A.Ruckelshausen et al. (Hrsg.): Digitale Marktplätze und Plattformen,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 155

Webservices auf heterogenen Datenbeständen – Methoden der Umsetzung am Beispiel der KTBL-Planungsdaten

Daniel Martini[1]

Abstract: Derzeitige Methoden der Bereitstellung von Webdiensten sehen in der Regel vor, dass für die auszuliefernden Objekte mit ihren Eigenschaften jeweils spezifischer Programmcode geschrieben wird. Die Bereitstellung von Daten aus Beständen, in denen komplexe Beziehungen von Daten und eine Vielzahl von Objekttypen mit jeweils unterschiedlichen Eigenschaften vorliegen, ist daher relativ aufwändig. Im vorliegenden Artikel wird ein alternativer Ansatz beschrieben, der Technologien des Semantic Web ausnutzt. Datenbestände beschreiben sich damit weitgehend selbst, sodass ein Dienst eigenständig darüber "reflektieren" und Informationen, die für die Bereitstellung notwendig sind, selbst extrahieren kann.

Keywords: Linked Open Data, Semantic Web.

1 Einleitung

Das KTBL bietet seit vielen Jahren Planungsdaten für die Landwirtschaft an. Diese dienen beispielsweise dazu, größere Investitionen – z. B. in Maschinen oder Anlagen – vorzubereiten oder Entscheidungen zum Produktionsprogramm zu unterstützen. Die hierfür notwendigen Datensätze sind – wie viele weitere landwirtschaftliche Datensätze – von Heterogenität geprägt. Eine Vielzahl von Objekttypen mit ihren jeweils unterschiedlichen Attributen stehen in landwirtschaftlichen Produktionsverfahren miteinander in Beziehung: Maschinen, Betriebsmittel, Prozessschritte usw. mit ihren Kennzahlen und Eigenschaften wie z. B. Reparaturkosten, Anschaffungspreise und Verbräuche. Eine Bereitstellung in Webservices aus vorhandenen Datenbanken über gängige Methoden des objekt-relationalen Mappings und der festen Abbildung o. g. Objekttypen im Programmcode der Dienste ist daher aufwändig und bei Änderungen an Datenbankstrukturen wartungsintensiv. Anstatt dessen sind Software-Komponenten notwendig, die Flexibilität bieten und sich möglichst weitgehend selbständig auf zugrundeliegende Daten einstellen.

Ziel der beschriebenen Arbeiten war daher die Entwicklung einer modularen Infrastruktur, die die Veröffentlichung von Daten über Webdienste möglichst unabhängig von zugrundeliegenden relationalen Strukturen unterstützt. Webservice-Endpoints sollten nur noch über eine möglichst einfache Abfragevorlage konfiguriert werden, ohne neuen Programmcode erstellen zu müssen. Die Bereitstellung sollte dabei gängigen Standards folgen und Linked-Data-Spezifikationen des W3C – z. B. die Linked Data Platform (LDP [Sp15]) und die Bereitstellung im RDF-Format Turtle [PC14] und als JSON-LD – erfüllen. Zudem

[1] Kuratorium für Technik und Bauwesen in der Landwirtschaft e. V., Datenbanken und Wissenstechnologien, Bartningstraße 49, 64289 Darmstadt, d.martini@ktbl.de

cbe doi:18.18420/provided-by-editor-02

sollte eine Auslieferung von Daten für Webbrowser im HTML-Format mit frei konfigurierbarer Darstellung unter Nutzung von HTML-Vorlagen mit Unterstützung von Mehrsprachigkeit möglich sein.

2 Material und Methoden

Die zugrundeliegende Gesamtarchitektur wurde bereits in [Ma15] beschrieben: Daten aus der relationalen Datenbank des KTBL werden über das Werkzeug D2RQ in RDF-Triples – eine graphenorientierte Darstellungsweise – umgewandelt. Hierfür muss lediglich ein Mapping erstellt werden, das Spalten und Einträge in relationalen Tabellen den Klassen, und Eigenschaften eines RDF-Vokabulars zuordnet. Letzteres definiert in maschinenlesbarer Form die Semantik der Dateninhalte. Im Zusammenspiel aus Daten und Vokabular erhält man so sich weitgehend selbst beschreibende Datenstrukturen. Die so konvertierten Daten werden in den Triple Store Jena Fuseki, der Abfragen in der Sprache SPARQL unterstützt, geladen. In der anfänglichen Umsetzung einer Linked-Data-Infrastruktur am KTBL erfolgte anschließend die Bereitstellung dieser Daten per HTTP über den Epimorphics Linked Data API-Server (ELDA).

In Bezug auf die Erkennung der Spracheinstellung des Nutzers und Auslieferung der entsprechenden Texte, Möglichkeit des Einsatzes von Vorlagen und Unterstützung verschiedener Datenformate weist dieser jedoch Limitationen auf: So unterstützt dieser die hierfür notwendige, sogenannte Content Negotiation nicht, zudem kann nur eine einzige HTML-Vorlage für eine gesamte Serverinstanz genutzt werden. Bei unterschiedlichen Anforderungen hinsichtlich der Darstellung der Dateninhalte je nach Objekttyp wird diese einzelne Vorlage daher sehr schnell recht komplex.

Deshalb wurde eine Serverkomponente in der Programmiersprache Go geschrieben, die die gegebenen Anforderungen erfüllt. Diese enthält keinen datenspezifischen Programmcode und erfordert nur noch die Konfiguration der Endpoints über die Angabe der jeweiligen SPARQL-Abfrage und HTML-Vorlage. Bei Eintreffen einer HTTP-Anfrage werden freie Variablen in der SPARQL-Abfrage – in der Regel die URL eines abgefragen Objekts wie z. B. einer Maschine – entsprechend befüllt und diese dann an den Triple Store durchgereicht. Der zurückgelieferte Datengraph wird anschließend eigenständig rekursiv von der Serverkomponente durchlaufen und dabei die notwendige Information zur Serialisierung in verschiedenen Formaten ermittelt. Dabei werden sowohl Ressourceninstanzen als auch Eigenschaften in dieselbe Struktur überführt, die die folgenden Felder enthält: Verweise auf Eltern- und Kindknoten über die alle im Datengraphen enthaltenen, auszuliefernden Knoten miteinander verknüpft sind; Typ (Ressource oder Eigenschaft) des vorliegenden Knotens; menschenlesbares Label; zugeordnete URL; Werte (nur für RDF-Literale) und der Verweis auf eine Einheit, sofern es sich um eine Eigenschaft und physikalische Größe handelt. Label werden je nach vom Browser übermittelter Spracheinstellung des Nutzers aus einem im Arbeitsspeicher gehaltenen assoziativen Datenfeld befüllt. Dabei kommt der Umsetzung eines der Grundprinzipien des Semantic Web zu Gute, dass

jede Ressource und Eigenschaft eine URI als eindeutigen Bezeichner zugewiesen bekommen. Diese können daher als Schlüsselfeld des assoziativen Datenfeldes genutzt werden, anhand dessen die richtigen Labels nachgeschlagen werden.

Die so befüllte Datenstruktur wird der Template-Engine übergeben, die die für die jeweilige URL der enthaltenen Ressourcen konfigurierte sowie für die übergebene Spracheinstellung spezifische HTML-Vorlage befüllt. Zum Einsatz kommt dabei die in der Go-Standardbibliothek enthaltene Template-Engine.

3 Ergebnisse und Diskussion

Gegenüber der derzeit gängigen Vorgehensweise Webservices aufzusetzen – in der Regel als ReSTful Services mit im Quellcode des Dienstes fest definierten Datenstrukturen für jedes ausgelieferte Objekt – weist die Umsetzung mit Technologien des Semantic Web eine Reihe von Vorteilen auf. So muss kein auf die Inhalte der anzubindenden Datenbank abgestimmter Quellcode mehr geschrieben werden. Es werden nur noch das Mapping von relationaler Datenbank zu Triples sowie Abfragen auf einen RDF-Triple-Store und HTML-Vorlagen für die Darstellung erstellt. Heterogene Datenbestände können so mit überschaubarem Aufwand für Webdienste aufgeschlossen werden. Die präsentierte Vorgehensweise kann daher als Modell für Webservice-Schnittstellen im landwirtschaftlichen Bereich dienen, bei denen eine Vielzahl von Objekten und komplexe Relationen an der Tagesordnung sind. In Go war für die Implementation relativ wenig Programmcode nötig. Die rekursive Funktion, die die oben beschriebene Datenstruktur der einzelnen Knoten befüllt, konnte beispielsweise in 40 Zeilen Programmcode realisiert werden. Die Antwortzeiten des Dienstes betragen bei schneller Netzanbindung nur wenige Millisekunden, auch dies ist eine deutliche Verbesserung gegenüber der ursprünglichen ELDA-basierten Umsetzung, die in der Regel um mindestens Faktor 10 höhere Antwortzeiten aufwies. Die Textlabel-Auslieferung direkt aus dem Arbeitsspeicher trägt hierzu bei und machte bei dem gegebenen Datenbestand keine Probleme. Es wird aktuell noch getestet, bis zu welcher Anzahl Label dieses Vorgehen bei gegebener Speichergröße noch gut funktioniert. Das assoziative Datenfeld wird derzeit einmalig jeweils mit dem Start des Servers befüllt. Ein Nachteil dieses Ansatzes ist, dass sich stetig ändernde Daten somit nicht gut bereitgestellt werden können. Für die relativ statischen Datenbestände des KTBL, die üblicherweise nur wenige Male im Jahr aktualisiert und neu publiziert werden, hat sich die oben beschriebene Vorgehensweise jedoch bewährt. In der vorliegenden Form der Umsetzung ist das System daher generell eher für sich wenig ändernde Stammdaten geeignet.

4 Ausblick

Derzeit werden im Rahmen des PAMrobust-Projektes Dienste aus dem Vorgängerprojekt PAM [Sc16] auf die beschriebene Infrastruktur migriert. Inhaltlich handelt es sich hierbei

um Daten zu Pflanzenschutzmitteln, insbesondere Mittel und deren zugeordnete Anwendungen sowie Wirkstoffe und geltende Abstandsauflagen. Ziel dabei ist es, den Datenbereitstellungsprozess zu vereinfachen. Die relativ umfassende, spezifische Programmierung des Webdienstes aus PAM wird hierbei verworfen und ersetzt durch eine wesentlich kürzere und einfacher zu pflegende Mappingdefinition und Server-Konfiguration. Zu Gute kommt hierbei der Aspekt, dass sämtliche oben beschriebenen Komponenten generisch sind und keinen spezifisch auf Fachinhalte abgestimmten Code enthalten. Gezeigt werden soll dabei außerdem, dass Abfragen, die mit relationalen Ansätzen nur sehr umständlich durchzuführen sind, wie z. B. der Einschluss von Mitteln, die für Oberklassen einer bestimmten Kultur (Bsp.: Apfel -> Kernobst) zugelassen sind, auf diese Weise relativ einfach durchzuführen sind.

Aktuell wird nur die Auslieferung von statischen Datenbankinhalten des KTBL unterstützt. Derzeit werden Konzepte erarbeitet, wie auch Dienste für dynamisch kalkulierte Planungsdaten von Vorteilen der Grundprinzipien des Semantic Web – beispielsweise der semantischen Beschreibung der Ressourcen und Eigenschaften und dem Vorhalten weltweit eindeutiger Bezeichner hierfür – profitieren können. Zudem ist geplant, das Nachladen von Textlabeln, die später hinzugekommen sind, in den im Arbeitsspeicher gehaltenen Bestand an Labeln zu ermöglichen.

Literatur

[Ma15] Martini, D.; Mietzsch, E.; Herzig, D.; Ladwig, G.: KTBL-Planungsdaten auf dem Weg in die Zukunft – Bereitstellung über Linked Open Data. In (Ruckelshausen, A., Schwarz, H.-P., Theuvsen, B., Hrsg.): Informatik in der Land-, Forst- und Ernährungswirtschaft 2015, Geisenheim. GI-Edition Lecture Notes in Informatics, Bonn, S. 105-108, 2015.

[PC14] Prud'hommeaux, E.; Carothers, G. (Hrsg.): RDF 1.1 Turtle – Terse RDF Triple Language. W3C Recommendation 25 February 2014. https://www.w3.org/TR/turtle/ (aufgerufen am 27.11.2017).

[Sc16] Scheiber, M.; Federle, C.; Feldhaus, J.; Golla, B.; Hartmann, B.; Kleinhenz, B.; Röhrig, M.; Martini, D.: Pflanzenschutz-Anwendungs-Manager (PAM): Automatisierte Berücksichtigung von Abtandsauflagen. Praktische Vorführung und Feldtestergebnisse. In (Ruckelshausen, A., Meyer-Aurich, A., Rath, T., Recke, G., Theuvsen, B., Hrsg.): Informatik in der Land-, Forst- und Ernährungswirtschaft 2016, Osnabrück. GI-Edition Lecture Notes in Informatics, Bonn, S. 177-180, 2016.

[Sp15] Speicher, S.; Arwe, J.; Malhotra, A. (Hrsg.): Linked Data Platform 1.0. W3C Recommendation 26 February 2015. https://www.w3.org/TR/ldp/ (aufgerufen am 27.11.2017).