

Entwurfsbeschreibung des Prototyps „Interaktiver Haushaltsrechner“

Verfasser:

Sebastian Zänker, Tobias Wieprich, Fabian Niehoff,
Janos Borst, Wolfgang Amann

4. April 2015

1 Allgemeines

Ziel des Projekts „Interaktiver Haushaltsrechner für die Stadt Leipzig“ (IHR) ist die Neugestaltung des Haushaltsrechners der Stadt Leipzig, aufbauend auf den Anforderungsanalysen der Gruppe „haushalt-14“. Der neue Haushaltsrechner soll eine übersichtlichere Darstellung des Finanzhaushalts beinhalten und eine adäquate Vorschlagfunktion mit Diskussionsforum zur Verfügung stellen.

Im Zuge dieses Projekts soll in dieser Phase zunächst ein Prototyp als Vorprojekt erstellt werden. Der Prototyp bildet die Grundlage für das weitere Vorgehen. Es soll die Datenschicht und die Programmierumgebung des geplanten Systems zum Ergebnis haben - das heißt die ersten zwei im Arbeitsplan definierten Arbeitspakete (Punkt 4.2 des Arbeitsplans) mit teilweiser Bearbeitung des dritten Schritts. Dazu gehören Drupal, Triplestore und eine Testseite.

2 Produktübersicht

Der Fokus des Vorprojekts ist die Funktionalität der Datenebene. Dazu soll zunächst eine Instanz von Virtuoso aufgesetzt werden (erreichbar unter <http://pcai042.informatik.uni-leipzig.de:1524/sparql>), auf der SPARQL-Abfragen an den Triplestore gestellt werden können. Anschließend wird eine Drupal-Instanz (erreichbar unter <http://pcai042.informatik.uni-leipzig.de/swp15-ihr/Drupal/>) so konfiguriert, dass dort mit entsprechenden SPARQL-Anfragen gewünschte Datensets in einem Kreisdiagramm visualisiert werden können.

Das Erstellen der Testseite wird - als Prototyp einer Webseite - zum Überprüfen der Funktionsfähigkeit der Arbeitsumgebung genutzt, soll aber auch als Grundstein für die weitere Arbeit gesehen werden. Durch die Erstellung des Prototyps soll so nicht nur das Know-How der grundsätzlichen Arbeitstechniken sicher gestellt, sondern auch einen Ansatzpunkt für die weitere Arbeit geliefert werden.

3 Grundsätzliche Struktur- und Entwurfsprinzipien

3.1 MVC-Modell

Da die Kernfunktion des Projekts auf einer festen und einheitlichen Basis von Datensätzen basiert, werden wir uns an das MVC-Architekturmodell halten. Dabei ist die Aufgabe des „Models“, die Daten aus einer RDF Datenbank zu lesen, sie in ihre Bestandteile zu zerlegen und sie dem View zugänglich zu machen. Das „View“ benutzt die Informationen, die es vom „Model“ erhält, um die Daten entsprechend zu visualisieren (z. B. in einem Kreisdiagramm). Daneben ist die zentrale Aufgabe des „Views“ die Verwaltung von Formularen und Dokumenten zur allgemeinen und übersichtlichen Informationspräsentation für die Benutzer.

4 Struktur- und Entwurfsprinzipien einzelner Pakete

4.1 Datenebene

Die Daten sind in Form von RDF zur Verfügung gestellt (bzw. wurden durch die Vorgruppe vom Excel-Format in einen RDF Data Cube transformiert). Diese Daten werden über eine SPARQL-Schnittstelle für die Diagrammdarstellung zur Verfügung gestellt.

4.2 Darstellungsebene

Das Aussehen der Website des interaktiven Haushaltsrechners der Stadt Leipzig soll sich stark an der vom Bundshaushaltsrechner halten.

Wie bei dem Bundshaushaltsrechner sollen die Daten in Form von Kreisdiagrammen angezeigt werden. Auf jeder anzeigbaren Seite existiert ein Button, der Benutzern dazu dient, Vorschläge einzureichen. Der Button soll später zur Anbindung an die Schnittstelle zur Partizipation verfügbar sein.

4.3 Forum

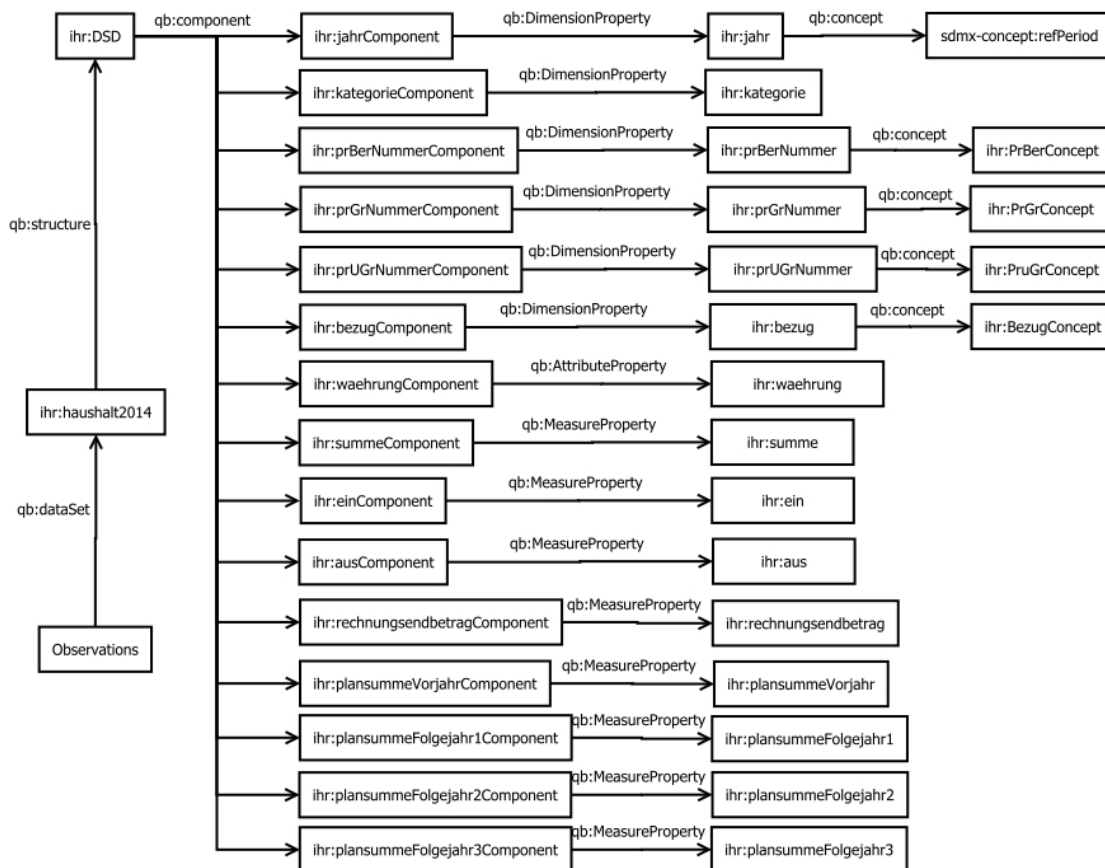
Das Forum besitzt eine strikt geregelte Hierarchie. Es gibt maximal eine geringe Anzahl von Administratoren, die alle Rechte besitzen (insbesondere die Rechte, alle anderen Mitglieder anderer Nutzergruppen auf- und absteigen zu lassen, sowie sie zu blockieren). Desweiteren ist es nur Administratoren erlaubt, Beiträge zu löschen. Als nächst tiefere Nutzergruppe gib es Moderatoren. Moderatoren sind von der Stadt zur Verfügung gestellte „Ordnungshüter“ und von Administratoren gewählte vertrauenswürdige normale Nutzer. Sie sorgen für allgemeine Ordnung, und können in speziellen Bereichen Threads erstellen. Sie sorgen für einen passenden Ton und können auch begrenzt normale Nutzer einschränken. Als letztes gibt es noch die normalen Nutzer. Sie nutzen das Forum für seinen gedachten Zweck: Diskussionen und Debatten über den Leipziger Stadthaushalt. Das Forum in sich selbst ist in mehrere Subforen unterteilt um eine bessere Spaltung der verschiedenen Bereiche des Haushalts zu ermöglichen.

4.4 Partizipation

Das Arbeitspaket „Partizipation“ umfasst im weiteren Sinne die Grundlage zur Mitarbeit der Leipziger Bürger. Wenn sie verbesserungswürdige Ausgaben oder Einnahmen sehen, können sie Vorschläge über ein Formular absenden. Diese Formulare sind auf die Ebene (Finanzbereich,

z.B. Ausgaben für Theater) zugeschnitten und ermöglichen so der Stadt Leipzig eine einfachere Verwaltung, Sortierung und Nutzung. Die Bürger können natürlich auch Vorschläge für gänzlich neuen Ein- und Ausgaben einreichen. Abgegebene Vorschläge erhalten einen eigenen Thread im Forum, in dem andere Bürger dann diesen Vorschlag kommentieren, verbessern und u. U. auch als unbrauchbar erklären können. Desweiteren können die Bürger im Forum Threads eröffnen und dort über aktuelle Themen debattieren und auch Vorschläge dort erstmal im vornherein schreiben und mit Hilfe der anderen Bürger diesen Vorschlag so verbessern, um ihn letzten Endes „einreichbar“ zu machen.

5 Datenmodell



Die vom Auftraggeber zur Verfügung gestellte Datengrundlage für unser Projekt besteht aus mehreren Excel-Dateien, die per dynamischer Listenausgabe aus SAP exportierte relevante Daten (derzeit noch aus dem Jahr 2014) enthalten. Im Vorprojekt wurde diese Datenbasis bereits schrittweise in das RDF-Data-Cube-Format übertragen. Die oben stehende Grafik zeigt die grundlegende Struktur unseres RDF-Cube-Modells.

6 Testkonzept

6.1 Komponententest

Jede Komponente, die erstellt wird, muss auf Funktionalität und Sicherheit getestet werden, damit die Funktionalität der Komponente gewährleistet wird. Als Werkzeug dient uns hierbei PHPUnit und soll vom jeden Teammitglied selber genutzt werden. Das Testen soll nach jeder Änderung oder Erweiterung ausgeführt werden. Fehler sollen in einem Testbericht festgehalten werden.

6.2 Integrationstest

Auch wenn jede Komponente bereits individuell getestet wurde, kann das Zusammenspiel fehlerhaft sein. Daher müssen die getesteten Komponenten noch auf Integrationsfähigkeit getestet werden. Durch das Zusammenführen von den Komponenten im Git-Repository kann man die Änderung auf das Gesamtsystem merken. Treten hier Fehler auf, können diese schnell behoben werden. Auch hier werden die Fehler in dem Testbericht festgehalten.

7 Glossar

Drupal ist ein freies (GPL) und kostenloses Content-Management-Framework auf der Basis von PHP. Es besteht dabei aus einem Core, der aus anderen Content-Management-Systemen bekannte Funktionen liefert und Modulen, mit denen man zusätzliche Funktionen implementieren kann.

Forum Ein Internetforum dient als virtueller Platz zum Austausch von Gedanken, Meinungen und Erfahrungen. Ein großer Vorteil dieser Kommunikationsform ist deren Asynchronität, d. h., dass Diskussionsteilnehmern ermöglicht wird, zeitversetzt zu antworten. Üblicherweise werden zur Unterhaltung sogenannte *Threads* erstellt (= Themen), in denen einzelne Diskussionsstränge zu einem übergeordnetem Thema gebündelt sind.

Model-View-Controller Als eines der beliebtesten Architekturmodelle der objektorientierten Programmierung ermöglicht das MVC-Muster eine klare Rollenteilung im Programm in die drei Einheiten Datenmodell (*model*), Präsentation (*view*) und Programmsteuerung (*controller*). Der „Observer“ (controller) nimmt hierbei eine zentrale Rolle ein – er ermöglicht es, auf Veränderungen in den zu beobachten Objekten zu reagieren. Diese werden durch das „Model“ und den „View“ beschrieben. Das „Model“ bietet Zugang zu zentralen Daten des Programms und Funktionen, um diese abzurufen oder zu verändern. Diese Daten werden dann in dieser Architektur vom „View“ dem Benutzer zugänglich gemacht. Auf Benutzereingaben reagiert von hier an wieder der „Observer“ und kann so die Daten abhängig vom Programmverlauf ändern.

PHPUnit ist ein freies (BSD-Lizenz) Framework zum (automatisierten) Testen von in PHP geschriebenen Anwendungen. Es dient somit zur frühen Fehlererkennung und Qualitätssicherung von Softwareprojekten.

RDF Das Resource Description Framework als Konzept erweitert die Möglichkeit des Webs, Inhalte miteinander zu verbinden. Hierbei wird versucht, nach semantischen Zusammenhängen zu ordnen. Die Aussagen die die eigentlichen Daten in RDF sind, betreffen sogenannte Ressourcen.

Diese Ressourcen sind eindeutig bezeichnete Dinge der Welt, die durch Tripel oder Graphen in eine Relation mit anderen Ressourcen gebracht werden. Nicht nur durch das mit RDF im Zusammenhang benutzten Vokabular ähnelt dieses Modell stark an die klassischen Modelle wie UML-Diagramme oder Entity-Relationship-Modelle, auch die Modellierungsweisen sind stark miteinander verknüpft, wodurch RDF als Grundansatz des Semantischen Webs an Bedeutung gewinnt.

RDF Data Cube ist ein spezielles Vokabular zur Beschreibung von Daten im RDF-Standard. Durch das Bereitstellen von einer beliebigen Anzahl von Dimensionen ist das Format vor allem zur Beschreibung von statistischen bzw. messbaren Daten geeignet.

SPARQL (rekursives Akronym für *SPARQL Protocol And RDF Query Language*) ist eine Abfragesprache für RDF, d. h. eine semantische Abfragesprache für Datenbanken, die dazu in der Lage ist, Daten, die im RDF-Format gespeichert sind, aufzurufen und zu manipulieren.

Triplestore Ein Triplestore ist eine Datenbank zur Speicherung und Abfrage von Tripeln durch semantische Abfragen. Ein Daten-Tripel besteht aus Subjekt, Prädikat und Objekt, z.B. „Franz kennt Fritz“.

8 Quellenverzeichnis

Letzter Aufruf von allen Internet-Quellen am 4. April 2015.

- <http://de.wikipedia.org/wiki/Drupal>
- <http://de.wikipedia.org/wiki/Internetforum>
- http://de.wikipedia.org/wiki/Model_View_Controller
- <http://home.edvsz.fh-osnabrueck.de/skleuker/CSI/Werkzeuge/PHP/PHPUnit.pdf>
- http://en.wikipedia.org/wiki/Resource_Description_Framework
- <http://www.w3.org/TR/vocab-data-cube/>
- <http://en.wikipedia.org/wiki/SPARQL>
- <http://de.wikipedia.org/wiki/Graphdatenbank>