

Projektangebot

Projekt Graphical SPARQL Builder
Gruppe s14.swp.gsb
Verantwortlich Gesamte Gruppe
Erstellt am 20. Januar 2014

Inhaltsverzeichnis

1	Zielbestimmung	2
2	Voraussetzungen	2
3	Designübersicht und Funktionalität	3
3.1	Nutzer Use Cases	3
3.1.1	Benutzer	3
3.1.2	Administrator	3
3.2	SPARQL Use Cases	3
4	Arbeitspakete und Meilensteine	6
4.1	Vorprojekt (30% des Gesamtprojekt)	6
4.2	Graphische Umsetzung (40%)	6
4.3	Endpoint-Anbindung (20%)	7
4.4	Benutzerhandbuch (10%)	7
5	Qualitätssicherung	7
6	Glossar	8

1 Zielbestimmung

Eine der wichtigsten Leistungen des World Wide Web ist sicherlich das Angebot immenser Mengen frei zugänglichen Wissens. Ein Teil dieses Wissens liegt dabei in einer Vielzahl von Datenbanken, welche dem Semantic Web zugeordnet sind. Dieses Konzept steht für die Bereitstellung von Daten auf eine maschinell verarbeitbare Weise, wodurch Suchanfragen, Austausch und Verknüpfungen von Daten, sowie strukturierte Erzeugung und Verwertung von Metadaten möglich werden. Für die Suche in solchen RDF-basierten Datenbanken stellt SPARQL als Anfragesprache ein mächtiges Werkzeug dar - für den eingelernten Nutzer. Dieses Projekt verfolgt als Ziel die Entwicklung eines Graphical-SPARQL-Builders (GSB) und spricht im Vergleich dazu vor allem diejenigen an, denen es an Erfahrungswerten in Bereichen der SPARQL-Syntax bzw. RDF-Graphen allgemein mangelt. Der GSB als graphische Webanwendung soll eine eingängige Benutzeroberfläche bereitstellen, die eine Vielzahl von Anfragen an SPARQL-Endpunkte ermöglicht. Der Schwerpunkt soll dabei vor allem auf einer einfachen, intuitiven Bedienung liegen, welche die darunterliegende SPARQL-Syntax weitestgehend abstrahiert, um ohne aufwendige Einarbeitung die Eingabe komplexer Anfragemuster zu ermöglichen.

2 Voraussetzungen

Dem GSB geht kein Projekt voraus und er muss somit komplett neu entwickelt werden. Technische Voraussetzungen sind die Standardwebtechnologien HTML und JavaScript, die von einem Webserver ausgeliefert und vom Browser des Nutzers interpretiert werden. Der Browser führt hierbei das Programm komplett aus und fragt nur zum Zwecke des Austauschs von Daten den SPARQL-Endpoint via AJAX an.

Webserver Für die erfolgreiche Nutzung des Programms muss ein Webserver bereitgestellt werden, auf dem das Produkt laufen kann und von dem aus online auf den Dienst zugegriffen werden kann.

Datenbanksystem Ein Datenbanksystem wird benötigt, welches die Tripel, die das Produkt verwendet, verwaltet. Beispiele für solche Triplestores sind DBpedia und CPL (catalogus professorum lipsiensis).

Client Der Client selbst benötigt einen Webbrowser zur Ausführung des Programms, bzw. zur Darstellung der Daten. Dieser muss in der Lage sein moderne Sprachen wie HTML5 und JavaScript zu unterstützen. Inbegriffen sind hierbei beliebte Browser wie Firefox, Chrome und Internet Explorer.

3 Designübersicht und Funktionalität

3.1 Nutzer Use Cases

3.1.1 Benutzer

Benutzer werden hier als diejenige (Ziel-)Gruppe von Personen definiert, die den GSB verwenden werden, um bestimmte Daten aus einer RDF-Datenbank zu extrahieren. Benutzer können Anfragen an den vom Administrator bestimmten Endpunkt der RDF-Datenbank mit Hilfe der zur Verfügung gestellten grafischen Oberfläche zusammenstellen und eine Ergebnisausgabe anfordern. Sie können außerdem eine Anleitung einsehen, die ihnen die Nutzung erleichtern soll.

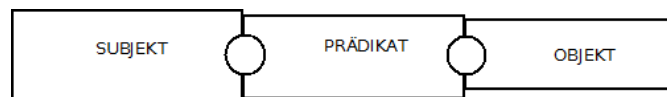
3.1.2 Administrator

Der Administrator ist diejenige Person, welche über erweiterte Zugriffsrechte verfügt, die ihm ermöglichen, Funktionalitäten der Anwendung bedarfsmäßig anzupassen. Dazu zählen hauptsächlich die Anpassung des Endpoints und der Spracheinstellungen. Denkbar sind außerdem Änderungen in der Anfragesyntax, wie etwa das Hinzufügen eines neuen Filters. Dies würde aber über den Rahmen des Projekts hinausgehen.

3.2 SPARQL Use Cases

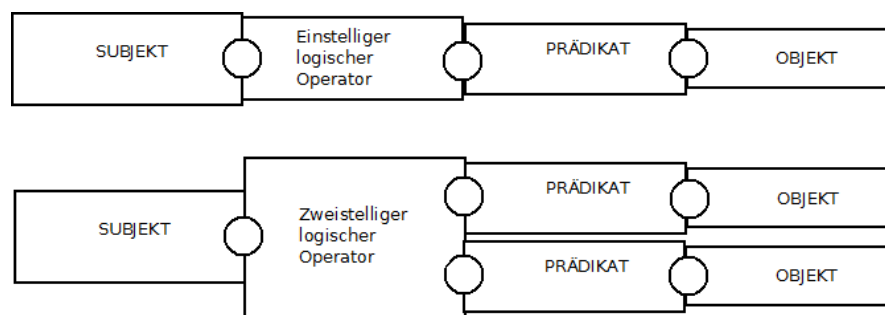
Im Folgenden wird die Struktur von Anfragen im GSB dargestellt und am Beispiel die daraus resultierende SPARQL-Anfrage aufgezeigt.

Die Standardstruktur



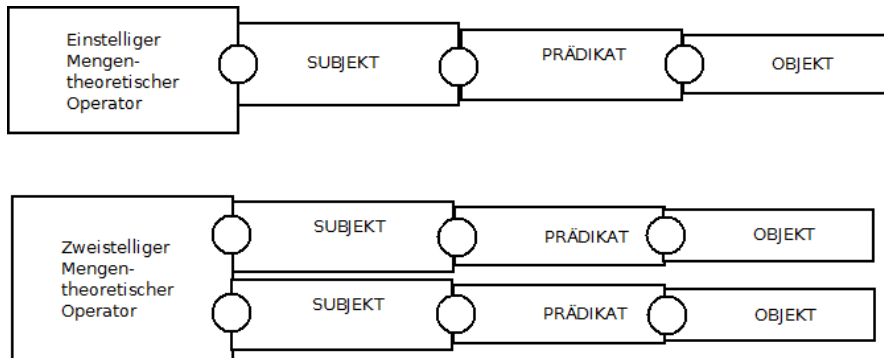
Die Standardstruktur ist dem RDF-Tripel nachempfunden. Das Subjekt entspricht einer Ressource, das Prädikat einem Attribut und das Objekt ist Teil der Menge der Ressourcen und konkreten Werte.

Logische Verknüpfungen



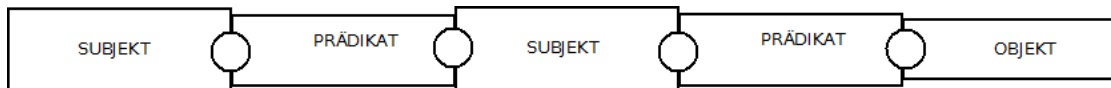
Logische Operatoren können zwischen Subjekt und Prädikat stehen. Sie werden nach Stelligkeit in zweistellige Operatoren (AND, OR, NAND, XOR) und den einstelligen Operator NOT unterteilt.

Mengentheoretische Verknüpfungen



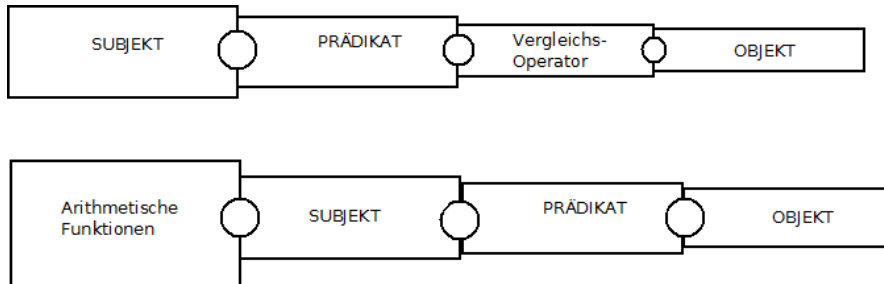
Mengentheoretische Operatoren stehen vor Subjekten. Sie werden wie die logischen Operatoren anhand ihrer Stelligkeit unterschieden: Zweistellige Operatoren (Vereinigung, Schnitt und Differenz) sowie der einstellige Mengenoperator Kardinalität (COUNT).

Reihungen



Strukturen können “in eine Reihe” gesetzt werden, wenn das Objekt der ersten Struktur das Ergebnis der zweiten Struktur sein soll.

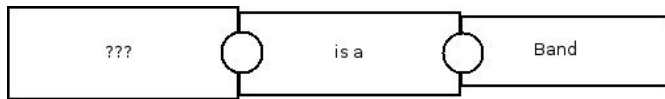
Arithmetische Verknüpfung



Arithmetische Operatoren werden in diesem Kontext unterteilt in Vergleichsoperatoren, welche vor dem Objekt stehen können und arithmetische Funktionen (MAX, MIN, AVG, SUM), welche vor dem Subjekt stehen und nur angewandt werden sollten wenn das Ergebnis der folgenden Struktur eine Menge von Zahlen ist.

Beispiele mit den entsprechenden SPARQL-Anfragen

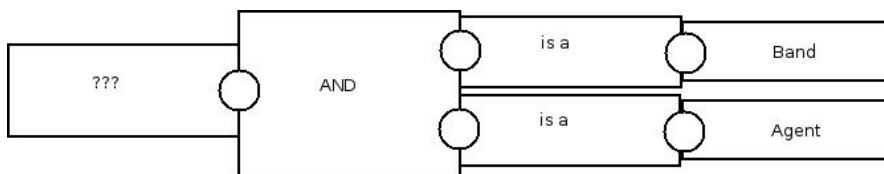
Die Standardstruktur



```

select ?s
where
{
?s a <http://dbpedia.org/ontology/Band>.
}
  
```

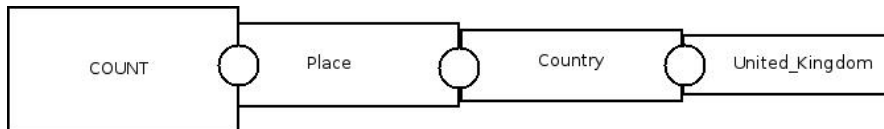
Logische Verknüpfungen



```

select ?s
where
{
?s a <http://dbpedia.org/ontology/Band>.
?s a <http://dbpedia.org/ontology/Agent>.
}
  
```

Mengentheoretische Verknüpfungen



```

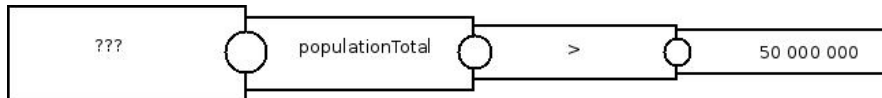
select count(*)
where
{
?s a <http://dbpedia.org/ontology/Place>.
?s <http://dbpedia.org/ontology/country> <http://dbpedia.org/resource/United_Kingdom>.
}
  
```

Reihungen



```
select ?bm
where
{
?stadt <http://dbpedia.org/ontology/leaderName> ?bm.
?stadt <http://dbpedia.org/ontology/country> <http://dbpedia.org/resource/United_Kingdom>.
}
```

Arithmetische Verknüpfung



```
select ?s
where
{
?s <http://dbpedia.org/ontology/populationTotal> ?o.
FILTER(?o > 50000000 ).
}
```

4 Arbeitspakete und Meilensteine

4.1 Vorprojekt (30% des Gesamtprojekt)

Muss-Ziele

/M1/001 Entwicklung eines graphischen Modells für SPARQL-Anfragen

/M1/002 Beispielhafte Umsetzung ausgewählter SPARQL-Anfragen mit AngularJS

4.2 Graphische Umsetzung (40%)

Muss-Ziele

/M2/001 Umsetzung notwendiger SPARQL-Anfragen

Kann-Ziele

/K2/001 Umsetzung nicht geforderter SPARQL-Anfragen

/K2/002 Modulare Entwicklung um Erweiterbarkeit sicherzustellen

/K3/003 Speichern von Anfragen / Beispielanfragen

/K3/004 i18n

4.3 Endpoint-Anbindung (20%)

Muss-Ziele

/M3/001 Anbindung über Konfigurationsdateien

/M3/002 Beispielhafte Anbindung des GSB an dbpedia Endpoint

Kann-Ziele

/K3/001 Anbindung an mehrere Endpoints

/K3/002 Caching von Daten um Reaktionszeiten zu verkürzen

4.4 Benutzerhandbuch (10%)

Muss-Ziele

/M4/001 Schriftliches Benutzerhandbuch

Kann-Ziele

/K4/001 Video-Anleitung für Benutzer

5 Qualitätssicherung

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität	×			
Zuverlässigkeit			×	
Benutzbarkeit	×			
Effizienz		×		
Anpassbarkeit			×	
Übertragbarkeit			×	

Erläuterungen: Da der GSB als eingängige, benutzerfreundliche Webapplikation ausgelegt ist, stehen Benutzbarkeit und Funktionalität klar im Vordergrund, da diese Aspekte als Kernargumente für die Nutzung der Anwendung klare Priorität haben. Zuverlässigkeit spielt eine weniger entscheidende Rolle. Abstürze sind natürlich nie erwünscht, doch müssen hier etwa keine wichtigen Daten vor einem Verlust bewahrt werden, oder Nutzertransaktionen mit allen Mitteln abgesichert werden. Effizienz ist definitiv wichtig, wenn

es um Operationen auf Datenbanken geht. Da in den meisten Fällen mit großen Mengen von Daten gearbeitet wird, und die Datenanfrage die zentrale Aufgabe des GSB darstellt, ist es wünschenswert, dass die Anwendung diese in zufriedenstellender Zeit bearbeiten und Ergebnisse präsentieren kann. Anpassbarkeit könnte besonders im Bezug auf die Sprache der Anwendungsoberfläche und eine Erweiterung der Anfragemöglichkeiten von Interesse sein. Doch da dieses Projekt sich mehr als 'Grundsteinlegung' einer graphischen SPARQL-Anfragehilfe versteht, werden sich auch die Ressourcen zunächst entsprechend anderweitig konzentrieren. Übertragbarkeit bezieht sich im Falle des GSB hauptsächlich auf die Möglichkeit, Anfragen an unterschiedliche Endpunkte zu stellen, bis hin zu einer kombinierten Anfrage auf mehrere Endpoints. Da Ersteres zunächst nur über eine Konfigurationsdatei abgehandelt wird, und Letzteres in seiner Komplexität den Rahmen dieses Projekts sprengen würde, wird diesem Aspekt ebenfalls eine abgeschwächte Priorität zugeordnet.

6 Glossar

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]

i18n internationalization and localization – Anpassung der Software an andere Sprachen und Kulturen ohne Quelltext zu ändern. Sprach- und Kulturspezifika werden über Konfigurationsdateien angepasst.

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. [6, 7, 8]

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. [6]

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. [9, 10, 11]

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. [12, 13]

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. [8]

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://www.w3.org/TR/2012/REC-owl2-overview-20121211/>
- [7] <http://de.wikipedia.org/wiki/Ontologie>
- [8] http://www.iais.fraunhofer.de/fileadmin/user_upload/Abteilungen/OK/PDFs/Alieva_Magisterarbeit.pdf
- [9] <http://www.w3.org/TR/rdf-primer/>
- [10] http://de.wikipedia.org/wiki/Resource_Description_Framework
- [11] <https://en.wikipedia.org/wiki/RDF/XML>
- [12] <http://www.w3.org/TR/rdf-sparql-query/>
- [13] <http://en.wikipedia.org/wiki/SPARQL>