The SDL Library: Querying a Relational Database with an Ontology, Rules and the Jess Engine

Jaroslaw Bak, Maciej Falkowski and Czeslaw Jedrzejek
Institute of Control and Information Engineering,
Poznan University of Technology,
M. Sklodowskiej-Curie Sqr. 5, 60-965 Poznan, Poland
{firstname.lastname}@put.poznan.pl

Abstract. In this demo we present the Semantic Data Library (SDL) which is used to query a relational database at a concept (ontological) level. The SDL integrates a rule engine, a relational database and a set of rules obtained from the transformation of an OWL ontology. This combination allows querying and inferring with data stored in a relational database using concepts, roles and rules. We propose an implementation of the method of querying relational database with extended rules and the transformation of OWL ontologies into sets of rules. Our demonstration is based on the previously presented financial crime 'minimal model' ontology and artificially generated data sets. Prospects of the future development of the SDL tool are presented.

Keywords. SDL library, Jess engine, rules, OWL ontology, query answering

1 Introduction

The most of data processed in modern applications come from relational databases. Such data is described only by their schema (a structure of data). Without strictly defined semantics there is often a mismatching problem with table and column names in databases. Moreover, it is rather difficult to query data at a more abstract level than only in a language of database relations and attributes. A lack of conceptual knowledge can be overcome by introducing ontologies. For the evaluation purposes, an ontology (and other knowledge) can be transformed into a set of rules (however, several of the OWL axioms cannot be transformed [1]). The additional rule-based knowledge allows reasoning and query answering at an appropriate abstract layer. Moreover, it simplifies posing a question than using structural constructions from SQL. This kind of query evaluation is called the rule-based query answering method.

As a result a user gets an easy way to query a relational database and both a query and an answer are based on the semantics defined in an OWL [2] ontology. The ontology describes data at the concept (ontological) level and introduces a formal definition of concepts and roles which do not exist directly in the database. For example, let us assume that we have a table *persons(id, fatherID, motherID, gender)*. In the corresponding OWL ontology we can define the following concepts:

Grandfather, *Grandmother*, *Cousin* etc. and roles: *hasBrother*, *hasSister*, *hasCousin* etc. These concepts/roles are not defined directly in the database. But with the use of the OWL ontology and SWRL [3] rules we can obtain instances of the above-mentioned terms. Moreover, we can use these terms in queries which are in the form of directed graphs.

In this paper we present a prototypical implementation of the Semantic Data Library (SDL) tool which integrates an OWL ontology, SWRL rules, the Jess [4] reasoning engine and a relational database. Our integration allows to pose a query to a relational database at concept (ontological) level. We assume that OWL ontology which is handled by the SDL can contain both OWL axioms and SWRL rules.

During the development and research process, we have proposed and implemented two methods of querying relational database: hybrid reasoning [5] and forward reasoning with extended rules [6]. In this work, we are focused on the implementation and the evaluation of the latter method. The paper makes the following contributions:

- We present the SDL library in details: characterizing the functionalities and the OWL to Jess transformation methods,
- We evaluate our 'minimal model' ontology with all our approaches achieved so far,
- We show that our approaches increase the scalability of the Jess engine and outperforms its rule-based query answering method.

The paper is organized as follows. Section 2 presents the SDL architecture and functionalities. Section 3 describes an example evaluation and application of the SDL tool to the previous constructed 'minimal model' ontology. Section 4 contains concluding remarks and future work plans.

2 SDL Architecture and Features

2.1 SDL Overview and Architecture

SDL integrates ontologies, relational data and rules which represent domain knowledge. We need such tool when we have to pose complicated queries to the standard relational database. Due to the formally defined semantics (OWL) we can pose a semantic query and get a corresponding semantic answer. The SDL generates rules automatically which is very important for knowledge bases that often change.

The architecture of this system is presented in Figure 1. The central part, which gathers input from other system elements and processes rules, are one [6] or two [5] Jess engines used for forward and backward chaining. The hybrid approach [5] exploits both forward and backward reasoning. The backward method is responsible for gathering data from a relational database and the forward chaining is used to answer a given query. One instance of the Jess engine is created for each reasoning method. It means that we use two instances of the Jess engine in the hybrid approach.

In the extended rules [6] approach we use one instance of the Jess engine, because only the forward reasoning method is used. *Extended* means that these rules are generated automatically from the basic ones for the evaluation purposes, and the modification is strongly connected with the magic transformation [7] method. The set of basic rules consists of rules which constitute the knowledge base. The rule-based knowledge base comes from an OWL to Jess transformation. The set of extended rules is semantically equivalent to the set of basic rules. The extended rules are generated in the goal- and dependency-directed transformation. In this method we are interested in dependencies between variables appearing in predicates inside each rule. Together with the mapping rules, the extended ones are used in the rule-based query answering algorithm.

The rule-based query answering method in this approach needs the different assumptions from the hybrid one because we use only one Jess engine to obtain relational data and answer a query. Obviously, we modified our query answering algorithm prepared for the hybrid system. More theoretical information can be found in [6].

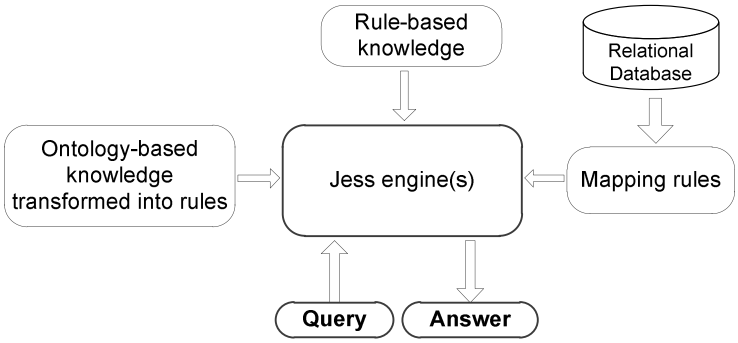

Figure 1. The architecture of the rule-based query answering system

Figure 2 presents the integration scheme of an OWL ontology with SWRL rules, the Jess engine and a relational database. We assume that the ontology is in the Horn-SHIQ language and contains SWRL rules (Horn-like clauses). Such OWL+SWRL ontology is transformed into a set of rules in the Jess language. The set of rules is stored as a Jess script file (*.clp). The script is then transformed into a set of extended rules (*ExRScript.clp*). A user can load: *ExRScript.clp* and a mapping rules Jess script; then establish a database connection and pose queries with SDL and Jess. It is worth noting that such a transformation needs to be done only once (besides changes of the OWL ontology, SWRL rules or the database schema).

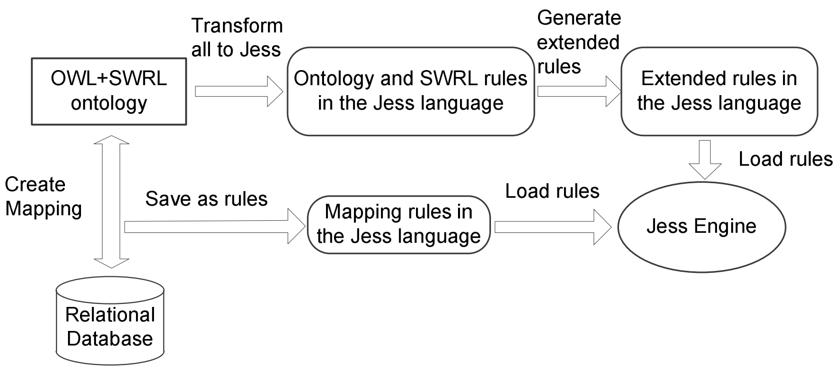

Figure 2. The integration scheme executed in the SDL library

2.2 SDL Features

The SDL tool is implemented in Java language. It is split into two modules:

- SDL-API (Application Programming Interface), which provides all functions,
- SDL-GUI (Graphical User Interface), which exploits SDL-API functions for defining the mapping between ontology terms and relational data; and provides automatic transformation of ontology into rules and the generation of Jess scripts.

The SDL-API module provides the following functionalities:

- reading a relational database schema,
- executing SQL query or procedure (results are added into Jess engine as facts),
- reading OWL ontology and Jess scripts,
- Jess scripts generation (forward and backward chaining, extended rules, Horn-SHIQ transformation) from OWL ontology,
- mapping between ontology concepts/roles and relational data,
- executing a Jess query which consists of the concepts and roles from OWL ontology or templates defined in Jess language,
- rule-based query answering methods: hybrid and extended rules,
- Jess engine reasoning management (in forward and backward chaining).

Due to SDL-GUI module the library enables executing the following functions:

- reading ontology and viewing of concepts/roles hierarchies; the view contains classes hierarchy, object properties hierarchy and datatype properties hierarchy. These hierarchies are calculated by the Pellet engine [8],
- viewing a relational database schema which contains tables, views, columns and data types,
- mapping between ontology concepts/roles and relational data,
- populating an ontology with data from a relational database according to the specified mapping,
- creating Jess facts from a relational database according to the specified mapping,
- transforming OWL ontologies to Jess scripts,
- transforming Jess scripts into Jess scripts with extended rules (only *triple* template of facts is currently supported).

SDL supports interaction with the Pellet engine (for TBox reasoning with ontology and its classification), exploits OWL API [9] (for handling OWL files) and uses JDBC library for MS SQL 2008 Server access. The taxonomies of ontology classes and properties are classified by SDL-GUI with Pellet 2.3.0 and prepared for a user, who can define SQL mapping queries on these calculated taxonomies.

Figure 3 presents our minimal model ontology loaded into SDL-GUI and established connection to the corresponding relational database. A user gets a presentation of tables and views which exist in a database.

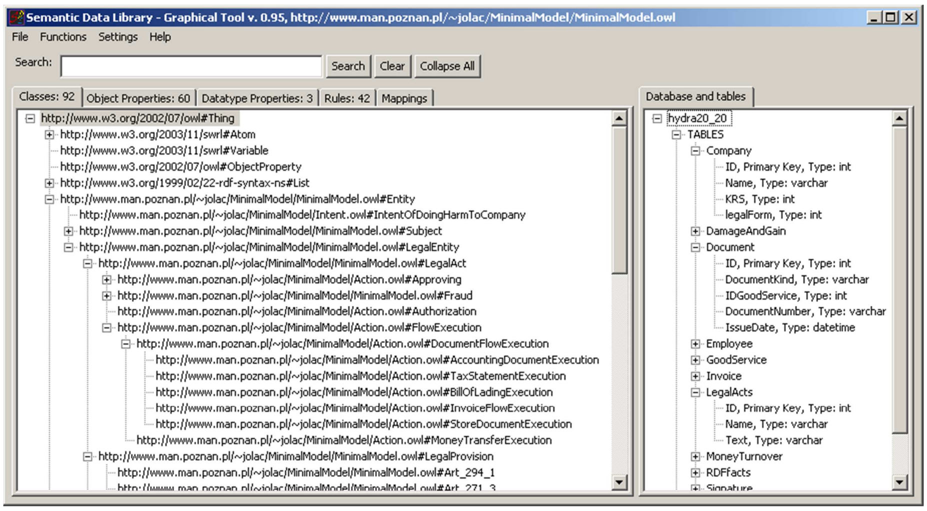

Figure 3. The SDL tool with minimal model ontology and database connection

SDL is available as a binary distribution and is free of charge for non-commercial academic usage (for universities only) and can be downloaded from the Web site [10].

2.3 OWL to Jess Transformation Methods

The SDL library supports two main methods of transforming OWL ontologies into rules expressed in Jess language: *simple* and *Horn-SHIQ*. The simple method transforms taxonomies of concepts and roles into Jess rules. These taxonomies are calculated by the Pellet engine first. SWRL rules and SWRLB [11] predicates are also transformed into rules and Jess expressions. The simple transformation can be done in the following modes:

1. Jess script assigned to forward chaining.
2. Jess script assigned to backward chaining.
3. Jess script assigned to forward chaining with extended rules.

The Horn-SHIQ transformation is an extension of the simple one. In this case, additional rules are generated according to (not all) OWL axioms. Rather than transforming the semantics of the OWL language into rules we create rules according to this semantics and a given ontology (in contrast to work presented in [12] and [13]). For example, when we have an ObjectProperty which is a SymmetricObjectProperty we create a rule which reflects that when an instance of this property occurs, a symmetric instance will also occur:

```
(defrule MAIN::HST-SymmetricProperty-inComplicityWith
  (triple (predicate "inComplicityWith") (subject ?x) (object ?y))
  =>
  (assert
     (triple (predicate "inComplicityWith") (subject ?y) (object ?x))))
```

Currently, the implementation is prototypical and does not support all Horn-SHIQ axioms from the W3C specification [14]. The SDL allows for use of simple atomic concepts (A, C), and roles (R). We assume that a concept C is simple if it is of the form: $A, \exists R.A, \forall R.A,$ or $\leq 1R.A$. Complex constructions are not supported. The universal and the existential quantifiers are used only as restrictions in the same way as presented in [15].

Currently supported OWL axioms are taken from the official Horn-SHIQ specification [14] and cover the following list:

a) class axioms:
   - equivalentClasses: URI | ObjectIntersectionOf | ObjectSomeValuesFrom
   - subClass: URI | ObjectUnionOf | ObjectIntersectionOf | ObjectSomeValuesFrom
   - superClass: URI | ObjectIntersectionOf

b) property axioms: URI | equivalentObjectProperties | subObjectPropertyOf | objectPropertyDomain | objectPropertyRange | functionalObjectProperty | inverseFunctionalObjectProperty | symmetricObjectProperty

The Horn-SHIQ transformation can be executed only in two modes: 1 and 3. The SDL also provides the Horn-SHIQ transformation without hierarchy rules. This feature can be helpful to use scripts in different Jess engines.

3 Example Evaluation

For a practical demonstration of the SDL library we used the 'minimal model' ontology (the one that fully models a fraudulent disbursement economic crime, but not other economic crimes) with artificially generated data sets. These data sets contain information about: companies, employees, documents, invoices, money turnovers, legal sanctions for this class of crimes, etc. We prepared three databases which differ in the size of the generated documents, values of money, turnovers, etc. The number of companies and employees are the same in every database (20 companies and 240 people). Generated databases contain the following numbers of documents (and money turnovers): 20, 100, 200. An example crime scheme and more information about 'minimal model' ontology are presented on the demo description site [10].

We executed five test queries for which description and graphical representation are presented on the demo site. Queries where executed on a computer with the following parameters: Intel Core2Duo 2GHz, 2GB Ram; Java Heap Space was set at 1024MB.

We compared the extended rules approach (marked 2011) with the results presented on the last RuleML Challenge [16] (marked 2010). The comparison is made using the same 5 queries as in 2010. Our current approach outperforms the hybrid one. Since we did not apply all possible optimizations, we are convinced that the efficiency of our method can be improved.

We compared our results with pure forward and backward reasoning in Jess system. Results of this comparison can be found in [10] in the Section called 'Evaluation'. In these tests while loading data from the third database, the size of the Java heap space was reached (in both engines), so the queries could not be executed.

It seems that for small databases, it is better to store data (facts) in the engines' working memory. But for the larger databases, the problem with scalability occurs. In such cases our extended rules approach seems promising.

Table 1. Results of queries execution in comparison to the RuleML Challenge 2010 results [6]

Query and info		Database 20		Database 100		Database 200	
		2010	2011	2010	2011	2010	2011
Query 1	[ms]	781	219	1 328	891	1 922	969
Results	[number]	54	54	474	474	1 036	1 036
Rules Fired	[number]	74	251	441	1 630	796	3 001
Query 2	[ms]	2 734	437	37 141	4 125	163 968	19 391
Results	[number]	1	1	1	1	1	1
Rules Fired	[number]	1076	1 506	36 260	13 179	225 381	29 593
Query 3	[ms]	2 875	359	36 344	14 938	183 047	116 593
Results	[number]	18	18	322	322	1004	1 004
Rules Fired	[number]	1 367	2 005	38 457	41 755	232 583	359 681
Query 4	[ms]	5 437	1 859	128 719	35 656	Time	347 110
Results	[number]	1	1	1	1	exceeded	1
Rules Fired	[number]	2 040	5 467	57 091	58 520	10 minutes	597 711
Query 5	[ms]	9 312	1 234	Time	34 500	Time	343 469
Results	[number]	1	1	exceeded	1	exceeded	1
Rules Fired	[number]	2 540	5 828	10 minutes	61 199	10 minutes	608 925

We also executed test queries with extended rules method and Horn-SHIQ transformation rules and compared them to the results achieved with the simple transformation rules. The results and the comparison are shown in [10], in the 'Evaluation' section. An addition of Horn-SHIQ rules makes query answering process more complicated and computationally demanding. It results from fact that Horn-SHIQ transformation contains more OWL axioms than the simple transformation.

Presented results confirm that our approach significantly improves a scalability of a rule-based system in the rule-based query answering. It is a very important, because in the forward chaining rule-based systems, facts have to be stored in the working memory which is, in general, limited by the RAM memory (we call it the traditional approach). If we store facts outside of the memory and load them only when they are needed, we achieve better scalability.

The SDL Demo with above test queries and presented query answering method are available on the demo site [10]. The 'minimal model' ontology is added to the demo material. On the demo site a user has an option to pose her/his own query constructed from concepts and roles from the minimal model ontology. Two databases are available: Database 20 and 100.

4 Conclusions and future work

In this paper we described the SDL library and demonstrated its application to the previously developed the 'minimal model' ontology. We presented a generalization (that is containing more OWL axioms) of the previously introduced hybrid method [5] to the case of transformation of an OWL ontology into Horn-SHIQ rules in the Jess language. The implementation was executed in the dedicated SDL framework. We also confirmed that our approaches significantly improve a scalability of a rule-based system compared with the pure Jess approach.

The SDL library is useful for queries creation because a user of our system gets an easier way to pose queries (due to ontology origin of rules) than using structural constructions from SQL. The creation of queries, presented in the performance evaluation, is extremely difficult when we want to use pure SQL constructions. The strictly defined semantics (in the form of an ontology) is another advantage of our tool.

In future, we are going to use other ontologies to test our tool. We will also extend our approach to handle predicates with an arbitrary number of arguments. We will improve the rule-based query answering algorithm by using optimizations that concern extended rules and magic transformation.

Acknowledgement. This work was supported by PUT BW 45-087/11 BW 2010 and PUT DS 45-085/11 PB grants.

References

1. Hitzler P., Parsia B., Ontologies and Rules, in: Steffen Staab and Rudi Studer (eds.), Handbook on Ontologies. Springer, 2nd Edition, 2009, pp. 111-132
2. McGuinness D., van Harmelen, F.:. Owl web ontology language overview. W3C Recommendation, 10 February 2004, http://www.w3.org/TR/owl-features/
3. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: Swrl: A semantic web rule language combining owl and ruleml. W3C Member Submission (May 21 2004), http://www.w3.org/Submission/SWRL/
4. Jess (Java Expert System Shell), http://jessrules.com/
5. Bak, J., Jedrzejek, C., Falkowski, M.: Usage of the Jess engine, rules and ontology to query a relational database. In: Governatori, G., Hall, J., Paschke, A. (eds.) RuleML 2009. LNCS, vol. 5858, pp. 216–230. Springer, Heidelberg (2009)
6. Bak J., Brzykcy G., Jedrzejek C., Extended Rules in Knowledge-based Data Access, F. Olten, M. Palmirani, D. Sottara (Eds.): RuleML - America 2011, LNCS 7018, pp. 112--127. Springer, Heidelberg (2011)
7. Beeri, C., Ramakrishnan, R.: On the Power of Magic. J. Log. Program.(1991) 255--299, (1991)
8. Pellet Reasoner, http://clarkparsia.com/pellet/
9. Matthew Horridge, Sean Bechhofer. The OWL API: A Java API for Working with OWL 2 Ontologies. OWLED 2009, 6th OWL Experienced and Directions Workshop, Chantilly, Virginia, October 2009, http://owlapi.sourceforge.net/
10. SDL demo and download page, http://draco.kari.put.poznan.pl/
11. SWRL Built-ins, http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/
12. Mei, J., Paslaru Bontas, E., Reasoning Paradigms for OWL Ontologies, FU Berlin, Fachbereich Informatik, Technical Reports B 04-12), 2004.
13. G. Meditskos, N. Bassiliades, "A Rule-based Object-Oriented OWL Reasoner", IEEE Transactions on Knowledge and Data Engineering (TKDE), 08 Oct 2007. IEEE Computer Society Digital Library. IEEE Computer Society, 9 October 2007
14. http://www.w3.org/2007/OWL/wiki/Tractable_Fragments#Horn-SHIQ
15. Grosof B., Volz R., Horrocks I. and Decker S. Description Logic Programs: Combining Logic Programs with Description Logics. In Proc. of the 12th International World Wide Web Conference (WWW 2003), 2003.
16. Bak J., Jedrzejek C., Falkowski M., Application of the SDL Library to Reveal Legal Sanctions for Crime Perpetrators in Selected Economic Crimes: Fraudulent Disbursement and Money Laundering, In Proceedings of the 4th International RuleML-2010 Challenge, Washington, DC, USA, October, 21-23, 2010. Edited by: Palmirani M., Omair Shafiq M., Francesconi E., Vitali F., Volume 649, Washington (2010)