

Recherchebericht

Projekt Graphical SPARQL Builder
Gruppe s14.swp.gsb
Verantwortlich Elvira Akhtyamova
Erstellt am 5. Januar 2014

Inhaltsverzeichnis

1	Begriffe	2
2	Konzepte	3
2.1	Semantic Web	3
2.2	SPARQL	4
2.3	Webapplikationen	4
2.3.1	HTML	4
2.3.2	JavaScript	4
2.4	MVC	5
3	Aspekte	5
3.1	Endpunktgebundenheit	5
3.2	Umsetzung der Komplexität von SPARQL	6
3.3	Frameworks	6
3.4	Kompatibilität	6
3.5	Intuitivität	6

1 Begriffe

API Application Programming Interface – Programmierschnittstelle. Eine API beschreibt, wie Software-Komponenten bzw. Programme miteinander interagieren können/sollten. Anders ausgedrückt: eine API ist der Teil eines Softwaresystems, der anderen Programmen zur Verfügung gestellt wird um mit dem Softwaresystem zu interagieren.

DBpedia DBpedia ist eine Datensammlung im RDF Format, deren Datensätze aus der Wikipedia extrahiert wurden. Ziel ist es strukturierte Daten für Webanwendungen zur Verfügung zu stellen. [1, 2, 3]

Endpoint Ein Endpoint ist eine Schnittstelle zwischen der Datensammlung und der Abfragesprache. Nachdem eine Anfrage an den Endpoint gesendet wurde (query) sendet selbiger die Ergebnisse zurück. Ein Beispiel für einen SPARQL-Endpoint ist der “Virtuoso SPARQL Query Editor”. [4]

GSB Graphical SPARQL Builder, der Name des Projekts. [5]

IRI “Internationalized Resource Identifier” kurz IRI ist eine spezialisierte Form der URI, die nur ASCII-Zeichen enthält. [6, 7]

JSON Javascript Object Notation – eine Dateiformat für den Datenaustausch zwischen Anwendungen. JSON soll für Mensch und Maschine gleichermaßen einfach lesbar sein und sich durch Kompaktheit auszeichnen. Dabei soll jedes gültige JSON Dokument gültiges Javascript darstellen. [8]

Ontologie Ontologien sind formalisierte Vokabulare von Begriffen. Diese Vokabulare beziehen sich meist auf eine bestimmte Domäne (Gegenstandsbereich) oder Nutzergruppe. Sie liegen in einer sprachlichen Form vor und umfassen die Begriffe einer Domäne sowie Beziehungen zwischen den Begriffen. [9, 10, 11]

OWL Die Web Ontology Language (OWL, aktuelle Version OWL2) ist eine Beschreibungssprache um Ontologien für das semantische Web zu erstellen und zu publizieren. OWL2-Ontologien können für Informationen verwendet werden, die in RDF geschrieben sind und werden hauptsächlich in Form von RDF-Dokumenten ausgetauscht. [9]

RDF “Resource Description Framework” kurz RDF ist eine Strukturierung von Daten nach dem Muster Subjekt-Prädikat-Objekt. Alle RDF-Daten werden in diesem Tripel-Format gespeichert. RDF gilt als eines der Basis-Elemente des semantischen Webs. Repräsentationen, also syntaktische Standards, des RDF-Prinzips sind N3 (Notation3), Turtle (Terse RDF Triple Language) sowie RDF/XML. Turtle und N3 gelten im Vergleich zu RDF/XML als benutzerfreundlicher. [12, 13, 14]

RDF-Modelle als gerichtete Graphen Aussagen werden in RDF als Tripel modelliert. Eine Menge solcher Tripel bildet einen gerichteten Graph, wobei Subjekte Objekte die Knoten und Prädikate die Kanten darstellen. [13]

RDFS “Resource Description Framework Schema” kurz RDFS ist ein Konzept für den Austausch von Daten. Durch Syntax-Vorgaben wird eine einheitliche und umfassendere Interpretation der Daten ermöglicht. [15]

Reasoning Als Reasoning bezeichnet man ein Ableitungsverfahren mit dem aus RDF-Daten durch gegebene Regeln und Strukturen neue Daten beziehungsweise deren Verknüpfung erzeugt werden können. [16, 17]

SPARQL “SPARQL Protocol And RDF Query Language” kurz SPARQL ist eine Abfragesprache für das Datenformat RDF. SPARQL ist graphbasiert und gilt nach dem W3C als Standard für RDF-Abfragen. [18, 19]

Triplestore Ein Triplestore ist ein System zur Speicherung, Verwaltung und Bearbeitung einer Sammlung von RDF-Tripeln. Ein Triplestore bietet gewöhnlich APIs, Reasoning-Verfahren sowie Abfragemöglichkeiten. [11]

URI “Uniform Resource Identifier” kurz URI sind Ressourcen-Identifikatoren. Sie bestehen aus Schema, Anbieter, Pfad, Abfrage und Teil, wobei nur Schema und Pfad zwingend vorhanden sein müssen. Ein RDF-Objekt wird durch eine URI eindeutig identifiziert. [20]

XML Die Extensible Markup Language (XML) ist eine Auszeichnungssprache. Sie dient der Darstellung hierarchisch strukturierter Daten in Form von Textdateien. Oberstes Design-Ziel ist dabei die Verwendbarkeit zum Datenaustausch über Netzwerke, speziell das Internet. Weiterhin stehen Implementierungs- und Plattformunabhängigkeit sowie Lesbarkeit für Menschen im Vordergrund. [21]

2 Konzepte

2.1 Semantic Web

Das *World Wide Web* ermöglicht den Zugriff auf eine gigantische Masse an Informationen. Doch als breit verteilte, ungeordnete Menge ist sie für Menschen nur schwer überschaubar – und auch für Maschinen kaum *semantisch*, also inhaltlich, erfassbar. Das macht es schwierig, heterogene Daten (etwa im Bezug auf Formate, Sprachen, Homepageaufbau) zu sammeln und einheitlich aufbereitet weiterzuverarbeiten, sowie implizites Wissen (etwa eine Folgerung aus mehreren dezentralen Fakten) zuverlässig zu extrahieren.

Das *Semantic Web* steht als Lösungsvorschlag dieser Problematik für das Konzept, Informationen auf eine Weise bereitzustellen, die eine maschinelle Verarbeitung möglich macht. Hierzu müssen einheitliche Standards gewährleistet werden, für die Beschreibung

der Bedeutung von Informationen (also deren Semantik) einerseits, und andererseits für den Austausch bzw. die Verknüpfung von Informationen verschiedener Anwendungen und Plattformen. XML, RDF(S) und OWL sind, als Informations-Spezifikationsprachen, Beispiele dieser Bemühungen, sowie auch die Anfragesprache SPARQL. [22]

2.2 SPARQL

SPARQL ist eine Anfragesprache für Datenbanken, deren Daten im RDF-Format vorliegen, also als RDF-Graphen. Sie nutzt daher einfache Tripelgraphen als grundlegendes Anfragemuster. Diese entsprechen den Tripelmustern von RDF, mit dem Unterschied, dass alle Bestandteile des Tripels, also Subjekt, Prädikat und Objekt Variablen darstellen können. Ein Teilgraph der RDF-Daten stimmt dann mit dem Anfragemuster überein, wenn RDF-Terme des Teilgraphen durch diese Variablen ersetzt werden können, und das Ergebnis wiederum äquivalent zum Teilgraphen ist.

Neben diesen einfachen Anfragen bietet SPARQL allerdings auch Möglichkeiten für die Konstruktion komplexerer Muster, zusätzlicher Filterbedingungen und die Formatierung der Ausgabe. So sind Konjunktionen, Disjunktionen und optionale Muster weitere wichtige Anfragefeatures. Damit ist SPARQL ein essentieller Teil des Semantic Webs. [23]

2.3 Webapplikationen

Der Graphical SPARQL Builder ist ausgelegt als Webapplikation, also als Anwendungsprogramm, welches einen Webbrowser zur Ausführung bzw. Darstellung nutzt, unabhängig von einem bestimmten Betriebssystem. Programmiert wird die Applikation mittels einer browsergestützten Programmiersprache, kombiniert mit einer Markup-Sprache. Für dieses Projekt werden dafür JavaScript bzw. HTML verwendet, da für diese eine breite Unterstützung in modernen Webbrowsern und offenen Standards zu finden sind.

2.3.1 HTML

HTML dient als Markup- bzw. Auszeichnungssprache der Beschreibung logischer Bestandteile, also des Formats, eines Dokuments. Sie zeichnet also etwa typische Elemente wie Überschriften, Textabsätze, Listen, Tabellen oder Grafikreferenzen als solche aus. HTML wird in Form hierarchisch gegliederter Tags geschrieben, die von spitzen Klammern umschlossen sind (etwa *<Beispiel>*) und zumeist in wiederum andere Elemente umschließenden Paaren (Starttag und Endtag) auftreten. Ein Browser übersetzt die so beschriebenen Metadaten dann in eine entsprechende Darstellung des Dokuments. Außerdem können auch Skripte von Programmiersprachen wie JavaScript miteingebettet werden, welche wiederum das Verhalten des Dokuments weiter beeinflussen. [24]

2.3.2 JavaScript

JavaScript ist eine von Netscape lizenzierte Skriptsprache, welche die Implementierung von Skripten ermöglicht, die, eingebettet in Web-Dokumenten, deren Funktionen beträchtlich erweitern können. Sie ist dynamisch typisiert, das heißt die Zuweisung von

Werten an Variablen unterliegt keinen typbasierten Einschränkungen, und unterstützt funktionale, objektorientierte und imperative Programmierstile. Des Weiteren stehen eine Vielzahl von JavaScript-Bibliotheken bereit, sogenannte Toolkits, welche Sammlungen von hilfreichen Funktionen zur Verfügung stellen, oder auch Frameworks, welche als eine Art Programmiergerüst verschiedene Aspekte der Sprache modifizieren können. [25, 26]

2.4 MVC

MVC steht für *model-view-controller* (Modell-Präsentation-Steuerung) und repräsentiert ein gängiges Software-Entwurfsmuster für die Planung und Implementierung eines graphischen User-Interfaces. Es steht für die Trennung der drei wesentlichen Komponenten:

- Das Objekt *Model* steht für das Anwendungsobjekt, welches die Daten und *Kernfunktionalitäten* bereitstellt. Es kennt seine Views und Controller, die zum Einsatz kommen und informiert diese über Änderung in den Anwenderdaten.
- *View* stellt die Schnittstelle der *Bildschirmrepräsentation* dar – also die Sicht auf das Model für den User und reagiert auf Änderungen im Model.
- Das Objekt *Controller* repräsentiert die Schnittstelle der *Benutzereingaben* und interpretiert bzw. übergibt die Eingabedaten dem Model.

Entscheidend im MVC-Muster ist die Unabhängigkeit der einzelnen Komponenten voneinander, die es ermöglicht Kernprogramm, Präsentation und Manipulation strikt zu trennen. Dadurch ist etwa ein unkomplizierter Austausch unterschiedlicher Views und/oder Controller für ein und dasselbe Model möglich.

Auch für dieses Projekt stellt MVC ein geeignetes Entwurfsmuster dar. Besonders die Austauschbarkeit der Komponenten wird von großem Nutzen sein, wenn es darum geht, verschiedene Ausgabe- bzw. Eingabevarianten zu testen. Eine enge Bindung dieser an das Kernmodell hätte dagegen eine impraktikable Aufwandserhöhung zur Folge. [27, 28]

3 Aspekte

3.1 Endpunktgebundenheit

Als mögliche Anwendungsgebiete des GSB sind die Ressourcen der DBpedia und des CPL (catalogus professorum lipsiensis) genannt. Es ergibt sich eine grundlegende Frage nach dem Grad der Gebundenheit an mögliche Endpunkte. Die DBpedia stellt zum Beispiel einen öffentlichen Virtuoso SPARQL Endpunkt bereit. Der Endbenutzer könnte auf die Eingabe einer solchen Quelle eine individuelle grafische Schnittstelle als Ausgabe bekommen, auf der er seine Anfragen an die eingegebene Quelle zusammenstellen kann. Alternativ könnten standardmäßig wenige Quellen zur Verfügung stehen, so dass der Endbenutzer ausschließlich Anfragen für diese grafisch zusammenstellen kann. Zusammenhängend mit dieser Entscheidung bestehen verschiedene Möglichkeiten zur Kommunikation mit den Endpunkten. Ausgehend von den bereitgestellten Schnittstellen muss

geklärt werden welches Datenformat (z. B. JSON/XML) beim Datenaustausch genutzt wird. [4, 29]

3.2 Umsetzung der Komplexität von SPARQL

SPARQL ist eine mächtige Abfragesprache, deren Komplexität unter Umständen zu Einschränkungen in der grafischen Bereitstellung der implementierten Sprachmerkmale führen könnte. Im Laufe des Projektes muss entschieden werden, welche Sprachmerkmale unverzichtbar sind und welche maximal optional implementiert werden könnten. Das W3C verabschiedete bisher zwei Versionen von SPARQL (Januar 2008 und März 2013). Für eine gute Arbeitsgrundlage am GSB sollte eine Version als die zu verwendende festgelegt werden. [18, 30]

3.3 Frameworks

Im Bereich der Entwicklung von Webanwendungen bieten Frameworks eine Rahmenstruktur, die HTML, CSS und/oder JavaScript um Funktionalitäten und Konzepte erweitert. Im Rahmen des Projekts sind Frameworks von Nöten, die

- das Verwenden des MVC-Modells (s. o.) ermöglichen und somit Entwicklung und Testen des GSB vereinfachen. Beispiele von JavaScript-Frameworks, die dies ermöglichen sind: AngularJS, Ember.js, Backbone.js
- die Manipulation des HTML-Dokuments ermöglichen und erleichtern (z. B. Drag & Drop, Dialoge, etc.). Beispiele für Frameworks dieser Art wären Prototype und JQuery + JQuery UI

Die dann genutzten Frameworks sollten leicht erlernbar sein und die Möglichkeit geben Code gut zu strukturieren.

3.4 Kompatibilität

Der GSB soll als Webanwendung im Browser des Benutzers ausgeführt werden. Aufgrund der unterschiedlichen technischen Möglichkeiten verschiedener Webbrowser muss sichergestellt werden, dass moderne Browser wie Firefox oder Chrome den GSB ausführen können. Ein weiterer noch zu klärender Aspekt ist die Kompatibilität zu nicht-lateinischen Zeichensätzen und Sprachen. Hierbei ist zwischen Eingabe und Ausgabe solcher zu unterscheiden.

3.5 Intuitivität

Die Intuitivität hängt stark von der Zielgruppe des GSB ab. Wenn der GSB vor allem ein Arbeitstool im akademischen Rahmen sein soll, kann man voraussetzen, dass der Benutzer sich mehr mit der Anwendung beschäftigt und zum Beispiel eine kurze Einarbeitungszeit verkraftet. Desweiteren ist die Intuitivität von der umgesetzten Komplexität von SPARQL abhängig. Es besteht die Möglichkeit, dass das Umsetzen komplexer Anfragen nicht auf intuitiven Weg möglich ist.

Literatur

- [1] <http://de.wikipedia.org/wiki/DBpedia>
- [2] <http://de.dbpedia.org/>
- [3] <http://wiki.dbpedia.org/Datasets>
- [4] <http://dbpedia.org/sparql>
- [5] <http://pcai042.informatik.uni-leipzig.de/~swp14-gsb/>
- [6] http://de.wikipedia.org/wiki/Internationalized_Resource_Identifier
- [7] <http://tools.ietf.org/html/rfc3987>
- [8] <http://www.json.org>
- [9] <http://www.w3.org/TR/2012/REC-owl2-overview-20121211/>
- [10] <http://de.wikipedia.org/wiki/Ontologie>
- [11] http://www.iais.fraunhofer.de/fileadmin/user_upload/Abteilungen/OK/PDFs/Alieva_Magisterarbeit.pdf
- [12] <http://www.w3.org/TR/rdf-primer/>
- [13] http://de.wikipedia.org/wiki/Resource_Description_Framework
- [14] <https://en.wikipedia.org/wiki/RDF/XML>
- [15] <http://www.w3.org/TR/rdf-schema/>
- [16] <http://www.w3.org/2004/12/rules-ws/paper/47/>
- [17] <http://elite.polito.it/files/courses/01LHV/2012/4-RDFreasoning.pdf>
- [18] <http://www.w3.org/TR/rdf-sparql-query/>
- [19] <http://en.wikipedia.org/wiki/SPARQL>
- [20] http://de.wikipedia.org/wiki/Uniform_Resource_Identifier
- [21] <http://www.w3.org/TR/xml/>
- [22] Hitzler, P., Krötzsch, M., Rudolph, S., Sure, Y., Semantic Web, Springer: 1. Auflage 2008.
- [23] <http://www.w3.org/TR/2013/REC-sparql11-query-20130321/>
- [24] <http://de.selfhtml.org/intro/technologien/html.htm>

- [25] <http://de.selfhtml.org/intro/technologien/javascript.htm>
- [26] <http://en.wikipedia.org/wiki/JavaScript>
- [27] http://www.informatik.uni-leipzig.de/~meiler/MuP.dir/MuPWS12.dir/Vorlesung/Kap13_MVC.pdf
- [28] <http://www.codeproject.com/Articles/25057/Simple-Example-of-MVC-Model-View-Controller-Design>
- [29] <http://wiki.dbpedia.org/OnlineAccess>
- [30] <http://www.w3.org/TR/sparql11-overview/>