A Reasoning Broker Framework for Protégé

Jürgen Bock, Tuvshintur Tserendorj,
Yongchun Xu, Jens Wissmann and Stephan Grimm

FZI Research Center for Information Technology
at the University of Karlsruhe
{bock,tserendo,xu,wissmann,grimm}@fzi.de

Abstract

We develop a reasoning broker framework for the combined use of existing OWL reasoners assisted by means for caching of results, scheduling of reasoning tasks and online-selection of appropriate reasoners. Here we demonstrate an integration of the first version of this reasoning broker framework into the Protégé ontology engineering environment. The demo allows for the configurable combined use of various OWL reasoners with features of parallel execution and runtime-selection of appropriate reasoners.

1 Reasoning Brokerage within Protégé

For the Web Ontology Language OWL, various systems for reasoning are available and being currently developed. Different systems focus on different aspects of OWL reasoning and are optimised for different types of reasoning, such as ABox or TBox reasoning. We present a reasoning broker framework called HERAKLES[1], which connects to different reasoners while it is itself acting as a single OWL reasoner. The centralised control of various external remote reasoners enables the implementation and combination of different features that provide added value compared to traditional reasoner systems. Such features are

- Reasoner Selection
- Parallel Execution of Reasoners
- Partitioning of Ontologies
- Load Balancing
- Real-time Benchmarking

So far, we implemented the features *Reasoner Selection* and *Parallel Execution of Reasoning Tasks* by providing different broker strategies.

HERAKLES is integrated with Protégé by using the *ReasonerFactory* OSGi extension point to make it available as a reasoner. We also provide user interface components to configure the broker in terms of selection of remote reasoners, as well as selection and configuration of broker strategies.

There are two Protégé plug-ins (HERAKLES reasoning broker, and anytime query tab) available at `http://www.fzi.de/downloads/ipe/herakles/herakles.zip`.

2 Demo Scenario

We demonstrate four facets of our contribution to Protégé:

1. Use of HERAKLES as a reasoner from within Protégé. HERAKLES can be selected via the *Reasoner* drop-down menu.

[1]The presented research was funded by the German Federal Ministry of Economics (BMWi) under the project THESEUS (number 01MQ07019).

2. Configuration of HERAKLES from within Protégé. A novel *Configuration Tab* firstly allows to select external remote reasoners to be used by HERAKLES. Secondly it provides a view to select strategies to be used by HERAKLES, which control the behaviour of the broker system. Thirdly, selected strategies can be configured in another view, which is currently realised for the *TaskSelection* strategy. Here, the user can define conditions, which determine the selection of reasoners for each reasoning method provided by the OWL API reasoner interface. In *basic mode* a user just selects the reasoner type, which should be used for a specific API method. In *expert mode* selection conditions can be defined manually using any properties a reasoner needs to match.

3. Monitoring of external remote reasoners connected to HERAKLES. As the reasoners are run competitively, the number of times a reasoner responded first to a given task is displayed along with the percentage relative to the other reasoners.

4. Anytime reasoning using HERAKLES via a novel *Anytime Reasoning Tab*. In contrast to the Protégé 4 DL Query tab, the *Anytime Reasoning Tab* delivers query results timely decoupled from the reasoning request. By combination of approximate reasoning systems, we further exploit reasoner properties of soundness and completeness to qualify query results by means of different colours and strike-through rendering.