Reasoning Performance Indicators for Ontology Design Patterns

Ontology Summit 2014 Track C Invited Talk

Karl Hammar

Jönköping University
karl.hammar@jth.hj.se

2014-02-06

About Karl

- PhD student at Linköping University
- Adjunct Lecturer at Jönköping University (BSc Program Manager)
- Researching reuse in Ontology Engineering, specifically Ontology Design Patterns (hereafter denoted ODPs)
- First time at Ontology Summit

Context of Work

- Part of speaker's Licentiate dissertation (Swedish postgraduate degree equivalent to 50 % of PhD)
- Licentiate project concerns development of quality characteristics, indicators, measurement methods, and recommendations for ODPs
- Project motivated by the lack of prior studies on ODP structure and design, as evidenced by the literature survey published [1] at the Workshop on Ontology Patterns (WOP) at ISWC in 2010
- The work presented in this talk was introduced in a paper by the same title [2], presented at WOP 2013

Motivation

- Semantic technologies, and consequently ontologies, are being used in new contexts where acceptable reasoning performance is a key requirement:
    - Stream reasoning / Complex event processing
    - Ubiquitous computing / Intelligent agents
    - Large-scale Information Extraction
- For many such tasks, existing ontologies are insufficient, and more specific application-focused ontologies need to be developed. ODPs are intended to simplify such work.
- ODPs are generally employed by specializing and importing pattern axioms into a the ontology.
- Consequently: if ODPs are to be used to aid in developing ontologies for reasoning-intensive purposes, they need to display suitable reasoning characteristics. We need to know more about what kind of performance indicators that can be applied to them, and how good existing patterns are.

Research Questions

1. Which existing performance indicators from literature known to affect the performance of reasoning with ontologies are also applicable to Ontology Design Patterns?
2. How do these performance indicators vary across published Ontology Design Patterns?
3. Which recommendations on reasoning-efficient Ontology Design Pattern design can be made, based on the answers to the above questions?

Note: The work presented here covers only Content Ontology Design Patterns, that is, small reusable OWL building blocks intended to be specialized.

Method Overview

1. Initial literature study to isolate candidate indicators. Covers some of the top conferences in the field, and their accompanying workshops: ISWC, ESWC, K-CAP, and EKAW, for the time period 2005 (initial appearance of ODPs) through 2012.
2. Evaluation study on how the candidate indicators found in the first step vary over the Ontology Design Patterns published in two popular pattern repositories on the web (`http://ontologydesignpatterns.org` and `http://odps.sourceforge.net`).

Literature Study

1. Downloaded all papers matching selection criteria (hundreds).
2. Read the papers' abstracts. Selected for further study those papers that in the abstract mentioned ontology metrics, indicators, language expressivity effects, classification performance improvements or performance analyses (16 papers).
3. Read the full papers. Selected as candidate indicators such performance-altering structures that are likely to exist or be relevant in small or reusable ontology modules, i.e., ODPs (8 papers).

Indicator Variance in Repositories

All pattern OWL files from the two repositories were downloaded, and the candidate indicator values were calculated for each[1]. The following steps were then repeated for each candidate indicator:

1. Sort all ODPs by the studied indicator.
2. Observe correlation effects against other indicators. Can any direct or inverse correlations be observed for whole or part of the set of patterns?
3. Observe distribution of values. Do the indicator values for the different patterns vary widely or not? Is the distribution even or clustered?
4. For any interesting observation made above, attempt to find an underlying reason or explanation for the observation, grounded in the OWL ontology language and established ODP usage or ontology engineering methods.

---

[1] A pluggable tool for calculating indicator values is available at `https://github.com/hammar/OntoStats`

Findings

- 11 reasoning performance indicators from literature, giving rise to:
- 12 recommendations on reasoning-friendly ODP design, in turn suggesting:
- 3 design principles for ODP development and use

Indicators

Table : ODP performance indicators found via literature study.

Indicator	Source
Average class in-degree	[3]
Average class out-degree	[3]
Cyclomatic complexity	[3]
Depth of inheritance	[4]
Existential quantification count	[3]
General concept inclusion count	[5]
OWL Horst adherence	[6, 7]
OWL 2 EL adherence	[8, 9]
OWL 2 QL adherence	[8, 9]
OWL 2 RL adherence	[8, 9]
Tree impurity / Tangledness	[3, 10]

Note: Cyclometic Complexity and OWL 2 Profile Adherence indicators not subsequently analyzed. The former unpractical to measure reliably, the latter extensively covered by Horridge et al. [8].

Avgerage Class In-Degree

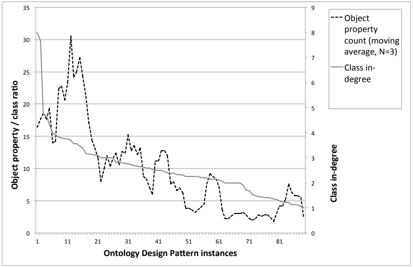

Figure : Class in-degree and object property per class distributions

- Varies between 0.75 and 8. Median 2.39, average 2.6
- Most below four - though a few stand out
- Appears to relate to object property count
- Cause: domain/range restriction axioms?

Average Class Out-Degree

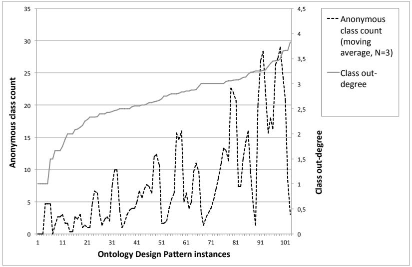

Figure : Class out-degree and Anonymous class count distributions

- Varies between 1 and 3.83. Median 2.75, average 2.64.
- Appears more common where class restrictions used more.
- Cause: subClassOf / equivalentClass axioms?

Depth of Inheritance

- Varies between 0 and 5.3. Median 1.5, average 1.7.
- Overall: Ontology Design Patterns have a shallow subsumption hierarchy. There is a large group (38 of 103 patterns) that display a depth of one or less.
- The ODPs from `http://odps.sourceforge.net` are generally deeper. A complicating factor is that these include examples in the same OWL file as the pattern itself, making direct comparison against the patterns from `http://ontologydesignpatterns.org` impossible.

Existential Quantifications

- 60 ODPs contain existential quantification axioms or (semantically equivalent) cardinality restriction axioms of size 1.
- 31 patterns contain one or two such axioms, most often used sparingly, when required.
- 29 patterns contain three or more such axioms. These are sometimes used in seemingly unneeded ways (restating axioms that were already asserted on superclasses, asserting coexistance of two individuals where one might well exist on its own, etc).

Note: These two groupings are obviously a simplification, and counterexamples exist.

General Concept Inclusion

- General Concept Inclusion axioms are axioms that have a complex class expression on the left hand side of a subclass or equivalent class axiom.
- Example[2]:

  ```
  (HeartDisease and hasLocation some HeartValve)
  SubClassOf:  CriticalDisease
  ```
- No patterns displayed General Concept Inclusion axioms.
- The likely cause of this is that there is little support for GCIs in common tools.
- The performance effects of GCIs in ODPs are negligible in practice today. The recommendation is to keep it that way.

[2] Courtesy of `http://ontogenesis.knowledgeblog.org/1288`

Tree Impurity / Tangledness

- Asserted tangledness in Ontology Design Patterns appears to be rare: only three ODPs displayed any such tangledness at all. All three of these patterns had only one single multi-parent class.
- Caveat: inferred tangledness may be significantly higher, but this is rather difficult to measure.

Reasoning-friendly ODP Design Recommendations

For a full list of design recommendations, see [2]

- Limit the use of class restrictions (i.e., enumerations, property restrictions, intersections, unions, or complements) to the minimum required by the ODP requirements.
- Limit the use of existential quantification axioms to the minimum required by the ODP requirements. Even if the addition of an axiom makes "real world" intuitive sense, consider whether it is strictly necessary for the purpose of the ODP.
- Rewrite GCI axioms into equivalent non-GCI axioms if possible.
- Limit the number of property domain and range definitions to the minimum required by the ODP requirements, as these may otherwise give rise to inefficient high class in-degree values.
- In constructing class restriction axioms, avoid restrictions that give rise to inferred tangledness, i.e., axioms which give rise to unions or intersections of classes from different branches in the subsumption tree.

Design Principles

- *Don't overspecify* - make requirements explicit, and fulfill only those requirements. Avoid completeness for the sake of neatness!
- *Rewrite if needed* - the solution proposed by the ODP may be sound, even though the provided building block is unsuitable for reasoning.
- *Ensure sufficient documentation exists* - supporting both of the above principles requires documentation on requirements, problem, and solution exist!

References I

K. Hammar and K. Sandkuhl.
The State of Ontology Pattern Research: A Systematic Review of ISWC, ESWC and ASWC 2005–2009.
In *Workshop on Ontology Patterns: Papers and Patterns from the ISWC workshop*, pages 5–17, 2010.

Karl Hammar.
Reasoning performance indicators for ontology design patterns.
In *Proceedings of the 3rd Workshop on Ontology Patterns :*, CEUR Workshop Proceedings, 2013.

Y-B. Kang, Y-F. Li, and S. Krishnaswamy.
Predicting Reasoning Performance Using Ontology Metrics.
In *The Semantic Web – ISWC 2012*, pages 198–214, 2012.

References II

P. LePendu, N. Noy, C. Jonquet, P. Alexander, N. Shah, and M. Musen.
Optimize First, Buy Later: Analyzing Metrics to Ramp-up Very Large Knowledge Bases.
In *The Semantic Web – ISWC 2010*, pages 486–501. Springer, 2010.

R. S. Goncalves, B. Parsia, and U. Sattler.
Performance Heterogeneity and Approximate Reasoning in Description Logic Ontologies.
In *The Semantic Web – ISWC 2012*, pages 82–98, 2012.

J. Urbani, S. Kotoulas, E. Oren, and F. Van Harmelen.
Scalable Distributed Reasoning using MapReduce.
In *The Semantic Web - ISWC 2009*, pages 634–649. Springer, 2009.

References III

J. Urbani, S. Kotoulas, J. Maassen, F. Van Harmelen, and H. Bal.
OWL reasoning with WebPIE: calculating the closure of 100 billion triples.
In *The Semantic Web: Research and Applications*, pages 213–227. Springer, 2010.

M. Horridge, M. E. Aranguren, J. Mortensen, M. Musen, and N. F. Noy.
Ontology Design Pattern Language Expressivity Requirements.
In *Proceedings of the 3rd Workshop on Ontology Patterns*, 2012.

W3C.
OWL 2 Web Ontology Language Profiles (Second Edition).
http://www.w3.org/TR/owl2-profiles/, checked on: 2013-02-27.

A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann.
Modelling Ontology Evaluation and Validation.
In *The Semantic Web: Research and Applications*, pages 140–154. Springer, 2006.