

# Anhang zum Pflichtenheft: Glossar

Gruppe SWP08-7

20. Mai 2008

**Erweiterung** Durch Erweiterungen werden in der vorliegenden Rahmenapplikation OLAT neue Funktionalitäten hinzugefügt. Dabei ist es nicht vonnöten, den bereits bestehenden Quellcode zu verändern, sondern die von den OLAT-Entwicklern zur Verfügung gestellten Schnittstellen, die sogenannten *ExtensionPoints* oder Erweiterungspunkte zu adaptieren. Dies erleichtert die Installation von Updates und schützt getätigte Investitionen. Beispiele für solche Erweiterungen wären ein zusätzlicher Navigationspunkt in der Hauptnavigation oder ein neuer Kursbaustein.

**Hauptfenster** Bezeichnet den inhaltlichen Teil des OLAT-Systems aus View-Sicht. In den meisten Fällen ist das “Main Panel” dreifach untergliedert in *Hauptmenü*, *Panel* und *Toolbaum*.

**Lernressource** In OLAT werden mit Lernressourcen einerseits die Ablage (als Funktionsraum in OLAT) und andererseits die darin abgelegte einzelne Lernressource bezeichnet. Diese sind unter anderem: Kurse, Tests, Fragebögen, CP-Lerninhalte, SCORM-Lerninhalte, Ressourcenordner, andere Dateien. Es handelt sich also um den *Tab* “Lernressourcen” und dessen Inhalt.

**Lernressourcen-Arten** sind verschiedene Kategorien von Lernressourcen:

- Kurse (OLAT-eigener Dokumententyp)
- Test (standardisierter Dokumententyp gemäss IMS-QTI)
- Fragebogen (standardisierter Dokumententyp gemäss IMS-QTI)
- CP-Lerninhalt (standardisierter Dokumententyp gemäss IMS-CP)
- SCORM-Lerninhalt (standardisierter Dokumententyp gemäss SCORM)
- andere Dateien (html, txt, doc, xls, ppt, mov, mp3, etc.)

**Lernressourcen-Instanz** ist ein Objekt einer Klasse. Z.B. ein Kurs oder Fragebogen, der einer bestimmten Gruppe zugeordnet wird.

**Schnittstelle / API** Application Programming Interface, ist eine Schnittstelle zur Anwendungsprogrammierung und dient dazu anderen Programmen Funktionalitäten zur Verfügung zu stellen. Damit wird auch eine höhere Wiederverwendbarkeit und Modularität erreicht.

**Systemverwalter** Systemverwalter haben uneingeschränkten Zugriff auf das System, und sind für die Verwaltung, Wartung und den Betrieb des Systems zuständig. OLAT Administratoren haben die Berechtigung alle internen Systeme zu steuern, neue Inhalte einzupflegen, Strukturen zu verändern oder zu erweitern und System-Daten und Statistiken auszulesen.

**Tab** die Bezeichnung für Register- oder Karteireiter, also die hervorstehenden, etikettierten Teile von Karteikarten oder Akten (beinhalten das *Hauptfenster*)

**Toolbaum** Der "allgemeine" Toolbaum stellt Funktionen für den Administrator bereit, um Instanzen von Lernressourcen zu erstellen, importieren und zu löschen. Die Angezeigten Ressourcen können den Toolbaum ihrerseits überschreiben und für eine Art "Unter-navigation" nutzen.

**dynamisch** ist das Gegenteil von "hard-coded". Z.B. eine "dynamische" Einbindung der Tabs (festgelegt in der Konfigurationsdatei "olat\_extensions.xml").

**MVC-Konzept** Das MVC-Paradigma ist ein grundlegendes Architekturkonzept, das Modell, Controller und View (Darstellung) klar voneinander trennt. Dabei können die einzelnen Komponenten nicht beliebig miteinander kommunizieren, sondern gehorchen dem MVC-Modell, so dass die einzelnen Teile möglichst unabhängig voneinander sind. Diese Trennung macht zum einen nahezu paralleles Arbeiten an diesen Teilen möglich, untertützt eine hohe Wiederverwendbarkeit sowie Wartbarkeit und macht es möglich, relativ leicht Änderungen und Erweiterungen vor zu nehmen.

**Model** Das Modell enthält die darzustellenden Daten und gegebenenfalls (abhängig von der Implementation des MVC-Patterns) auch die Geschäftslogik.

**View** Die Präsentations-Schicht ist für die Darstellung der benötigten Daten aus dem Modell und die Entgegennahme von Benutzerinteraktionen zuständig. Sie kennt sowohl ihre Steuerung als auch das Modell, dessen Daten sie präsentiert, ist aber nicht für die Weiterverarbeitung der vom Benutzer übergebenen Daten zuständig. Im Regelfall wird die Präsentation über Änderungen von Daten im Modell unterrichtet und besorgt sich daraufhin die aktualisierten Daten. Die Präsentation entspricht dem Entwurfsmuster "Kompositum".

**Controller** Die Steuerung verwaltet die Präsentation(en), nimmt von ihnen Benutzeraktionen entgegen, wertet diese aus und agiert entsprechend. Zu jeder Präsentation existiert ein Modell. Es ist nicht die Aufgabe der Steuerung, Daten zu manipulieren. Die Steuerung entscheidet aufgrund der Benutzeraktion in der Präsentation, welche Daten im Modell geändert werden müssen. Sie enthält weiterhin Mechanismen, um die Benutzerinteraktionen der Präsentation einzuschränken. Präsentation und Steuerung entsprechen zusammen dem Entwurfsmuster „Strategie“, wobei die Steuerung die Strategie ist.

**Actiontext** Als Actiontext / Linktext bezeichnen wir Text, der mit einem Verweis auf eine Zieladresse verbunden ist.

**Description** Ist der Text, der erscheint, wenn man mit dem Mauszeiger über einen Link fährt (“MouseOver-Effekt”).

**Klasse** Klasse ist in der Objektorientierung ein abstrakter Oberbegriff für die Beschreibung der gemeinsamen Struktur und des gemeinsamen Verhaltens von Objekten (Klassifizierung). Sie dient dazu, Objekte zu abstrahieren. Im Zusammenspiel mit anderen Klassen ermöglichen sie die Modellierung eines abgegrenzten Systems in der objektorientierten Analyse und dem objektorientierten Design. Die Struktur einer Klasse bilden die Attribute (auch Eigenschaften), das Verhalten die Methoden (auch Operation, Funktion, Prozedur) der Klasse.

**Hauptmenü** Das Hauptmenü befindet sich auf der linken Seite des *Hauptfensters* und stellt die Navigationselemente dar. Ein Menüeintrag erlaubt dem Benutzer zu bestimmten Inhalten zu navigieren.

**Menüeintrag** Ist ein Eintrag (“Link” mit *Actiontext* und *Description*) im Hauptmenü.

**ActionExtension** Ist ein Interface in OLAT welche *Actiontext*, *Description* und *Controller* beinhaltet. Sie wird genutzt um dynamisch Links im Hauptmenü des Tabs “Home” zu erzeugen.

**Panel** Als Panel bezeichnen wir den mittleren Teil der 3-teiligen Gliederung (*Hauptmenü*, *Panel* und *Toolbaum*).

**RepositoryMainController** Die Klasse “RepositoryMainController” ist verantwortlich für den Aufbau des Hauptfensters im Tab “Lernressourcen”.

**buildTreeModel** Ist die Methode im RepositoryMainController, die das Hauptmenü generiert.

**statische Inhalte / dynamische Inhalte** Unter “dynamischen Inhalten” verstehen wir Inhalte, die nicht fest im OLAT-Core implementiert sind, sondern über Schnittstellen von Extensions bereitgestellt werden. So besteht z.B. das Hauptmenü des Tabs “Home” aus statischen Inhalten (“Settings”, “Calendar”, ...) und zusätzlich aus dynamischen Einträgen die Extensions einbinden können.

**ClosableModelController** Ist eine Art “PopupController”, der erscheint, wenn man im Toolbaum vom Tab Lernressourcen eine neue Instanz einer Lernressource erstellt, importiert oder löscht.