JÖNKÖPING UNIVERSITY
School of Engineering

Ontology, Ontology Language and Ontology Development

TKRR25 Knowledge Representation and Reasoning

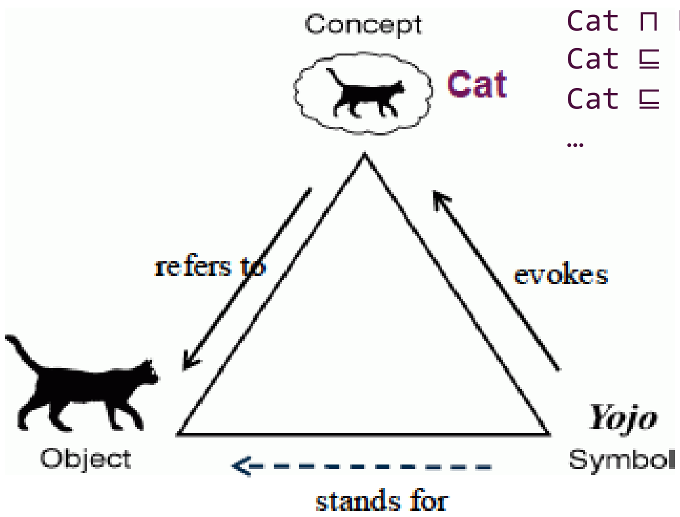
He Tan, Autumn 2025

Outline

- Introduction to Ontology
- Web Ontology Language (OWL)
- Ontology Development

A Brief History of Ontology (1)

- It was called "*first philosophy*" by Aristotle (384 BC – 322 BC) in his Book IV of his Metaphysics.
- Categorization is a fundamental cognitive process that people engage in regularly.



$\text{Cat} \sqcap \text{Dog} \equiv \bot$
$\text{Cat} \sqsubseteq \text{Pet}$
$\text{Cat} \sqsubseteq \text{Carnivore}$
…

Family Example

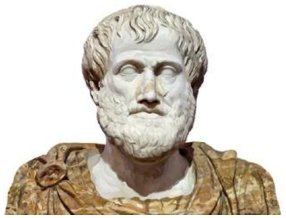
- John is a father, John is married to Mary, Alice is John's child
- . . .

individuals

- A man is a male person, a woman is a female person
- A husband is married to a woman, a wife is married to a man
- A parent is a person who has a child who is also a person
- A father is a man and a parent, a mother is a woman and a parent
- A grandparent is a person who has a child who is also a parent
- Every person is a grandchild
- A happy father has at most two children
- A happy mother has at least one daughter

family ontology

Aristotle's Ontology



- Aristotle proposed that **everything that exists (being)** can be classified according to **categories**.

Denomination	Question	Example
Substance	What is something	Man, horse
Quantity	How much/big is something?	Two inches long
Qualities	What are the features?	White, able to read
Relation	In what relationship is something (to something)?	double, half, bigger, daughter of, was born in
Location	Where is something?	On the table, in the swimming pool
Time	When is something?	Yesterday, in the future
Position/orientation	What orientation does something have?	standing, sitting
Having	What does something have?	Has shoes on his feet, is armed
Doing	What does something do?	Cut, burn
Experience	What experiences something?	Being cut, being burned

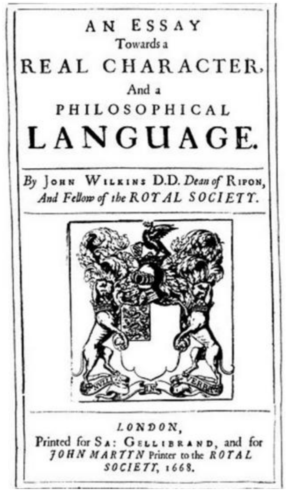
https://plato.stanford.edu/entries/aristotle-categories/

A Brief History of Ontology (2)

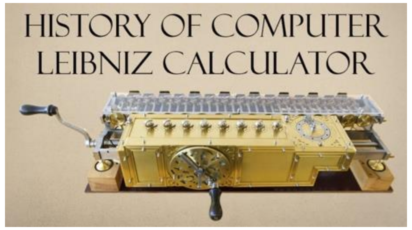
- Medieval scholastics: Aquinas, Scotus, Ockham, ... (1200 – 1600)
  - Integration of Aristotle with Christian theology
  - Distinction between its **essence** and its **existence** of a thing.
  - **Hierarchical structure** of beings
  - **Objective grounding**: concepts are not arbitrary labels but real structures of the world (being).

A Brief History of Ontology (3)

- John Wilkins (1614 – 1672)
  - tried to classify all concepts and entities into a **tree-like taxonomy.**
  - attempted to design a universal, rational classification of **all knowledge.**
  - proposed a constructed **symbolic language.**



- Gottfried Wilhelm Leibniz (1646 – 1716)
  - proposed a **universal logic language** that could represent knowledge with **symbolic precision**.
  - saw concepts as **sets of characteristics**.
  - proposed a mechanical procedure for **automatic reasoning**.



Deutsches Museum

A Brief History of Ontology (4)

- Immanuel Kant (1724 – 1804)
  - Ontologies are **conceptual models**, not not a passive mirror of reality.
  - Ontology connects perception to concepts - **symbol grounding.**

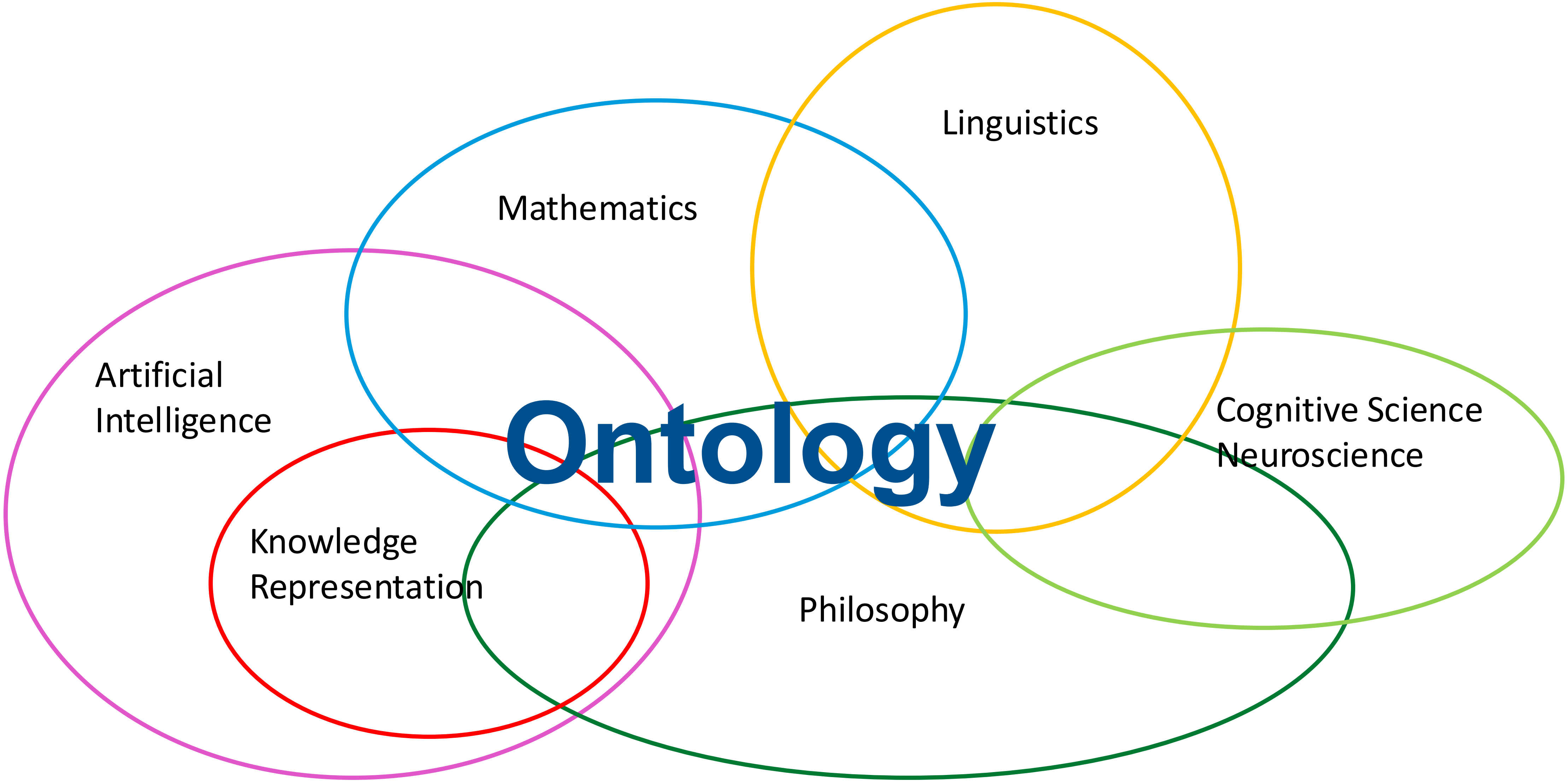
Categories are pure concepts of understanding.

A Brief History of Ontology (5)

- Franz Brentano (1838 - 1917) and His student, Edmund Husserl (1859 - 1938) brought ontology into the study of **conscious experience**.

A Brief History of Ontology (6)

- After ontology was introduced into CS and AI
  - Barry Smith, American mathematician, philosopher, and researcher in the field of **Applied Ontology**.
    - Co-developer of Basic Formal Ontology (BFO)
    - Ontologies should correspond to entities in reality, not just data models.
  - Tom Gruber, American computer scientist
    - defined ontology in AI terms: "***An ontology is an explicit specification of a conceptualization***." (1993)
    - pioneered **knowledge sharing** and **interoperability**
  - **…**


Linguistics
Mathematics
Artificial Intelligence
Ontology
Cognitive Science
Neuroscience
Knowledge Representation
Philosophy

Defintions in this Course

- An ontology is an **explicit**, **formal specification** of a **shared conceptualization**.

- A description logic knowledge base

Outline

- Introduction to Ontology
- Web Ontology Language (OWL)
- Ontology Development

Ontology Languages

- XOL (XML based Ontology Exchange Language)
- OBO (Ontology Language for Biological Ontologies)
- OML (Ontology Markup Language)
- KL-ONE (The Protege ODE, frame-based + logic)
- Ontolingua (First-order Logic (KIF))
- SHOE (Simple HTML Ontology Extension)
- OIL (Ontology Interchange Language)
- DAML+OIL (DARPA Agent Markup Language + OIL)

OWL - Web Ontology Language

- It was developed for the Semantic Web at that time.
  - It becomes the W3C recommendation in 2004.
  - Ontology become the foundation for linking, understanding, reasoning, and interoperability.

- It exploits results of 15+ years of DL research.

- It becomes the standardized ontology language today.

OWL - Web Ontology Language

- But it was designed as an ontology language for the **Semantic Web**
  - using **URIs** as identifiers
  - compatible with RDF and RDF Schema
  - using web-enabled syntaxes, e.g. XML/RDF
  - including many **technical aspects**, e.g. datatypes, annotations, versioning

Components of an Ontology in OWL

- An ontology is composed of:
  - ***Concepts***
    - `Student, Course, Mother`
  - ***Roles***
    - `teachesCourse, hasChild`
  - ***Individuals***
    - `Mary, Alice, John`
    - objects, names, places etc.
  - ***Constructors and Axioms***
    - state about how concepts or roles are related to each other
    - $\text{Mother} \equiv \text{Woman} \sqcap \exists \text{hasChild}.\text{Person}$
    - $\text{Cat} \sqsubseteq \text{Animal} \sqcap \forall \text{eats}.\text{Animal}$

Terminologies

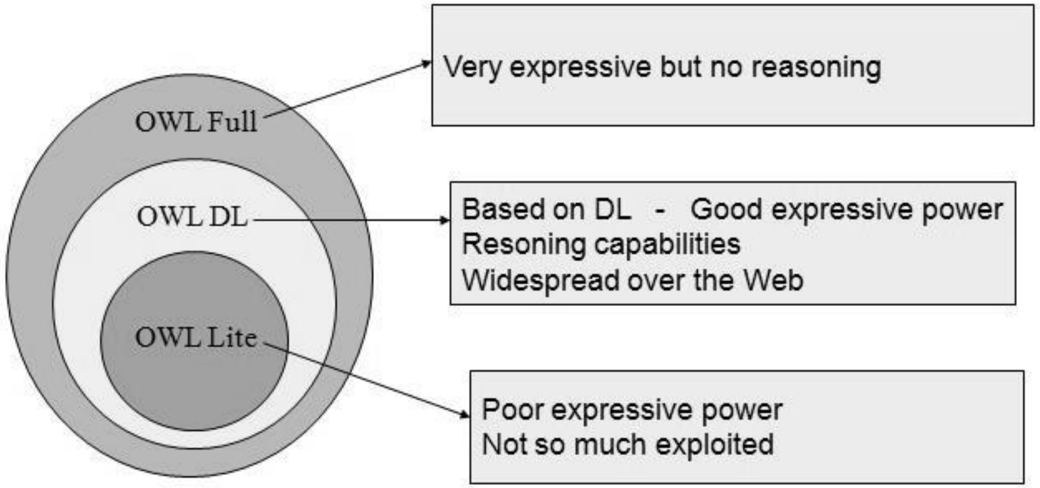
- The corresponding terms used in the OWL and DL.

OWL	DL
ontology	knowledge base
vocabulary	terminology/vocabulary/signature
class	concept
object property	role
axiom	axiom
individual	individual/instance
individual name	individual name

- OWL includes datatype properties, datatype, annotations, etc.

The OWL 1 Family of Languages

- OWL Lite
  - corresponds to the DL SHIF(D).
- OWL DL
  - corresponds to the DL SHOIN(D).
- OWL Full
  - not a description logic.
  - a very high expressiveness and all syntactic freedom of RDF (a graph-oriented formalism).



OWL 2

- OWL was extended into OWL 2 in 2009, as the W3C recommendation
  - Syntactic sugar
  - New constructs for properties
  - Extended datatypes
  - Punning
  - Extended annotations
  - Some innovations
  - Minor features

- Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., & Sattler, U. (2008). OWL 2: The next step for OWL. *Journal of Web Semantics*, *6*(4), 309-322.
- https://www.w3.org/TR/2012/REC-owl2-new-features-20121211/

OWL 2 DL and OWL 2 Full

- The OWL 2 DL is an extension of **SROIQ**.
  - https://www.w3.org/TR/owl2-direct-semantics/
  - Decidable
  - There are several production quality reasoners that cover the entire OWL 2 DL language, e.g. FaCT++, HermIT, Pellet, RacerPro
- OWL 2 Full is based on the RDF-compatible semantics
  - https://www.w3.org/TR/owl2-rdf-based-semantics/
  - Undecidable, no reasoner for OWL 2 Full

https://www.w3.org/TR/owl2-primer/#OWL_2_DL_and_OWL_2_Full

	Syntex	Semantics
$\mathcal{SROIQ}$ **constructors**		
universal concept	$\top$	Δ^I
bottom concept	$\perp$	$\perp$
atomic concept	A	A^I
intersection	$C \sqcap D$	$C^I \cap D^I$
union	$C \sqcup D$	$C^I \cup D^I$
complement	$\neg C$	$\Delta^I \backslash C^I$
universal restriction	$\forall R.C$	$\{a \in \Delta^I \mid \forall b.(a,b) \in R^I \rightarrow b \in C^I\}$
existential restriction	$\exists R.C$	$\{a \in \Delta^I \mid \exists b.(a,b) \in R^I \rightarrow b \in C^I\}$
at-least restriction	$\geq n\ \exists R.C$	$\{a \in \Delta^I \mid \#\{b \mid (a,b) \in R^I \rightarrow b \in C^I\} \geq n\}$
at-most restriction	$\leq n\ \exists R.C$	$\{a \in \Delta^I \mid \#\{b \mid (a,b) \in R^I \rightarrow b \in C^I\} \leq n\}$
local reflexivity	$\exists R.Self$	$\{a \mid (a,a) \in R^I\}$
nominal	$\{a\}$	$\{a^I\}$
atomic role	R	R^I
inverse role	R^-	$\{(a,b) \mid (b,a) \in R^I\}$
universal role	U	$\Delta^I \times \Delta^I$
individual name	a	a^I
$\mathcal{SROIQ}$ **axioms**		
concept assertion	$C(a)$	$a^I \in C^I$
role assertion	$R(a,b)$	$(a^I, b^I) \in R^I$
individual equality	$a \approx b$	$a^I = b^I$
individual inequality	$a \neq b$	$a^I \neq b^I$
concept inclusion	$C \sqsubseteq D$	$C^I \subseteq D^I$
concept eqivalence	$C \equiv D$	$C^I = D^I$
role inclusion	$R \sqsubseteq S$	$R^I \subseteq S^I$
role equivalence	$R \equiv S$	$R^I = S^I$
complex role inclusion	$R_1 \circ R_2 \sqsubseteq S$	$R_1 \circ R_2 \subseteq S$
role disjointness	$Disjoint(R,S)$	$R_1 \cap R_2 = \emptyset$

Table 1: $\mathcal{SROIQ}$ constructors and axioms

OWL 2 DL and $\mathcal{SROIQ}$

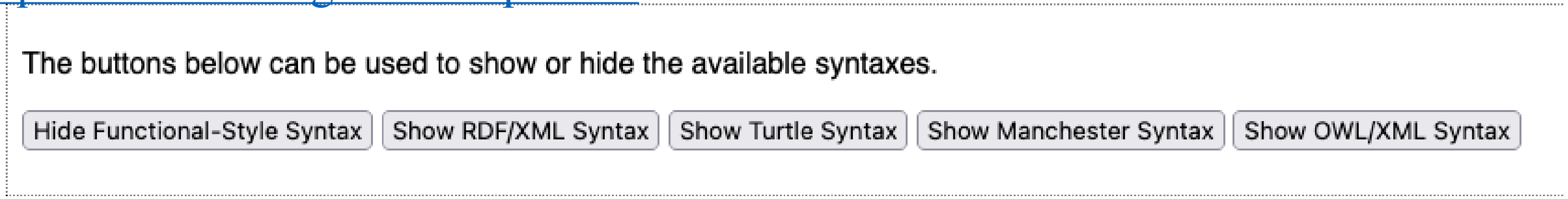
- Some differences between OWL 2 DL and $\mathcal{SROIQ}$
  - OWL provides a lot more operators that, though logically redundant, can be convenient as shortcuts for compound DL axioms, e.g., OWL has special constructs for specifying domain and range of a property
  - OWL includes some expressive features that are not included in *SROIQ,* e.g., OWL 2 DL supports for datatypes and datatype literals.
  - OWL has logical feature Keys.
  - OWL allows to importing ontology elements from one to another.
  - OWL supports annotations like comments in a programming language.

- Motik, B., & Horrocks, I. (2008, October). OWL datatypes: Design and implementation. In *International Semantic Web Conference* (pp. 307-322). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Parsia, B., Sattler, U., & Schneider, T. (2008). Easy Keys for OWL. In *OWLED* (Vol. 432).

OWL Syntaxes

- **The Manchester syntax**
- The Functional-Style syntax
- The RDF based syntax
  - XML-based serialization format (XML/RDF). It is mandatory to be supported by all OWL 2 tools.
  - Turtle (Terse RDF Triple Language)
- The OWL/XML syntax is an XML syntax for OWL defined by an XML schema

https://www.w3.org/TR/owl-primer/



Ontology Editor

- Protégé is a free, open-source ontology editor.
  - was developed at Stanford University.
  - supports OWL 2, RDF(S) and description logics reasoning.
  - integrated with reasoners (e.g., HermiT, Pellet, FaCT++) to check consistency.
  - DL Query tab allows hands-on exploration of class expressions.

protégé

Class Declaration

- Declares the classes (i.e., atomic concept/primitive concept)

```
Class: Person
```

the Manchester syntax

Class Disjointness

- State that membership in one class specifically excludes membership in another.

$\text{Woman} \sqcap \text{Man} \sqsubseteq \bot$ — DL syntax

```
DisjointClasses: Woman Man
```

the Manchester syntax

Class Constructors (Cont.')

- (Qualified) Exact Cardinality :

$\geq 3.hasChild.Parent \sqcap \leq 3.hasChild.Parent$

```
hasChild min 3 Parent
```

$\geq 5 \exists hasChild \sqcap \leq 5 \exists hasChild.Parent$

```
hasChild exactly 3 Parent
```

Class Constructors (Cont.')

- Describe classes of individuals that are related to one particular individual.

JohnsChildren ≡ hasParent.{John}

```
Class: JohnsChildren
   EquivalentTo: hasParent value John
```

Property Declaration

- Declare object properties (i.e., atomic role).

```
ObjectProperty: hasWife
   Domain: Man
   Range: Woman
```

- Specify the relationship between classes via object property

hasWife Domain: Man

$\exists hasWife.\ \top \sqsubseteq Man$

$\{\ a \in Man \mid \exists b.(a,\ b) \in haswife \rightarrow b \in \Delta^{I}\ \}$

hasWife Range :Woman

$\forall hasWife.Woman \sqsubseteq \top$

$\{\ a \in \Delta^{I} \mid \forall b.(a,\ b) \in hasWife \rightarrow b \in Woman\}$

Property Characteristics

```
InverseObjectProperties:  hasParent, hasChild
```

```
ObjectProperty: hasHusband
  Characteristics: Functional
```

```
ObjectProperty: hasHusband
   Characteristics: InverseFunctional
```

```
ObjectProperty: hasAncestor
   Characteristics: Transitive
```

```
DisjointProperties: hasParent, hasSpouse
```

```
ObjectProperty: hasSpouse
  Characteristics: Symmetric
```

```
ObjectProperty: hasChild
  Characteristics: Asymmetric
```

```
ObjectProperty: hasRelative
   Characteristics: Reflexive
```

```
ObjectProperty: parentOf
   Characteristics: Irreflexive
```

Individuals Assertion

- Specify how the individuals relate to other individuals.

  - (John, Mary) : hasWife

```
Individual: John
   Facts: hasWife Mary
```

  - (Bill, Mary): ¬ hasWife

```
Individual: Bill
   Facts: not hasWife Mary
```

Datatype Properties

the XML Schema datatypes

```
DataProperty: hasAge
    Domain: Person
    Range: xsd:nonNegativeInteger
```

- Relate individuals to data values

```
Individual: John
   Facts: hasAge "51"^^xsd:integer

Individual: Jack
   Facts: not hasAge "53"^^xsd:integer
```

Advanced used Datatypes

```
Datatype: personAge
   EquivalentTo: integer[<= 0 , >= 150]
```

```
Datatype: toddlerAge
    EquivalentTo: { 1, 2 }
```

```
Datatype: majorAge
    EquivalentTo: personAge and not minorAge
```

```
Class: Teenager
  SubClassOf: hasAge some integer[< 12 , >= 19]
```

Keys

- Assign a collection of properties as a key to a class expression.

```
Class: Person
HasKey: hasSSN
```

Annotation Properties

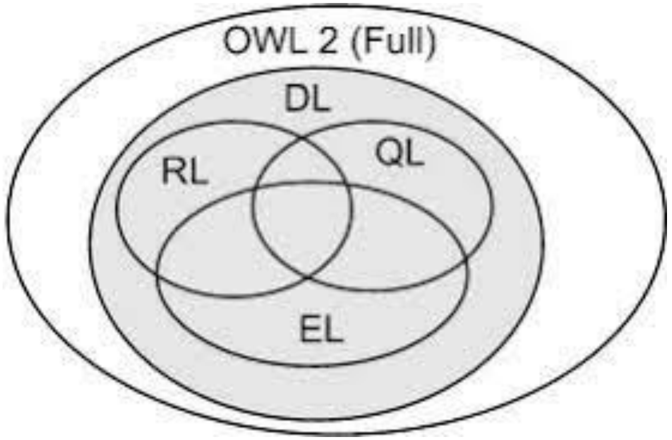
- provides additional (non-logical) information

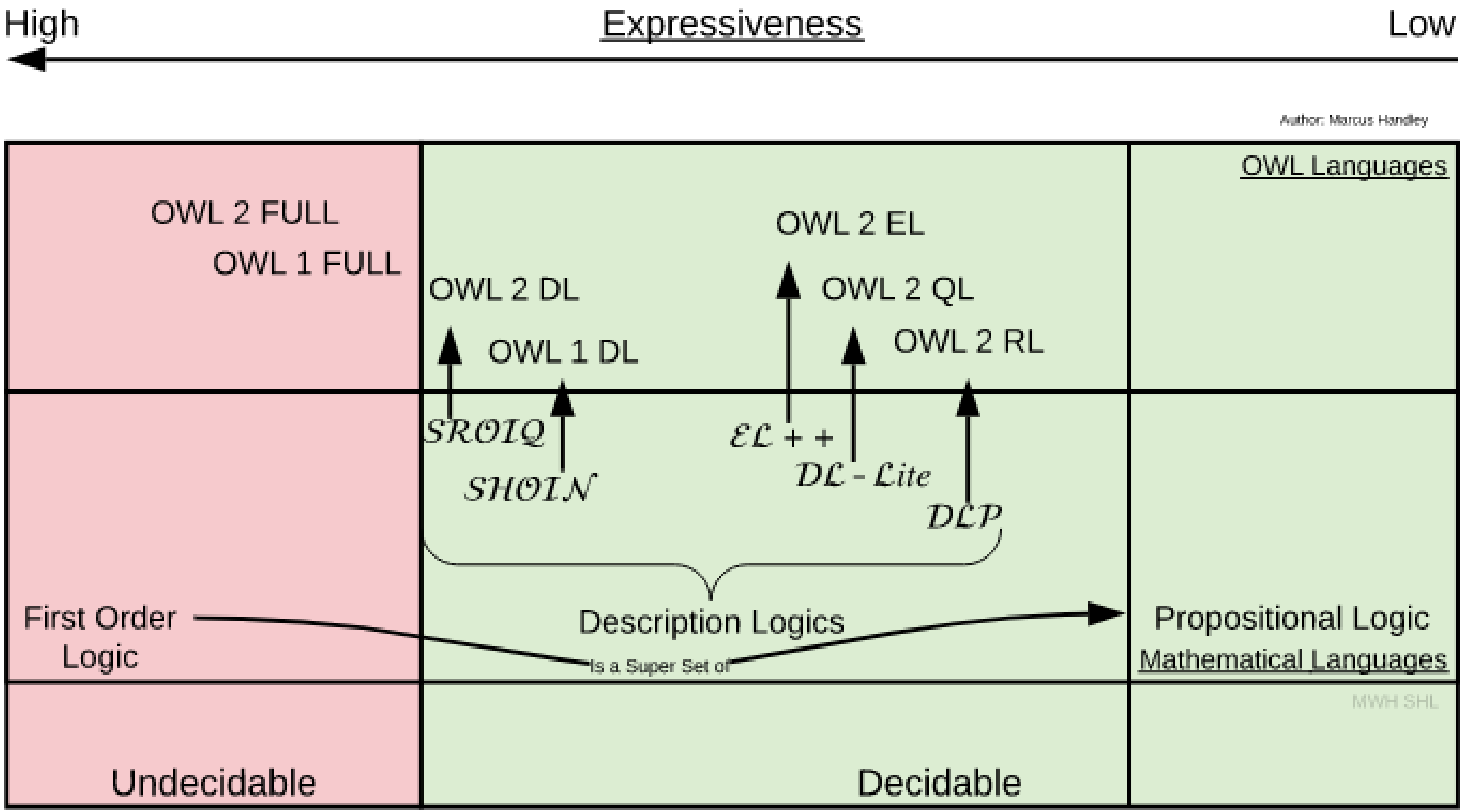
```
Class: Person
   Annotations: rdfs:comment "Represents the set of all people."
```

Three Profiles of OWL DL 2

- **OWL 2 EL** is particularly suitable for applications employing ontologies that define very large numbers of classes and/or properties
- **OWL 2 QL** is aimed at applications that use very large volumes of instance data, and where query answering is the most important reasoning task.
- **OWL 2 RL** is aimed at applications that require scalable reasoning without sacrificing too much expressive power.

https://www.w3.org/TR/owl2-profiles




High
Expressiveness
Low
Author: Marcus Handley
OWL Languages
OWL 2 FULL
OWL 1 FULL
OWL 2 EL
OWL 2 DL
OWL 2 QL
OWL 1 DL
OWL 2 RL
SROIQ
SHOIN
EL + +
DL - Lite
DLP
First Order Logic
Description Logics
Is a Super Set of
Propositional Logic
Mathematical Languages
MWH SHL
Undecidable
Decidable

OWL Ontology

- Starts with the namespaces
  - **namespace**: tells which vocabulary a name is from. It is to address the problem of naming conflicts.
  - **namespace prefix** declares a pair of the namespace reference and the namespace URI that is associated with this prefix.

default namespace refers to this vocabulary

```
Prefix: : <http://example.com/owl/families/>
Prefix: xsd: <http://www.w3.org/2001/XMLSchema#>
Prefix: owl: <http://www.w3.org/2002/07/owl#>
Prefix: rdfs: <http://www.w3.org/2000/01/rdf-schema#>

Prefix: otherOnt: <http://example.org/otherOntologies/families/>

Import <http://example.org/otherOntologies/families.owl>
```

How to Build an Ontology?

- It is not an easy task.

nature > nature genetics > commentary > article

Commentary | Published: May 2000

Gene Ontology: tool for the unification of biology

Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin & Gavin Sherlock

Nature Genetics 25, 25–29 (2000) | Cite this article

Genomic sequencing has made it clear that a large fraction of the genes specifying the core biological functions are shared by all eukaryotes. Knowledge of the biological role of such shared proteins in one organism can often be transferred to other organisms. The goal of the Gene Ontology Consortium is to produce a dynamic, controlled vocabulary that can be applied to all eukaryotes even as knowledge of gene and protein roles in cells is accumulating and changing. To this end, three independent ontologies accessible on the World-Wide Web (http://www.geneontology.org) are being constructed: biological process, molecular function and cellular component.

The Gene Ontology Resource: 20 years and still GOing strong

The Gene Ontology Consortium Author Notes

Nucleic Acids Research, Volume 47, Issue D1, 08 January 2019, Pages D330–D338, https://doi.org/10.1093/nar/gky1055
Published: 05 November 2018 **Article history ▾**

PDF | Split View | Cite | Permissions | Share ▾

Abstract

The Gene Ontology resource (GO; http://geneontology.org) provides structured, computable knowledge regarding the functions of genes and gene products. Founded in 1998, GO has become widely adopted in the life sciences, and its contents are under continual improvement, both in quantity and in quality. Here, we report the major developments of the GO resource during the past two years. Each monthly release of the GO resource is now packaged and given a unique identifier (DOI), enabling GO-based analyses on a specific release to be reproduced in the future. The molecular function ontology has been refactored

The Gene Ontology knowledgebase in 2023

The Gene Ontology Consortium, Suzi A Aleksander, James Balhoff, Seth Carbon, J Michael Cherry, Harold J Drabkin, Dustin Ebert, Marc Feuermann, Pascale Gaudet, Nomi L Harris ... Show more
Author Notes

Genetics, Volume 224, Issue 1, May 2023, iyad031, https://doi.org/10.1093/genetics/iyad031
Published: 03 March 2023 **Article history ▾**

PDF | Split View | Cite | Permissions | Share ▾

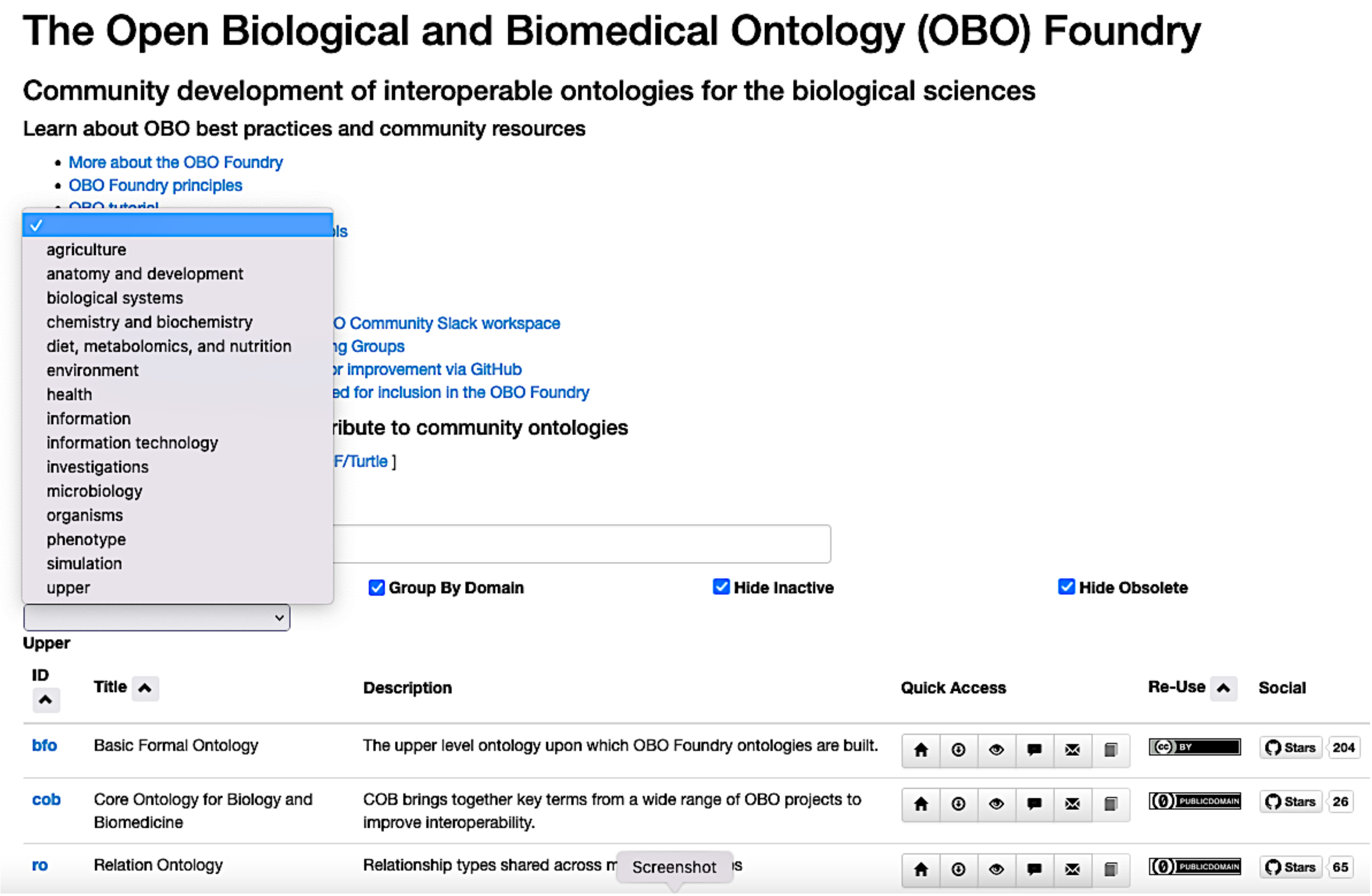
Abstract

The Gene Ontology (GO) knowledgebase (http://geneontology.org) is a comprehensive resource concerning the functions of genes and gene products (proteins and noncoding RNAs). GO annotations cover genes from organisms across the tree of life as well as viruses, though most gene function knowledge currently derives from experiments carried out in a relatively small number of model organisms. Here, we provide an updated overview of the GO

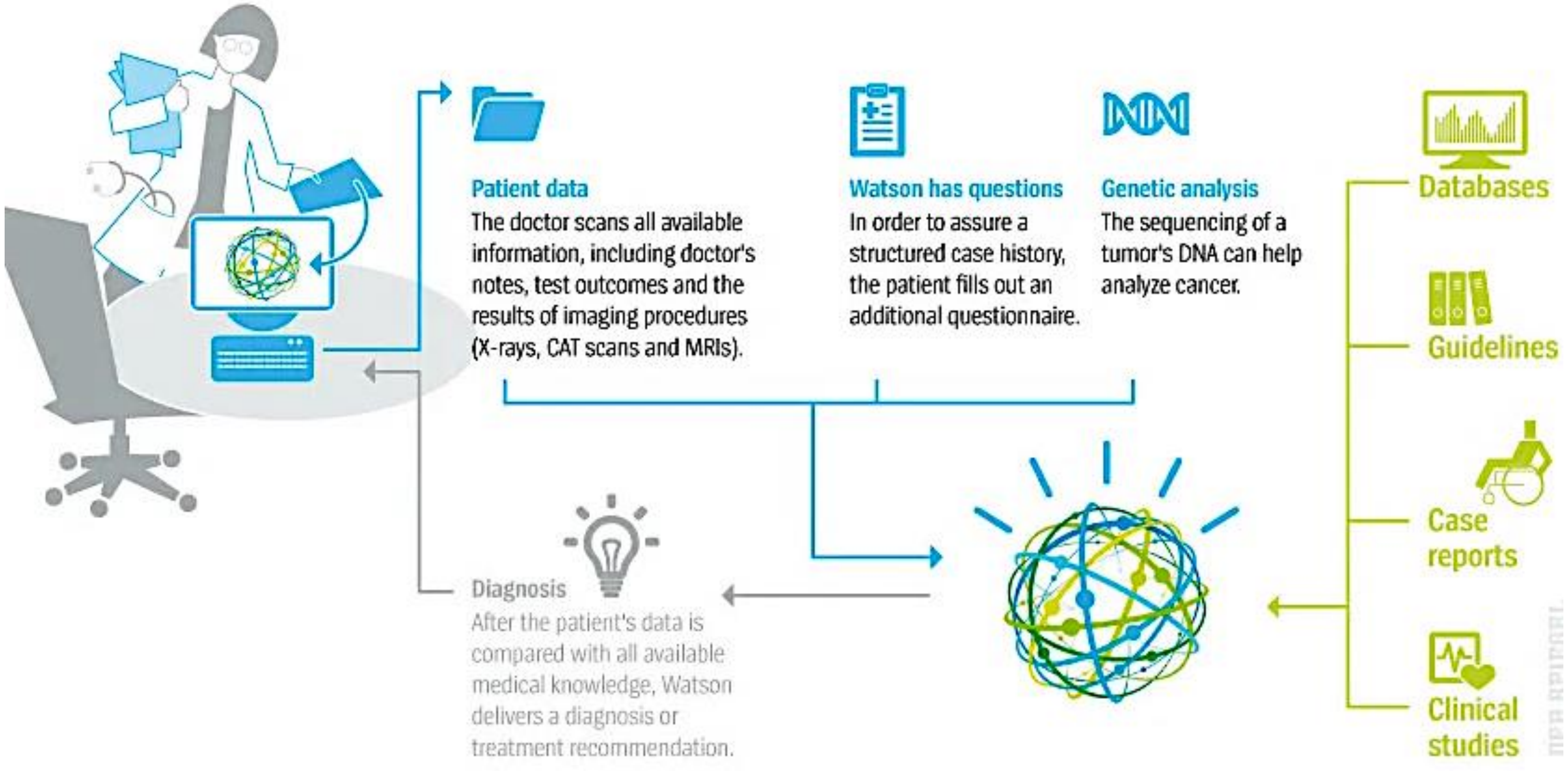
GENEONTOLOGY
Unifying Biology

Example Domain Ontologies

- https://obofoundry.org/



An Example – IBM Watson for Oncology



Memorial Sloan Kettering Cancer Center

IBM Watson

Outline

- Introduction to Ontology
- Web Ontology Language (OWL)
- Ontology Development

Ontology Building Methodologies

- Macro level methodologies
  - METHONTOLOGY (1997)
  - On-To-Knowledge (2001)
  - DILIGENT (2004)
  - The NeON Methodology (2008)
  - Ontology Design Pattern (2009)
  - Ontology Modularization (2009)
  - The eXtreme Design Methodology (2012)
- Micro-level methodologies
  - Ontology Development 101 (2001)
  - DiDOn (2012)

Ontology Editors

- It is like an IDE (integrated development environment) but for ontologies.
  - Protégé
  - TopBraid Composer
  - NeOn Toolkit (less active now)

Ontology Quality and Evaluation

- Completeness
- Accuracy
- Consistency
- Conciseness
- Clarity
- Extendibility
- Etc.

Ontology Tools

- Ontology development tools
- Ontology merge and alignment tools
- Ontology evaluation tools
- Ontology-based annotation tools
- Ontology storage and querying tools
- Ontology learning tools
- Ontology debugging and completion tools

Literature

- Guarino, N., Oberle, D., & Staab, S. (2009). What is an ontology?. *Handbook on ontologies*, 1-17.
- Smith, B. (2012). Ontology. In *The furniture of the world* (pp. 47-68). Brill.
- OWL 2 Web Ontology Language Primer (Second Edition): https://www.w3.org/TR/owl2-primer/
- G'abor N. (2007) Ontology Development. In: Studer R., Grimm S., Abecker A. (eds) Semantic Web Services. Springer, Berlin, Heidelberg.
- Suárez-Figueroa, M. C., Gómez-Pérez, A., & Villazón-Terrazas, B. (2009). How to write and use the ontology requirements specification document. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 966-982). Springer, Berlin, Heidelberg.
- "Ontology Development 101: A Guide to Creating Your First Ontology" by Natalya F. Noy and Deborah L. McGuinness

JÖNKÖPING UNIVERSITY

School of Engineering