

Entwurfsbeschreibung des Wikis

1. Allgemeines

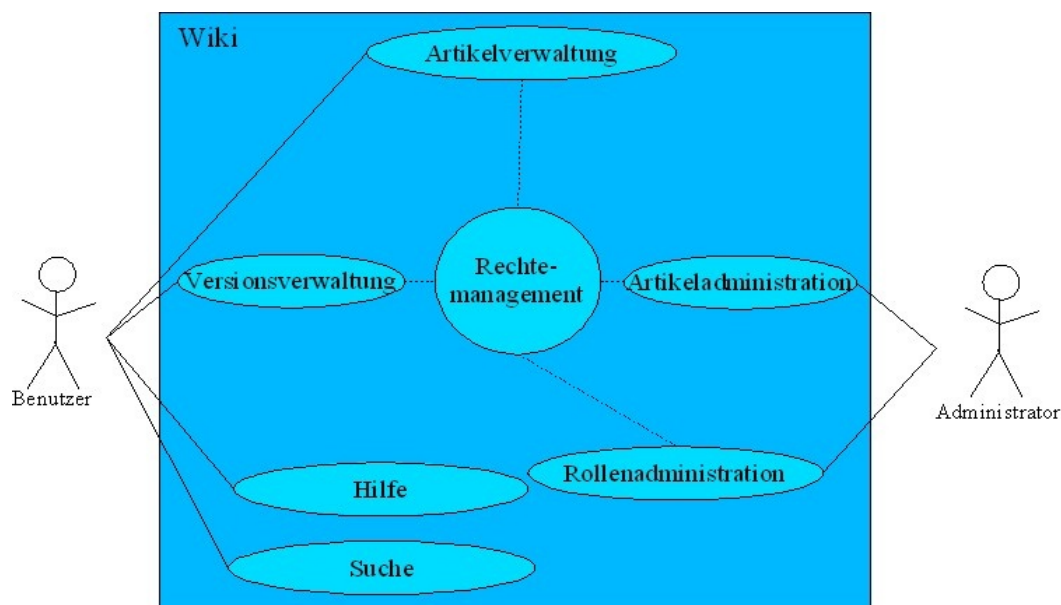
1.1 Charakterisierung

Das Wiki ist ein Portlet, das unter Jetspeed-2 (bzw. unter Portalen die auf Jetspeed-2 aufbauen) läuft. Unterschiedliche Benutzergruppen aus Jetspeed-2 sollen dabei auch in Wiki unterschiedliche Rechte haben. Artikel im Wiki sollen von allen Benutzern, die entsprechende Rechte haben, über einen Webbrowser angelegt, gelesen und bearbeitet werden können. Diese Beiträge werden dann als XML-Dateien gespeichert. Zur Formatierung eines Artikel wird eine WikiSyntax verwendet. Das Wiki dient im Allgemeinen der Bereitstellung von Informationen, welche durch die Benutzer verwaltet und gepflegt werden. Das Wiki soll vor allem im Rahmen von Hochschulveranstaltungen verwendet werden, um Hinweise und Informationen bereitzustellen, die von den Lehrenden leicht bearbeitet werden können. Das Wiki läuft innerhalb eines Internet-Portals (z.B. des elatePortals) und wird von Studenten, Dozenten und Professoren genutzt, die keine tiefergehenden Computerkenntnisse benötigen.

1.2 Systemvoraussetzungen

Um das Wiki einsetzen zu können benötigt man einen Webserver (z.B. Apache) auf dem Tomcat und ein Portalserver (Jetspeed-2) läuft. Die Benutzer, die das Wiki verwenden sollen, benötigen nur einen Webbrowser.

2. Produktübersicht



3. Grundsätzliche Struktur- und Entwurfsprinzipien des Gesamtsystems

3.1 Aufbau des Wikis

Das Wiki besteht im wesentlichen aus Artikeln. Diese bestehen aus einem Titel, seiner aktuellen Version und einer History. Die Version besteht dabei aus dem Artikeltext in WikiSyntax, KeyWords, seiner Rechtekonfiguration und einem Attribut „gehörtZu“. Die KeyWords enthalten Stichwörter, die kurz beschreiben worum es in dem Artikel geht, um ihn über die Suche schnell zu finden, während die WikiWords der automatischen Verknüpfung von Artikeln dienen.

Die Artikel sind in einer hierarchischen Ordnung aufgebaut. Jeder Artikel hat ein Attribut („gehörtZu“) in dem gespeichert ist, zu welchem Artikel er gehört. (Bei Artikeln die keinem übergeordneten Artikel zugeordnet sind, ist dieser Wert NULL.) So lässt sich ein Artikel-Baum aufbauen. Sinn davon ist es, dass Wiki-Artikel gruppiert werden können, weil sie verwandte Themen beinhalten (z.B. gehören zu einer Vorlesung die Seiten: Skript, Übungen, etc.).

Des Weiteren gibt es zu jedem Artikel mehrere Versionen, die in der History verwaltet werden. Alle Benutzer die über entsprechende Rechte verfügen, können jetzt den Artikel bearbeiten, wodurch eine neue Version entsteht. Administratoren können auch alte Versionen wieder „zurückholen“ („VersionRecall“).

3.2 Rechtemanagement

Das Rechtemanagement orientiert sich von der Struktur her am UNIX-Rechtesystem.

Es gibt drei Benutzergruppen (Admin, User und Guest) und 4 Rechte-Optionen (lesen, bearbeiten, anlegen und löschen).

Für die Gruppen gibt es folgende Standardrechte:

- Gäste dürfen nur lesen
- Benutzer dürfen lesen, anlegen und bearbeiten
- Admins dürfen alles

Außerdem können die Admins Rechte für einen Artikel vergeben, die die Standardrechte überschreiben. Durch die hierarchische Struktur der Artikel werden die Rechte vererbt.

Bsp: Der Artikel zur Vorlesung „Softwaretechnik“ erlaubt es Gästen, diesen zu Bearbeiten. Dann können Gäste auch die Subartikel „Skript“ und „Übungen“ bearbeiten, sofern für diese nicht andere Rechte gesetzt sind.

Die technische Umsetzung verläuft folgendermaßen: Beim Aufruf eines Artikels wird das Rechteobjekt desselben überprüft und entsprechend die erlaubten Funktionen „freischaltet“. Hat ein Artikel kein Rechteobjekt (bzw. ist dieses NULL), dann wird der übergeordnete Artikel aufgerufen und dessen Rechteobjekt auf die gleiche Weise überprüft. Hat ein Artikel keine festgelegten Rechte und keinen übergeordneten Artikel, dann werden die oben beschriebenen Standardrechte verwendet.

Softwaretechnik-Praktikum SS 2006

tr-06-2

Projektleiter: Marcus Lechner

3.3 die Engine

Die Engine ist die zentrale Verwaltungseinheit des Wikis. Beim Initialisieren des Wikis sowie beim Anlegen oder Bearbeiten eines Artikels muss sich jeder Artikel an der Engine anmelden. Die Engine registriert den Artikel dabei in verschiedenen Verwaltungsstrukturen die schnellen Zugriff auf die Artikel gewährleisten sollen. Die Engine registriert z.B. auch die KeyWords jedes Artikels, damit später die Suche nach Stichwörtern schnell ausgeführt werden kann.

3.4 Interner Portlet Layer

Da die Portlet API nur eine sehr dünne Schicht zur Behandlung von Portlet Ereignissen bereitstellt (GenericPortlet), wird diese nicht verwendet sondern eine selbst entwickelte, etwas mächtigere, Schicht. Ein Portlet erhält 2 Typen von Ereignissen die es behandeln muss: RenderRequests und ActionRequests. Diese Einteilung wird weiter verwendet, jedoch wird die Behandlung der Ereignisse in jeweils ein Interface, PortletRequestDispatcher und ActionDispatcher, verlagert. Aktionen, die die Ereignisse verarbeiten sind dann genau Klassen welche diese Interfaces implementieren. Dies ermöglicht eine saubere Trennung zwischen Darstellung und Behandlung von Benutzeraktionen und auch zwischen den einzelnen Aktionen überhaupt (Modularität!). Da Darstellung und Reaktion auf Benutzereingaben letztendlich zusammen eine Seite, welche dem Benutzer angezeigt wird, ausmacht wird beides im Interface Pages zusammengeführt. Diese ist dazu gedacht die Kompositionen von PortletRequestDispatcher und ActionDispatcher anzupassen durch eine Implementierung des Page Interfaces, jedoch nicht dazu um die Trennung aus Anzeige und Benutzerereignisbearbeitung aufzuheben. Die Implementierung speichert ein PortletRequestDispatcher und ein ActionDispatcher Objekt und leitet die Anfragen entsprechend weiter. Um bei einer Anfrage an das Portlet überhaupt zu wissen welche Seite derzeit angezeigt wird, wird das Selector Interface verwendet. Es enthält eine Methode um aus einem PortletRequest die anzuzeigende Seite zu ermitteln. Zur Zeit wird eine Implementierung verwendet, welche einfach den Wert eines Parameters der Anfrage an das Portlet gegen feste Zeichenketten vergleicht und dadurch die richtige Seite auswählt. Die gleiche Technik kommt derzeit auch zum Einsatz um bei einem ActionRequest den richtigen ActionDispatcher auszuwählen. Durch die strikte Verwendung von Interfaces und durch die Wahl sehr allgemeiner Methoden ist die Erweiterbarkeit des Wikis jederzeit gewährleistet.

Softwaretechnik-Praktikum SS 2006

tr-06-2

Projektleiter: Marcus Lechner

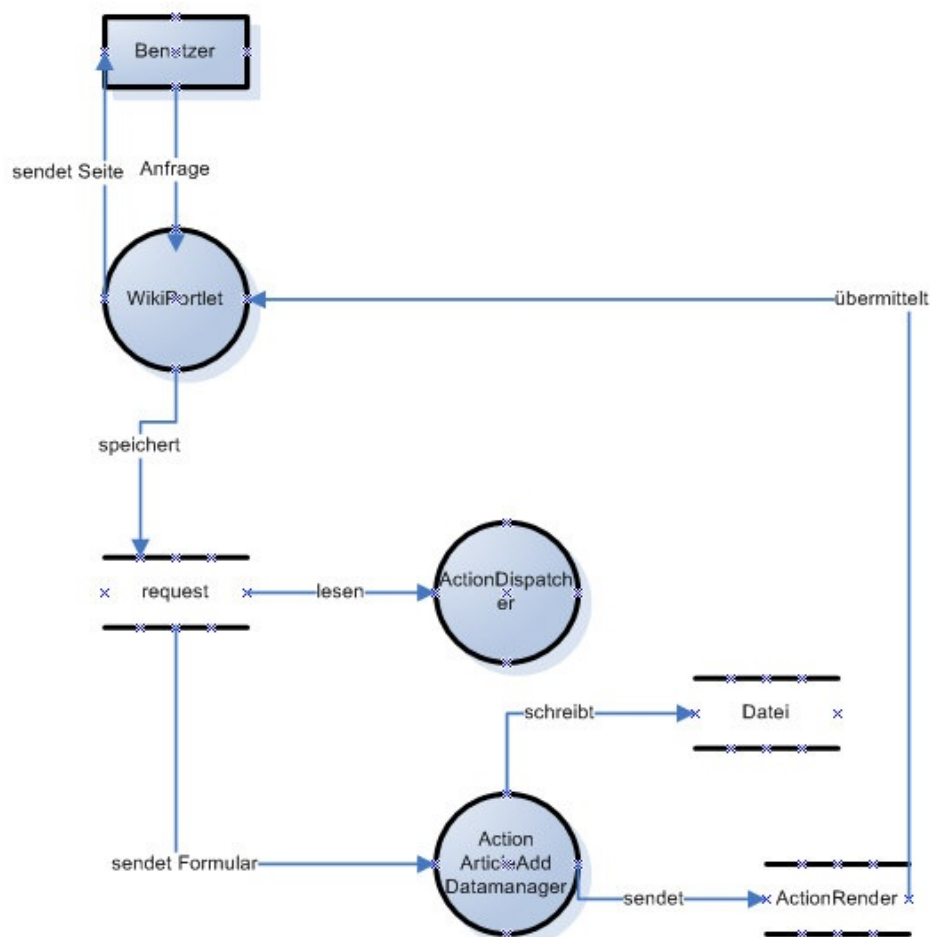
3.5 Aufruf von Funktionen

Wenn in dem Wiki eine Aktion ausgeführt wird, sendet das Portlet ein REQUEST zum ActionDispatcher, der entscheidet welche Action für diese Anfrage zuständig ist und diese startet. Dazu werden Methoden aus dem Model benutzt.

Bsp: Ein Benutzer ruft die Funktion „Artikel anlegen“ auf. Später ruft er dann die Funktion „Artikel bearbeiten“ auf. Beide Funktionen starten den gleichen ActionDispatcher und dieser ruft dann im ersten Fall eine Funktion zum Artikel-Anlegen und im zweiten Fall eine Funktion zum Artikel-Bearbeiten auf.

3.6 Fluss von Ein- und Ausgabe

Der Fluss von Ein- und Ausgaben folgt immer dem selben Prinzip und soll hier anhand eines Beispiels beschrieben werden:



Softwaretechnik-Praktikum SS 2006

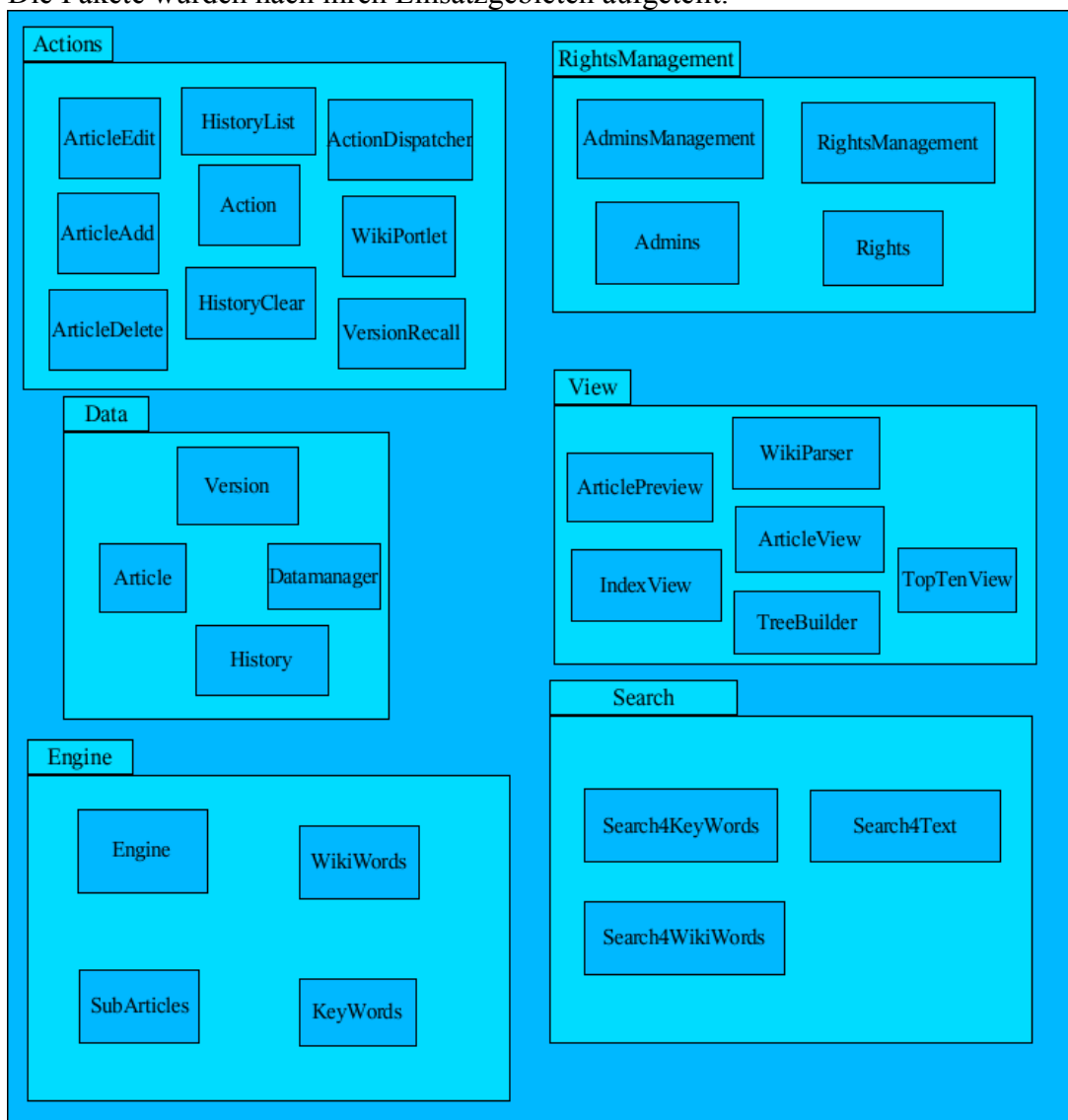
tr-06-2

Projektleiter: Marcus Lechner

Sendet der Benutzer eine Anfrage an das "WikiPortlet" wird diese in einem "Request-Object" gespeichert. Der ActionDispatcher liest dieses und ruft, wie in 3.5 beschrieben, die entsprechende Action auf. In diesem Beispiel sei das ActionAdd. Die Daten des Request-Objects werden dort ausgewertet und bearbeitet. In diesem Fall würde der Artikel über den "DataManager" gespeichert. Die Ausgabe wird in ein ActionRender-Object geschrieben, dann über die entsprechende JSP vom WikiPortlet gelesen und formatiert an den Benutzer ausgegeben.

4. Grundsätzliche Struktur- und Entwurfsprinzipien der einzelnen Pakete

Die Pakete wurden nach ihren Einsatzgebieten aufgeteilt.



Ihre Klassen übernehmen folgende Funktionen:

Softwaretechnik-Praktikum SS 2006

tr-06-2

Projektleiter: Marcus Lechner

4.1 Actions

Dieses Paket beinhaltet alle Aktionen. Sie dienen zum Verwalten des Wikis.

- ◆ Action: Oberklasse für alle Aktionen, bekommt Requestobjekt und schreibt Ausgabe in Renderobjekt
- ◆ ActionDispatcher: weist übergebener Action die auszuführende Methode zu
- ◆ PortletRequestDispatcher: Speichert und übermittelt Ausgabe der Actions
- ◆ ArticleDelete: informiert Articleobjekt über seine Löschung und löscht ihn anschließend
- ◆ ArticleAdd: erzeugt neues Articleobjekt
- ◆ ArticleEdit: ändert Articleobjekt
- ◆ HistoryClear: leert das Historyobjekt eines Articleobjekts
- ◆ HistoryList: erstellt einen Index der Versionen eines Articles
- ◆ VersionRecall: ersetzt Article durch Version und löscht jüngere Versionen aus History
- ◆ WikiPortlet: eigentliches WikiPortlet, Controller

4.2 RightsManagement

Das Paket Rechteverwaltung dient dazu, die Rechte innerhalb des Wikis zu verwalten. Man kann damit die Rechte für die einzelnen Artikel festlegen.

- ◆ AdminsManagement: konfiguriert wer Admin ist
- ◆ Admins: beinhaltet Liste der Administratoren
- ◆ RightsManagement: konfiguriert Artikelrechte
- ◆ Rights: beinhaltet die Rechteinstellungen und gibt Auskunft Rechte des aktuellen Benutzers

4.3 Data

Das Daten-Paket enthält alles was in XML-Dateien gespeichert werden muss. Um die XML-Dateien zu erstellen und zu verwalten, gibt es den DataManager, der diese Aufgaben übernimmt.

- ◆ DataManager: Verwaltet das Lesen und Speichern von Objekten. Dadurch, dass der DataManager allein für den direkten Dateizugriff zuständig ist, kann man leicht ein anderes Speichermedium (z.B. Datenbank) verwenden. Dazu tauscht man den aktuellen durch einen entsprechenden DataManager aus.
- ◆ Article: beinhaltet Articledaten und zugehöriges Historyobjekt, fügt neue Version in History ein, ändert Articledaten
- ◆ History: beinhaltet alle Versionen eines Articles
- ◆ Version: beinhaltet alle Daten einer Version (Text, Datum,...)

Softwaretechnik-Praktikum SS 2006

tr-06-2

Projektleiter: Marcus Lechner

4.4 View

Dieses Paket realisiert die grafische Oberfläche des Wikis. Die gespeicherten Artikel werden geparkt um dann im Webbrowser korrekt angezeigt zu werden.

- ◆ WikiParser: formatiert Artikelquelltext entsprechend Wikisyntax
- ◆ ArticleView: stellt Artikel dar
- ◆ ArticlePreview: stellt Artikel als Vorschau dar
- ◆ TreeBuilder: erstellt Hierarchie der Artikel
- ◆ IndexView: zeigt einen Index aller Artikel an
- ◆ TopTenView: zeigt 10 zuletzt geänderten Artikel an

4.5 Search

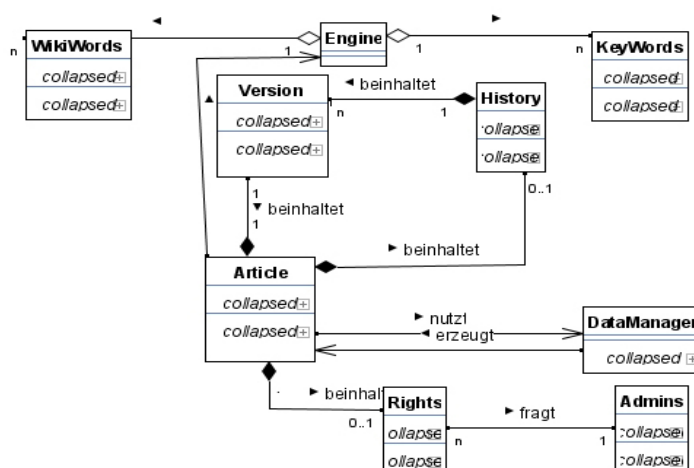
Die Suche enthält 3 Klassen welche die Artikel nach unterschiedlichen Gesichtspunkten durchsuchen.

- ◆ Search4Keywords: sucht nach Stichwörtern in allen Artikeln
- ◆ Search4WikiWords: durchsucht einen Artikel nach WikiWords.
- ◆ Search4Text: sucht im Text aller Artikel

4.6 Engine

Die Engine ist das wichtigste Paket des Wikis. Sie enthält eine Übersicht der WikiWords, KeyWords und der Artikel.

- ◆ Keywords: verwaltet die KeyWords und die zugehörigen Artikel
- ◆ WikiWords: verwaltet alle WikiWords mit zugehörigen Artikeln
- ◆ Engine: verwaltet WikiWords und KeyWords der Artikel

4.7 Übersicht des Zusammenspiels

Dieser Ausschnitt des Klassendiagramms zeigt übersichtlich den Aufbau eines Artikels und die verwaltende Rolle der Engine, die alle Artikel 'kennt'.

Der DataManager übernimmt dabei das Laden und Speichern von Artikeln.