



## DIP D1.10: WSML-DL Reasoner

### Prototype Fact Sheet, 10 Jan 2007

**Latest version:**

<http://tools.deri.org/wsml2reasoner/DIPFactSheet.html>

**Authors:**

Holger Lausen, DERI Innsbruck  
Nathalie Steinmetz, DERI Innsbruck

**Reviewers:**

Stephan Grimm, FZI  
Damyán Ognyanoff

Copyright 2006 by [DIP](#). All Rights Reserved. [DIP](#) liability, trademark, document use, and software licensing rules apply.

## Document Information

<b>IST Project Number</b>	FP6 - 507483		<b>Acronym</b>	DIP
<b>Full Title</b>	WSML-DL Reasoner			
<b>Project URL</b>	<a href="http://dip.semanticweb.org">http://dip.semanticweb.org</a>			
<b>Document URL</b>	<a href="http://tools.deri.org/wsml2reasoner/DIPFactSheet.html">http://tools.deri.org/wsml2reasoner/DIPFactSheet.html</a>			
<b>EU Project Officer</b>	Werner Janusch			
<b>Deliverable Number</b>	1.10	<b>Title</b>		
<b>Work package Number</b>	1	<b>Title</b>	Ontology reasoning and querying	

<b>Date of Delivery</b>	<b>contractual</b>	M36	<b>actual</b>	20-December-2006
<b>Status</b>	<b>version</b>	0.2	draft	

<b>Nature</b>	Prototype Report Dissemination Ontology			
<b>Dissemination Level</b>	Public Consortium			
<b>Authors</b>	Nathalie Steinmetz, Holger Lausen			
<b>Responsible Author</b>	Nathalie Steinmetz		<b>Email</b>	Nathalie.Steinmetz@deri.org
	<b>Partner</b>	DERI Innsbruck	<b>Phone</b>	
<b>Version Log</b>				
<b>issue date (dd-mm-yy)</b>	<b>revision no.</b>	<b>author</b>	<b>change</b>	
21-11-06	001	Holger Lausen	first internal version (version 0.1)	
16-12-06	002	Nathalie Steinmetz	second internal version (version 0.2)	
10-01-07	003	Nathalie Steinmetz	final version after 1st review (v 0.3)	
<b>Reviewer Information</b>				
1	Stephan Grimm		<b>Email</b>	Grimm@fzi.de
	<b>Partner</b>	FZI	<b>Phone</b>	+49 (0)721 9654 816
2	Damyan Ognyanoff		<b>Email</b>	damyan@sirma.bg
	<b>Partner</b>	Ontotext Lab.	<b>Phone</b>	

---

## Table of contents

### 1. Availability and Contacts

### 2. Purpose and Functionality

[2.1. Architecture Overview](#)

[2.2. Ontology Transformation](#)

[2.3. Reasoning with WSML-DL Ontologies](#)

### 3. Requirements

### 4. Licensing

[4.1. Licensing of WSML2Reasoner](#)

[4.2. Licensing of Third Party Libraries](#)

### 5. Installation and Usage

[5.1. Installation of WSML2Reasoner](#)

[5.2. Usage Examples](#)

### 6. Future Plans

### References

---

# 1. Availability and Contacts

**Version:** 0.3, 10 January 2007.

**Download:** [http://tools.deri.org/wsml2reasoner/releases/v0\\_2/wsml2reasoner-dl-v0\\_2/](http://tools.deri.org/wsml2reasoner/releases/v0_2/wsml2reasoner-dl-v0_2/)

**Source control:** Available from CVS of the [DERI CVS](#).

## 2. Purpose and Functionality

The Web Service Modeling Language (WSML) is a family of formal Web languages. Its semantics is based on Description Logics, Logic Programming and First-order Logic, with influences from F-Logic and frame-based representation systems. Conforming to the different influences, there exist five variants of WSML: WSML-Core, WSML-DL, WSML-Flight, WSML-Rule and WSML-Full.

Deliverable D1.10 provides a reasoner prototype for WSML-DL, the variant that captures the expressive Description Logic SHIQ(D). Using this prototype we can, among others, perform the reasoning tasks of checking ontology consistency, entailment and instance retrieval. It also validates WSML-DL ontologies and allows to serialize the latter to OWL DL ontologies. However the serialization to OWL DL is not complete; for details about the restrictions see [Restrictions](#)).

### 2.1. Architecture overview

The WSML2Reasoner framework [Figure 1](#) is a flexible and highly modular architecture for easy integration of external reasoning components.

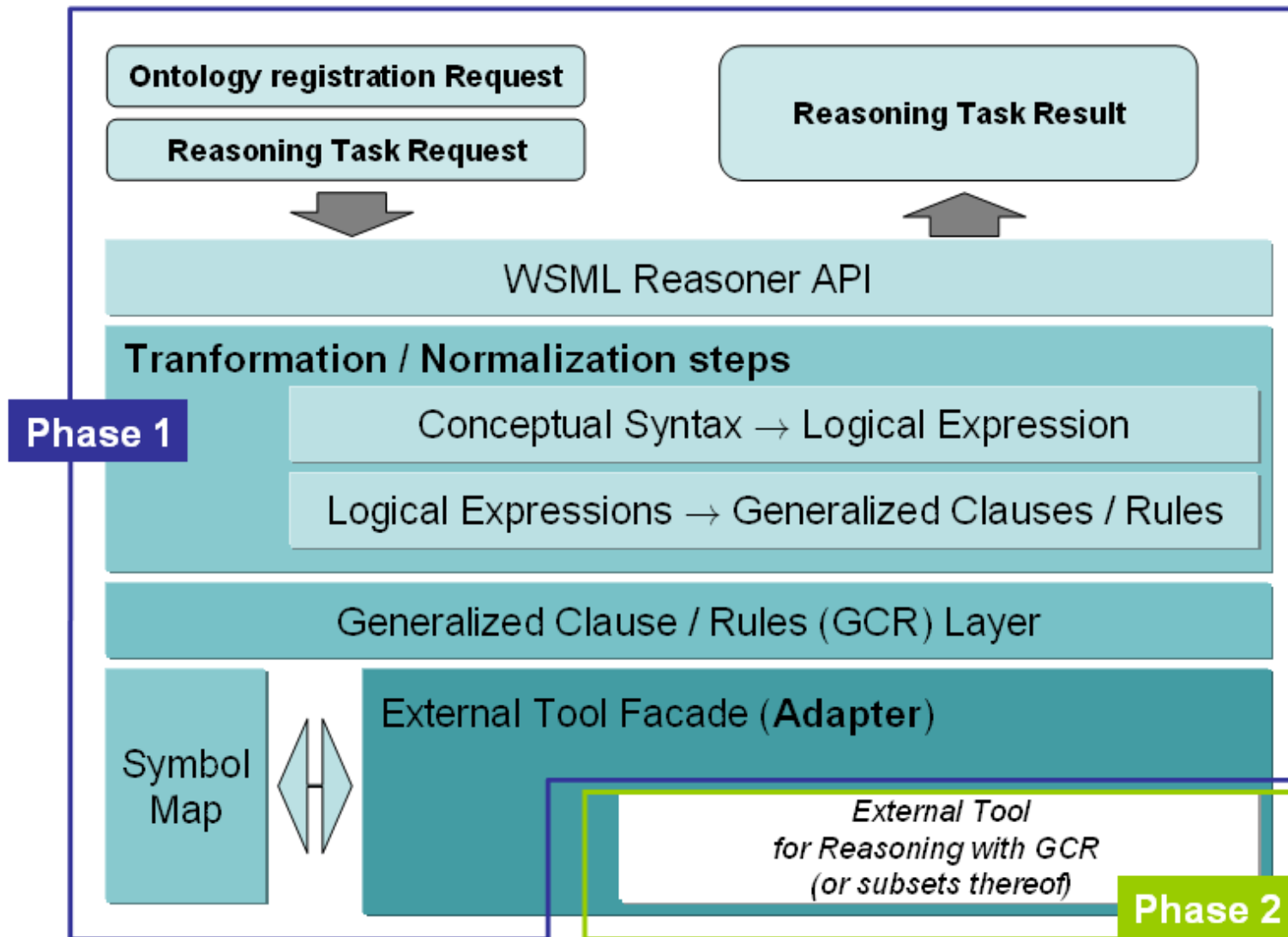


Figure 1. WSML2Reasoner framework

Instead of implementing new reasoners, existing reasoner implementations can be used for WSML through a wrapper that translates WSML expressions into the appropriate syntax for the reasoner. This wrapper contains various validation, normalization and transformation functionalities that are reusable across different WSML variants. For more details look at [Ontology Transformation](#). We use the [Wonderweb OWL API](#) to translate WSML-DL to OWL DL.

So far, we have embedded the OWL DL reasoners [Pellet](#) and [KAON2](#) as first reasoning engines in the WSML2Reasoner framework infrastructure:

- **Pellet** - Pellet is wrapped by a Pellet reasoner façade, which mediates between the OWL DL ontology produced by the transformation and the Pellet-specific internal representation.

- **KAON2** - KAON2 is wrapped by a KAON2 reasoner façade, which mediates between the OWL DL ontology produced by the transformation and the KAON2 API used for interacting with the KAON2 reasoner.

For each new DL reasoner that is integrated into the framework, such an adapter façade has to be written.

## 2.2. Ontology Transformation

The transformation of a WSML-DL ontology to an OWL DL ontology is done in a line of single transformation steps that are applied subsequently.

### **Relations to Attributes.**

Replace relations, subrelations and relation instances by attributes and axioms, according to the preprocessing steps described in [\[ReasWSMLDL\]](#).

### **Axiomatization.**

All conceptual elements are converted into appropriate axioms specified by logical expressions. The resulting set of logical expressions is semantically equivalent to the original WSML ontology.

### **Implication Reduction Rules.**

Replace equivalences and right-implications in logical expressions by left-implications.

### **InverseImplication Reduction Rules.**

Replace conjunctions on the left side and disjunctions on the right side of an inverse implication by left implications.

### **Molecule Decomposition Rules.**

Replace complex molecules inside a logical expression by conjunctions of simple ones.

### **OWL API Transformation.**

All logical expressions that are resulting from the transformation and normalization steps described above, are processed one by one. Each logical expression is translated into the corresponding OWL Description, using the [Wonderweb OWL API](#).

[Figure 2](#) shows a detailed perspective on the transformation steps for the WSML-DL variant. It also emphasizes the highly modular internal architecture of the WSML-DL component of the WSML2Reasoner framework.

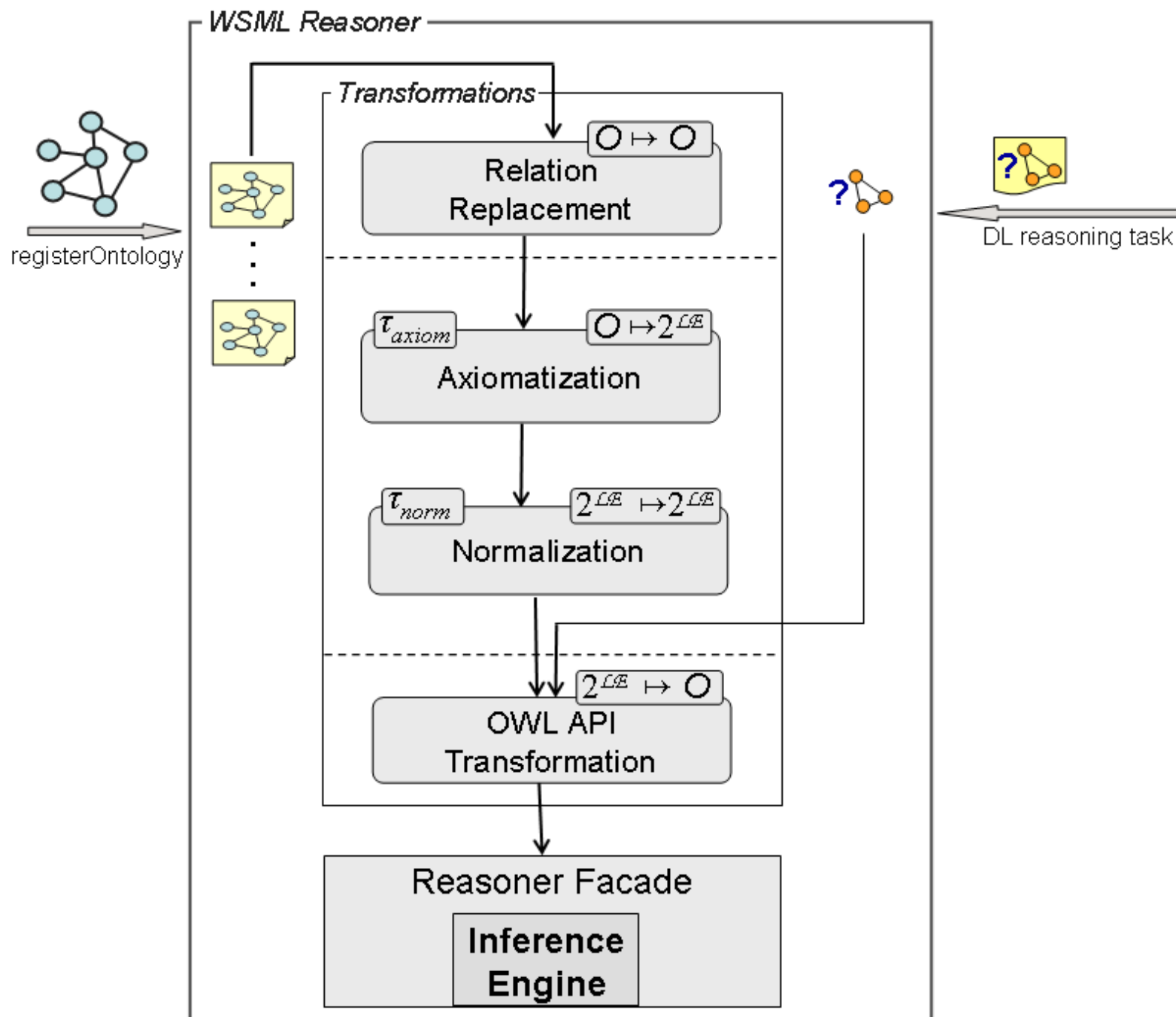


Figure 2. Internal framework architecture

## Restrictions to the transformation

The transformation from WSML-DL to OWL DL is not complete. The restrictions are mainly due to the differences of the Description Logics underlying WSML-DL (SHIQ(D)) and OWL DL (SHOIN(D)). OWL DL does not support the following features:

- **Datatype predicates** - Datatype predicates are lost during the transformation.
- **Qualified Number Restrictions** - Qualified Number Restrictions (QNRs) are lost during the transformation. In [\[ReasWSMLDL\]](#) we mention a possible workaround and a non-endorsed OWL extension, that both would allow to translate QNRs into OWL DL (only approximated with the workaround). We have planned to implement the workaround as a next step and, in the long run, intend to extend the OWL API for support of the non-endorsed OWL extension.

## 2.3. Reasoning with WSML-DL Ontologies

In Description Logics, there are different basic reasoning tasks for reasoning with TBoxes or ABoxes, the two parts of DL knowledge bases. TBoxes contain the terminological knowledge of a knowledge base, e.g. concept definitions. ABoxes contain assertional knowledge, which is knowledge about the individuals of a domain. As described in [\[DLHandBook\]](#), the main inference procedures with TBoxes are Concept subsumption and Concept satisfiability. With ABoxes, the main reasoning tasks are ABox consistency and Instance checking.

The OWL community focuses on entailment and query answering as the key inference services. Entailment can be reduced to satisfiability, while query answering amounts to compute the result of a query over a database, or an ABox respectively.

In [\[ReasWSMLDL\]](#) and [\[DLHandBook\]](#) you will find descriptions of the main standard reasoning tasks as described above, as well as of some main non-standard inference tasks.

The following is the functionality that is supported by the WSML-DL Reasoner prototype:

- Check Ontology, Concept or Logical Expression consistency.
- Get all concepts, instances or attributes from the ontology.
- Get all constraint or inferring attributes from the ontology.
- Get all subconcepts or superconcepts of a specified concept.
- Get all direct subconcepts or superconcepts of a specified concept.
- Get all concepts or attributes equivalent to a specified concept or attribute.
- Check if two concepts are equivalent.
- Check if a specified concept is a subconcept of another specified concept.
- Check if a specified instance is a member of a specified concept.
- Get all instances of a specified concept.
- Get all direct concepts a specified instance is member of.
- Get all (direct and indirect) concepts a specified instance is member of.
- Get all sub- or superrelations (-attributes) of a specified relation(attribute).
- Get all direct sub- or superrelations (-attributes) of a specified relation (attribute).
- Get all relations(attributes) inverse to a specified relation(attribute).
- Get all concepts from a specified attribute.
- Get all instance or datavalue ranges from a specified inferring or constraint attribute.
- Get all inferring or constraint attributes with the corresponding values or datavalues from a specified instance.

- Get all instances that have values for a specified constraint or inferring attribute, with the corresponding values.
- Check if a specified instance has a specified constraint or inferring attribute value.
- Get values or datavalues for a specified instance and a specified constraint or inferring attribute.

## 3. Requirements

**Nature:** A Java library without user interface.

**Interfaces (API, Web Services):** a Java API.

**Platform:** JDK 1.5.

**Supported standards:**

- WSMML-DL [[wsml0.2](#)] as knowledge representation language.
- OWL DL [[OWL](#)] as knowledge representation language.

**Required Libraries:**

- **WSMML2Reasoner DL Core Libraries:**

- [wsml0.2](#) is an API and a reference implementation for building Semantic Web Services applications compliant with the Web Service Modeling Ontology ([WSMO](#)). The version used in the current version of the WSMML2Reasoner framework is 0.5.2 from 28-Mar-2006, or newer.
- [OWL API](#) - Mandatory. The OWL API provides a programmatic interface for accessing, manipulating and reasoning over OWL ontologies. It comes along with a simple reference implementation. The version used in the current version of the WSMML2Reasoner framework is 1.4.3.
- [econn OWL API](#) (Mindswap)
- [log4j](#) (Apache)
- [rdfapi](#) (KAON)

- **Libraries needed for reasoning with Pellet:**

- [Pellet](#) - Sample Reasoner. The version used in the current version of the WSMML2Reasoner framework is 1.3.

The Pellet reasoner furthermore requires the following libraries:

- [xsdlib](#) (Sun Microsystems)
- [relaxngDatatype](#) (Sun Microsystems)
- commons-logging 1.1 (Apache)
- [aterm](#) (Centrum voor Wiskunde en Informatica (CWI))

- **Libraries needed for reasoning with Kaon2:**

- [KAON2](#) - Sample Reasoner. The KAON2 reasoning engine is not shipped in the WSMML2Reasoner release, but wrappers for it are implemented and tested.

## 4. Licensing

### 4.1. Licensing of WSML2Reasoner

This library is free software; you can redistribute it and/or modify it under the terms of the [GNU Lesser General Public License](#) as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

### 4.2 Licensing of Third Party Libraries

Licensing of third party libraries and components required for the WSML-DL Reasoner component of WSML2Reasoner:

- [wsmo4j](#) - (c) Copyright Ontotext Lab, Sirma and University of Innsbruck. It is an open-source library, available under the same [LGPL](#) conditions.
- [OWL API](#) - (c) Copyright University of Manchester and University of Karlsruhe. It is an open-source library, available under the same [LGPL](#) conditions.
- [Pellet](#) - (c) Copyright 2003 Ron Alford, Mike Grove, Bijan Parsia, Evren Sirin. It is an open-source library, available under the [MIT License](#).
- [econn OWL API](#) - (c) Copyright Mindswap. It is an open-source library, available under the [LGPL](#) conditions.
- [xsdlib](#) - (c) Copyright Sun Microsystems. This library is available under the [Sun Microsystems License](#).
- [relaxngDatatype](#) - (c) Copyright Sun Microsystems. This library is available under the Thai Open Source Software / Sun Microsystems license.
- [commons-logging 1.1](#) - (c) Copyright Apache. This library is available under the [Apache License](#).
- [log4j](#) - (c) Copyright Apache. This library is available under the [Apache License](#).
- [rdfapi](#) - (c) Copyright KAON. This library is available under the [LGPL](#) conditions.
- [aterm](#) (c) Copyright Centrum voor Wiskunde en Informatica (CWI). This library is available under the [LGPL](#) conditions.
- [KAON2](#) - (c) Copyright Ontoprise GmbH. KAON2 is available as a precompiled binary distribution and is free of charge for research and academic purposes. To use KAON2 in a commercial setting, please contact [Ontoprise GmbH](#).

## 5. Installation and Usage

### 5.1. Installation of WSML2Reasoner

WSML2Reasoner is distributed as a ZIP archive, which should be extracted in a separate folder.

#### Generic WSML2Reasoner DL Release

The archive file is originally named [wsm2reasoner-dl.zip](#) and has the following contents:

- `wsm2reasoner.jar` - the WSML2Reasoner framework provided as a Java library.
- folders containing the required libraries (jar files and licensing information);

- `license.txt` file - contains the LGPL license;

In the `lib` folder are all the required libraries. Please note that in this release contains no reasoning engines. For reasoning with KAON2 please download the libraries at <http://kaon2.semanticweb.org/#download>.

## Pellet WSML2Reasoner DL Release

The archive file is originally named [wsml2reasoner-pellet.zip](#) and has the following contents:

- `wsml2reasoner.jar` - the WSML2Reasoner framework provided as a Java library.
- folders containing the required libraries (jar files and licensing information);
- `license.txt` file - contains the LGPL license;

In the `lib` folder are all the required libraries.

## 5.2. Usage Examples

A simple usage scenario is provided as an illustration of the functionality of the WSML-DL Reasoner prototype. It is available as Java source in the `src/example` folder.

- [PelletDLReasonerExample](#) - parsing an WSML-DL ontology, validating it and transforming it into an OWL DL ontology. This OWL DL ontology is registered at the Pellet reasoner façade and is used for executing different reasoning tasks.
- [Kaon2DLReasonerExample](#) - parsing an WSML-DL ontology, validating it and transforming it into an OWL DL ontology. This OWL DL ontology is registered at the KAON2 reasoner façade and is used for executing different reasoning tasks.

## Future Plans

In the future more reasoners will be added to the framework, e.g. [FaCT++](#) and [RACER](#). To be able to add reasoners that do not support the OWL API interface, we will add an export from OWL DL to the Description Logic Interface [DIG](#).

## References

**[DLHandBook]** Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, Peter F. Patel-Schneider: *The Description Logic Handbook*. 2003. Cambridge University Press. 0-521-78176-0.

**[OWL]** McGuinness, D. et al. (2004). *OWL Web Ontology Language*. W3C Recommendation, <http://www.w3.org/TR/owl-features/>

**[ReasWSMLDL]** Uwe Keller, Cristina Feier, Nathalie Steinmetz, Holger Lausen: *Report on reasoning techniques and prototype implementation for the WSML-Core and WSML-DL languages*. Deliverable d1.2 v1.0, RW2, 2006. <http://rw2.deri.at/pub/d1.2.pdf>

**[WSML0.2]** J. de Bruijn, H. Lausen, R. Krummenacher, A. Polleres, L. Predoiu, M. Kifer, D Fensel: *The Web Service Modeling Language WSML*.

Deliverable d16.1v0.2, WSML, 2005. <http://www.wsmo.org/TR/d16/d16.1/v0.2/>



\$Date: 2007/01/16 16:36:37 \$