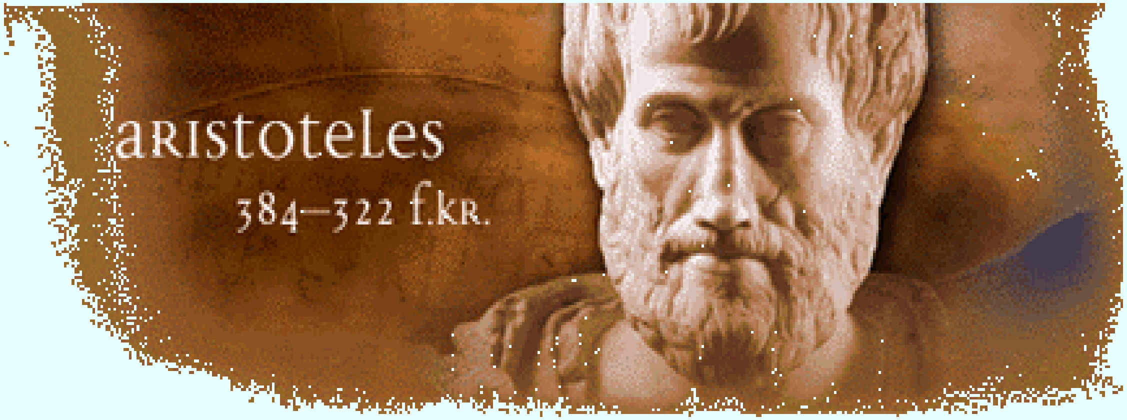
Ontology-Driven Conceptual Modelling

Πληροφοριακά Συστήματα Διαδικτύου

What is Ontology?

- A discipline of Philosophy
  - Meta-physics dates back to Aristotle
  - Ontology dates back to 17th century
- The science of what is ("being qua being")
- Ontology derives from the Greek word "*on*" (being) and "*logos*" (*word*, *speech*, or *reason*).



What is *an* Ontology?

- A specific **artifact** designed with the purpose of expressing the **intended meaning** of a (shared) **vocabulary**
- A shared vocabulary plus a specification (**characterization**) of its intended meaning

  "An ontology is a specification of a conceptualization" [Gruber 95]
- ...i.e., an ontology accounts for the **commitment** of a language to a certain **conceptualization**

What is *an* Ontology?

***A*n ontology is a formal, explicit specification of a shared conceptualization - Gruber**

- ‘Conceptualization’ refers to an abstract model of phenomena in the world by having identified the relevant concepts of those phenomena.
- ‘Explicit’ means that the type of concepts used, and the constraints on their use are explicitly defined.
- ‘Formal’ refers to the fact that the ontology should be machine readable.
- ‘Shared’ reflects that ontology should capture consensual knowledge accepted by the communities

Ontologies in AI and Beyond

- An ontology in AI means a collection of concept definitions: "...In the context of knowledge sharing, I use the term ontology to mean a specification of a conceptualization. That is, an ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist… it is certainly a different sense of the word than its use in philosophy…" [Gruber91].
- Ontologies are used to facilitate "knowledge sharing". Ontologies have a big role to play in data integration as well.

Levels of Ontological Depth

- **Lexicon** -- a vocabulary with definitions.
- **Simple Taxonomy** -- captures taxonomic relationships.
- **Thesaurus** -- taxonomy plus related-terms; captures synonymy, homonymy, etc.
- **Relational Model** -- Unconstrained use of arbitrary relations
- **Fully Axiomatized Theory** -- universal, ontologically neutral language; can specify/characterize fully a conceptualization.

Important Distinctions

- A taxonomy – also from Greek "*taxis*" (arrangement, order, division) and "*nomos*" (law) - is usually applied to structure existing objects and assumed to be a single hierarchical architecture, while an ontology can apply to constructs and can have different kinds of architectures.
- An ontology is more than a taxonomy or classification of terms. Ontologies include richer relationships between terms. It is these rich relationships that enable the expression of domain-specific knowledge. This is a key distinction.

Main components of an Ontology

Five kinds of components:

(1) classes:

- concepts of the domain or tasks, which are usually organized in taxonomies

(2) relations:

- a type of interaction between concepts of the domain (e.g. subclass-of, is-a)

(3) instances:

- to represent specific elements (e.g. Student called Peter is the instance of Student class

Main components of an Ontology

(4) functions:

- a special case of relations in which the n-th element of the relationship is unique for the n-1 preceding elements (e.g. Price-of-a-used-car can define the calculation of the price of the second-hand car on the car-model, manufacturing data and kilometers)

(5) axioms:

- model sentences that are always true (e.g. if the student attends both A and B course, then he or she must be a second year student)

Kinds of ontologies

- Knowledge Representation ontologies
  - capture the representation primitives used to formalize knowledge in KR paradigm
- General/Common ontologies
  - vocabulary related to things, events, time, space, etc.
- Meta-ontologies
  - reusable across domains
- Domain ontologies
  - vocabularies about the concepts in a domain

Kinds of ontologies

- Task ontologies
  - a systematic vocabulary of the terms used to solve problems associated with tasks that may or may not from the same domain
- Domain-task ontology
  - task ontology reusable in a given domain
- Application ontology
  - necessary knowledge for modeling a particular domain

Why Ontologies ?

- The reason ontologies are becoming popular is largely due to what they promise: *a shared and common understanding of a domain that can be communicated between people and application systems*.
- Semantic Interoperability
  - Generalized database integration
  - Virtual Enterprises
  - e-commerce
  - e-Science
- Information Retrieval
  - Decoupling user vocabulary from data vocabulary
  - Query answering
  - Natural Language Processing

Purpose and Benefits

- Fundamentally, ontologies are used to improve communication between either humans or computers. Broadly, these may be grouped into the following three areas:
  - to assist in **communication** between humans. Here, an unambiguous but informal ontology may be sufficient.
  - to achieve **interoperability** among computer systems achieved by translating between different modelling methods, paradigms, languages and software tools. Here, the ontology is used as an interchange format.
  - to improve the process and/or quality of engineering software systems.

Purpose and Benefits

Systems Engineering Benefits: In particular,

- *Re-Usability:* the ontology is the basis for a formal encoding of the important entities, attributes, processes and their inter-relationships in the domain of interest. This formal representation may be a re-usable and/or shared component in a software system.
- Search: an ontology may be used as metadata, serving as an index into a repository of information.
- *Reliability:* a formal representation also makes possible the automation of consistency checking resulting in more reliable software.

Purpose and Benefits

Systems Engineering Benefits

- *Specification:* the ontology can assist the process of identifying requirements and defining a specification for an IT system (knowledge based, or otherwise).
- *Maintenance*: use of ontologies in system development, or as part of an end application, can render maintenance easier in a number of ways.
- *Knowledge Acquisition:* using an existing ontology as the starting point and basis for guiding knowledge acquisition when building knowledge-based systems may increase speed and reliability.

Types of Ontologies

- ***Static Ontologies*** encompass static aspects of an application, described in terms of concepts such as Entity, Attribute, Relationship, Resource,...
- ***Dynamic Ontologies*** encompass dynamic aspects within an application, described in terms of Process, Activity, Action, Plan, Procedure, Event,...or State, Transition,...
- ***Intentional Ontologies*** describe the world of things agents (human or otherwise) believe in, want, prove, argue about, e.g., Issue, Goal, Softgoal, Supports, Denies, SubgoalOf, ...
- ***Social Ontologies*** describe social settings in terms of social relationships among agents, such as Authority, Commitment, Responsibility, Actor , Position, Role, Goal/Task/ Resource Dependency,...

Ontology languages and Conceptual Data Models

- An ontology language usually introduces **concepts** (classes, entities), **properties** of concepts (slots, attributes, roles), **relationships** between concepts (associations), and additional **constraints**.
- Ontology languages may be simple (e.g., having only concepts and taxonomies), frame-based (having only concepts and properties), or logic-based (e.g. OWL).
- Ontology languages are typically expressed by means of diagrams.
- Entity-Relationship schemas and UML class diagrams can be considered as ontologies.

Ontology tools

The tool environment addresses three key aspects:

- Acquiring ontologies and linking them with large amounts of data. For reasons of quality this process requires the human in the loop to build and manipulate ontologies using ontology editors.
- Storing and maintaining ontologies and their instances.
- Querying and browsing semantically enriched information sources.

References

- [Gruber91] Gruber, T., “The Role of a Common Ontology in Achieving Sharable, Reusable Knowledge Bases,” Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, Cambridge, 601-602.
- [Guarino00] Guarino, N. and Welty, C., “Ontological Analysis of Taxonomic Relations,” in A. Länder and V. Storey (eds.), Proceedings of ER-2000: The International Conference on Conceptual Modeling, Springer Verlag LNCS 1920
- [Van Heijst97] Van Heijst, G., Schreiber, A.Th., Wielinga, B.J., “Using explicit ontologies in KBS development”, *Int. J. Human-Computer Studies*, Vol. 45, pp. 183-292, 1997.
- [Studer98] Studer, R., Benjamins, V.R., Fensel, D., “Knowledge engineering: Principles and methods”, *Data & Knowledge Engineering*, Vol. 53, pp. 161-197, 1998.

Metadata

What is Metadata?

- Simply defined, metadata is "data about data.“ Metadata consist of information that characterizes data. Metadata are used to provide documentation for data products.
- In essence, metadata answer **who**, **what**, **when**, **where**, **why**, and **how** about every facet of the data that are being documented.
- Metadata can also describe the content, quality, condition, and other characteristics of data.
- The term "meta" comes from a Greek word that denotes "alongside, with, after, next."

Why bother with Metadata?

- Metadata help people to find the data they need and determine how to use them. Undocumented data can lose their value.
- Subsequent workers may have little understanding of the contents and uses for a digital data base and may find they can't trust results generated from these data.
- The lack of knowledge about other organizations' data can lead to duplication of effort.
- It may seem burdensome to add the cost of generating metadata to the cost of data collection, but in the long run metadata is worth it.

Why bother with Metadata?

- What is now very much needed on the Web is *metadata*. W3C's Metadata Activity is concerned with ways to model and encode metadata. A particular priority of W3C is to use the Web to document the *meaning* of the metadata.
- The association of standardized descriptive metadata with networked objects has the potential for substantially **improving resource discovery** capabilities by
  - enabling field-based searches (e.g., author, title)
  - permitting indexing of non-textual objects
  - allowing access to the surrogate content that is distinct from access to the content of the resource itself

Some metadata standards

- FGDC (Federal Geographic Data Committee) http://www.fgdc.gov/metadata/metadata.html
  - This standard is designed to describe all possible geospatial data.
  - Complex structure and domain specific.
  - There are 334 different *elements* in the FGDC CSDGM standard, 119 of which exist only to contain other elements. These *compound elements* are important because they describe the relationships among other elements.

Some metadata standards

- DC (Dublin Core) http://dublincore.org/
  - The Dublin Core Metadata Initiative is an open forum engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models.
  - Simple structured generic format
  - There are 15 core elements

    (Title, Subject, Description, Creator, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage, Rights)

XML is not enough

- Limitations of XML
  - Many ways to say the same thing. Multiple valid structures for the same data
  - Not impose a common interpretation of a data
    - heading vs. title
    - price vs. cost

RDF

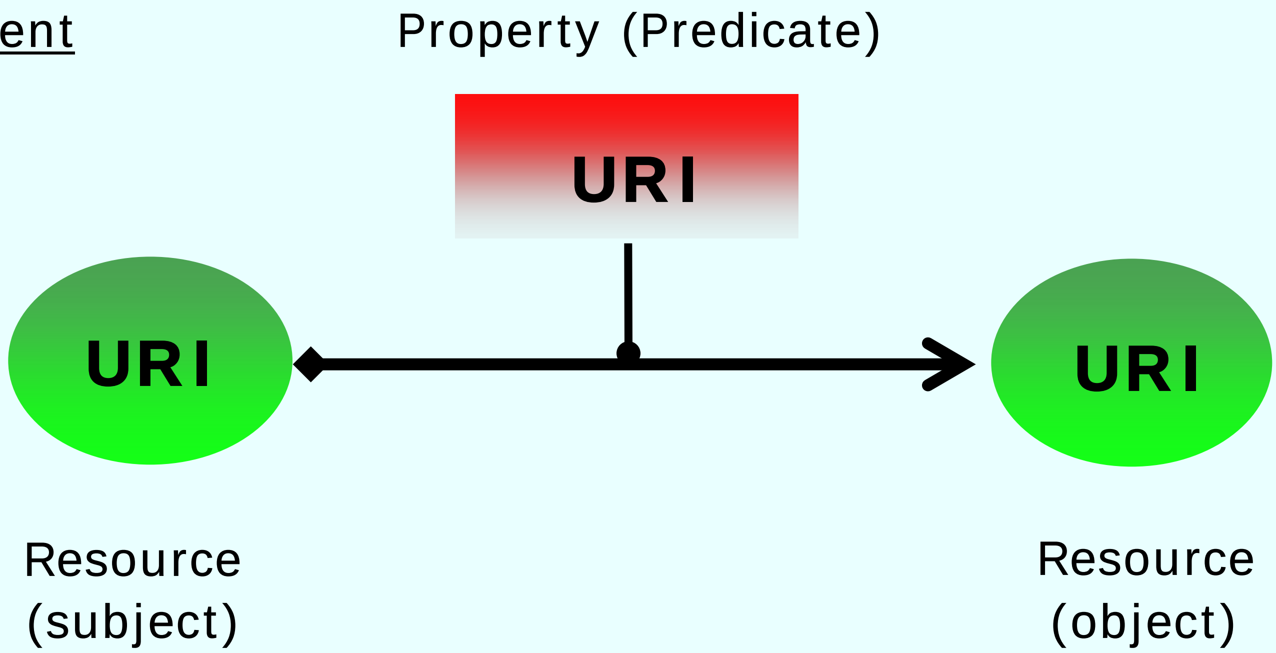
- RDF (Resource Description Framework) is an infrastructure that enables the encoding, exchange and reuse of structured metadata.
- RDF is a foundation for processing metadata
  - It provides **interoperability** between applications that exchange machine-understandable information on the Web.
- RDF is an application of XML that imposes needed structural constraints to provide unambiguous methods of expressing semantics.

RDF Model

- The basic structure of RDF is very simple. There are three components in an ***RDF triple*** :
  - the subject
  - the predicate (property)
  - the object
- Resources
  - All things being described by RDF expressions
- Properties
  - A specific aspect, characteristic, attribute, or relation used to describe a resource
- Statements
  - A specific resource together with a named property plus the value of that property for that resource is an RDF statement.

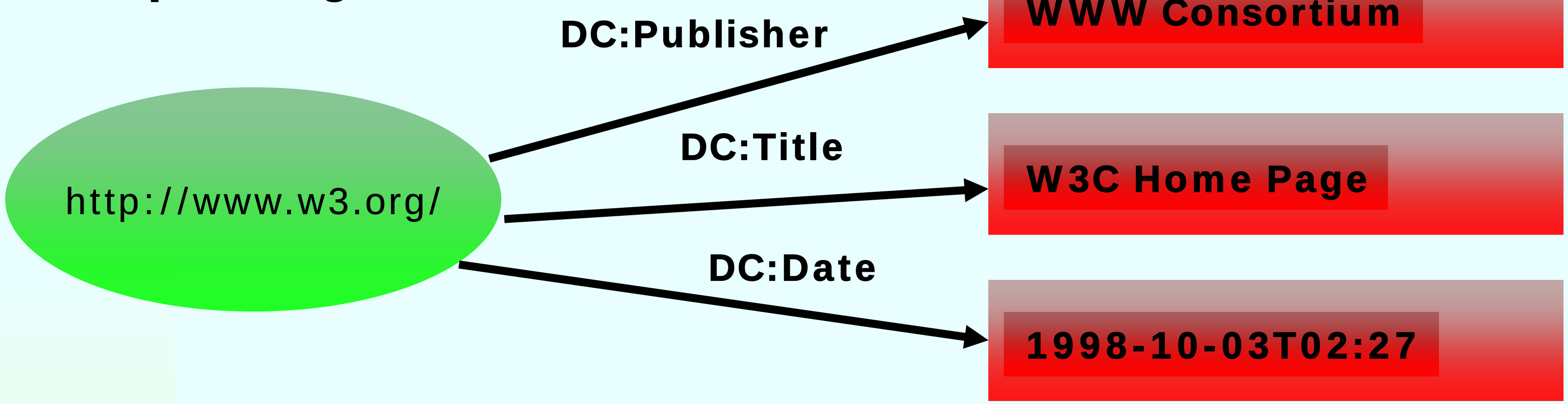
RDF Graph Model

Statement



RDF Example

Graph Diagram



XML Serialization

```
<rdf:RDF>
  <rdf:Description about=“http://www.w3c.org”>
    <DC:Publisher>World Wide Web Consortium</DC:Publisher>
    <DC:Title>W3C Home Page</DC:Title>
    <DC:Date>1998-10-03T02:27</DC:Date>
  </rdf:Description>
</rdf:RDF>
```

XML vs. RDF

- XML was designed for documents, not data.
  - Many features (like attributes and entities) are document-oriented, not for expressing data
  - There are many ways to say the same thing in XML
  - Hybrid tree structure: confusing and nonstandard
  - Makes basic operations more complex (e.g. merging)

- RDF was designed for statements, or data
  - Simple structure: triples
  - Merging two documents are simply combining two into one

Ontology-driven metadata

- Domain specific metadata could be extracted from the ontology.
- Ontologies are a useful mechanism to classify metadata of various resources.
- Metadata extraction involves named entity identification and semantic disambiguation to extract syntactic and contextually relevant semantic metadata.