Reasoning in the FIBO ontology – A challenge

Pawel Garbacz[1], Elisa Kendall[2]

[1]EDM Council, 10101 East Bexhill Drive Kensington, MD 20895, USA
[2]Thematix, 954 Lexington Avenue, New York, USA

Abstract
The paper discusses some challenges one must face when using automatic reasoners for more complex Semantic Web ontologies. The case in question is FIBO - an enterprise-level ontology for the financial industry.

Keywords
ontology consistency, DL reasoner, performance, financial industry

Introduction

One of the founding principles of the Semantic Web ontologies is their decidability. In principle, this should allow one to perform more or less complex reasoning tasks. Unfortunately, it is well-known that more complex ontologies present a serious challenge to this goal because even a simple task of checking whether a DL ontology is consistent may take up more time than the ontology's developers are willing to spend. This paper presents a more troublesome example: an enterprise-level ontology for the financial industry.

In section 1 we describe this ontology and the methodology of its development. Section 2 summarises the issues with using the standard DL reasoners for the basic reasoning tasks.

1. FIBO ontology

FIBO ontology evolved out of concerns that arose during the 2008 financial crisis among individuals who worked together in data governance and management. The most pressing issue at the time was the need for a *shared*, common vocabulary, focused on financial contracts and related concepts, that could be used for analysis and regulatory reporting purposes. Since then, FIBO has been sponsored and hosted by the Enterprise Data Management Council (https://edmcouncil.org), a global association for data management professionals, initially focused on financial services that have since expanded to other domains. Some selected modules of FIBO have been standardized by the Object Management Group (https://www.omg.org), and a new baseline standard is in work.

FIBO exists, so to speak, at two levels: the production version (FIBO PROD), which is the ready-to-use release, where all known issues are resolved, and the development version (FIBO

SemREC'21: Semantic Reasoning Evaluation Challenge, ISWC'21, Oct 24 – 28, Albany, NY
pawel.garbacz@makolab.com (P. Garbacz); ekendall@thematix.com (E. Kendall)
0000-0003-4145-7528 (P. Garbacz)

DEV), which ranges in maturity from “almost releasable” to “really, really rough,” is more than 40 percent larger. Both versions are available from https://github.com/edmcouncil/fibo.

Currently, three FIBO primary content development teams are working in parallel on different but related topics. In order to coordinate continuous integration of new and revised material, facilitate collaboration between ontologists, and ensure continuous quality improvement, leadership and process teams were put in place several years ago. One of the products of their work is a development framework created to automate aspects of ontology “unit-level testing” to guarantee a minimum level of quality.

The original motivation for FIBO was the failure of financial institutions and regulatory agencies to clearly exchange and integrate data about financial contracts and their counterparties, as demonstrated by the industry’s failure to roll up the risk with respect to those contracts. The initial FIBO use case was to provide an industry glossary that financial institutions and other market participants can use to meet regulatory requirements such as Dodd-Frank[1] in the U.S. and the MiFID II[2] framework in the EU for regulating financial markets. That use case was extended to cover additional requirements for data governance, data management, and enterprise glossaries mandated in the EU by the Basel Committee on Banking Supervision (BCBS) for risk data aggregation and reporting (BCBS 239[3]). Over the last few years, we have refined our approach as recommended in [1] to create instrument- or topic-specific use cases that add incremental value, resulting in significant progress by each working group. The use cases include several usage scenarios and a number of competency questions per scenario, which are used to test the efficacy of the ontology as the work progresses.

The FIBO effort is organized into working groups, each consisting of at least one ontologist and some number of subject matter experts, which meet weekly to (1) review the use cases, (2) find areas in the ontologies where gaps remain, (3) refine and extend the ontologies to address those gaps and other issues raised by users, and (4) develop examples that answer the competency questions based on the revisions to the ontologies. Given an issue, use case, or partial use case, such as one scenario, the development process is roughly as follows:

1. In the context of a working group teleconference, review the existing ontology to determine what aspects of the ontology can be used to answer the question(s)
2. Identify the specific gap(s) and raise an issue to address the gap
3. Identify any missing concepts and work together to develop definitions and other annotations for those concepts and any important relationships based on a combination of appropriate resources (online financial dictionaries, offline financial dictionaries, ISO and other financial standards, etc.) and record our findings, discussion, and references in our minutes in the working group wiki
4. Create a branch in GitHub for the issue
5. Identify the ontology(ies) that need to be revised, where in the class hierarchy the concept(s) belong, and, importantly, whether or not there are existing patterns we can leverage in order to integrate the material

[1] See: https://www.govinfo.gov/content/pkg/PLAW-111publ203/html/PLAW-111publ203.htm
[2] See: https://www.esma.europa.eu/policy-rules/mifid-ii-and-mifir/
[3] See: https://www.bis.org/publ/bcbs239.pdf

6. Integrate the new content into the relevant ontology(ies), reusing existing classes and properties as much as possible and extending them as needed
7. Run at least one reasoner and perform SPARQL queries to ensure that the semantics seem reasonable and that the ontology(ies) remain logically consistent
8. Check the changes into GitHub and push them to a remote branch so that other members of the working group can review the results, automatically invoking the RDF serializer described below that ensures consistent serialization of the resulting RDF/XML via a custom Git hook
9. Create example individuals (or update existing individuals) and test whether or not the competency question(s) can now be answered by the ontology (as appropriate), and check-in any examples that might be used as guidance for FIBO users
10. Once the working group members are comfortable with the revisions, perform a pull request in GitHub to get a broader review, which automatically kicks off the infrastructure presented below; address any issues uncovered as a consequence
11. Once the pull request passes all of the stages in the publication cycle, at least two qualified reviewers must sign off (currently active members of at least one of the working groups plus other process team members have this privilege)
12. Finally, one of the process teams will merge the pull request after it has been approved.

We iterate through steps 6-9, as needed, depending on the issue's complexity, until we reach a consensus on the resulting ontologies. Additional information regarding the methodology, minimal criteria for metadata and ontology content, and unit-level hygiene testing is outlined in our ontology guide – see: https://github.com/edmcouncil/fibo/blob/master/ONTOLOGY_GUIDE.md.[4]

2. Reasoning challenge

FIBO has been growing over the years, and the task of validating its consistency has changed over this period accordingly. For the purpose of this paper, we will investigate its latest release - see: https://github.com/edmcouncil/fibo/releases/tag/master_2022Q2.

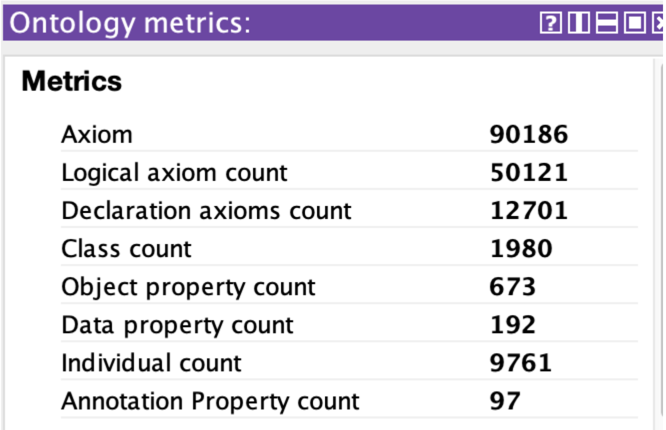
To appreciate the reasoning challenge in question, note that FIBO uses the full strength of the SROIQ(D) logic and is rich in terms of logical axioms - see Figures 1 and 2.

For the purposes of this paper, we run the Openllet reasoner (commit `97c43dd3`) as a standalone application and the Pellet and Hermit plugins (with the default configurations) to the Protege editor using a Ubuntu Linux virtual machine with 48 vCPUs (Intel Xeon 2.4 GHz) and 89 GiB RAM. The results can be found in table 1 – bear in mind that Pellet and Hermit check both the consistency of an ontology and the satisfiability of all its classes. The Hermit processes were terminated by us after 4 days and the Pellet processes after 2 days of running – that's where the unknown values come from.

We also tried Konclude, but it terminated with the following error:

```
{info} 12:51:55:173 >> Starting Konclude ...
```

[4]This section summarises a more detailed exposition of FIBO from [2].

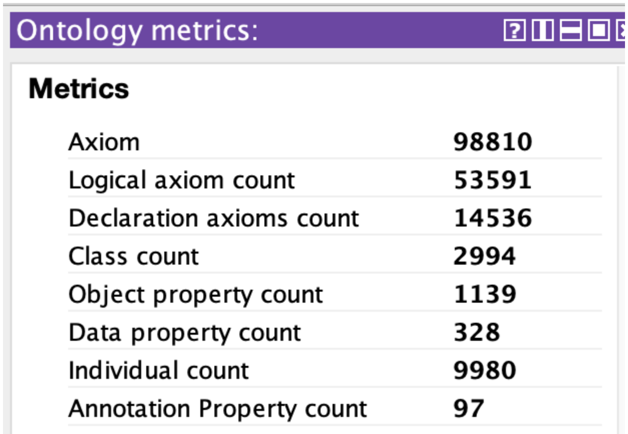


Ontology metrics:

Metrics

Axiom	90186
Logical axiom count	50121
Declaration axioms count	12701
Class count	1980
Object property count	673
Data property count	192
Individual count	9761
Annotation Property count	97

Figure 1: FIBO PROD metrics



Ontology metrics:

Metrics

Axiom	98810
Logical axiom count	53591
Declaration axioms count	14536
Class count	2994
Object property count	1139
Data property count	328
Individual count	9980
Annotation Property count	97

Figure 2: FIBO DEV metrics

```
{info} 12:51:55:174 >> Konclude - Uni Ulm Parallel Reasoner
{info} 12:51:55:174 >> Reasoner for the SROIQV(D) Description Logic, 64-bit,
Version v0.7.0-1135 - 91b3e331 (Mar 16 2021)

{info} 12:51:55:181 >> Starting consistency checking for 'MergedAboutFIBOProd.owl'.
{info} 12:51:55:183 >> Initializing reasoner. Creating calculation context.
{info} 12:51:55:189 >> Reasoner initialized with 1 processing unit(s).
{warning} 12:51:55:195 >> Annotations are currently not handled.
{error} 12:51:55:668 >> Skipped parsing of not supported datatype expression/axiom.
{error} 12:51:55:668 >> Skipped parsing of not supported datatype expression/axiom.
{error} 12:51:55:668 >> Skipped parsing of not supported datatype expression/axiom.
{info} 12:51:55:700 >> Query 'UnnamedConsistencyQuery' processed in '0' ms.
{info} 12:51:55:701 >> Preprocessing ontology 'http://konclude.com/test/kb'.
```

```
./Konclude: line 2: 73572 Bus error: 10          ./Binaries/Konclude $*
```

Ontology	Reasoner	Elapsed Time (milliseconds)
FIBO PROD	Openllet	16,987,331
FIBO DEV	Openllet	41,045,148
FIBO PROD	Pellet	unknown
FIBO DEV	Pellet	unknown
FIBO PROD	Hermit	unknown
FIBO DEV	Hermit	unknown

Table 1
Reasoners' performance

Now the problem with this performance is that we cannot integrate a simple consistency check into our DevOps infrastructure that supports the ontology development process. Currently, i.e., when consistency is not automatically validated, the process takes, on average, less than 30 minutes, so adding just the consistency check for PROD would extend it more than eight times. Obviously, finding unsatisfiable classes is out of the question.

So the challenge that the FIBO development process creates for DL reasoners is to reduce the time of automatic consistency check so that it is comparable to the whole process, i.e., its average execution takes less than 1,800 seconds.

3. Conclusion

The experiments described in this paper indicate that none of the bog-standard DL reasoners can support reasoning over logically complex ontologies like FIBO. The need for much faster and more scalable tools is imminent.

References

[1] E. F. Kendall, D. L. McGuinness, Ontology Engineering, Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool Publishers, 2019. doi:10.2200/S00834ED1V01Y201802WBE018.

[2] D. Allemang, P. Garbacz, P. Grądzki, E. Kendall, R. Trypuz, An infrastructure for collaborative ontology development, in: Formal Ontology in Information Systems, IOS Press, 2021, pp. 112–126.