Ontologies in NLP

Corina Forăscu

corinfor@info.uaic.ro

ooNLP: outline

1. Who did what to whom, when & where
2. Fundamentals of ontologies and ontological engineering
3. Modeling and knowledge representation languages
4. Wrapping-up

Definitions: philosophy

Ancient Greeks (Aristotle): ὄντος + λογία

- What is existence?
- Is existence a property?
- Which entities are fundamental?
- What constitutes the identity of an object?

First written mention: 1606

Bailey's dictionary (1721): "an Account of being in the Abstract"

Definitions: Information & Computer Science

(Gruber, 1993): “an explicit specification of a conceptualisation”

Abstract model and simplified view of a phenomenon, of a domain

Concepts, properties, relations, functions, constraints, axioms

Definitions: Information & Computer Science

- A
- *Formal*,
- *Partial Specification* of a
- *Conceptualization*
- *Conceived* by some
- *Rational agent* for some (good or bad)
- *Reason*, and made in order to
- *Negotiate* that conceptualization with
- *Someone* else, or to
- *Reuse* it.

Logic
Representation
Meaning
Cognition
Embodiment
Context
Agreement
Society
Culture

(Gangemi, 2004)

Definitions: Information & Computer Science

(Gruber, 2008): "a set of representational primitives with which to model a domain of knowledge or discourse"

classes (or sets), attributes (or properties), and relationships (or relations among class members)

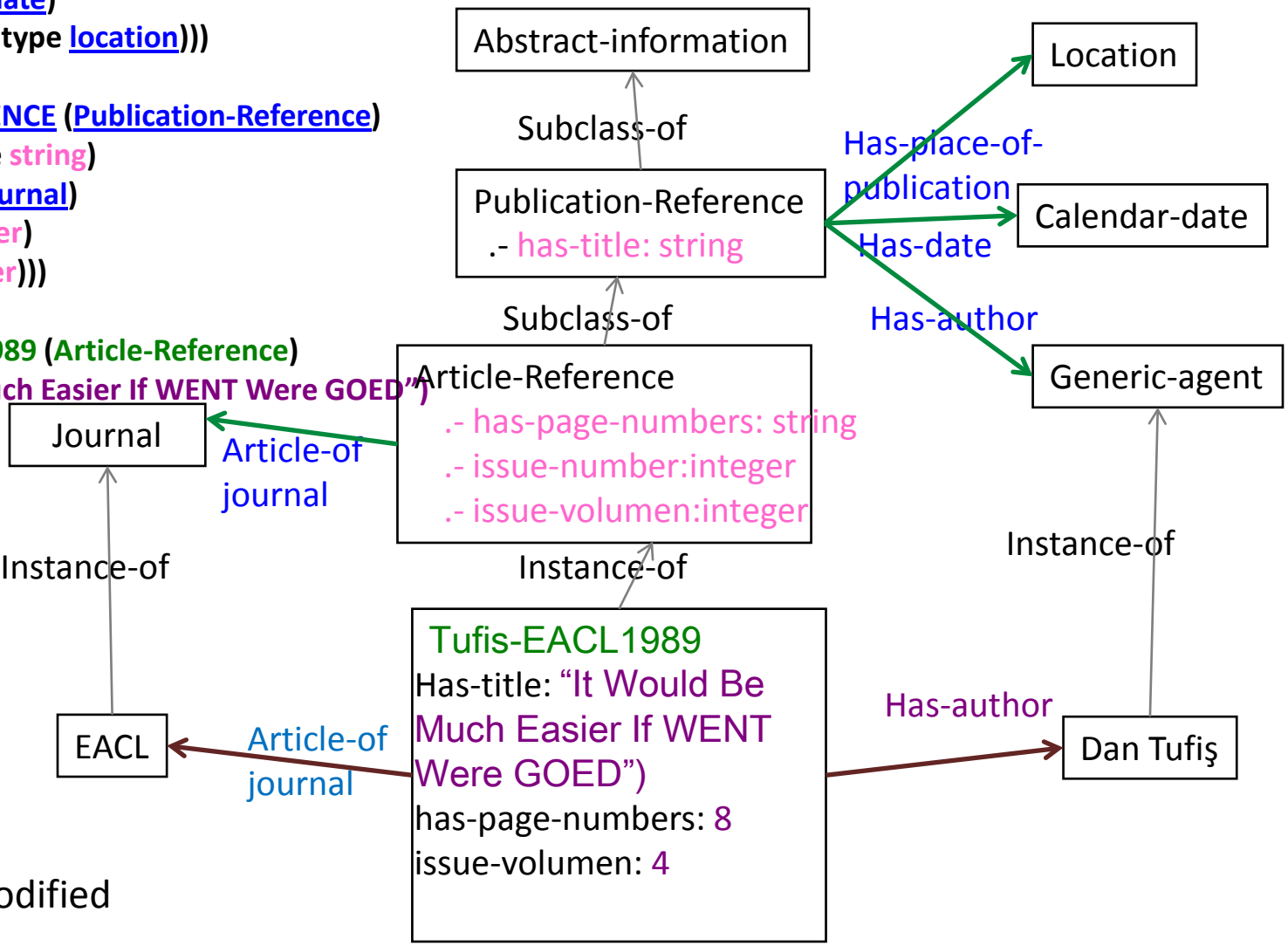
set of objects that can be represented in a declarative formalism for the knowledge of a domain

Dirty hands

(def-class PUBLICATION-REFERENCE (abstract-information)
"we have decided that a publication reference is an intangible, abstract information"
((has-title :type string)
(has-author :type generic-agent)
(has-date :type calendar-date)
(has-place-of-publication :type location)))

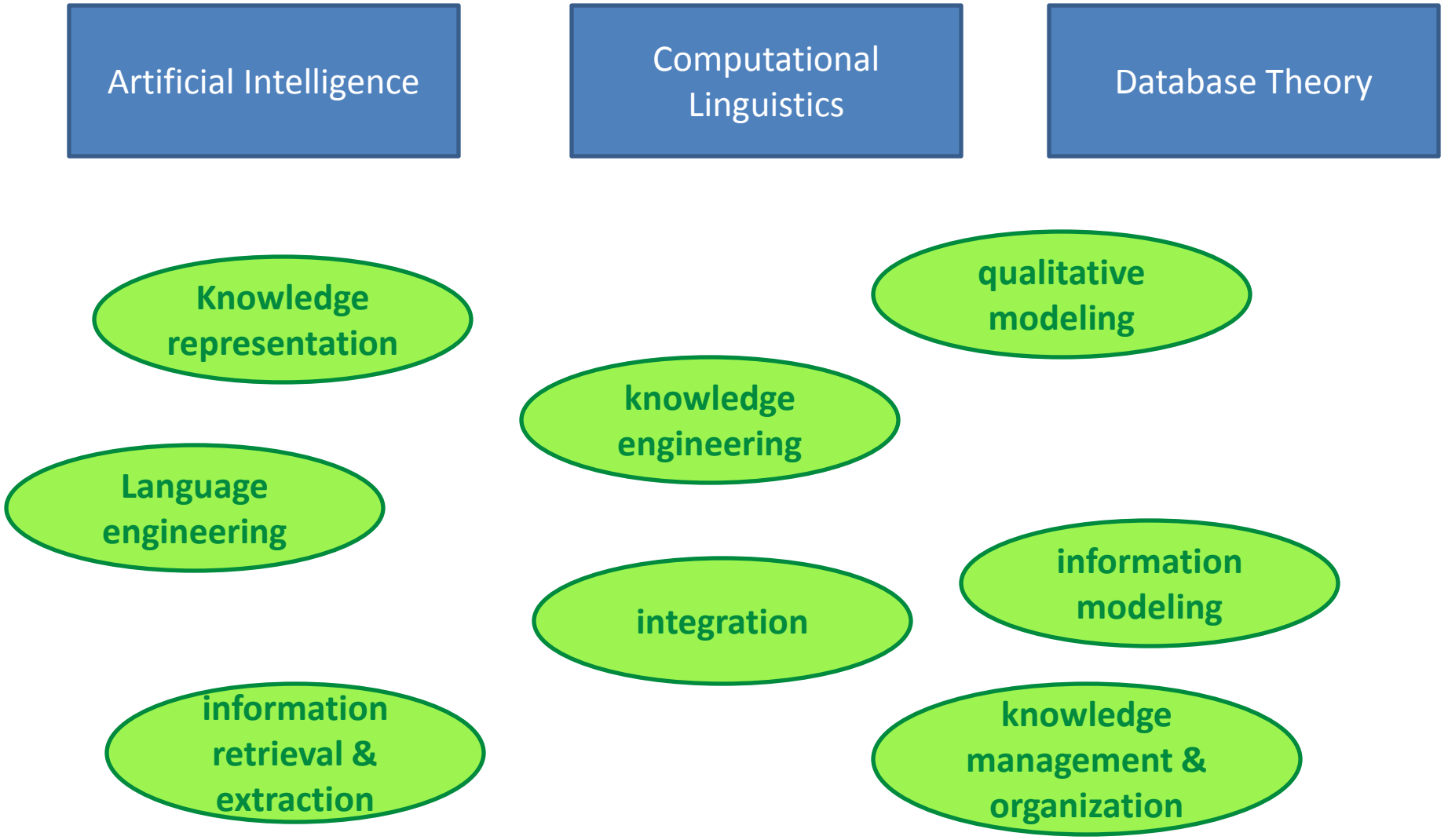
(def-class ARTICLE-REFERENCE (Publication-Reference)
((has-page-numbers :type string)
(article-of-journal :type journal)
(issue-number :type integer)
(issue-volume :type integer)))

(def-instance Tufis-EACL1989 (Article-Reference)
(has-title "It Would Be Much Easier If WENT Were GOED")
(has-author Dan Tufiş)
(has-date 1989)
(has-page-numbers 8)
(article-of-journal EACL)
(issue-volume 4)

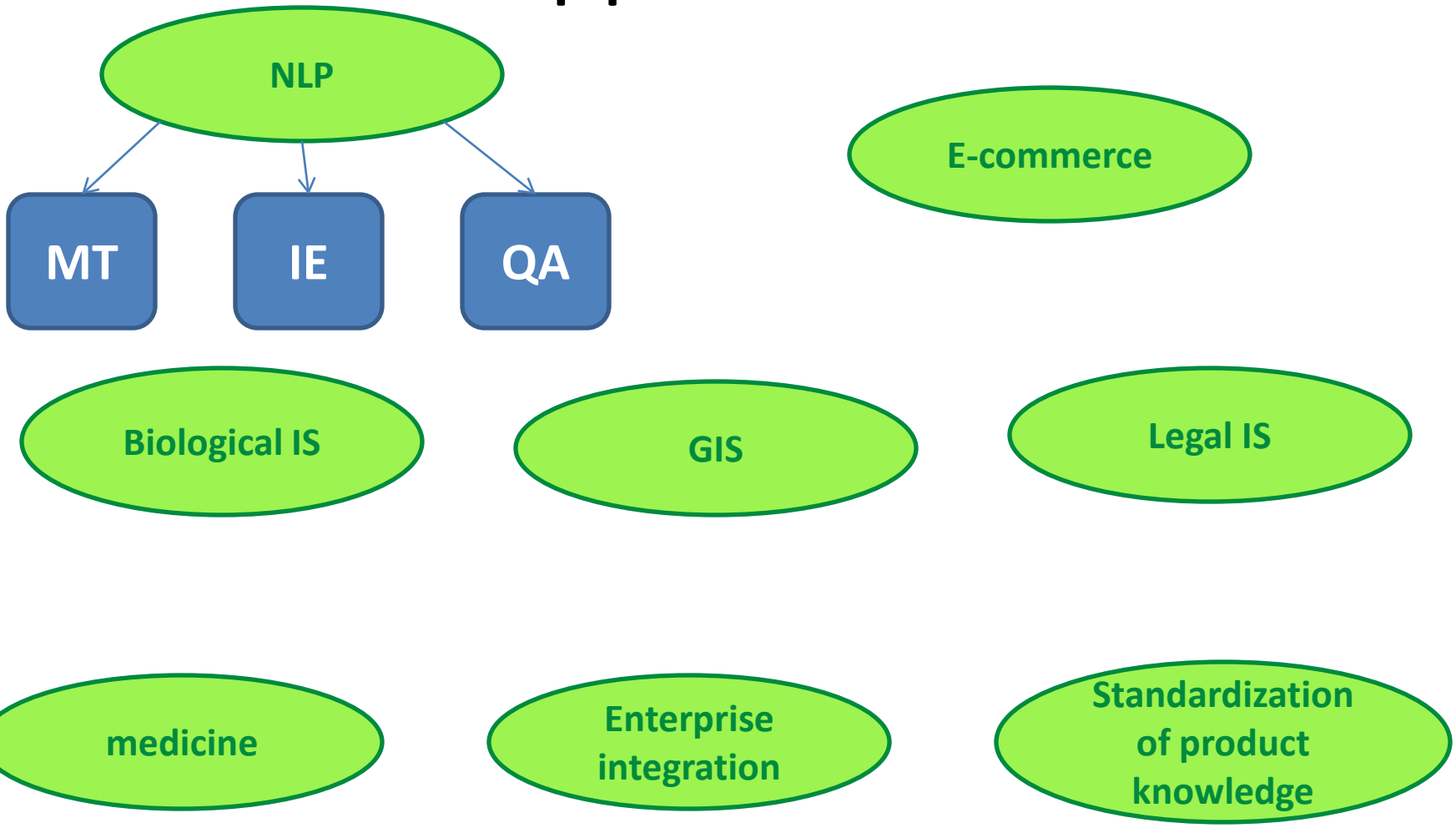


(Gomez, 2004), modified

1. Research Fields



1. Applications



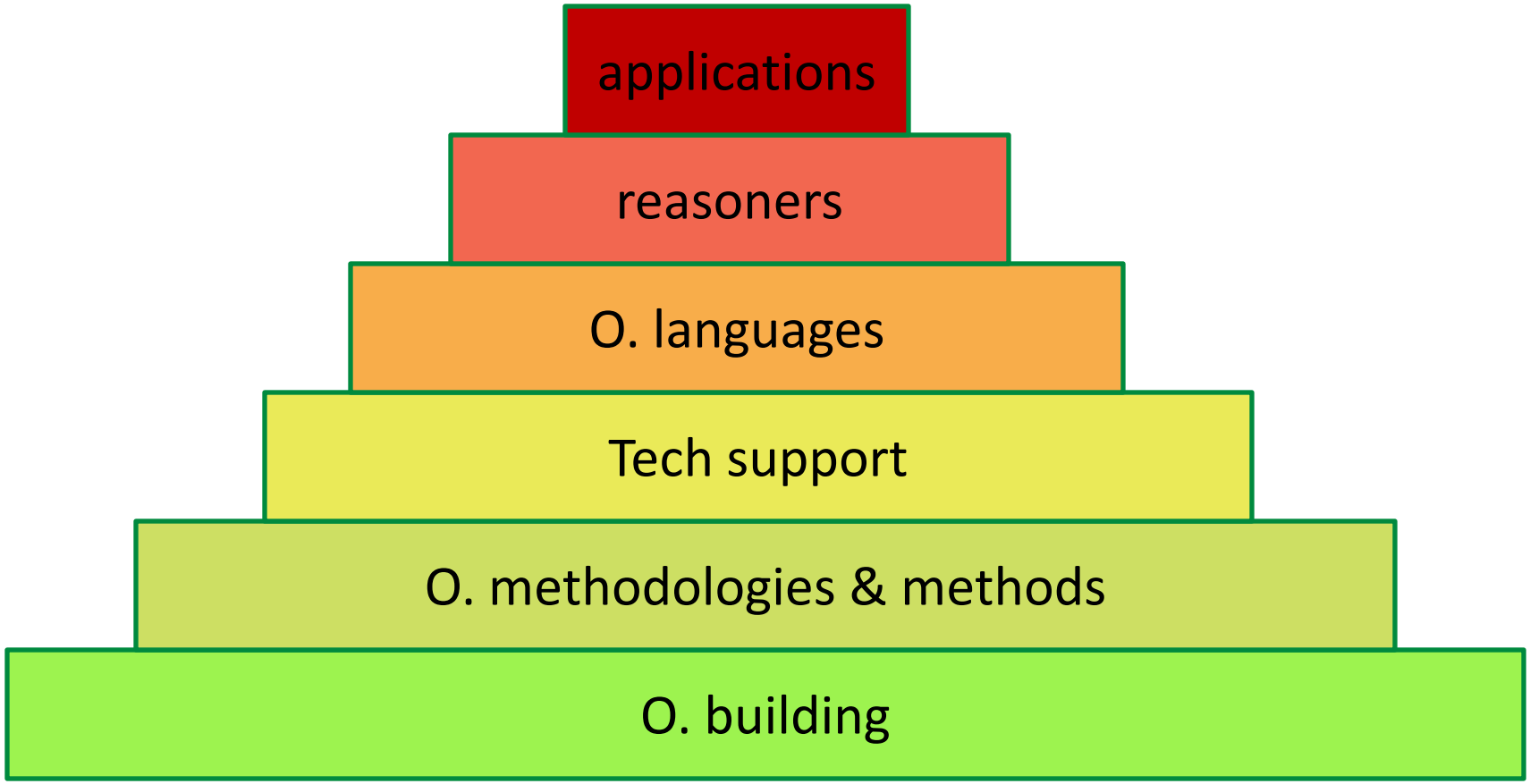
O. libraries

1. DAML ontology library http://www.daml.org/ontologies/
2. Protege ontology library http://protege.stanford.edu/download/ontologies.html
3. Ontolingua ontology library http://ontolingua.stanford.edu/
4. SWEET ontologies http://sweet.jpl.nasa.gov/ontology/
5. SHOE ontology library http://www.cs.umd.edu/projects/plus/SHOE/onts/index.html
6. WebODE ontology library http://webode.dia.fi.upm.es/WebODEWeb/Ontologies.html
7. semanticweb.org ontologies http://semanticweb.org/wiki/Ontology
8. AKT ontology http://www.aktors.org/ontology/

ooNLP: outline

1. Who did what to whom, when & where
2. **Fundamentals of ontologies and ontological engineering**
3. Modeling and knowledge representation languages
4. Wrapping-up

What's in?


applications
reasoners
O. languages
Tech support
O. methodologies & methods
O. building

1. Components (1)

- **Individuals**: instances or objects - the basic objects
  - concrete objects: people, animals, tables, cars, molecules
  - abstract individuals: numbers, measures, words
- **Classes**: sets, collections, concepts, types of objects, or kinds of things
  - *Person*, the class of all people
  - *Class*, the class of all classes
  - *Vehicle*, the class of all vehicles
  - *Thing*, the class of all things
- **Attributes**: aspects, properties, features, characteristics, or parameters that objects (and classes) can have
  - Has-name
  - Has-income
  - Has-author
  - Has-owner
- **Relations**: ways in which classes and individuals can be related to one another
  - *subsumption* relation (*is-a-superclass-of*, the converse of *is-a, is-a-subtype-of* or *is-a-subclass-of*)
  - *meronymy* relation (part-of,): how objects combine together to form composite objects

1. Components (2)

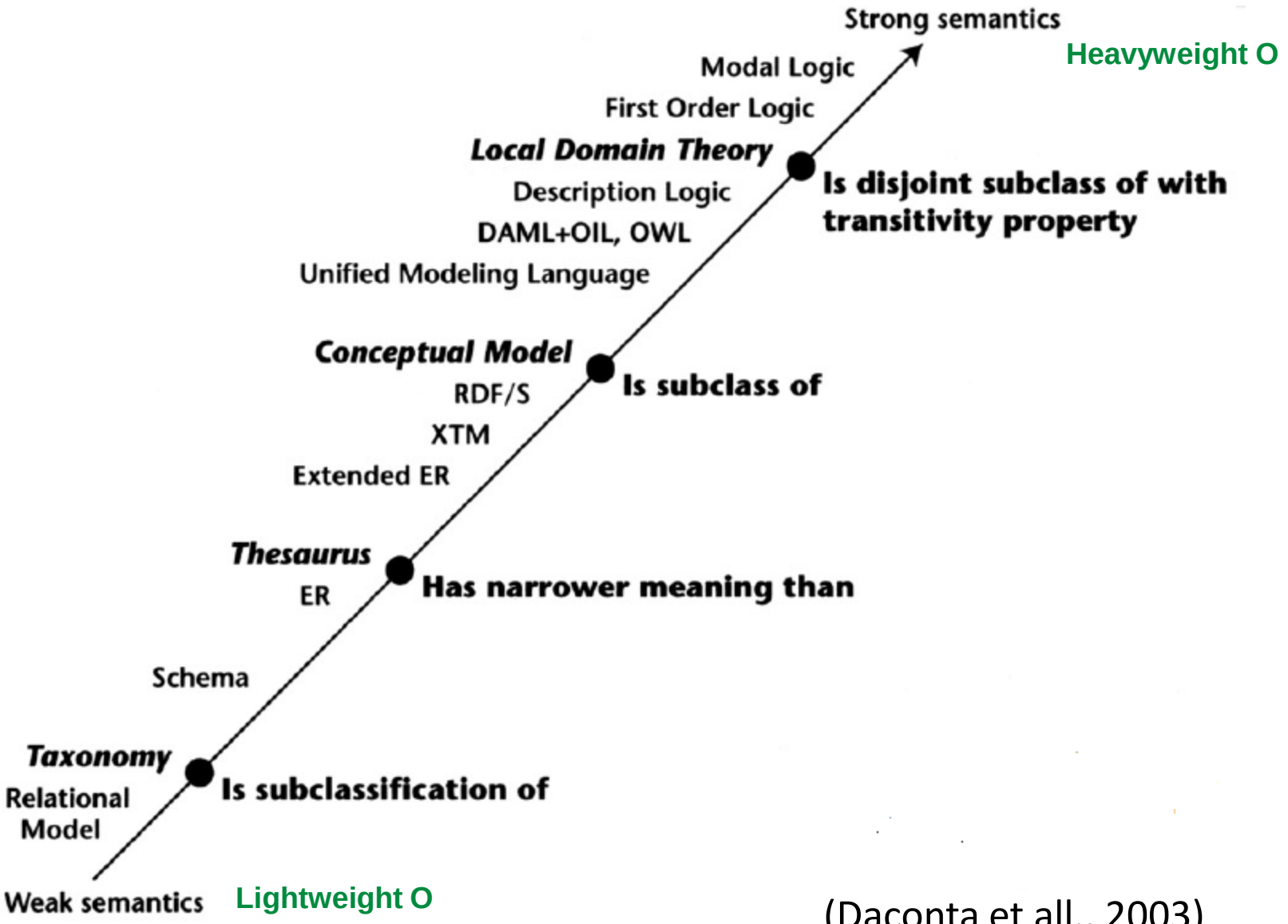
- **Function terms**: complex structures formed from certain relations that can be used in place of an individual term in a statement
- **Restrictions**: formally stated descriptions of what must be true in order for some assertion to be accepted as input
- **Rules**: statements in the form of an if-then sentence describing the logical inferences that can be drawn from a particular assertion
- **Axioms**: assertions (including rules) in a logical form that together comprise the overall theory that the ontology describes in its domain of application
- **Events**: the modification of attributes or relations

Types (taxonomy) of ontologies (1)

- Term Lists
  - Authority Files
  - Glossaries
  - Dictionaries
  - Gazetteers
- Classifications and Categories
  - Subject Headings
  - Classification Schemes
  - Taxonomies
  - Categorization Schemes
- Relationship Lists
  - Thesauri
  - Semantic Networks
  - Ontologies

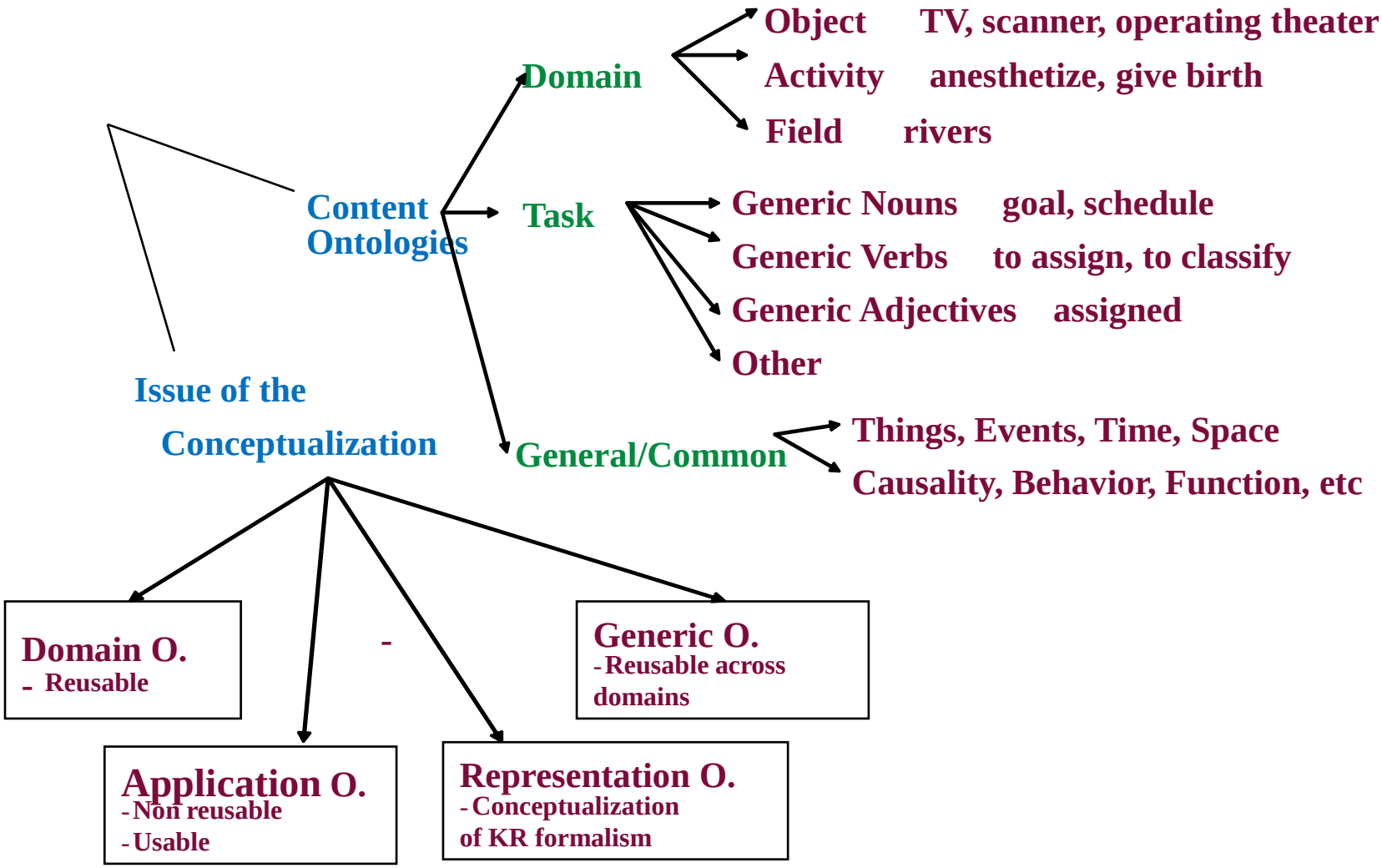
(Dhiman, 2009)

Types (taxonomy) of ontologies (2)



(Daconta et all., 2003)

Types (taxonomy) of ontologies (3)



(Gomez Perez, 2008)

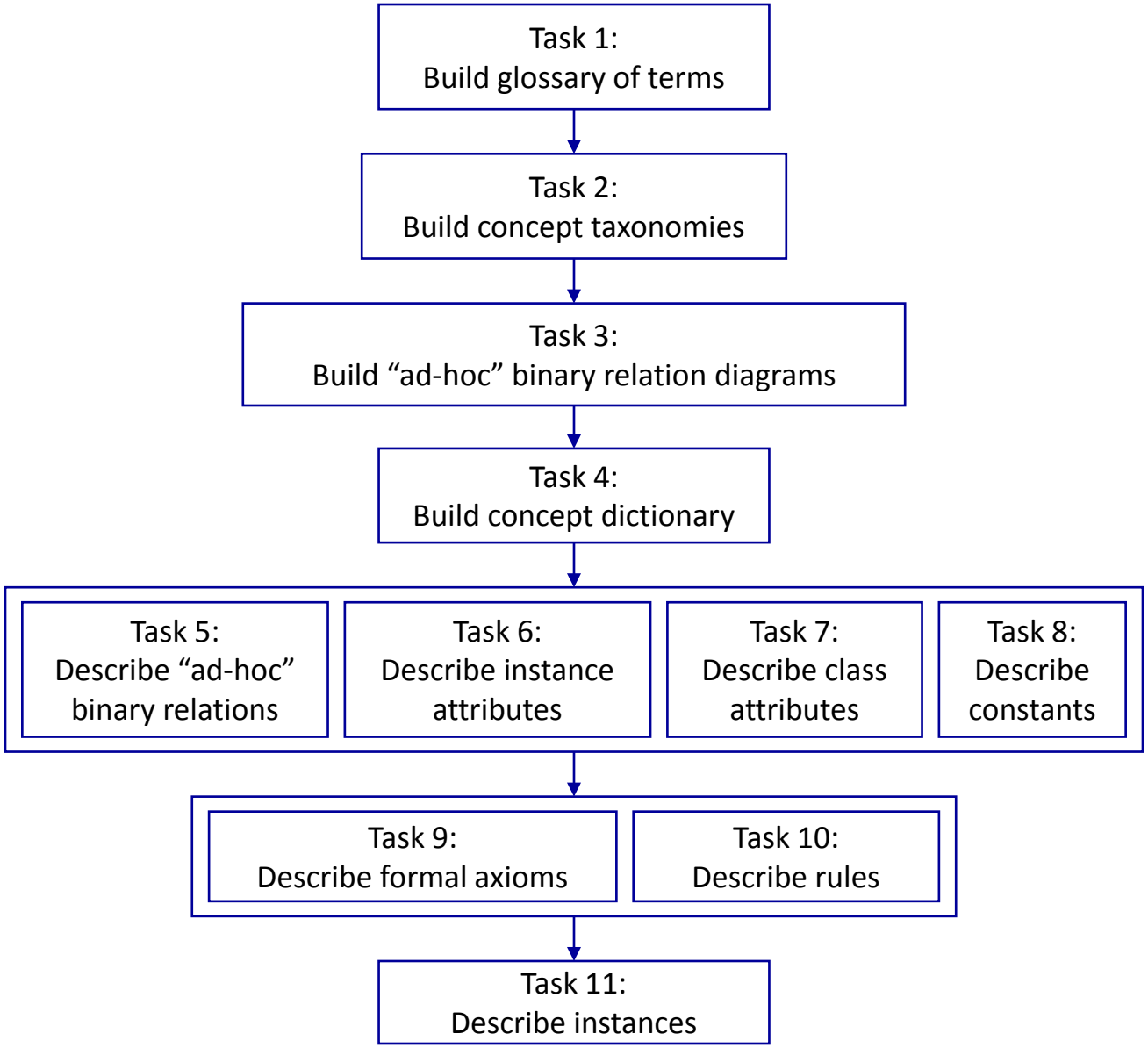
Building Os.

- From scratch
  - Specification
  - Knoweldge Acquisition
  - Conceptualization
  - Integration
  - Implementation.
  - Evaluation
  - Documentation
- Learning from
  - Text
  - Dictionaries
  - Knowledge bases
  - Semi-structured schemata
  - Relational schemata

O. specification

- *Content:*
  - Purpose
  - Scenarios of use
  - Possible end users
  - Level of formality of the ontology
    - highly informal
    - semi-informal
    - semi-formal
    - rigorously formal
  - Scope
  - Granularity
- *Language:*
  - Informal
  - Semi-formal
  - Competency Questions

O. conceptualization



O. Learning

- accelerates the knowledge acquisition process necessary to build an ontology from scratch
- reduces the time required to enrich an existing ontology
- speeds up the construction of Os to be used for different purposes in the Semantic Web.

Techniques in OL

OL from text
- NLP Techniques
- Clustering techniques
- Machine learning
- Statistical approach

OL from dictionary
- NLP
- Statistical approach

OL from knowledge bases
- Rules

OL from semi-structured schemata
- Graph Theory
- Machine Learning
- Pattern Recognition
- Clustering
- Ontological Techniques

OL from relational schemata
- Mapping Techniques
- Reverse Engineering

OL from texts

- Pattern-based extraction
- Association rules
- Conceptual clustering
- Ontology pruning
- Concept learning

OL from texts

- Pattern-based extraction: a relation is recognized when a sequence of words in the text matches a pattern.
- Conceptual clustering: concepts are grouped according to the semantic distance between each other to make up hierarchies. The formulae to calculate the semantic distance between two concepts may depend on different factors and must be provided in these methods
- Concept learning: a given taxonomy is incrementally updated as new concepts are acquired from real-world texts

OL from texts: Association rules

"Given a set of transactions, where each transaction is a set of literals (called items), an association rule is an expression of the form X implies Y, where X and Y are sets of items. The intuitive meaning of such a rule is that transactions of the database which contain X tend to contain Y".

Association rules have been used to discover non–taxonomic relations between concepts, using a concept hierarchy as background knowledge.

OL from texts: Ontology pruning

It builds a domain-ontology based on different heterogeneous sources, in three steps:

1. a generic core ontology is used as a top level structure for the domain-specific ontology
2. a dictionary which contains important domain terms described in natural language is used to acquire domain concepts. These concepts are classified into the generic core ontology
3. domain-specific and general corpora of texts are used to remove concepts that were not domain specific. Concept removal follows the heuristic that domain-specific concepts should be more frequent in a domain-specific corpus than in generic texts

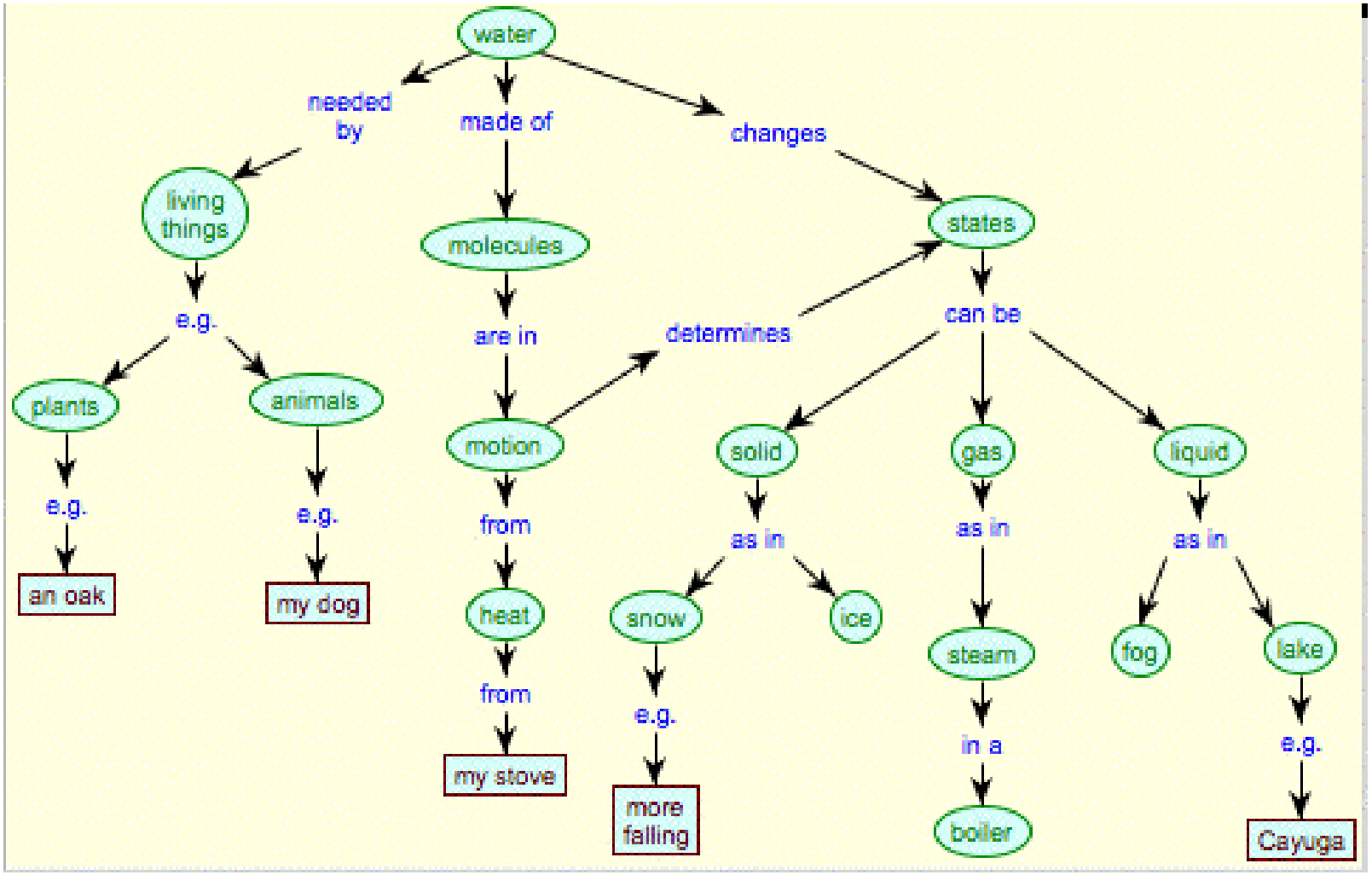
ooNLP: outline

1. Who did what to whom, when & where
2. Fundamentals of ontologies and ontological engineering
3. **Modeling and knowledge representation languages**
4. Wrapping-up

Ontology languages

- Graphical notations
  - Semantic networks
  - Topic Maps
  - UML
  - RDF
- Logic Based
  - Description Logics
  - Rules
  - FOL
  - Conceptual Graphs

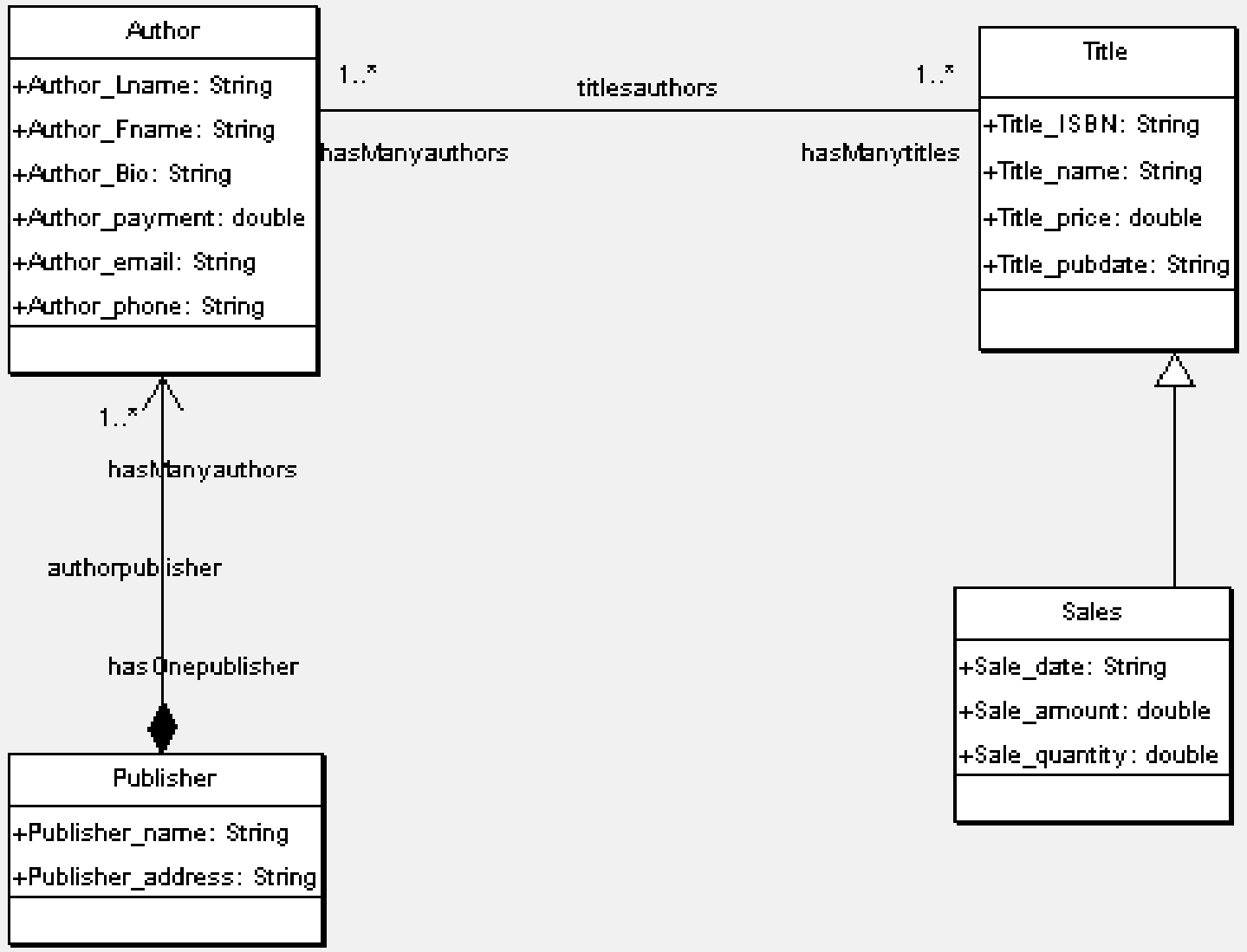
Ontology languages: semantic networks



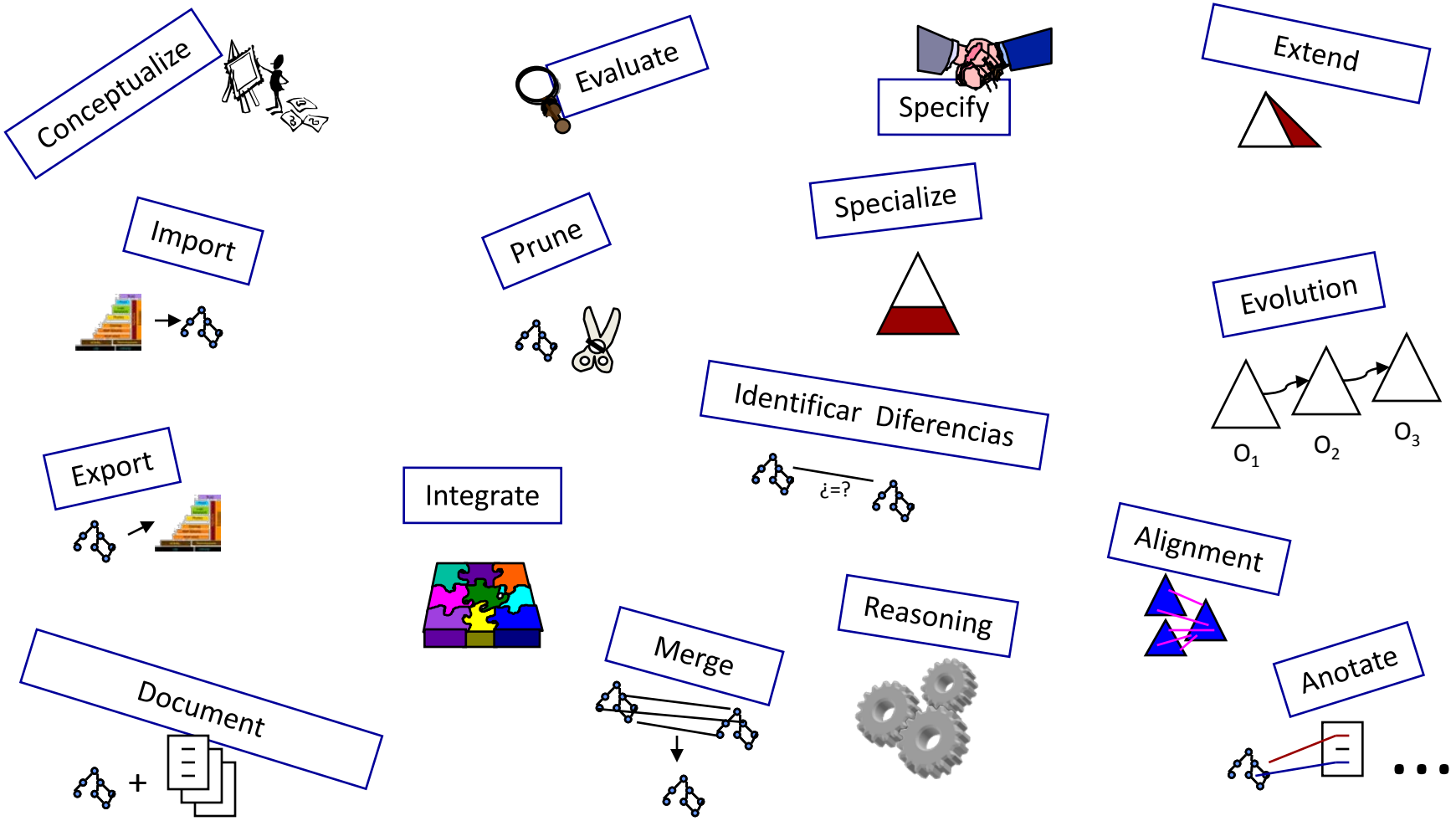
Ontology languages: Topic Maps

- **Topics** might have ***types***, what expresses memberships in classes, that are topics as well, called ***topic types***. A topic can have one or more ***names***.
- **Associations** are n-ary relations (with $n \geq 2$) between topics. Associations might have ***role*** and ***roles types***.
- **Occurences** are "information resources relevant to a topic", i.e. ***instances of a class***. An occurrence might have one or several ***types*** characterising its "relevance to a topic", i.e. memberships into classes.

Ontology languages: UML



Building ontologies



(Gomez-Perez, 2007)

Ontology tools

1. **Protégé** ontology editor and knowledge-base framework: http://protege.stanford.edu/
2. **Apollo**: http://apollo.open.ac.uk/
3. **WebODE** Ontology Engineering Platform: http://webode.dia.fi.upm.es/WebODEWeb/index.html
4. **OntoSaurus** editor: http://www.isi.edu/isd/LOOM/PowerLoom/index.html#Ontosaurus
5. **CLaRK** - corpora & ontology development tool: http://www.bultreebank.org/clark/index.html

Ontology tools

1. **Chimaera** – creating, maintaining, merging and diagnostic web-based browser ontology environment: http://www-ksl.stanford.edu/software/chimaera/
2. **PROMPT** – Protégé plugin for semi-automatic guided ontology merging
3. **MnM** annotation tool for populating ontologies and for adding semantic contents to web pages: http://projects.kmi.open.ac.uk/akt/MnM/
4. **Magpie** semantic filter for the interpretation of web documents: http://projects.kmi.open.ac.uk/magpie/main.html
5. **AquaLog** – portable QA system which takes NL queries and an ontology as input and returns answers drawn from one or more KBs http://kmi.open.ac.uk/projects/akt/aqualog/
6. **Watson** web service for collecting the available semantic content on the Web, for analyzing it to extract useful metadata and indexes, and for efficient query facilities to access these data, based on ontologies http://kmi.open.ac.uk/technologies/name/Watson
7. **Swoogle** - search engine for Semantic Web documents, Web terms and data http://swoogle.umbc.edu/

References

1. T. R. Gruber (1993) A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220
2. Tom Gruber (2008), Ontology. In *Encyclopedia of Database Systems*, Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag
3. Aldo Gangemi (2004) *Ontology Engineering*, SSSW'04 Summer School
4. Asun Gómez-Pérez, Fernández-López, M.; Corcho, O. (2003) *Ontological Engineering*. Springer Verlag
5. Nicola Guarino (1998) *Formal ontology in information systems.* IOS Press
6. Natalya F. Noy and Deborah L. McGuinness. *Ontology Development 101: A Guide to Creating Your First Ontology*
7. http://semanticweb.org/wiki/Ontology
8. http://www.jfsowa.com/ontology/
9. http://www.isi.edu/natural-language/projects/ONTOLOGIES.html
10. http://www.w3.org/2001/sw/WebOnt/