Two Case Studies of Ontology Validation

Doug Foxvog

University of Maryland Baltimore County, Baltimore, MD, USA
doug@foxvog.org

Abstract. As ontologies and ontology repositories have proliferated, the need for a discipline of ontology validation and quality assurance has grown more urgent. This report describes two case studies of ontology validation by converting ontologies to a more powerful reasoning language and analyzing them using logical queries. The lessons learned and directions for continuing work are discussed.

Keywords: ontology, quality assurance, validation

1 Introduction

In computer science, an ontology is a formalization of the concepts of a specific field, specifying the types (classes) of things that are dealt with in the field, relations that may apply among instances of those classes, rules applying to instances of those classes, and possibly specific instances of those classes. It may define subsumption and disjointness between classes or relations and may constrain the argument types of relations. An ontology is not a definition of terms in a natural language, although many ontologies provide mappings between the terms of the ontology and natural language terms.

If an ontology accurately constrains the relations and classes of a model of the domain with restrictions that prevent assertions that could not be true in the modeled domain and does so in a logically consistent manner, then it can be used to encode valid information in the field, conclude additional information that is implied by the stated information, and detect or block statements that are inconsistent with the domain model.

However, an ontology that does not accurately model the domain would allow logically invalid statements to be asserted, prevent true statements from being made, or both. An ontology may be incorrect not only due to some of its statements being incorrect, but also due to missing assertions. An ontology that accurately encodes a domain model and yet is logically invalid indicates that the model itself is invalid.
For these reasons, it is important to validate ontologies before use and whenever they are modified. Not only can sets of logically inconsistent statements be identified, but omission of argument constraints and class disjointness assertions can be flagged.

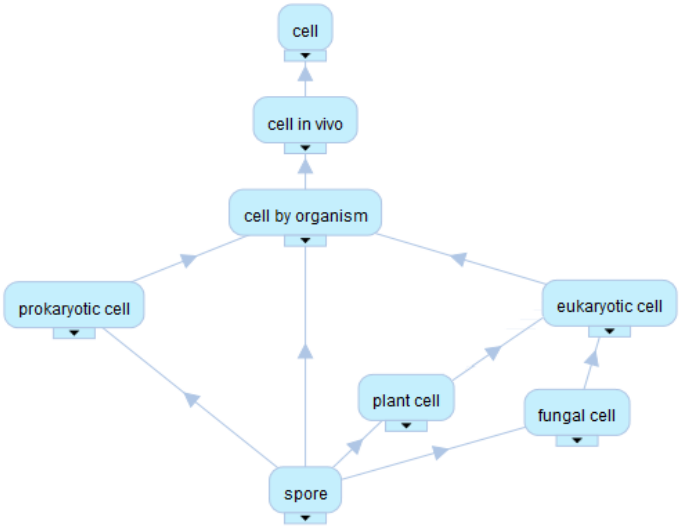

Fig. 1. Class with three disjoint superclasses[1]

Our method does not cover ways to verify if an ontology corresponds to reality or to an external model, but deals with design flaws and logical issues that could be examined with an inference engine. This paper presents the results of validating two standard ontologies with strong user bases and communities (the Cell Line Ontology and Plant Ontology described in Section 2) using this technique.

As the ontology language for the selected case studies lacks the reasoning capabilities for logically detecting all the considered types of ontology flaws, we translated the ontologies into a richer language (CycL [1]) in order to perform the analysis[2]. Some errors are flagged as the translated ontology is being input to the Cyc system, while others can only be detected by asking the inference engine queries about the ontology.

2 Source of Ontologies for Validation

The National Center for Biomedical Ontology maintains hundreds of ontologies for the biomedical field with versions in several formats [2].

We selected as case studies two associated ontologies hosted by the NCBO: the Cell Line Ontology and the Plant Ontology, downloading them in the OBO format [3]. The 5 May 2009 version of the Cell Line Ontology [4] included thousands of cell types including types of animal cells, plant cells, fungal cells, and prokaryotic cells.

[1] http://bioportal.bioontology.org/ontologies/39927/?p=terms&conceptid=CL%3A0000522

[2] Although CycL is formally an undecidable language, the queries used here (taxonomic, local closed world, queries ranging over locally defined predicates, etc.) were not. The Cyc inference engine specifies whether lists of answers provided are known to be complete.

The Plant Ontology [5] covers plant anatomy (including types of plant cells), morphology, growth, and development. The Cell Line Ontology and Plant Ontology had hundreds of pairs of plant cell concepts, with the same name and same or similar English definition, but different IDs.

Disjointness violations we detected in the Cell Line Ontology (see 3.1) suggested that at least the one ontology had not been created with automated logical verification of its statements. We decided to do a more complete analysis of the two ontologies to determine if there were additional issues.

3 Cell Line Ontology

3.1 Introduction

In the Cell Line Ontology terms for several classes of cells, such as "epidermal cell," which are used by botanists and anatomists to refer to cells with similar functions in both plants and animals, had been created with some plant cell subclasses and other animal cell subclasses. However, at some point these terms were defined as subclasses of animal cell. Without disjointness constraints between plant and animal cell, this situation was not detected when the statements were made. The term for "spore" was similarly a subclass of three disjoint classes: "prokaryotic cell," "plant cell," and "fungal cell" (Fig. 1) and "zygote" was a subclass of "animal cell" and "plant cell."

These disjointness issues, which were detected by Cyc when we attempted to add disjointness assertions to a translated version of the 2009 Cell Line Ontology, were corrected with the separation of plant cell types from the Cell Line Ontology in December 2011 [6].

3.2 Analysis

For a more complete analysis, we downloaded an updated (13/1/2012 09:59) version of the Cell Line Ontology. This version had obsoleted all plant cell types, referring the user to the Plant Ontology for such terms, and distinguished prokaryotic spores from fungal spores. The ontology defines 1928 non-obsoleted cell types, 29 binary predicates, and 32 (new) disjointness assertions among cell types.

A collection of terms from other ontologies (including PR for protein, UBERON - cross-species anatomy, NCBI - biological taxa, ChEBI - chemical entities, PATO-phenotypic qualities) are also included to be specified as arguments to relations restricting the cell type definitions. 4233 such assertions are included in the ontology.

To perform an analysis, the ontology was converted to CycL, loaded into OpenCyc [7], and then queries were asked using the OpenCyc interface.

Formal criteria Analysis of the logical constraints for the Cell Line Ontology showed that the cell types were arranged in a directed acyclic graph rooted on a term for "cell" and that there were no shared subclasses of any of the defined disjoint pairs

(Table 1, column 1). Cyc was not needed for such a determination – OWL reasoners can detect intersections of disjoint classes.

Table 1. Queries of Cell Line Ontology

Disjoint classes that have a common subclass	Cell types that develop from Eukaryotic cells, but are not known to be Eukaryotic	Eukaryotic cell types that develop from cell types not known to be Eukaryotic
(and (ist-Asserted[3] CL_Mt (disjointWith ?C1 ?C2)) (genls ?C0 ?C1) (genls ?C0 ?C2))	(and (allRelationExists ?C1 CL_developsFrom ?C2) (genls ?C2 EukayoticCell) (unknownSentence (genls ?C1 EukayoticCell)))	(and (allRelationExists ?C1 CL_developsFrom ?C2) (genls ?C1 EukayoticCell) (unknownSentence (genls ?C2 EukayoticCell)))
Answers: 0	Answers: 19	Answers: 22

Informal Criteria – Completeness. Nine of the 29 binary relations had argument restrictions defined, all of which were to the PATO Ontology's term for "Quality" (PATO:00000001). Five of these relations were defined as transitive, two of them having an identical domain and range defined, and the rest having neither. These relations were only used in expressing intersection with a property [See Table 2], and in all cases the classes were consistent with the argument restrictions. The lack of argument restrictions on most relations is a significant incompleteness.

One of the properties defined for many cell types is that they develop from other cell types. Logically, cells that develop from types of EukayoticCell[4] (or Prokaryotic_Cell or Animal_Cell) should themselves be types of EukayoticCell (or Prokaryotic_Cell or Animal_Cell). The inference engine finds 19 violations of this principle. Similarly, if a subtype of one of these general classes of cells is known to develop from another type, it is quite possible that the second type is also a subtype of the general class. The inference engine finds 22 cases in which the cell type from which a eukaryotic cell type develops is not known to be a eukaryotic cell type. Table 1 (columns 2 and 3) provides the queries asked of the inference engine, their English translations, and the number of answers.

[3] The CycL relation ist-Asserted relates a specified context to a specific statement made in it; the relation genls is the CycL subclass relation; allRelationExists means that for every instance of a class (first argument) the specified relation (second arg.) relates it to some instance of another class (third arg.); unknownSentence means that the statement that is its only argument is neither stated nor derivable through taxonomic reasoning. Variables in CycL are prefixed by a question mark ("?"). The relation ist-Asserted can not be expressed in FOL.

[4] The Cell Ontology uses IDs such as CL:0000003. For clarity, we use the phrase provided by the name field to specify each term.

Table 2. Germ line stem cell defined as intersection of germ-line cell and being capable of stem cell division (OWL format)

```
:CL_0000014 rdf:type owl:Class ;
            owl:equivalentClass
             [ rdf:type owl:Class ;
               owl:intersectionOf
               ( :CL_0000039
                 [ rdf:type owl:Restriction ;
                   owl:onProperty : capable_of ;
                   owl:someValuesFrom GO:0017145
                 ]
               )
             ]
```

Only 32 disjointness assertions are defined, all of which apply to types of white blood cells and blood progenitor cells. Cell types near the top of the hierarchy include cell types by number of nuclei (none, some, one, greater than one) and cell types by organism type (prokaryotic, eukaryotic, animal, and fungal — plant cells having been removed from the ontology), which strongly indicated missing partitions and disjointness assertions (Fig. 2).

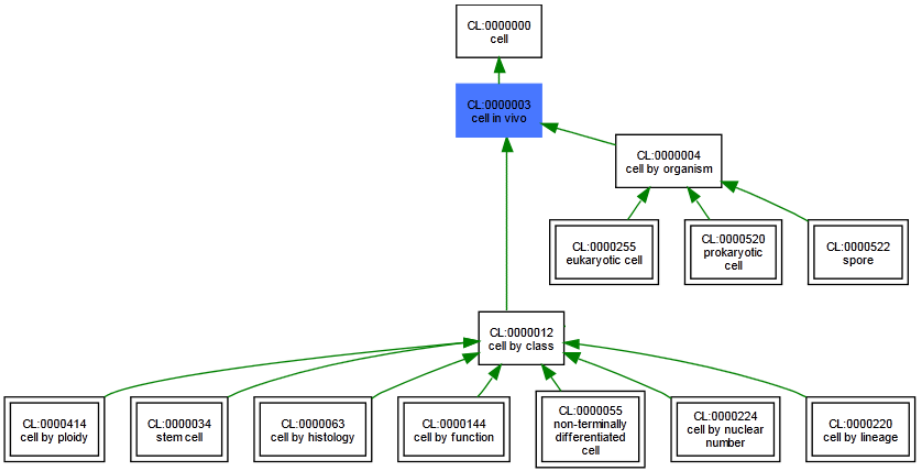

Fig. 2. Top Layers of Cell Ontology hierarchy, from http://proto.informatics.jax.org/prototypes/GOgraphEX/CL_Graphs/CL_0000003.svg

Informal Criteria – Abstraction Level. A brief analysis of the very top levels of the hierarchy showed that sets of classes were being treated as normal classes, with their members being labeled as subclasses. The initial partition (as described by textual descriptions) is `Cell_in_Vitro` vs. `Cell_in_Vivo` (renamed `Native_Cell`) with almost every other cell type a subclass of `Cell_in_Vivo`.

An ontologist might prefer to make this distinction orthogonal to other distinctions (since *in vitro* cells might reasonably be considered to be muscle/nerve/etc. cells, and

to have a nucleus or not even though they are not in a body). Cell_in_Vivo has the subclasses Cell_by_Organism and Cell_by_Class. Cell_by_Class has the subclasses Cell_by_Nuclear_Number, Cell_by_Ploidy, Cell_by_Lineage, Cell_by_Histology, Cell_by_Function, and Nonterminally_Differentiated_Cell. The textual descriptions of each of the "Cell_by_X" classes start with "a classification of cells by …," making it clear that each are intended to be sets of classes, i.e. metaclasses, since their descriptions are not generally applicable to the various subclasses of those cell types that are defined as their direct subclasses. The definitions of these classes are meaningless with respect to the individual cells that are supposed to be their instances [8].

Some of these sets of cell types seem to naturally be disjoint sets. Under Cell_by_Organism there is Prokaryotic_Cell and Eukaryotic_Cell and under Eukaryotic_Cell there is Animal_Cell, Fungal_Cell, and Mycetozoan_Cell (Plant_Cell has been obsoleted), all of which are cells distinguished by the type of organism of which they are a part. Since every organism is either a prokaryote or a eukaryote, the first division is a partition on Cell although it has not been so declared in the ontology. The three directly specified subclasses of Eukaryotic Cell are all disjoint, but this is not stated in the ontology; these subclasses do not cover all eukaryotic cells, so it is not a partition.

A similar analysis covers Cell_by_Nuclear_Number and Cell_by_Ploidy, each of which has instances (although defined as subclasses) that partition Cell. These instances each have very few subclasses even though most cell types fall under the definition of an instance of each of these metaclasses.

Table 3. Cell Line Ontology Property Issues

Cell types defined as the intersection of another cell type and having some property	Cell types defined as the intersection of a metatype and having some property	Cell types that have a property and are not stated as being a subclass of the intersection of a superclass with the property
(and (isIntersectionOf ?C ?C1 ?PRED ?V) (isa ?C1 CellType) (isa ?C CellType))	(and (isIntersectionOf ?CT ?MCT ?PRED ?VALUE) (genls ?MCT CellType) (genls ?CT Cell))	(and (isIntersectionOf ?C1 ?C0 ?PRED ?V) (genls ?C2 ?C0) (allRelationExists ?C2 ?PRED ?V) (unknownSentence (genls ?C2 ?C1)))
Answers: 547	Answers: 10	Answers: 10

Formal criteria – Internal Consistency. Over 4200 cell types in the ontology are defined as having some property (e.g., haploid, mononucleate, etc.). Over 500 cell types are defined as being the intersection of a more general cell type and having a specific property. Ten of the cell types which end up being instances of one of the metaclasses are also defined as being an intersection of a metaclass and having one of

these properties. However, in the Cell Line Ontology, the more specific cell types specified as having a property are not always (through the subclass hierarchy) declared to be subclasses of the class which is an intersection of that property and one of their superclasses. Although many reasoners (including OWL reasoners) can derive the subclass relationship, BioPortal's browser for the Cell Type Ontology at the time of the ontology release did not conclude them.

A query by the inference engine yielded ten such missing subclass relationships, which makes the ontology internally inconsistent for insufficiently powerful systems such as the BioPortal ontology browser of January 2011. Table 3 provides queries asked of the inference engine, their English translations, and the number of answers.

3.3 Resolution

We converted the metaclasses at the top of the ontology to actual metaclasses, converting the subclass assertions of their instances to instantiation assertions. This meant that those classes which had their `subclassOf` relationships removed needed to have new `subclassOf` relationships asserted if no others existed. Subclass assertions were made from the instances of these metaclasses to `Cell`, not to `Native_Cell`. `Prokaryotic_Cell` was made a subtype of `Anucleate_Cell` and `Nucleate_Cell` was made a subclass of `Eukaryotic_Cell`. Other subclass assertions are needed at this level; for example, a number of the (now) instances of `Cell_Type_by_Function` should be declared to be subclasses of `Animal_Cell` or `Eukaryotic_Cell`, but such work is the responsibility of a developer or subject matter expert.

Defined subclasses of these now direct instances of the metaclasses were examined to determine whether they should also be instances of the metaclass and were so asserted only if judged appropriate. For example, `Cell_By_Nuclear_Number` had `Mononucleate_Cell`, `Binucleate_Cell`, and `Multinucleate_Cell` added as instances while remaining as subclasses of `Nucleate_Cell`.

Other former direct subclasses were examined to determine whether they should be subclasses of direct instances of the metaclass, and not instances themselves. For example, `Cell_by_Nuclear_Number` had its instances restricted to `Anucleate_Cell`, `Nucleate_Cell`, `Mononucleate_Cell`, and `Multinucleate Cell`, with its other former direct subclasses (`Mononuclear_Osteoclast`, `Multinuclear_Osteoclast`, …) being asserted as subclasses of the appropriate direct instance as indicated by their comments.

Disjointness statements were made for the instances of the newly restructured metaclasses, `Cell_by_Organism`, `Cell_by_Nuclear_Number`, and `Cell_by_Ploidy`. `Cell_by_Organism` was made a subclass of `Cell_by_Class`.

We added rules to the CycL version of the ontology conclude subclass relationships:

- A rule was added so that cell types that are defined as developing from eukaryotic or animal cell types are concluded to also be subclasses of `Eukaryotic_Cell`

or Animal_Cell, respectively. This resulted in 26 subclass assertions being derived.
- A rule was added so that if one class is defined as an intersection of a class and a property, subclasses of that class that have that property are concluded to be subclasses of the intersection class. This resulted in a further ten subclass assertions being derived.
- A rule was added so that if one class is defined as an intersection of a metaclass and a property, other classes with that property are concluded to be subclasses of the direct instance of the metaclass. The intersection assertion was changed to being an intersection of Cell and the property. This resulted in nine subclass assertions being derived.

The Cell Ontology obsoleted the metaclasses in March 2012 [8]. A more recent OBO Library browser does conclude subclass relationships derived from intersection definitions.

Other detected problems still need to be resolved. Such work is not the responsibility of a validator, but of a developer or subject matter expert. We recommend that Cell Line Ontology developers:

- Define subclasses of Mononucleate_Cell and other instances of Cell_by Nuclear_Number so that every cell type that has a restricted nuclear number is defined as such by the subclass hierarchy.
- Define subclasses of Diploid_Cell and other instances of Cell_by_Ploidy so that every cell type that has a restricted ploidy is defined as such by the subclass hierarchy.
- Define those instances of Cell_by_Function which of necessity are subclasses of Animal_Cell or Eukaryotic_Cell as being so. For those instances which are not so restricted, check their direct subclasses to determine whether they should be subclasses of Animal_Cell or Eukaryotic_Cell.
- In cases in which a subclass of Eukaryotic_Cell (or Animal_Cell) is declared to develop from a cell type that is not such a subclass, the second class should be examined to determine whether it should be a subclass of Animal_ Cell or Eukaryotic_Cell.
- Add many more disjointness assertions among sibling classes, as appropriate.
- Define appropriate argument restrictions on the predicates in the ontology.

4 Plant Ontology

4.1 Introduction

The 2 April 2012 version of the Plant Ontology contains 1593 terms, 1181 of which are types of plant anatomical entity, 272 of which are types of plant structure developmental stage, eight of which are binary relations, and 132 of which are obsoleted. 37 disjointness assertions among cell types are included. The Plant Ontology includes

64 assertions specifying that one class is an intersection of another class with having a specific property.

The intersection assertions are accepted as a way of stating subclass relationships that are to be concluded instead of directly stated. This was done in order to avoid directly stating "dual parentage" in the ontology [5, p. 4].

4.2 Analysis

To analyze the ontology, it was converted to CycL, loaded into OpenCyc, and queries were asked using the OpenCyc interface.

Formal criteria – Logical constraints. Analysis of the logical constraints for the Plant Ontology showed that the classes were arranged in two directed acyclic graphs rooted on terms for "plant anatomical entity" and "plant structure development stage," and that there were no shared subclasses of any of the defined disjoint pairs. There was no violation of logical constraints.

Formal criteria – Internal Consistency. Over 800 classes in the ontology are defined as having some property. 64 of the classes are defined as being an intersection of a more general class and having one of these properties. By querying the inference engine, we found that in 63 cases, the more specific classes are not (directly or indirectly) defined as subclasses of the class-property intersection. Two examples of this are types of plant cell that have the property of being part of a plant embryo, but are not known to be subclasses of `EmbryonicPlantCell`. For systems with limited reasoning capabilities, these are violations of internal consistency.

Table 3 provides the queries asked of the inference engine, their English translations, and the number of answers.

Table 2. Plant Ontology Property Issues

Classes which are defined to have some property that are not defined to be subclasses of the intersection of a superclass with that property	Plant cell types that are part of plant embryos, but are not known to be embryonic plant cells

`(and`
` (isIntersectionOf`
`  ?P1 ?P0 ?PRED ?V)`
` (allRelationExists ?P2 ?PRED ?V)`
` (genls ?P2 ?P0)`
` (unknownSentence`
`  (genls ?P2 ?P1)))`
	`(and`
	`  (allRelationExists ?P1`
	`  PO_part_of PlantEmbryo)`
	`  (genls ?P1 PlantCell)`
	`  (unknownSentence`
	`  (genls ?P1`
	`  EmbryonicPlantCell)))`

`Answers: 63`	`Answers: 2`

Informal Criteria – Completeness. Disjointness assertions were missing from the developmental stage hierarchy and from near the top of the anatomical hierarchy. None of the binary relations had argument restrictions defined. Three of these relations were defined as transitive; none as symmetric or reflexive. Only 37 disjointness assertions are present, all of which are well down in the cell type hierarchy. There are

significant gaps in the ontology in both argument type restrictions and disjointness assertions.

Informal Criteria – Abstraction Level. Unlike the Cell Line Ontology, the Plant Ontology had no metaclasses near the top of the hierarchy that were used as subclasses.

4.3 Resolution

We added a rule to conclude subclass relationships:

- A rule was added so that if one class is defined as an intersection of a class and a property, subclasses of that class that have that property are concluded to be subclasses of the intersection class. This resulted in 63 assertions being derived.

A more recent Plant Ontology browser does conclude subclass relationships derived from intersection definitions. Much is still missing, e.g., disjointness assertions and argument type restrictions. Such work is not the responsibility of a validator, but of a developer or subject matter expert.

We recommend that Plant Ontology developers:

- Specify disjointness among sibling classes as appropriate.
- Define appropriate argument restrictions on the predicates in the ontology.
- Review comments which state that a class has a certain property, and encode those that are valid and are not already encoded or derivable from properties of superclasses.

5 Lessons Learned and Conclusion

We analyzed two ontologies that have strong user bases and communities for ensuring their validity. Significant problems were discovered with each ontology as a result of verification queries.

We note that public ontologies are not static. Early problems in the class hierarchy of the Cell Line Ontology, discovered when making high-level disjointness assertions (e.g., plant vs. animal cell) flagged common subclasses, were corrected before our in-depth analysis and the Plant Ontology was disconnected from the Cell Type Ontology in December of 2011. The in-depth analyses of the two ontologies discovered no remaining disjointness problems. Only a domain expert can determine whether this is due to the validity of the current subclass hierarchy or the sparseness of disjointness assertions.

One of the two ontologies erroneously treated metaclasses as normal subclasses of the root term. This led to numerous missing subclass assertions (evidently because the subclass does not fit the definition of the metaclass). These metaclasses have since been obsoleted. They could be reinstated as metaclasses if they are recognized as such.

The omission of argument restrictions can be readily determined. The lack of disjointness assertions among sibling classes can also be readily determined, but a subject matter expert should determine whether such sibling classes are actually disjoint.

Both ontologies had statements that instances of certain classes had certain properties, and that other classes were the intersection of superclasses with having some property. Such statements were initially not executable rules in the provided ontology viewer, so that in both cases subclass assertions that could be concluded based on these rules were missing. These examples emphasize that ontology evaluation needs to cover more than just whether the statements in an ontology lead to a logical contradiction.

When an ontology includes statements that could be mapped to rules from which subclass relationships or disjointness between classes can be concluded, an ontology evaluation step in a sufficiently rich semantic language can draw such conclusions and check if the conclusions are entailed by the encoded subclass and disjointness statements. If they are not already present, the concluded statements can then be added to the ontology.

The presence of metaclasses erroneously defined as normal classes in a subsumption hierarchy cannot be concluded from automatic analysis of the statements in an ontology. Such problems may be more likely to occur near the root of a subclass hierarchy and can be manually detected by reading the descriptions of the terms. Such situations can be resolved by determining which of the defined subclasses of the metaclass are normal classes and which are metaclasses, converting the normal classes to be instances instead of subclasses of the metaclass, and adding disjointness assertions, as appropriate, among them.

It is noteworthy that the problems found in these case studies consisted of systematic repetition of narrow categories of errors, rather than many different errors. If one were to evaluate the ontologies using a checklist of validity criteria or common errors, they would have gotten few black marks; yet a large proportion of their concepts was affected. If it can be shown that this pattern is typical, an ontology validation and correction strategy could be optimized accordingly.

Although a discipline of ontology validation and quality assurance is still evolving, our experiences so far have been positive and instructive. Potential future work includes the creation of an updated, comprehensive reference to ontology validity criteria, informed by a survey of previous case studies and the performance of additional new case studies.

Acknowledgement

This work was funded under NSF award #0934364 to the University of Maryland, Collaborative Research Establishing a Center for Hybrid Multicore Productivity Research..

References

1. Matuszek, C., Cabral, J., Witbrock, M., DeOliveira, J.: An Introduction to the Syntax and Content of Cyc. In Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, Stanford, CA, (2006).
2. OBO Download Matrix, http://www.berkeleybop.org/ontologies .
3. The OBO Flat File Format Specification, version 1.2, http://www.geneontology.org/GO.format.obo-1_2.shtml .
4. Cell Line Ontology version 1.0, May 2009, http://bioportal.bioontology.org/ontologies/39927/.
5. L. D. Cooper et al., "The Plant Ontology: Development of a Reference Ontology for all Plants," http://icbo.buffalo.edu/F2011/slides/AnatomyWorkshop/ FAOI2011.08.Cooper.pptx, 2011.
6. Mungall, C.: Cell Ontology 2012-12-13[STET – typo in year] release, http://sourceforge.net/mailarchive/message.php?msg_id=28544354 , December 15 2011.
7. OpenCyc, http://www.opencyc.org/, 2012.
8. Delete meaningless upper level terms, Cell-Ontology Issues Wiki, http://code.google.com/p/cell-ontology/issues/detail?id=1 .