



Retrieval auf RDF

Momentan existiert noch keine standardisierte Anfragesprache für RDF Dokumente. Auf Grund der existierenden XML Repräsentation von RDF liegt es jedoch nahe, eine Anfragesprache an XML für das Retrieval auf RDF zu verwenden. Mit XPath [XPath] liegt eine standardisierte Anfragesprache an XML vor, die in den wichtigsten Programmiersprachen implementiert ist.

Es zeigt sich jedoch sehr schnell, dass XPath nur in Spezialfällen für das Retrieval auf RDF geeignet ist. Da die Entwicklungen von RDF Anfragesprachen jedoch noch weit von einer Standardisierung und Implementierung entfernt sind, gibt es zu dem Umweg über XML häufig keine Alternative.

Aus diesem Grund wird zunächst das Retrieval auf XML Dokumenten mit Hilfe von XPath und die Anwendung dieser Technik auf RDF Dokumente vorgestellt. Im letzten Kapitel wird ein Überblick über den aktuellen Stand der Entwicklung von speziellen RDF Anfragesprachen gegeben. Insbesondere wird mit RQL [RQL] eine Anfragesprache vorgestellt, die sowohl RDF als auch RDF Schema unterstützt. Leider gibt es zur Zeit noch keine Implementierung von RQL.

Retrieval auf XML

Ein XML Dokument zeichnet sich im Wesentlichen durch seine baumartige Struktur aus:

```
<document class="H.3.3">
  <author>John Smith</author>
  <title>XML Retrieval</title>
  <chapter>
    <heading>Introduction</heading>
    This text explains all about XML and IR.
  </chapter>
  <chapter>
    <heading>Extensible Style
      Language</heading>
    <section>
      <heading>Examples</heading>
    </section>
    <section>
      <heading>Syntax</heading>
    </section>
  </chapter>
</document>
```

Die grundlegende Funktionsweise von XPath wird nun anhand einiger Beispiele demonstriert.

Die XPath Anweisung `//heading` liefert als Ergebnis die Werte alle `<heading>` Elemente zurück: `Introduction`, `Extensible Style Language` und `Syntax`. Die Anweisung `//chapter/heading` liefert alle `<heading>` Elemente, deren Elternelement ein `<chapter>` ist. Im obigen Beispiel würden also nur die Werte `Introduction` und `Extensible Style Language` als Ergebnis geliefert werden. Durch das Anhängen eines Filters `[..]` an den Pfad lässt sich die Ergebnismenge weiter einschränken. Zum Beispiel liefert die Anfrage `//chapter[heading]` alle `<chapter>` zurück, die ein Kindelement `<heading>` haben. (Man vergleiche dies mit der Anweisung `//chapter/heading`, die als Ergebnis `<heading>` und keine `<chapter>` liefert!)

Weiterhin gibt es eine Reihe von Vergleichsoperatoren: Die Anfrage `//chapter[heading= "XML"]` liefert alle `<chapter>` Elemente zurück, deren Inhalt durch den String "XML" gegeben ist. Auch Attribute lassen sich über diesen Mechanismus abfragen. Die Anfrage `/document[@class = "H.3.3" \&\$author = "John Smith"]` liefert alle Dokumente zurück, deren Attribut `class` und deren Kindelement `<author>` mit den angegebenen Werten übereinstimmen.

Die hier vorgestellten Anfragen beschreiben nur einen kleinen Teil der Funktionalität von XPath. Eine vollständige Beschreibung liefert die Spezifikation [XPath].

Anwendung auf RDF

In diesem Abschnitt wird nun gezeigt, wie sich Anfragesprachen an XML für das Retrieval auf RDF einsetzen lassen und welche Probleme bei dieser Vorgehensweise auftreten.

Beispiel 1

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/" >
  <rdf:Description>
    <dc:creator>Stefan Kokkelink</dc:creator>
  </rdf:Description>
  ...
</rdf:RDF>
```

Die Anweisung `//dc:creator` liefert alle `dc:creator` zurück, unabhängig davon, an welcher Stelle diese Property in dem RDF Dokument vorkommt. Durch die Angabe eines vollständigen Pfades `/rdf:RDF/rdf:Description/dc:creator` lässt sich die Resultatmenge einschränken. Auch komplexere Anfragen lassen sich auf diese Weise an RDF Dokumente stellen:

Beispiel 2

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/" >
  <rdf:Description about="http://www.iuk-initiative.de/iwi/Document_A">
    <dc:creator>Hartmut Polzer</dc:creator>
  </rdf:Description>
  <rdf:Description about="http://www.iuk-initiative.de/iwi/Document_B">
    <dc:publisher>Springer-Verlag</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

Die Anfrage `//rdf:Description[dc:*]/@about` liefert alle URIs zurück, die durch Dublin Core Metadaten beschrieben wurden.

Beispiel 3

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/" >
  <rdf:Description ID="Document">
    <dc:publisher>Springer-Verlag</dc:publisher>
    <dc:creator>
      <rdf:Description>
        <vCard:FN>Hartmut Polzer</vCard:FN>
        <vCard:EMAIL>hartmut@gmx.de</vCard:EMAIL>
        .....
      </rdf:Description>
    </dc:creator>
  </rdf:Description>
</rdf:RDF>
```

Die Anfrage

`//dc:creator/rdf:Description/vCard:FN[../../../../dc:publisher="Springer-Verlag"]` liefert alle Autorennamen, die eine Arbeit im Springer-Verlag publiziert haben.

Das hier beschriebene Retrieval auf RDF läßt sich nur dann mit Hilfe von XML Anfragesprachen sinnvoll durchführen, wenn man die XML Struktur des zugrunde liegenden RDFs genau kennt. Das kann z.B. dann der Fall sein, wenn man die XML Präsentation selbst erstellt hat.

Im Allgemeinen eignet sich diese Form des Retrievals auf RDF nicht, wie die folgenden Überlegungen zeigen:

1. Für einen RDF Graphen kann es viele Repräsentationen in XML geben.
2. Das Datenmodell von RDF ist ein gerichteter Graph, das Datenmodell von XML ist im Wesentlichen ein Baum.

Beispiel 4

Die beiden folgenden RDF Dokumente sind äquivalent (d.h. der Parser erzeugt dasselbe Tripelset):

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/" >
  <rdf:Description about="http://www.iuk-initiative.de/iwi/Document_A">
    <dc:creator>Hartmut Polzer</dc:creator>
    <dc:publisher>Springer-Verlag</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/" >
  <rdf:Description about="http://www.iuk-initiative.de/iwi/Document_A">
    <dc:creator>Hartmut Polzer</dc:creator>
  </rdf:Description>
  <rdf:Description about="http://www.iuk-initiative.de/iwi/Document_A">
    <dc:publisher>Springer-Verlag</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

Es sollen nun alle Autoren gefunden werden, die im Springer-Verlag publiziert haben. In der ersten XML Version würde die Anfrage `//dc:creator[../dc:publisher="Springer-Verlag"]` das gewünschte Resultat bringen. In der zweiten XML Version gibt es jedoch auf diese Anfrage keinen Treffer, obwohl die zugrunde liegenden RDF Graphen identisch sind!

RQL: RDF Query Language

Im letzten Abschnitt wurde deutlich, dass sich Anfragesprachen für XML nur sehr beschränkt für das Retrieval auf RDF eignen. Die folgende Liste fasst den aktuellen Stand (Mai 2000) der Entwicklung von speziellen Anfragesprachen für RDF zusammen:

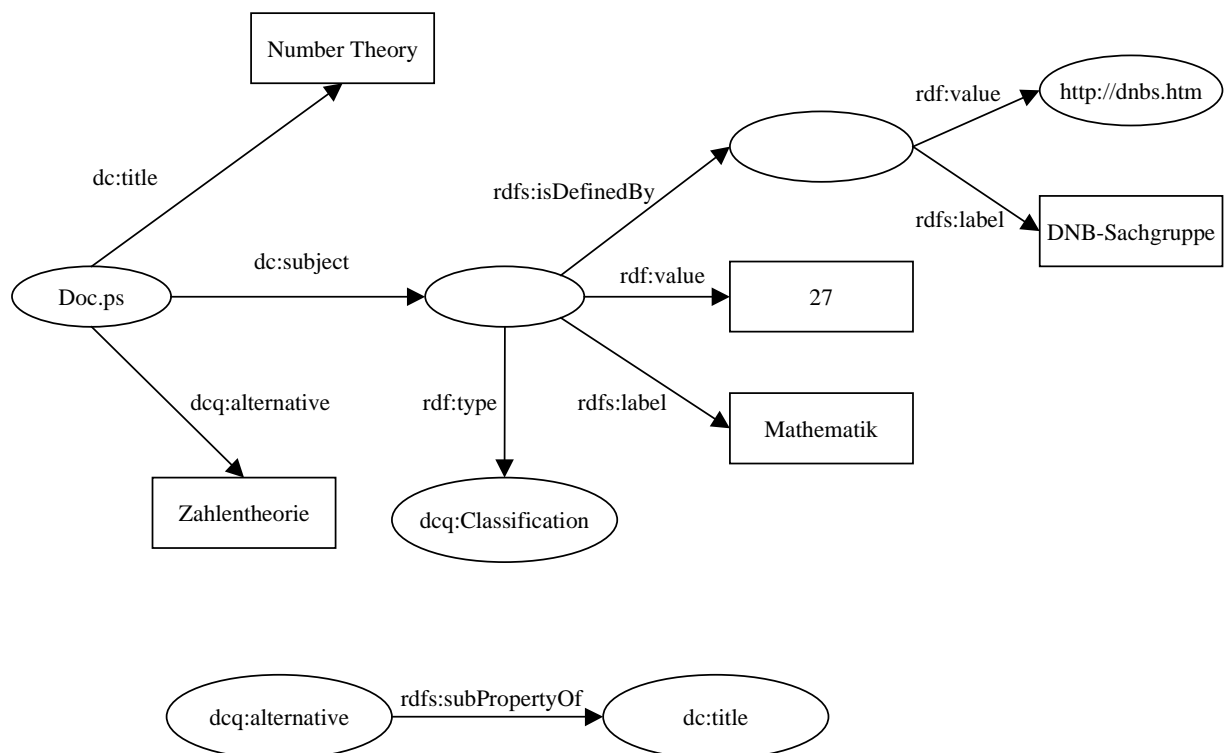
1. Metalog - The RDF querying system (1999-05-02) [Metalog]
2. RDF Query Specification (1998-12-03) [RDFQS]
3. A Query and Inference Service for RDF (1998-11-18) [QISRDF]
4. Querying Semistructured (Meta)data and Schemas on the Web: The case of RDF&RDFS (2000-02-25) [RQL]

Eine echte Unterstützung von RDF Schema bietet nur die in [RQL] spezifizierte Anfragesprache RQL. Aus diesem Grund wird sie hier vorgestellt, auch wenn es noch keine Implementierung von RQL gibt.

Die folgende Liste fasst die wesentlichen Eigenschaften von RQL zusammen:

- RQL basiert auf dem Graphenmodell von RDF.
- Die Syntax von RQL ähnelt SQL.
- RQL unterstützt RDF Schema , insbesondere die durch (subClassOf, subPropertyOf) definierten Relationen von Klassen und Properties.

Die unterschiedlichen Anfragemöglichkeiten von RQL werden anhand des folgenden RDF Graphen demonstriert.



Einfache Anfragen

Das Vorhandensein bestimmter Properties in einem RDF Graphen kann durch die Angabe ihrer URI überprüft werden. Zum Beispiel liefert die Anfrage `dcq:alternative` (die URI wird wie üblich mit Hilfe des jeweiligen Namespaces abgekürzt dargestellt) als Ergebnis das folgende Paar:

Source	Target
Doc.ps	Zahlentheorie

Die Anfrage `rdf:value` ergibt:

Source	Target
genID_2	http://dnbs.htm
genID_1	27

RQL berücksichtigt die Relation `subPropertyOf` der zugrunde liegenden RDF Schemas. Die Anfrage `dc:title` ergibt:

Source	Target
Doc.ps	Number Theory
Doc.ps	Zahlentheorie

Es werden also automatisch auch die Werte alle Subproperties von `dc:title` angezeigt! Man kann diese Funktionalität durch den `^` Operator verhindern. So würde die Anfrage `^dc:title` den einen Wert "NumberTheory" liefern.

Analog lassen sich Klassen abfragen. Auch hier wird immer bezüglich der Relation `subClassOf` expandiert.

Filter

Um die Graphstruktur von RDF auszunutzen, bietet RQL einen `select-from-where` Filter analog zu SQL an. Zum Beispiel ist die Anfrage `dc:title` äquivalent zur Anfrage **`select X,Y from {X}dc:title{Y}`**. Diese Anfrage setzt sich aus zwei Teilen zusammen: Zunächst werden die Objekte angegeben, die als Resultat geliefert werden sollen (hier Source X und Target Y), dann wird ein Teilgraph angegeben, in denen die Objekte auftreten müssen (hier **`where {X}dc:title{Y}`**). Möchte man nur die Ressourcen erhalten, von denen eine Property `dc:title` startet, so müsste man folgende Anfrage stellen: **`select X from {X}dc:title{Y}`**.

Es können noch weitere Bedingungen formuliert werden. Im folgenden Beispiel werden im Ergebnis alle Titel (`dc:title`) von Ressourcen angezeigt, deren `dc:subject` auf eine Ressource zeigt, deren `rdf:value` "Mathematik" ist:

```
select {Y} from {X}dc:title{Y},{X}dc:subject{Z}.rdf:value{W}  
where W="Mathematik"
```

Bezüglich des obigen RDF Graphen würde diese Anfrage als Ergebnis "Zahlentheorie" und "Number Theory" liefern. (Zur Erinnerung: Es wird bezüglich der Relation `supPropertyOf` expandiert.)

Auf diese Weise lassen sich sehr komplexe Anfragen realisieren. Für eine Übersicht über die gesamte Funktionalität von RQL sei auf die Spezifikation verwiesen.

Referenzen

[XPath]: XML Path Language (XPath) Version 1.0
<http://www.w3.org/TR/xpath>

[XQL]: XML Query Language (XQL)
<http://www.w3.org/TandS/QL/QL98/pp/xql.html>

[Metalog] Metalog
<http://www.w3.org/RDF/Metalog/>

[RDFQS]: RDF Query Specification
<http://www.w3.org/TandS/QL/QL98/pp/rdfquery.html>

[QISRDF]: A Query and Inference Service for RDF
<http://www.ilt.bris.ac.uk/discovery/rdf-dev/purls/papers/QL98-queryservice/>

[RQL]: Querying Semistructured (Meta)Data and Schemas on the Web:
The case of RDF & RDFS
<http://www.ics.forth.gr/proj/isst/RDF/rdfquerying.ps>