DIG 2.0 Reference Middleware

Timo Weithöner and Thorsten Liebig
and Olaf Noppens
University of Ulm, Dept. of AI
89069 Ulm, Germany
{timo.weithoener, thorsten.liebig, olaf.noppens}@uni-ulm.de

Marko Luther and Sebastian Böhm
DoCoMo Euro-Labs
Landsberger Strasse 312
80687 München, Germany
{luther,boehm}@docomolab-euro.com

Abstract

The DIG interface defines a protocol for the communication of Description Logic (DL) applications. The current DIG 1.1 specification has some shortcomings, which gave rise to the definition of its successor DIG 2.0. In this paper, we introduce the DIG 2.0 Reference Middleware. It allows for the integration of applications that comply with any of the two language versions. In addition to a language translation service, this middleware adds told and retraction functionalities to the DL reasoner when needed.

1 Introduction

The DIG interface [1], which is developed by the Description Logics Implementation Group (DIG), defines a protocol that enables a standardized access to Description Logics (DL) implementations. The DIG language consists of “tell” and “ask” messages, used to specify axioms and to access reasoning results. Messages are thereby serialized as XML and transmitted via HTTP.

The current release of the DIG interface (DIG 1.1) is already well established in the field of DL implementations. It is mainly used to connect ontology editors (e.g., OntoTrack [3], OilEd1, Protege2), middleware components (e.g., Instance Store3, Jena24), applications (e.g., McAnt [2]) and DL reasoners (e.g., CEL5, FaCT++6, KAON27, Pellet8, and Racer9).

DIG 2.010 introduces a number of improvements. Most importantly, it now includes full support for OWL 1.111. Furthermore, it has a well-defined mechanism for extending the core language. Predefined extensions allow for the retraction of axioms, the examination of previously told information as well as the integration of non-standard inferences. Clients and servers can use XML Schema definitions for the core language and its extensions to negotiate the common language subset by exchanging schema URIs.

2 Reference Implementation

System developers now face the challenge of adapting their implementations to the new DIG 2.0 standard interface. During this transition, there will be a heterogeneous set of systems, some already supporting DIG 2.0 and others still relying on DIG 1.1. Therefore, the DL Implementation Group decided to develop a Reference Middleware, rather than reference implementations of DIG 2.0 clients and servers. The purpose of the middleware is to lower the burden of adapting the new specification, to provide the developers a reliable tool for testing, and to validate the extended specification.

2.1 Architecture

The DIG 2.0 Reference Middleware mediates between a DIG client and a DIG server. It implements a DIG server that accepts DIG 1.1 and 2.0 messages and passes them on to the reasoner (either via DIG 1.1 or DIG 2.0, depending on the actual reasoner capabilities). As a result, the middleware is able to extend a DIG 1.1 reasoner or client with a DIG 2.0 interface. Figure 1 depicts the overall architecture and the interaction between the middleware and connected clients as well as servers. Internally, a Told Knowledge Base (ToldKB) keeps track of previously sent axioms, able to process told queries and retraction requests. This way, the middleware also provides an implementation of the DIG 2.0 told and retraction extensions.

To summarize, the DIG 2.0 Reference Middleware offers the following functionalities:
- Mediation between DIG 1.1 and DIG 2.0 components
- Query processing for told information
- Realization of a naive retraction mechanism

The processing steps of the middleware are as follows. At first, received DIG requests (1) are translated into the internal DIG 2.0 representation (if necessary). Axioms are copied into the ToldKB and told queries as well as retraction requests

1http://oiled.man.ac.uk
2http://protege.stanford.edu
3http://instancestore.man.ac.uk
4http://jena.sourceforge.net
5http://lat.inf.tu-dresden.de/systems/cel/
6http://owl.man.ac.uk/factplusplus
7http://kaon2.semanticweb.org
8http://www.mindswap.org/2003/pellet
9http://www.racer-systems.com
10http://www.cs.man.ac.uk/~seanb/dig/
11http://owl1_1.cs.manchester.ac.uk
are processed (2). The reminder of the request is sent to the reasoner (3). The answer received from the reasoner (4) is merged with the results of the internal processing (5), translated (if necessary) and sent back to the DIG client (6).

2.2 Implementation

The DIG 2.0 Reference Middleware is implemented in Java and can be seen as a specialized HTTP-Proxy. It accepts DIG messages that are processed and passed on to a DIG server. The response from the DIG server is combined with the results computed internally and eventually sent back to the original client.

Whenever a DIG message is received by the system, it is translated by an integrated XSLT engine into the DIG 2.0 language, which serves as the internal data format. Similarly, DIG servers are accessed either directly via DIG 2.0 or DIG 1.1, again translated by the XSLT engine. This actually depends on the type of language supported by the target system.

DIG 2.0 messages containing axioms (“tells”) are identified using XPath expressions and copied into the internal ToldKB. The resulting KB is kept in memory as a DOM object, periodically serialized as XML and stored in the file system for persistence. Whenever the DIG 2.0 Reference Middleware is restarted, the ToldKB is restored and all axioms are retransmitted to the reasoner. This allows for a complete recovery in case of a system crash, even if the client(s) do not hold a complete copy of all axioms sent to the server. The ToldKB is used to answer told requests, in case the reasoner is unable to do so. If a told query is identified within a DIG message, the ToldKB is scanned for matching axioms using a corresponding XPath expression.

The ToldKB also enables retraction, if the connected reasoner does not support this extension natively. Retraction requests, again identified with the help of XPath, are applied to the ToldKB. Once all corresponding axioms have been removed, the resulting ToldKB is sent to the reasoner, replacing the previously loaded knowledge base.

2.3 Summary

The general objective of the DIG 2.0 Reference Middleware is to support the transition towards DIG 2.0 by providing a testbed for the development and validation of DIG components. Thereby, the focus lies on functionality and correctness rather than performance. The implementation makes use of XPath expressions taken directly from the DIG 2.0 specification. It is able to extend existing DIG reasoners by told and retraction facilities, which become increasingly important for realizing feasible Semantic Web applications. The implemented middleware component will be freely available to the community.

Acknowledgements

The work described in this paper was inspired and encouraged by the members of the DL Implementation Group. We would like to thank the DIG members for their support and comments.

References


---

12 http://www.w3.org/TR/xslt
13 http://www.w3.org/TR/xpath