Entwurfsbeschreibung V1

Gruppe SWP08-7

27. Mai 2008

Inhaltsverzeichnis

1 Allgemeines

RepositoryExtensions sind eine Erweiterung des Extension-Modells von OLAT, sie sollen es ermöglichen die vorhandenen Lernressourcen funktional aus dem Gesamtsystem auszuglieden und auch neue Lernressourcen zu OLAT hinzuzufügen.

2 Produktübersicht

Bei den RepositoryExtensions handelt es sich um eine Erweiterung des Extension-Systems von OLAT. Bislang gibt es die Möglichkeiten über jar-Archive, die eine definierte Klassenstruktur beinhalten, neue Inhalte zum Home hinzuzufügen, oder auch einen eigenen Tab zu erstellen. Dieses System wird nun dahingehend erweitert, dass es auch möglich ist eigene Lernressourcen in OLAT zu integrieren. Da die Lernressourcen innerhalb des Gesamtsystems sehr verzahnt sind, ist die notwendige Definition einer solchen Basis-Klasse ungleich komplex. So müssen 3 verschiedene Funktionalitäten abgedeckt werden:

- eine Klasse die im Zusammenhang der Lernressourcen fungiert,
- eine Klasse die die Interaktion zwischen einem Kurs und einer Lernressourcen-Instanz erledigt, sowie
- die eigentliche Funktionalität der jeweiligen Lernressource.

Zusätzlich wird als "proof of concept" eine derartige Lernressource erstellt, die in Anlehnung an ein "Schwarzes Brett", es Nutzern eines Kurses ermöglicht kurze Nachrichten für alle anderen Nutzer sichtbar zu hinterlassen.

3 Grundsätzliche Entwurfsprinzipien für das Gesamtsystem

3.1 Übersicht

Die RepositoryExtension wird eine Schnittstelle anbieten, mit welcher die oben genannten Funktionalitäten einer Lernressourcen-Erweiterung transparent für das OLAT-System zugänglich werden.

3.2 Architekturkonzept

Da es sich um eine Modifikation eines bestehenden Systems handelt, werden die schon vorhandenen Konzepte konsequent umgesetzt. Es wird, wie auch schon in OLAT umgesetzt, das MVC-Konzept verfolgt, welches die Funktionalität in Modelle, Views und Controller teilt.

4 Grundsaetzliche Struktur- und Entwurfsprinzipien der einzelnen Paktete

Wie schon in 2. beschrieben, wird die Funktionalität einer Lernressource in 3 Teile aufgeteilt. Diese verschiedenen Teile, sowie die schon vorhandenen Schnittstellen zu dieser sollen

nun im folgenden dargestellt werden. Insbesondere wird auf notwendige Änderungen eingegangen werden, die an bestehenden Code vorgenommen werden. Prinzipiell kann man zusammenfassen, dass die Änderungen vorwiegend derart sind, dass schon vorhandene Abstraktionen konsequent genutzt werden um bisher statische Zugriffe auf fest definierte Lernressourcen durch eine Form von "Wrapper" auf beliebig viele solche Klassen abzubilden.

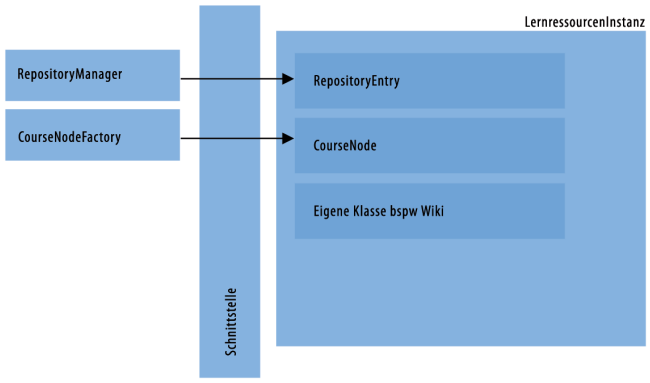

Abbildung 1: Klassen mit Bezug zu Lernressourcen

org.olat.repository.*

DetailsForm.java

Stellt einen Dialog dar in dem Eigenschaften an einer Lernressourcen-Instanz verändert werden können. Dazu werden Get/Set-Methoden der RepositoryEntry-Klasse verwendet. An dieser Klasse ist vorerst nichts zu ändern.

DisplayCourseInfoForm.java

Zeigt Eigenschaften eines bestimmten Kurses an. An dieser Klasse ist nichts zu ändern.

DisplayInfoForm.java

Zeigt Eigenschaften einer Lernressourcen-Instanz an, dazu werden Get-Methoden der RepositoryEntry-Klasse verwendet. An dieser Klasse ist nichts zu ändern.

EditDescForm.java

Stellt einen Dialog dar in dem der Beschreibungstext einer Lernressourcen-Instanz verändert werden kann. Dazu werden Get/Set-Methoden der RepositoryEntry-Klasse verwendet. An dieser Klasse ist nichts zu ändern.

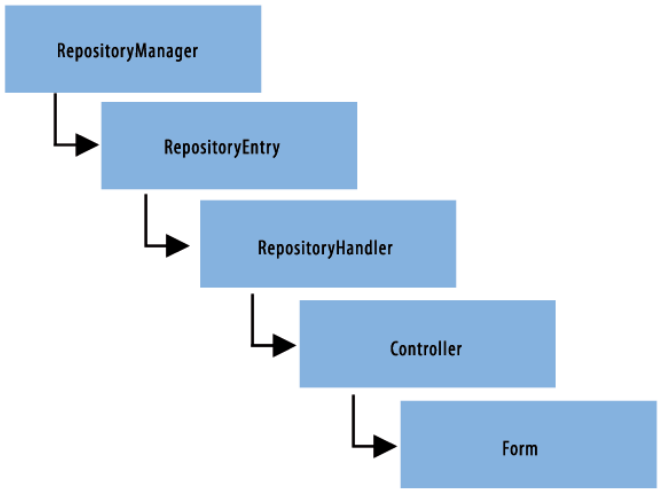

Abbildung 2: Zugriff auf Lernressourcen-Instanzen-Funktionalitaet

MetaDataElement.java

Stub.

PropPupForm.java

Stellt einen Dialog dar in dem Eigenschaften an einer Lernressourcen-Instanz verändert werden können. Dazu werden Get/Set-Routinen der RepositoryEntry-Klasse verwendet. An dieser Klasse ist vorerst nichts zu ändern.

RepoJumpInHandlerFactory.java

Klasse für JumpIn, was auch immer das ist. An dieser Klasse ist vorerst nichts zu ändern.

RepositoryEntry.java

Basis-Klasse für einen Lernressourcen-Eintrag erbt von OLATResourceable und besteht vor allem aus Get/Set-Methoden für die Eigenschaften. An dieser Klasse ist vorerst nichts zu ändern.

RepositoryEntryImportRenderer.java

Diese Klasse gibt einen CSS-Icon für eine Lernressourcen-Instanz zurück, sie ermittelt das Icon sowie den "Hover-Text" aus der RepositoryEntry-Klasse. An dieser Klasse ist vorerst nichts zu ändern.

RepositoryEntryImportExport.java

Importiert/exportiert Eigenschaften und Daten einer Lernressourcen-Instanz. Dazu werden Methoden aus dem zugehörigen Handler aufgerufen. An dieser Klasse ist vorerst nichts zu ändern.

RepositoryEntryTypeColumnDescriptor.java

uninteressant.

RepositoryManager.java

Abbildung 3: Klassendiagramm RepositoryManager

Diese Klasse verwaltet die Lernressourcen und Lernressourcen-Instanzen. Also Erstellen/Laden/Speichern/Aktualisieren/Löschen von Lernressourcen-Instanzen, Suchen von Lernressourcen-Instanzen, Verwaltung spezieller Eigenschaften (wie Download-Counter). An dieser Klasse ist vorerst nichts zu ändern.

RepositoryTableModel.java

Klasse die das Datenmodell der Lernressourcen-Instanzen-Liste definiert. An dieser Klasse ist vorerst nichts zu ändern.

RepositoryUIFactory.java

Diese Klasse übernimmt die Einbettung der Darstellung einer Lernressourcen-Instanz in OLAT. An dieser Klasse ist vorerst nichts zu ändern.

SearchForm.java

Stellt einen Dialog dar, der eine Maske für die Suche nach Lernressourcen-Instanzen erlaubt. An dieser Klasse ist vorerst nichts zu ändern.

SharedFolderSecurityCallback.java

uninteressant

org.olat.repository.controllers.*

AddFileResourceController.java

Diese Klasse erlaubt es Dateien als Lernressourcen-Instanz zu OLAT hinzuzufügen. An dieser Klasse ist vorerst nichts zu ändern.

EntryChangedEvent.java

Klasse die angibt ob sich eine bestimmte Lernressourcen-Instanz verändert hat. An dieser Klasse ist vorerst nichts zu ändern.

IAddController.java

Interface für eine Lernressourcen-Instanz die beim Erstellen einer solchen Instanz verwendet wird. Beinhaltet Methoden die eine Lernressourcen-Klasse beinhalten muss. An dieser Klasse ist vorerst nichts zu ändern.

ReferencableEntriesSearchController.java

Diese Klasse stellt eine weitere Suchfunktion für Lernressourcen-Instanzen bereit, mit Fokus auf die für einen bestimmten Benutzer editierbaren Instanzen. An dieser Klasse ist vorerst nichts zu ändern.

RepositoryAddCallback.java

Ober-Klasse für das Erstellen einer neuen Lernressourcen-Instanz? An dieser Klasse ist vorerst nichts zu ändern.

RepositoryAddController.java

Diese Klasse erlaubt es einem Kurs (?) eine bestimmte Lernressourcen-Instanz hinzuzufügen. Bislang werden die vorhandenen Ressourcen statisch eingebunden, und bei dem jeweiligen Event die zugehörigen Methoden aus der Handler-Klasse aufgerufen. Sie wird um weitere dynamische Einträge für Lernressourcen erweitert werden. Wird ein solcher Eintrag vom Nutzer ausgewählt muss die richtige Handler-Klasse gewählt und mit dieser der Workflow fortgesetzt werden.

RepositoryCopyController.java

Diese Klasse behandelt das Kopieren von Lernressourcen-Instanzen (von einer vorhandene Instanz wird eine neue Kopie angelegt), sie verwendet Methoden aus RepositoryEntry. An dieser Klasse ist vorerst nichts zu ändern.

RepositoryDetailsController.java

Diese Klasse erlaubt es für eine Lernressourcen-Instanz Einstellungen vorzunehmen, sowie Aktionen (bspw. Download) mit ihr durchzuführen. Die meisten verwendeten Methoden stammen aus der Klasse RepositoryEntry, wenige sind spezifisch aus dem Handler für die Kurse. Die Methoden für die Aktionen werden auch von der Klasse RepositoryMainController verwendet. An dieser Klasse ist vorerst nichts zu ändern.

RepositoryEditDescriptionController.java

Diese Klasse erlaubt es den Beschreibungstext einer Lernressourcen-Instanz zu ändern. Es wird die Klasse EditDescForm verwendet. Die Methoden zum Zugriff auf die Lernressourcen-Instanz stammen aus RepositoryEntry. An dieser Klasse ist vorerst nichts zu ändern.

RepositoryEditPropertiesController.java

Diese Klasse erlaubt es für eine Lernressourcen-Instanz Einstellungen vorzunehmen. Es wird die Klasse PropPupForm verwendet. Die meisten verwendeten Methoden stammen aus der Klasse RepositoryEntry, wenige sind spezifisch aus dem Handler für die Kurse. Die Methoden für die Aktionen werden auch von der Klasse RepositoryMainController verwendet. An dieser Klasse ist vorerst nichts zu ändern.

RepositoryEntryImageController.java

Diese Klasse erlaubt es einer Lernressourcen-Instanz ein Bild zuzufügen. Die verwendeten Methoden stammen aus der Klasse RepositoryEntry. An dieser Klasse ist vorerst nichts zu ändern.

RepositoryMainController.java

Diese Klasse beinhaltet die Funktionalität für das “Hauptfenster” des Tabs Lernressourcen. Es werden die Menüs auf der linken und rechten Seite erstellt und damit Menüpunkte für das Erstellen/Löschen/Suchen/Kopieren etc. von Lernressourcen-Instanzen verwal-

tet. Bisher sind die vorhandenen Lernressourcen statisch eingebunden, das heißt das die Menüpunkte statisch eingetragen werden und die Methoden des zugehörigen Handlers bei einem Event aufgerufen werden. Das Menü wird um weiter dynamisch hinzugefügte Lernressourcen-Arten erweitert, wird einer dieser Einträge angewählt muss der spezifische Handler ausgewählt werden um den Workflow korrekt weiterzuführen.

RepositorySearchController.java

Diese Klasse beinhaltet die Funktionalität für die Suche nach bestimmten Lernressourcen-Instanzen. An dieser Klasse ist vorerst nichts zu ändern.

org.olat.repository.delete.*

Diese Klassen werden beim Löschen von Lernressourcen-Instanzen verwendet. Vorerst werden diese nicht näher behandelt, da es sich beim Löschen um eine Kann-Funktion handelt. Zum gegebenen Zeitpunkt wird diese Sektion erweitert werden.

- ReadyToDeleteController.java
- RepositoryEntryDeleteTableModel.java
- SelectionController.java
- StatusController.java
- TabbedPaneController.java

org.olat.repository.handlers.*

RepositoryHandlersFactory.java

Diese Klasse verwaltet die verschiedenen Handler der Lernressourcen. Bislang werden diese statisch eingetragen. Sie wird dahingehend verändert, dass dynamisch neue Handler hinzugefügt werden können.

RepositoryHandler.java

Diese Klasse definitiert die Basis-Klasse einer Lernressource und beinhaltet Funktionen die in den verschiedenen Workflows in der Lernressource aufgerufen werden. Eine Lernressourcen-Erweiterung muss folglich all diese Funktionen implementieren.

org.olat.repository.site.*

Diese Klassen beinhalten die Funktionalität für die Schnittstelle eines Tabs für den Tab Lernressourcen.

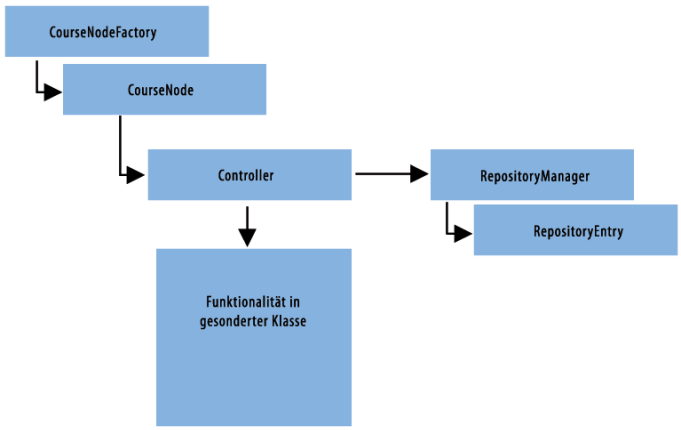

Abbildung 4: Verknuepfung Kurs / Lernressourcen Instanz

org.olat.course.nodes.*

CourseNode.java

Diese Klasse stellt die Basisklasse für eine Lernressource dar, wie sie von einem Kurs aus angesteuert wird.

SchwarzesBrettNode.java

Diese Klasse beinhaltet die Methoden für die Ansteuerung eines "Schwarzen Bretts" aus einem Kurs heraus.

org.olat.course.nodes.sb.*

Hier befinden sich die Controller die von org.olat.course.nodes.SchwarzesBrettNode.java verwendet werden. Unter anderem ein Run-Controller und ein Edit-Controller.

org.olat.core.extensions.*

ExtensionsManager.java

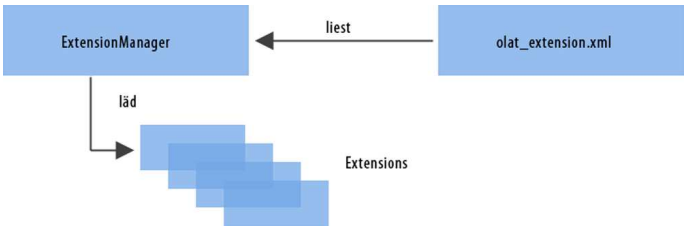

Abbildung 5: Laden der Erweiterung

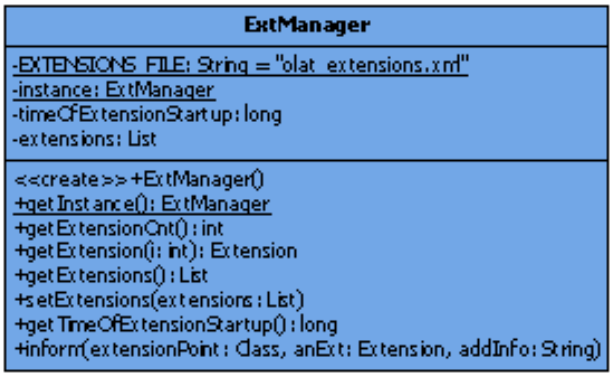

Abbildung 6: Klassendiagramm Extensionmanager

Verwaltet die verschiedenen Erweiterungen die in der "olat_extensions.xml" aufgelistet sind. Diese werden von dieser Klasse geladen und sind auch über diese referenzierbar.

org.olat.core.extensions.repository.*

RepositoryExtension.java

Diese Klasse beinhaltet die zu schaffende Basisklasse für Lernressourcen-Erweiterungen. Sie beinhaltet Funktionen, die es den verschiedenen Stellen innerhalb von OLAT (CourseNodes, RepositoryHandler) auf spezifische Lernressourcen zuzugreifen.