Axioms & Templates: Distinctions & Transformations amongst Ontologies, Frames, & Information Models

or

OWL, UML, and Frames

Alan Rector
rector@cs.manchester.ac.uk
Common Questions:

► How do I convert between UML and OWL? Frames & OWL?
  ► How do I determine which properties go with which classes?
    • The “Sanctioning” problem

► “Can I use OWL as a richer schema for databases?”
  • Or to enhance / check database schemas

► How do I say “may” or “typically” in OWL?
  ► How do I manage defaults and exceptions in OWL?

► Why is it so hard for people to switch from frames to OWL
  ► Why do people still use frames? Why switch to OWL?

► How do I get back to what was easy in 1985?
Why I use OWL (for the record)

► Composite concepts and definitions
  ► Left_leg ≡ Leg & has_laterality value left
    ▪ Avoid combinatorial explosions – the “exploding bicycle” or…
      ‣ It’s even made the NY Times:
        ‧ “Roughed up by an Orca? There’s a code for that” http://www.nytimes.com/2013/12/30/technology/medical-billing-nears-a-new-era-of-ultra-specific-codes.html?_r=0
        ‧ >500 codes for kinds of bicycle injuries
        ‧ >200 codes for accidents to space craft crew

► Maintain parallel hierarchies
► Propagate definitions consistently

► Validation & error detection
  ► Difficult, but less difficult than with totally asserted hierarchy

► Basis for Natural Language Generation of Labels
  ► From definitions

► Because it is a standard – and I live in that community
The role of ontologies

▸ Ontology == Knowledge Representation

▸ Is OWL/DLs a general KR language?

▸ Need KR languages be based on logic and axioms?
  ◀ Should they be?
  ◀ Can they be?

▸ How to select a technology for an application?
One approach: Refactor the problem

Key Distinctions

- Ontology vs background knowledge vs information model
- Axiom-based vs Template-based representations
- Class expressions vs Queries in OWL/DLs
- Models of the domain vs Models of Information about that domain

Illustrate starting with UML and OWL; Then discuss frames
Ontology vs background knowledge base vs Information model

New look at an old architecture:

Background Knowledge

- Ontology (Narrow sense)
- Universal Knowledge
- ...Contingent & Other Knowledge...

Information Model / Schema
What is an ontology?

► Is it the same as a knowledge base?
  ► “Conceptualisation of a domain” imprecise
    • If it means everything it means nothing

► Original philosophical meaning: the study of what there is
  ► Useful KR interpretation: Ontology (narrow sense)
    The definitions and essential properties of the entities that can be represented
    • What is necessarily true
      ‣ “by definition”
      ‣ As universal/essential characteristics
        - Representable in logic statements beginning $\forall x \ldots$ 
    • Corresponds to subset of OWL/DL T-Box
Universal Knowledge

- Pneumonia is a lung disease
- Rashs are located on the skin (epithelium)
- Penicillin is an antibiotic

Contingent/Particular Knowledge

- Pneumonia may be caused by bacteria.
- Meningitis *may* cause a rash (Rash is a symptom of Meningitis)
- Penicillins may be used to treat Bacterial Meningitis
Ontology (Narrow Sense)

Universally qualified statements about the domain: true in all possible models/worlds

OWL/DL statements are a subset of such statements in F2

- B subClassOf A \[ \forall x . B(x) \implies A(x) \]
- B subClassOf p some C \[ \forall x . B(x) \implies \exists y . C(y) \land p(x,y) \]
- B EquivalentTo A & p some C \[ \forall x . B(x) \iff A \land \exists y . C(y) \land p(x,y) \]
- B EquivalentTo A & p value c \[ \forall x . B(x) \iff A \land \exists y . C(y) \land p(x,c) \]

Examples

- All pneumonias are lung disease;
  Pneumonia is defined as an Inflammation localised to the lung
  ...

Excludes “contingent” knowledge:
True of given world

- “may”, “typically”, “probably”, “with probability X”, ...

FOL approximations beginning \( \exists \)

FOL approximations that are ground clauses \( p(a,b) \)

- Almost all of a DL A-Box
**Axioms vs Templates**

**Axioms**
- Axioms from which to draw inferences
- Definitions and necessary truths (Universal knowledge)
  - Monotonic, open world, negation as unsatisfiability
  - Composite concepts
- Strictly first order
  - Metaclasses impossible (or kluged)
- Restrict what may be said
  - What may not be said
- Global
- Inferred existence, underspecification
  - “John has a sister”
- Classification inferred & asserted
- Built in two steps
  - Assertion + reasoning (“compiled”)
  - Validation delayed to reasoning-time

**Templates**
- Data structures to be queried.
- Statements, universal & contingent (undistinguished)
  - Non-monotonic (usually), closed world, negation as failure
  - Primitive concepts only
- Higher order
  - Metaclasses essential to representation
- Permit new things to be said
  - What may be said (“sanctioning”)
- Local (to class & descendants)
- Explicit existence (+ skolemization)
  - “John’s sister is Mary”
- Classification asserted only
- Built in one step (“interpreted”)
  - Validation immediate
Domain Knowledge vs Information

**Domain Knowledge Model**
- About the domain
- True or false or uncertain
- Open, at least in parts
- Inferred existence
  - “Has no body temperature” *makes no sense*
- Represents our understanding of a domain
- Variables range over domain entities

**Information Model**
- About the information structures
- Entered or missing
- Closed
- Explicit existence
  - “Missing entry for body temperature” *makes sense*
- Specifies structures to hold information motivated by that understanding
- Variables range over data structures & symbols
# Axioms vs templates, Knowledge vs Data schemas

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Axioms</th>
<th>Templates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge</td>
<td>OWL, Logics, Conceptual Graphs (existential logic) GRAIL axioms</td>
<td>Frames, Conceptual Graphs (cannonical graphs) GRAIL sanctions</td>
</tr>
<tr>
<td>Data schemas</td>
<td>OCL constraints on UML</td>
<td>UML, Archetypes, XML, ...</td>
</tr>
</tbody>
</table>
Three possible reconciliations

► Hybrid models
  ► Represent ontology (narrow sense) in OWL and use for values in UML/Frames

► Represent Templates in OWL or OWL in Templates
  ► Tried representing OWL in templates in Protégé 3
    • problematic
  ► Explore representing templates in OWL
    • Illustrates issues clearly
    • Practical set of transformations and limitations
    • So far explored only with toy examples – needs tooling for larger scale work

► Treat OWL as having dual semantics
  ► Axioms + queries & annotations for templates
  ► Works in HOBO ontology programming environment
Example: What cause pneumonia?

► UML:

Disorder \( \ast \) \( 1..\ast \) Agent

- Disorder entries must be linked to one or more agent entries by the CausedBy association
  - NB: All UML associations are linked implicitly to a class
- Also, any agent can be linked to any number of disorders
- Reciprocal: associations can be traced in both directions
- The agent is mandatory for Disorder; Disorder is optional for agent
  An exception will be raised for missing agents

► Obvious OWL:

Property: CausedBy domain Disorder; range Agent

Class: Disorder subclassOf causedBy some Agent

- All disorders are caused by some agent (even if we don’t know which)
- Unidirectional – & does not generalise easily to other multiplicities
- An agent will be inferred to exist whenever a disorder exists
- Domain/range constraints axioms for inference rather than constraints
  - What properties apply to Disorder hard to determine
Alternative OWL: Model the template
Make Associations classes

► UML:

► Alt OWL:

Property to functional
Property from functional
Class DomainEntity
Class Association \rightarrow to some DomainEntity &
from some DomainEntity
key(to, from)

Class CausedBy \rightarrow Association
Class Disorder \rightarrow DomainEntity &
inv(from) some (CausedBy &
to some Agent)

► Similar meaning but:
  ‣ Schema symmetrical – generalises naturally to all multiplicities
  ‣ Easy to retrieve the associations relevant to any DomainEntity
  ‣ Has direct transformation to/from original for cases where possible
key declaration:

- OWL 2 construct so that each Association instance links exactly one pair of DomainEntities – analogous to prohibiting duplicate rows in a database.
- Multiplicities always associated with DomainEntities, never the association itself

Gain

- Agents may cause Disorders
  - Natural extension to other uses of “may”
  - Natural representation of contingent knowledge
  - Naturally reciprocal
- Ability to say other things about association – e.g. strength, time, etc.
- DL expressions for Association to or from any DomainEntity

Lose

- Transitive relations and property paths (& other property characteristics except functional and inverseFunctional)

Still must content with

- Domain and range declarations are axioms rather than constraints
Comparison to frames

► For “association” substitute “slot”
  ► Almost identical structure

► Gain for frames…
  ► Composition and inferred classification
  ► Clear criteria to distinguish “ontology (narrow sense)”
    • Axioms with DomainEntities on left-hand side

► But still …
  ► No metadata or meta classes
    • except by punning or annotation
  ► Domain & range constraints behave as axioms
    • Inference when reasoning rather than constraints when entering

► Loss to OWL
  ► Transitivity and property paths, etc.
    • Powerful additions to inferences
Restoring transitivity and property paths

Extensions via preprocessing

► Domain and range
  ► Replace with Motik style constraints
    Limited support in current classifiers but easy in preprocessing

► Transitivity and property paths
  ► Specialise to, from & Association for each property
  ► Define a bridging property
  ► Filter out Associations from query results

► Property paths almost work, but queries would include CausedBy class
► Restrict by transformations, e.g.

► (causedBy some X) ➞ (DomainEntity & causedBy some X)
In more detail

Properties:
- to_causedBy ➞ to;
- from_caused_by ➞ from;
- causedByT ➞ bridgingProp, causedByT transitive

property_path: inv(to_caused_by) o from_caused_by ➞ causedBy

Enforce:
- CausedBy ➞ to some C ➞ CausedBy ➞ to_causedBy some C
  ➞ from some C ➞ CausedBy ➞ from_causedBy some C
- Enforce: causedByT some C ➞ DomainEntity & causedByT some C
Metaknowledge & Metaclasses

▶ Use in frames
  ▶ Define templates
    • OWL: Dealt with by Axiomization
  ▶ Annotations
    • OWL: Annotation properties suffice
  ▶ Higher order statements
    ‣ Classes as values – “books about lions”
    ‣ Statements about classes – “Lion is an endangered species”
    • OWL: No fully satisfactory solution
      ‣ Work arounds using Puns & additional post processing
      ‣ Work arounds using annotation properties & additional post-processing
      ‣ Proposed “rich annotations” & layered OWL
        - Neither made it into OWL 2 recommendation
Set of “nearest” existential restrictions or annotations

“Touretzky distance”

Set usually a singleton in a well constructed ontology

Example Tourezky distance measure

\[ t_{\text{nearest}}(p, E) \text{ almost always a singleton} \]

\[
\begin{align*}
p \text{ some } V1 & \leftarrow F \\
B & \rightarrow p \text{ some } V2 \\
C & \\
D & \\
E & \\
\end{align*}
\]

\[ t_{\text{nearest}}(p, E) = \{ V2 \} \]
Knowledge about associations

- Strength, uncertainties
  - Extension to link to Bayesian probabilities a challenge for research

- Evidence / provenance

- Typicality
  - Links to exceptions
Summary: Beware of Differences

- Fundamental distinctions
  - Axioms & templates
  - Ontology (narrow sense) & Contingent/Particular knowledge

- Trade-offs of axioms vs templates
  - *Axioms* – Composition and Classification - ontologies
  - *Templates* – Contingent knowledge and data structures, Higher order (meta) knowledge

- One possible reconciliation & compromise
  - Alternative OWL with reified properties & enforced transformations
    - Gains but expressivity loses other
    - Basis for further extensions and expressivity
    - May sacrifice completeness

- Practical experiments & more theoretical studies needed
  - Specialised environments & tools