mindswap
maryland information and network dynamics lab semantic web agents project
MINDSWAP

# A Survey of the Web Ontology Landscape

Taowei David Wang[1]
Bijan Parsia[2]
James Hendler[1]
[1]MINDSWAP, University of Maryland at College Park,
[2]University of Manchester

ISWC 2006

mindswap
maryland information and network dynamics lab semantic web agents project
MINDSWAP

## Motivation

- Two pieces of information are imperative for good tool design
  - Users and their tasks
  - The characteristic of the data to be manipulated
- Many Semantic Web tools for dealing with ontologies are created without careful analysis of these variables
- Here we surveyed 1300 OWL ontologies and RDFS files to offer tool designers what ontologies in the wild look like.

# Outline

- Ontology Collection
- Statistics Collection
  - Tools used
  - Statistics collected
- Analyses
  - OWL species with respect to DL expressivity
  - Tractable fragments of OWL
  - OWL construct usage
  - OWL class hierarchy analyses
- Final words



# Ontology Collection

- Collected over 4000 documents from Swoogle 2005[1] using "sort:ontology"
- Collected 218 OWL ontologies from Google using "owl ext:owl"
  - Much has changed in ways how Google indexes .owl files, now the number is orders of magnitudes bigger
- Manually added ontologies from well-known repositories
  - Protégé OWL Library[2]
  - DAML Ontology Library[3]
  - Open Biological Ontologies Repository[4]
  - SchemaWeb[5]

# Collection Clean Up

- We first pruned off the duplicate URIs
- Threw away unsuitable data:
  - DAML files from Swoogle
  - Test files for OWL from W3C, Jena
    - Syntactically correct, but are only used to verify tools or show use cases.
    - All versions from the SVN
  - Pruned away around 1000 WordNet RDFS files
    - Useful as a whole, some meanings are dropped when viewing specific fragments
- After cleaning and pruning, we had roughly 700 OWL ontologies, and 600 RDFS files.

mindswap
maryland information and network dynamics lab semantic web agents project
MINDSWAP

# Statistics Collection

- We used Swoop[6] to gather statistics about an ontology and the class graph structure.
- We used Pellet[7] to check consistency, classify, and perform species validation.
- We used Jena[8] to collect statistics regarding the OWL construct usage.

mindswap
maryland information and network dynamics lab semantic web agents project

# OWL Species vs Expressivity

| **Species** | RDFS | Lite | DL | Full | Error |
|---|---|---|---|---|---|
| **Count** | 587 | 199 | 149 | 337 | 3 |

- We split RDFS and OWL files by presence of the OWL namespace, then performed species validation on OWL files
- Notice the large number of OWL Full files
  - Are they really beyond OWL DL?

mindswap
maryland information and network dynamics lab semantic web agents project

# OWL Fullness

- Bechhofer and Volz (2004)[9] categorized OWL Full documents:
  - Syntactically OWL Full
    - Missing type triples
    - Structural sharing
  - Redefinition of Known Vocabulary
  - Mixing *Classes*, *Properties*, and *Individuals*
  - Beyond OWL DL
- They also showed that many are of the "Missing Type Triples" category, and can be syntactically patched.
- Here we apply the same technique

## Patching OWL Full

| Species | RDFS(DL) | Lite | DL | Full | Error |
|---|---|---|---|---|---|
| **Count** | 556 | 391 | 264 | 61 | 3 |

- Only 61 Full files left: 30 OWL, 31 RDFS files
- Of the patched OWL Full files
  - 2/3 became OWL Lite
  - 1/3 became OWL DL
- Now the majority are OWL Lite (let's investigate!)



## DL Expressivity Binning

- We binned the files by their expressivity
  - Bin 4: contains nominals (*O*) or number restrictions (*N*), e.g. *SHOIN*
  - Bin 3: contains inverse (*I*) or complements (*C*), e.g. *SHIF*
  - Bin 2: contains role hierarchies (*H*) or functional properties (*F*), e.g. *ALHF*
  - Bin 1: The rest, e.g. *AL*

mindswap
maryland information and network dynamics lab semantic web agents project
MINDSWAP

# Expressivity Distribution

| Bin | Bin 1 | Bin 2 (*H*,*F*) | Bin 3 (*I*,*C*) | Bin 4 (*O*,*N*) |
|---|---|---|---|---|
| **Count** | 793 | 55 | 262 | 151 |

- Number of OWL Lite files: 391 (after patching)
- By (Bin 1 + Bin 2 – RDFS)
  - Number of OWL Lite files that do not use *I* or *C*: 261
  - 67% of OWL Lite documents use very little above RDFS
- Possible explanations
  - OWL Lite syntax keeps modelers away from *SHIF*.
  - RDFS modelers want to use a little bit of OWL
- There seems to be a subset of OWL Lite that is very widely used.

mindswap
maryland information and network dynamics lab semantic web agents project
MINDSWAP

# Tractable Fragments of OWL[10]

| RDFS(DL) | DL-Lite | EL++ | Not Tractable |
|---|---|---|---|
| 230 | 94 | 56 | 287 |

| Constructs | *DL-Lite* | $\mathcal{EL}++$ |
|---|---|---|
| owl:Intersection | 1(1%) | 3(5%) |
| owl:Restriction | 35 (37%) | 36 (64%) |
| owl:ObjectProperty | 45 (48%) | 43(77%) |
| owl:DatatypeProperty | 44 (0.47%) | 4 (7% |
| owl:FunctionalProperty | 20 (20%) | 0 (0%) |
| owl:Cardinality | 21 (22%) | 0 (0%) |
| owl:SomeValuesFrom | 0(0%) | 33(60%) |

mindswap
maryland information and network dynamics lab semantic web agents project

# OWL Construct Usage

Looking only at the OWL files now:

Table 5: OWL Construct Usage

| Construct | Count |
|---|---|
| Ontology-Related Constructs | |
| owl:Ontology | 567 |
| owl:OntologyProperty | 0 |
| owl:BackwardCompatibleWith | 0 |
| owl:Imports | 221 |
| owl:InCompatibleWith: | 1 |
| owl:PriorVersion | 8 |
| owl:VersionInfo | 305 |
| Class-Related Constructs | |
| owl:Class | 580 |
| owl:ComplementOf | 21 |
| owl:DataRange | 14 |
| owl:DeprecatedClass | 2 |
| owl:DisjointWith | 97 |
| owl:EquivalentClass | 77 |
| owl:IntersectionOf | 69 |
| owl:OneOf | 43 |
| owl:Union | 109 |
| Property-Related Constructs | |
| owl:AnnotationProperty | 28 |
| owl:DatatypeProperty | 277 |
| owl:DeprecatedProperty | 2 |
| owl:EquivalentProperty | 25 |
| owl:FunctionalProperty | 114 |
| owl:InverseFunctionalProperty | 30 |
| owl:InverseOf | 128 |
| owl:ObjectProperty | 462 |
| owl:SymmetricProperty | 20 |
| owl:TransitiveProperty | 39 |
| Individual-Related Constructs | |
| owl:AllDifferentFrom | 6 |
| owl:DifferentFrom | 5 |
| owl:DistinctMembers | 6 |
| owl:SameAs | 18 |
| Restriction-Related Constructs | |
| owl:AllValuesFrom | 118 |
| owl:Cardinality | 120 |
| owl:hasValue | 48 |
| owl:MaxCardinality | 60 |
| owl:MinCardinality | 99 |
| owl:onProperty | 263 |
| owl:Restriction | 263 |
| owl:SomeValuesFrom | 85 |

mindswap
maryland information and network dynamics lab semantic web agents project

# OWL Construct Usage

- As expected, ObjectProperty used in more ontologies than DatatypeProperty

| | |
|---|---|
| ObjectProperty | 462 |
| DatatypeProperty | 277 |

- Modelers may want to use InverseFunctional(30), Symmetric(20), Transitive(39), InverseOf(128), which, in OWL DL, are only available for ObjectProperties.

mindswap
maryland information and network dynamics lab semantic web agents project

# OWL Construct Usage

| | |
|---|---|
| owl:Union | 109 |
| owl:Intersection | 69 |

- Union appears in more ontologies than Intersection.
  - In OWL, you can get intersection by subclassing. So modelers can often get around not using the intersection construct to achieve the same meaning.
  - Protégé assumes the union semantic for range/domains, and will use owl:union by default when modelers say “R has range C1” and “R has range C2.

mindswap
maryland information and network dynamics lab semantic web agents project

# OWL Construct Usage

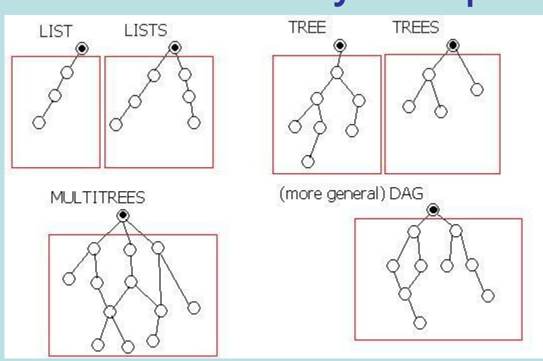
- Of the 688 OWL ontologies, 221 used owl:Imports.
  - Don’t know the distribution of imports, however.
- 253 OWL ontologies define instances
  - But very few ontologies use instance constructs
    - AllDifferentFrom(6), DistinctMembers(6), DifferentFrom(5), SameAs(18)

# Motivation for Hierarchy Analysis

- Lots of tree visualizers are used to visualize class hierarchies, including tree widgets
- Are they appropriate? Can we do better?
  - To what complex graph form can OWL class hierarchies take?
  - How do the told and inferred structures of the hierarchy impact the visualization?
  - How does having multiple inheritance impact the visualization? Do they occur often?

mindswap
maryland information and network dynamics lab semantic web agents project
MINDSWAP

# Class Hierarchy Morphology

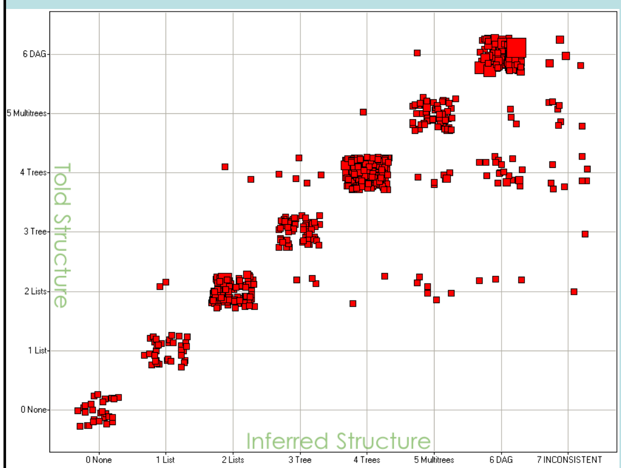



- Ignoring owl:Thing as the root, OWL ontologies can have these structures.
- How do the structures change from told to inferred? Do they change often?

mindswap
maryland information and network dynamics lab semantic web agents project

MINDSWAP

# OWL Class Graph Morphology



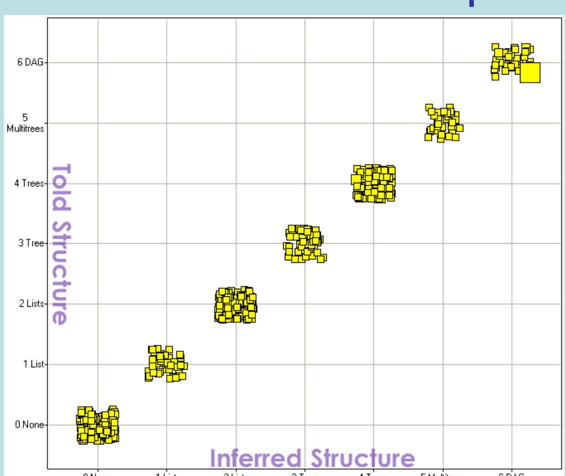

| Complexity Change | |
|---|---|
| Up | 42 |
| Down | 12 |

- 34 ontologies had no multiple inheritance in told structure, but has at least one in inferred structure
- 21 inconsistent

mindswap
maryland information and network dynamics lab semantic web agents project

MINDSWAP

# RDFS Class Graph Morphology



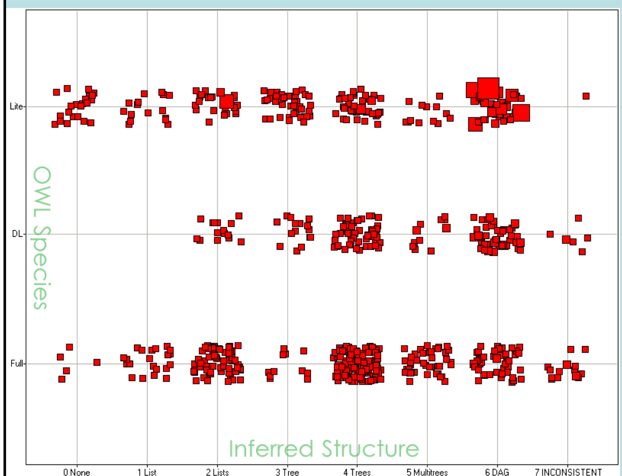

- Contrast this with the OWL version
- No cycles in RDFS

mindswap

maryland information and network dynamics lab semantic web agents project

MINDSWAP

# Large Ontologies




- Many large OWL Lite files are DAGs
- 19 ontologies with 2000+ classes
  - 14 have *ALC*, 2 *S*, 2 *SHIF*, 1 *SHOIF*
- 6 ontologies with 10000+ classes, 5 belong to (DAG, Lite)
  - 4 *ALC*, 1 *S*
- Complex class structures, but no OWL DL
- OBO

mindswap

maryland information and network dynamics lab semantic web agents project

MINDSWAP

# Summary

- Most OWL Full files can be patched
  - Tool support to explicitly add type triples?
  - No real need for OWL Full tool support
- Lots of ‘light weight’ OWL ontologies out there
  - People are using tractable fragments
  - Language standardization effort should take this into account (e.g. OWL 1.1)
  - Choosing the right reasoners for the right jobs
- Class morphology can change wildly
  - Changes between told/inferred structures are telling
  - To show topology differences should be a visualization requirement.

mindswap
maryland information and network dynamics lab semantic web agents project

MINDSWAP

# Conclusions and Discussions

- Do we need to do future surveys of this type?
  - There can be shifts in how people use ontologies
  - State of Semantic Web tools may improve and mature to a point so finer analyses are required
- Future work:
  - Wider scope, other analyses: import structure, partitioning, instances outside of ontologies (foaf)
  - The other half of the equation
    - Investigate what users do with ontologies

mindswap
maryland information and network dynamics lab semantic web agents project

MINDSWAP

# References

1. Swoogle 2005: http://swoogle.umbc.edu/2005/
2. Protégé Ontology Library: http://protégé.stanford.edu/plugins/owl/owl-library/
3. DAML Library: http://www.daml.org/ontologies/
4. Open Biological Ontologies Repository: http://obo.sourceforge/net/main.html
5. Schemaweb: http://www.schemaweb.info/
6. Swoop: K. Kalyanpur, B. Parsia, J. Hendler. A tool for working with web ontologies. International Journal on Semantic Web and Information Systems. 1(1), 2004.
7. Pellet: http://www.mindswap.org/2003/pellet/
8. Jena: J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: Implementing the semantic web recommendations. *Proceedings ISWC 2004.*
9. S. Bechhofer and R. Volz. Patching syntax in owl ontologies. *Proceedings ISWC 2004.*
10. *OWL: 1.1 Web Ontology Language: Tractable Fragments (http://owl1_1.cs.manchester.ac.uk/tractable.html)*

# Thank You

Special thanks to **Aditya Kalyampur** and **Evren Sirin** for their valuable inputs and discussions

**Result download**: http://www.mindswap.org/~tw7/work/survey/results/

mindswap
maryland information and network dynamics lab semantic web agents project
MINDSWAP

# Backup Slides

- Buggy OWL ontologies
- Pruning of ontologies

## Some Buggy OWL Ontologies

- 21 ontologies are inconsistent
  - 18 due to missing type on literal values
  - 3 contain logical contradictions
- 17 consistent ontologies contain at least one unsatisfiable classes.
  - 12 belong to bin 4
  - 5 belong to bin 3



## Details of Unpatchable Documents

- OWLfiles
  - structure sharing(1)
  - metamodeling(8)
  - beyond-DL(2)
    - Inverse on DatatypeProperty, transitivity on functional
  - Redefining existing vocabulary
    - e.g. subproperty of rdfs:label
- RDFS
  - Redefining existing vocabulary
    - e.g. Subclassing xsd:string