MDE

FINAL TOPICS

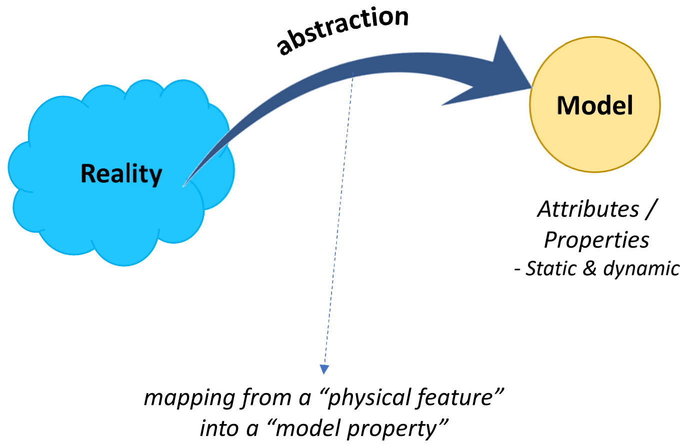
Luis M. Camarinha-Matos

NOVA University of Lisbon
lcm@fct.unl.pt or cam@uninova.pt

REVISITING THE BASICS

- "A model is an **abstract representation** of an environment, system, or entity in the physical, social, or logical world."

- The models are necessarily incomplete (intentionally ignoring non-relevant details), reducing their complexity to levels appropriate to the purposes for which they are intended.



Modeling language

- DB SQL
- Prolog
- Frames
- UML
- …
- …

Mechanisms:

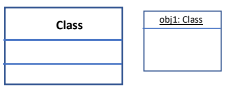
- Classification
- Generalization
- Aggregation or composition
- Partition

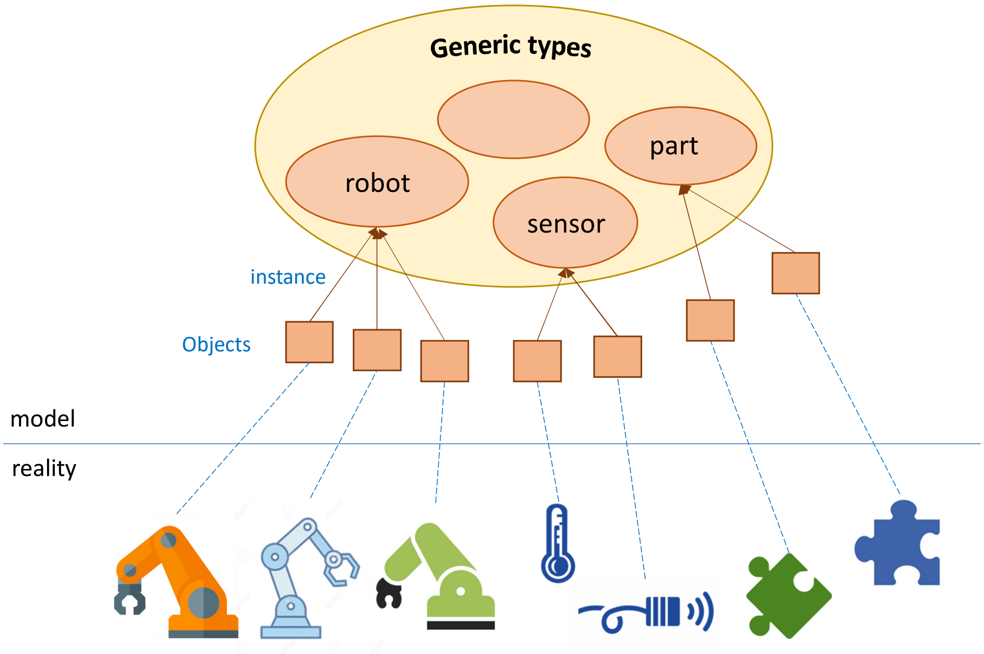
MECHANISMS: CLASSIFICATION

Objects defined as occurrences or **instances** of generic types

- **instance**
- **is-occurrence-of**

In UML:



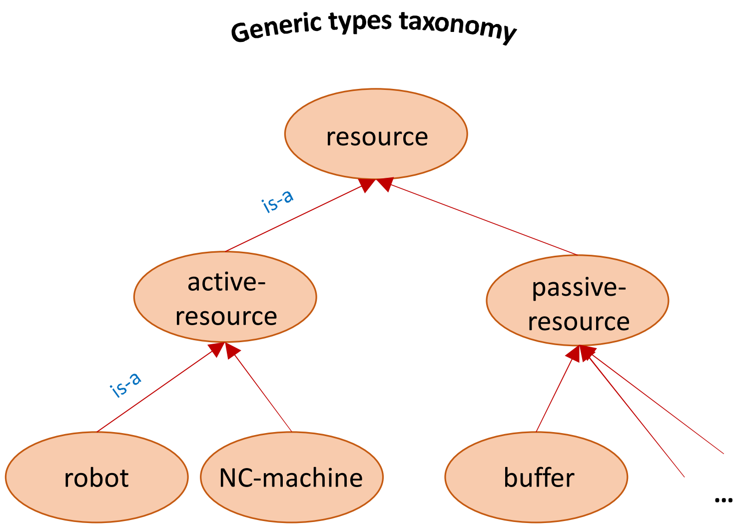


MECHANISMS: GENERALIZATION

One type is related to a more general one
=> **Taxonomy**

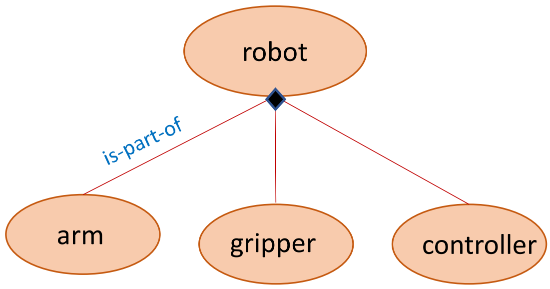
- **is-a**
- **ako**

In UML:



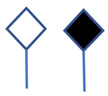
MECHANISMS: AGGREGATION OR COMPOSITION

Entities are composed of other (simpler) entities / components



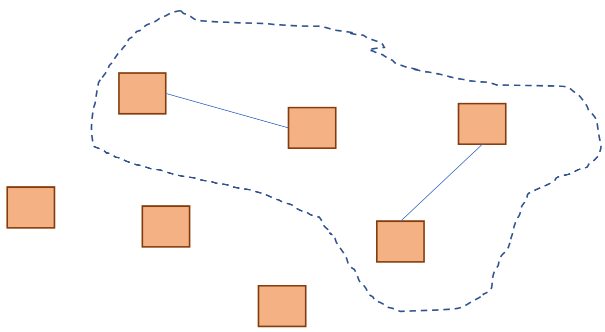
- **is-part-of**
- **has-component**

In UML:



MECHANISMS: PARTITION

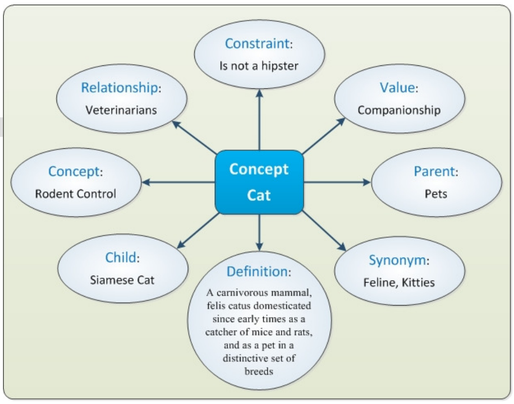
A new “class” can be identified through a relation between similar objects



- **is-married-to**
- **has-a-link-to**

ONTOLOGIES

ONTOLOGIES

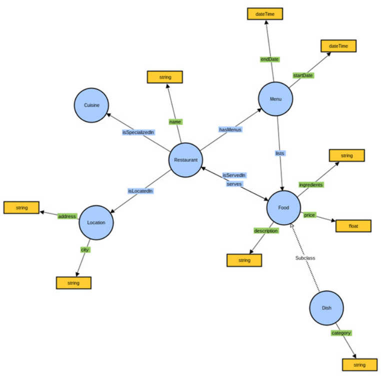


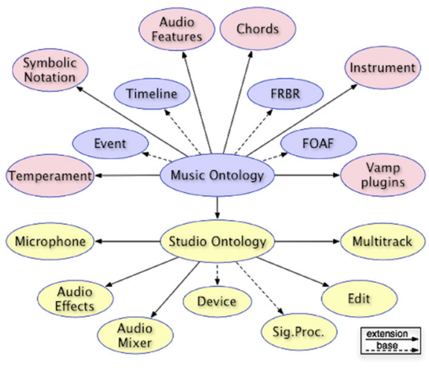
Ontology – formal explicit description of

- concepts in a domain of discourse [***classes*** or ***concepts***],
- properties of each concept describing various features and attributes of the concept [***slots***, or ***properties***], and
- restrictions on slots [***facets***].

Examples:

A class can have ***subclasses*** that represent concepts that are more specific than the superclass.





ONTOLOGIES …

In Philosophy, ontology means a systematic explanation of being, or the study of what exists.

More broadly, it studies **concepts** that directly relate to being, in particular becoming, existence, reality, as well as the basic categories of being and their relations. [Wikipedia]

In the framework of computer science, **ontology** represents a formal representation of knowledge as a set of concepts within a domain, and the relationships between these concepts.

''An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary''
Neches et al. 1991

Knowledge base – an ontology together with a set of individual instances of classes.

See also: http://www.cs.man.ac.uk/~stevensr/onto/node3.html

ONTOLOGIES …

An "ontology" describes the common words, concepts and relationships between concepts used to describe and represent an area of knowledge.

An ontology can range from a

- Taxonomy (knowledge with minimal hierarchy or a parent/child structure) to a …
- Thesaurus (words and synonyms) to a ….
- Conceptual Model (with more complex knowledge) to a…
- Logical Theory (with very rich, complex, consistent and meaningful knowledge).

A well-formed ontology is one that is expressed in a well-defined syntax that has a well-defined machine interpretation consistent with the above ontology definition

https://www.scss.tcd.ie/Owen.Conlan/CS7063/6_Introduction%20to%20Ontology.pdf

ONTOLOGIES …

Gruber, T. *A translation Approach to portable ontology specifications.* **Knowledge Acquisition**. Vol. 5. 1993. 199-220

"An ontology is a formal, explicit specification of a shared conceptualization"

Machine readable

Concepts, properties, relations, functions, constraints, axioms, are explicitly defined

Consensual knowledge

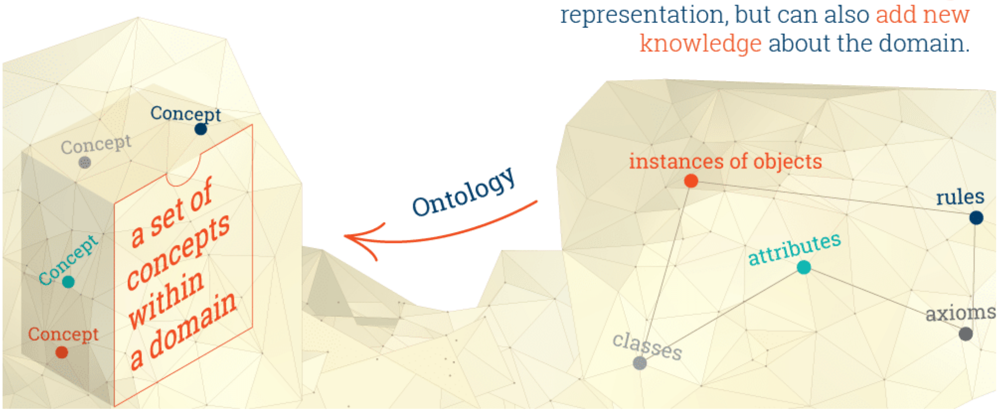
Abstract model and simplified view of some phenomenon in the world that we want to represent

axioms are used to constrain values for classes or instances

Studer, Benjamins, Fensel. Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering*. 25 (1998) 161-197

ONTOLOGIES …

Ontologies do not only introduce a sharable and reusable knowledge representation, but can also add new knowledge about the domain.



https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/

"An ontology is a **formal description** of knowledge as a set of concepts within a domain and the relationships that hold between them.

To enable such a description, we need to formally specify components such as individuals (instances of objects), classes, attributes and relations as well as restrictions, rules and axioms. As a result, ontologies do not only introduce a sharable and reusable knowledge representation but can also add new knowledge about the domain."

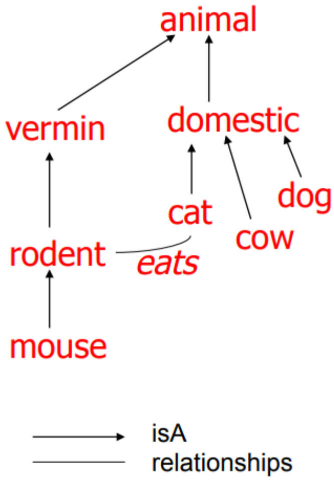
ONTOLOGIES …

Concepts (**class**, set, type, predicate)

- event, gene, gammaBurst, atrium, molecule, cat

Properties of concepts and *relationships* between them (**slot**)

- *Taxonomy*: generalisation ordering among concepts *isA*, *partOf*, *subProcess*
- *Relationship*, *Role* or *Attribute*: *functionOf*, *hasActivity* location, *eats*, size



Constraints or *axioms* on properties

- value: integer
- domain: cat
- cardinality: at most 1
- range: 0 <= X <= 100
- oligonucleiotides < 20 base pairs
- cows are larger than dogs
- cats cannot eat only vegetation
- cats and dogs are disjoint

https://www.scss.tcd.ie/Owen.Conlan/CS7063/6_Introduction%20to%20Ontology.pdf

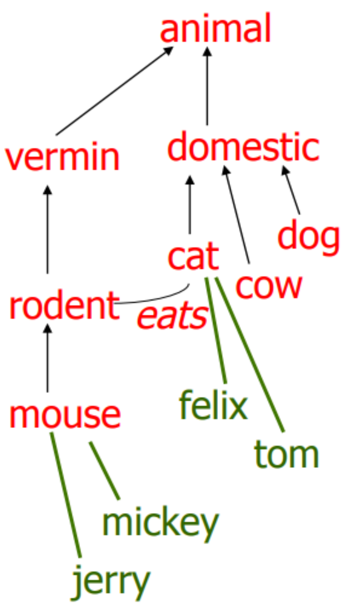
ONTOLOGY & KNOWLEDGE BASE

Individuals or *Instances*

- sulphur, trpA Gene, **felix**

Ontology **versus** *Knowledge Base*

- An *ontology* = concepts+properties+axioms +values
- A *knowledge base* = ontology+instances

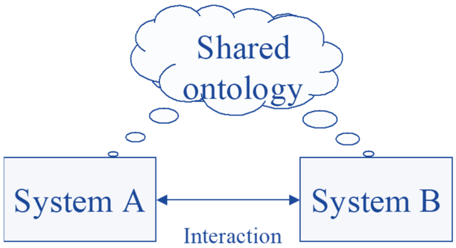


https://www.scss.tcd.ie/Owen.Conlan/CS7063/6_Introduction%20to%20Ontology.pdf

ONTOLOGIES: MOTIVATION

Why creating ontologies?

- To **share common understanding** of the structure of information among people or software components.



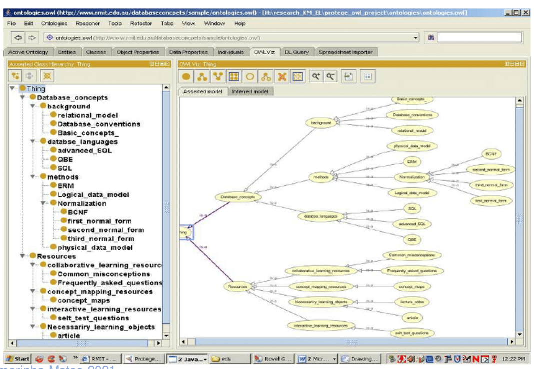
- To **enable reuse** of knowledge.
- To **make domain assumptions explicit**.
- To separate **domain knowledge** from the operational knowledge.
- To **analyze domain knowledge**.

ONTOLOGY DEVELOPMENT

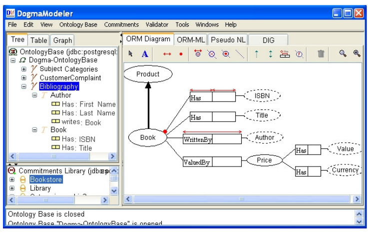
Developing an ontology includes:

- Defining classes in the ontology
- Arranging the classes in a taxonomic hierarchy
- Defining slots and describing allowed values for the slots
- Filling in the values for slots for instances

Tools to build ontologies



PROTÉGÉ
http://protege.stanford.edu



DOGMA
http://www.jarrar.info/Dogmamodeler/

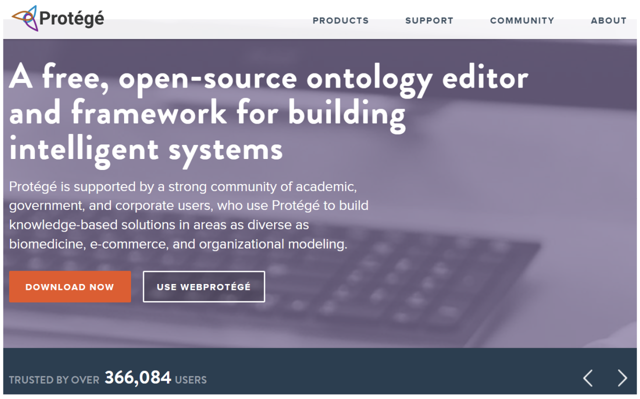
OWL tools
https://www.w3.org/OWL/

… and more

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

PROTÉGÉ

Protege is a free, open-source platform to construct domain models and knowledge-based applications with ontologies.



Stanford Center for Biomedical Informatics Research (BMIR)

https://protege.stanford.edu/

Protege supports the following ways of Ontology Engineering:
- Frame-based
- OWL (Web Ontology Language)

- Protege Frames editor: enables users to build and populate ontologies that are frame-based, in accordance with OKBC (Open Knowledge Base Connectivity Protocol).
  - Classes
  - Slots for properties and relationships
  - Instances for class

- Protege OWL editor:
  - Classes
  - Properties
  - Instances
  - Axioms

https://protege.stanford.edu/download/protege/old-releases/Protege%201.x/1.9/ [**Protége 2000** version]

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

ONTOLOGY DEVELOPMENT: STEPS

(Based on Noy & McGuinness, 2000)

STEP 1: Determine the domain and scope of the ontology

- What is the domain that the ontology will cover?
- For what we are going to use the ontology?
- For what types of questions the information in the ontology should provide answers?
- Who will use and maintain the ontology?

Domain: Factory automation

Purpose: Elaborate digital twin model

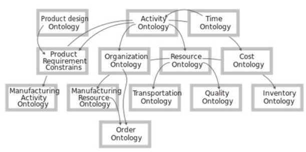
Questions: Status? Control?

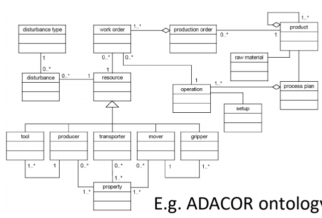
Users: Automation engineer

STEP 2: Consider reusing existing ontologies

Reusing existing ontologies may be a requirement if our system needs to interact with other applications (systems) that have already committed to particular ontologies or controlled vocabularies.

E.g. TOVE ontology

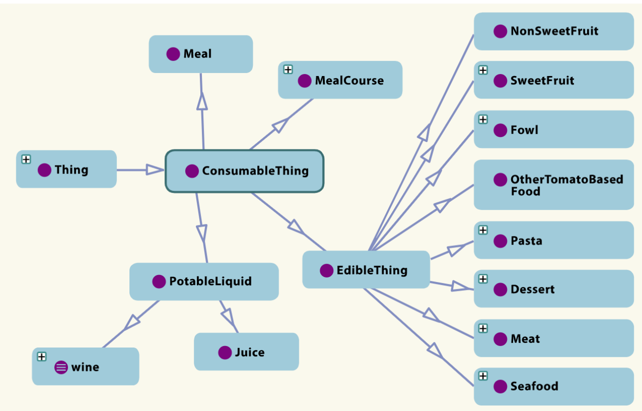




E.g. ADACOR ontology

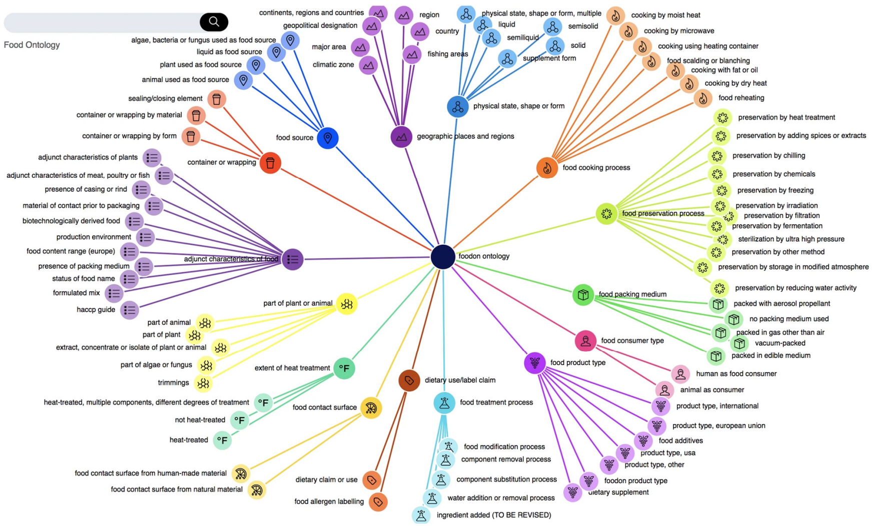
ONTOLOGY EXAMPLES

FOOD ontology



https://ceweb.br/livros/dados-abertos-conectados/capitulo-4/

FOODON ontology



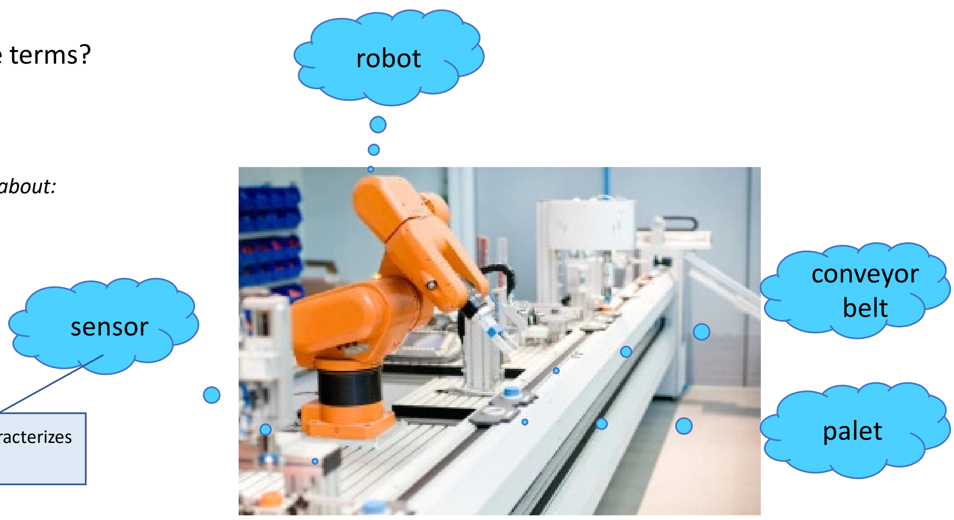
https://www.nature.com/articles/s41538-018-0032-6

ONTOLOGY DEVELOPMENT: STEPS

STEP 3: Enumerate important terms in the ontology

- What are the terms we would like to talk about?
- What properties do those terms have?
- What would we like to say about those terms?

Get a comprehensive list of terms without worrying about:
- *overlap between concepts they represent,*
- *relations among the terms,*
- *or any properties that the concepts may have,*
- *or whether the concepts are classes or slots.*



ONTOLOGY DEVELOPMENT: STEPS

STEP 4: Define the classes and the class hierarchy

- Top-down process – starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts.

- Bottom-up process – starts with the definition of the most specific classes, the leaves of the hierarchy, with subsequent grouping of these classes into more general concepts.

- Combined process – define the more salient concepts first and then generalize and specialize them.

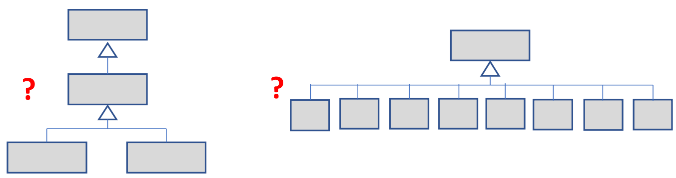
From the list created in Step 3, we select the terms that describe objects having independent existence

These terms will be classes in the ontology and will become anchors in the class hierarchy

- **"is-a" relation** – *a subclass of a class represents a concept that is a "kind of" the concept that the superclass represents*.
- **Transitivity** – *if B is a subclass of A and C is a subclass of B, then C is a subclass of A.*
- **Cycles** – *we should avoid cycles in the class hierarchy.*

STEP 4 ...

- **How many is too many and how few are too few**?
  - *If a class has only one direct subclass there might be a modeling problem or the ontology is not complete.*
  - *If there are more than a dozen (less?) subclasses for a given class then additional intermediate categories may be necessary.*



- **When to introduce a new class (or not)**
  - *Subclasses of a class usually*
    - *Have additional properties that the superclass does not have*
    - *Have restrictions different from those of the superclass*
    - *Participate in different relationships than the superclasses*
  - *However, sometimes it may be useful to create new classes even if they do not introduce any new properties*.

E.g. a conference room may have a property "booked by" that room does not

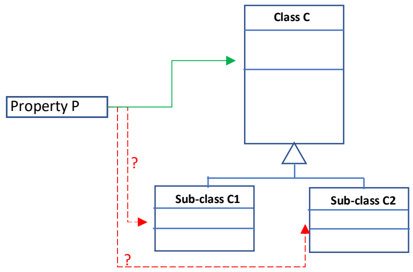
- **Multiple inheritance** – *in some systems a class can be a subclass of several classes.*

ONTOLOGY DEVELOPMENT: STEPS

STEP 5: Define the properties of classes – slots

- Once we have defined some of the classes, we must describe the **internal structure** of the concepts.
- Several types of object **properties** that can become slots:
  - “intrinsic” properties (e.g. *flavor* of a wine, *capacity* of a room)
  - “extrinsic” properties (e.g. wine’s *name / origin, room’s name*)
  - parts, if the object is structured (e.g. parts of a car, courses of a meal)
  - relationships to other individuals - relationships between individual members of the class and other items (e.g. wine and wine maker)
- A **subclass** of a class inherits the slots of that class.
- A slot should be attached at the **most general class** that can have that property.
  - Inverse slots

If both sub-class C1 and sub-class C2 have Property P, then it should be added to the super class C



ONTOLOGY DEVELOPMENT: STEPS

STEP 6: Define the facets of the slots

- **Slot cardinality** – how many values a slot can have
- **Slot-value type** – what types of values can fill in the slot
  - String
  - Number
  - Boolean
  - Enumerated
  - Instance … definition of relationships
  - …
  - Default values

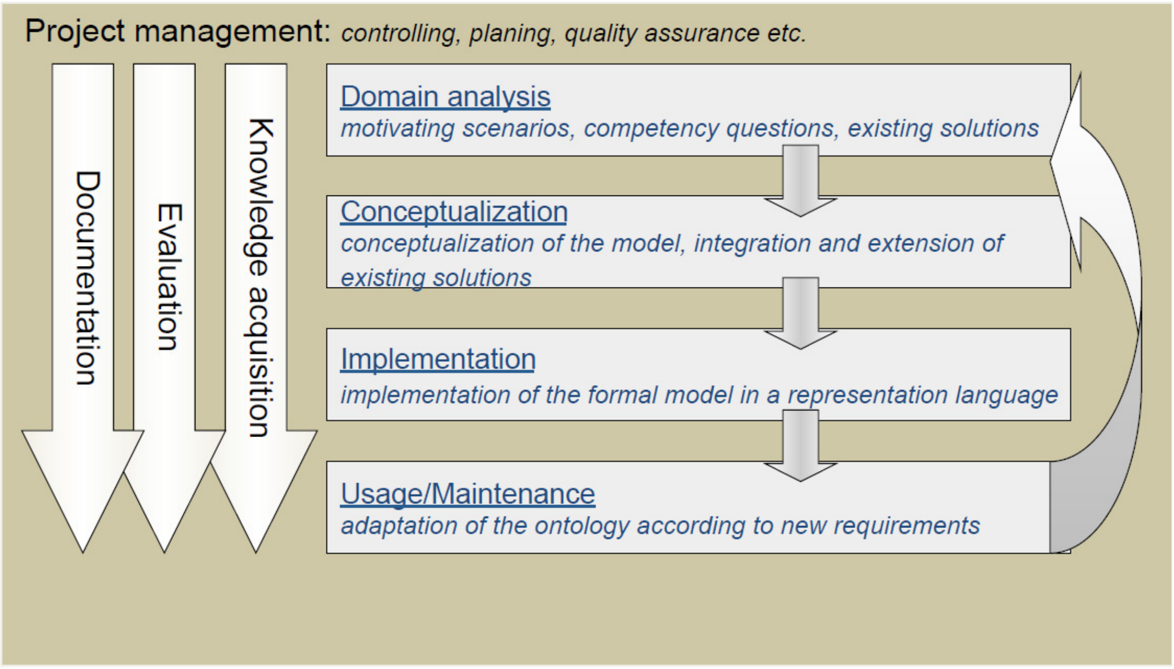
Ontology

KB

STEP 7: Create instances (optional)

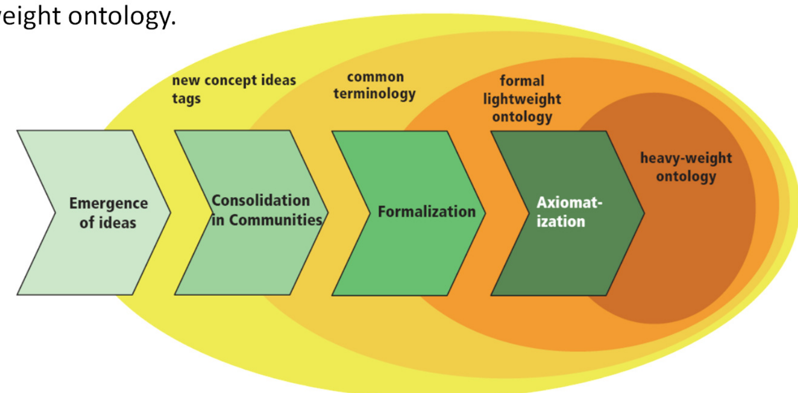
Defining an individual **instance** of a class requires:
- Choosing the class
- Creating an individual instance of that class
- Filling in the slot values

ONTOLOGY LIFECYCLE

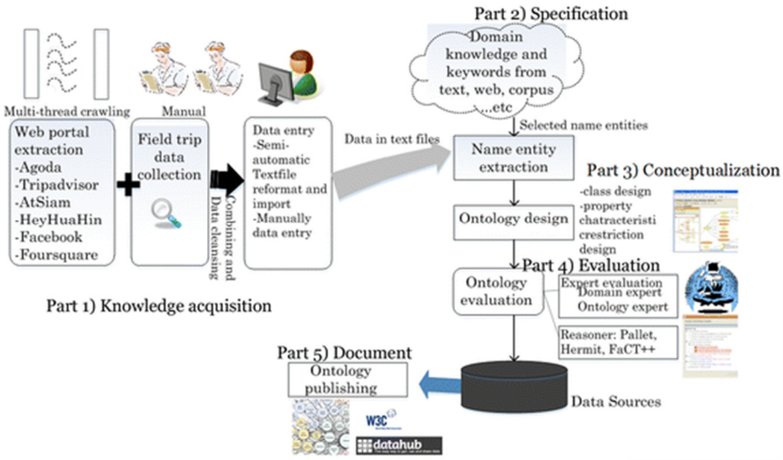


ONTOLOGY MATURING

1. **Emergence of ideas**. In the first phase, new concept ideas are collected in an ad-hoc fashion. This is done using simple tags.
2. **Consolidation in communities**. The concept symbols generated in the first phase are reused and adapted by the user community. The phase aims at extracting concepts from the available tags leading to a common terminology.
3. **Formalization**. This phase adds taxonomic and ad-hoc relations to the common terminology yielding lightweight but formal ontologies.
4. **Axiomatization**. The final step in the methodology addresses knowledge workers, i.e. ontology engineers, to add axiom leading to a heavy-weight ontology.



AN EXAMPLE OF ONTOLOGY CREATION



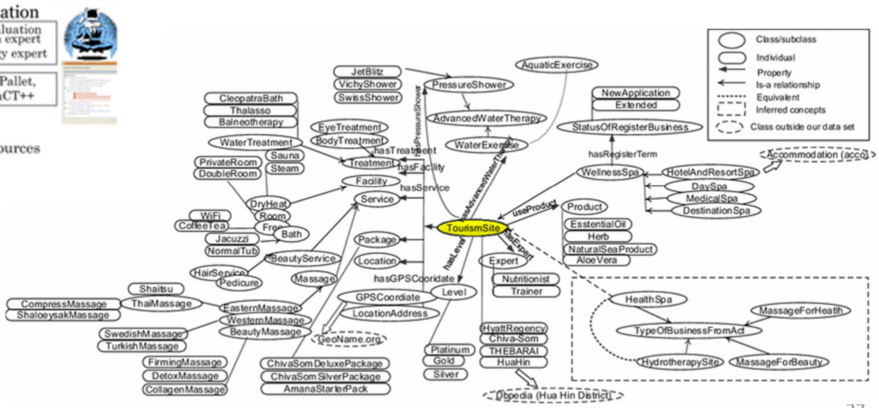
Ontology construction and application in practice case study of health tourism in Thailand

Chantrapornchai, C., Choksuchat, C.

SpringerPlus **5,** 2106 (2016).

https://doi.org/10.1186/s40064-016-3747-3

https://springerplus.springeropen.com/articles/10.1186/s40064-016-3747-3#citeas



KNOWLEDGE GRAPHS

KNOWLEDGE GRAPHS

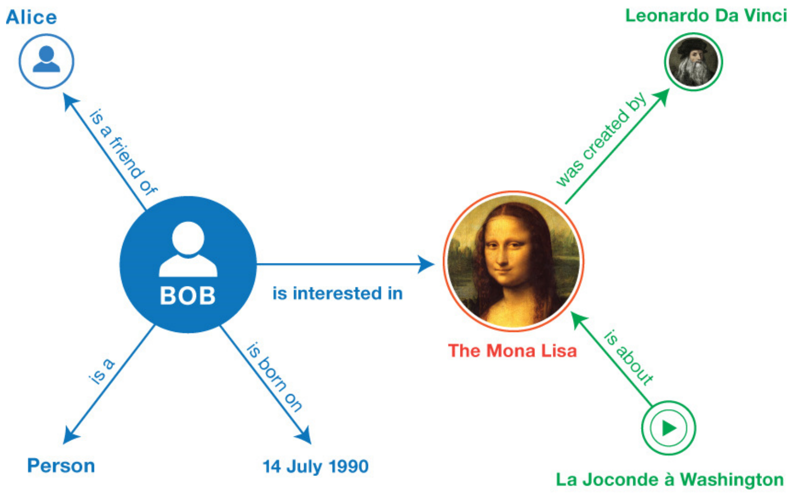
We are surrounded by “entities”, which are connected by relationships.

Graphs are a natural way to represent entities and their relationships.

A **knowledge graph** provides a semantic description of entities and their relationships.

A knowledge graph is a directed heterogeneous multigraph whose node and relation types have domain-specific semantics.

It allows us to encode the knowledge into a form that is human interpretable and amenable to automated analysis and inference.



http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/example-graph.jpg

KNOWLEDGE GRAPHS ...

What is a Knowledge Graph?

From a Knowledge Engineer’s perspective

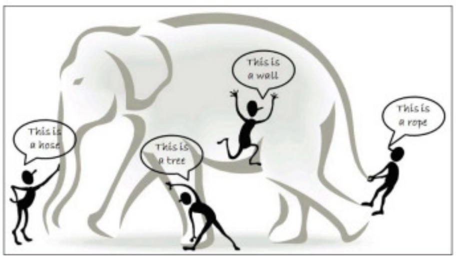
A Knowledge Graph is a ***model of a knowledge domain*** *created by subject-matter experts with the help of intelligent machine learning algorithms.*

From a Data Architect’s perspective

Structured as an additional virtual data layer, the KG lies on top of existing databases or data sets to ***link all your data together at scale*** *– be it structured or unstructured.*

From a Data Engineer’s perspective

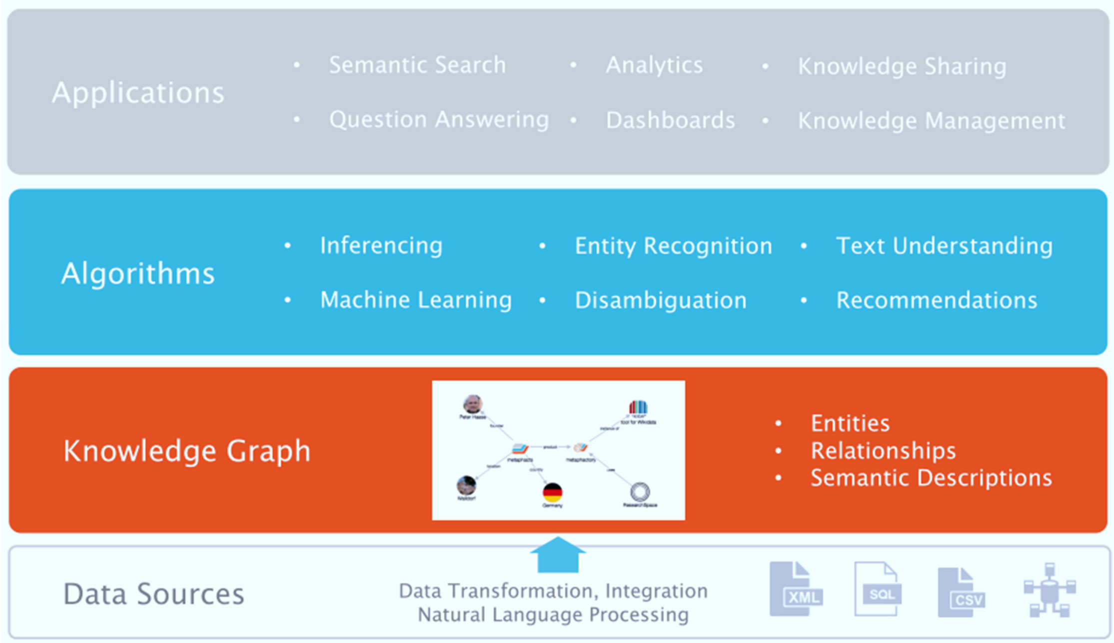
It provides a ***structure and common interface*** *for all of your data and enables the creation of smart multilateral relations throughout your databases.*



https://www.slideshare.net/semwebcompany/introduction-to-knowledge-graphs-and-semantic-ai

The vertices of the knowledge graph are often called **entities** and the directed edges are often called **triplets** and are represented as a **(h, r, t)** tuple, where h is the head entity, t is the tail entity, and r is the relation associating the head with the tail entities.

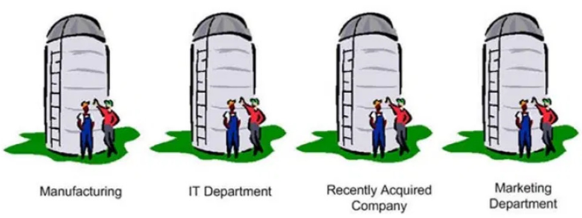
KNOWLEDGE GRAPHS ...



https://www.slideshare.net/phaase/getting-started-with-knowledge-graphs?from_action=save

A Knowledge Graph **connects data across different sources** (structured and unstructured) and provides a semantic enriched structure that enables discovery, insight and empowers AI capabilities.

A problem with data: Silos



In the enterprise, data silos allow local control and governance in a way that is often valuable.

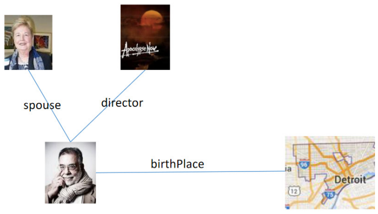
But a silo is a disconnected thing that prevents larger structures from being composed easily. Silos impede everything: app dev, data science and analytics, reporting, compliance, and AI initiatives.

Data silos mean unconnected data

https://www.stardog.com/blog/what-is-a-knowledge-graph/

Key Benefits of Knowledge Graphs

The **more relationships** created, the **more context** the entities/things have, which then provides a bigger picture of a context or situation, helping users make informed decisions with connections that they may have never found.



https://www.cedrus.digital/post/an-introduction-to-the-world-of-knowledge-graphs

1.Combine Disparate Data Silos: Knowledge Graphs help to combine disparate silos of data, giving an overview of all the organization’s knowledge – not only departmentally but also across departments and global organizations.

2.Bring Together Structured and Unstructured Data: Knowledge Graph technology means being able to connect different types of data in meaningful ways and supporting richer data services than most knowledge management systems. In addition, any graph can be linked to other graphs as well as relational databases. Organizations will then use this technology to extract and discover deeper and more subtle patterns with the help of AI and Machine Learning technology.

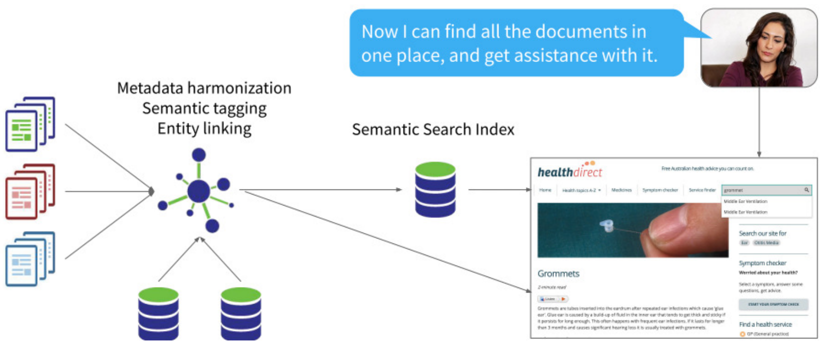
3.Make Better Decisions by Finding Things Faster: Knowledge Graph technology can help provide enriched and in-depth search results, helping to provide relevant facts and contextualized answers to specific questions. Knowledge Graphs can do this because of its networks of “things” and facts that belong to these “things”. “Things” can be any business objects or attributes and facets of these business objects, such as: projects, products, employees or their skills.

4.Data Standards and Interoperability: Knowledge Graphs are compliant with W3C standards, allowing for the re-use of publicly available industry graphs and ontologies (e.g., FIBO, CHEBI, ESCO, etc.), as well as the ISO standard for multilingual thesauri.

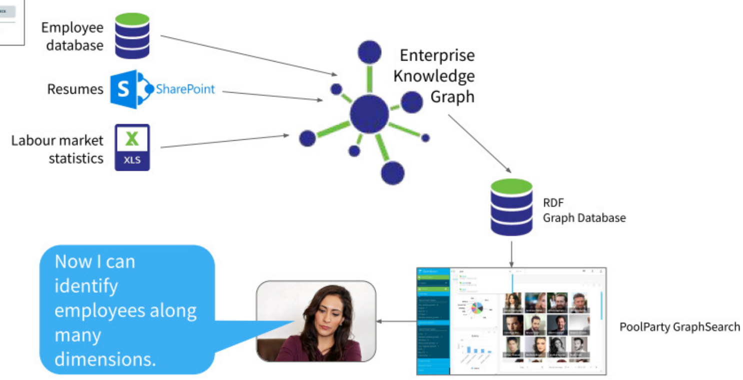
5.AI Enablement: Data from unstructured data sources up to highly structured data, can be harmonized and linked so that the resulting higher data quality can be used for additional tasks, such as machine learning (ML). Knowledge Graphs are the linking engine for the management of enterprise data and a driver for new approaches in Artificial Intelligence

KNOWLEDGE GRAPHS …

Knowledge Graphs for **Information Retrieval**

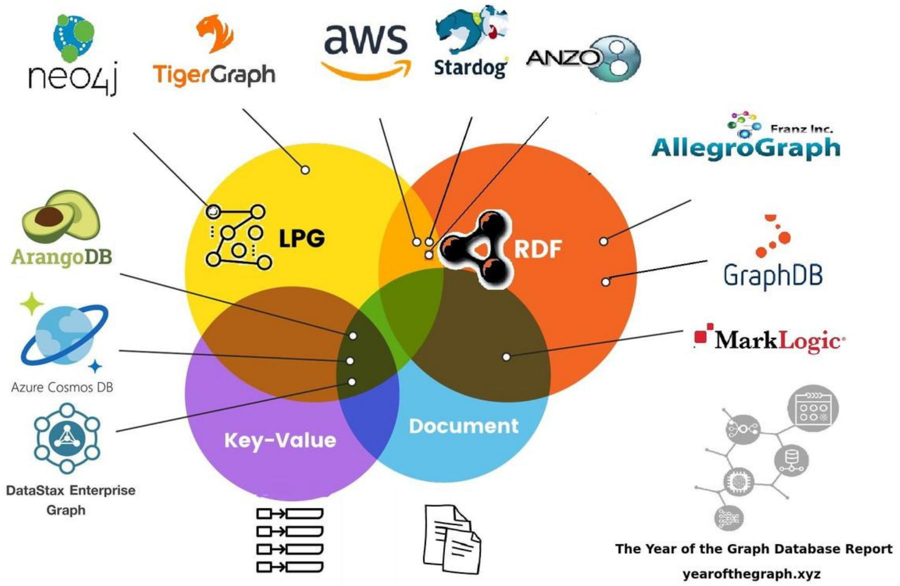


Knowledge Graphs for **Data Integration & Analytics**



https://www.slideshare.net/semwebcompany/introduction-to-knowledge-graphs-and-semantic-ai

KNOWLEDGE GRAPHS – SOME TOOLS

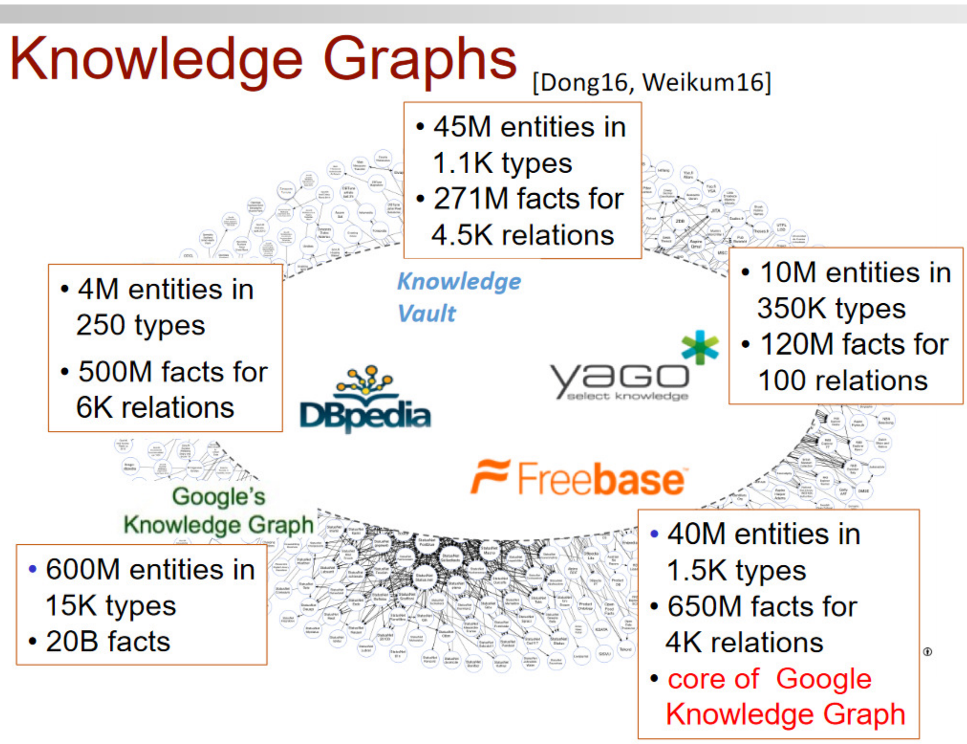


https://www.zdnet.com/article/knowledge-graph-evolution-platforms-that-speak-your-language/

Data engineering tools for building knowledge graphs

https://www.thehyve.nl/articles/tools-building-knowledge-graphs

KNOWLEDGE GRAPHS – EXAMPLES

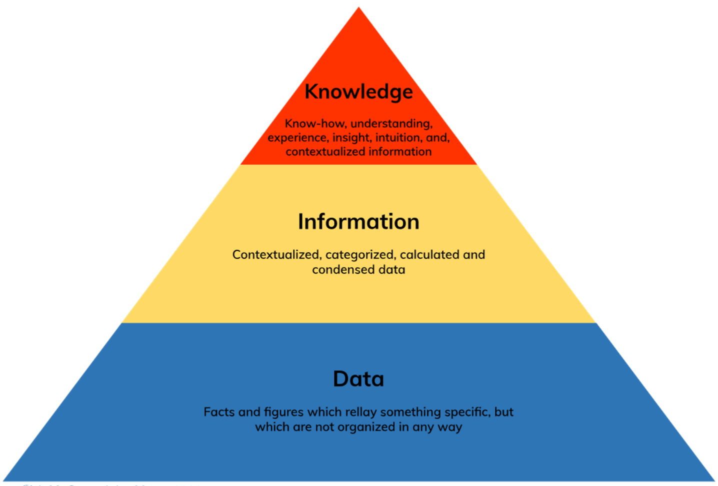


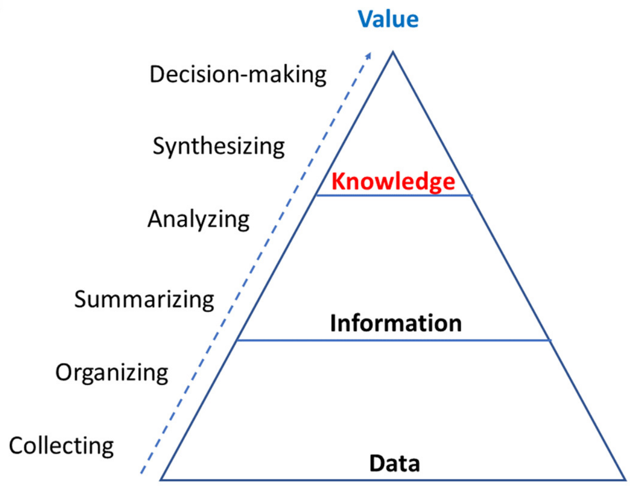
Knowledge representation

Knowledge representation

Knowledge Representation

Subfield of Artificial Intelligence whose main goal is to study how **knowledge** about the world can be represented and what kinds of reasoning can be done with that **knowledge**

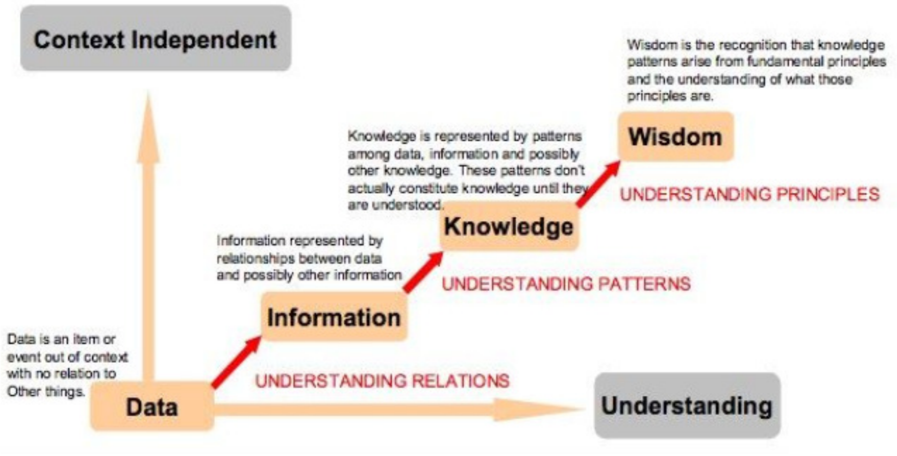


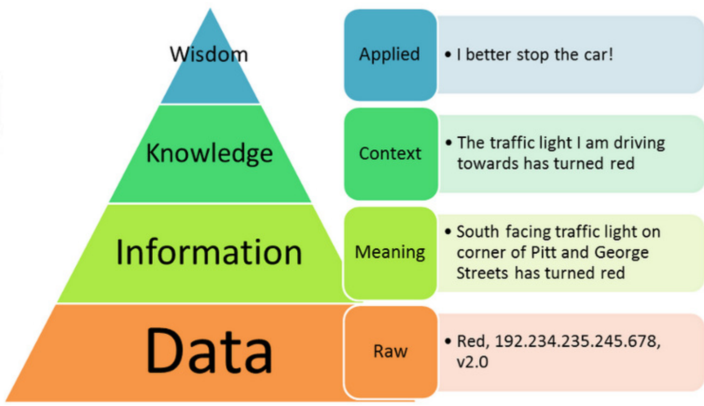


Knowledge Representation and Reasoning

Has a fundamental assumption that an agent's knowledge is explicitly represented in a declarative form, suitable for processing by dedicated reasoning engines

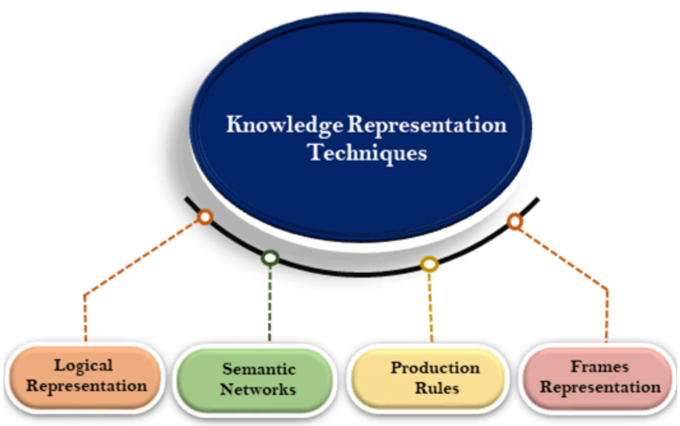
Knowledge representation …





NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Knowledge representation …



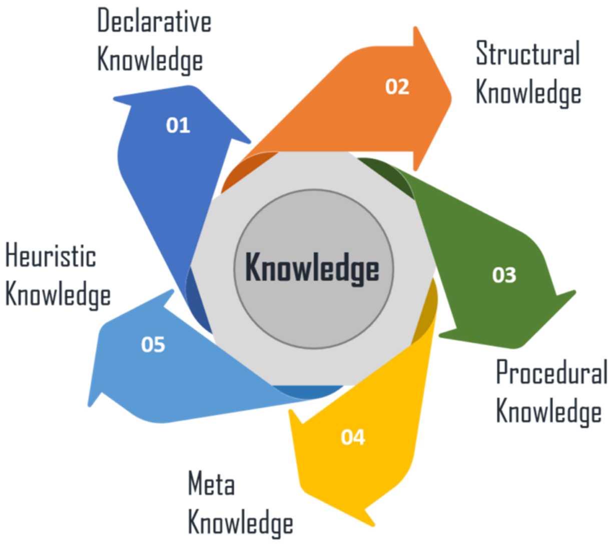
How can knowledge be represented ?

- **Symbolic methods**
  - Declarative Languages (Logic)
  - Imperative Languages (C, C++, Java, etc.)
  - Hybrid Languages (Prolog)
  - Rules
  - Frames
  - Semantic Networks
  - …
- **Non – symbolic methods**
  - Neural Networks
  - Genetic Algorithms

https://www.javatpoint.com/ai-techniques-of-knowledge-representation

https://www.slideshare.net/sajidmarwatt/knowledge-representation-reasoning-31745847

Knowledge representation ...

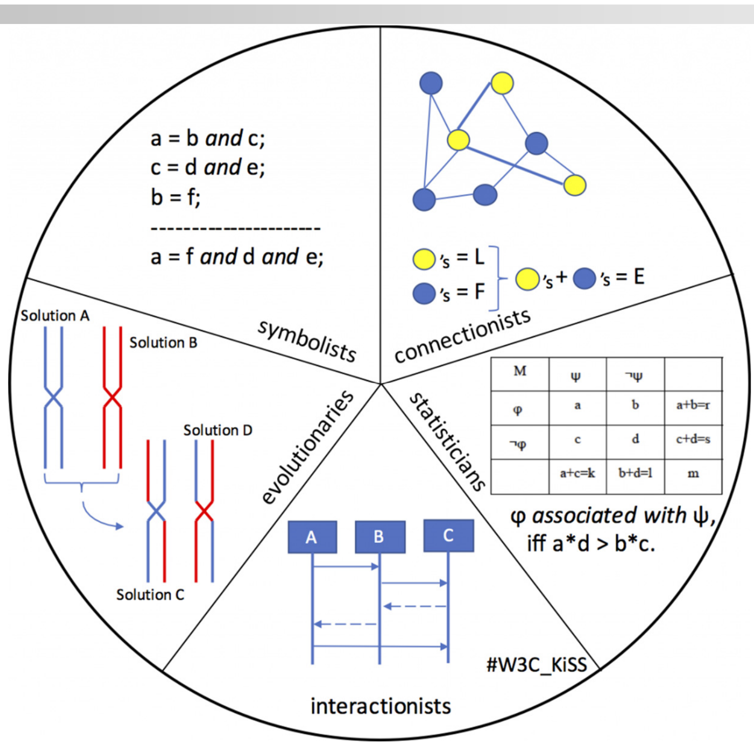


- **Declarative Knowledge** – It includes concepts, facts, and objects and expressed in a declarative sentence.
- **Structural Knowledge** – It is a basic problem-solving knowledge that describes the relationship between concepts and objects.
- **Procedural Knowledge** – This is responsible for knowing how to do something and includes rules, strategies, procedures, etc.
- **Meta Knowledge** – Meta Knowledge defines knowledge about other types of Knowledge.
- **Heuristic Knowledge** – This represents some expert knowledge in the field or subject.

https://www.edureka.co/blog/knowledge-representation-in-ai/

Knowledge representation ...

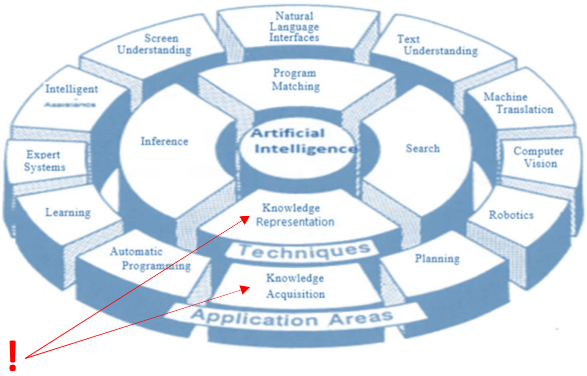
5 schools of thought

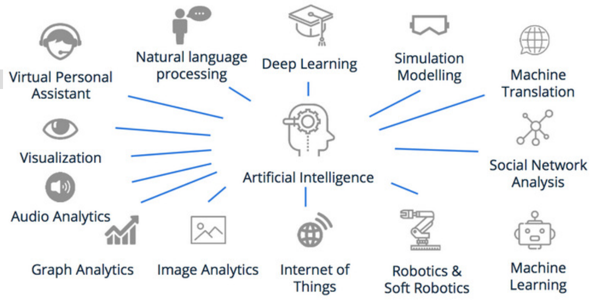


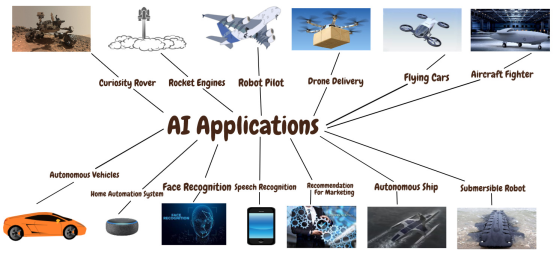
Different "schools" prefer different "languages"

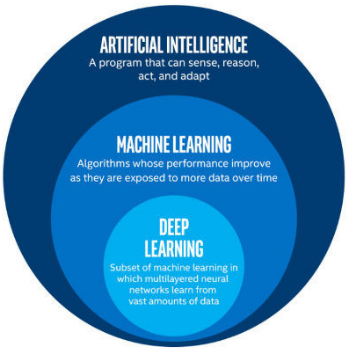
NOVA SCHOOL OF SCIENCE & TECHNOLOGY

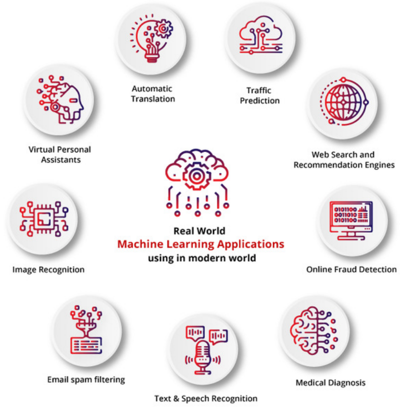
Artificial Intelligence





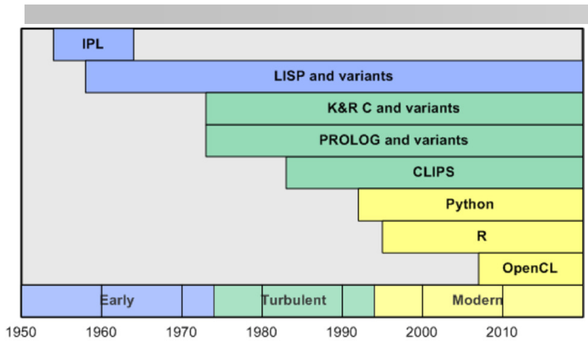




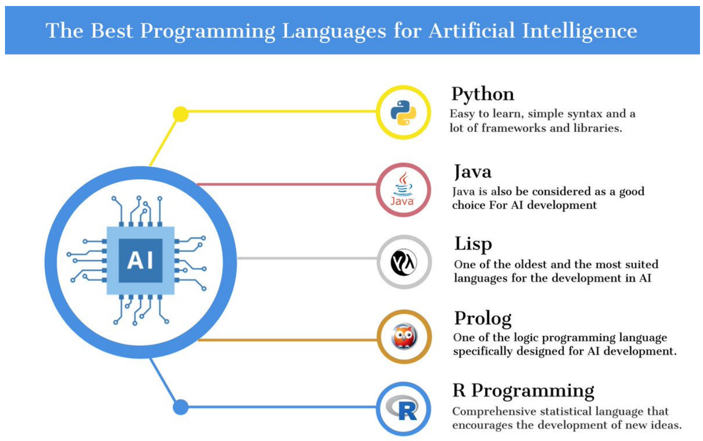


NOVA SCHOOL OF SCIENCE & TECHNOLOGY

The languages of AI



https://developer.ibm.com/technologies/artificial-intelligence/articles/cc-languages-artificial-intelligence/



https://twitter.com/digitalogycorp/status/1202837668603711488

https://www.researchgate.net/figure/Some-Programming-Languages-of-Artificial-Intelligence_fig1_331907158

FURTHER READING

Ontology Development 101: A Guide to Creating Your First Ontology
Natalya F. Noy and Deborah L. McGuinness
https://protege.stanford.edu/publications/ontology_development/ontology101.pdf

https://wiki.hl7.org/Ontology_Development_Methodology

Methodologies, tools and languages for buildingontologies. Where is their meeting point?
Oscar Corcho, Mariano Fernaandez-Loopez, Asuncioon Goomez-Peerez, Data & Knowledge Engineering 46 (2003) 41–64.
http://oa.upm.es/2637/1/JCR02.pdf

Examples of Ontology Model Usage in Engineering Fields
Larysa Globa, Rina Novogrudska, Alexander Koval and Vyacheslav Senchenko, 2018
https://www.intechopen.com/books/ontology-in-information-science/examples-of-ontology-model-usage-in-engineering-fields

An Introduction to Ontologies and Ontology Engineering
Catherine Roussey, Francois Pinet, Myoung Ah Kang, and Oscar Corcho
http://oa.upm.es/10381/1/An_Introduction.pdf

An Introduction to Ontology Engineering
C. Maria Keet, 2020
https://people.cs.uct.ac.za/~mkeet/files/OEbook.pdf