

A T-Box Generator for testing scalability of OWL mereotopological patterns

Martin Boeker¹, Janna Hastings², Daniel Schober¹, and Stefan Schulz³

martin.boeker@uniklinik-freiburg.de

¹ Institute of Medical Biometry and Medical Informatics,
University Medical Center Freiburg, Germany

² Chemoinformatics and Metabolism, EBI, Hinxton, UK

³ Department of Medical Informatics, Medical University Graz, Austria



OWLED
San Francisco
June 6th 2011

Background

- @neurlST** Integrated Biomedical Informatics for the Management of Cerebral Aneurysms (European Community - 6th FP)
- DebugIT** Detecting and Eliminating Bacteria Using Information Technology (European Community - 7th FP)
- BioTop** A Top-Domain Ontology for the Life Sciences
- GoodOD** Good Ontology Design (DFG grant JA 1904/2-1, SCHU 2515/1-1)

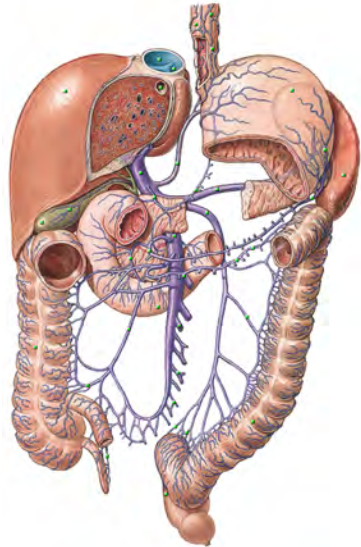
<http://www.cistib.upf.edu/aneurist1/>

<http://www.debugit.eu/>

<http://www.imbi.uni-freiburg.de/ontology/biotop/>

Taxonomy

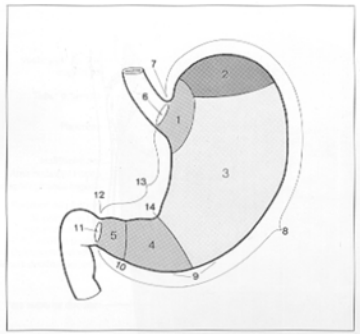
- *A* subClassOf *B*
- *Stomach* subClassOf *CavitatedOrgan*



Mutual disjointness

$\text{DisjointClasses } (C_1, C_2, \dots, C_n) =_{\text{def}} \{ C_1 \text{ subclassOf not } C_2; \dots; C_1 \text{ subclassOf not } C_n; C_2 \text{ subclassOf not } C_n; \dots \}$

- mutual disjointness: there are no entities both member of class C_1 and C_2 (... C_n)



Rauber, Anatomie des Menschen

locusOf-hasLocus implementation in BioTop

The screenshot displays the BioTop ontology editor interface. On the left, a hierarchical tree view shows the structure of the ontology, with 'topObjectProperty' expanded to show various relations. The 'locusOf' relation is highlighted. On the right, the 'definition' tab is active, showing the definition of 'locusOf' as a very broad relation. Below the definition, the 'Class hierarchy' tab is active, showing the hierarchy of classes. The 'locusOf' relation is shown as a 'disposition' that can be an 'immaterial object', 'information object', 'material object', 'process', 'quality', or 'role'. The 'hasLocus' relation is also shown as a 'disposition' that can be an 'immaterial object' or 'material object'.

topObjectProperty

- DEPRECATED_RELATION
- 'abstractly related to'
- 'causally related to'
- 'has participant'
- 'participates in'
- 'spatially related to'
 - 'has locus'
 - 'inherits in'
 - 'physically located in'
 - 'physical part of'
 - 'physically contained in'
 - 'has origin'
 - 'locus of'
 - 'bearer of'
 - 'physical location of'
 - 'has physical part'
 - 'has proper physical part'
 - 'has component part'
 - 'has granular part'
 - 'has originating physical part'
 - 'physically contains'
 - 'originates from'
 - 'physically bounds'
 - 'physically bounded by'
 - 'physically connected to'
 - 'physically disconnected from'
 - 'temporally related to'

definition

"locusOf is a very broad relation which relates the place at which an entity occurs, inheres, or is part of"

Class hierarchy

Disposition: 'locusOf'

(Disposition: 'locusOf')

- 'immaterial object'
- or 'material object'

disposition

- or 'immaterial object'
- or 'information object'
- or 'material object'
- or process
- or quality
- or role

Disposition: 'hasLocus'

- 'immaterial object'
- or 'material object'

<http://www.imbi.uni-freiburg.de/ontology/biotop/>

Spatial disjointness

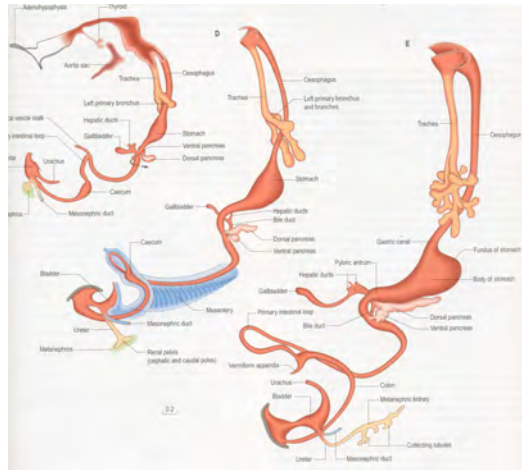
C_1 subClassOf *locusOf* only (not (*hasLocus* some C_2))
 C_2 subClassOf *locusOf* only (not (*hasLocus* some C_1))

- spatial disjointness: “the location of an entity can only be where no spatial disjoint entity is located”.
 - why do we need that: mutual disjoint entities can be overlapping or in parthood relation



Partonomy

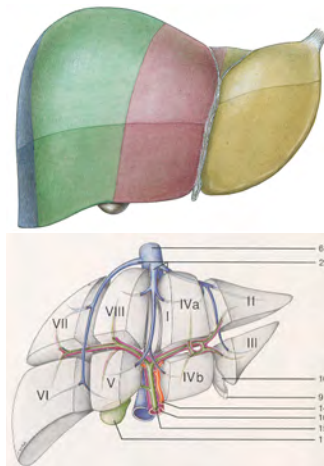
- Part implies whole
 - *Stomach* partOf *SOME GastrointestinalSystem*
- Whole implies part
 - *GastrointestinalSystem* hasPart *SOME Stomach*



Gray's Anatomy, Editor: Susan Standring, 39th Edition

Exact cardinality

- Exact cardinalities in in biostructures: 1, 2, 5, n
 - Hand* hasPart exactly 5 *Finger*
 - Hand* hasPart exactly 1 *Thumb*
 - Liver* hasPart exactly 8 *LiverSegment*



Sobotta, Atlas der Anatomie des Menschen

Rohen, Anatomie des Menschen. Fotografischer Atlas der systematischen und topografischen Anatomie.

Properties of the T-Box Generator

- Configuration of the
 - Number of hierarchical levels
 - Number of siblings on each level
 - Number of mutual disjoint classes on each level
 - Number of spatial disjoint classes on each level
 - Definition of partonomic restrictions as subclass or equivalent class expressions
 - Quantification of partonomic relation as existential or exact cardinal
 - Type of *locusOf-hasLocus*, *hasPart-partOf*: inverse and/ or transitive

Implementation in Scala

- Scala version 2.8.2
 - object functional language for the JVM
- ca. 170 loc
- OWL API version 3.2.1 (Feb 4 2011)

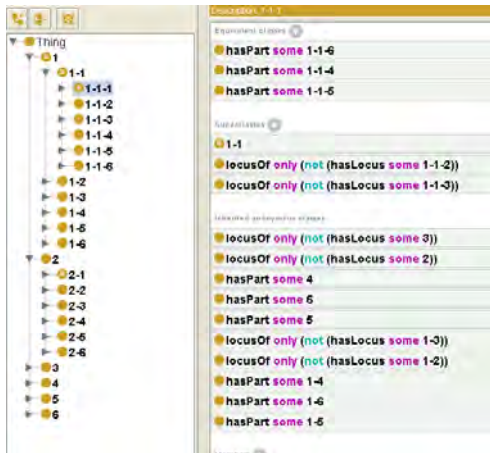
```
98  /**
99   * process = List[OWLCClass] depending on @cl (nested class level)
100   * call generateClassList for every parent class in List[OWLCClass] recursively,
101   */
102  def processClassList(cl: Int, classList: List[OWLCClass]): Unit = {
103    // do some restrictions
104    setDisjoints(classList)
105
106    // do some hierarchy stuff
107    if (cl == 1) {
108      for (cls <- classList)
109        manager.addAxiom(ontology, factory.getOWLDeclarationAxiom(cls))
110    }
111    if (cl < Depth) {
112      for (cls <- classList) {
113        val pref = cls.toStringID.split("#")(1) + "-"
114        val nextLevel = generateClassList(2, pref,
115                                          List(factory.getOWLCClass(IRI.create(
116                                            ontology.getIRI + "#" + pref + 1))))
117        for (ncls <- nextLevel) {
118          manager.addAxiom(ontology, factory.getOWLSubClassOfAxiom(ncls, cls))
119        }
120        //recurse!!!
121        processClassList(cl+1, nextLevel)
122      }
123    }
124  }
```

<http://www.scala-lang.org/>

<http://owlapi.sourceforge.net>

Download at: <http://www.imbi.uni-freiburg.de/ontology/t-box-generator.zip>

Example ontology: 6^5 classes, partonomy as equivalent classes, existential quantification



Estimating the performance of classification

- three groups of seven ontologies
 - each ca. 8000 classes: $6^5, 20^3, 90^2$ (*siblings*^{hierarchical levels})
- “increasing complexity”
 - hierarchy only
 - add 50% siblings mutually disjoint and 50% siblings mutually spatially disjoint
 - add partonomical restrictions to mutually disjoint as subclass expressions with existential quantification
 - change restrictions to equivalent class expression with existential quantification
 - set properties of *locusOf-hasLocus*, *hasPart-partOf* relations to invers and transitive
 - change restrictions to exact cardinal quantification (1, no. of disjoints/ 2)

Experimental setting

- 1.6GHz Intel(R) Core(TM) i7 Q720, 4GB RAM, Windows 7 64-Bit, Java 1.6.25 64-Bit
- -Xmx3000m: effective memory allocation of 2796MB
- Three DL Reasoners
 - Fact++ (1.5.2), HermIT (1.3.3), Pellet (2.2.2)
- Three sequential measurements with Protegé (4.1 RC2 build 228)
 - unload Protege after each sequence

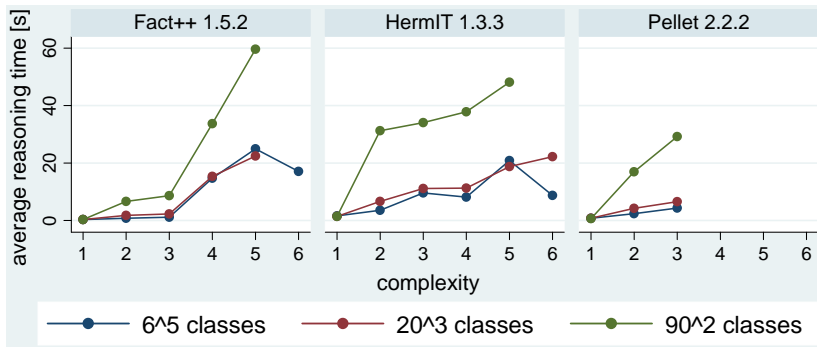
Results: Fact++ 1.5.2

disjoints		partonomy		subclasses or equivalent classes		object relations invers/ transitive		quantification		complexity level		6 ⁵ =7776 [3; 2]		20 ³ =8000 [10; 5]		90 ² =8100 [45; 22]	
0	no									1	0.3	0.3	0.3				
n										2	0.7	1.6	6.7				
	yes	sub	yes	existential	3	1.1	2.0	8.4									
			no		4	14.5	15.4	33.5									
		eq	yes		5	24.9	22.3	59.3									
					cardinality 1	6	17.1	to	to								
					cardinality m	7	to	to	oom								

Results: HermIT 1.3.3

disjoints		partonomy		subclasses or equivalent classes		object relations invers/ transitive		quantification		complexity level		6 ⁵ = 7776 [3; 2]		20 ³ = 8000 [10; 5]		90 ² = 8100 [45; 22]	
0	no							1	0.7	0.5	0.8						
n														2	1.8	3.8	26.1
	yes	sub	yes	existential	3	7.6	7.6							30.4			
			no		4	6.0	8.5							29.8			
		eq			yes	5	18.3							15.9	42.5		
			cardinality 1	6		5.8	18.0							387.4			
			cardinality m	7		276.4	oom							oom			

Results: comparison of average reasoning time



graphs produced with Stata 11.2

Limitations and further development

- introduce other object relations (pairwise inverse)
- relate to child classes of siblings
- introduce other patterns

Summary

- patterns of biostructure
 - taxonomy
 - mutual disjointness
 - spatial disjointness
 - partonomy
 - exact cardinality
- T-Box generator
 - configurable to produce ontologies with patterns
 - estimation of reasoner performance
 - performance of Fact++ and HermIT equal on the tested ontologies