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

Martin Boeker\textsuperscript{1}, Janna Hastings\textsuperscript{2}, Daniel Schober\textsuperscript{1}, and Stefan Schulz\textsuperscript{3}

\textit{martin.boeker@uniklinik-freiburg.de}

\textsuperscript{1} Institute of Medical Biometry and Medical Informatics, University Medical Center Freiburg, Germany
\textsuperscript{2} Chemoinformatics and Metabolism, EBI, Hinxton, UK
\textsuperscript{3} Department of Medical Informatics, Medical University Graz, Austria

OWLED
San Francisco
June 6\textsuperscript{th} 2011
Background

@neurIST 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*

Sobotta, Atlas der Anatomie des Menschen

Institute of Med. Biometry and Med. Informatics

A T-Box Generator testing scalability of mereotopological patterns
Mutual disjointness

\[
\text{DisjointClasses } (C_1, C_2, \ldots, C_n) = \text{def } \{ C_1 \text{ subclassOf not } C_2; \\
\ldots; C_1 \text{ subclassOf not } C_n; C_2 \text{ subclassOf not } C_n; \ldots \}
\]

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

Rauber, Anatomie des Menschen
representation of biomedical structure
A T-Box Generator

locusOf-hasLocus implementation in BioTop

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

Institute of Med. Biometry and Med. Informatics
A T-Box Generator testing scalability of mereotopological patterns
Spatial disjointness

\[ C_1 \text{ subClassOf } \text{locusOf} \text{ only } (\text{not } (\text{hasLocus some } C_2)) \]

\[ C_2 \text{ subClassOf } \text{locusOf} \text{ only } (\text{not } (\text{hasLocus some } C_1)) \]

- 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

Sobotta, Atlas der Anatomie des Menschen
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 biostructures: 1, 2, 5, n
  - Hand hasPart exactly 5 Finger
  - Hand hasPart exactly 1 Thumb
  - Liver hasPart exactly 8 LiverSegment

Sobotta, Atlas der Anatomi des Menschen
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)

```scala
// process a List[OWLClass] depending on cls (nested class level).
// call generateClassList for every parent class in List[OWLClass] recursively.
/
/def processClassList(cl: Int, classList: List[OWLClass]): Unit = {
// do some restrictions
setDisjoint(classList).

if (cl == 1) {
  for (cls <- classList) {
    manager.addAxion(ontology, factory.getOWLDeclarationAxiom(cls))
  }

  if (cl < depth) {
    for (cls <- classList) {
      val pref = cls.toStringID.split("/")(1) + "="
      val nextLevel = generateClassList(2, pref,
        List(factory.getOWLClass(IRI.create (ontologyIRI + "#" + pref + 1))))
      for (nccls <- nextLevel) {
        manager.addAxion(ontology, factory.getOWLSubClassOfAxiom(nccls, cls))
      }
    }
  }

  processClassList(cl+1, nextLevel)
}
```
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: $6^5, 20^3, 90^2 (\text{siblings}^{\text{hierarchical levels}})$
- “increasing complexity”
  - hierarchy only
  - add 50% siblings mutually disjoint and 50% siblings mutually spatially disjoint
  - add partonomical restrictions to mutually disjoints as subclass expressions with existential quantification
  - change restrictions to equivalent class expression with existential quantification
  - set properties of $\text{locusOf}-\text{hasLocus}, \text{hasPart}-\text{partOf}$ relations to invers and transitive
  - change restrictions to exact cardinal quantification ($1, \text{no. of disjoints/ 2}$)
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

<table>
<thead>
<tr>
<th>disjoints</th>
<th>partonomy</th>
<th>subclasses or equivalent classes</th>
<th>object relations</th>
<th>invers / transitiv</th>
<th>quantification</th>
<th>complexity level</th>
<th>$6^5 = 7776$ [3; 2]</th>
<th>$20^3 = 8000$ [10; 5]</th>
<th>$90^2 = 8100$ [45; 22]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>2</td>
<td>0.7</td>
<td>1.6</td>
<td>6.7</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>1.1</td>
<td>2.0</td>
<td>8.4</td>
</tr>
<tr>
<td>4</td>
<td>14.5</td>
<td>15.4</td>
<td>33.5</td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eq</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>24.9</td>
<td>22.3</td>
<td>59.3</td>
</tr>
<tr>
<td>6</td>
<td>17.1</td>
<td>to</td>
<td>to</td>
<td></td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cardinality 1</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cardinality m</td>
<td>7</td>
<td>to</td>
<td>to</td>
<td></td>
<td></td>
<td>7</td>
<td></td>
<td></td>
<td>oom</td>
</tr>
</tbody>
</table>
## Results: HermIT 1.3.3

<table>
<thead>
<tr>
<th>disjoints</th>
<th>partonomy</th>
<th>subclasses or equivalent classes</th>
<th>object relations</th>
<th>quantification</th>
<th>complexity level</th>
<th>6^5=7776</th>
<th>20^3=8000</th>
<th>90^2=8100</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>0.7</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>1.8</td>
<td>3.8</td>
<td>26.1</td>
</tr>
<tr>
<td>n</td>
<td></td>
<td>sub</td>
<td>yes</td>
<td></td>
<td>3</td>
<td>7.6</td>
<td>7.6</td>
<td>30.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>no</td>
<td></td>
<td>existential</td>
<td>4</td>
<td>6.0</td>
<td>8.5</td>
<td>29.8</td>
</tr>
<tr>
<td></td>
<td>eq</td>
<td>yes</td>
<td></td>
<td></td>
<td>5</td>
<td>18.3</td>
<td>15.9</td>
<td>42.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cardinality 1</td>
<td>yes</td>
<td></td>
<td>6</td>
<td>5.8</td>
<td>18.0</td>
<td>387.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cardinality m</td>
<td></td>
<td></td>
<td>7</td>
<td>276.4</td>
<td>oom</td>
<td>oom</td>
</tr>
</tbody>
</table>

Institute of Med. Biometry and Med. Informatics
A T-Box Generator testing scalability of mereotopological patterns
# Results: comparison of average reasoning time

<table>
<thead>
<tr>
<th>Reasoner</th>
<th>Complexity</th>
<th>Average Reasoning Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fact++ 1.5.2</td>
<td>$6^5$ classes</td>
<td><img src="graph1.png" alt="Graph" /></td>
</tr>
<tr>
<td>HermIT 1.3.3</td>
<td>$20^3$ classes</td>
<td><img src="graph2.png" alt="Graph" /></td>
</tr>
<tr>
<td>Pellet 2.2.2</td>
<td>$90^2$ classes</td>
<td><img src="graph3.png" alt="Graph" /></td>
</tr>
</tbody>
</table>

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