

KWQL, Querying for Social Semantic Software



François Bry, Klara Weiland
University of Munich

Semantic Wiki characteristics

Collaboration

- ▶ Inconsistencies
- ▶ Disagreement

Work in progress

- ▶ Emerging content
- ▶ Emerging semantics

Social relationships

- ▶ Varying user expertise
- ▶ Connections between users

Heterogenous data

- ▶ Formal and informal annotation
- ▶ Text and multimedia
- ▶ Document structure

KiWi Querying: KWQL

KWQL can access all elements the user interacts with

- ▶ Combination of selection criteria from several data sources in one query
- ▶ Combined querying of text, annotation and meta data
- ▶ Querying of informal to formal annotations
- ▶ SPARQL integration
- ▶ Querying of user approval and conflicts

KWQL Examples

Java

Select documents containing "Java"

author: "Mary"

Select documents authored by Mary or containing tags assigned by Mary

ci(text:Java OR (tag(name:XML) AND author:Mary))

Select documents that either have "Java" in their text or that have the tag "XML" and were authored by Mary

ci(tag(name:Java) link(target:ci(title:Lucene) tag(name:uses)))

Select documents with the tag "Java" that contain a link tagged "uses" to a document with the title "Lucene"

ci(author:\$A text:\$A OPTIONAL tag(name:\$T))

Bind variable \$A to the names of authors whose name also appears in the text and, if the document has any tags, bind variable \$T to their names.

\$Y@ci(URI:\$X) SPARQL:(SELECT ?Y WHERE \{ ?X http://purl.org/dc/elements/1.1/title> ?Y .\})

Use SPARQL to retrieve all document titles

ci(title:Contents text:(\$A "-" ALL(\$T,"")) @ ci(title:\$T author:\$A)

Retrieve the titles and authors of all documents and display them in a new document

A Content Item is the basic part of KiWi. Every Content Item is a Wiki page. A Content Item can contain Text, Text Fragments, Links, and other Content Items

Welcome to the KiWi Query Builder. Start by selecting an element in the menu and adding children elements to it. You can delete elements any time by dragging them outside the colored workarea.

KWQL: Parse

Varying complexity of queries

- ▶ Simple label-keyword queries
- ▶ Conjunction/disjunction/optional
- ▶ Structural queries
- ▶ Link traversal

Aggregation and construction

- ▶ Data construction
- ▶ Embedded queries
- ▶ Continuous queries

Result navigation

- ▶ Ranking
- ▶ Faceted Browsing

Simple query construction

- ▶ visKWQL, KWQL's visual counterpart
- ▶ Query by example paradigm
- ▶ Roundtripping between KWQL and visKWQL

KWQL as a basis for reasoning

Querying induces reasoning

- ▶ Data selection is the basis of further processing
- ▶ Rules give rise to a simple kind of reasoning
- ▶ Users should not need to learn two separate languages

Current Status

- ▶ KWQL grammar defined
- ▶ Parser and semantic verification implemented
- ▶ Implementation underway

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.