Semi-Automatic Information and Knowledge Systems:
Ontology Merging & Integration

Monika Lanzenberger
Outline

- Ontology Reuse
- Integration
- Merging
- Tools
Some Relevant Terms

An ontology is a tuple:

\[ O := (C, H_C, R_C, H_R, I, R_I, A) \]

Combining ontologies \( O_1 \) and \( O_2 \) is done by:

- Merging
- Integration
- Mapping
- Alignment

Ontology Reuse
Ontology Matching

Ontology Reuse

Ontology are artifacts shared by different applications

- Create common ontologies
- Extend them for specific domains and applications
- Using same top-level ontology alleviates reuse problems

However, combination problems need to be solved ...
Typical problems when combining ontologies:

- Practical Problems
- Mismatches between ontologies (or entities)
- Synchronization of the changes made to source ontologies (Versioning)

[Klein 2001]
Problems

problems in ontology combination tasks

practical problems

mismatches between ontologies

language level

finding alignments

diagnosis

repeatability

ontology level

versioning

identification

traceability

translation

conceptualization

syntax

logical representation

semantics of primitives

language expressivity

coverage

concept scope

explication

terminological modelling style encoding

synonyms homonyms concept description paradigm

[Klein 2001, Su 2004]
Syntax (e.g., Class definition):
- `<rdfs:Class ID = "Car">`(RDF Schema)
- `(defconcept Car)` (LOOM)

Logical Representation (e.g., representing disjointness):
- `disjoint A B`
- `A subclass of (NOT B), B subclass-of (Not A)`

Semantics of primitives e.g., same name but different meanings:
- several interpretations of `A equalTo B`

Expressivity: which notions can be expressed (e.g., negation, quantification etc.)

[Klein 2001, Su 2004]
Conceptualization mismatches: difference in the way a domain is interpreted (conceptualized), which results in different ontological concepts or different relations between those concepts.

- Model coverage and granularity: mismatch in the part of the domain that is covered by the ontology, or the level of detail to which that domain is modeled e.g., one vehicle ontology might model cars but not trucks

- Scope: two classes seem to represent the same concept, e.g. employee is described differently depending on the scope

[Klein 2001, Su 2004]
Explication: difference in the way the conceptualization is specified.

Terminological
- Synonym terms: same thing is represented by different names
- Homonym terms: same term has different meanings depending on the context

Modeling style
- Paradigm:
  Different paradigms can be used to represent time, action, plans, causality, etc.
  E.g., time interval versus point
- Concept description:
  several choices can be made for the modeling of concepts, e.g.,
  dissertation < book < scientific publication < publication
  dissertation < scientific book < book < publication
  or as subclass of both book and scientific publication

Encoding, different formats and different languages
- date dd/mm/yyyy or mm-dd-yyyy
- miles or kilometers
- Deutsch or English

[Klein 2001, Su 2004]
Two different types of ontology reuse...

merging:
- building an ontology in one subject reusing two or more different ontologies on that subject
- sources are unified into a single one
- it can be difficult to identify unchanged regions
- truly different ontologies, not simple revisions, improvements or variations of the same ontology

integration:
- building an ontology in one subject reusing one or more ontologies in different subjects
- source ontologies are aggregated, combined, assembled together
- possibly after reused ontologies were changed (extension, specialization or adaptation)
- identification of unchanged regions taken from source ontologies is easier
- integration is more complex than merging

[Pinto, et al. 1999]
Outline

- Ontology Reuse
- Integration
- Merging
- Tools
Integration process takes place along the entire ontology building life cycle

**METHONTOLOGY**

Stages of the building process:
- Specification
- Conceptualization
- Formalization
- Implementation
- Maintenance

...integration should begin as early as possible.

[Pinto and Martins, 2000; Fernández et al. 1999]
Different ontologies: $O, O_1, \text{ and } O_2$
Different domains: $D', D'', \text{ and } D'''$
Integration Activities

Identify

...the possibility of integration

...the modules of the ontology

...the assumptions and ontological commitments for the ontology and each module

...what knowledge should be represented in each module

[Pinto and Martins 2000]
Integration Process

[Diagram showing the steps of an integration process: Identify integration possibility, Identify modules, Identify assumptions & ont. commitments, Identify knowledge to be represented.

- Identify candidate ontologies: find, choose
- Get candidate ontologies: (translate, reengineering)

[Pinto and Martins 2000]
Integration Process

1. Study candidate ontologies
   - evaluate
   - assess

2. Choose most adequate source ontologies

3. Apply integration operations

4. Analyze resulting ontology

[Pinto and Martins 2000]
Choose Candidate Ontologies

Basic requirements:
- appropriate domain
- is the ontology available?
- formalism paradigms in which the ontology is available
- main assumptions and ontological commitments
- main concepts represented

Additional criteria:
- where is the ontology available?
- at what level is the ontology available?
- what kind of documentation is available
- where is the documentation available?

[Pinto and Martins 2000]
Evaluate Candidate Ontologies

Domain experts evaluate the ontologies in terms of:

- what knowledge is missing (concepts, classification criteria, relations, etc),
- what knowledge should be removed,
- which knowledge should be relocated,
- which knowledge sources changes should be performed,
- which documentation changes should be performed,
- which terminology changes should be performed,
- which definition changes should be made,
- which practices changes should be made

[Pinto and Martins 2000]
Assess Candidate Ontologies

Ontology engineers assess the ontologies in terms of:

- the overall structure of the ontology
- appropriateness classification criteria
- the relation used to structure knowledge
- the naming convention rules
- the quality of the definitions, such as using unified patterns, simple, clear, concise, consistent, complete, correct —semantically and syntactically—, precise and accurate
- the quality of the documentation
- appropriateness and completeness of the knowledge pieces (entities) represented or included

[Pinto and Martins 2000]
Taxonomy of features in the **first** stage:

**General**
- Generality
- Formality
- Development status

**Development Content**

[Pinto and Martins 2000]
Taxonomy of features in the **first** stage:

**General**

**Development**
- Knowledge acquisition
  - Quality of knowledge sources
  - Adequacy of knowledge acquisition practices
- Maintenance
  - Is it maintained?
  - Who does maintenance?
  - How is maintenance done?
- Documentation
  - Quality of the documentation available
  - Is the available documentation complete?
- Implementation
  - Language issues

**Content**

[Pinto and Martins 2000]
Taxonomy of features in the **first** stage:

**General Development**

**Content**
- Level of detail
- Modularity
- Adequacy from the domain expert point of view
- Adequacy from the ontologist point of view

[Pinto and Martins 2000]
Taxonomy of features in the second stage:

Content

- Completeness
- Compatibility
  - Terminology of common concepts
  - Definitions of common concepts
Integration of Knowledge

Criteria to guide integration of knowledge:

- Modularize
- Specialize
- Diversify each hierarchy
- Minimize the semantic distance between sibling concepts
- Maximize relationships between taxonomies
- Standardize names of relations

[Ref: Pinto and Martins 2000, Arpirez-Vega et al. 1998]
Analysis of Resulting Ontology

- Clarity, coherence, extendibility, minimal encoding bias and minimal ontological commitment
- Correctness – semantically and syntactically –, completeness, conciseness, consistency, coherency, expandability, sensitiveness and robustness
- Regular level of detail: no ”islands” of exaggerated level of detail

[Pinto and Martins 2000, Gruber 1995, Gómez-Pérez et al. 1995]
• Ontology Reuse
• Integration
• Merging
• Tools
Ontology Merging

...the creation of a single coherent, consistent and non-redundant ontology made up of two or more source ontologies.

Steps in ontology merging:

- Find the places in the ontologies where they overlap - discover mapping candidates
- Relate concepts that are semantically close via equivalence and subsumption relations
- Check the consistency, coherency and non-redundancy of the result

[McGuinness et al. 2000]
Ontology Merging

[Jakoniene 2003]
Merging Example

Ontology_A

Library
  part_of
  part_of
  People
    {name, uid, address, type}
    is_a
    Periodical-Publication
      {title, editor, pages, periodicity}
      is_a
      Journal
        {area, issue_no}
      Newspaper
        {date}

Ontology_B (preferred)

Library
  part_of
  Person
    {name, id, address, role}
    is_a
    Publication
      {title, editors, pages}
      is_a
      Magazine
        {category, issueNo, periodicity}
      Book
        {author, ISBN, style}
      Newspaper
        {date}

Merged_ontology

Library
  part_of
  Person
    {name, id, address, role}
    is_a
    Publication
      {title, editors, pages}
      is_a
      Periodical-Publication
        {periodicity}
        is_a
        Magazine
          {category, issueNo, periodicity}
        Book
          {author, ISBN, style}
        Newspaper
          {date}

[Jakoniene 2003]
Two major architectures for mapping discovery between ontologies exist:

- **Using information sources**
  - A common reference ontology
  - Lexical information
  - Ontology structure
  - User input
  - External resources
  - Prior matches

- **Mapping methods**
  - Heuristic and Rule-based methods
  - Graph analysis
  - Machine-learning
  - Probabilistic approaches
  - Reasoning, theorem proving

[ML]

[Noy 2004, Noy 2005]
Common Reference Ontology

Upper level or reference ontologies designed to support information integration

- designed in principled way
- provide common reference terminology
- Cyc, SUO, DOLCE

Domain-specific interlingua

- Process Specification Language (PSL)

[Noy 2004, Noy 2005]
Using Lexical Information

String normalization
- upper and lower case
- blanks and delimiters
- diacritics
- stop-words

String distance
- Hamming distance
- Levenshtein distance (edit distance)

Soundex

Thesaurus

[Noy 2005]
Natural-language analysis of concept names and definitions

- splitting composite names
- finding common substrings
- finding the ratio of common words in definitions

Hierarchy information of taxonomies

[Noy 2005]
Graph analysis

Treat ontologies as graphs and compare the corresponding subgraphs, e.g. Anchor-Prompt

[Noy 2005, Noy and Musen 2001]
Examples for representation of identified mappings:

- As instances in an ontology of mappings
- Defining bridging axioms to represent transformations
- Using views to describe mappings from a global ontology to local ontologies

[Noy 2005]
Example: Merge Classes with Prompt

- Agency employee
- Employee
- Agent
- Customer
- Traveler

- subclass of
- has client
- agent for

[ML Noy 2001]
Tools: Protégé and Prompt - Merge

Choose the first source project:
file:/Users/monika/Desktop/ontologies/tourismA.ppr

Choose the second source project:
file:/Users/monika/Desktop/ontologies/tourismB.ppr

Choose the algorithm to use in initial comparison:
- Lexical matching

Choose the options for storing mappings:
- Store mapping using a simple mapping ontology
- Choose the Mapping project (optional, must have been generated by)

Click here to begin
Tools: Protégé and Prompt - Compare

[Protégé 3.2]

ISO1 Protégé 3.2 (file:/Users/monika/Desktop/downloads/ISOXML%202/ISO1.pprj, OWL / RDF Files)

CLASS BROWSER
For Project: ISO1
Subclass
- owl:Thing
  - Abstract_objects
    - Class
    - Multidimensional_object
    - Relationship
  - Possible_individual
    - Activity
    - Actual_individual
    - Arranged_individual
  - Event
    - Period_in_time
  - Physical_object
    - Whole_life_individual

CLASS EDITOR
For Class: Activity (instance of owl:Class)

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs:comment</td>
<td>A possible_individual that brings about change by causing the event at the beginning or the event that marks the ending of a possible_individual.</td>
</tr>
<tr>
<td>rdfs:label</td>
<td>Activity</td>
</tr>
</tbody>
</table>

Possible_individual
Outline

- Ontology Reuse
- Integration
- Merging
- Tools
Tools for Integration and Merging

Ontolingua
http://www.ksl.stanford.edu/software/ontolingua/

Protégé (current version 3.3)
http://protege.stanford.edu/

OntoStudio
http://www.ontoprise.de

Chimaera
http://ksl.stanford.edu/software/chimaera/

PROMPT
http://protege.stanford.edu/plugins/prompt/prompt.html

WebODE

CORE: A Tool for Collaborative Ontology Reuse and Evaluation
http://km.aifb.uni-karlsruhe.de/ws/eon2006/eon2006fernandezetal.pdf
References & Resources


References & Resources


