RESO®

2018 Spring Technology Summit

Project ORREO Foundry: Open Residential Real Estate Ontology Foundry

IT'S GAME ON FOR DATA STANDARDS!

SPEAKER

BIO PAGE

Tavi Truman | CTO

BIO: Celebrating more than 25 years in Software sciences, Information, Cognitive and Applied Ontology Science and Commercial software development. Currently, Tavi leads all aspects of the RocketUrBiz Engineering. His focus is on Ontological Engineering, Real Estate Informatics, Automated Reasoning and Workflow processing.

RocketUrBiz
Harnessing Technology, Empowering People

Email:tavi@rocketurbiz.com

Phone: +1 408-214-0916

Website: http://www.rocketurbiz.com

RESO®
2018 Spring Technology Summit

What Is Ontology …

2018 Spring Technology Summit

Robert Arp, Ph.D.

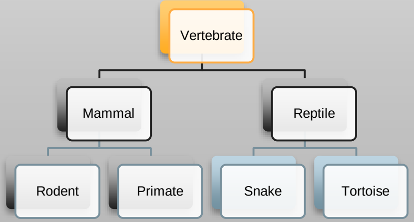
The word ‘ontology’ can refer to a branch of Western philosophy—having its origins in ancient Greece with philosophers such as Parmenides, Heraclitus, Plato, and Aristotle—the concern of which is the study of what is, of the kinds and structures of objects, properties, events, processes, and relations in every area of reality.

From this philosophical perspective, ontology seeks to provide a definitive and exhaustive classification of entities in all spheres or domains of being. As a theoretical discipline concerned with accurately describing the taxonomy of all things that exist, philosophical ontology is synonymous with classical metaphysics.

Dr. Barry Smith, Ph.D.
SUNNY Award – Applied Ontology

ontology = def. a representational artifact, comprising a taxonomy as proper part, whose representations are intended to designate some combination of universals, de-fined classes, and certain relations between them.

- Universal = type, kind of thing or Entity



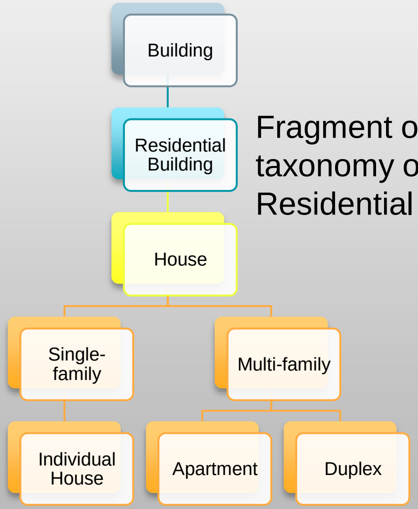
What is Ontology (continued)

taxonomy = def. a hierarchy consisting of terms denoting types (or universals or class-es) linked by subtype relations

By “ types ” or “ universals ” we mean the entities in the world referred to by the nodes (appearing here as boxes) in a hierarchy; in the case of figure 1.1 , biological phyla, classes, and orders.

entity = def. anything that exists, including objects, processes, and qualities “ Entity ” thus comprehends also representations, models, images, beliefs, utterances, documents, observations, and so on.

A **building,** or **edifice,** is a structure with a roof and walls standing more or less permanently in one place.

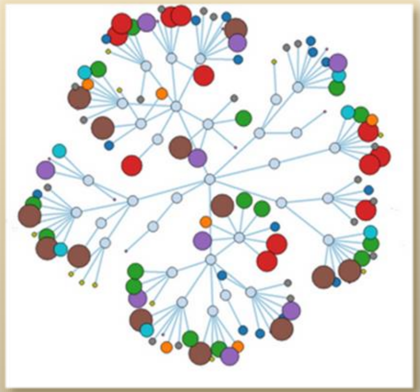


Fragment of a simple taxonomy of Residential Building

What must an Ontology contain

Visualizations in Graph Theoretic form

- Terms (represent types in reality)
  - Preferred labels
  - Synonyms
- Unique IDs
  - Alphanumeric identifiers for each term
  - Namespace ID
- Nodes - terms
  - Complex collections of things
  - Noun or noun phrases
- Edges
  - relations
- Definitions
- Axioms
  - Governs how the terms are to be understood



2018 Spring Technology Summit

The Vision of the Semantic Web powered by Ontology

Tim Berners-Lee, inventor of the internet:
“sees a more powerful Web emerging, one where documents and data will be annotated with special codes allowing computers to search and analyze the Web automatically. The codes … are designed to add meaning to the global network in ways that make sense to computers”

RESO®

2018 Spring Technology Summit

Why do we need Ontology

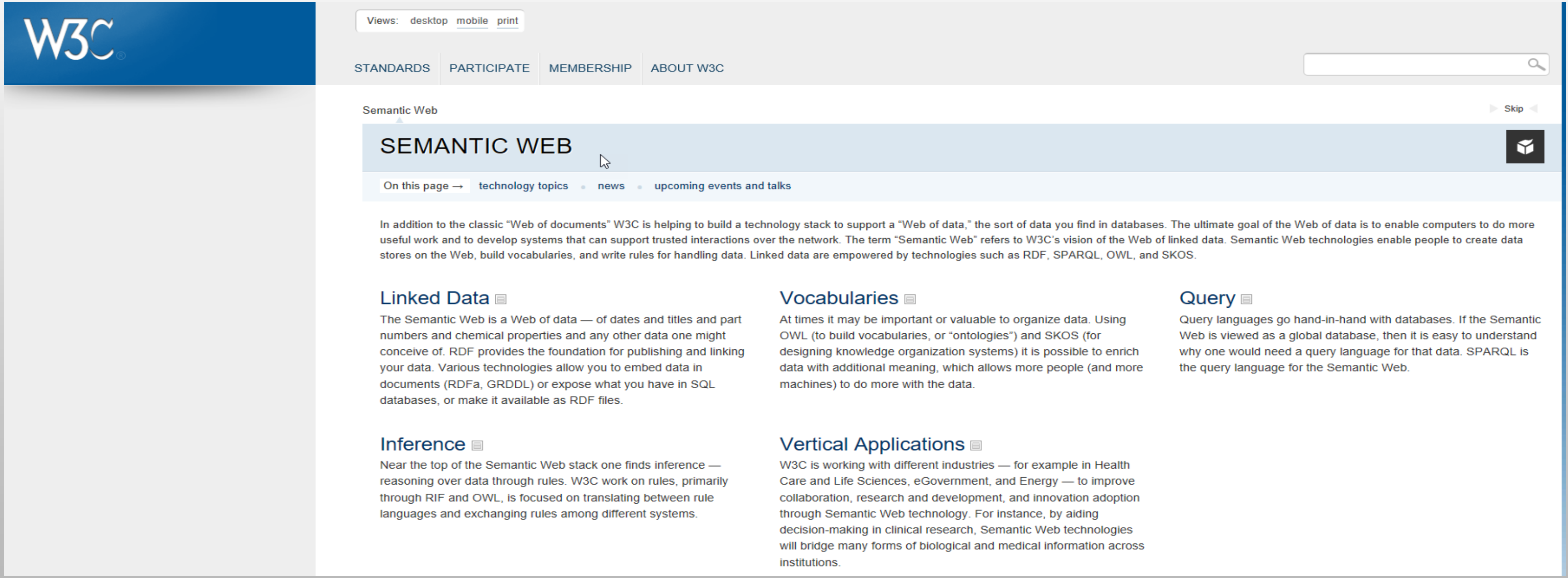
Tim Berners-Lee:
hyperlinked vocabularies, called ‘ontologies’ will be used by Web authors “to explicitly define their words and concepts as they post their stuff online.

“The idea is the codes would let software ‘agents’ analyze the Web on our behalf, making smart inferences that go far beyond the simple linguistic analyses performed by today's search engines.”

RESO®

2018 Spring Technology Summit

Working together: W3C Standards for the Semantic Web



W3C

Views: desktop mobile print

STANDARDS PARTICIPATE MEMBERSHIP ABOUT W3C

Semantic Web

Skip

SEMANTIC WEB

On this page → technology topics • news • upcoming events and talks

In addition to the classic "Web of documents" W3C is helping to build a technology stack to support a "Web of data," the sort of data you find in databases. The ultimate goal of the Web of data is to enable computers to do more useful work and to develop systems that can support trusted interactions over the network. The term "Semantic Web" refers to W3C's vision of the Web of linked data. Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data. Linked data are empowered by technologies such as RDF, SPARQL, OWL, and SKOS.

Linked Data

The Semantic Web is a Web of data — of dates and titles and part numbers and chemical properties and any other data one might conceive of. RDF provides the foundation for publishing and linking your data. Various technologies allow you to embed data in documents (RDFa, GRDDL) or expose what you have in SQL databases, or make it available as RDF files.

Vocabularies

At times it may be important or valuable to organize data. Using OWL (to build vocabularies, or "ontologies") and SKOS (for designing knowledge organization systems) it is possible to enrich data with additional meaning, which allows more people (and more machines) to do more with the data.

Query

Query languages go hand-in-hand with databases. If the Semantic Web is viewed as a global database, then it is easy to understand why one would need a query language for that data. SPARQL is the query language for the Semantic Web.

Inference

Near the top of the Semantic Web stack one finds inference — reasoning over data through rules. W3C work on rules, primarily through RIF and OWL, is focused on translating between rule languages and exchanging rules among different systems.

Vertical Applications

W3C is working with different industries — for example in Health Care and Life Sciences, eGovernment, and Energy — to improve collaboration, research and development, and innovation adoption through Semantic Web technology. For instance, by aiding decision-making in clinical research, Semantic Web technologies will bridge many forms of biological and medical information across institutions.

RESO

2018 Spring Technology Summit

NextGen Information Systems – Powered by Ontology: The OBO Foundry

About ▾ Principles ▾ Ontologies ▾ Citation ▾ Participate ▾ FAQ ▾ Legacy ▾ Search Ontobee Submit

The OBO Foundry

The Open Biological and Biomedical Ontology (OBO) Foundry is a collective of ontology developers that are committed to collaboration and adherence to shared principles. The mission of the OBO Foundry is to develop a family of interoperable ontologies that are both logically well-formed and scientifically accurate. To achieve this, OBO Foundry participants voluntarily adhere to and contribute to the development of an evolving set of principles including open use, collaborative development, non-overlapping and strictly-scoped content, and common syntax and relations, based on ontology models that work well, such as the Gene Ontology (GO).

The OBO Foundry is overseen by an Operations Committee with Editorial, Technical and Outreach working groups. The processes of the Editorial working group are modelled on the journal refereeing process. A complete treatment of the OBO Foundry is given in "The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration".

On this site you will find a table of ontologies, available in several formats, with details for each, and documentation on OBO Principles.

You can contribute to this site using GitHub OBOFoundry/OBOFoundry.github.io or get in touch with us at obo-discuss@sourceforge.net.

RESO®

2018 Spring Technology Summit

Modern Information System Design moving to Ontology

Download table as: [YAML | JSON-LD | RDF/Turtle]

bfo	Basic Formal Ontology (CC BY)	The upper level ontology upon which OBO Foundry ontologies are built. Detail	
chebi	Chemical Entities of Biological Interest (CC BY)	A structured classification of molecular entities of biological interest focusing on 'small' chemical compounds. Detail	
doid	Human Disease Ontology (PUBLICDOMAIN)	An ontology for describing the classification of human diseases organized by etiology. Detail	
go	Gene Ontology (CC BY)	An ontology for describing the function of genes and gene products Detail	
obi	Ontology for Biomedical Investigations (CC BY)	An integrated ontology for the description of life-science and clinical investigations Detail	
pato	Phenotype And Trait Ontology (CC BY)	An ontology of phenotypic qualities (properties, attributes or characteristics) Detail	
po	Plant Ontology (CC BY)	The Plant Ontology is a structured vocabulary and database resource that links plant anatomy, morphology and growth and development to plant genomics data. Detail	

2018 Spring Technology Summit

Project ORREO Foundry

The Open Residential Real Estate Ontology (ORREO) Foundry is a collective of ontology developers that are committed to collaboration and adherence to shared principles. The mission of the Foundry is to develop a family of interoperable ontologies that are both logically well-formed and scientifically accurate. To achieve this, ORREO Foundry participants voluntarily adhere to and contribute to the development of an evolving set of principles including open use, collaborative development, non-overlapping and strictly-scoped content, and common syntax and relations, based on ontology models that work well, such as the Gene Ontology (GO).

The ORREO Foundry is overseen by an Operations Committee with Editorial, Technical and Outreach working groups. The processes of the Editorial working group are modelled on the journal refereeing process. A complete treatment of the ORREO Foundry is given in "The ORREO Foundry: coordinated evolution of ontologies to support residential real estate data integration".

RESO
2018 Spring Technology Summit

Project ORREO Foundry Resources

- ORREO Foundry @ Github.com
  - MIT Commons License
- Website: http://www.orreofoudry.org
  - Tools
  - Ontological Browser
  - Protégé Support
- OWL 2 DL
  - OWL/XML, RDF/XML, JSON-LD, Turtle serialization
  - DL Reasoners
  - Common Logic Reasoner
  - DL Expression Explorer
- BFO and IAO Upper Ontologies
- Modular Design pioneered by Dr. Barry Smith (Ontological Engineering)
- Autodesk BIM Integration
  - Building Information Model Integration
- MS Azure IoT
- Smart City Collaboration

Project ORREO Foundry and RESO

- ORREO Foundry is about Real Estate and should be collaborative in it workings
- Ontology Development effort is never considered to reach a complete state but Ontology do come to a “good-enough” state
- RESO will play an important role providing guidance and experience and insights
- RESO+ORREO will bring many more Real Estate professionals to the party; thus expanding reach of knowledge, learnings, experience and drive new business and technological innovation

- People Ontology
- Email Ontology
- Emotion Ontology
- Material Ontology
- Geographical Ontology
- Many others

- ORREO will create new ontologies that span all aspects of reality where real estate is applicable

RESO
2018 Spring Technology Summit

Project ORREO Foundry: Multidisciplinary Teams

- Building Architects
- Home Builders
- Landscape Designers and Engineers
- Interior Decorates
- Appliance Manufactures
- Power Utility (Gas, Water, Electric, Solar)
- Heating and Cooling
- Home & Property Insurance
- Mortgage Lending (FIBO – Financial Industry Business Ontology)
- Semantic Bank Compliance Ontology + Legal Knowledge Interchange Format (LKIF)
- Construction
- Home Security
- Whole-House Systems Approach Ontology @ Energy.gov
- More..

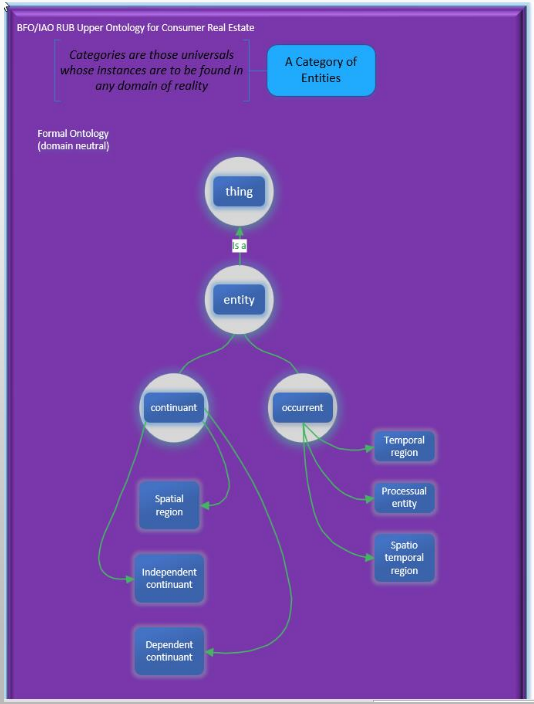
RESO®

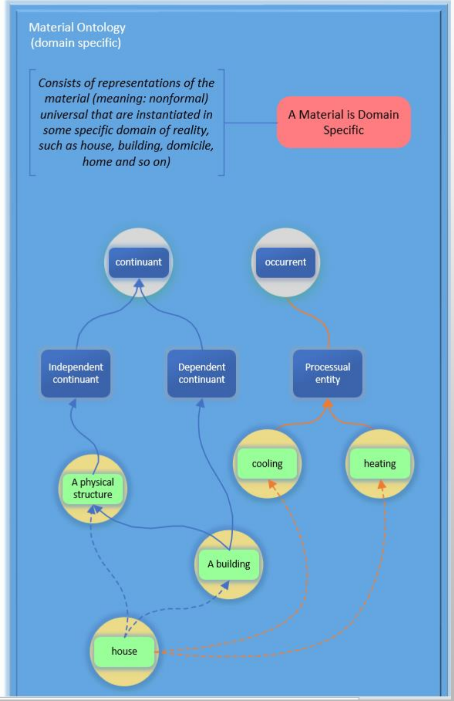
2018 Spring Technology Summit

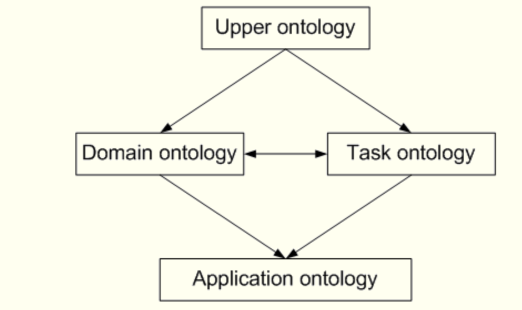
Protégé Demo

2018 Spring Technology Summit

Start with BFO and IAO Upper Ontology







Modularization of ontologies depending on the scope and partial ordering defined by inheritance

RESO®

2018 Spring Technology Summit

We will be able to use ontologies to help us share data only if

- they are ontologically coherent (intelligible to a human user)
- and logically coherent
- and computationally tractable
- and work well together
    - evolve together
    - created according to the tested rules

RESO®

2018 Spring Technology Summit

Data Model - Purpose

- To provide a consistent and efficiently functioning data store for a particular business application(s)
  - Represents specific business concepts in a way that determines organization of data in the store
  - Commonly used representations are relational and graph; they are supported by data management technologies, e.g. relational
  - Oracle and MySQL, graph – Neoj4, RDF/OWL stores.
- Efficiency requires
  - Application-specific representations
  - Store only data needed by the application
  - Objective (shared) representation of the domain is not the purpose – multiple data models for the same domain to accommodate different business applications

Ontology - Purpose

- Objectivity of representation of reality
- Commonly used representation is graph, it is supported by RDF-based semantic technologies
- Objective (shared) representation of the domain
    - **one authoritative ontology for the domain of reality meant for re-use**
- **Storing vast volumes of data is not the purpose**

RESO®

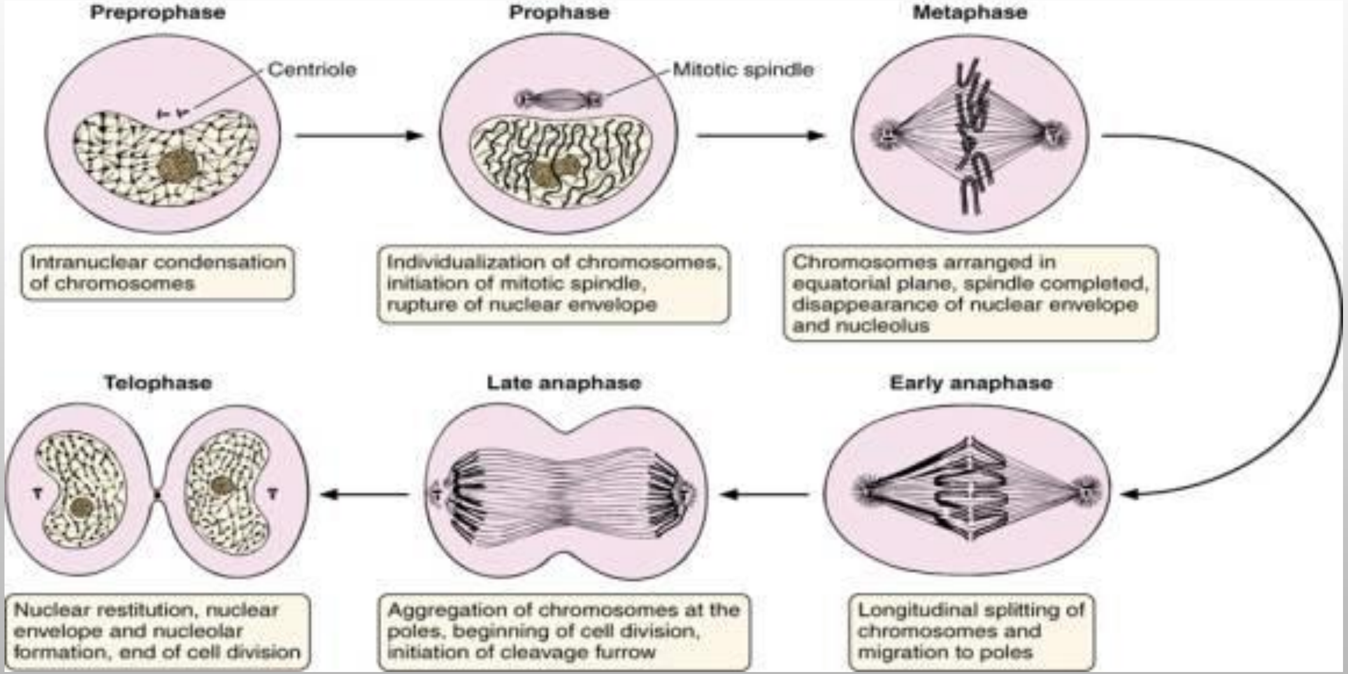
2018 Spring Technology Summit

Ontology - Organization

- Each type appears only once in the ontology hierarchy.
- The ontology view of reality is synoptic – it represents in non-redundant fashion an entire hierarchy of types at different levels of generality. Each term is associated in an intelligible way with its subsuming and subsumed terms (and thus with the ancestor and descendant types) in the hierarchy of more and less general
- Representation is more flexible, changes are easier to make, and changes are not as disruptive

2018 Spring Technology Summit

Knowledge Designed for Human Understanding

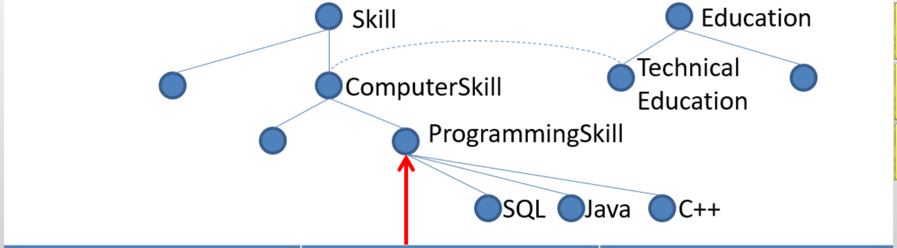


Knowledge Designed for Machine Understanding

MKVSDRRKFEKANFDEFESALNNKNDLVHCPSITLFESIPTEVRSFY
EDEKSGLIKVVKFRTGAMDRKRSFEKVVISVMVGKNVKKFLTFVED
EPDFQGGPISKYLIPKKINLMVYTLFQVHTLKFNRKDYDTLSLFYLNR
GYYNELSFRVLERCHEIASARPNDSSTMRTFTDFVSGAPIVRSLQK
STIRKYGYNLAPYMFLLLHVDELSIFSAYQASLPGEKKVDTERLKRD
LCPRKPIEIKYFSQICNDMMNKKDRLGDILHIILRACALNFGAGPRGG
AGDEEDRSITNEEPIIPSVDEHGLKVCKLRSPNTPRRLRKTLDAVKA
LLVSSCACTARDLDIFDDNNGVAMWKWIKILYHEVAQETTLKDSYRIT
LVPSSDGISLLAFAGPQRNVYVDDTTRRIQLYTDYNKNGSSEPRLKT
LDGLTSDYVFYFVTVLRQMQICALGNSYDAFNHDPWMDVVGFEDP
NQVTNRDISRIVLYSYMFLNTAKGCLVEYATFRQYMRELPKNAPQKL
NFREMRQGLIALGRHCVGSRFETDLYESATSELMANHSVQTGRNIY
GVDFSLTSVSGTTATLLQERASERWIQWLGLESDYHCSFSSTRNAE
DVDISRIVLYSYMFLNTAKGCLVEYATFRQYMRELPKNAPQKLNFRE
MRQGLIALGRHCVGSRFETDLYESATSELMANHSVQTGRNIYGVDF
SLTSVSGTTATLLQERASERWIQWLGLESDYHCSFSSTRNAEDV

Semantic Enhancement of Data Models by Ontology

- Semantic Enhancement (SE) is realized with the help of ontologies that are used to explicate data models and annotate data instances
  - Vocabulary of ontologies used for explications and annotations provides agile horizontal integration
  - Ontologies, by virtue of their nature and organization, provide semantic enhancement of data



PersonID	Name	Description
111	Java	Programming
222	SQL	Database

The Meaning of 'Enhancement'

- Semantic enhancement/enrichment of data = arm's length approach (no change to data) – through simple explication we associate an entire knowledge system with a database field
    - enables analytics to process data, e.g. about computer skills, "vertically" along the Skill hierarchy, as well as "horizontally" via relations between Skill and Education.
    - and further… while data in the database does not change, its analysis can be richer and richer as our understanding of the reality changes
- For this richness to be leveraged by different communities, persons, and applications it needs to have the properties mentioned above and be constructed in accordance with the principles of the SE (see References)

Semantic Enhancement and Data Integration

- Traditional integration approaches involve creation of a new model in
  - A new physical store (data warehouse)
  - Another data store – rigid (potential data silo), interoperable with other stores
  - Querying the data sources via it
    - Fragile
  - Both entail loss and or distortion of data and semantics, and provide only 'local' integration (do not lead to interoperability with other sources)

- SE of a store
  - Does not require data reorganization and creation of another store
  - Changes to it are non-intrusive
  - Leads to integration of the store with other stores, enhanced previously or in the future

2018 Spring Technology Summit

LIKQ Demo

RESO®

2018 Spring Technology Summit

The Languages of Ontology, Knowledge and Reasoning

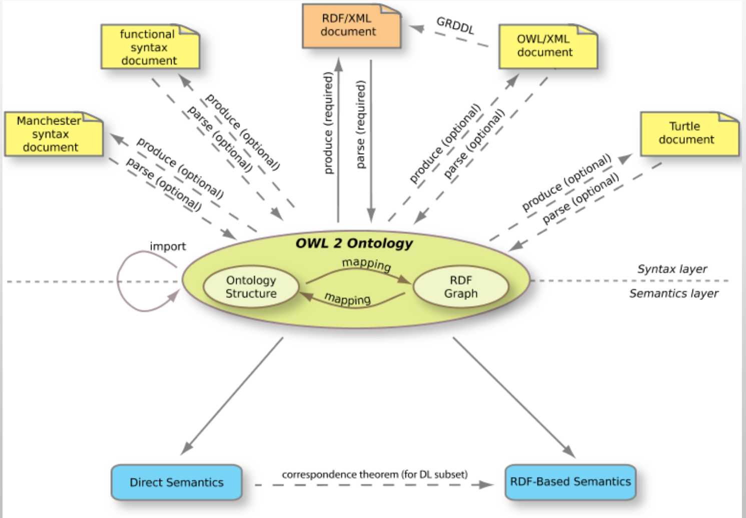
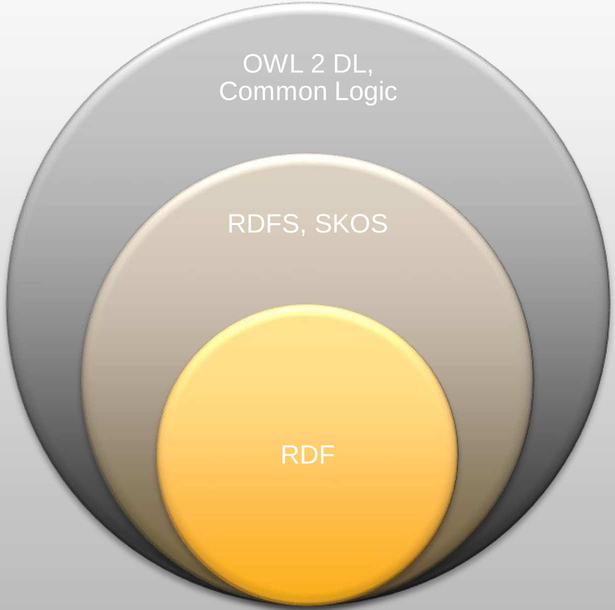


Figure 1. The Structure of OWL 2



2018 Spring Technology Summit

The Languages of Ontology, Knowledge and Reasoning

OWL Language

- Three species of OWL
  - *OWL Full* is union of OWL syntax and RDF
  - *OWL DL* restricted to FOL fragment (≅ DAML+OIL)
  - *OWL Lite* is "simpler" subset of OWL DL
- Semantic layering
  - OWL DL ≅ OWL full within DL fragment
- OWL DL based on SHIQ Description Logic
- OWL DL Benefits from many years of DL research
  - Well defined semantics
  - Formal properties well understood (complexity, decidability)
  - Known reasoning algorithms
  - Implemented systems (highly optimised)

```
<owl:Class rdf:about="&snap;Continuant">
        <rdfs:subClassOf rdf:resource="&bfo;Entity"/>
        <owl:equivalentClass>
                <owl:Class>
                        <owl:unionOf rdf:parseType="Collection">
                                <owl:Class rdf:about="&snap;DependentContinuant"/>
                                <owl:Class rdf:about="&snap;IndependentContinuant"/>
                                <owl:Class rdf:about="&snap;SpatialRegion"/>
                        </owl:unionOf>
                </owl:Class>
        </owl:equivalentClass>
        <owl:disjointWith rdf:resource="&span;Occurrent"/>
        <rdfs:label>continuant</rdfs:label>
        <skos:prefLabel>continuant</skos:prefLabel>
        <skos:definition>An entity [bfo:Entity] that exists in full at any time in which it exists at all, persists through
        <skos:example>a heart</skos:example>
        <skos:example>a person</skos:example>
        <skos:example>the color of a tomato</skos:example>
        <skos:example>the mass of a cloud</skos:example>
        <skos:example>a symphony orchestra</skos:example>
        <skos:example>the disposition of blood to coagulate</skos:example>
        <skos:example>the lawn and atmosphere in front of our building</skos:example>
        <skos:altLabel>endurant</skos:altLabel>
</owl:Class>
```

Reasoning with Description Logics

Syntax	Semantics	Comment
A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$	atomic concept
R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$	atomic role
$\top$	$\Delta^{\mathcal{I}}$	top (most general) concept
$\bot$	$\emptyset$	bottom (most specific) concept
$\neg A$	$\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$	atomic negation
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	intersection
$\forall R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in R^{\mathcal{I}} \Rightarrow b \in C^{\mathcal{I}}\}$	value restriction
$\exists R.\top$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in R^{\mathcal{I}}\}$	limited existential quantification

AL (attributive language) logic syntax and semantics

Name	Syntax	Semantics	Comment
$\mathcal{U}$	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	union of two concepts
$\mathcal{E}$	$\exists R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$	full quantification
$\mathcal{N}$	$\geqslant nR$ $\leqslant nR$	$\{a \in \Delta^{\mathcal{I}} \mid \; \lvert\{b \mid (a,b) \in R^{\mathcal{I}}\}\rvert \geq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \; \lvert\{b \mid (a,b) \in R^{\mathcal{I}}\}\rvert \leq n\}$	number restriction
$\mathcal{C}$	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$	negation of arbitrary concept

Examples of AL logic basic extensions

Some further extensions of *ALC* logic that will be of interest for us are as follows.

- *S* - role transitivity *Trans(R)* (asserting that role is transitive)
- *H* - role hierarchy $R \subseteq S$ (asserting hierarchy of roles)
- *I* - role inverse R^- (creating inverse role)
- *F* - functionality $\leqslant 1\ R$ (functional role in concept creation)
- *O* - nominals $\{a_1, \ldots, a_n\}$ (concept declared by enumeration)

Abstract Syntax	DL Syntax	Semantics
Descriptions (C)		
A (URI Reference)	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
`owl:Thing`	$\top$	$\texttt{owl : Thing}^{\mathcal{I}} = \Delta^{\mathcal{I}}$
`owl:Nothing`	$\bot$	$\texttt{owl : Nothing}^{\mathcal{I}} = \emptyset$
`intersectionOf`$(C_1\ C_2 \ldots)$	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
`unionOf`$(C_1\ C_2\ \ldots)$	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
`complementOf`(C)	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
`oneOf`$(o_1 \ldots)$	$\{o_1, \ldots\}$	$\{o_1^{\mathcal{I}}, \ldots\}$
`restriction`$(R$ `someValuesFrom`$(C))$	$\exists R.C$	$\{x \mid \exists y\ (x,y) \in R^{\mathcal{I}}$
`restriction`$(R$ `allValuesFrom`$(C))$	$\forall R.C$	$\{x \mid \forall y\ (x,y) \in R^{\mathcal{I}}$
`restriction`$(R$ `hasValue`$(o))$	$R : o$	$\{x \mid (x, o^{\mathcal{I}}) \in R^{\mathcal{I}}\}$
`restriction`$(R$ `minCardinality`$(n))$	$\geqslant nR$	$\{a \in \Delta^{\mathcal{I}} \mid \; \lvert\{b \mid (a,b)$
`restriction`$(R$ `maxCardinality`$(n))$	$\leqslant nR$	$\{a \in \Delta^{\mathcal{I}} \mid \; \lvert\{b \mid (a,b)$
`restriction`$(U$ `someValuesFrom`$(D))$	$\exists U.D$	$\{x \mid \exists y\ (x,y) \in U^{\mathcal{I}}$
`restriction`$(U$ `allValuesFrom`$(D))$	$\forall U.D$	$\{x \mid \forall y\ (x,y) \in U^{\mathcal{I}}$
`restriction`$(U$ `hasValue`$(v))$	$U : v$	$\{x \mid (x, v^{\mathcal{I}}) \in U^{\mathcal{I}}\}$
`restriction`$(U$ `minCardinality`$(n))$	$\geqslant nU$	$\{a \in \Delta^{\mathcal{I}} \mid \; \lvert\{b \mid (a,b)$
`restriction`$(U$ `maxCardinality`$(n))$	$\leqslant nU$	$\{a \in \Delta^{\mathcal{I}} \mid \; \lvert\{b \mid (a,b)$
Data Ranges (D)		
D (URI reference)	D	$D^{\mathcal{D}} \subseteq \Delta_{\mathcal{D}}^{\mathcal{I}}$
`oneOf`$(v_1 \ldots,)$	$\{v_1 \ldots,\}$	$\{v_1^{\mathcal{I}} \ldots,\}$
Object Properties (R)		
R (URI reference)	R	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
	R^-	$(R^{\mathcal{I}})^-$
Datatype Properties (U)		
U (URI reference)	U	$U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_{\mathcal{D}}^{\mathcal{I}}$
Individuals (o)		
o (URI reference)	o	$o^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
Data Values (v)		
v (RDF literal)	v	$v^{\mathcal{D}}$

Abstract Syntax	DL Syntax	Semantics
Classes		
`Class`$(A$ `partial` $C_1 \ldots C_n)$	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} \subseteq C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
`Class`$(A$ `complete` $C_1 \ldots C_n)$	$A \equiv C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} = C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
`EnuneratedClass`$(A\ o_1 \ldots o_n)$	$A \equiv \{o_1, \ldots, o_n\}$	$A^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots, o_n^{\mathcal{I}}\}$
`SubClassOf`$(C_1\ C_2)$	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
`EquivalentClasses`$(C_1 \ldots C_n)$	$C_1 \equiv \ldots \equiv C_n$	$C_1^{\mathcal{I}} = \ldots = C_n^{\mathcal{I}}$
`DisjointClasses`$(C_1 \ldots C_n)$	$C_i \sqcap C_j = \bot, i \neq j$	$C_i^{\mathcal{I}} \cap C_j^{\mathcal{I}} = \emptyset, i \neq j$
`Datatype`(D)		$D^{\subset}\Delta_{\mathcal{D}}^{\mathcal{I}}$
Datatype Properties		
`DatatypeProperty(`		
U `super`$(U_1) \ldots$ `super`(U_n)	$U \sqsubseteq U_i$	$U^{\mathcal{I}} \subseteq U_i^{\mathcal{I}}$
`domain`$(C_1) \ldots$ `domain`(C_m)	$\geqslant 1\, U \sqsubseteq C_i$	$U^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta_{\mathcal{D}}^{\mathcal{I}}$
`range`$(D_1) \ldots$ `range`(D_l)	$\top \sqsubseteq \forall U.D_i$	$U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times D_i^{\mathcal{I}}$
`[Functional])`	$\top \sqsubseteq \leqslant 1U$	U_i is functional
`SubPropertyOf`$(U_1\ U_2)$	$U_1 \sqsubseteq U_2$	$U_1^{\mathcal{I}} \subseteq U_2^{\mathcal{I}}$
`EquivalentProperties`$(U_1 \ldots U_n)$	$U_1 \equiv \ldots \equiv U_n$	$U_1^{\mathcal{I}} = \ldots = U_n^{\mathcal{I}}$
Object Properties		
`ObjectProperty(`		
R `super`$(R_1) \ldots$ `super`(R_n)	$R \sqsubseteq R_i$	$R^{\mathcal{I}} \subseteq R_i^{\mathcal{I}}$
`domain`$(C_1) \ldots$ `domain`(C_m)	$\geqslant 1\, R \sqsubseteq C_i$	$R^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta_{\mathcal{D}}^{\mathcal{I}}$
`range`$(C_1) \ldots$ `range`(C_l)	$\top \sqsubseteq \forall R.C_i$	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C_i^{\mathcal{I}}$
`[inverseOf`(R_0)`]`	$R \equiv (R_0^-)$	$R^{\mathcal{I}} = (R_0^{\mathcal{I}})^-$
`[Symmetric]`	$R \equiv (R^-)$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^-$
`[Functional]`	$\top \sqsubseteq \leqslant 1R$	$R^{\mathcal{I}}$ is functional
`[InverseFunctional]`	$\top \sqsubseteq \leqslant 1R^-$	$(R^{\mathcal{I}})^-$ is functional
`[Transitive])`	$Tr(R)$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^+$
`SubPropertyOf`$(R_1\ R_2)$	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$
`EquivalentProperties`$(R_1 \ldots R_n)$	$R_1 \equiv \ldots \equiv R_n$	$R_1^{\mathcal{I}} = \ldots = R_n^{\mathcal{I}}$
Annotation		
`AnnotationProperty`(S)		
Individuals		
`Individual(`		
o `type`$(C_1) \ldots$ `type`(C_n)	$o \in C_i$	$o^{\mathcal{I}} \in C_i^{\mathcal{I}}$
`value`$(R_1\ o_1) \ldots$ `value`$(R_n\ o_n)$	$\{o, o_i\} \in R_i$	$\{o^{\mathcal{I}}, o_i^{\mathcal{I}}\} \in R_i^{\mathcal{I}}$
`value`$(U_1\ v_1) \ldots$ `value`$(U_n\ v_n))$	$\{o, v_i\} \in U_i$	$\{o^{\mathcal{I}}, v_i^{\mathcal{I}}\} \in U_i^{\mathcal{I}}$
`SameIndividual`$(o_1 \ldots o_n)$	$o_1 = \ldots = o_n$	$o_1^{\mathcal{I}} = \ldots = o_n^{\mathcal{I}}$
`DifferentIndividual`$(o_1 \ldots o_n)$	$o_i \neq o_j, i \neq j$	$o_i^{\mathcal{I}} \neq o_j^{\mathcal{I}}, i \neq j$

RESO®

2018 Spring Technology Summit

Reasoning with Description Logics

Complexity of reasoning in Description Logics

Note: the information here is (always) incomplete and updated often

Base description logic: $\mathcal{A}$ttributive $\mathcal{L}$anguage with $\mathcal{C}$omplements

$\mathcal{ALC}$ ::= $\bot$ | $\top$ | A | $\neg C$ | $C \sqcap D$ | $C \sqcup D$ | $\exists R.C$ | $\forall R.C$

Concept constructors:

- ☐ $\mathcal{F}$ – functionality[2]: $(\leq 1\ R)$
- ☐ $\mathcal{N}$ – (unqualified) number restrictions: $(\geq n\ R)$, $(\leq n\ R)$
- ☐ $\mathcal{Q}$ – qualified number restrictions: $(\geq n\ R.C)$, $(\leq n\ R.C)$
- ☐ $\mathcal{O}$ – nominals: $\{a\}$ or $\{a_1, \ldots, a_n\}$ ("one-of")
- ☐ μ – least fixpoint operator: $\mu X.C$

Forbid complex roles[5] in number restrictions[6]

Role constructors: trans reg

- ☐ $\mathcal{I}$ – role inverse: R^-
- ☐ $\cap$ – role intersection[3]: $R \cap S$
- ☐ $\cup$ – role union: $R \cup S$
- ☐ $\neg$ – role complement: $\neg R$ full
- ☐ $\circ$ – role chain (composition): $R \circ S$
- ☐ $*$ – reflexive-transitive closure[4]: R^*
- ☐ id – concept identity: $id(C)$

TBox (concept axioms):

- ◉ empty TBox
- ○ acyclic TBox ($A \equiv C$, A is a concept name; no cycles)
- ○ general TBox ($C \sqsubseteq D$, for arbitrary concepts C and D)

RBox (role axioms): OWL-Lite OWL-DL OWL 1.1

- ☐ $\mathcal{S}$ – role transitivity: Tr(R)
- ☐ $\mathcal{H}$ – role hierarchy: $R \sqsubseteq S$
- ☐ $\mathcal{R}$ – complex role inclusions: $R \circ S \sqsubseteq R$, $R \circ S \sqsubseteq S$
- ☐ s – some additional features (click to see them)

Reset

You have selected a Description Logic: $\mathcal{ALC}$

Complexity[7] of reasoning problems[8]		
Concept satisfiability	**PSpace-complete**	• Hardness for $\mathcal{ALC}$: see [80]. • Upper bound for $\mathcal{ALCQ}$: see [12, Theorem 4.6].
ABox consistency	**PSpace-complete**	• Hardness follows from that for concept satisfiability. • Upper bound for $\mathcal{ALCQO}$: see [17, Appendix A].
Important properties of the Description Logic		
Finite model property	**Yes**	$\mathcal{ALC}$ is a notational variant of the multi-modal logic $\mathbf{K}_m$ (cf. [77]), for which the finite model property can be found in [4, Sect. 2.3].
Tree model property	**Yes**	$\mathcal{ALC}$ is a notational variant of the multi-modal logic $\mathbf{K}_m$ (cf. [77]), for which the tree model property can be found in [4, Proposition 2.15].

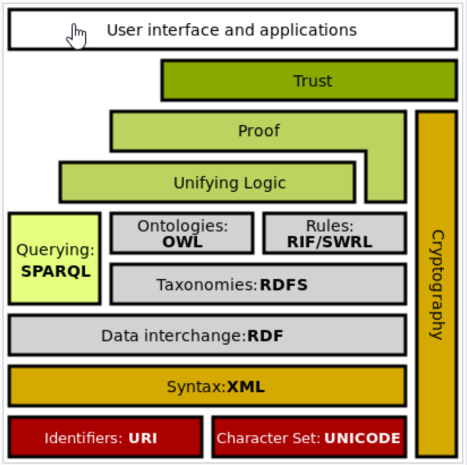
Maintained by: Evgeny Zolin
Please see the list of updates

Any comments are welcome: EZolin@cs.man.ac.uk

RESO®
2018 Spring Technology Summit

Reasoning with Description Logics

- The purpose of authoring ontologies is also reusing of knowledge. Once ontology is created for a domain, it should be (at least to some degree) reusable for other applications in the same domain. To simplify both ontology development and reuse, modular design is beneficial. The modular design uses inheritance of ontologies - upper ontologies describe general knowledge, and application ontologies describe knowledge for a particular application



Semantic Web Stack

2018 Spring Technology Summit

Questions?

Thank You,

Everyone!