

SPARQLBot: The Semantic Web Command Line (Scripting Challenge Submission)

Benjamin Nowack

semsol, Bielefelder Str. 5, 40468 Düsseldorf, Germany
bnowack@semsol.com

Abstract. SPARQLBot is an RDF-driven agent that loads structured information from the Web and reacts to user-defined questions and commands via an IRC interface. The bot is implemented using a small number of PHP scripts and ARC, an open-source PHP/MySQL-based RDF system for storage and query functionality.

1 Motivation

SPARQLBot was developed during a single-day coding session to demonstrate a) the potential of tools that support structured and linked Web data, b) the developer-friendliness of SPARQL, and c) how very little custom code can be used to build useful applications. It consists of three core user interface components: An HTML-based command editor that simplifies the definition of custom bot operations, a long-running process that listens to command calls on an IRC channel¹, and a standard SPARQL endpoint for simplified debugging of SPARQL Queries and HTTP-based data access.

2 Implementation

SPARQLBot is built on top of two open-source toolkits. For core RDF functionality (RDF storage, parsing, microformats extraction, querying, etc.), it uses ARC², a light-weight RDF toolkit written in PHP, and the user-facing applications are implemented as Trice³ modules. Trice is a Web development framework that extends ARC with session management, HTML forms processing, IRC access, themes, and similar standard Web CMS functionality.

¹ <http://semsol.org/semcamp/sparqlbot>, to be moved to <http://sparqlbot.semsol.org/> soon

² #sparqlbot on <http://irc.freenode.net/>

³ <http://semsol.org/semcamp/sparqlbot/sparql>, to be moved to <http://sparqlbot.semsol.org/bot/sparql> soon

⁴ <http://arc.semsol.org/>

⁵ <http://trice.semsol.org/>

2 Benjamin Nowack

The SPARQLBot-specific code consists of only four custom PHP Classes (~25KB / 800 LOC altogether). A generic RequestHandler dispatches HTTP requests to the three user-facing Sub-Handlers (the IRC ProcessHandler that implements the bot, a Command Editor, and the SPARQL endpoint). Only very few commands had to be built directly into the bot (e.g. "quit", or "smush"). As ARC supports LOAD, INSERT, and DELETE via SPARQL, both read and write operations can be defined using the command editor.

3 Examples: XFN Lookups

XFN⁶, the "Xhtml Friends Network" is a widely deployed microformat⁷ to specify a person's social network in XHTML pages. SPARQLBot's RDF toolkit can convert XFN markup to RDF triples and then make them accessible to SPARQL queries. The code below shows the command's essential parts which can be defined using an online form.

Command Pattern (a regular expression):

```
(.*)'s? (contact|friend|me)s
```

Triggered SPARQL Query (*\$i* can be used for command matches):

```
SELECT DISTINCT ?name WHERE {  
  {?res foaf:nick "$1"} UNION {?res foaf:name "$1"}  
  ?res xfn:$2 [ foaf:name ?name ] .  
}
```

Result Template (*\$var* can be used for result bindings):

```
$nick, I found {$name, }
```

Example Conversation:

```
<bengee> sparqlbot, load http://twitter.com/bengee  
<sparqlbot> 290 triples loaded in 4.9s seconds  
<bengee> sparqlbot, smush  
<sparqlbot> OK  
<bengee> sparqlbot, Benjamin Nowack's contacts  
<sparqlbot> bengee, I found Danny Ayers, Tom,  
Gregory Williams, Arto Bendiken, Paul Miller, John  
Breslin,Uldis Bojars, Alexandre Passant, ...
```

⁶ <http://gmpg.org/xfn/>

⁷ <http://microformats.org/>