Ontology based semantics and graphical notation as directed graphs

Johann Höchtl, Danube University Krems, Austria
Member of Research Staff, OASIS SET TC Member

johann.hoechtl@donau-uni.ac.at
www.donau-uni.ac.at/egov
Overview

- The Business Problem with Ontology, Semantics and Interoperability
- An Information Preparation Methodology
- Visualization of Ontologies
  - Tool Overview
- Ontology Similarity
  - Selected Methodologies and Tools
The Business Problem of (Semantic) Interoperability Application

- The leaders do not understand what it is all about
- Sometimes even not your customers
  - Plethora of definitions, common understanding but definition varies from organization to organization
  - IEEE: “Interoperability is the ability of two or more systems to exchange information and to use the information that has been exchanged”
  - ISO, ETSI, EICTA, …
Quotes from a study

“Interoperability is not yet a well-defined subject that can easily be applied to project work. It remains an innovative process in which conceptual issues have to be clarified on the job”

“At all phases of the project awareness of the results and benefits to potential users have to be widely publicised to ensure comprehensive marketing to the client base and maximum user take up.”

“differences between stakeholder groups or countries of different sizes or between different levels of government do exist [...] could cause problems of mutual understanding”

Ever heard of *Information Maps*?

The method to have your say in a concise manner, even your customer (boss) will understand you

- Obviously you still need the chance to be heard

- A research-based approach for creating structured documents and communications that are clear, concise, and user-focused

  - 200 common “block types”
  
  - Information Mapping produces measurable results, changing the way people write and work
Get your message through

- Key element: Visualization, more precise visual preparation of information
- Home of Robert Horn:
  - Probably wont get a design price
  - More examples:
  - NOT NOW!
Visualization is Key

Visualization is Key

- Visual representation of information and knowledge is nothing new
- Directed graphs are a natural way to represent knowledge
- All OWLs/RDFs/... can be transformed to Directed Acyclic Graphs
  - Want a formal prove? http://www.w3.org/2007/OWL/wiki/RDF-Based_Semantics
- Actually every DAG is_a DG, tautologies do not add information so are omitted → DAG
  - An Ontology is_a Ontology is_a Ontology is_a ...
Ontology Visualization

- **Tools Overview**
  - Licensed, Closed Source:
    - SemanticWorks (Altova)
    - TopBraid Composer (TopQuadrant)
    - Integrated Ontology Development Toolkit (IODT) (IBM)
    - Integrated Ontology Development Environment (IODE) (Ontology Works)
Ontology Visualization

- Free, Open Source Tools Overview
  - IsaViz (W3C)
    http://www.w3.org/2001/11/IsaViz/
  - Protégé (Stanford University)
    http://protege.stanford.edu/
    - Various Plugins for Visualization: Jambalaya, OWLViz, TGViz
  - IsaViz and various Protégé plugins (but not all!) use Graphviz as the underlying rendering engine
- Graphviz (AT&T)
  http://www.graphviz.org/
  - Based by AT&T research, released under a liberal “Common Public License” with sources
  - Domain free – no affiliation to semantic knowledge representation
  - Uses proprietary scripting language – DOT Language – to render directed graphs
digraph finite_state_machine {
  rankdir=LR;
  size="8,5"
  node [shape = doublecircle]; LR_0
  LR_3 LR_4 LR_8;
  node [shape = circle];
  LR_0 -> LR_2 [ label = "SS(B)" ];
  LR_0 -> LR_1 [ label = "SS(S)" ];
  LR_1 -> LR_3 [ label = "S($end)" ];
  LR_2 -> LR_6 [ label = "SS(b)" ];
  LR_2 -> LR_5 [ label = "SS(a)" ];
  LR_2 -> LR_4 [ label = "S(A)" ];
  LR_5 -> LR_7 [ label = "S(b)" ];
  LR_5 -> LR_5 [ label = "S(a)" ];
  LR_6 -> LR_6 [ label = "S(b)" ];
  LR_6 -> LR_5 [ label = "S(a)" ];
  LR_7 -> LR_8 [ label = "S(b)" ];
  LR_7 -> LR_5 [ label = "S(a)" ];
  LR_8 -> LR_6 [ label = "S(b)" ];
  LR_8 -> LR_5 [ label = "S(a)" ];
}
Tool Overview: IsaViz (W3C)

- Visual representation and modeling of RDFs in N3, N-Triple and RDF-XML Notation
- ONLY RDF, no OWL!
- Merge and Extend support – a real editor!
- Visual component of knowledge representation (DAGs) is a first class citizen through GSS – Graph Style Sheets: IsaViz parses SVG output of GraphViz
- Implemented in Java
- Uses Xerces (Apache) and Jena
  http://jena.sourceforge.net/
- W3C license, GPL compatible
Tool Overview: IsaViz

- **GSS** – Combination of Scalable Vector Graphics (SVG) and CSS allows dynamic visualizations and 'views'
  - Operates on Node-Edge principle of RDFs
- **Fresnel** – Successor of GSS, defined in OWL
  - Representation independent, no Node-Edge principle
  - *What to display* – “Lense”
  - *How to represent*
  - Has it's own selector language opposed to SPARQL
Tool Overview: IsaViz – Fresnel Transformation

Apply Fresnel Lens and style Information
Tool Overview: Protégé (Stanford University / Manchester)

- An Umbrella Project – Pluggable Architecture like Eclipse Platform
- Two distinct modeling methodologies:
  - Frames-Editor, rather “object-oriented”, everything is disallowed until specified
  - OWL-Editor: everything is allowed until restricted (“open world” principle)
- Supports Import of OWL(Lite, DL, Full), RDF, RDFS
- Implemented in Java
  - Uses familiar Open Sourced libraries like Xerces, Jena, Lucene
  - Provides the “de-factor standard OWL API”
- Released under Mozilla Public License, MPL
- Visualization Tools: OWLViz, Jambalaya, and more!
Tool Overview: Protégé: OWLViz

- Enables class hierarchies in an OWL Ontology to be viewed and incrementally navigated
- uses Graphviz as layout engine
- Unlike IsaViz no editor, viewer only
- Here: Pizza Example of Protégé
Tool Overview: Protégé:Jambalaya

- Fundamentally different visualization principle
  - domain-independent visualization technique designed to enhance how people browse and explore complex information spaces
  - uses a nested graph view to present hierarchically structured information.
  - introduces the concept of nested interchangeable views to allow a user to explore multiple perspectives of information at different levels of abstraction.

- University of Victoria's Department of Computer Science, Canada

- A Protégé-Plugin as well a tool on it's own
  http://sourceforge.net/projects/chiselgroup
Tool Overview: Protégé:Jambalaya

Nested View

Nested Treemap

Nested Composite View

useful ontology information is ubiquitous
### Visualization Technologies

- **Treeviews:**
  - easy to construct
  - easy to collapse, “focus” on certain aspects

- **Network views**
  - identify similarity
  - full potential when employed interactively

- **Cluster Maps**
  - identify similarity
  - query result visualization
  - complex ontologies
As OWLs can be reliably represented as DAGs, comparing DAGs for similarity solves the problem of OWL comparison.

Research in graph theory as old as IT research (compiler construction), boost through search engine research
- eg. Map-Reduce algorithm by Google

However: Not all algorithms applicable to Graphs make sense in Ontology context
- Optimizations ("Graph rewriting"), shortest path (traveling salesman), ...

Brief overview over research projects
- COMA++
- BayesOWL
- SOQA-SimPack
Notions of Similarity – Structural

- **Tree Edit Distance** - Number of Edits to transform one tree into another
  - Operations: insert, delete, relabel
  - purely structural, no notion of synonymy
  - Here: Add F, relabel A->G, delete C

- **Maximum common subgraph/ Minimum common supergraph**
  - identifying the `smallest’ graph that contains both graphs

- **Adjacent Matrix**
  - Two graphs are similar if the neighborhoods for every Node are similar.
    - 1st. Graph: Vertice from Node N to M?
    - Establish a Matrix for all 1 .. N nodes and their interconnection
    - Do so for second graph
    - “Compare” Matrices using mathematics
DAG/OWL Similarity

- **Structural Similarity is not enough**

- **But there clearly is similarity**
- **Measure similarity through combined approaches**
  - Adjacent Matrices, String matches, full text similarity match (stemming algorithms!), distance of concepts within ontologies, ...

- **Still not enough**
  - Abbreviations, AssocWords with delimiters (ArrivalAirportIn), Suffix/Prefix (hasName), misspellings, free invented words
Graphical user interface
- Map SQL, XSD, XDR and OWL
- compose, merge and compare of mappings
- **Repository** to persistently store all match-related data, **Model** and **Mapping Pools** to manage schemas, ontologies and mappings in memory, the **Match Customizer** to configure matchers and match strategies, and the **Execution Engine** to perform match operations
- Models are uniformly represented by directed graphs!
- **Mapping Pool** maintains all generated mappings
  - Operations to manipulate them: Diff / Intersect, Merge, MatchCompose → find common minimum
- **Matchers**
  - **Taxonomy Matcher**: Intermediary Ontology for similarity lookup. No use of WordNet, “Hand-taylored Matching Ontology”
  - Experiments with different Lexical Matchers:
  - Divide and Conquer-Strategy: Divide large matching problems into smaller schema fragments
    - Identify similar fragments
    - Each identified similar fragment represents a match problem on its own
  - **Reuse-Oriented Matching**: Based on automated learning and component statistics, reuse previous match results
COMA++

- http://dbs.uni-leipzig.de/Research/coma.html
BayesOWL

- Based on the principle of uncertainty matching
- quantify the degree of the overlap or inclusion between two concepts
- augment OWLs with uncertainty for reasoning in Baysian Networks (BN)
- Convert OWL into DAG and enhance with conditional probability table (CPT) to form a BN
- The CPT is constructed entirely on the information of rdfs:subClassOf, owl:equivalentClass, and owl:disjointWith (structural evaluation) and owl:unionOf, owl:intersectionOf, and owl:complementOf (logical relations)
  - No evaluation of properties and datatypes
- Structural Nodes are translated 1:1 to BN (DAG) nodes, logical relations are constructed using an algorithmus
- The so constructed BN is enhanced with probabilities from the CPT
- The actual mapping of two enriched BN is still an area of research:
- Sources: http://bayesowl.svn.sourceforge.net/viewvc/bayesowl?view=rev&revision=1
SOQA-SimPack Toolkit

- Operates on Ontologies encoded in OWL, PowerLoom, DAML, WordNet
- Supports different concepts of similarity
  - Vector Similarity: Mapping of OWL properties of two ontologies, encoding as binary vectors, calculate vector similarity eg. cosine, jaccard or overlap algorithm
  \[
  \text{sim}_{\text{cosine}}(x, y) = \frac{x \cdot y}{\|x\|_2 \cdot \|y\|_2}
  \]
  - String based similarity: Traverses the underlying graph representation of the ontology resources ie. concepts or data values. The string similarity is expressed as the edit distance, calculated by the levenshtein distance
  - Full text similarity: The whole ontology is expressed in textual representation, normalized using the porter-stemmer and indexed using the Apache Lucene http://lucene.apache.org/ full text indexer
  - Distance based Similarity: A very intuitive measure as the number of sub- or superconcepts in between two ontonologies. Efectively means finding the shortest path between two concepts and measure the distance.
- Supports lexical ontologies like WordNet and ontologies supplied by knowledge bases like CYC http://opencyc.org/

- Architectural overview:
  - SOQA:SIRUP Ontology Query API
    - Defines it’s own Query Language, similar to SQL
    - Uses the generic SimPack Java class library
    - http://www.ifi.uzh.ch/ddis/simpact.html
SOQA-SimPack Toolkit Browser

Concept Hierarchy

- Super_Thing
  - PL-USER-THING
    - COURSE_OWL:TIMERELATEDTHING
    - COURSE_OWL:TIMEUNRELATEDTHING
      - COURSE_OWL:ADDRESS
      - COURSE_OWL:AGENT
        - COURSE_OWL:ORGANIZATIONUNIT
        - COURSE_OWL:PERSON
      - COURSE_OWL:DOCUMENT
      - COURSE_OWL:LOCATION
      - COURSE_OWL:PROGRAMOFSTUDY
      - COURSE_OWL:RESEARCHTOPIC
  - uni-bench_owl:Person
  - uni-bench_owl:Organization
  - uni-bench_owl:Employee
    - uni-bench_owl:AdministrativeStaff
    - uni-bench_owl:Director
    - uni-bench_owl:Faculty
    - uni-bench_owl:ResearchAssistant
  - uni-bench_owl:Student
  - uni-bench_owl:TeachingAssistant
  - uni-bench_owl:Publication
  - uni-bench_owl:Schedule
  - uni-bench_owl:Work

Calculation Type

Similarity(subtree root concept, subtree root concept, single measure)

- First Subtree Root Concept
  - uni-bench_owl:Employee
- Second Subtree Root Concept
  - daml+owl:Thing

Number of Similar Concepts

100

Similarity Measure

Levenshtein

Result

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Concept</th>
<th>Compared Concept</th>
<th>Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>general1_0_damo:Employee</td>
<td>uni-bench_owl:Employee</td>
<td>0.150300000</td>
</tr>
<tr>
<td>1</td>
<td>base1_0_damo:Professor</td>
<td>uni-bench_owl:Professor</td>
<td>0.14705882</td>
</tr>
<tr>
<td>2</td>
<td>base1_0_damo:Dean</td>
<td>uni-bench_owl:Dean</td>
<td>0.14705882</td>
</tr>
<tr>
<td>3</td>
<td>base1_0_damo:Faculty</td>
<td>uni-bench_owl:Faculty</td>
<td>0.140625</td>
</tr>
<tr>
<td>4</td>
<td>base1_0_damo:PostDoc</td>
<td>uni-bench_owl:PostDoc</td>
<td>0.140625</td>
</tr>
<tr>
<td>5</td>
<td>base1_0_damo:Lecturer</td>
<td>uni-bench_owl:Lecturer</td>
<td>0.140625</td>
</tr>
<tr>
<td>6</td>
<td>general1_0_damo:Employee</td>
<td>uni-bench_owl:AdministrativeStaff</td>
<td>0.13333333</td>
</tr>
<tr>
<td>7</td>
<td>base1_0_damo:AdministrativeStaff</td>
<td>uni-bench_owl:AdministrativeStaff</td>
<td>0.13333333</td>
</tr>
<tr>
<td>8</td>
<td>general1_0_damo:Employee</td>
<td>uni-bench_owl:ClericalStaff</td>
<td>0.13333333</td>
</tr>
<tr>
<td>9</td>
<td>base1_0_damo:ClericalStaff</td>
<td>uni-bench_owl:ClericalStaff</td>
<td>0.13333333</td>
</tr>
<tr>
<td>10</td>
<td>general1_0_damo:Employee</td>
<td>uni-bench_owl:SystemsStaff</td>
<td>0.13333333</td>
</tr>
<tr>
<td>11</td>
<td>base1_0_damo:SystemsStaff</td>
<td>uni-bench_owl:SystemsStaff</td>
<td>0.13333333</td>
</tr>
<tr>
<td>12</td>
<td>general1_0_damo:Employee</td>
<td>uni-bench_owl:Director</td>
<td>0.13333333</td>
</tr>
<tr>
<td>13</td>
<td>base1_0_damo:Director</td>
<td>uni-bench_owl:Director</td>
<td>0.13333333</td>
</tr>
<tr>
<td>14</td>
<td>general1_0_damo:Employee</td>
<td>uni-bench_owl:ResearchAssistant</td>
<td>0.13333333</td>
</tr>
<tr>
<td>15</td>
<td>base1_0_damo:ResearchAssistant</td>
<td>uni-bench_owl:ResearchAssistant</td>
<td>0.13333333</td>
</tr>
<tr>
<td>16</td>
<td>base1_0_damo:ResearchAssistant</td>
<td>uni-bench_owl:AssociateProfessor</td>
<td>0.13235794</td>
</tr>
</tbody>
</table>
Thank you!

Questions & Discussion
Links and further reading: Visualization

- GraphSynth: the first publicly available approach to creating graph grammars
  http://www.me.utexas.edu/~adl/graphsynth/
- OntoSphere: more than a 3D ontology visualization tool: SWAP 2005:
- Ontology-based Information Visualization: Towards Semantic Web Applications:
  http://www.cs.vu.nl/~frankh/postscript/VSW05.pdf
- SIMILE http://en.wikipedia.org/wiki/SIMILE
Links and further reading: Graph Similarity

- http://dbs.uni-leipzig.de/file/comaplusplus.pdf
- http://www.thechiselgroup.org/publications#ontology
  - Overview about mapping tools as well as visualization support, no in-depth analysis though
  - Good introductory stuff about semantic matching
  - The Aesthetics of Graph Visualization – Why good visualization matters for the performance of understanding
- Further readings: