KTBL-Planungsdaten auf dem Weg in die Zukunft – Bereitstellung über Linked Open Data

Daniel Martini, Esther Mietzsch, Mario Schmitz, Daniel Herzig, Günter Ladwig

Team Datenbanken und Wissenstechnologien
Kuratorium für Technik und Bauwesen in der Landwirtschaft e.V. (KTBL)
Bartningstraße 49
64289 Darmstadt
d.martini@ktbl.de
e.mietzsch@ktbl.de
m.schmitz@ktbl.de
herzig@searchhaus.net
ladwig@searchhaus.net

Abstract: Das KTBL liefert Daten für die Planung landwirtschaftlicher Produktion – in der Vergangenheit in Büchern und Tabellenwerken, seit einigen Jahren aber auch in Webanwendungen. Nun wurden Dienste implementiert, die Daten nach den Prinzipien und technischen Methoden des Semantic Web und Linked Open Data bereitstellen. Das nun aufgesetzte Produktivsystem ersetzt den im Rahmen der letzten GIL-Jahrestagung präsentierten Prototypen. Daten und Dienste beschreiben sich selbst und ein Abruf in maschinenlesbaren Formaten im Sinne von Webservices ist möglich. Der Beitrag zeigt die genutzten Komponenten und geht auf die Erfahrungen ein, die in der Umsetzung gemacht wurden.

1 Zielsetzung

Das KTBL liefert seit Langem Daten für die Planung von Produktion, Arbeitsvorgängen und Investitionen für die Landwirtschaft. Seit einigen Jahren werden diese Daten auch in interaktiven Webanwendungen bereitgestellt. In letzter Zeit wurde jedoch von Datenkonsumenten zunehmend Bedarf nach flexiblen Auswertungsmöglichkeiten und Einbindung in externe Anwendungen (z. B. Farmmanagement-Informationssysteme) formuliert. Daher wurden Dienste implementiert, die Daten jetzt auch nach den Prinzipien und technischen Methoden des Semantic Web und Linked Open Data bereitstellen. Die verwendeten semantischen Technologien bieten für den Anwender einen komfortableren Zugriff mit individueller Zusammenstellung benötigter Daten. Außerdem liefern sie selbstbeschreibende Daten und Dienste durch Nutzung von Standardvokabularien und den Abruf in maschinenlesbaren Formaten über einen Webdienst, um eine Einbindung in weitere Applikationen zu ermöglichen. Durch die für Linked Open Data typische Zuweisung von URLs an Entitäten können die Daten schließlich weltweit mit anderen Datensätzen verknüpft werden.

Das nun aufgesetzte Produktivsystem ersetzt den im letzten Jahr im Rahmen der GIL-Tagung bereits präsentierten Prototypen und bietet eine Reihe von zusätzlichen Funktionalitäten [MKH14].

2 Aufbau des Systems

Abbildung 1 zeigt die grundlegende Architektur des gesamten Systems mit allen seinen Komponenten. Letztere erfüllen dabei jeweils eine spezielle Funktionalität und haben eine klar definierte, meist standardisierte Schnittstelle, sodass diese grundsätzlich gegen andere Implementationen austauschbar sind. Im Vorfeld wurde jedoch eine von der Firma SearchHaus erstellte Übersicht und Empfehlung zu am Markt befindlichen Werkzeugen, die die Kernanforderungen berücksichtigte, ausgewertet und verschiedene Zusammensetzungen von Komponenten erprobt. Betrachtet wurden dabei die Teilbereiche „Erstellung eines semantisch angereicherten Datenbankextrakts“, „Graphenorientierte Speicherung und SPARQL-Abfrageendpunkt“ sowie „ReSTful http-Webservice zur Datenbereitstellung“.

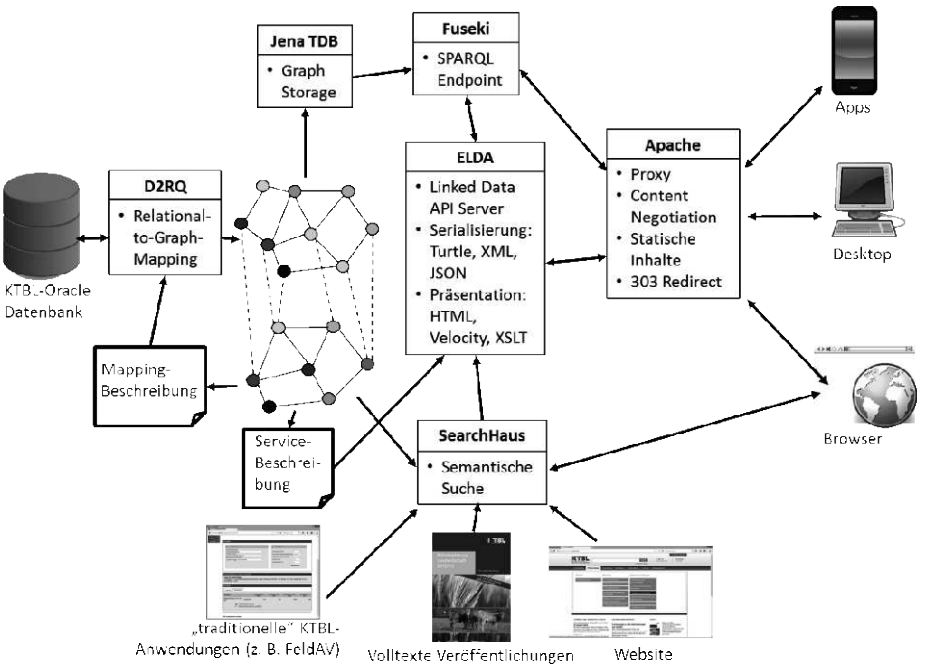

Abbildung 1: Architektur des Linked Open Data Services mit Suchserverkomponente

Grundlage des Linked Open Data Services sind die Daten aus der Oracle-Datenbank des KTBL, die auch für die interaktiven Online-Anwendungen genutzt werden. Aus diesen Daten wird mittels des Mappingwerkzeuges d2rq [DR14] eine in der Turtle-Syntax des

Ressource Description Framework (RDF, [CWL14]) abgebildete Graphenrepräsentation erstellt. Notwendig hierfür ist eine Mappingbeschreibung, die spezifiziert, wie Datenbanktabellen und -spalteninhalte in RDF-Klassen und -Eigenschaften zu überführen sind. Derzeit erfolgt anschließend eine automatisierte, skriptgesteuerte Nachbearbeitung mit gängigen UNIX-Textprocessing-Werkzeugen, die bestimmte Konstrukte ergänzen, die von d2rq nicht unterstützt werden. Die Klassen und Eigenschaften selbst werden in einer einfachen Ontologie ebenfalls in Turtle-Syntax beschrieben.

Sowohl der Datenbankdump als auch die Ontologie wird anschließend in einen sogenannten Triple-Store geladen, auf den der SPARQL-Server Apache Fuseki [Fu14] aufsetzt. Die Abfragesprache SPARQL [HS13] ermöglicht das gezielte Auffinden von Knoten und Kanten im RDF-Graphen. Auch komplexe Abfragen, die in einem relationalen Datenbanksystem nur mit umfangreichen JOIN-Operationen oder über eine serielle Abarbeitung mehrerer Queries möglich wären, können hiermit einfach und effizient formuliert werden.

Um Daten jedoch auch im Sinne gängiger Prinzipien von Linked Open Data und über einfache URI-Aufrufe in ReSTful Webservices ohne die Notwendigkeit der Nutzung von SPARQL zugänglich zu machen, wurde hierauf eine weitere Serverkomponente aufgesetzt. Zum Einsatz kommt dabei ELDA (Epimorphics Linked Data API, [El14]), eine Implementierung der offenen Spezifikation des Linked Data API [El14]. Die Daten können hiermit in einer Reihe von Formaten, unter anderem auch in der bei Entwicklern derzeit beliebten Java Script Object Notation (JSON) [Br14], abgerufen werden. Für den interaktiven Aufruf über den Browser steht eine HTML-Ansicht zur Verfügung. Diese wird mit Hilfe des Apache Velocity Template Engines [Ve14] erzeugt und kann daher durch Erstellung entsprechender HTML-Vorlagen, die dann zur Laufzeit entsprechend befüllt werden, beliebig angepasst werden. ELDA selbst wird über eine ebenfalls in RDF abgebildete API-Spezifikation gesteuert, die beschreibt, welche Daten in welcher Form ausgeliefert werden sollen.

Zudem sind dieselben Daten über einen semantischen Suchserver, der von SearchHaus implementiert wurde, zugänglich. Dieser erlaubt die gezielte Suche nach Schlüsselwörtern. Zusätzlich kann er bestimmte Eigenschaften darstellen, die semantisch mit seinem Suchbegriff verknüpft sind, und spontan Verknüpfungen zwischen den Datensätzen folgen. Vorgeschaltet ist dem ganzen System ein Apache Webproxy, der für Funktionalitäten wie die bei ReSTful Webservices übliche Content Negotiation und bestimmte Redirects zuständig ist.

3 Ergebnisse und Ausblick

Die gewählte Zusammenstellung von Komponenten für die o. g. Teilbereiche „Datenbankextrakt“, „Graphenorientierte Speicherung“ und „Webservice“ spiegelt im Rahmen der gegebenen funktionalen Anforderungen und Rahmenbedingungen hinsichtlich Datenumfang, vorhandener Serverinfrastruktur, verfügbarer Ressourcen für Wartung und Pflege usw. derzeit die einfachste und effizienteste Lösung wider. Mittelfristig ist für die Datenbankextraktion jedoch geplant, auf das Werkzeug db2triples umzusteigen, da

dieses die vom W3C spezifizierte Standardsprache R2RML unterstützt und bestimmte im Datensatz notwendige Konstrukte – z. B. sogenannte language tags an Textfeldern, die über einen Tabellen-JOIN generiert werden oder blank nodes – ohne Zwischenschritte erzeugt werden können.

Mit überschaubarem Aufwand können auch große Datenmengen aus bestehenden relationalen Datenbanken semantisch aufbereitet und für Abfragen, Suchvorgänge und über Webdienste zugänglich gemacht werden, die bislang nur sehr umständlich umzusetzende Funktionalitäten und Auswertungen auf einfache Art und Weise ermöglichen. Zudem hat sich gezeigt, dass sich sämtliche Abläufe in der oben beschriebenen Architektur nahezu vollständig automatisieren lassen. Bei einer Ergänzung von Daten sind keinerlei Anpassungen an Serverkomponenten notwendig. Das Hinzufügen von weiteren Klassen und Eigenschaften erfordert lediglich das Einfügen einiger Zeilen in die Mappingbeschreibung für d2rq und in die API-Spezifikation für ELDA. Mittelfristig ist geplant, auch den letzteren Schritt weitgehend zu automatisieren. Zur Verfügung zu stellende Service-URLs können nämlich auch aus der zu den Daten gehörenden Ontologie bereits erkannt werden. Durch den Linked Open Data Service werden die KTBL-Daten so aufbereitet, dass Anwendungsentwickler sie direkt in ihre Applikationen einbinden können und Endnutzern neue und komfortable Suchen ermöglicht werden.

Literaturverzeichnis

[Br14] Bray, T. (2014): The JavaScript Object Notation (JSON) Data Interchange Format, RFC 7159. https://tools.ietf.org/html/rfc7159, aufgerufen am 12.11.2014.

[CWL14] Cyganiak, R., Wood, D., Lanthaler, M. (2014): RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation. https://www.w3.org/TR/rdf11-concepts/, aufgerufen am 20.11.2014.

[DR14] The D2RQ Platform – Accessing Relational Databases as Virtual RDF Graphs. http://d2rq.org/, aufgerufen am 12.11.2014.

[El14] Elda 1.3.0 An implementation of the linked-data API, Elda quickstart. http://epimorphics.github.io/elda/docs/E1.3.0/index.html, aufgerufen am 20.11.2014.

[Fu14] Apache Jena - Fuseki: serving RDF data over HTTP. http://jena.apache.org/documentation/serving_data/, aufgerufen am 12.11.2014.

[HS13] Harris, S., Seaborne, A. (2013): SPARQL 1.1 Query Language. W3C Recommendation. http://www.w3.org/TR/sparql11-query/, aufgerufen am 20.11.2014.

[LD14] linked-data-api – API and formats to simplify use of linked data by web developers. https://code.google.com/p/linked-data-api/wiki/Specification, aufgerufen am 20.11.2014.

[MKH14] Martini, D., Kunisch, M., Herzig, D., Ladwig, G. (2014): Planungsdaten schnell finden und einfach nutzen: Linked Open Data und semantische Suche im Einsatz für das KTBL-Datenangebot. In: Referate der 34. GIL-Jahrestagung; GI-Edition – Lecture Notes in Informatics (LNI); Bonn, 2014; S. 225-228.

[Ve14] Apache Velocity User Guide. http://velocity.apache.org/engine/releases/velocity-1.7-/user-guide.html, aufgerufen am 20.11.2014.