

SPARQL, linked-data

Námět na využití v projektech kombinujících
různé datové zdroje

Small or medium-scale focused research project (STREP)

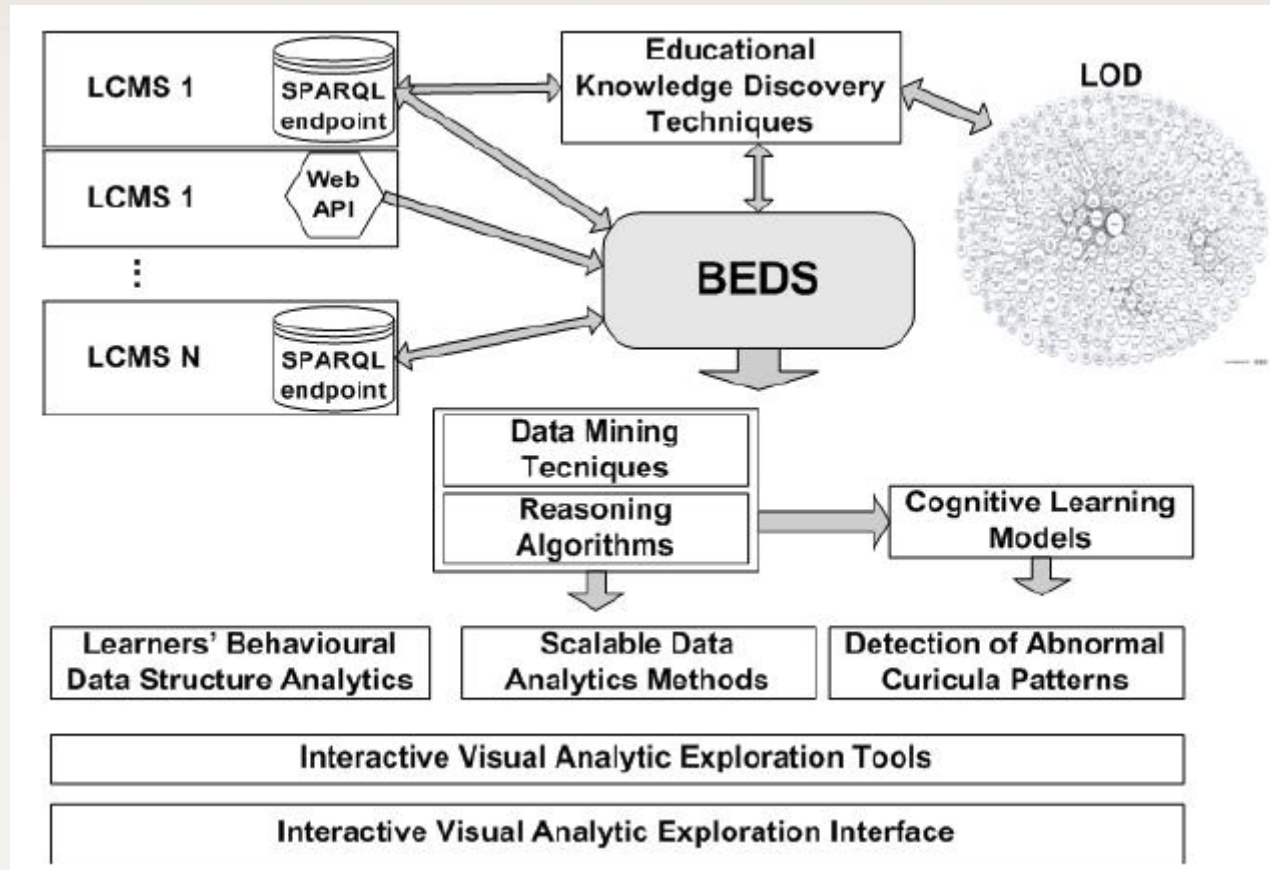
ICT Call 11

FP7-ICT-2013-11

**Big Educational Data Analytics
for Medical Educational Knowledge Discovery -
Medical Learning Analytics**

CHIRON¹





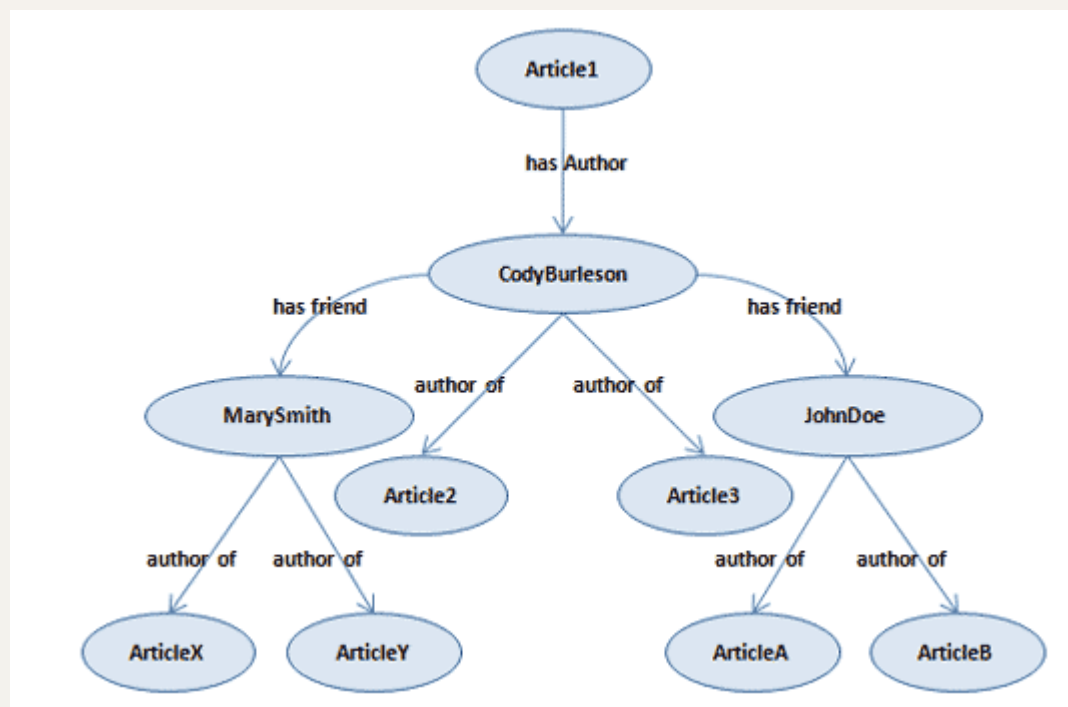
- klíčové technologie pro vývoj sémantického webu
- SPARQL: **S**imple **P**rotocol **A**nd **R**DF **Q**uery **L**anguage
 - je jazyk, vycházející z SQL, určený k manipulaci (primárně dotazování) s RDF databázemi
- RDF: **R**esource **D**escription **F**ramework
 - je datovým modelem pro metadata
- W3C:
 - standard pro RDF
 - Standard pro SPARQL
(1.0 – 2008, 1.1 – 03/2013)

Resource Description Framework

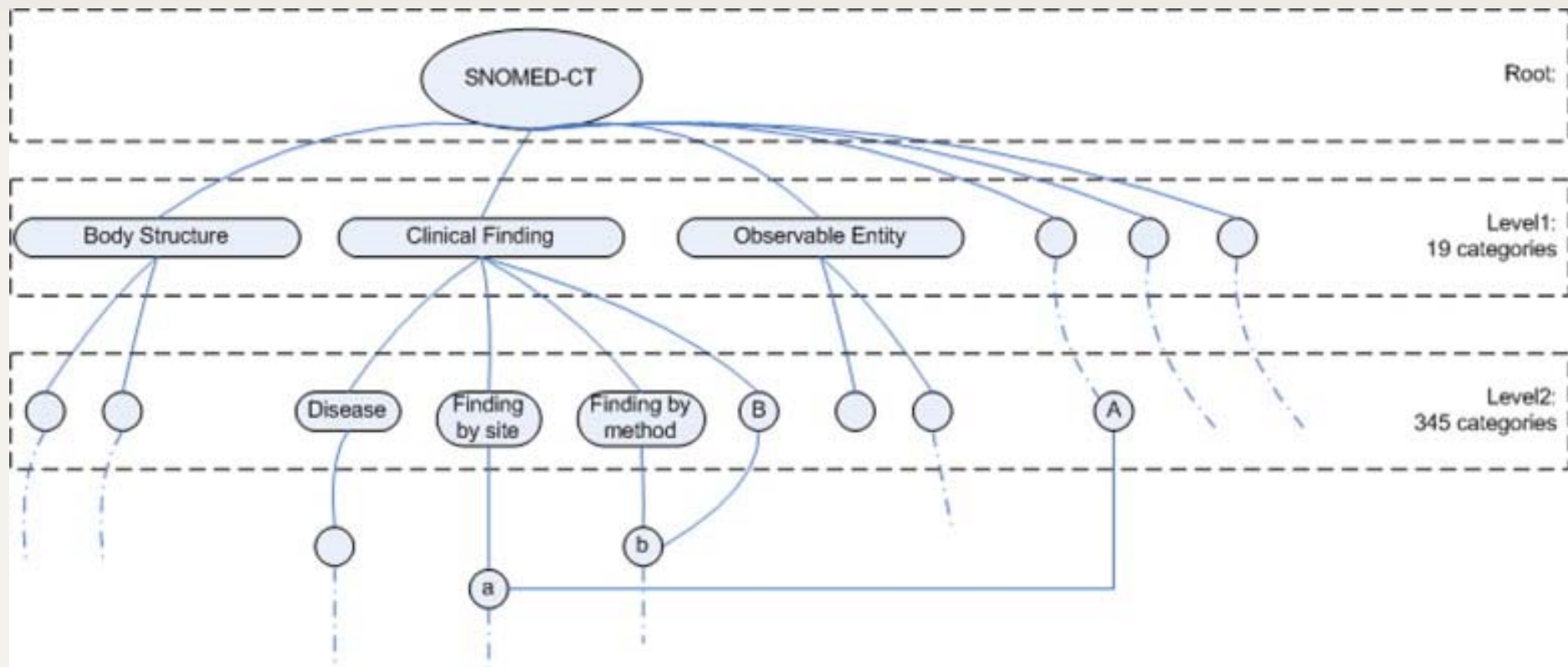
- Odpověď na poptávku po jednotném modelu pro výměnu informací v decentralizované heterogenní síti systémů.
- Pokud vyvíjíte aplikaci a chcete umožnit, aby informace byly sdíleny a využívány jinými aplikacemi, publikujte tyto informace v RDF.

Resource Description Framework

- model vyjadřování informací pomocí trojice:
 - podmět – přísudek – předmět (objekt)
- množina trojic = RDF graf. Jak serializovat?
 - RDF/XML
 - RDFa
 - Turtle
 - N-triples
 - ...



SNOMED-CT

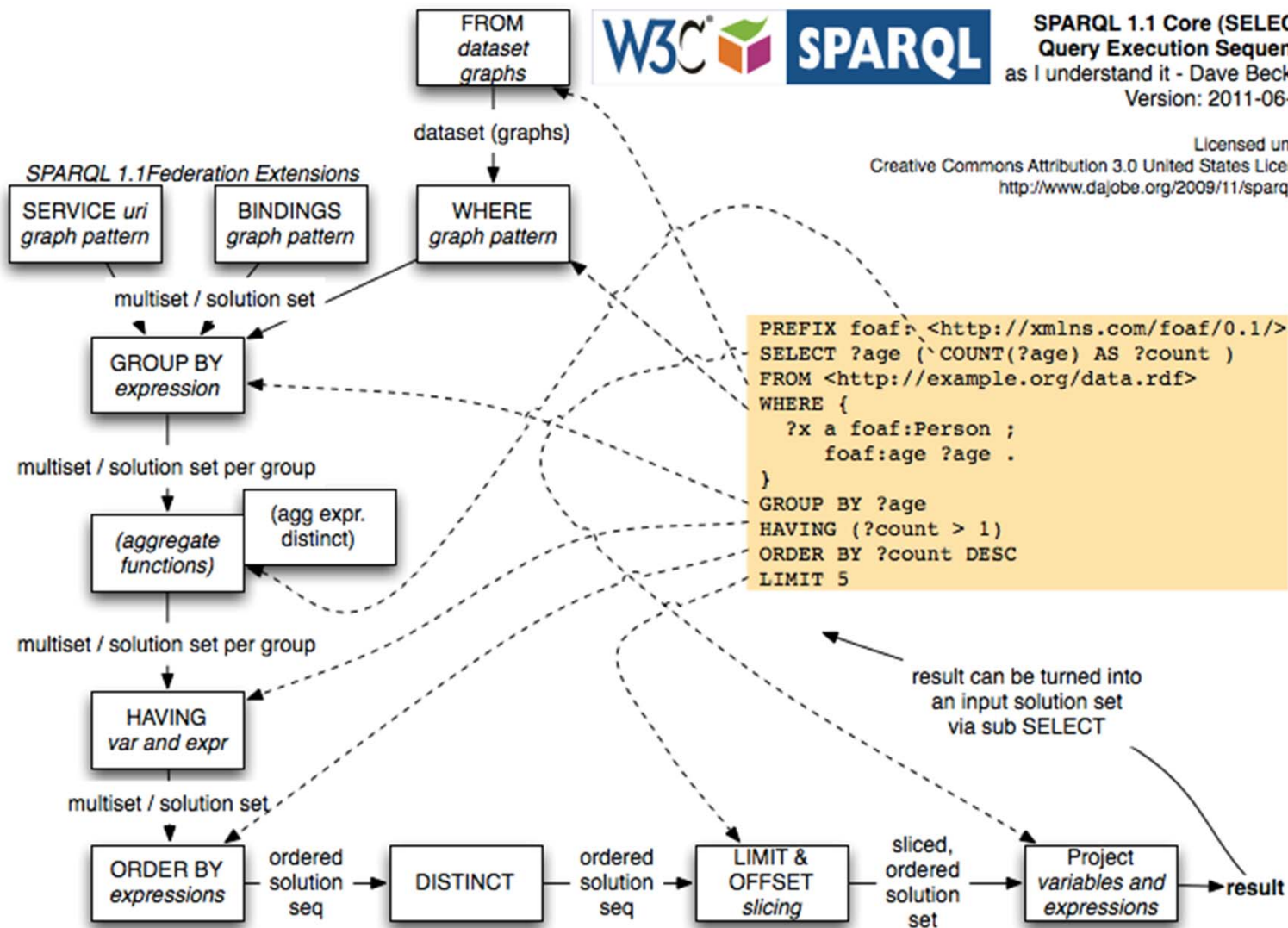


Federated query



SPARQL 1.1 Core (SELECT)
Query Execution Sequence
as I understand it - Dave Beckett
Version: 2011-06-16

Licensed under
Creative Commons Attribution 3.0 United States License
<http://www.dajobe.org/2009/11/sparql11/>



- I. Dbpedia: www.dbpedia.org
- II. Meducator: www.meducator.net
- III. GeoNames: www.geonames.org

<http://querybuilder.dbpedia.org/>

<http://www.w3.org/wiki/SparqlEndpoints>

sparqlib.php

Simple library to query SPARQL from PHP.
©2010-12 Christopher Gutteridge, University of Southampton.

Intro

This is a very simple RDF library to query SPARQL from PHP. It currently ignores language and datatype information to make it feel as similar as possible to the normal PHP SQL libraries.

Download: [sparqlib.php](#) (LGPL)

If you want to get started really quickly, the following command line will install sparqlib.php. You should run it in the same directory as where your PHP code resides.

```
curl -s http://graphite.ecs.soton.ac.uk/download.php/sparqlib.php -o sparqlib.php
```

Or get the [latest version from Github](#).

Also hosted on this site is [Graphite](#), a simple PHP library for querying RDF data.

Classic mysql_query style

The library provides functions very similar to mysql_* for comfort.

```
Code
<?php
require_once( "sparqlib.php" );

$db = sparql_connect( "http://rdf.ecs.soton.ac.uk/sparql/" );
if( !$db ) { print sparql_errno() . ": " . sparql_error(). "\n"; exit; }
sparql_ns( "foaf", "http://xmlns.com/foaf/0.1/" );

$sparql = "SELECT * WHERE ( ?person a foaf:Person . ?person foaf:name ?name ) LIMIT 5";
$result = sparql_query( $sparql );
if( !$result ) { print sparql_errno() . ": " . sparql_error(). "\n"; exit; }

$fields = sparql_field_array( $result );

print "<p>Number of rows: " . sparql_num_rows( $result ) . " results.</p>";
print "<table class='example_table'>";
print "<tr>";
foreach( $fields as $field )
{
    print "<th>$field</th>";
}
print "</tr>";
while( $row = sparql_fetch_array( $result ) )
{
    print "<tr>";
    foreach( $fields as $field )
    {
        print "<td>$row[$field]</td>";
    }
    print "</tr>";
}
,
```

[Intro](#)
[Quick Interface](#)
[Classic Interface](#)
[Object Interface](#)
[Rows](#)
[Capabilities](#)
[Contact](#)

<http://www.w3.org/wiki/SparqlImplementations>

- Clients
- Query engines
- Parsers

Sorry, I don't speak SPARQL – Translating SPARQL Queries into Natural Language

Axel-Cyrille Ngonga
Ngomo
Universität Leipzig, IFI/AKSW
PO 100920, D-04009 Leipzig
ngonga@informatik.uni-leipzig.de

Lorenz Böhmann
Universität Leipzig, IFI/AKSW
PO 100920, D-04009 Leipzig
buehmann@informatik.uni-leipzig.de

Christina Unger
Bielefeld University, CITEC
Universitätsstraße 21–23,
33615 Bielefeld
cunger@cit-ec.uni-bielefeld.de

Jens Lehmann
Universität Leipzig, IFI/AKSW
PO 100920, D-04009 Leipzig
lehmann@informatik.uni-leipzig.de

Daniel Gerber
Universität Leipzig, IFI/AKSW
PO 100920, D-04009 Leipzig
dgerber@informatik.uni-leipzig.de

ABSTRACT

Over the past years, Semantic Web and Linked Data technologies have reached the backend of a considerable number of applications. Consequently, large amounts of RDF data are constantly being made available across the planet. While experts can easily gather information from this wealth of data by using the W3C standard query language SPARQL, most lay users lack the expertise necessary to proficiently interact with these applications. Consequently, non-expert users usually have to rely on forms, query builders, question answering or keyword search tools to access RDF data. However, these tools have so far been unable to explicate the queries they generate to lay users, making it difficult for these users to i) assess the correctness of the query generated out of their input, and ii) to adapt their queries or iii) to choose in an informed manner between possible interpretations of their input. This paper addresses this drawback by presenting SPARQL2NL, a generic approach that allows verbalizing SPARQL queries, i.e., converting them into natural language. Our framework can be integrated into applications where lay users are required to understand SPARQL or to generate SPARQL queries in a direct (forms, query builders) or an indirect (keyword search, question answering) manner. We evaluate our approach on the DBpedia question set provided by QALD-2 within a survey setting with both SPARQL experts and lay users. The results of the 115 filled surveys show that SPARQL2NL can generate complete and easily understandable natural language descriptions. In addition, our results suggest that even SPARQL

Keywords

Natural language generation, query verbalization, SPARQL

1. INTRODUCTION

An ever-growing number of applications rely on RDF data as well as on the W3C standard SPARQL for querying this data. While SPARQL has proven to be a powerful tool in the hands of experienced users, it remains difficult to fathom for lay users. To address this drawback, approaches such as question answering [28], keyword search [25] and search by example [18] have been developed with the aim of hiding SPARQL and RDF from the user. Still, these approaches internally construct SPARQL queries to address their data backend, without providing lay users with a possibility to check whether the retrieved answers indeed correspond to the intended information need. Consider for example the natural language question What is the birth date of Li Ling?, for which TBSL [28] returns more than 50 possible interpretations, including the birth date of the pole vaulter Li Ling and the age of the sinologist Li Ling. Since each of the interpretations is realized as a SPARQL query, a lay user cannot pinpoint the set of results that correspond to the person he is actually interested in, nor can he easily detect the source of possible errors. Similar problems occur in keyword-based systems. For example, the keywords Jenny Runacre husbands leads to SINA [25] generating queries for the husbands of Jenny Runacre as well as for the role of Jenny Runacre in the movie "The Husbands". We address this drawback by presenting SPARQL2NL, a novel approach