

# Entwurfsbeschreibung

v1.2

21.06.2010

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines</b>	<b>2</b>
<b>2</b>	<b>Produktübersicht</b>	<b>3</b>
<b>3</b>	<b>Struktur des Gesamtsystemes</b>	<b>4</b>
<b>4</b>	<b>Struktur der Einzelkomponenten</b>	<b>5</b>
4.1	Übersicht . . . . .	5
4.2	Klassenübersicht . . . . .	7
4.3	Editor . . . . .	9

## 1 Allgemeines

OntoWiki ist ein Online-Wissensmanagement-System, dessen Inhalte von den Nutzern nicht nur gelesen, sondern auch nach eigenen Vorstellungen geändert und ergänzt werden können. “Onto” bezieht sich dabei auf die semantische Struktur. Die Open-Source-Software OntoWiki ist also eine Plattform zur agilen Verwaltung von Semantic-Web-Wissensbasen und vermag Informationen komplex zu verknüpfen. Semantische Daten werden in unterschiedlichen Ansichten präsentiert und können intuitiv verändert werden.

Die Erweiterung für OntoWiki soll als Alternative zu der Triple-Ansicht eine Darstellung als Wiki-Seite ermöglichen. Für den Nutzer ist diese dann wie in einem gewöhnlichen Wiki-Seite zu bearbeiten und zu pflegen.

## 2 Produktübersicht

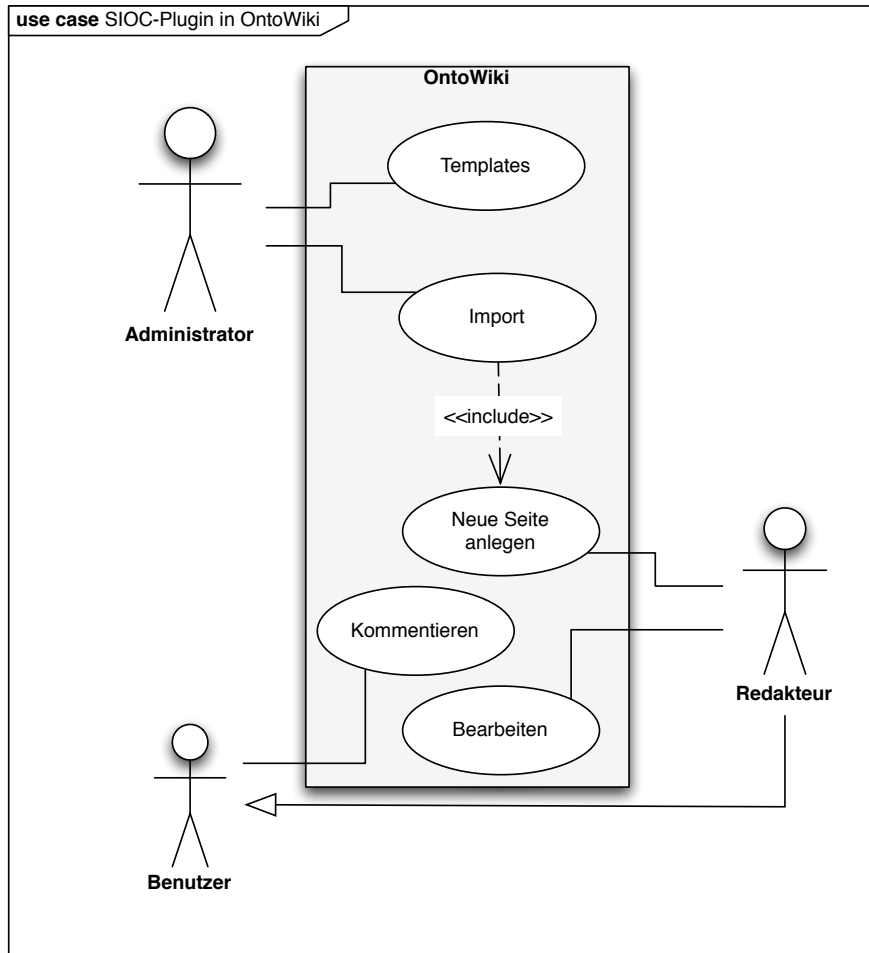


Abbildung 1: Produktübersicht

Das Produkt ermöglicht die Darstellung der Daten im OntoWiki als Wiki-Seite in einem eigenen Tab. Diese Wiki-Seiten können über einen eingebundenen Editor bearbeitet werden. Darüber hinaus können diese Artikel auch kommentiert und bewertet werden. Die Wikiseiten sind untereinander verlinkt und können aber auch auf externe Seiten verweisen. Es besteht ebenfalls die Möglichkeit bereits vorhandene Wikis zu importieren. Bei Bedarf können auch ältere Versionen einer Wiki-Seite aufgerufen, miteinander verglichen oder wiederhergestellt werden.

### 3 Struktur des Gesamtsystemes

Das Projekt wird als eine Erweiterung vom Typ Komponente realisiert. Die OntoSioc-Komponente beinhaltet den OntoSiocController, den OntoSiocHelper, die WikiPage, die in templates liegende view.phtml und das EditorWidget.

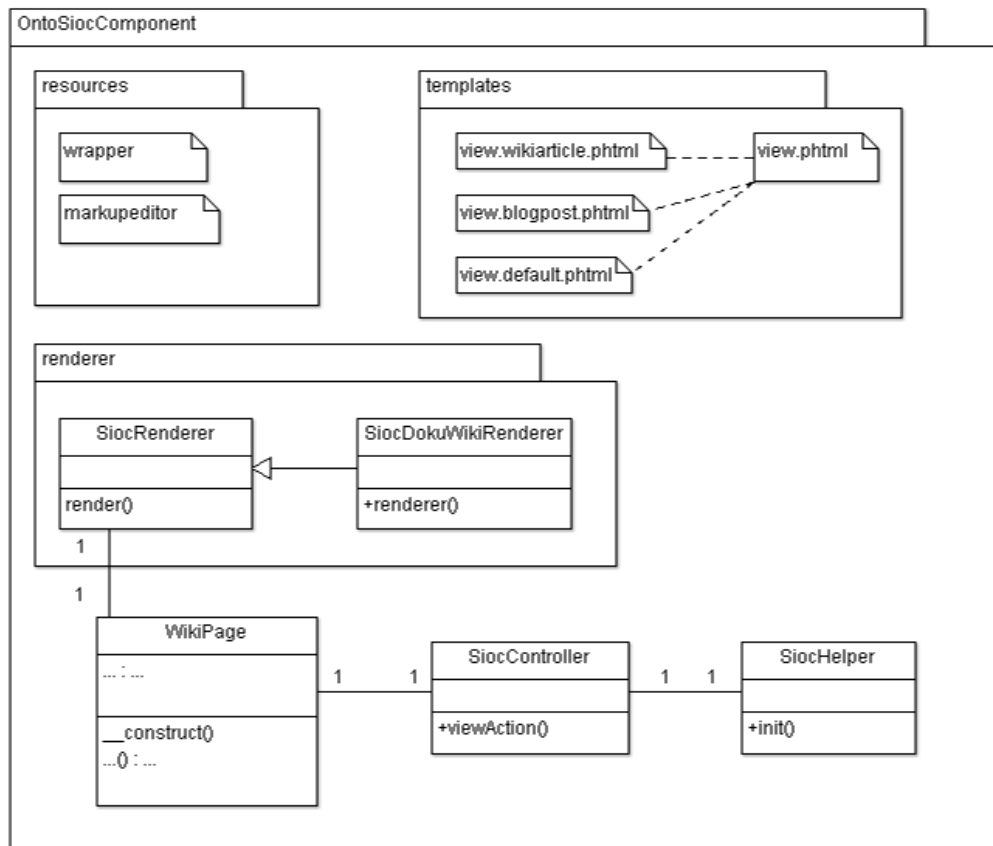


Abbildung 2: Paketdiagramm

## 4 Struktur der Einzelkomponenten

### 4.1 Übersicht

Der Benutzer kann nach dem Aufrufen der Startseite sich eine konkrete Wikiseite anzeigen lassen oder ein Objekt ohne SIOC. Im Objekt ohne SIOC kann eine neue Seite angelegt werden mittels des Editors. Wenn die Wikiseite angezeigt wird kann sie über den Editor bearbeitet oder gelöscht werden. Nach dem Löschen gelangt man auch wieder auf ein Objekt ohne SIOC. Weiterhin können die Wikiseiten durch Aufruf der Kommentarfunktion kommentiert werden. Zusätzlich kann von der Wikiseite aus die Versionshistorie angezeigt werden und gegebenenfalls wieder hergestellt werden. Ist der Benutzer als Administrator eingeloggt, kann er von der Startseite aus auch andere Wikis importieren.

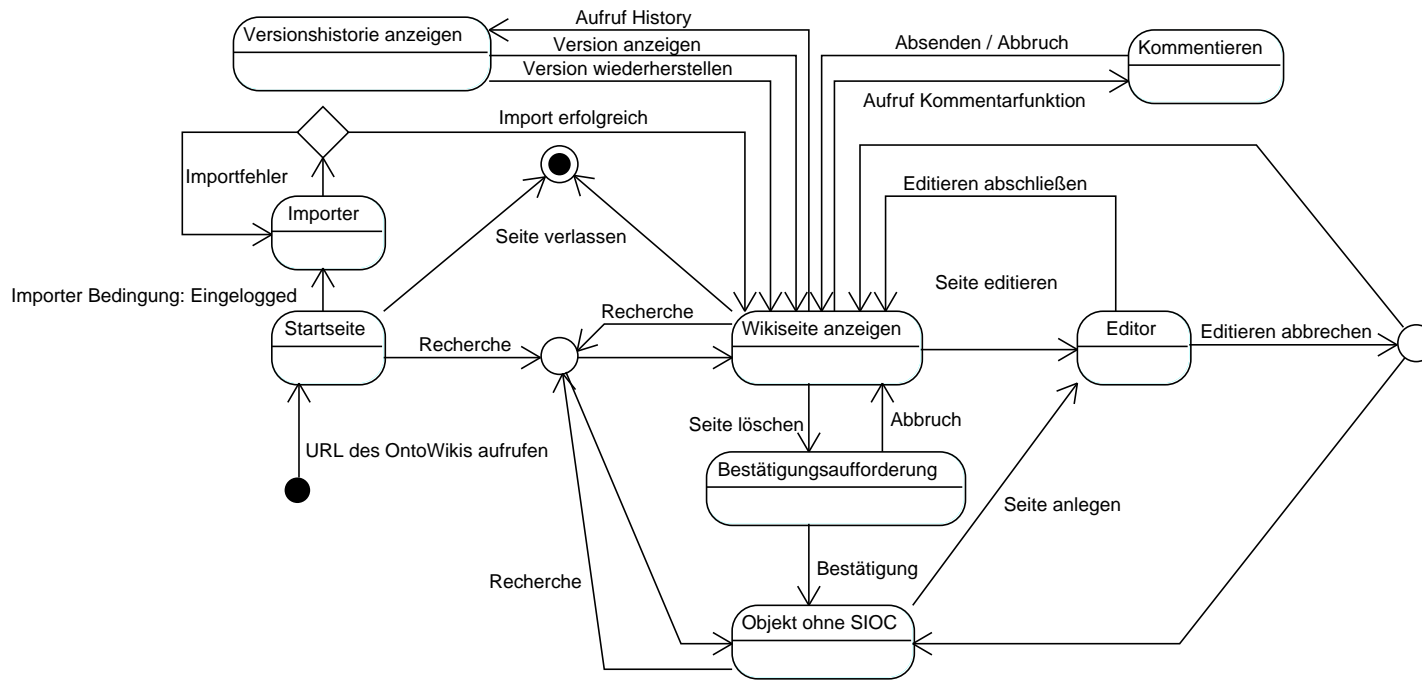


Abbildung 3: Zustandsdiagramm

## 4.2 Klassenübersicht

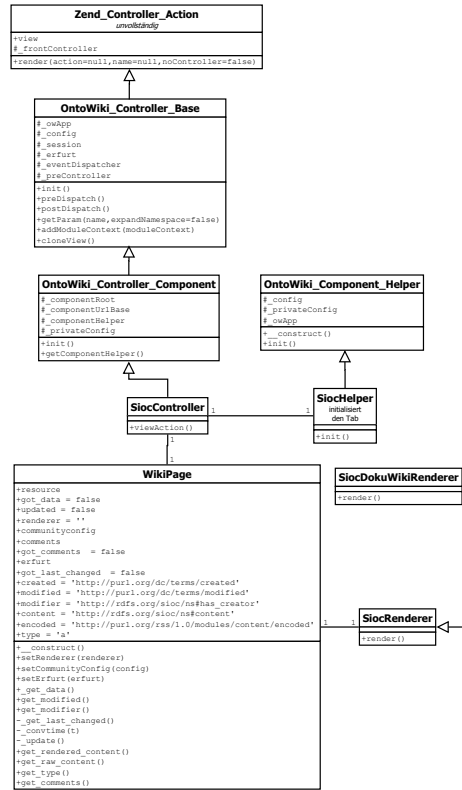


Abbildung 4: Klassendiagramm

### 4.3 Editor

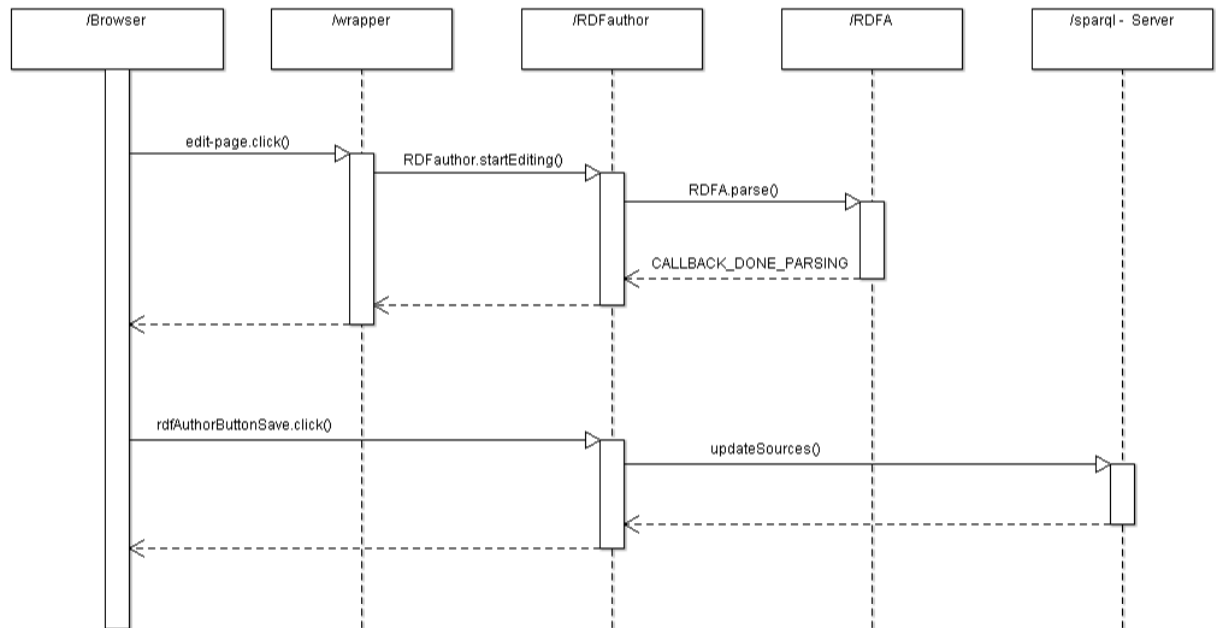


Abbildung 5: Sequenzdiagramm

Damit Wiki-Markup von Benutzern komfortabel editiert werden kann, wird der im OntoWiki verwendete Editor `RDFauthor` um ein zusätzliches Widget namens `EditorWidget` erweitert. Dieses wird dann mit Hilfe der Funktion `RDFauthor.registerWidget()` zu den bereits in `RDFauthor` vorhandenen Widgets hinzugefügt.

Für die konkrete Editierfunktion wird wie auch im Properties-Tab des OntoWikis `RDFauthor` verwendet. Dazu bindet die Wikiseite die Javascript Dateien ein, die den `RDFauthor` und das von uns implementierte `EditorWidget` bereitstellen. Dabei wird auch der “Editieren”-Button mit dem `RDFauthor` verknüpft.

Wird nun ein Editiervorgang gestartet, so traversiert `RDFauthor` den Baum unserer Wiki-seite, die neben der Darstellung als (X)HTML auch noch die Repräsentation unserer Da-



ten in RDFa enthält. Aus den über die RDFa Daten erhaltenen Informationen konstruiert nun der `RDFauthor` ein Editorfenster, wobei das `EditorWidget` für die entsprechenden Felder verwendet wird.

Wird der Editiervorgang positiv durch eine Betätigung des “Speichern”-Buttons beendet, so analysiert der `RDFauthor` die vorgenommenen Änderungen und überträgt diese in Form eines oder mehrerer SPARQL/Update Requests an den Server, der diese samt Versionshistorie in der Datenbank speichert. Damit ist der Prozess des Editierens abgeschlossen.