visKWQL, a Visual Renderer for a Semantic Web Query Language

Andreas Hartl, Klara Weiand, François Bry

Institute for Informatics, University of Munich
Oettingenstr. 67, 80538 München, Germany
http://pms.ifi.lmu.de

Abstract. Querying a Wiki must be simple enough for beginning users, yet powerful enough to accommodate experienced users. To this end, the keyword-based KiWi query language (KWQL) supports queries ranging from simple lists of keywords to expressive rules for selecting and reshaping Wiki (meta-)data. In this demo, we showcase visKWQL, a visual interface for the KWQL language aimed at supporting users in the query construction process. visKWQL and its editor are described, and their functionality is illustrated using example queries. The editor provides guidance throughout the query construction process through hints, warnings and highlighting of syntactic errors. The editor enables round-tripping between the twin languages KWQL and visKWQL, meaning that users can switch freely between the textual and visual form when constructing or editing a query. It is implemented using HTML, JavaScript, and CSS, and can thus be used in (almost) any web browser without any additional software.

1 Introduction

Web query languages like XQuery and SPARQL allow for the precise and targeted selection and transformation of Web data. While these languages are powerful tools, they require their users to be knowledgeable about the language itself as well as the structure and schema of the queried data. This requirement excludes a large part of the web's user base (and thus the potential user base of the Semantic Web) from the benefits of these languages and the functionality they provide.

Visual languages have two advantages over textual languages that specifically benefit beginning users [2]: First, their visual structure can make them easier to learn and understand than textual languages. Secondly, editors for visual languages can support users in the creation of valid queries by providing guidance and preventing editing operations that would result in incorrect queries.

This demonstration presents visKWQL[1], a visual query interface for the Semantic Wiki KiWi [4]. visKWQL is not so much a separate query language but

[1] A detailed demonstration description as well as a demo of visKWQL are available at http://www.pms.ifi.lmu.de/visKWQL/

rather a visual rendering of KWQL, the keyword-based KiWi query language. KWQL [1] is a rule-based query language based on the label-keyword paradigm that combines a low entry barrier with powerful querying and ease of use of keyword search with advanced features and capabilities as used in traditional query languages in order to accommodate users with varying levels of expertise.

visKWQL aims at extending textual KWQL—which itself has been designed to be easy to use—to achieve two cohesive and tightly integrated querying modi in the KiWi Wiki and enable user-friendly and powerful querying.

The work described here has been presented before as a demonstration at the 2010 WWW conference [3].

2 visKWQL

visKWQL provides a visual alternative to textual KWQL. It fully supports KWQL in that every KWQL query can be expressed as an equivalent visKWQL query. Further, in order to avoid introducing additional constructs and thus additional complexity, visKWQL stays close to the textual language in its visual representation.

2.1 Visual Formalism

visKWQL uses a form-based approach, in which all KWQL elements are represented as boxes, and associations between them are represented as nestings (see Figure 1 for an example). Boxes consist of a *label*, in which the name of the represented KWQL element is included, and a *body*, which can hold child boxes.

This approach stays close to KWQL's textual structure, making it easier to learn visKWQL and to translate between the two representations; it also lends itself well to rendering in HTML.

2.2 Round-tripping

One of the key features of visKWQL is round-tripping, to achieve a tight coupling between KWQL and visKWQL.

Whenever the user makes a change to the visual query, the change is immediately represented in the textual version. The textual query can further be edited and parsed by the system to display it in its visual form.

This allows the user to make changes in the representation of his choice at any time during the query construction process, to import KWQL queries easily into visKWQL, and has the additional benefit of teaching the user KWQL while he experiments with the visKWQL editor.

2.3 User Guidance

User support in visKWQL is provided via tooltips, error prevention, and error and problem display and correction.

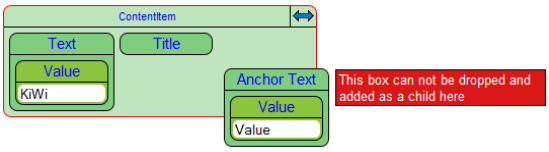

Fig. 1. Information hiding and error prevention

Tooltips: A text area below the workspace displays an explanation of the KWQL element represented by the box currently under the mouse cursor.
Error Prevention: A large number of syntactic errors result from invalid box nestings, and can be actively prevented by the editor during drag and drop actions. When a box is being dragged, the system continuously checks the validity of a child inclusion or a type switch with the box underneath it.

If dropping the box in its current location would result in a syntactic error, the border of the box underneath it is colored red, and a tooltip informs the user that he may not drop the box (see Figure 1).
Error Reporting and Correction: Some errors cannot be prevented during editing. These include variable names or values containing invalid characters, empty strings, misplaced operators and references to undefined variables.

After every user action, the query is checked for such errors. When an error is found, the label of the node is colored red and a tooltip indicating the error is displayed next to it.To make these errors easy to locate within the query, even if the erroneous node is currently hidden, the labels of all its parent boxes will also be colored red, and display a tooltip that a child box contains an error.

Errors that are less severe and can be corrected automatically, like empty boxes, cause the box label to be colored orange. A tooltip and a message below the workspace inform the user about the source of the problem.

3 Acknowledgements

The research leading to these results is part of the project "KiWi - Knowledge in a Wiki" and has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 211932.

References

1. F. Bry and K. Weiand. Flavours of KWQL, a keyword query language for a semantic wiki. In *Proceedings of SOFSEM 2010*, 2010.
2. T. Catarci, M. Costabile, S. Leviladi, and C. Batini. Visual Query Systems for Databases: A Survey. *Journal of Visual Languages and Computing*, 8(2), 1997.
3. A. Hartl, K. A. Weiand, and F. Bry. visKWQL, a visual renderer for a semantic web query language. In *WWW*, 2010.
4. S. Schaffert, J. Eder, S. Grünwald, T. Kurz, and M. Radulescu. Kiwi - a platform for semantic social software. In *ESWC*, 2009.