NITELIGHT: A Graphical Editor for SPARQL Queries

Alistair Russell
School of Electronics and Computer Science
University of Southampton
Southampton,
SO17 1BJ, United Kingdom
ar5@ecs.soton.ac.uk

Paul R Smart
School of Electronics and Computer Science
University of Southampton
Southampton,
SO17 1BJ, United Kingdom
ps02v@ecs.soton.ac.uk

ABSTRACT

Query formulation is a key aspect of information retrieval, contributing to both the efficiency and usability of many semantic applications. A number of query languages, such as SPARQL, have been developed for the Semantic Web; however, there are, as yet, few tools to support end users with respect to the creation and editing of semantic queries. In this paper we present NITELIGHT, a graphical tool for semantic query design. NITELIGHT uses a Visual Query Language (VQL), called vSPARQL, which provides graphical formalisms for SPARQL query specification. NITELIGHT is a highly reusable Web-based component, and it can be easily embedded in a variety of different Web applications. This paper provides an overview of the NITELIGHT tool and the vSPARQL specification.

Keywords

sparql, visual query system, semantic web, graphical query language, ontology, owl.

1. INTRODUCTION

Information retrieval is a key capability on the Semantic Web, contributing to both the efficiency and usability of many semantic applications. The availability of semantic query languages, such as SPARQL [1], is an important element of information retrieval capabilities; however, query developers are likely to benefit from the additional availability of tools that assist them with respect to the process of query formulation (i.e. the process of creating or editing a query). Ideally, query formulation tools should avail themselves of user interaction capabilities that contribute to the efficient design of accurate queries, while maximally exploiting the power and expressivity provided by the constructs of the target query language.

Most attempts to support the user with respect to query formulation have focused on graphical or visual techniques in the form of Visual Query Systems (VQSs) [2]. VQSs provide a number of advantages relative to simple text editors. Most obviously, such systems support the user in developing syntactically valid queries: they serve to constrain or guide editing actions so as to militate against the risk of lexical or syntactic errors. Other potential advantages include improved efficiency, understanding and reduced training requirements.

In this paper we introduce a graphical tool for semantic query construction based on the SPARQL language specification [1]. SPARQL is one of a number of query languages that have been proposed for the Semantic Web (others include RQL [3] and RDQL [4], although only SPARQL benefits from W3C endorsement). The tool we present in this paper (called NITELIGHT) enables users to create SPARQL queries using a set of graphical notations and GUI-based editing actions. The tool is intended primarily for users that already have some familiarity with SPARQL (although it could also potentially serve as a support tool for novice users who aim to acquire SPARQL expertise).

2. VISUAL SPARQL

The development of a graphical tool for SPARQL query formulation necessarily entails the development of a set of graphical notations that support the visual representation of SPARQL query components. Following an analysis of the SPARQL syntax specification [1], we developed a set of graphical notations to support the representation of SPARQL queries. These notations comprise the basis of a SPARQL VQL that we refer to as vSPARQL.

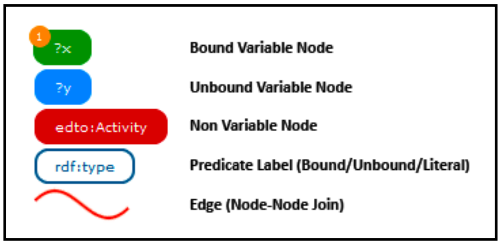
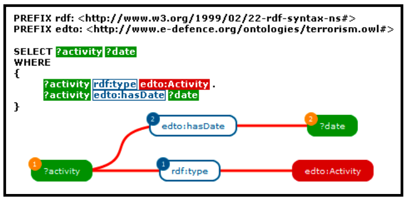


Figure 1. Core vSPARQL graphical notations.

Because SPARQL queries exploit the triple-based structure of RDF models, graph-based representations comprising a sequence of graphical nodes and links can be used to represent the core of most SPARQL queries. The nodes in this case correspond to the subject and object elements of an RDF triple, while the links correspond to the predicates. Colour is used to differentiate between different types of subject and object nodes (e.g. Bound Variable Nodes) (see Figure 1). Bound Variable Nodes represent variables whose value bindings are returned as part of the query resultset, Unbound Variable Nodes are variables that are not returned in the resultset (they are used as part of the query execution process) and Non-Variable Nodes are nodes that represent a URI, literal value or blank node. All nodes are associated with a label that indicates the URI, literal value or query variable represented by the node. Figure 2 shows a simple SPARQL query expressed using vSPARQL formalisms.



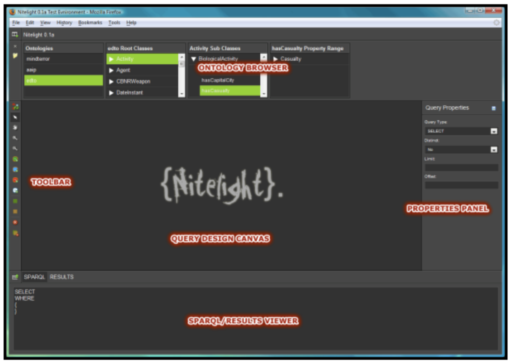
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX edto: <http://www.e-defence.org/ontologies/terrorism.owl#>

SELECT ?activity ?date
WHERE
{
        ?activity rdf:type edto:Activity .
        ?activity edto:hasDate ?date
}
```

Figure 2. vSPARQL representation of a SPARQL SELECT query.

3. NITELIGHT

The NITELIGHT tool is a Web-based application written entirely in JavaScript. The main user interface (see Figure 3) has five components, each of which works together to provide a visually compelling environment for graphical query formulation. The main user interface components are briefly described in subsequent sections.



Query Type: SELECT
Distinct: No
Limit:
Offset:

```
SELECT
WHERE
{
}
```

Figure 3. The NITELIGHT tool main user interface.

3.1 Query Design Canvas

The Query Design Canvas (see Figure 3) is the center-piece of the NITELIGHT tool. It provides a canvas for the graphical rendering of SPARQL queries using the constructs of the vSPARQL language. Many of the vSPARQL constructs, once rendered on the Query Design Canvas, are selectable objects that can be edited to change the underlying structure of the SPARQL query.

3.2 Ontology Browser

To facilitate the process of query formulation, and to provide users with a starting point for query specification, the NITELIGHT tool includes an Ontology Browser component (see Figure 3). The browser allows users to view the contents (classes and properties) of any loaded ontology. It also enables users to drag and drop ontology elements onto the Query Design Canvas as part of the query construction process.

3.3 Quick Toolbar

The Quick Toolbar provides access to commonly used tools for manipulating the Query Design Canvas and its graphical query contents.

3.4 Properties Inspector Panel

The Properties Inspector Panel allows users to view and edit the properties associated with a selected vSPARQL object.

3.5 SPARQL Syntax Viewer/Query Results Viewer

The SPARQL Syntax Viewer provides a read-only text-based view of the SPARQL query being edited in the NITELIGHT tool. The viewer also allows users to execute the current query against a SPARQL endpoint and view the resulting query output.

4. WEB APPLICATION EXTENSION

NITELIGHT is written using the standard prototypeJS [5] and script.aculo.us [6] libraries. These allow for 3rd party application developers to integrate the NITELIGHT component into custom built applications using lightweight code development techniques.

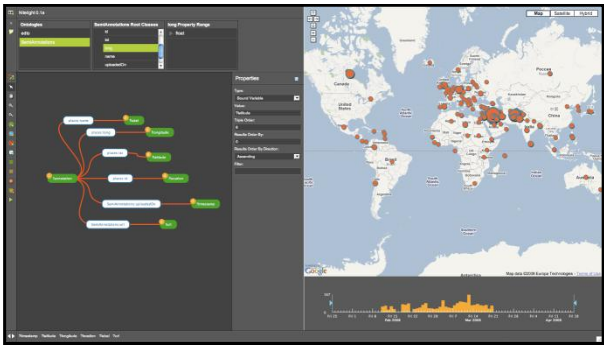


Figure 4. The NITELIGHT component used as a query generator in a 3rd party Web application.

Figure 4 shows the NITELIGHT tool embedded in a custom Google Maps mash-up application. The use of the NITELIGHT tool in this application context is intended to support the user with respect to semantic query construction against an application-specific ontology. The results of query execution are visualized using both timeline and map-based UI components.

5. FUTURE WORK

The NITELIGHT tool is currently being developed for open source distribution. Future research efforts will focus on user evaluation and comparative analyses with other graphical query designers such as iSPARQL [7] and SPARQLViz [8].

6. CONCLUSIONS

This paper has presented a Web-based graphical editing tool for semantic query design, called NITELIGHT. NITELIGHT combines a number of features to support end-users with respect to the creation of SPARQL queries. These include an ontology browser, an interactive graphical design surface, a SPARQL-compliant VQL (called vSPARQL), a SPARQL syntax viewer and an integrated semantic query results browser. The tool is currently at an early stage of development, but is ultimately intended for open source distribution and use within a variety of 3rd party Web applications.

7. REFERENCES

[1] http://www.w3.org/2001/sw/DataAccess/rq23/ - "SPARQL Query Language for RDF"

[2] Catarci, T., Costabile, M. F., Levialdi, S., and Batini, C. Visual Query Systems for Databases: A Survey. Journal of Visual Languages and Computing 8, 2 (1997), 215-260.

[3] Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., and Scholl, M., "RQL: a declarative query language for RDF," in 11th International World Wide Web Conference. Budapest, Hungary, 2002, pp. 592-603.

[4] http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/ - "RDQL - A Query Language for RDF"

[5] http://prototypejs.org/ - “Prototype JavaScript Framework”

[6] http://script.aculo.us/ - “script.aculo.us – Web 2.0 JavaScript”

[7] http://demo.openlinksw.com/isparql/ - “OpenLink iSPARQL”

[8] http://sparqlviz.sourceforge.net/ - “SPARQLViz Project Page”