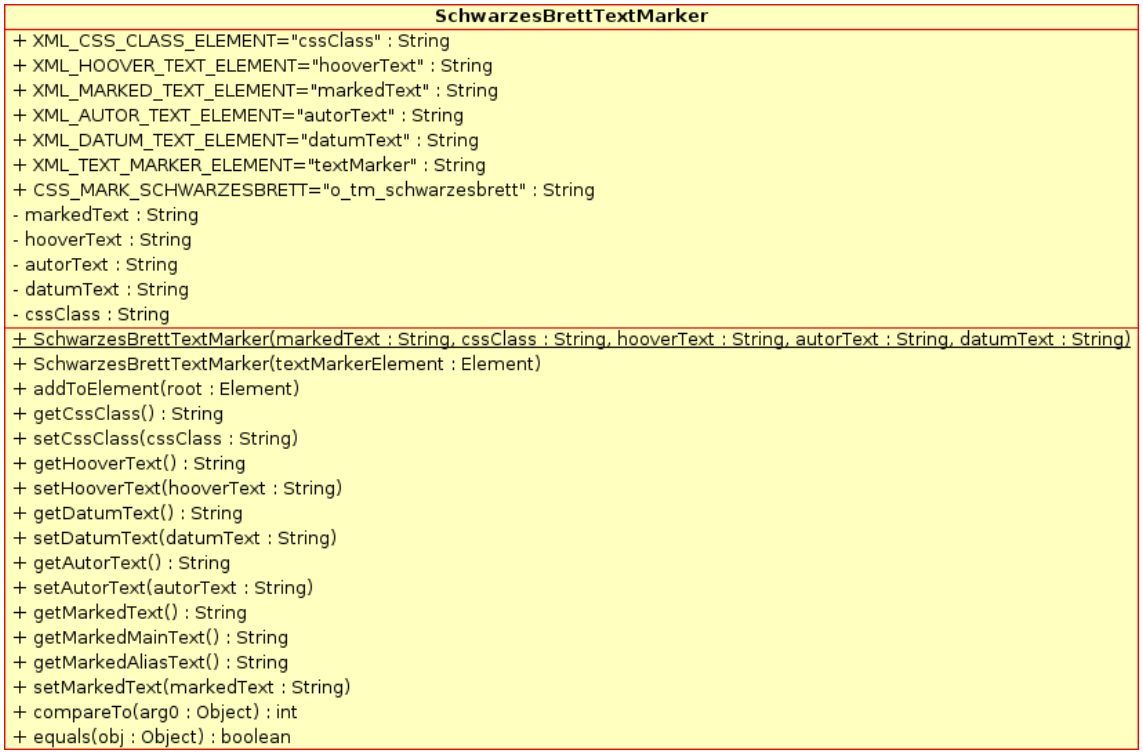

Abbildung 12: SchwarzesBretTextMarker.java

CreateNewSchwarzesBrettController.java

Diese Klasse dient dem Erstellen einer Instanz des SchwarzesBrett

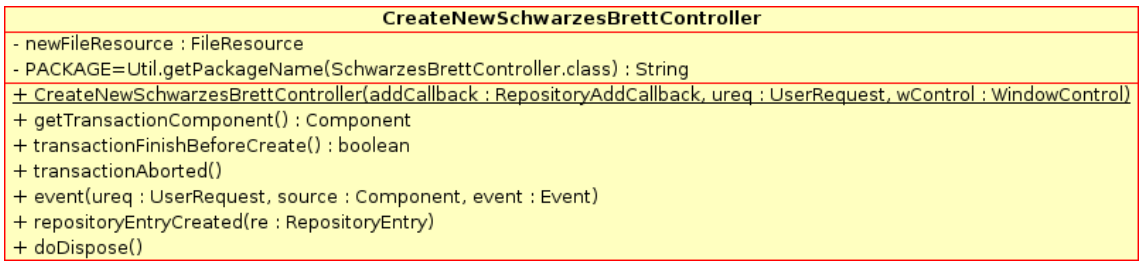

Abbildung 13: CreateNewSchwarzesBrettController.java

SchwarzesBrettController.java

Beinhaltet alle Funktionalitäten um mit einer SchwarzenBrett-Instanz zu arbeiten.

Abbildung 14: SchwarzesBrettController.java

SchwarzesBrettDisplayController.java

Beschreibung siehe Olat Dokumentation.

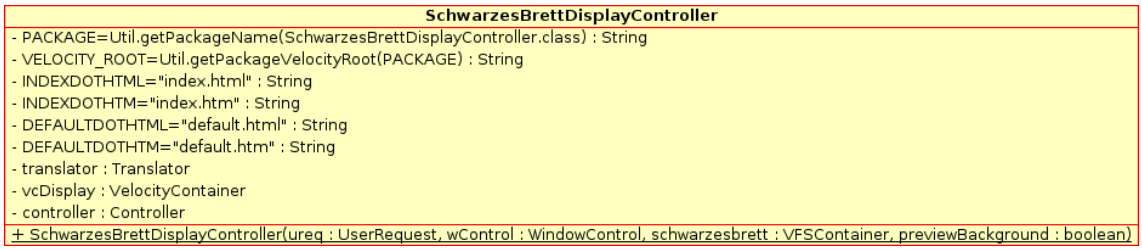

Abbildung 15: SchwarzesBrettDisplayController.java

SchwarzesBrettForm.java

Generiert die Oberfläche um ein neues Thema zu einem Schwarzen Brett hinzufügen zukönnen.

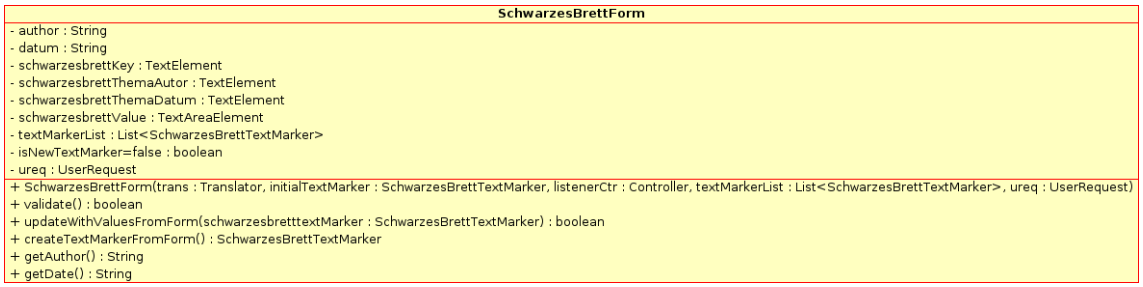

Abbildung 16: SchwarzesBrettForm.java

3.3 Übersicht aller Klassen des Schwarzen Brettes

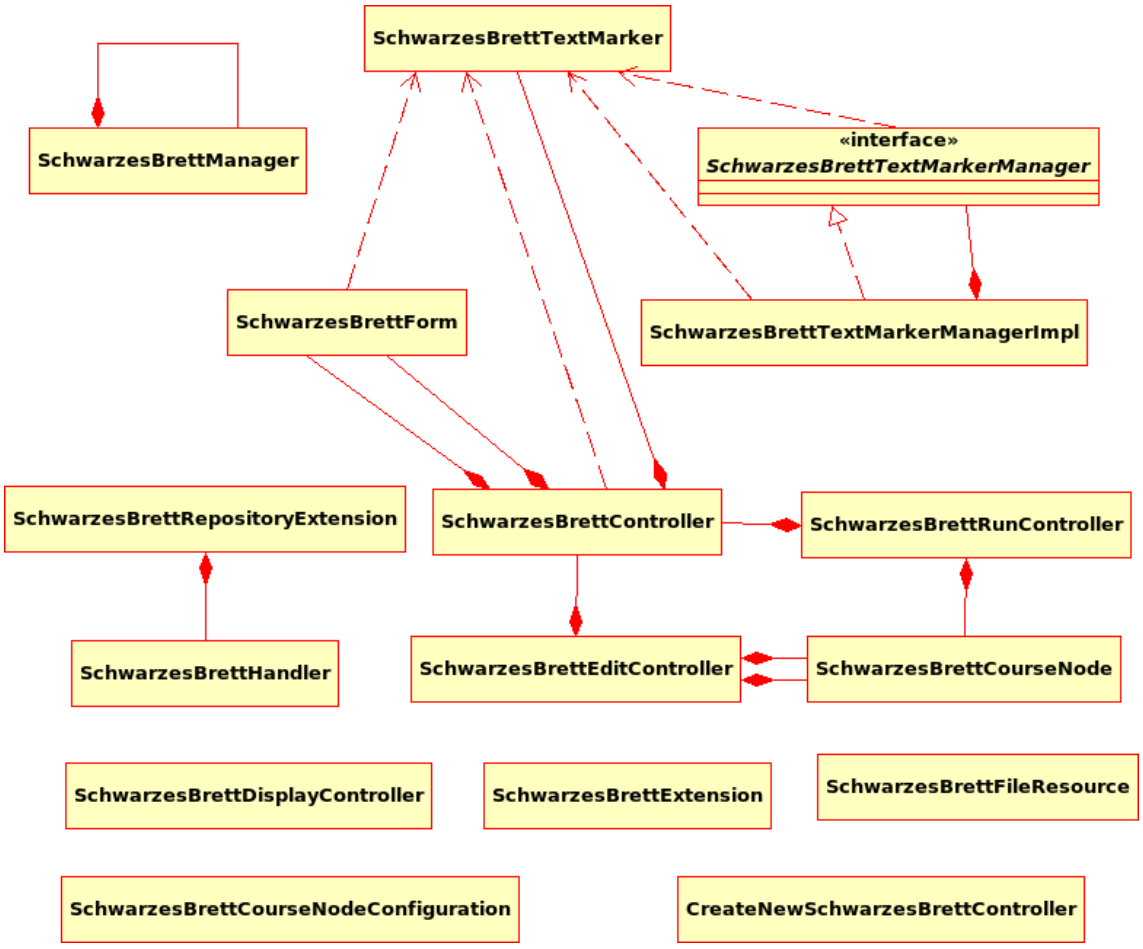

Abbildung 17: Klassenübersicht

4 Schnittstellenbeschreibung RepositoryExtension

In diesem Abschnitt wird das Interface RepositoryExtension beschrieben und somit jene Methoden, die von der Lernressource zur Verfügung gestellt werden müssen um die Integration in den Tab "Lernressourcen" umzusetzen.

public String getTypeName()

gibt den Namen der Lernressource zurück, wie sie innerhalb von OLAT benannt werden soll. Dieser wird dann für die Bearbeitung der Aktionen im Tab "Lernressourcen" verwendet.

public String getActionAddText(Locale loc)

gibt den Namen des Menüeintrags für den "Importieren" Teil des "Tool-Baums" zurück. Dieser sollte anhand des übergebenen Parameters übersetzt werden.

public String getActionAddDescription()

gibt einen Identifier-String zur Auswertung des Events des Menüeintrags im "Importieren" Teil des "Tool-Baums" zurück.

public String getActionNewText(Locale loc)

gibt den Namen des Menüeintrags für den "Erstellen" Teil des "Tool-Baums" zurück. Dieser sollte anhand des übergebenen Parameters übersetzt werden.

public String getActionNewDescription()

gibt einen Identifier-String zur Auswertung des Events des Menüeintrags im "Erstellen" Teil des "Tool-Baums" zurück.

public RepositoryHandler getHandler()

gibt die Instanz der Handler-Klasse zurück, welche sich - wie im vorherigen Kapitel erwähnt - um die Verteilung der verschiedenen Aktionen des Tabs "Lernressourcen" auf die verschiedenen Controller kümmert. Für eine detailierte Beschreibung sollten die relevanten OLAT-Dokumente bzw. der Quellcode der Demo-Extension "Schwarzes Brett" herangezogen werden.

public String getTranslatedName(Locale loc)

gibt den Namen der Lernressource, wie er dem Nutzen angezeigt wird zurück. Dieser sollte anhand des übergebenen Parameters übersetzt werden.

public String getCSSClass()

gibt den Namen für das Icon der Lernressource zurück, welcher, wie im Abschnitt 2.2 beschrieben, identischer mit dem relevanten Eintrag in der CSS-Datei sein muss.