Jörg Schönfisch, Florian Lautenbacher, Julian Lambertz, Willy Chen

Softplant Living Semantic Platform

A platform for semantic applications
ISWC Industry Track 2011
Softplant GmbH
Living Enterprise Architecture

- German (Ingolstadt and Munich) based IT-service provider
- We enable a sustainable alignment of our customer’s IT to constantly evolving business requirements
- Service Portfolio:
  - Enterprise architecture management
  - Project services, e.g. requirements and test management
- We offer:
  - support in introducing, organizing and developing a living enterprise architecture
  - active support during planning and implementation of IT projects
  - build bridges between strategic IT management and operative project implementation
- Ideas and innovation are our driving force for changes
Agenda

- Motivation
- Preliminaries
- Softplant Living Semantic Platform
- Demo
- Evaluation
- Outlook
Motivation

- We aim for developing business applications based on lightweight ontologies focussing on query answering as primary use case
- Possible application areas:
  - data integration
  - dependency management
  - ...
- Standards for ontology representation (OWL2-QL) and querying (SPARQL) have been published as W3C recommendation
- Relational databases are widespread within enterprises with mature operational services (performance tuning, backup, ...)
- Algorithms and prototypes for OWL2-QL reasoning exist, but not ready for commercial use in closed source products and customer projects
- Missing integration of collaborative ontology editing features

→ We require a platform to do this!
Preliminaries

OWL2 Language Profiles

- OWL is the Web Ontology Language standardized by the W3C
- Version 2.0 includes several profiles:
  - OWL2 EL
    - captures the expressive power used by many large-scale ontologies and
    - for which the following reasoning problems can be decided in polynomial time: satisfiability, subsumption, classification, and instance checking.
  - OWL2 QL
    - enables conjunctive queries to be answered in LogSpace using standard relational database technology
    - suitable for applications where relatively lightweight ontologies are used to organize large numbers of individuals and where it is useful or necessary to access the data directly via relational queries (e.g., SQL)
  - OWL2 RL
    - support rule-based reasoning engines
    - restricts the use of expressions (no disjoint, negation, reflexive object properties)

Reference of 1 and image source: OWL2 Overview, www.w3.org/TR/owl2-overview
Preliminaries

OWL2-QL - Query Rewriting

Advantages of OWL2-QL and the use of rewriting algorithms:

- Reasoning during query time, not during load time
- Changes to instances have no impact on the internal state of the reasoner
  - no additional materialization or reasoning step
- Concurrent editing and reasoning
- Allows handling of very large A-Boxes
- Based on robust RDBMS
- Multiple rewriting algorithms with different characteristics were proposed
Preliminaries

Rewriting Algorithms

Existing rewriting algorithms either produce a union of conjunctive queries (UCQ) or a non-recursive (nr-) Datalog program.

- PerfectRef/CGLLR [1] [5]
  - first proposed rewriting algorithm
- RQR/REQUIEM [2]
  - optimized version of PerfectRef resulting in smaller rewritings
- Presto [3]
  - alternative approach using nr-Datalog programs
- TGD-rewrite [4]
  - optimized rewriting to take RDBMS characteristics into account
## Preliminaries

Other OWL2-QL Capable Reasoner and Triple Stores

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Reasoning</th>
<th>License</th>
<th>Activity</th>
<th>Editing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owlgres</td>
<td>rewriting</td>
<td>query time</td>
<td>GPL (AGPL 3)</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>REQUIEM</td>
<td>rewriting</td>
<td>query time</td>
<td>LGPL</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Quill</td>
<td>rewriting</td>
<td>query time</td>
<td>unknown</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Nyaya/IRIS</td>
<td>rewriting</td>
<td>query time</td>
<td>LGPL</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>MASTRO / QuOnto</td>
<td>rewriting</td>
<td>query time</td>
<td>evaluation only</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Stardog (beta)</td>
<td>RDF triple store</td>
<td>query-time</td>
<td>commercial</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>OWLIM</td>
<td>RDF triple store</td>
<td>load time</td>
<td>commercial</td>
<td></td>
<td>triple-based</td>
</tr>
<tr>
<td>OpenLink Virtuoso</td>
<td>RDF triple store</td>
<td>query-time</td>
<td>commercial</td>
<td></td>
<td>triple-based</td>
</tr>
<tr>
<td>Allegro Graph</td>
<td>RDF triple store</td>
<td>query time</td>
<td>commercial</td>
<td></td>
<td>triple-based</td>
</tr>
<tr>
<td>Oracle Sem. Tech.</td>
<td>RDF triple store</td>
<td>load-time</td>
<td>commercial</td>
<td></td>
<td>triple-based</td>
</tr>
</tbody>
</table>
Preliminaries

MASTRO System

- Similarly to our implementation, based on DL-Lite with query-time reasoning through query rewriting and relational databases to store the A-Box
- However, the Mastro System focuses on different aspects and partially employs different technologies:
  - Only integration of external data sources through mappings, no data management (e.g. insertions or updates of instances)
  - Based on the description logic DL-Lite$_{A,\text{id}}$, not the OWL2 standard
  - Proprietary API and query interface, no SPARQL support
  - PerfectRef algorithm for rewriting, which produces UCQs contrary to nr-Datalog programs in Presto and in general larger rewritings
Softplant Living Semantic Platform

Goals and Features

- **Goals:**
  - Integrating ontology editing, searching and reasoning within a single platform
  - Focus on query answering as primary use case
  - Ready-to-use in business applications
  - Lightweight and extendable platform

- **Features:**
  - Scalable query answering for large A-Boxes based on OWL2-QL subset and relational database management systems (RDBMS)
  - Collaborative ontology editing (A-Box, T-Box)
  - Concurrent reasoning and editing
  - Fulltext search
  - Usage of robust and mature technologies
  - Open for extensions

- We chose Presto for our implementation, as it produces the smallest set of rewritings in a reasonable amount of time (TGD-rewrite was not published when we started the project).
Softplant Living Semantic Platform
System architecture

Motivation | Preliminaries | Softplant Living Semantic Platform | Demo | Evaluation | Outlook

Pure Java Application
UI built with Google Web Toolkit (GWT)

Editing Services
- A-Box Editing
- T-Box Editing
- OWL Import & Export (OWL API)

Fulltext Search Services
- Search UI
- Lucene based Search Engine
- Lucene

SPARQL Querying Services
- Ontology Visualizations
- Facted Search
- OWL2-QL Query Rewriting (Presto Algorithm), Sesame SPARQL Parser

Persistence Mapper
- CRUD Operations on RDBMS (Hibernate)
- (any) RDBMS
- Tested with Oracle, Postgresql, HSQLDB

CRUD Operations
- Hibernate
Demo

Latest GUI-Screenshots (ongoing work)
Evaluation
First Comparison with a Triple Store

- Performance comparison with mature triple store
- Benchmark: LUBM ontology

→ Performance and Scalability promising, even without optimizations
Outlook

There are still a lot of things to do:

- Thorough tests
- Performance improvements
  - concurrent reasoning
  - optimize query rewriting
  - caching
- Improve and complete editing and searching features
- More tests!

- We plan to make the software available under an open source license (decision pending – depending on your interest!).
- Contact us, if you are interested in using the Living Semantic Platform in your projects: living-semantic-platform@softplant.de
Softplant GmbH

Friedrichshofener Str. 12
85049 Ingolstadt, Deutschland
Phone +49 (0) 841. 370 81 - 0
Fax +49 (0) 841. 370 81 - 111

Agnes-Pockels-Bogen 1
80992 München, Deutschland
Phone +49 (0) 89. 550 603 - 46
Fax +49 (0) 841. 370 81 - 111

welcome@softplant.de | www.softplant.de
References