

Entwurfsbeschreibung des Gesamtprojekts „Interaktiver Haushaltsrechner“

Verfasser:

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

18. Mai 2015

1 Allgemeines

Im Rahmen der Beteiligungsprozesse wurde durch die Leipziger Agenda21-Gruppe in Zusammenarbeit mit der Stadtverwaltung über mehrere Jahre (seit 2007) ein Haushaltsrechner angeboten, über den Bürger*innen im Zuge der Verabschiedung des jeweiligen Haushalts durch den Stadtrat interaktiv Vorschläge zur Änderung der Haushaltsplanung einbringen konnten. Diese Möglichkeit wurde zuletzt im Jahr 2012 für den Haushalt 2013 in überarbeiteter Form angeboten. Für den Haushalt 2014 wurde diese Beteiligungsoption aufgegeben. 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 des Vorprojektes wurde die ersten beiden Arbeitspakete erledigt. Dies beinhaltet das Aufsetzen der Drupalumgebung auf den Server [<http://pcai042.informatik.uni-leipzig.de/swp15-ihr/Drupal/>], sowie das Auslesen von Daten aus den RDF auf eine kleine Testseite [<http://pcai042.informatik.uni-leipzig.de/swp15-ihr/Drupal/>]. In der abschließenden Phase des Projekts soll der zuvor erstellte Prototyp erweitert und mit allen in der Anforderungsanalyse geforderten Funktionen ausgestattet werden. Da die Arbeitsschritte zur Informationsebene schon größtenteils abgeschlossen wurden, beschränkt sich die kommende Arbeit vor allem auf Designkonzept, Testphasen und die Umsetzung der Partizipationsebene (d. h. Arbeitspakete [3] bis [7]).

2 Produktübersicht

Während des Vorprojektes wurde die Funktionalität der Datenebene implementiert. Dabei wurde eine Instanz von Virtuoso aufgesetzt, auf der SPARQL-Abfragen an den Tripelstore gestellt werden. Zudem wurde auf dem Praktikumsserver eine Instanz von Drupal aufgesetzt, welche der Grundstein für die spätere Webseite sein soll. Als nächstes wird die Informationsebene

implementiert, welche die Grundzüge der späteren Webseite enthält. Die Ebene dient dazu, die Daten zu visualisieren. Dabei werden Feinheiten wie das Mouseover-Tool und ähnliches zur besseren Bedienung eingebaut. Als letzter großer Abschnitt wird ein Forum eingebaut, welches die zugehörigen Usergruppen besitzt.

3 Grundsätzliche Struktur- und Entwurfsprinzipien

3.1 Barrierefreiheit

Die gesamte Website soll gemäß BITV 2.0 barrierefrei gestaltet sein. Bei der Wahl der Farben muss also auf die üblichen Farbschwächen / Farbenblindheit geachtet werden. Beim Aufbau der Website sollte darauf geachtet werden, dass Programme und Techniken, die Webseiten für blinde Personen aufbereiten, möglichst wenig behindert werden. Dies ermöglicht die Partizipation für alle Bürger*innen und wirkt nicht ausgrenzend, was für öffentliche Institutionen und für einen demokratischen Prozess notwendig ist.

3.2 MVC-Architekturmodell

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“ bei der Datenvisualisierung, die Daten aus einer RDF Datenbank zu lesen, sie in ihre Bestandteile zu zerlegen und sie dem View zugänglich zu machen. Dazu wurde von uns die BorderCloud SPARQL 1.1 PHP Bibliothek genutzt, nachdem vorige Versuche mit Arc2 an der fehlenden SPARQL 1.1 Unterstützung scheiterten. 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. Der „Controller“ reagiert nach Signalübermittlung des Views und führt die entsprechenden - d. h. vom Nutzer gewünschten - Anfragen aus.

3.3 PAC-Architekturmodell

Die Verwaltung des Forums und der Vorschläge wird komplett über Drupal-interne Funktionen abgewickelt. Das bedeutet, dass hier - anders als den von uns eigens programmierten Inhalten - das PAC-Architekturmodell („Presentation-abstraction-control“) zum Einsatz kommt.

3.4 Semantische Technologien

Der Haushaltsrechner verwendet auf der Datenebene die semantische Technologie RDF in Form eines RDF Data Cubes. Das bedeutet, dass die in unserer Datenbank dargestellten Ressourcen eindeutig bezeichnete Dinge der Welt sind, die durch Darstellung in Tripel in eine Relation mit anderen Ressourcen gebracht werden.

3.5 Datensicherheit

Da bei der Registrierung von Nutzern Klarnamen verlangt werden, ist es nötig, eine ausreichende Datensicherheit zu gewährleisten - diese wird durch das Nutzerhandling von Drupal auch vollständig beachtet. Auch Passwörter werden nur verschlüsselt gespeichert.

4 Struktur- und Entwurfsprinzipien einzelner Pakete

4.1 Partizipationspaket

4.1.1 Forum

Die erste zentrale Komponente der partizipativen Ebene des „Interaktiven Haushaltsrechners“ ist das Forum, welches sowohl für allgemeine Diskussionen, als auch zur Ausarbeitung oder Diskussion von Vorschlägen verwendet werden kann.

Das Forum wird durch die Drupal-eigenen Module *Forum* und *Advanced Forum* implementiert. Die Nutzerverwaltung wird so von Drupal selbst übernommen und muss nicht extern verwaltet werden. Weiterhin ist es so einfacher für Informationsebene und Partizipationsebene ein einheitliches Design zu erstellen, da beiden dasselbe Drupal Theme zu Grunde gelegt werden kann.

Es ist wie folgt zu strukturieren:

- Das Forum muss moderiert sein.
Hierzu werden verschiedene Userrollen im Forum benötigt. Diese werden direkt von Drupal verwaltet und mit entsprechenden Rechten versehen.
- Es gibt bei der Registrierung einen Klarnamenzwang. Dieser Klarname wird nicht öffentlich angezeigt.
- Es existiert ein Melde-Button zum Melden von Beiträgen.
Der Melde-Button wird durch die Drupal-Module *Flag* und *Flag Abuse* verwirklicht.
- Diskussionen zu einem konkreten Vorschlag finden sowohl unter dem Vorschlag, als auch im Forum statt. Ein Kommentar unter dem Vorschlag wird jedoch im Forum angezeigt und umgekehrt.
- Die Vorschläge und Kommentare sollen bewertet werden können. Dazu kommt das Drupal-Modul *Rate* zum Einsatz.
- Der Aufbau vom Forum sollte sich an folgenden Bereichen orientieren:
 1. Allgemeine Diskussionen
 2. Vorschläge (Unterteilung in die Kategorien, verknüpft mit den Vorschlägen, Diskussion / Fragen im Forum, liken etc beim Vorschlag)
 3. Konkrete Themen zum Diskutieren (z.B. „Wie gehen wir mit Investitionen in Leipzig in Zukunft um?“)

4.1.2 Vorschläge

Die zweite zentrale Komponente des Partizipationspakets sollte sowohl bei der jeweiligen Kategorie im Informationsbereich zur Verfügung stehen, als auch in einem extra Formular. Nutzer sollen so in der Lage sein, Vorschläge zur Verbesserung des städtischen Haushalts einreichen zu können. Dazu soll es in der Informationsebene zu jeder Hierarchieebene des Haushalts einen Button geben der direkt auf ein Formular verweist über das ein Thread im Forum erstellt wird. Jeder Vorschlag wird in eine Kategorie eingeteilt, die der höchsten Abstraktionsebene entspricht. Die Kategorien, welche keine Möglichkeit zur Mitbestimmung liefern, sollten auch

nicht auswählbar sein. Es sollte die Möglichkeit bestehen, Vorschläge zu bewerten und zu kommentieren. Die Vorschläge und die Diskussion dazu sind mit dem Forum verbunden.

4.2 Informationspaket

Die Arbeiten zum Informationspaket wurden wie oben angemerkt größtenteils bereits in der Prototyp-Phase abgeschlossen, wodurch sich im Entwurf des Gesamtprojekts die Hauptarbeit auf die Verbesserung der Nutzerfreundlichkeit beschränkt.

4.2.1 Abstraktionsebenen

Die Präsentation der Daten sollte über verschiedene Abstraktionsebenen möglich sein. Die Abstraktionsebenen haben den Zweck, die Menge an Daten übersichtlich zu präsentieren und ein intuitives Verständnis sowohl der Größenverhältnisse als auch der Verteilung der Ausgaben und Einnahmen zu gewinnen. Die Abstraktionsebenen orientieren sich hierbei an den Hierarchien aus dem Haushaltsplan der Stadt Leipzig. Hierbei sollten möglichst alle Hierarchieebenen bis zum kleinsten Element verfügbar sein.

4.2.2 Designkonzept

Das Aussehen der Website des interaktiven Haushaltsrechners der Stadt Leipzig soll sich stark an der vom Bundeshaushaltsrechner halten. Wie beim Bundeshaushaltsrechner werden die Daten in Form von Kreisdiagrammen angezeigt. Zusätzlich werden Produktbereiche, die kleine Beträge aufweisen (im Diagramm $\leq 2\%$) zu einem Segment *Sonstiges* zusammengefasst. Dieses soll in einer Tabelle unterhalb des Diagramms aufgeschlüsselt dargestellt werden. Grundlage für die Implementation des Designs bildet das *Bootstrap*-Framework (<http://getbootstrap.com/>) bzw. dessen Portierung für Drupal 7 (<https://www.drupal.org/project/bootstrap>).

5 Datenmodell

Da die Datenkonvertierung bereits in der Prototyp-Phase abgeschlossen wurde, gibt es beim Datenmodell keinerlei Änderungen gegenüber des Prototyps. 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 (per SPARQL-Abfragen erreichbar unter <http://pcai042.informatik.uni-leipzig.de:1524/sparql>). In der Entwurfsbeschreibung des Prototyps gibt es eine genauere Darstellung des Datenmodells.

6 Testkonzept

Die Komponententests und Integrationstests werden größtenteils in der Implementierungsphase in den jeweiligen Gruppen ausgeführt. Die Ergebnisse der Tests wird in einem Testbericht für jedes Releasebündel zur Verfügung gestellt. Zum Abschluss der Implementierungsphase sind Testphasen für abschließende Integrationstests und Systemtests angedacht.

6.1 Komponententest

Jede Komponente, die erstellt wird, muss auf Funktionalität, Sicherheit etc. getestet werden, damit die Funktionalität der Komponente gewährleistet wird. Als Werkzeug dient 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 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, kann man diese schnell behoben werden. Auch hier werden die Fehler in dem Testbericht festgehalten.

6.3 Systemtest

In diesem Testteil wird nicht mehr aus Sicht der Entwickler getestet, sondern aus Sicht der User bzw. des Kunden. Dabei muss das System in der späteren Anwendungsumgebung getestet werden. Hierbei wird das Testtool Selenium genutzt, damit man die spätere Anwendungen durch User bzw. den Kunden simulieren kann. Somit können leicht Fehler in dem Programm gefunden und behoben werden. Fehler werden ebenfalls im Testbericht festgehalten.

6.4 Abnahmetest

Der Abnahmetest erfolgt mit den Kunden. Dabei wird das fertige Produkt in der Umgebung, welche ungefähr der späteren Umgebung ähnelt, auf die von Kunden vorgegebenen Anforderungen getestet. Sollten hierbei keine Probleme auftreten, kann das fertige Produkt den Kunden übergeben werden.

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 beobachteten 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.

Presentation-Abstraction-Control ist ein Architekturmodell, das MVC ähnelt. Es teilt ein System in die Komponenten „Presentation“, „Abstraction“ und „Control“ ein. „Presentation“ ist dabei für die Darstellung der Daten zuständig, „Abstraction“ für die Verwaltung der Daten und „Control“ für Flusskontrolle und Kommunikation zwischen den beiden anderen Komponenten.

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.

Selenium ist ein Tool für das automatisierte Testen von Webanwendungen.

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“.

BorderCloud SPARQL 1.1 PHP Bibliothek Eine Programmierbibliothek für PHP, die Funktionen zum Senden von Anfragen und Verwalten von Antworten an einen Tripelstore-Endpoint zur Verfügung stellt.

Bootstrap-Framework Open-Source Framework für HTML, CSS und JavaScript zur Erstellung von Webseiten.

8 Quellenverzeichnis

Letzter Aufruf von allen Internet-Quellen am 18. Mai 2015.

- <http://de.wikipedia.org/wiki/Drupal>
- <http://de.wikipedia.org/wiki/Internetforum>
- http://de.wikipedia.org/wiki/Model_View_Controller
- <http://de.wikipedia.org/wiki/Presentation-abstraction-control>
- <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>