

Entwurfsbeschreibung des LilaWikis (1. ergänzte und überarbeitete Fassung)

1. Allgemeines

1.1 Charakterisierung

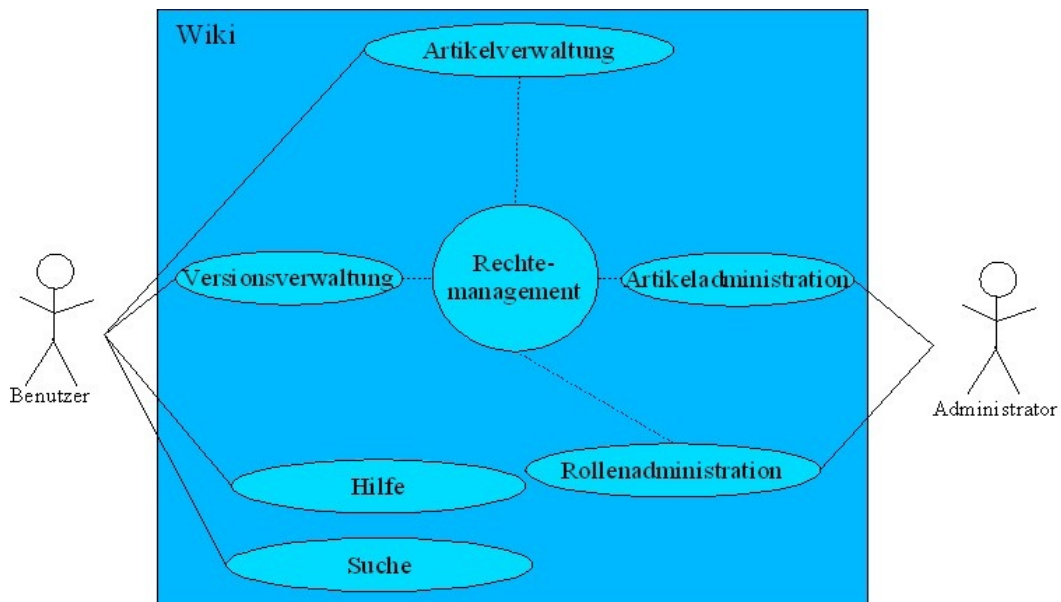
LilaWiki ist eine Portletapplikation, die unter Jetspeed-2-Portalen betrieben wird. Der Schwerpunkt liegt auf der Bereitstellung von Artikeln. Sie sind hierarchisch strukturiert und sollen von den Portal-Benutzern möglichst einfach über einen Webbrowser angelegt, gelesen und bearbeitet werden können. Zusätzlich wird artikelspezifische Rechtevergabe unter Verwendung der Rollen und Gruppen des Portals ermöglicht. Artikelinhalte können zur besseren Darstellung mit einem speziellen WikiSyntax formatiert und miteinander verknüpft werden. Die Datenhaltung selbst erfolgt über Dateien im XML-Format.

Es dient der Bereitstellung von Informationen, welche durch die Benutzer selbst verwaltet und gepflegt werden und soll vor allem im Rahmen von Hochschulveranstaltungen verwendet werden.

1.2 Systemvoraussetzungen

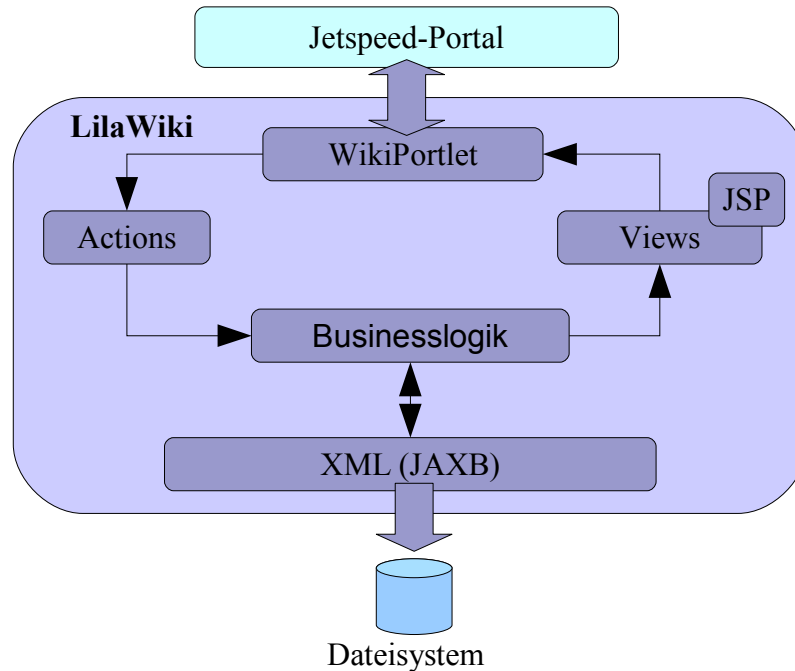
Um LilaWiki einsetzen zu können benötigt man einen Webserver auf dem Tomcat und ein Portalserver (Jetspeed-2) läuft. Der Nutzung erfolgt über einen Webbrowser.

2. Produktübersicht



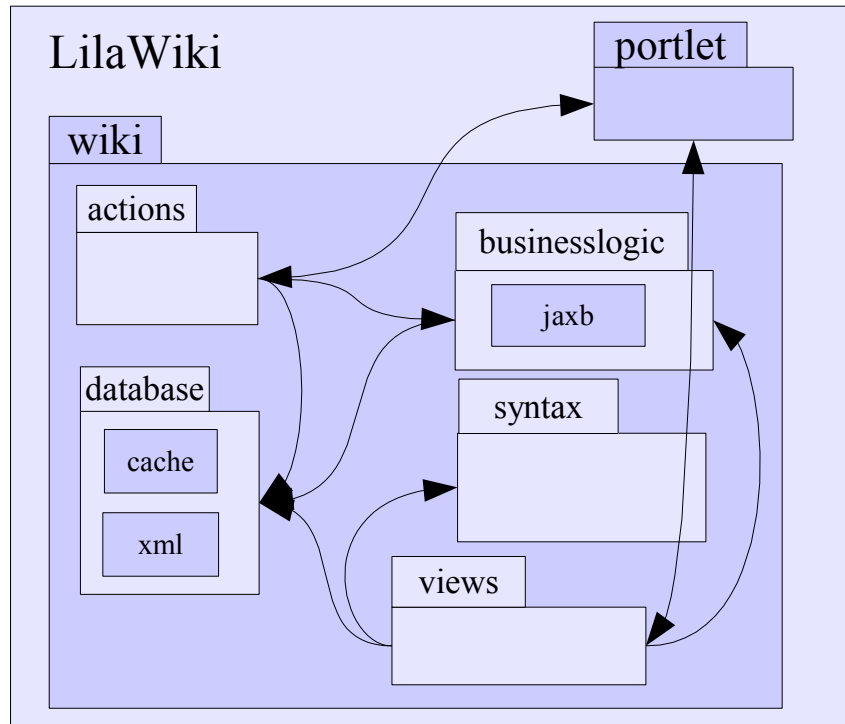
3. Grundsätzliche Struktur- und Entwurfsprinzipien des Gesamtsystems

3.1 Gesamtübersicht



Die Benutzer greifen über ein Portal auf das WikiPortlet zu, welches die Anfragen übersetzt und über eine bestimmte Aktion ausführt bzw. beantwortet. Diese Antwort wird dann über Viewklassen unter Zuhilfenahme von JSPs ausgegeben. Die Ausführung der Aktion wird von der Businesslogik übernommen, welche auf ein XML-basierende Datenhaltung im Dateisystem zurückgreift.

3.2 Paketübersicht



Das Wiki besteht aus den Paketen wiki und portlet.

Das Paket portlet regelt die Ansicht und das Paket wiki enthält die Pakete für das Modell, die Daten, die Aktionen und die Details der Ansicht.

Aus dem Portlet-Paket werden alle Aktionen weitergeleitet. Wohin sie weitergeleitet werden, wird dabei vom ActionSelector bestimmt.

Das Paket actions übernimmt alle Änderungen des Modells. Außerdem benutzen die Aktionen das Paket database um alle Änderungen umzusetzen.

Auch das Paket views greift auf database zu, um die Darstellung des Modells zu realisieren. Dazu wird das Paket Wikisyntax benutzt, was alle Informationen über die zu benutzende Syntax enthält.

Schließlich steuert view die Ansicht des Portlets für den Benutzer.

Softwaretechnik-Praktikum SS 2006

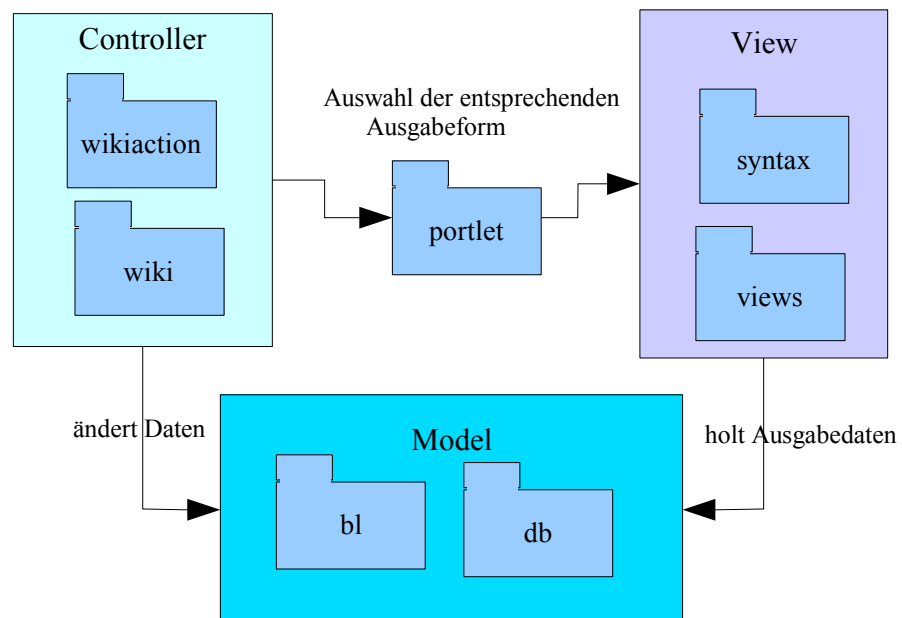
tr-06-2

Projektleiter: Marcus Lechner

3.3 MVC-Umsetzung im Wiki

Das MVC-Modell ist ein Modell für die Architektur von Softwareprodukten, bei der das Produkt in 3 Schichten aufgeteilt wird. In das Modell, den View und den Controller. Das Modell beinhaltet alle Komponenten für die optimale Datenverwaltung. Der View präsentiert die Daten. Der Controller steuert das Programm, d.h. es nimmt die Benutzeraktionen entgegen, wertet sie aus und ändert gegebenenfalls Daten im Modell.

Für das Wiki sieht diese Aufteilung folgendermaßen aus:

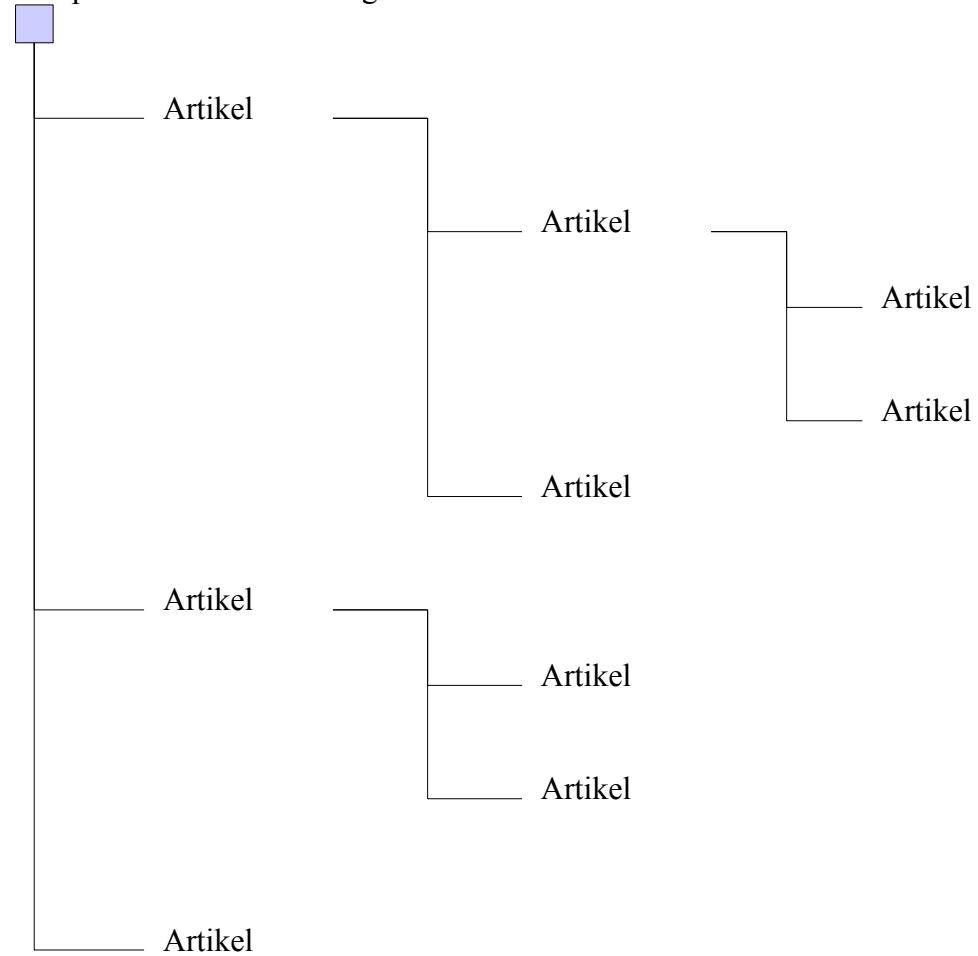


Das Package views enthält Klassen mit deren Hilfe, die entsprechende jsp ausgewählt wird. Das Package syntax enthält die Klassen, die nötig sind um die Artikeltexte darzustellen, indem sie den gespeicherten Artikeltext von Wikisyntax in HTML-Syntax formatieren. Zusammen bilden diese beiden Packages den View. Das Modell wird durch die Packages db (database) und bl (businesslogic) gebildet. Dabei ist das Package db für die Speicherung der Daten verantwortlich und das Package bl für das Rechtmanagement. Klassen die sich im Package wiki oder wikiaction befinden gehören zum Controller. Durch sie wird eine Aktion im Wiki bemerkt und die entsprechenden Daten ins Modell übertragen. Die Verbindung von Controller und View bildet das Package portlet, mittels welchen der, der Action entsprechender, View gewählt wird, welches sich dann die benötigten Daten aus dem Modell holt.

3.4 Struktur des Wikis

Artikel im Wiki werden in einer hierarchischen Baumstruktur angeordnet. Die Zuordnung erfolgt über den „gehört-zu“-Attribut eines jeden Artikels, der ihn als Unterartikel eines anderen beschreibt.

Beispielhafte Strukturierung der Artikel:



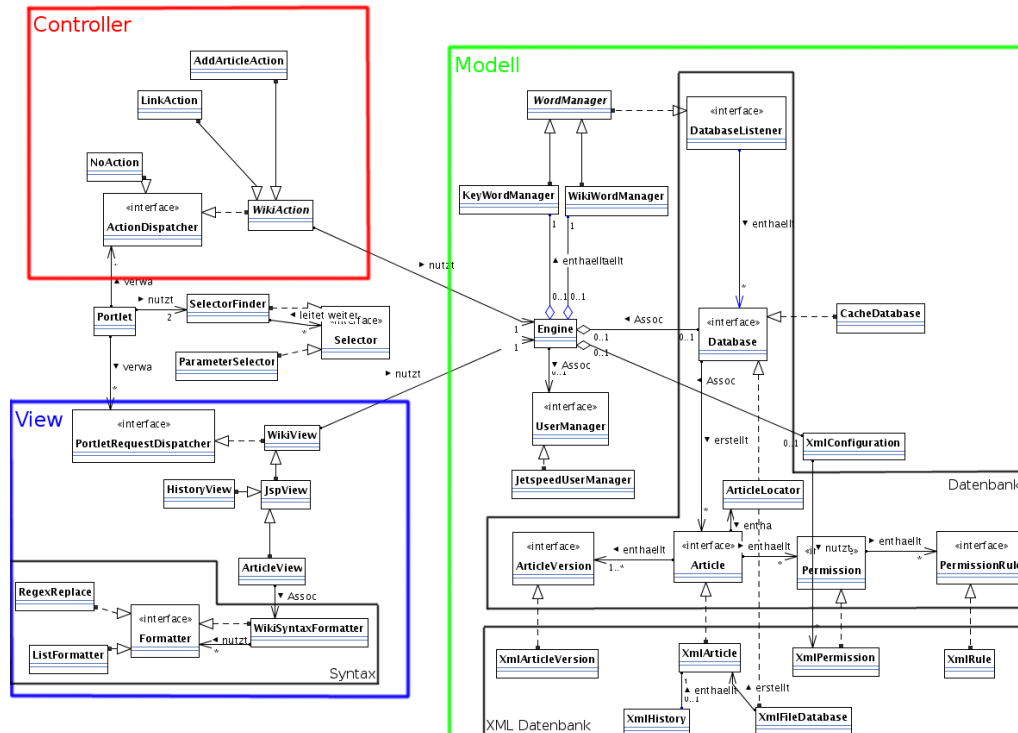
Softwaretechnik-Praktikum SS 2006

tr-06-2

Projektleiter: Marcus Lechner

4. Grundsätzliche Struktur- und Entwurfsprinzipien der einzelnen Pakete

4.1 Klassen-Übersicht



Der Grundaufbau des Wiki basiert auf dem MVC Prinzip, welches eine strenge Trennung von Model,View und Controller vorsieht. Die Klassen zwischen View und Controller dienen dazu das Modell besser umsetzen zu können als es die Portlet API von Haus aus erlaubt.

Alle Anfragen des Portlet Containers werden an eine Instanz von Portlet gestellt, welches dann mit Hilfe mehrerer Selector Instanzen den richtigen View oder die richtige Action zur Behandlung der Anfrage auswählt. Die restliche Arbeit wird dann von dem gewählten Objekt verrichtet.

Alle Views der Wiki sind Unterklassen der Klasse WikiView, welche wiederum PortletRequestDispatcher, die Schnittstelle zur Portlet API, implementiert. WikiView stellt gemeinsame Methoden zur Verfügung und fängt wikispezifische Fehler ab. Views die JSPs nutzen leiten von JspView ab, so das abgeleitete Klassen nur noch Attribute setzen müssen und nicht die eigentliche Weiterleitung mehr durchführen. Konkrete Views wie ArticleView sind dann einfach weitere Unterklassen. Zur Ausgabe von Wiki Artikeln wird das Paket syntax verwendet, welches eine Reihe von Klassen zur Formatierung von Wiki Syntax in HTML bereitstellt. Diese Klassen werden alle in WikiSyntaxFormatter zusammengefasst und dort nacheinander auf einen Artikel in Wiki Syntax angewandt.

Analog zur View Komponente leiten alle Controllerklassen, so genannte

Softwaretechnik-Praktikum SS 2006

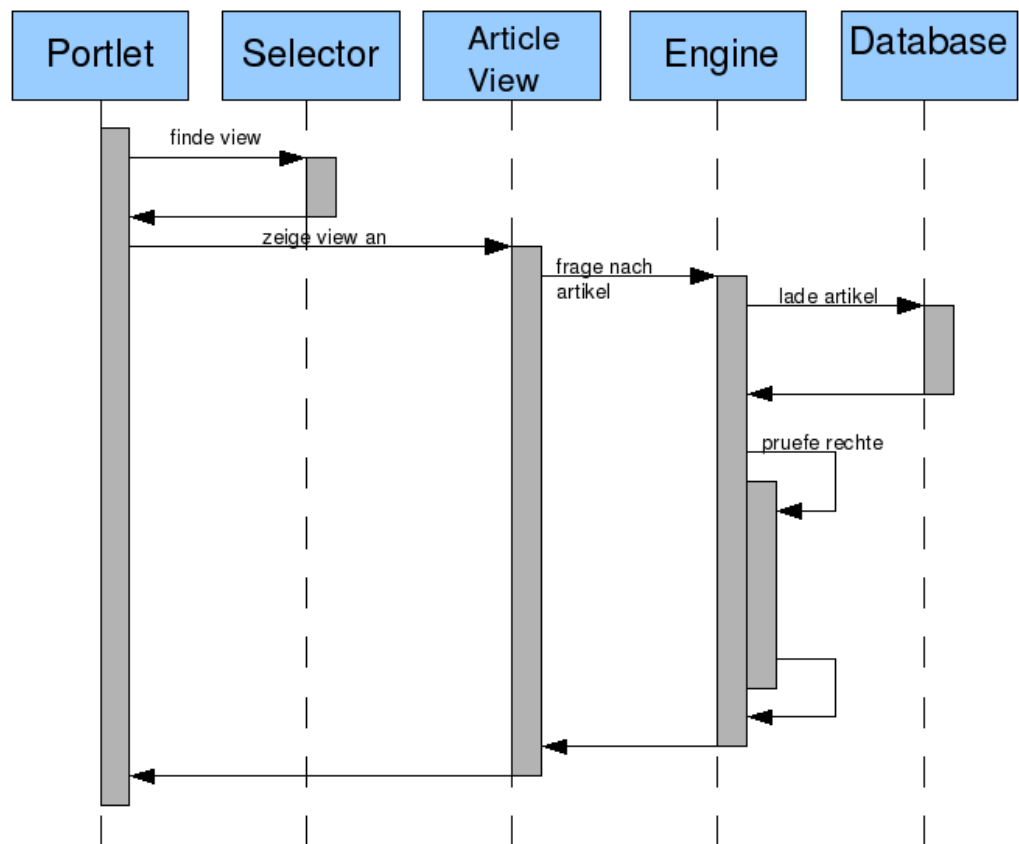
tr-06-2

Projektleiter: Marcus Lechner

Action, welche genau eine Aktion durchführen, von WikiAction ab, welches wiederum ActionDispatcher als Schnittstelle zur Portlet API implementiert. WikiAction, ähnlich wie WikiView, stellt Methoden und Daten zur Verfügung die von jeder Aktion des Wikis benötigt werden.

Das Modell der Anwendung stellt die eigentliche Wiki Logik inklusive der Datenhaltung dar. Die alles zusammenfassende Klasse ist hier Engine, sie stellt sozusagen ein Wiki dar. Um die Datenhaltung flexibel zu halten kommen hier fordergründig nur Interfaces zum Einsatz. Sie stellen das Verhalten aller datenhaltungsrelevanten Klassen zur Verfügung, nicht jedoch ihre eigentliche Implementierung. Als konkrete Implementierung kommt XmlFileDatabase zum Einsatz, welche natürlich Database implementiert. Die von ihr gelesenen Artikel sind hintergründig XmlArticle Instanzen und nutzen intern auch weitere Xml* Objekte. Eine weitere Datenbankimplementierung ist CacheDatabase welche oft gelesene Artikel im Speicher hält und andernfalls an eine zugrunde liegende Datenbank weiterleitet. Dadurch sollen teure Dateisystemzugriffe verhindert werden.

Die Konfiguration einer Engine Instanz wird immer in XML gespeichert und intern als ein XmlConfiguration Objekt dargestellt.

*Viewzuordnung*

4.2 Einsatz von Action- und Render-Objekten

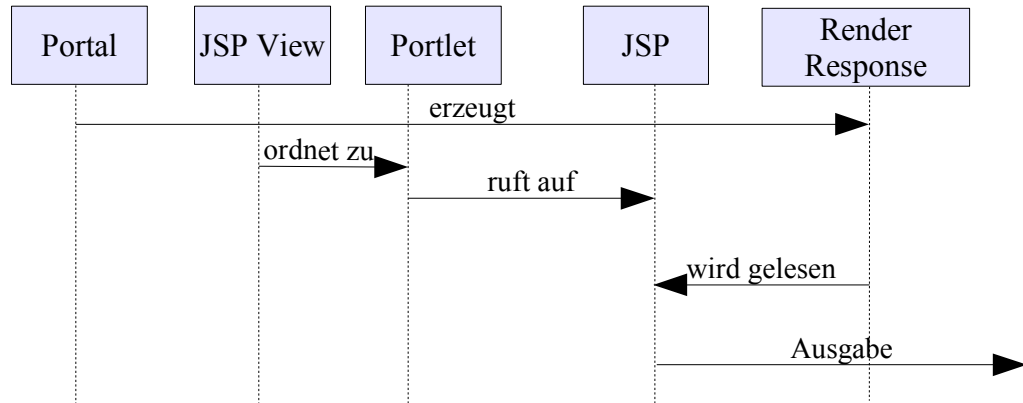
Portlets unter Jetspeed-2 unterscheiden bei ihren Aufgaben zwischen Action- und Render-Aufgaben. Wenn ein Portlet eine Anfrage bekommt (ein User klickt auf einen Link/Button/etc.), weiß das Portlet, ob dies ein Action- oder Render-Request ist. Je nachdem welche von beiden Aufgaben angefordert wird, wird PortletRequest-Objekt oder ein ActionRequest-Objekt erzeugt.

Der wesentliche Unterschied der beiden Aufgaben liegt im Dateizugriff. Beim Rendern werden bestehende Datenbestände ausgelesen und für die Anzeige verarbeitet. Dazu gehört das Parsen eines Artikels, etc.

Bei einer Action wird der Datenbestand geändert. Wenn ein Benutzer einen Artikel bearbeitet und dann auf „Absenden“ klickt, wird eine Action aufgerufen, die dann den Artikel prüft und abspeichert. Die Action greift dabei auf die Engine zurück um mit den Dateien arbeiten zu können.

Der Einfachheit halber wird dies in den Übersichten aber nicht gesondert aufgeschlüsselt. Beide Anfragetypen werden weiterhin als Aktion bezeichnet.

4.3 Einsatz von JSPs



Die konkrete Seitendarstellung wird von den JSPView-Klassen gesteuert. Das Portal erzeugen zunächst ein RenderResponse-Objekt, das von der ausgeführten Action mit Ausgabedaten beladen wird (hier nicht zu sehen). Außerdem ordnen sie dem Portlet diese Response und eine bestimmte JSP zur Ausgabe zu. Vom Portlet aus wird diese JSP dann aufgerufen um die Daten des angelegten RenderResponse-Objektes schließlich in HTML formatiert auszugeben.

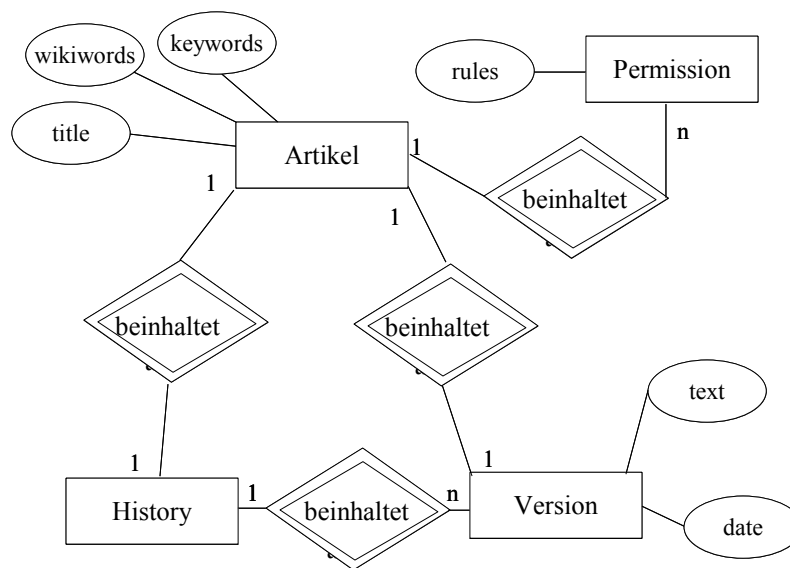
Softwaretechnik-Praktikum SS 2006

tr-06-2

Projektleiter: Marcus Lechner

4.4 Artikelaufbau

Ein Artikel des Wikis besteht aus seinem Titel, seiner aktuellen Version, den Stichwörter, seiner WikiWords seiner History und seinen Rechten. In der aktuellen Version ist der Text des Artikel sowie das Datum der Erstellung der Version. Die History speichert alle alten Versionen des Artikels. In den Rechteobjekt (Permission) werden die Zugriffsrechte des Artikels mittels Regeln gespeichert. Stichwörter (keywords) sind diejenigen Wörter, die für die Stichwortsuche verwendet werden. Die WikiWords eines Artikels sind die Wörter über welcher er im Wiki verlinkt wird. Sowohl die Stichwörter als auch die Wikiwords werden in Stringlisten gespeichert. Außerdem hat jeder Artikel eine ArticleLocator mit Datenbank, welcher genutzt wird um den Artikel im Dateisystem zu identifizieren, da der Dateiname nicht nur der Titel sondern die Hierarchie des Artikels wiedergibt. Das heißt im Dateiname eines Artikel stehen zusätzlich noch die Titel seiner Oberartikel. Die aktuelle Version des Artikels wird im Dateisystem im Ordner Article und seine History im Ordner History gespeichert. Diese beiden Ordner werden vom Wiki automatisch erstellt.

Artikelaufbau

Softwaretechnik-Praktikum SS 2006

tr-06-2

Projektleiter: Marcus Lechner

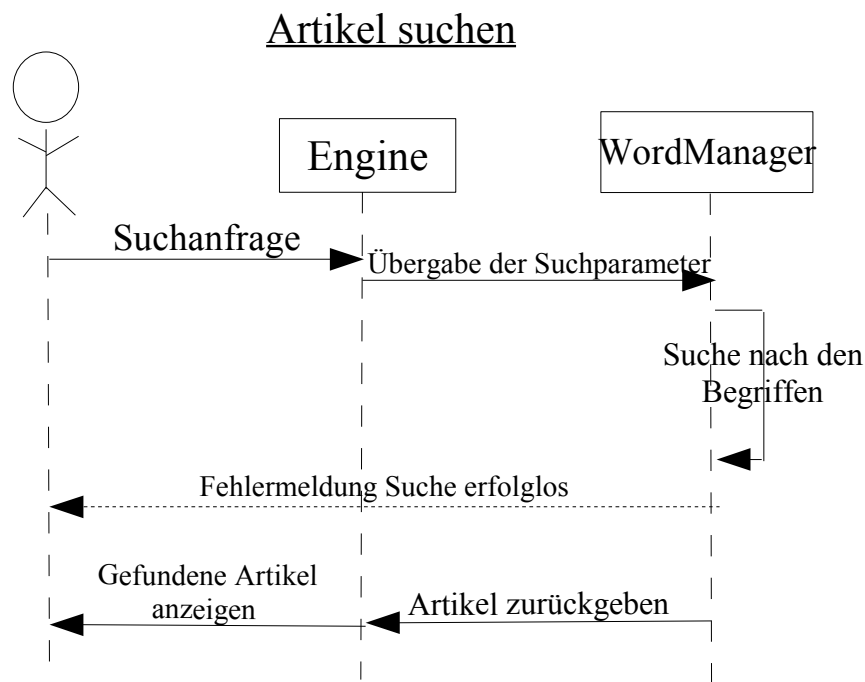
4.5 Engine

Die Engine bildet das Kernstück des Businesslogic-Pakets. Über sie laufen alle Anfragen an die Anwendungslogik. Die Geschäftsfälle (wie z.B. Artikel suchen, lesen, erstellen) werden von ihr koordiniert.

Dazu unterhält sie u.a. die Instanz der Datenbank und Manager-Objekten.

4.6 Manager-Objekte

Von Manager-Objekt abgeleitet sind WikiWordManager, KeyWordManager und SubArticleManager. Sie bilden über Indexstrukturen auf den aktuellen Datenbestand ab und ermöglichen so schnelles Durchsuchen der Datenbank nach bestimmten Kriterien. Damit die Indizierung stets aktuell bleibt, sind sie bei der Datenbank als Listener (d.h. Beobachter) eingetragen und verfolgen so alle Änderungen des Datenbestandes.

4.7 Artikelsuche

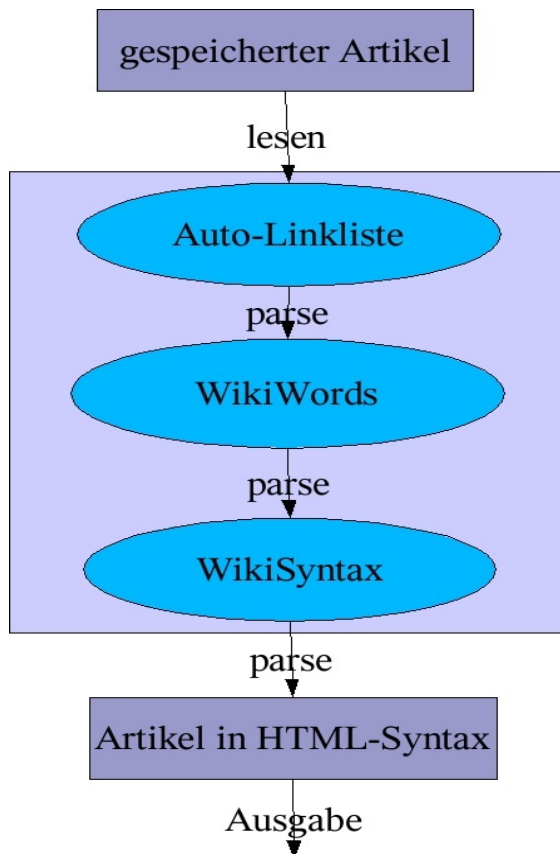
Anfragen zur Suche von Artikeln werden an die Engine gestellt, welche sie dann an das entsprechende Managerobjekt weiterleitet. Diese ermitteln anhand ihrer Indexstrukturen passende Artikel und übergeben ihren ArticleLocator, damit sie aus der Datenbank gelesen werden können.

4.8 Text-Linkliste

Zusätzlich zur WikiWorderkennung verfügt das Wiki über eine weitere Funktion zu automatischen Verknüpfung von Artikelinhalten:

Die Auto-Linkliste ist eine simple Liste, die einem Wort oder einer Wortgruppe eine komplette URL zuordnet. Würde z.B. der Wortgruppe „Uni Leipzig“ die URL „<http://www.uni-leipzig.de>“ zugeordnet, so würde jedes Vorkommen dieser Wortgruppe in allen Artikeln automatisch mit dieser URL verknüpft werden. Dies ermöglicht die statische Einbindung externer Quellen (z.B. Verknüpfungen zum Elate-Portal oder Uni-Seiten), aber auch Wiki-interner Artikel.

4.9 Parser



Softwaretechnik-Praktikum SS 2006

tr-06-2

Projektleiter: Marcus Lechner

Der Artikel-Parser bekommt einen gespeicherten Artikel und wandelt diesen in HTML-Format um. Dabei werden mehrere Parse-Ebenen durchlaufen:

1. Systembegriffe

Die Systembegriffe bestehen aus einer Liste von Wörtern, die automatisch verlinkt werden. In der Liste stehen z.B. Namen von Professoren oder Vorlesungen und die dazugehörigen Seiten, die für diese Worte verlinkt werden sollen. (Hier sind vollständige Pfadangaben zu verwenden!) Die Liste der Systembegriffe muss manuell gewartet werden, da sie speziell für die Nutzung im elatePortal gedacht ist.

In der ersten Parse-Stufe wird der Artikel nach diesen Begriffen abgesucht und ersetzt die Begriffe durch entsprechende Links in Wikisyntax.

2. Wikiwords

In der zweiten Parse-Stufe werden alle anderen WikiWords zu entsprechenden Links in Wikisyntax umgewandelt. Wikiwords müssen als solche vom Benutzer gekennzeichnet werden der den Text erstellt. Bsp: [IchBinEinWikiWord] würde dann auf die Seite IchBinEinWikiWord verweisen.

3. Wikisyntax

Im dritten und letzten Parse-Schritt wird die Wikisyntax des Artikels in HTML-Format überführt. Unten haben wir eine vorläufige Übersicht der umzusetzenden Wikisyntax aufgelistet.

Bsp: !Überschrift 1 wird zu <h1>Überschrift</h1>

Ein Ausschnitt der Wikisyntax-Liste:

```
! <h1>Überschrift 1</h1>
!! <h2>Überschrift 2</h2>
!!! <h3>Überschrift 3</h3>
# nummerierte Liste Stufe 1
## nummerierte Liste Stufe 2
### nummerierte Liste Stufe 3
* ungeordnete Liste Stufe 1
** ungeordnete Liste Stufe 2
*** ungeordnete Liste Stufe 3
"kursiver Text – 2 Apostrophs"
"fetter Text – 3 Apostrophs"
""fetter kursiver Text – 5 Apostrophs""
```

4.10 Rechtemanagement

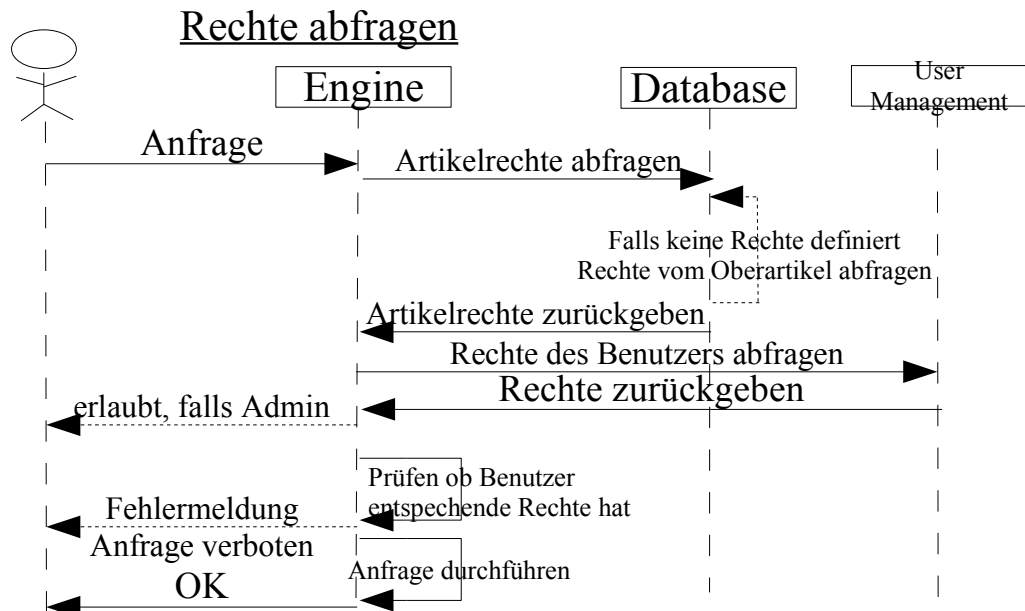
Für jeden Artikel könne eigene Zugriffsrechte vergeben werden:

	Benutzer in Rollen		Benutzer in Gruppen		
	Benutzer	Gäste	DBS	ADS	SWT
Lesen	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ändern	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Anlegen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Löschen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Ein Haken bedeutet dabei, erlaubt. Kein Haken, verboten.

Ein Benutzer darf eine Aktion genau dann ausführen, wenn für seine Eigenschaften (Mitgliedschaft in Gruppen, oder übernehmen einer Rolle) mindestens ein "Erlaubt" gefunden wird. D.h. für das Bild: Bin ich normaler Benutzer kann ich den Artikel nur ändern, wenn ich in der Gruppe SWT bin. Mitgliedschaften in anderen Gruppen beeinflussen das nicht. Löschen kann den Artikel niemand. Administratoren dürfen dabei aber grundsätzlich alles. Deswegen findet für sie auch keine separate Rechtevergabe statt. Wurden keine Rechte definiert, so wird der Oberartikel befragt (Vererbung). Hat ein Artikel keinen Oberartikel, werden für definierte Standartrechte angewendet.

Die Überprüfung eine Anfrage läuft folgendermaßen ab:



4.11 Rollenzuordnung

Rollen mit Administrator-Rechten:



Für den Betrieb des Wikis werden im Prinzip nur 3 Rollen benötigt:

Administrator, normaler Benutzer und Gast.

Um die Übernahme der in Jetspeed zugeordneten Rollen zu erleichtern, werden diese auf die Wiki-internen Rollen gemappt. Wer nicht als Administrator eingetragen wurde ist dabei automatisch normaler Benutzer.

Die Gast-Rolle von Jetspeed wird automatisch übernommen und muss nicht gesondert vergeben werden.