Analysing the Evolution of the NCI Thesaurus

Rafael S. Gonçalves
goncalvj@cs.man.ac.uk
School of Computer Science
University of Manchester, UK

Bijan Parsia
bparsia@cs.man.ac.uk
School of Computer Science
University of Manchester, UK

Uli Sattler
sattler@cs.man.ac.uk
School of Computer Science
University of Manchester, UK

Abstract

The National Cancer Institute (NCI) Thesaurus (NCIt) is a biomedical ontology which has been developed for over a decade. Nearly every month for the past 8 years the NCI has published an updated version of the NCIt to the Web in a variety of formats. We collected all 88 versions of the NCIt available in OWL format since 2003, and conducted a cross-sectional study on this corpus to investigate and characterize the evolution of the NCIt. In particular, we gathered and analyzed various axiom and entity statistics, and carried out a reasoner performance test over the corpus. Additionally, we extracted two complete sets of pairwise, consecutive diffs: the first set was generated by a purely syntactic difference analysis (based on OWL's notion of "structural equivalence"); for the second set, we also checked whether the additions or removals changed the set of entailments between versions. We discovered a high level of "merely syntactic" removals and additions. We develop a categorization of such changes based on a heuristic inference of the impact of the change. As a result, not only do we get a rich, purely analytic characterization of the change history of the NCIt, but also we generate a realistic test corpus for incremental classification.

1 Introduction

Since 2000, the Enterprise Vocabulary Services (EVS) project of the National Cancer Institute of the United States (NCI) has been developing and publishing a biomedical reference terminology: National Cancer Institute Thesaurus (NCIt).[1] Part of the development methodology for the NCIt is to base it on an ontology encoded in a description logic. Since 2003, the monthly releases of the NCIt have included a version in the W3C's Web Ontology Language (OWL) resulting in a set of 88 OWL ontologies. While there have been a number of analyses of the NCIt [7, 5, 8, 1, 14, 12, 13], these have all been of particular versions (typically, either a known snapshot, such as the one described in [8], or the "latest" version). What history oriented comments there are typically merely give an indication of the overall rate of growth (e.g., 700-900 "new entries" a month).

However, 88 temporally evenly-spaced versions of the same ontology form a unique resource for studying ontology evolution. Since the NCIt is a central resource for a lot of ontology engineering research,[2] and the full corpus is freely downloadable[3] from the web, it is surprising that there has been no analysis of this corpus. In fact, the general approach is to take some particular version of the NCIt without much regard to the typicality of that version. Nor has there been any systematic attempt to characterize the evolution of the NCIt. While the EVS does provide "concept change" logs with each release, these are intended for consumers of the terminology, not people interested in the ontological portions of the NCIt. Internally, the EVS keeps detailed "edit based" change logs [6] which are deeply embedded in the development and quality assurance process (according to [6]), however these are not public. Thus modellers who wish to understand the scope of the changes from version to version need to rely on post-facto diff analysis.

The full corpus is challenging to work with. The latest version has nearly 90,000 classes and over 1.2 million axioms. The total corpus weighs in at 12 GB for the uncompressed OWL files. In this paper, we provide a detailed analysis of the 88 versions of the NCIt, charting its change over the past 8 years. In addition to synchronic statistics about each version (in both asserted and fully classified form), we apply two post-facto diffing techniques to each sequential pair of versions. Strikingly, a significant portion of the changes to the logical axioms of the NCIt do not change the set of entailments of the new version, that is, have no logical impact. This highlights the shortcomings of

[1] http://ncit.nci.nih.gov/

[2] In addition to driving the development of the Protégé OWL plugin, it has been used in research on reasoning [11], ontology diff analysis [10], modularity [4], to name but a few.

[3] http://evs.nci.nih.gov/ftp1/NCI_Thesaurus

purely syntactic diff approaches. In order to investigate this phenomenon, we develop a classification of these logically ineffectual changes.

Finally, we examine the behavior of three freely available OWL 2 DL reasoners,[4] FaCT++, Pellet (both standard and incremental modes), and HermiT on the entire corpus. As a result, we believe that reasoner benchmarking using any single version of the NCIt is likely to be misleading.

2 Preliminaries

We assume the reader to be reasonably familiar with OWL [15], as well as the underlying description logics (DLs) [9], though detailed knowledge is not required. We do use the notion of entailment [3], which is identical to the standard first order logic entailment (albeit restricted to certain syntactic forms for consequences, typically atomic subsumption). When comparing two versions of NCIt we refer to an earlier version as $\mathcal{O}_{old}$, and a later one as $\mathcal{O}_{new}$.

An axiom α is asserted if $\alpha \in \mathcal{O}$; and α is inferred if not asserted but $\mathcal{O} \models \alpha$. A subclass relation gives rise to a transitive relation on class names, whereas equivalences give rise to cycles.[5] The number of atomic equivalences is the number of classes involved in cycles minus 1. The number of atomic subsumptions is the number of direct subclass relations between cycles (after cycles have been collapsed to nodes). Classes which are equivalent to $\top$ or $\bot$ are not counted.

3 The NCIt corpus

The NCIt archive contains 88 versions of the ontology in OWL format, two of which were unparsable (05.03F and 05.04d) with the OWL API,[6] and consequently Protégé.[7] The experiment machine is an Intel Xeon Quad-Core 3.20GHz, with 12Gb DDR3 RAM dedicated to the Java Virtual Machine (JVM v1.5). The system runs Mac OS X 10.6.7, and all tests were run using the OWL API (v3.1).

All gathered test data is available from `https://sites.google.com/site/ncitanalysis/`, a part of it is published on Google Public Data Explorer,[8] and can be visualised at `http://www.google.com/publicdata/overview?ds=fion1t13i4l96_`. A Virtual Machine (VM) to replicate our results, with the same settings used, can be supplied upon request.

[4] For more details on these, and other OWL reasoners, see `http://owl.cs.manchester.ac.uk/reasoners.html`.

[5] See technical report at `https://sites.google.com/site/ncitanalysis/`.

[6] `http://owlapi.sourceforge.net/`

[7] `http://protege.stanford.edu/`

[8] `http://www.google.com/publicdata/home`

3.1 Parsing times

Given the size of the NCIt, it is often presumed that the parsing time is high. However, we argue that nowadays the parse time of the NCIt is a non-issue. The parsing times across the NCIt corpus increased linearly, and there was very little difference between the times within the OWL API and Protégé (v4.1).[9] An early version, 6, is parsed in 6.1 seconds using the OWL API (6.8 seconds in Protégé), and having the logical part split from the non-logical (annotations) as two distinct ontologies, the latter loads in 5 seconds in the OWL API (5.9 seconds in Protégé), and the earlier in 3 seconds (3.8 seconds in Protégé). The latest version in the corpus, 86, loads in 14.5 seconds using the OWL API (around 17.2 seconds in Protégé); the logical part takes 5.2 seconds (6.4 seconds in Protégé), and the non-logical 11.3 seconds (12.4 seconds in Protégé). We can see that the logical part does not create much overhead on the parsing times of the NCIt. In fact the non-logical part alone takes nearly as much as the whole ontology to load.

3.2 Asserted information

We observe a steady growth in terms of axioms over the corpus (see Figure 1), with a few exceptions; there was a drop in annotation axioms from version 59 to 60, and also a drop in subclass and annotation axioms from 16 to 17. The majority of axioms in each NCIt version are entity annotations, with an average of 84% over the whole set (the minimum percentage is 75%). Aside from the removal of around 180,000 entity annotations (from 59 to 60), the annotation growth is consistent across the corpus. The expressivity of the underlying logic increased throughout the NCIt timeline, except that from versions 14 and 62 to their respective subsequent versions the use of datatypes in data property ranges was dropped. In versions 14 and 16 we came across a parsing issue; these ontologies have an annotation property and a data property both named “code”. This presumably results in the parsing of data property assertions rather than annotations, therefore leading to the mass-creation of individuals. However, this is a tool artifact rather than a fundamental modelling choice. By excluding these two versions, we observe that the growth in terms of entities (namely classes and properties) is linear across the corpus. Individuals are only present in three NCIt versions; the mentioned 14 and 16, and version 27 where the same issue occurs - with the exception that no terms use these data properties, and so the mass-generation of individuals does not take place.

[9] However Protégé takes some time to render the class hierarchy, which was not accounted for.

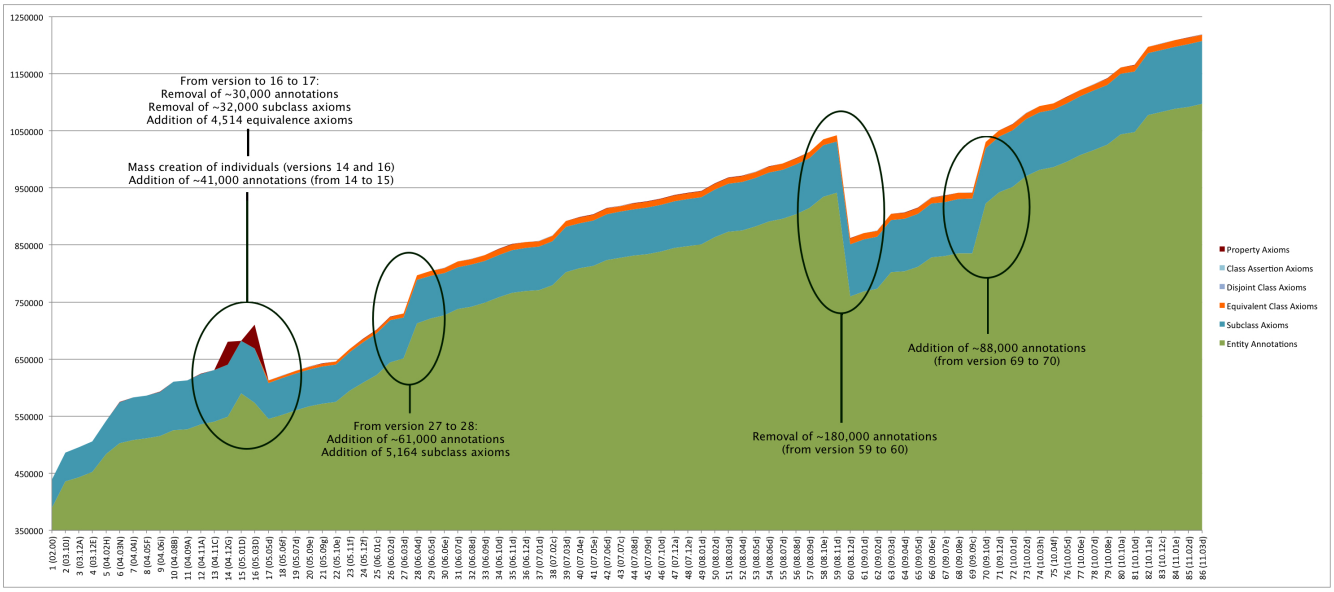

Figure 1. Axiomatic growth of NCIt. Annotation axioms dominate

3.3 Entailments analysis

In order to analyse the logical content of the NCIt, we used an entailment counter which outputs all atomic subsumptions and equivalences. Aside from three particular cases in the corpus, the ratio of asserted against inferred entailments is constant (see Figure 2). The mentioned cases are versions 16, 58 and 59, all of which contain a high number of unsatisfiable classes (37,436, 21,819 and 21,866 respectively). However they are distinct cases; on the one hand, version 16 has only 4,009 satisfiable classes out of 41,535, therefore causing a high subsumption entailment drop. Versions 58 and 59, on the other hand, reveal a rise in equivalences which causes the decline in subsumptions.

3.4 Reasoner Performance

It is often the case that, for reasoner testing, only a few or even one ontology version is tested against. There is no reported reasoner benchmark using a corpus of the same kind as the one here described. So, in the process of analysing the NCIt, we evaluate how modern reasoners handle all published OWL versions of the NCIt. Three major DL reasoners were put to the test; FaCT++ (v1.5.1), Pellet (v2.2.2) and HermiT (v1.3.3). Since we also possess the axioms in the difference between NCIt versions, this allows us to test incremental reasoning as well.[10] We note (see Figure 3) that, of the three reasoners put to test, FaCT++ behaves consistently faster than Pellet and HermiT. The latter on the other hand shows a poorer performance, and even did not terminate (after more than 10 hours) when processing NCIt versions 14 and 16. This performance test also showed that, to some degree, incremental reasoning provides a big advantage when handling large ontologies, in terms of reasoning time. However, like HermiT, it did not terminate upon classifying versions 14 and 16. This is due to the abundance of individuals: incremental reasoning is based on locality-based modules [2], and these behave poorly in the presence of individuals. Aside from these two cases, the timings gathered using the incremental classifier were consistently below 5 seconds per version, across the corpus.

[10] As implemented within Pellet.

4 Comparing NCIt versions

The first step in computing the difference between versions is purely syntactic, based on OWL's notion of "structural equivalence". This diff gives us a set of syntactic additions and removals; $\mathcal{O}_{old} \setminus \mathcal{O}_{new}$ and $\mathcal{O}_{new} \setminus \mathcal{O}_{old}$ respectively. In the second step we apply an entailment diff, that checks which axioms in the additions (or removals) affects the set of entailments of $\mathcal{O}_{old}$ (or $\mathcal{O}_{new}$). In other words, we check whether $\mathcal{O}_{old}$ (or $\mathcal{O}_{new}$) entails added (or deleted) axioms. This results in a set of non-retracting removals, and another of non-adding additions. Finally, for all axioms in these two sets (that have no logical impact), we apply a categorisation method that identifies the kind of (non-)impact caused.[11] As such, an axiom is called:

[11] For precise definitions see technical report at `https://sites.google.com/site/ncitanalysis/`.

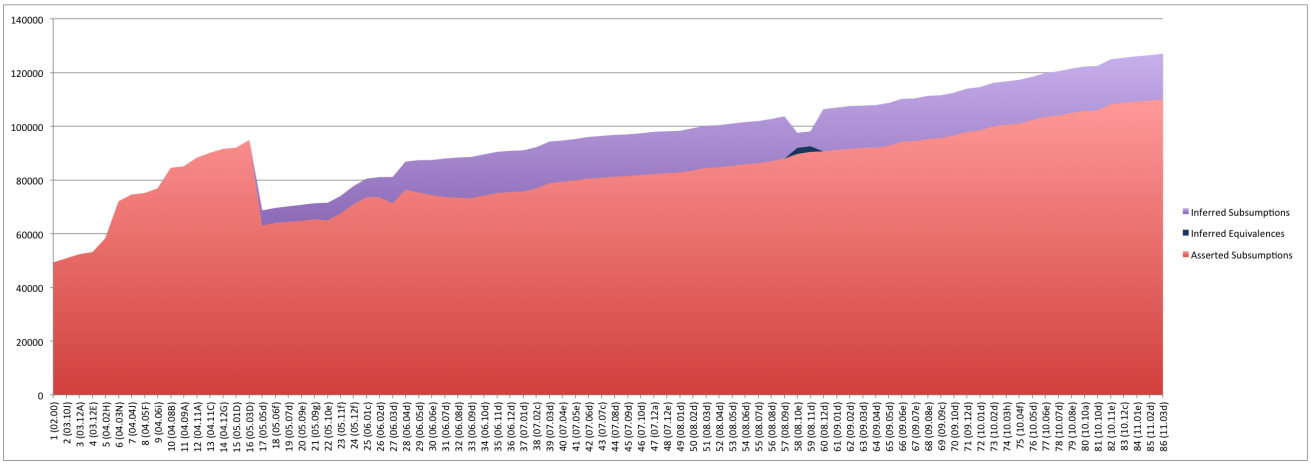

Figure 2. Entailment growth of NCIt; asserted vs inferred (number of axioms)

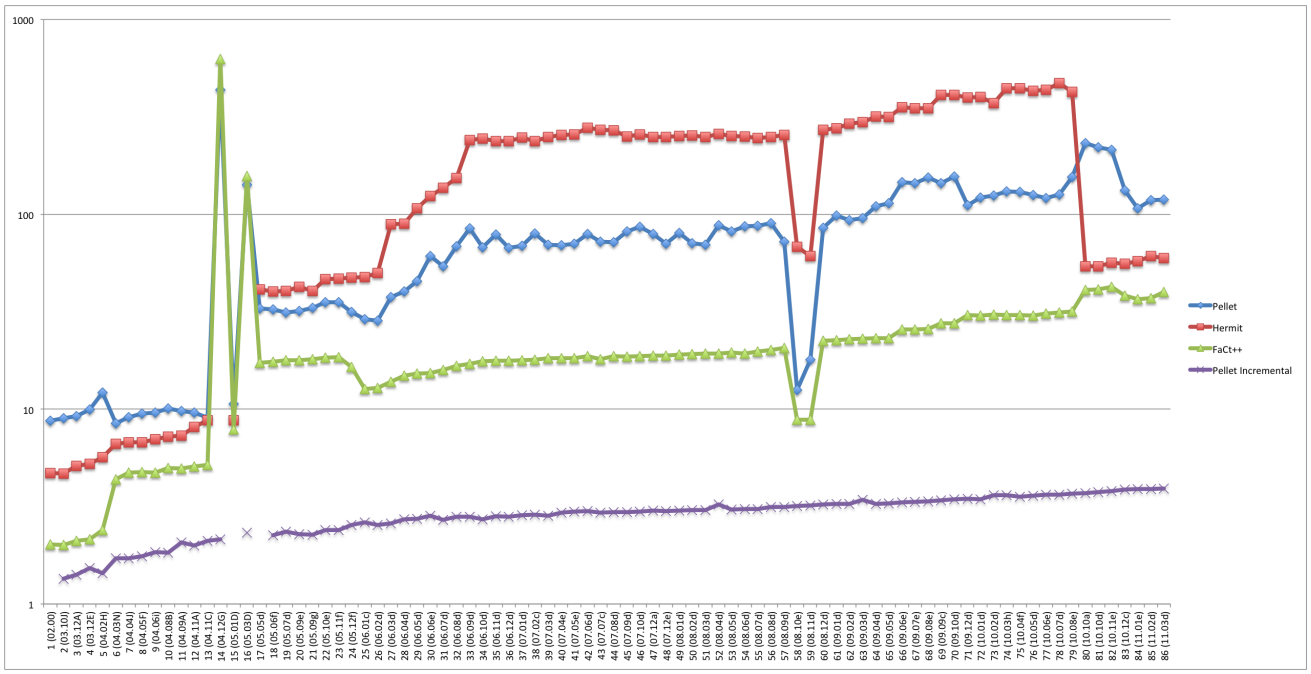

Figure 3. Reasoner performance across NCIt (time in seconds, logarithmic scale)

- Strenghtened; if the axiom is less constraining than its replacement (e.g. $A \sqsubseteq B$ changed into $A \sqsubseteq B \sqcap C$).
- Weakened;[12] if the axiom is looser than its original.
- Rewritten; if both rewriting and rewritee entail one another (e.g. $A \equiv B$ changed into $A \sqsubseteq B, B \sqsubseteq A$).
- Redundant; if the added or removed axiom is entailed by $\mathcal{O}_{old} \cap \mathcal{O}_{new}$.

[12]The reverse effect of strengthening.

Furthermore, we devised a particular notion of annotation difference, specifically taking into account common annotation types in NCIt. As such we compute the difference of "definition" type annotations, and entity labels (the most relevant annotations in NCIt). From these we differentiate additions, removals and edits. In the case of definition edits we also check, based on the Levenshtein distance between strings, whether details were added, deleted or minor refinements carried out (e.g. capitalisation of definition text, updating entity names to their most recent version).

4.1 Logical Difference

The logical difference throughout the NCIt timeline consists mostly of subclass axioms (see Figure 4, and for complete results the mentioned website), with an average of 90% (excluding versions 14 and 16). Logical removals are not substancial, and consist of mostly subclass axioms. However it should be noted that, despite the large number of annotations, NCIt developers devoted considerable efforts towards the logical part of the ontology. On average, almost 800 classes were added each month throughout the corpus. In terms of deleted classes, these are roughly at 70 per month. However this metric is distorted by the mass class removal (or renaming) in version 6. This version is a curious case, where a large number of classes (5170) were removed, along with corresponding annotations and 14,418 subclass axioms. This indicates a possible re-modelling of the NCIt at this point. More evidence to support this include the addition of 30,859 subclass axioms, 9070 classes and 23 object properties (and 240,000 entity annotation axioms). Similarly in version 25 a series of changes were carried out to the subsumption hierarchy, with the removal of 8,231 subclass axioms and 2,899 equivalent class axioms compared to the previous version, and also the addition of 10,591 subclass axioms and 3,011 equivalent class axioms.

Overall there is a fair amount of syntactic removals in the corpus (see Figure 4), which, semantically speaking, are non-retracting removals (i.e. still entailed by $\mathcal{O}_{new}$). The majority of these turned out to be strenghtened axioms, and some removed redundancies. A small proportion of the syntactic additions were non-adding additions, among these in many cases there were added redundancies, but the majority of these axioms were weakened axioms. We also identified a number of rewrites in the corpus. Particularly from version 32 to 33 there are 227 re-written axioms.

We noted a recurring trend throughout the NCIt corpus, which is the addition of redundancies. This trend has more incidence up until version 8, but there are high values in the rest of the corpus as well, such as version 35 with 145 added redundant axioms (see Figure 4). The highest value found is in version 5, where 190 redundant axioms were added. Upon investigating this phenomenon, we found that such added redundancies were, in most or all cases, entailments from previous versions. These entailments were those derived from the transitivity of the subclass relationship, e.g. $\mathcal{O}_1 = \{\alpha_1 : A \sqsubseteq \exists r.B, \alpha_2 : C \sqsubseteq A\}$, $\mathcal{O}_2 = \{\alpha_1, \alpha_2, \alpha_3 : C \sqsubseteq \exists r.B\}$. From the example we see that α_3 is redundant; $C \sqsubseteq A$ suffices for $C \sqsubseteq \exists r.B$ to hold.

4.2 Annotations Difference

The majority of changes across NCIt versions are based on annotations, of which we observed were mostly additions. In terms of changes to NCIt definitions, there is a significant number of edits in entity annotation definitions (55% of definition changes), with around 1,158 edits per version. Of these, 42% of definitions were increased, 46% were reduced, while only 12% were refined with minor details. On average, 424 definitions were removed per version, and 542 were added (20% and 25% of definition changes, accordingly). In version 6 there was a major overhaul in definitions, with 17,343 removals and 17,009 additions. It is the only version where no edits took place, meaning that the definitions were completely removed (possibly to be edited "offline"), and then re-inserted into the ontology. Version 63 also had a fair share of annotations changes, with the addition of over 700,000 annotation axioms, and the removal of around 675,000 annotations.

The difference in labels between versions consists mostly of edits, with an average of 955 per version. In version 63 there was a mass refining of labels, with 72,414 changes. This version contains 72,674 entities (of which 72,402 classes), so the degree of label changes indicates a nearly complete refinement of labels. In terms of additions and removals, very few took place across the corpus (averages of 2.82 and 0.13 per version, respectively). The only exception is the addition of 220 labels in version 33.

5 Discussion and Conclusions

The analysis of the NCIt's evolution shows that its major focus is (not surprisingly) annotations, while there were consistent efforts on the logical part. Nevertheless, a significant portion of the changes did not alter the set of entailments of its subsequent (or earlier) version. Throughout the corpus, there is a strikingly[13] high number of added redundancies. We identified at least two re-modelling phases in versions 6 and 25.

With respect to the diffing methods applied, we conclude that merely syntactic diffs do not provide enough insight into the type and impact of changes carried out. Employing our semantic diff allows us to identify axioms in the changes that are logically redundant, but it still does not convey the desired insight. The axiom categorisation we devised allows ontology engineers to understand the impact of their changes, and possibly refine their changes before publishing newer versions. The advantages of doing so are the awareness of the type of changes carried out, and in particular if redundancies are present, these would be encouraged to be revised.

The reasoner performance test, as carried out in this diachronic study, suggests that reasoner benchmarking should take into account more than a single version of an ontology. We have shown in Section 3.4 that reasoning times fluctuate

[13] Particularly given the type of redundancy being added (Section 4.1).

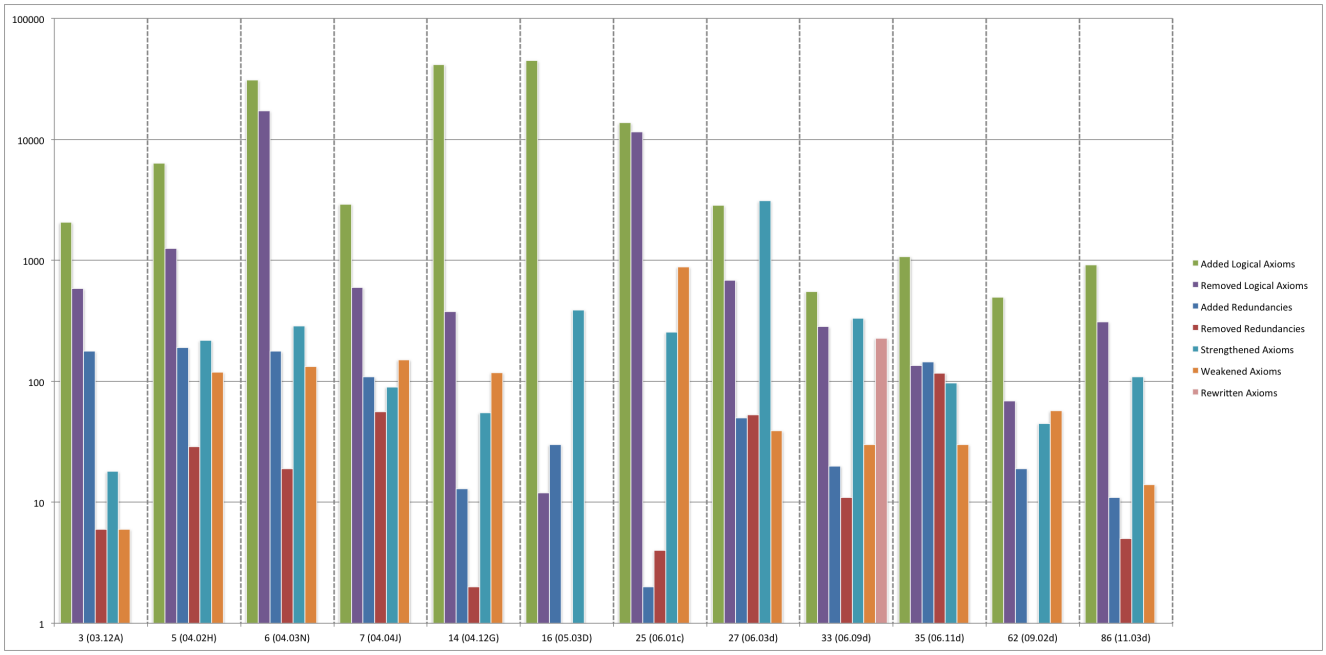

Figure 4. Logical diff across selected versions of the NCIt (number of axioms)

(significantly in some cases) across the corpus, particularly HermiT from version 79 to 80. Furthermore, we demonstrate the advantage (in terms of time) of using incremental reasoning for ontology engineering tasks, especially when large and complex ontologies are involved.

References

[1] W. Ceusters, B. Smith, and L. Goldberg. A terminological and ontological analysis of the NCI Thesaurus. *Methods of Information in Medicine*, 44(4):498–507, 2005.

[2] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *J. of Artificial Intelligence Research*, 31:273–318, 2008.

[3] B. Cuenca Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *J. of Web Semantics*, 2008.

[4] B. Cuenca Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularity and web ontologies. In P. Doherty, J. Mylopoulos, and C. A. Welty, editors, *Proc. of KR-06*, pages 198–209. AAAI Press, 2006.

[5] S. de Coronado, M. W. Haber, N. Sioutos, M. S. Tuttle, and L. W. Wright. NCI Thesaurus: Using science-based terminology to integrate cancer research results. *Studies in Health Technology and Informatics*, 107(1):33–37, 2004.

[6] S. de Coronado, L. W. Wright, G. Fragoso, M. W. Haber, E. A. Hahn-Dantona, F. W. Hartel, S. L. Quan, T. Safran, N. Thomas, and L. Whiteman. The NCI Thesaurus quality assurance life cycle. *Journal of Biomedical Informatics*, 42(3):530–539, June 2009.

[7] G. Fragoso, S. de Coronado, M. Haber, F. Hartel, and L. Wright. Overview and utilization of the NCI Thesaurus. *Comparative and Functional Genomics*, 5(8):648–654, 2004.

[8] J. Golbeck, G. Fragoso, F. Hartel, J. Hendler, J. Oberthaler, and B. Parsia. The National Cancer Institute's Thésaurus and ontology. *J. of Web Semantics*, 1(1):75–80, 2003.

[9] I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible $\mathcal{SROIQ}$. In *Proc. of KR-06*, pages 57–67, 2006.

[10] E. Jiménez-Ruiz, B. Cuenca Grau, I. Horrocks, and R. Berlanga Llavori. Building ontologies collaboratively using ContentCVS. In B. Cuenca Grau, I. Horrocks, B. Motik, and U. Sattler, editors, *Proc. of DL 2009*, volume 477 of *CEUR (`http://ceur-ws.org/`)*. CEUR-WS.org, 2009.

[11] Y. Kazakov. Consequence-driven reasoning for horn $\mathcal{SHIQ}$ ontologies. In *Proc. of IJCAI-09*, pages 2040–2045, 2009.

[12] N. F. Noy, S. de Coronado, H. Solbrig, G. Fragoso, F. W. Hartel, and M. A. Musen. Representing the nci thesaurus in owl. *Applied Ontology*, 3(3):173–190, 2008.

[13] S. Schulz, D. Schober, I. Tudose, and H. Stenzhorn. The pitfalls of thesaurus ontologization – the case of the NCI Thesaurus. In *Proc. of the 2010 AMIA Symposium*, pages 727–731, Washington, D.C., October 2010.

[14] N. Sioutos, S. de Coronado, M. W. Haber, F. W. Hartel, W.-L. Shaiu, and L. W. Wright. NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, 40(1):30–43, February 2007.

[15] W3C OWL Working Group. OWL 2 Web Ontology Language: Document overview. W3C Recommendation, 27 Oct 2009. `http://www.w3.org/TR/owl2-overview/`.