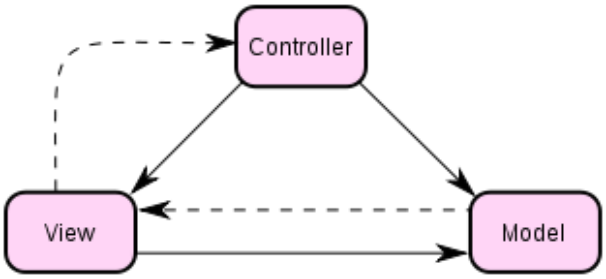
Entwurfsbeschreibung qTranslate

1. Allgemeines

Das Ziel des Projektes RDF2WP ist es, ein Plugin zu entwickeln, mit welchem sich semantische Daten (RDF) in ein WordPress-Blogeintrag einbetten lassen.
Im Rahmen der Fremdprojektanalyse sollte sich unser Team mit Wordpress und dessen Erweiterungsmöglichkeiten befassen. Diese Studie soll uns eine vertiefte Einsicht in das zu bearbeitende Praktikumsprojekt geben. Unsere Aufgabe war es, die Aspekte der Umsetzung von Model-View-Controller-Architektur im Wordpress sowie der Implementierung von eigener Sichten zu untersuchen. Darüber hinaus haben wir uns im Detail mit dem Wordpress-Plugin qTranslate beschäftigt.

2. Produktübersicht

Die Model-View-Controller-Architektur besteht aus den Einheiten Datenmodell (model), Präsentation (view) und Programmsteuerung (controller). Das Datenmodell enthält die darzustellenden Daten. Die Präsentationsschicht ist für die Darstellung der Daten aus dem Modell zuständig. Die Steuerung verwaltet Präsentationen und entscheidet welche Daten im Modell aufgrund Benutzeraktion geändert werden sollen.

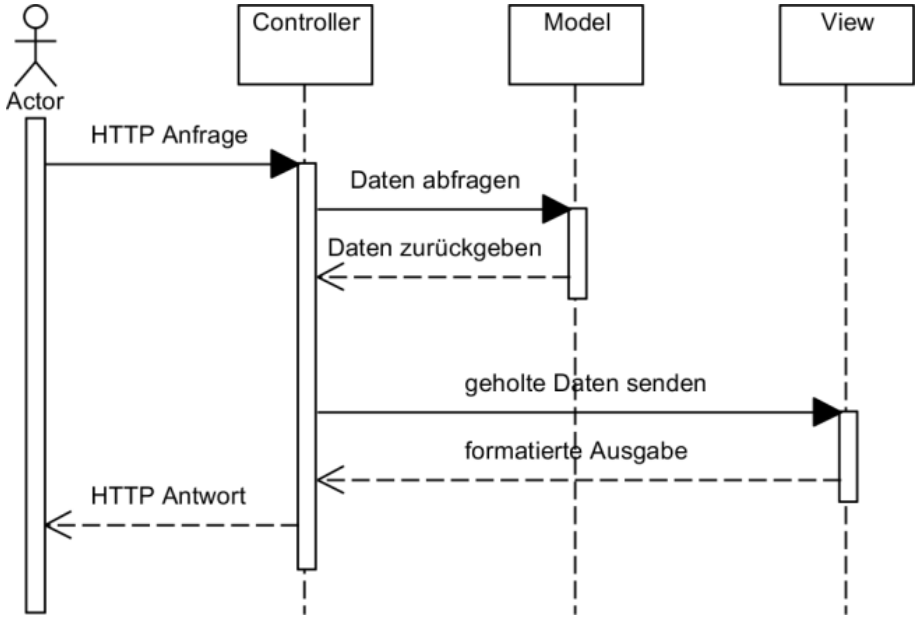


Im Allgemeinen kann die Wordpress-Architektur nicht als eine MVC-Architektur betrachtet werden. Allerdings ist es möglich, Wordpress-kompatible Plugins in der MVC-Architektur zu gestalten. Eine Möglichkeit stellt z.B. das WP-Plugin Tina MVC bereit. Mit diesem Plugin lassen sich andere Plugins erstellen, die auf dieser Architektur basieren.
Das im Rahmen dieser Studie untersuchte Plugin qTranslate bietet die Möglichkeit, einen Blogeintrag gleichzeitig in mehreren Sprachen zu verfassen. Die Sprachen, in denen gewöhnlich Einträge geschrieben werden, können bei der Wordpress unter *Settings→Languages* ausgewählt werden. Als Option kann gewählt werden, dass die Sprache des Browsers automatisch erkannt und der Inhalt in der entsprechenden Sprache angezeigt wird. In der Adressleiste erscheint dann der zweibuchtabige ISO-Sprachcode.

3. Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem

MVC in PHP

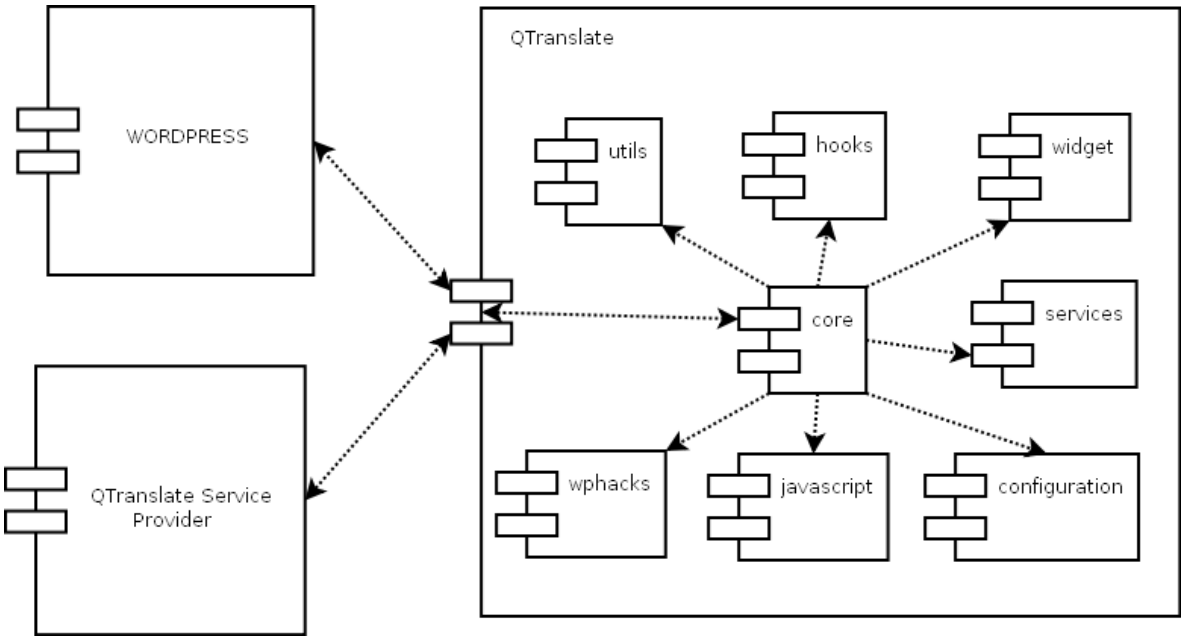
Das Prinzip von Model-View-Controller wird in drei Module aufgeteilt: Modell, Präsentation und Steuerung.
Das Modell (Model) ist verantwortlich fur das Verwalten der Daten; es beinhaltet und holt sich (meistens aus einer Datenbank) die Entitäten, die von der Applikation verwendet werden.
Die Präsentationsschicht (View) ist verantwortlich für das Anzeigen der von dem Modell bereitgestellten Daten in einem spezifischen Format.
Die Steuerung (Controller) ermöglicht die Zusammenarbeit zwischen dem Modell und den Präsentationsschichten. Die Steuerung erhalt eine Abfrage von dem Klienten, bringt das Modell dazu die abgefragten Operationen auszuführen und sendet die Daten an die Präsentationsschicht.



Struktur des qTranslate Plugins

Das qTranslate Plugin besteht aus acht Komponenten, die in je einer Datei gespeichert sind. Die Aufgaben der Komponenten sind:

core	initialisiert qTranslate
utils	enthält Funktionen zum Konvertieren von Strings
hooks	fügt Action-Hooks und Filter hinzu
widget	dient als Sprachauswahl
services	bietet Dienste zum Übersetzen von Texten
wphacks	behebt WP bugs und erstellt GUI Elemente in WP
javascript	stellt Javascript Funktion bereit
configuration	verwaltet qTranslate in WP



4. Grundsätzliche Struktur und Entwurfsprinzipien der einzelnen Pakete

Allgemeines

- Aufbau grundsätzlich prozedural nicht OOP (außer das Hinzufügen von Widgets)
- qTranslate ändert in den Versionen ab 2 nichts mehr an der WordPress-Datenbank, sondern verwendet ausschließlich die vorgesehenen Datenfelder, allen voran post_title und post_content in der Tabelle wp_posts.
- qTranslate verwendet für die Zuordnung der Inhalte zu den Sprachversion die PHP-Funktion Gettext. `<!--:de-->Hier erscheint nun der deutsche Text<!--:-->` Andere Sprachversionen, etwa Englisch werden direkt im Anschluss angefügt: `<!--:en-->Here we have the english text<!--:-->`

- für jede Sprache muss ein separater Text erstellt werden, auch wenn die Inhalte gleich sind
- Um die Erstellung unterschiedlicher Sprachversion komfortabel zu ermöglichen, teilt qTranslate das Editorfenster entsprechend auf. Die Titeleingabefelder werden für jede Sprache repliziert. Innerhalb des Haupteditorfensters kennzeichnen Tabs mit Angabe der jeweiligen Sprache den jeweiligen Eingabebereich.

MVC Module in PHP

Das Beispiel zeigt die grundsätzliche Struktur der einzelnen Pakete.
index.php
Erzeugt die Steuerungsklasse und startet ihre Funktionalität.

```
<?php
    include_once('BookController.php');
    $controller = new BookController();
    $controller->invoke();
?>
```

BookController.php
Die Steuerungsklasse empfangt HTTP-Request und erzeugt mit den durch die GET-Methode extrahierten Attributen ein neues Modell. Danach wird die Präsentationsschicht eingebunden.

```
<?
    php include_once('BookModel.php');
    class BookController
    {
        public function invoke()
        {
            $model = new BookModel($_GET['title'], $_GET['author']);
            include 'BookView.php';
        }
    }
?>
```

BookModel.php

Das Modell zum Buch beinhalten zwei Attribute: der Titel und der Author.

```
<?php
class BookModel
{
    public $title;
    public $author;
    public function __construct($title, $author)
    {
        $this->title = $title;
        $this->author = $author;
    }
}
?>
```

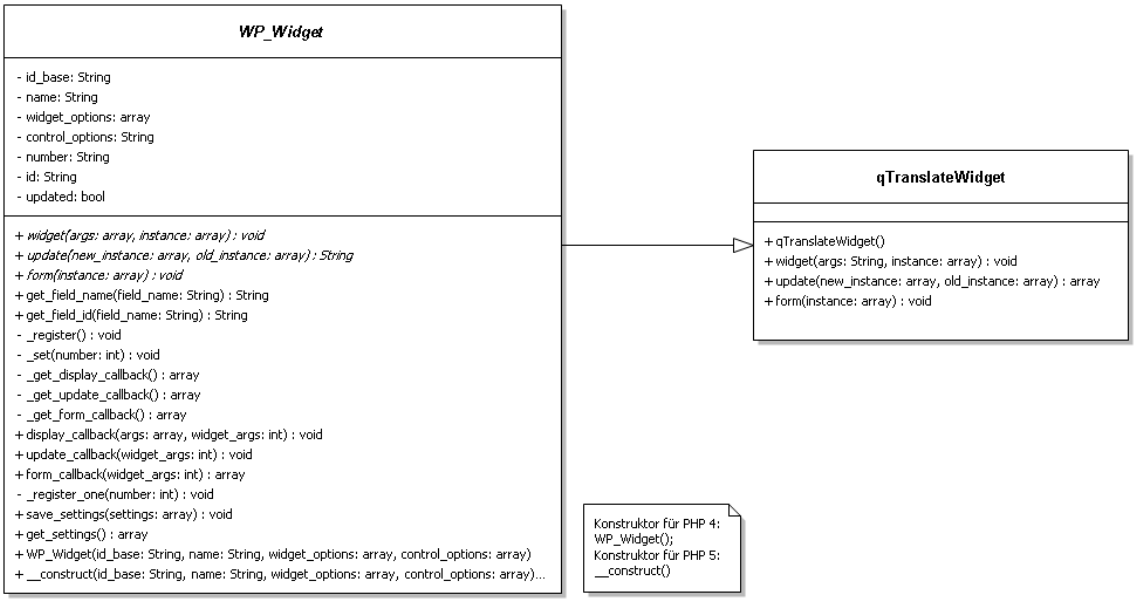
BookView.php

HTML-Ausschnitt mit dem erwarteten vordefinierten Modell zum Buch, um eigenen Inhalt zu vervollständigen.

```
<table>
    <tr>
        <th>Title</th>
        <td><?php echo $model->title; ?></td>
    </tr>
    <tr>
        <th>Author</th>
        <td><?php echo $model->author; ?></td>
    </tr>
</table>
```

qTranslate Widget

Das qTranslate Widget erlaubt es dem Nutzer eine Sprache für sein WordPress zu wählen. Im Admin Bereich kann hierbei gewählt werden, in welcher Weise die Sprachen und der Titel im Widget angezeigt werden sollen. Um ein Widget in WordPress zu erstellen, muss die Klasse `WP_Widget` erweitert werden. Dabei überschreibt die qTranslateWidget Klasse die drei Funktionen `widget(), update() und form()`.



Die Funktion `widget()` erhält über das assoziative Array `$instance` die im Widget Menü gesetzten Optionen. Die Bezeichner des Arrays sind „titel“, „hide-titel“ und „type“. „titel“ enthält den Titel des Widgets. Ist hier nichts angegeben so ist der Titel „Language“ (in der jeweiligen Sprache). „hide-titel“ entscheidet ob der Titel angezeigt werden soll und „type“ entscheidet wie die aktivierten Sprachen anzuzeigen sind. Die Funktion ist also für die Ausgabe des aktuellen Inhalts verantwortlich. Die Funktion `update()` sorgt dafür, dass geänderte Werte aus dem Widget Menü übernommen werden. „`$new_instance`“ enthält dabei die geänderten Werte und „`$old_instance`“ die alten Werte. Hier wird entschieden in welcher Weise neue Werte zu initialisieren sind und das neue Aussehen des Widgets über ein Array, welches die oben genannten Optionen enthält zurückgegeben. Im qTranslate Widget werden neue Werte direkt übernommen. Die Funktion `form()` definiert das Formular der verfügbaren Optionen im Widgets Menü. Der Parameter `$instance` enthält die aktuell gesetzten Optionen. Im qTranslate Widget werden vier radio-Button, ein check-Button und ein Text Feld definiert. Die Zuweisung des name- und id- Attributes der Input Felder wird durch die Funktion `get_field_name(<option>)` bzw. `get_field_id(<option>)` aus der Widget Klasse realisiert. Der Rückgabewert ist ein in WordPress einmaliger Identifizierer, der von der `update()` Funktion zum Speichern neuer Eingaben benötigt wird.

Komponenten

qTranslate.php

- Setzen der Konfiguration aus dem Settings-Menü, z.B. default language, browser language detection, im assoziativen Array `$q_config` (Global)
- Laden bzw. Einbinden aller benötigten Files (über `require_once()`)

qTranslate_configuration.php

- Definiere Admin Menü, wenn mehrere Sprachen aktiv sind
- Definition Language HTML Formular im Settings-Menü
- Auswerten des Formulars durch updaten der Werte in `$q_config`
- Validieren der Formulareingabe
- Eintrag der Sprachdaten der Posts in WP Datenbank
- Editieren und Löschen einzelner Sprachen
- Aktiviere und Deaktiviere Sprachen im Languages Menü
- Ordnen der Sprachen
- Erstellen Formular Language Management
  - General Settings
  - Advanced Settings
  - Services Settings
- Erstellen Tabelle zum Aktivieren und Deaktivieren von Sprachen (Menü Languages)

qtranslate_core.php

- Initialisiere qTranslate (`check QTRANS_INIT`)
- Setze alle Werte zurück mit `delete_option()`
- Setze Option für Cookie (`cookie_enabled`)
- Initialisiere JavaSkript Funktionen
- Set Cookie `\qtrans_admin_language"`
- Lade Angaben im Settings Menü

- Wenn nichts angegeben, lade defaults
- Lade gesetzte Werte in das `q_config()` Array
- Lade die Konfiguration in WP (`do_action()`)
- Speichere die Kofiguration
- Update der GETTEXT Datenbanken der verwendeten Sprachen (setzen der Update Zeit)
- DATE TIME FUNCTIONS

qtranslate_javascript.php

- Definieren der JavaSkript Funktionen
- Diese werden im Globalen Array `q_config` als Strings gespeichert, wie folgt:
- Aufteilung des Arrays in `q_config['js'][<JavaSkript Funktionsname>]`
- Einsatz von jQuery z.B. um Texteingabefelder zu laden und auszulesen
- Häufiger Zugriff auf HTML Elemente über deren Id-Attribut mit `document.getElementById()` z.B. um Wert (value) der `enabled_languages` zu erhalten: `document.getElementById('qtrans_tag_".`

qtranslate_hooks.php

- Realisieren der Settings, wie nicht übersetzte Posts oder Pages nicht anzeigen (Posts werden direkt aus WP-Datenbank abgefragt)
- Setzen von css für den Header
- Setze Einträge für Posts Tabelle im Posts Menü
- Hinzufügen von Action-Hooks (`add_action()`) und Filtern (`add_filter()`), time critical und non-time-critical
- Im Admin-Bereich werden einige Hooks nicht eingebunden

qtranslate_services.php

- Realisiert Übersetzungsfeatures durch qTranslate Service Provider
- Definieren von Konstanten
- Definiere Feld für Error Messages

- Hinzufügen von Hooks, z.B `save, init, load, config_hook`
- Verschlüsselungs- und Entschlüsselungsfunktionen
- Verbindung mit qTranslate Server (Socket)
- Sicherer Datenaustausch über openSSL Protokoll (Austausch des public key ...)
- `qTranslate_services` use professional human translation services
- Hinzufügen von HTML für die Weiterleitung auf die Homepage der Services
- Hinzufügen von Formularen für Service Settings
- Für die Übersetzung eines Posts wird dieser verschlüsselt (SSL) und an den qTranslate Service geschickt. Dieser übersetzt ihn und schickt die Übersetzung zurück.
- Submit mittels jQuery
- Liegt ein Text bereits in der Sprache vor in die er von einem Service übersetzt werden soll, so wird er überschrieben

qtranslate_utils.php

- Funktionen die den Status bzw. einzelne Settings wiedergeben, z.B. enabled Languages, AvailableLanguages, durch Auslesen von `q_config[]` (Getter)
- Datumsformate in String umwandeln
- Konvertieren von Datums- und Zeitformaten für andere Sprachen

qtranslate_widget.php

- Erweitern der WP Widget Klasse (`class qTranslateWidget extends WP_Widget ()`)
- Überschreiben von drei Funktionen, die für die Erstellung von Widgets unbedingt benötigt werden (`form(), widget(), update()`).
- Funktion `qTranslateWidget()` definiert qTanslate Plugin (Konstruktor)
- Sprachauswahl für Nutzer, die keine Widgets nutzen.
- Registrieren des Widgets in WP

qtranslate_wphacks.php

- Behebe WordPress bugs (fix wpautop bug)
- Überprüfe auf kompatible WP Version und Erstelle das Editiermenü zum Editieren von mehrsprachigen Posts (in der Variablen `$content`)
- Hinzufügen von JavaScript Code (zu *$content*), um den Admin Bereich zu ändern.
- Erstellen eines neuen Editors, Hinzufügen der Editor -Toolbar
- Hinzufügen von Tabs für die aktivierten Sprachen, HTML und Visual im Editor
- Verbergen des alten Text- und Titel-Editors
- Hinzufügen von neuen Titel-Editors für aktivierte Sprachen
- Hinzufügen des Inhalts beim Ändern bestehender Posts
- Hinzufügen von TextArea‘s mittels jQuery