

RDF-S3 und eRQL: RDF Technologien für Informationsportale

Karsten Tolle¹, Fabian Wleklinski²

¹ Institut für Informatik, Datenbanken und Informationssysteme (DBIS)
Johann Wolfgang Goethe-Universität Frankfurt am Main
D-60325 Frankfurt am Main
tolle@dbis.informatik.uni-frankfurt.de

² eWorks
D-60487 Frankfurt am Main
Wleklinski@eworks.de

Abstract: Semantische Technologien versprechen einen Mehrwert für Informationsportale. Grundvoraussetzung für diese Technologien ist die semantische Modellierung der Daten, wobei das *Resource Description Framework (RDF)* hierfür eine Möglichkeit darstellt. Existierende RDF Anwendungen sind jedoch nicht für einen Einsatz in Informationsportalen gedacht und werden daher den dortigen besonderen Bedürfnissen nicht gerecht. Um diesen Bedürfnissen entgegen zu kommen, haben wir mit *RDF-S3* eine Speichermöglichkeit geschaffen, welche durch Quellinformationen die Überprüfbarkeit und Glaubhaftigkeit der Daten erhöht, sowie mit *eRQL* eine Anfragesprache erstellt, die besonders gut für einen Einsatz in einem Informationsportal geeignet ist.

1 Einführung

Für das heutige Internet sind Web- oder Informations-Portale von großer Bedeutung. Sie werden als Einstiegspunkte zum Internet oder zu bestimmten Themengebieten verwendet und bieten dem Nutzer Such- und Klassifizierungsfunktionen an. Die Nützlichkeit und damit auch der Erfolg eines Web Portals hängt direkt mit der Qualität dieser Funktionen zusammen. Eine Möglichkeit zur Verbesserung dieser Qualität ist, die Informationen so zu modellieren, dass auch ihre Semantik (Bedeutung) durch Computer ausgewertet werden kann. Derart modellierte Informationen können einfacher eingebunden werden, ermöglichen bessere Suchergebnisse und sogar das Ableiten neuer Informationen. Das *Resource Description Framework (RDF)* [1] ermöglicht eine solche Modellierung. Es wurde vom W3C entwickelt und bestehend aus mehreren Standards. Bei der Verwendung von RDF werden die Informationen durch einfache Aussagen (Tripel), die aus *Subjekt*, *Prädikat* und *Objekt* bestehen, zusammengesetzt.

Grundlegend für den Umgang mit RDF Daten sind die Art und Weise der Speicherung, die Anfragemöglichkeit und die Darstellung von Anfrageergebnissen. Es existieren bereits Anwendungen, die die genannten Funktionalitäten unterstützen, jedoch gehen diese nicht gezielt auf die Bedürfnisse der Informationsportale ein. Mit unserer Arbeit

wollen wir genau diese Bedürfnisse aufdecken und durch neue Technologien unterstützen.

In den folgenden drei Abschnitten werden nacheinander die Bereiche Speicherung, Anfragemöglichkeiten und Darstellungsformen von Ergebnissen im Bezug auf Informationsportale diskutiert und unsere Lösungsvorschläge genannt. Im abschließenden 5. Abschnitt fassen wir kurz zusammen und geben einen Ausblick auf unsere künftigen Vorhaben.

2 Speicherung: RDF-S3

Nutzer von Informationsportalen erwarten heutzutage, dass ihre Anfragen in kürzester Zeit beantwortet werden. Das Durchsuchen der Informationsquellen an sich, wie z.B. Web Seiten, wäre zu langsam und kommt hierfür nicht in Frage. Vorab gesammelte und gespeicherte Daten bilden daher die Grundlage für die Beantwortung von Anfragen. Informationen, die weder in diesen Daten enthalten, noch aus ihnen ableitbar sind, können auch nicht zurückgegeben werden. Dies gilt auch für Zusatzinformationen, wie z.B. woher die Daten stammen (Quellinformation), oder wann und von wem sie abgespeichert wurden. Für die Nutzer von Informationsportalen und auch für das manuelle Aufbereiten und Überprüfen der Daten durch die Betreiber, sind diese Zusatzinformationen von besonderem Interesse: nur so kann bei der Quelle nach weiteren Informationen gesucht, die Daten überprüft oder sich nur anhand der Zusatzinformationen ein Bild über die Glaubwürdigkeit und Aktualität der Daten gebildet werden. Wie das folgende Beispiel zeigt, ist dies insbesondere im Fall von RDF wichtig, bei dessen Verwendung jede beteiligte Person beliebige Aussagen hinzufügen kann.

Beispiel: Angenommen ein Produkt P wird von verschiedenen Anbietern angeboten. Sei weiterhin der in Abbildung 1 abgebildete RDF Graph gespeichert. Der Graph beinhaltet die Aussagen, dass ein Anbieter xyz P für 100 \$ anbietet und ein weiterer Anbieter abc hierfür nur 80 \$ verlangt. Die erstgenannte Aussage mag stimmen, kann jedoch ebenso gut vom Anbieter abc kommen, um vorzutäuschen er sei günstiger – wobei in Wirklichkeit der Anbieter xyz P für nur 70 \$ anbietet. Man kann den Daten also nicht ohne weiteres trauen! Wüsste man zusätzlich, woher die Informationen stammen, würde dies die Glaubwürdigkeit und Überprüfbarkeit erhöhen.

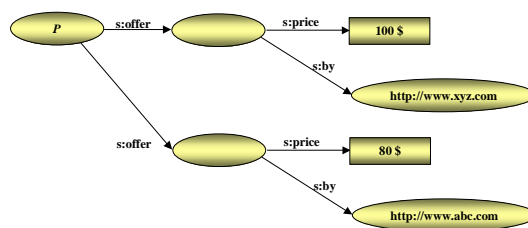


Abbildung 1 Beispiel eines RDF Graphen, der zwei Angebote für das Produkt P darstellt.

Es gibt verschiedene Möglichkeiten, die genannten Zusatzinformationen zu speichern. Eine Möglichkeit wäre, innerhalb von RDF *Aussagen über Aussagen* (*reified statements*) hinzuzufügen. Dies bereitet jedoch das Problem, dass auch diese anschließend nicht von Aussagen dritter unterschieden werden könnten. Man stände so wieder am Anfang: Welchen Aussagen kann man trauen, und welchen nicht? Eine weitere diskutierte Möglichkeit ist die Erweiterung der Tripel zu Quadrupel, wobei der vierte Teil für Zusatzinformationen genutzt wird. Bei der Verwendung von Quadrupel wird davon ausgegangen, dass dieser vierte Teil auch wieder innerhalb der normalen Tripel verwendet werden kann, z.B. um Aussagen über diese Quellinformation zu treffen. Dies ist zwar sehr aussagekräftig, ist aber nicht mehr konform mit dem RDF Model, was dazu führt, dass andere existierende RDF Anwendungen hiermit nicht arbeiten können. Insofern ist diese Möglichkeit ebenfalls nicht praktikabel. Eine Lösung stellt die Verwendung von *eingeschränkten Quadrupeln* dar. Hierbei wird der vierte Teil nicht innerhalb der Tripel verwendet. Dieses Vorgehen ermöglicht, dass existierende RDF Anwendungen – durch Ignorieren dieser Zusatzinformationen – weiter verwendet werden können und dennoch jedes Tripel mit seiner Quellinformation hinterlegt werden kann.

Das von uns entwickelte *RDF-Source related Storage System*¹ (*RDF-S3*) [4] verwendet diesen Ansatz und speichert zu jedem RDF Tripel zusätzlich seine Quellinformation, also die URL von der die Daten in RDF-S3 gespeichert wurden. Hierdurch wird dem Nutzer ermöglicht a) an der Informationsquelle nach weiteren Informationen zu suchen, b) die Aktualität der Informationen anhand des Speicherdatums mit ein zu beziehen, c) durch externe Ranking Systeme über die Quellen oder Präferenzen des Nutzers die Glaubwürdigkeit der Daten zu bestimmen, oder d) durch minimale Anpassungen verschiedene Versionen eines Dokumentes nebeneinander abzulegen und so dessen Entwicklung zu dokumentieren. Des weiteren bringt es enorme Vorteile für die Verwaltung der Daten, da so einzelne Quellen wieder gelöscht oder aktualisiert werden können.

Zusätzlich ist zu erwähnen, dass RDF-S3 die beiden gängigen Speicherstrategien der Generischen- und Schemaspezifischen-Repräsentation kombiniert und so die spezifischen Nachteile der einzelnen Strategien kompensieren kann. Daher können sowohl Daten- als auch Schema-Anfragen unter RDF-S3 schnell beantwortet werden. Dies unterscheidet RDF-S3 von anderen Anwendungen wie der RDFSuite [8] oder Jena [6].

3 Anfragemöglichkeit: eRQL

Um Anfragen an ein Informationsportal stellen zu können, ist eine entsprechende Benutzungsschnittstelle notwendig – beispielsweise auf einer Abfragesprache basierend. Diese Abfragesprache muss auch für technisch unversierte Benutzer ähnlich intuitiv einsetzbar sein, wie z.B. die Abfragesprache der Internetsuchmaschine Google.

¹ RDF-Source related Storage System (RDF-S3) ist ein 100% Java™ Open Source Anwendung, erhältlich unter: <http://www.dbis.informatik.uni-frankfurt.de>.

Weiterhin muss es mittels dieser Abfragesprache möglich sein, ohne Kenntnisse der Datenstruktur zu operieren, wobei für erfahrene Nutzer eine erweiterte Suche, die Zusatzwissen über die RDF Schema Struktur der Daten einbezieht, durchaus sinnvoll ist.

Existierende Abfragesprachen für RDF erfüllen nicht alle dieser Forderungen. Die gängigsten und sehr mächtigen Anfragesprachen, wie z.B. RQL [2] oder RDQL [3], lehnen sich an SQL an. Sie sind damit jedoch von Nutzern ohne Vorwissen nur schwer bedienbar. Weiterhin ist festzustellen, dass ein Nutzer, der nicht genau weiß wonach er sucht, mit den existierenden Anfragesprachen nur schwer zu einem brauchbaren Ergebnis kommt. Diese Anfragesprachen erwarten einerseits eine gewisse Präzision, z.B. liefert RQL bereits bei abweichender Eingabe in Punkto Groß- und Kleinschreibung die entscheidenden Fundstellen nicht mehr, andererseits enthält das Ergebnis einer RQL-Abfrage ausschließlich diejenigen Tripel, auf welche die Anfrage zutrifft. Dies reicht unserer Meinung nach nicht aus, um ein Anfrageergebnis zu verstehen oder sinnvoll weiter zu verwenden. Insbesondere ist dies der Fall, wenn das Ergebnis aus mehreren Teilgraphen besteht, die zu der Anfrage passen – die Fundstellen werden dann als voneinander unabhängige Tripel zurückgegeben, die Gruppierung zu den Teilgraphen geht in der Regel verloren.

Um nun den genannten Anforderungen an ein Informationsportal gerecht zu werden, haben wir eine neue Anfragesprache entwickelt, die auf einer existierenden Anfragesprache aufsetzt. Hierbei haben wir uns aufgrund seiner Mächtigkeit für RQL entschieden. In Anlehnung hieran haben wir sie *easy RQL (eRQL)* [5] genannt. Die Syntax von eRQL ist extrem einfach, unserer Beobachtung und Einschätzung nach die einfachste Syntax aller RDF-Abfragesprachen. eRQL bietet sowohl Ein-Wort-Abfragen als auch boolesche Verknüpfungen, wie sie jeder Nutzer von Internet Suchmaschinen kennt. Zwischen Groß- und Kleinschreibung unterscheidet eRQL nicht.

Als weitere nützliche Erweiterung zu RQL bietet eRQL die Möglichkeit, zusätzlich zu jeder Fundstelle auch deren umgebende Tripel zurückzugeben. Der Nutzer kann hierbei den Radius dieser Umgebung durch die Anfrage spezifizieren. Dies ermöglicht dem Nutzer das Ergebnis in seinem Kontext zu sehen, um dieses besser verstehen zu können. Mit der *eRqlEngine*² bieten wir zurzeit eine Prototyp Implementierung an, um eRQL zu testen. Hierbei wird der Ergebnisgraph gegenwärtig aber nur in Form textueller Tripel ausgegeben (siehe Abbildung 2).

² *eRqlEngine* ist eine 100% Java™ Open Source Anwendung, erhältlich unter: <http://www.dbis.informatik.uni-frankfurt.de/~tolle/RDF/> oder <http://www.wlekliniski.de/rdf/>.

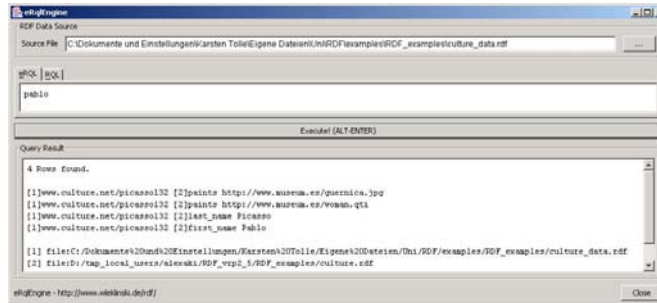


Abbildung 2 Bildschirmfoto der *eRqlEngine*, hier mit einer Ein-Wort-Abfrage „Pablo“ und dem resultierendem Ergebnis in der unteren Hälfte. Es ist zu erkennen, dass Tripel aus den Umgebungen der Fundstellen (hier mit dem Radius 1) zurückgegeben werden, welche den Suchbegriff selbst nicht enthalten.

4 Darstellung von Ergebnissen

Textuelle Tripel, wie sie zurzeit von der *eRqlEngine* als Ergebnis zurückgegeben werden, sind für einen Menschen nur sehr schwer lesbar. Zur Darstellung sollten daher andere Möglichkeiten, wie die Graph-Repräsentation der Tripel, gewählt werden. Dies könnte durch gängige Anwendungen wie *IsaViz* [7] erfolgen. Für Informationsportale ist es unserer Meinung nach mit der einfachen Darstellung des Ergebnisses als Graph jedoch nicht getan. Insbesondere größere Ergebnisgraphen werden unübersichtlich. Um die Lesbarkeit und das Verständnis der Ergebnisse zu verbessern, sind wir dabei eine Anwendung zu entwickeln, die den Antwortgraphen in seine zusammenhängenden Teilgraphen aufteilt. Innerhalb dieser Teilgraphen können einzelne Tripel nach voreingestellten Methoden, z.B. durch ein Ranking der Namensräume, gewichtet werden. Überschreiten die Teilgraphen eine gewisse Größe, so könnten die „wichtigen“ Tripel graphisch hervorgehoben werden. Als Beispiel sei hierbei erwähnt, dass insbesondere Literale³ für das menschliche Verständnis eine hohe Bedeutung haben und daher höher gewichtet werden könnten.

5 Zusammenfassung und Ausblick

Wie wir gezeigt haben, werden grundlegende Funktionalitäten für Informationsportale von heutigen RDF Anwendungen noch nicht ausreichend unterstützt. Mit den Anwendungen *RDF-S3* und *eRQL* bieten wir Lösungen für die Speicherung und die Anfragemöglichkeit an, die den nötigen Anforderungen an ein Informationsportal gerecht werden. Bis spätestens Mitte 2004 ist geplant, *eRQL* so zu erweitern, dass es die Zusatzinformationen, die *RDF-S3* anbietet, unterstützt und eine Anfrage über diese ermöglicht. Hiermit könnte es den Nutzern eines Informationsportals ermöglicht werden, sowohl gezielt in einzelnen Quellen zu suchen, als auch ihnen nicht vertrauenswürdige

³ Literale sind lesbare Texte und Textfragmente, Zahlen, Datumsangaben, etc.

Quellen auszublenden. Auch bei der Visualisierung von Anfrageergebnissen ist geplant, die Zusatzinformationen aus RDF-S3 auszunutzen.

Links

- [1] Webseite vom W3C zum Resource Description Framework: <http://www.w3.org/rdf/>
- [2] RQL – RDF Query Language: <http://athena.ics.forth.gr:9090/RDF/RQL/index.html>
- [3] RDQL – RDF Data Query Language: <http://www.hpl.hp.com/semweb/rdql.htm>
- [4] RDF-S3: <http://www.dbis.informatik.uni-frankfurt.de/~tolle/RDF/RDFS3/index.html>
- [5] eRQL – easy RQL: <http://www.dbis.informatik.uni-frankfurt.de/~tolle/RDF/eRQL/index.html>
oder <http://www.wleklinski.de/rdf/>
- [6] Jena von HP: <http://www.hpl.hp.com/semweb/>
- [7] IsaViz: A Visual Authoring Tool for RDF: <http://www.w3.org/2001/11/IsaViz/>
- [8] ICS-FORTH RDFSuite: <http://athena.ics.forth.gr:9090/RDF/>