THE 3RD ANNUAL
DataCentric
Architecture Forum

Ontology Design By Enrichiching Taxonomies

Alexis Dimitriadis and Heather Hedden

February 3, 2021

About the Speakers

Alexis Dimitriadis
Data and Knowledge Engineer
Semantic Web Company

Alexis holds a Ph.D. degree in Linguistics from the University of Pennsylvania. He has research and teaching experience in ontologies and linked data, database design, natural language processing, and machine learning and classification. www.linkedin.com/in/alexisdimitriadis

Heather Hedden
Data and Knowledge Engineer
Semantic Web Company

Heather has over 25 years of experience in developing and managing taxonomies, metadata, and other knowledge organization systems for various organizations and applications. She provides training in taxonomy creation and is author of the book *The Accidental Taxonomist*. www.linkedin.com/in/hedden

Semantic Web Company is a leading provider of graph-based metadata, search, and analytic solutions and the developer of PoolParty Semantic Suite software.

SEMANTIC WEB COMPANY

Outline

1. Data, Information, and Knowledge Problems
2. Knowledge Organization Systems
3. Ontology Types and Approaches
4. Using and Applying Ontologies to Taxonomies
5. Creating Knowledge Graphs
6. Ontology Modeling Issues and Tips

1.
Introduction

Data, information, and knowledge problems and solutions

Data, Information, and Knowledge Problems & Solutions

Problems

- Data silos
- Heterogeneous data sources
- Mix of unstructured and structured data
- Same things with different names
- Localized meanings
- Change

Causing

- Inefficiencies
- Missed opportunities
- Poor decisions

Solutions

- Sharing data
- Reusable data sets
- Semantic links
- Semantic data fabric
- Unified views

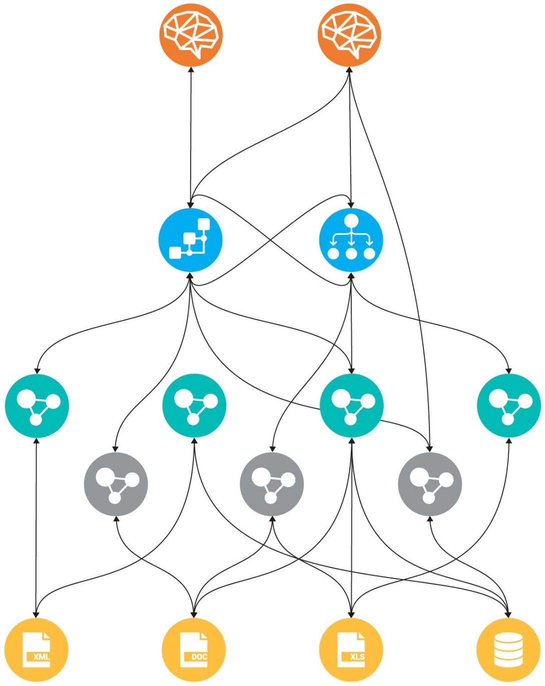
Provided by

- Data-centric architecture
- Knowledge graphs
- Ontologies

Results in

- Better decisions
- Customer satisfaction
- Knowledge discovery

Knowledge Graphs



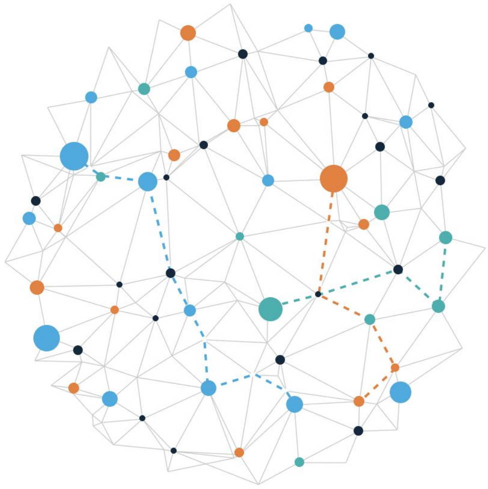
SEMANTIC AI APPLICATION

CONCEPTUAL AND LINGUISTIC MODEL

(VIRTUAL) DATA GRAPH

CONTENT & DATA LAYER

ENTERPRISE KNOWLEDGE GRAPH



An Enterprise Knowledge Graph contains business objects and topics that are closely linked, classified, semantically enriched, and connected to existing data and documents.

Ontologies

Ontologies provide:

- The semantic structure of a knowledge graph:
  - A template for the types, attributes and possible relationships between entities
  - The meaning of the defined nodes and edges
- A standard method to name and link all business objects
- A knowledge model for the domain

➢ The term “ontology” is sometimes used to mean any structured knowledge model, including vocabularies. We’ll use it in the above stricter sense only.

2. Knowledge Organization Systems

Taxonomies, thesauri, and ontologies: types, comparisons, and standards

Knowledge Organization Systems

Knowledge models and knowledge organization systems (KOS)

- **Knowledge model** - names of entities and their relationships in a particular domain, to support knowledge and reasoning about what is in the domain.

- **Knowledge organization system (KOS)** - a system or structure of concepts to support the organization of knowledge and information in order to make their management and retrieval easier.

- A knowledge model may comprises one or more KOS.
- Sometimes a knowledge model and a KOS are the same (e.g., an ontology)
- A KOS may be of limited use, and thus not constitute a knowledge model.

(e.g., a single taxonomy)

Knowledge Organization Systems

- Any system of concepts, terminology, classification, etc. to organize, define, manage, and/or retrieve information.
- A scheme to organize concepts/terms for organizing, classifying, defining, tagging, or retrieving information, rather than any method to organize knowledge directly.
- Includes more than just “controlled vocabularies”

KOS types:

term lists
synonym rings
name authorities
taxonomies
thesauri

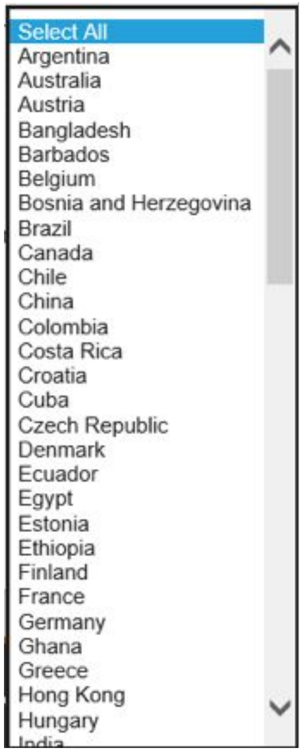
Controlled Vocabularies

glossaries
dictionaries
gazetteers
terminologies
categorization schemes
classification schemes
subject heading schemes
semantic networks
ontologies

Knowledge Organization Systems: Common Types

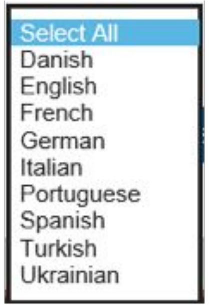
Term list

- Also called a “pick list”
- A simple, flat list of terms
- Usually alphabetical, but could be in other logical order
- Lacking synonyms, it is usually short enough for quick browsing
- Often used for various metadata values
- Part of a *set* of controlled vocabularies, such as facets of a faceted taxonomy
- Sometimes “controlled vocabularies” is used to mean term lists, because they are the most basic kind.



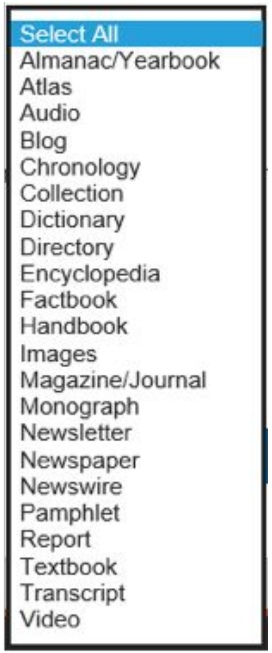
Select All
Argentina
Australia
Austria
Bangladesh
Barbados
Belgium
Bosnia and Herzegovina
Brazil
Canada
Chile
China
Colombia
Costa Rica
Croatia
Cuba
Czech Republic
Denmark
Ecuador
Egypt
Estonia
Ethiopia
Finland
France
Germany
Ghana
Greece
Hong Kong
Hungary
India

Country of publication



Select All
Danish
English
French
German
Italian
Portuguese
Spanish
Turkish
Ukrainian

Language of publication



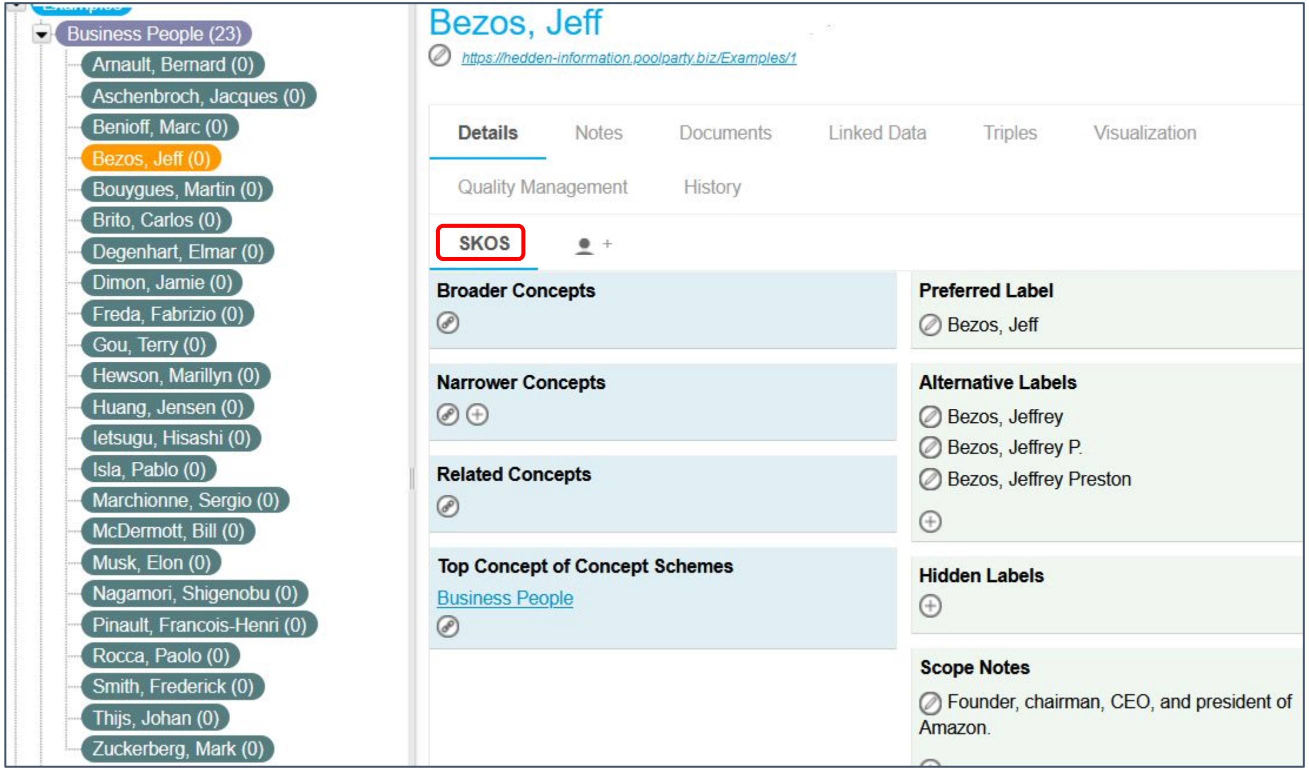
Select All
Almanac/Yearbook
Atlas
Audio
Blog
Chronology
Collection
Dictionary
Directory
Encyclopedia
Factbook
Handbook
Images
Magazine/Journal
Monograph
Newsletter
Newspaper
Newswire
Pamphlet
Report
Textbook
Transcript
Video

Publication format

Knowledge Organization Systems: Common Types

Name authority

- Also called Authority file
- For named entities/ concrete entities/ proper nouns
- A controlled vocabulary with preferred names and variant/alternative names.
- May or may not have hierarchical relationships between named entities.
- Usually has additional information/attributes (metadata) for each named entity (although limited in SKOS)ß



- Business People (23)
  - Arnault, Bernard (0)
  - Aschenbroch, Jacques (0)
  - Benioff, Marc (0)
  - Bezos, Jeff (0)
  - Bouygues, Martin (0)
  - Brito, Carlos (0)
  - Degenhart, Elmar (0)
  - Dimon, Jamie (0)
  - Freda, Fabrizio (0)
  - Gou, Terry (0)
  - Hewson, Marillyn (0)
  - Huang, Jensen (0)
  - Ietsugu, Hisashi (0)
  - Isla, Pablo (0)
  - Marchionne, Sergio (0)
  - McDermott, Bill (0)
  - Musk, Elon (0)
  - Nagamori, Shigenobu (0)
  - Pinault, Francois-Henri (0)
  - Rocca, Paolo (0)
  - Smith, Frederick (0)
  - Thijs, Johan (0)
  - Zuckerberg, Mark (0)

Bezos, Jeff

https://hedden-information.poolparty.biz/Examples/1

Broader Concepts

Narrower Concepts

Related Concepts

Top Concept of Concept Schemes

Business People

Preferred Label

Bezos, Jeff

Alternative Labels

Bezos, Jeffrey

Bezos, Jeffrey P.

Bezos, Jeffrey Preston

Hidden Labels

Scope Notes

Founder, chairman, CEO, and president of Amazon.

Knowledge Organization Systems: Common Types

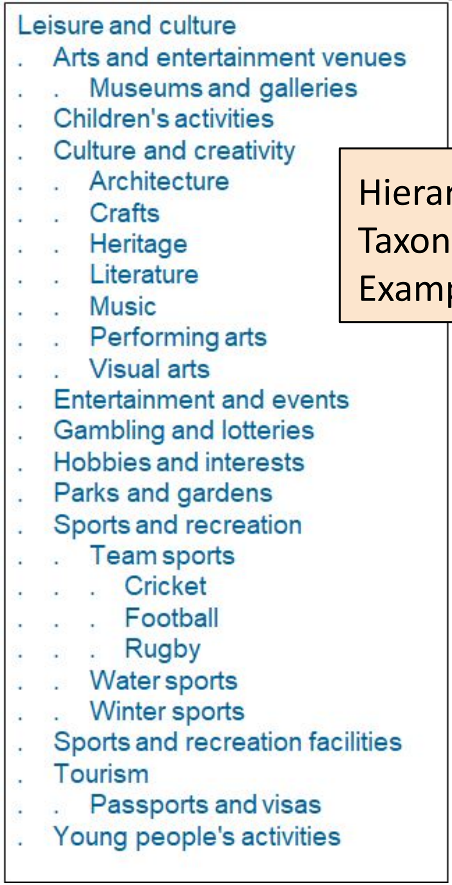
Taxonomy

- A KOS with broader/narrower (parent/child) relationships that include all concepts to create a hierarchical structure.
- Has a focus on categorizing and organization concepts.
- May or may not have “synonyms” to point to the correct, preferred terms/labels
- May comprise several hierarchies or facets. (A facet can be considered as a hierarchy.)

➢ “Taxonomy” sometimes refers to any kind of controlled vocabulary (term list, authority file, synonym ring, classification scheme, thesaurus, etc.)

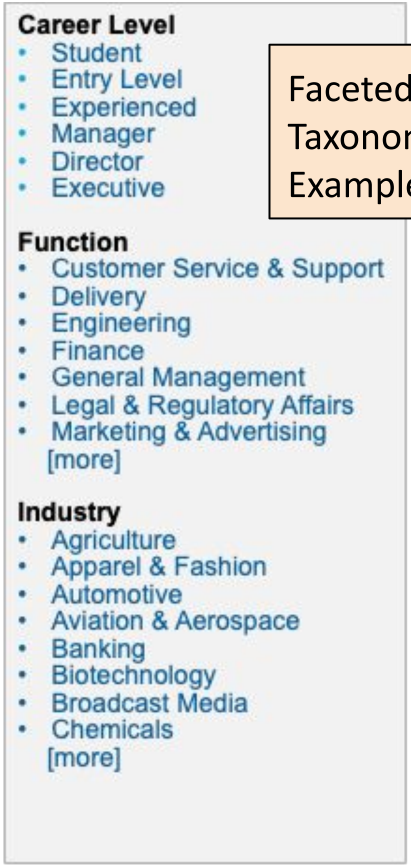
Knowledge Organization Systems: Common Types

Taxonomy Examples



- Leisure and culture
  - Arts and entertainment venues
    - Museums and galleries
  - Children's activities
  - Culture and creativity
    - Architecture
    - Crafts
    - Heritage
    - Literature
    - Music
    - Performing arts
    - Visual arts
  - Entertainment and events
  - Gambling and lotteries
  - Hobbies and interests
  - Parks and gardens
  - Sports and recreation
    - Team sports
      - Cricket
      - Football
      - Rugby
    - Water sports
    - Winter sports
  - Sports and recreation facilities
  - Tourism
    - Passports and visas
  - Young people's activities

Hierarchical Taxonomy Example



Career Level
- Student
- Entry Level
- Experienced
- Manager
- Director
- Executive

Function
- Customer Service & Support
- Delivery
- Engineering
- Finance
- General Management
- Legal & Regulatory Affairs
- Marketing & Advertising

[more]

Industry
- Agriculture
- Apparel & Fashion
- Automotive
- Aviation & Aerospace
- Banking
- Biotechnology
- Broadcast Media
- Chemicals

[more]

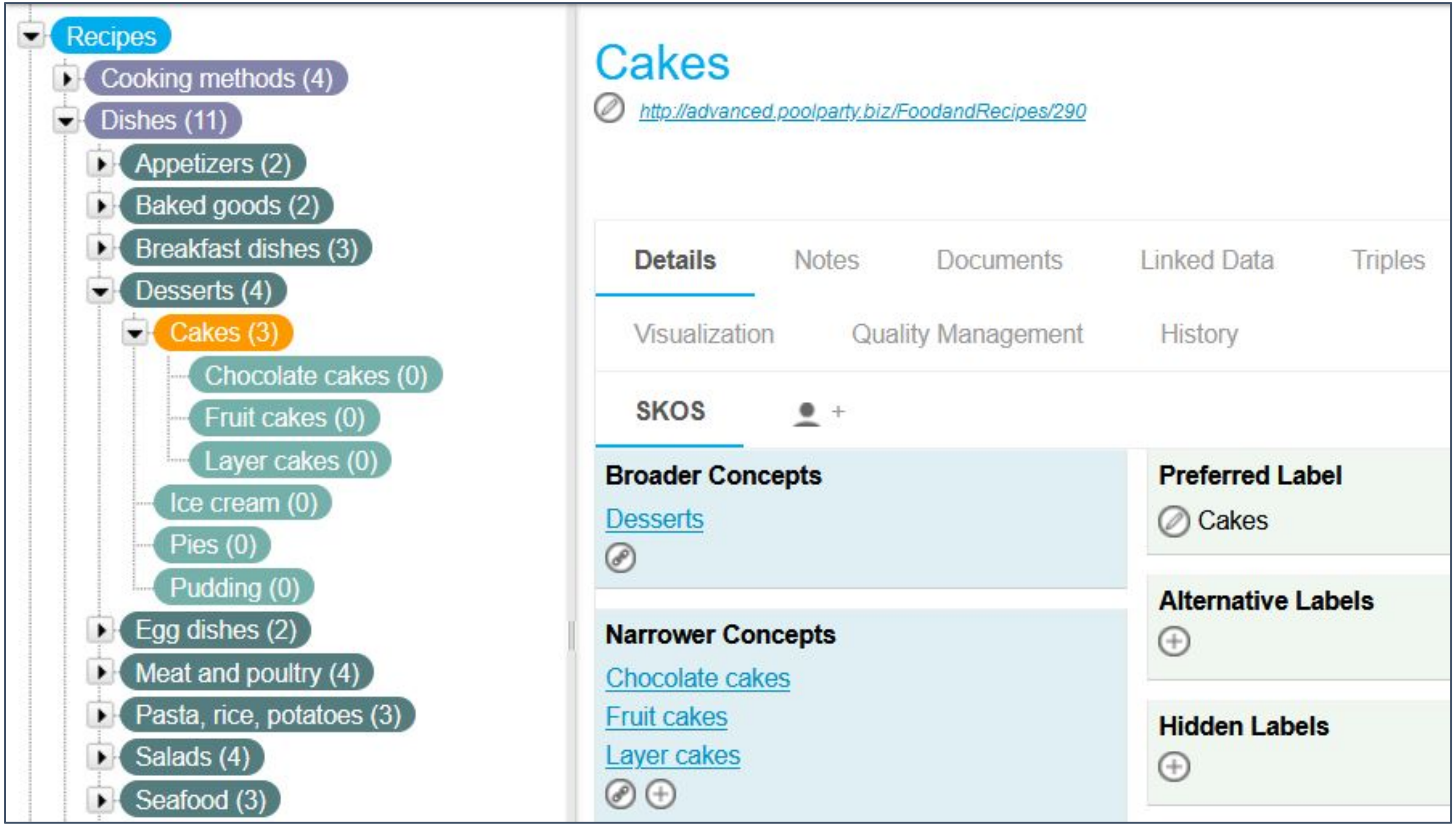
Faceted Taxonomy Example

Knowledge Organization Systems: Common Types

Hierarchical taxonomy example

Concepts have broader concepts and narrower concepts.

Screenshot
from Poolparty



- Recipes
  - Cooking methods (4)
  - Dishes (11)
    - Appetizers (2)
    - Baked goods (2)
    - Breakfast dishes (3)
    - Desserts (4)
      - Cakes (3)
        - Chocolate cakes (0)
        - Fruit cakes (0)
        - Layer cakes (0)
      - Ice cream (0)
      - Pies (0)
      - Pudding (0)
    - Egg dishes (2)
    - Meat and poultry (4)
    - Pasta, rice, potatoes (3)
    - Salads (4)
    - Seafood (3)

Cakes

http://advanced.poolparty.biz/FoodandRecipes/290

Details | Notes | Documents | Linked Data | Triples

Visualization | Quality Management | History

SKOS | +

Broader Concepts

Desserts

Narrower Concepts

Chocolate cakes

Fruit cakes

Layer cakes

Preferred Label

Cakes

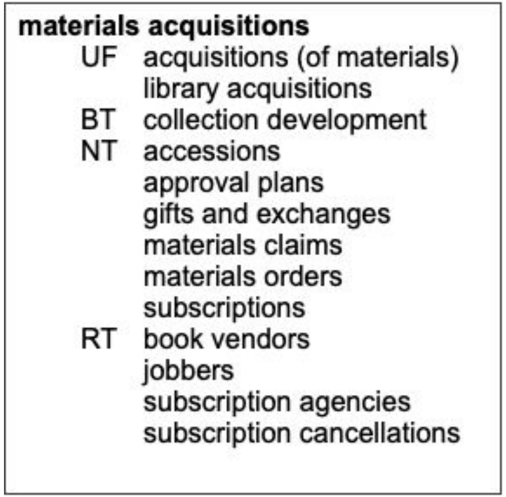
Alternative Labels

Hidden Labels

Knowledge Organization Systems: Common Types

Thesaurus

- A KOS that has standard structured relationships between terms/concepts
  - Hierarchical: broader term/narrower term (BT/NT)
  - Associative: related terms (RT)
  - Preferred terms and Alternative terms (as equivalence relationship USE/UF) or preferred labels and alternative labels.
- Created in accordance with standards:
  - ISO 25964-1 Part 1, *Thesauri and interoperability with other vocabularies*
  - ANSI/NISO Z39.19 *Guidelines for Construction, Format, and Management of Monolingual Controlled Vocabularies*
- The kind of KOS most used in indexing articles for library/ academic research; existed, originally in print, since 1960s



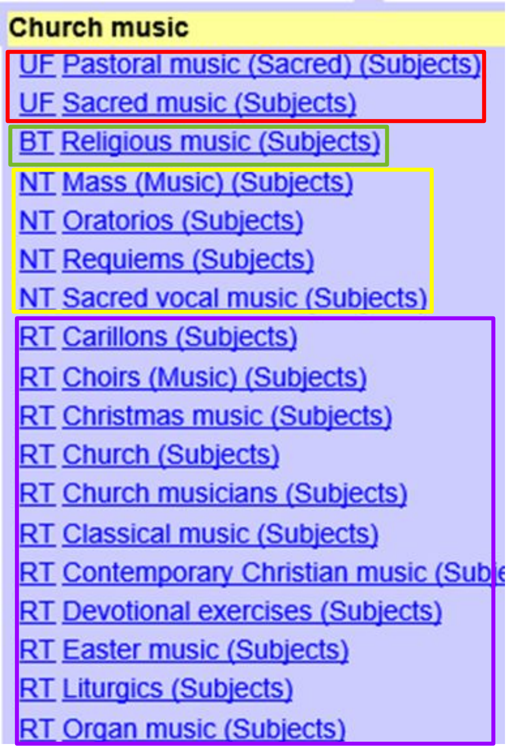
materials acquisitions

UF	acquisitions (of materials) library acquisitions
BT	collection development
NT	accessions approval plans gifts and exchanges materials claims materials orders subscriptions
RT	book vendors jobbers subscription agencies subscription cancellations

ASIS&T Thesaurus entry example

Knowledge Organization Systems: Common Types

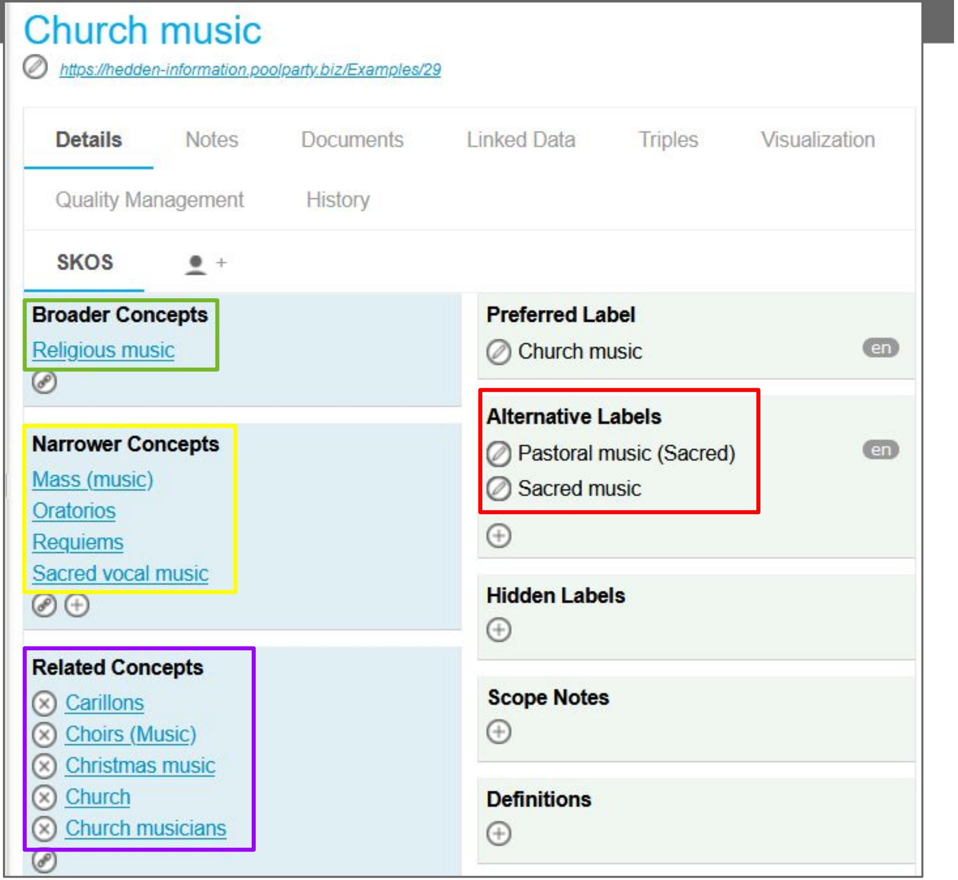
ANSI/NISO thesaurus model and SKOS model compared



Church music

UF Pastoral music (Sacred) (Subjects)
UF Sacred music (Subjects)
BT Religious music (Subjects)
NT Mass (Music) (Subjects)
NT Oratorios (Subjects)
NT Requiems (Subjects)
NT Sacred vocal music (Subjects)
RT Carillons (Subjects)
RT Choirs (Music) (Subjects)
RT Christmas music (Subjects)
RT Church (Subjects)
RT Church musicians (Subjects)
RT Classical music (Subjects)
RT Contemporary Christian music (Subj
RT Devotional exercises (Subjects)
RT Easter music (Subjects)
RT Liturgics (Subjects)
RT Organ music (Subjects)

ANSI/NISO thesaurus model



Church music

https://hedden-information.poolparty.biz/Examples/29

Broader Concepts
Religious music

Narrower Concepts
Mass (music)
Oratorios
Requiems
Sacred vocal music

Related Concepts
Carillons
Choirs (Music)
Christmas music
Church
Church musicians

Preferred Label
Church music en

Alternative Labels
Pastoral music (Sacred) en
Sacred music

Hidden Labels

Scope Notes

Definitions

SKOS thesaurus model (in PoolParty)

Knowledge Organization Systems: Common Types

Ontology

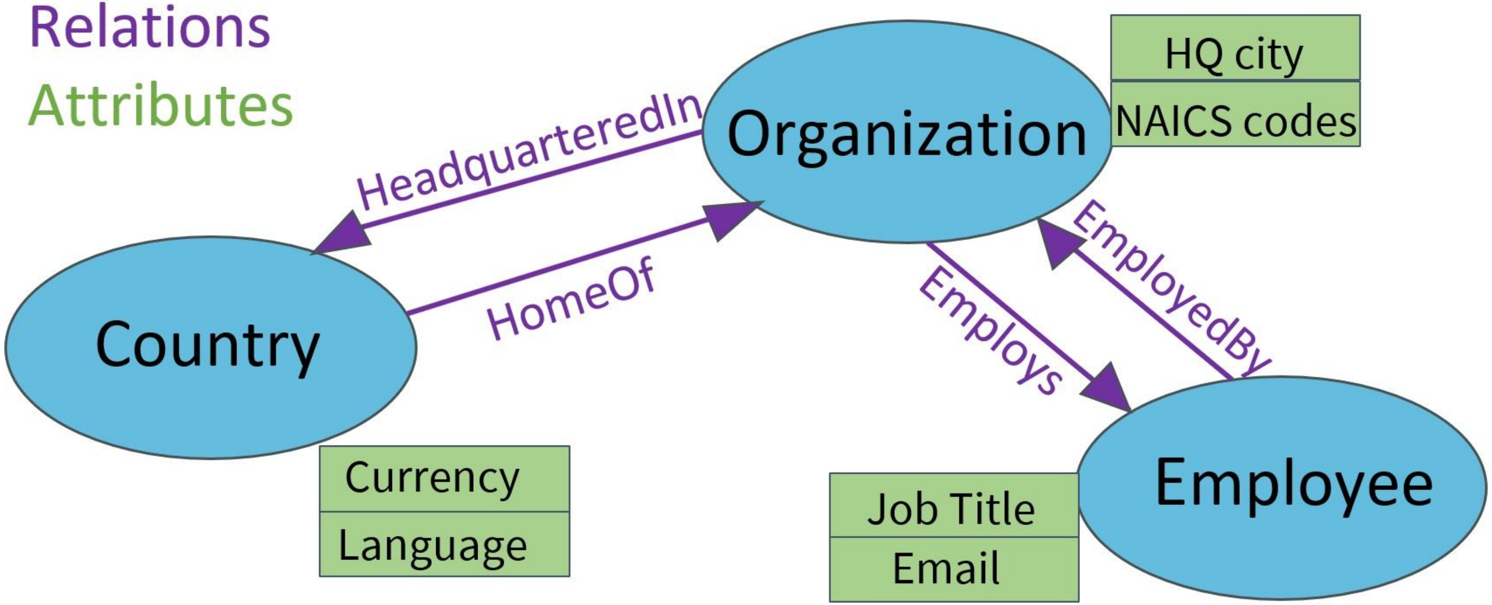
- The most complex or semantically rich kind of KOS
- Arguably not even a KOS, as it’s for knowledge *representation*, not *organization*
- A formal naming and definition of the types, properties and interrelationships of entities in a particular domain.
- Comprises classes, relations, and attributes. Theses are linked in triples.
- Relations contain meaning, are “semantic.”
- W3C guideline: OWL Web Ontology Language Guide W3C Recommendation (2004) http://www.w3.org/TR/2004/REC-owl-guide-20040210/

Knowledge Organization Systems: Common Types

Classes

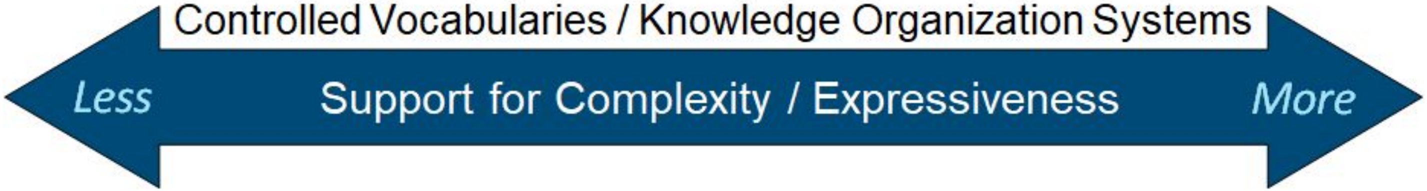
Relations

Attributes



Ontology model example

Knowledge Organization Systems: Common Types


Controlled Vocabularies / Knowledge Organization Systems
Less
Support for Complexity / Expressiveness
More

Term List	Name Authority	Taxonomy	Thesaurus	Ontology
Ambiguity control	Ambiguity control Synonym control (Attributes)	Ambiguity control (Synonym control) Hierarchical relationships	Ambiguity control Synonym control Hierarchical relationships Associative relationships	Ambiguity control (Synonym control) Semantic relationships Attributes Classes

Knowledge Organization System Standards

Standards are of two basic types:

1. *Standards for design*
   – supports an expected experience and results by varied users without training
2. *Standards for specifications* (measurements, protocols, coding, etc.)
   – supports exchange and interoperability

Standards for knowledge organization systems of each type:

1. Standards for design:
   ISO 25964 (2011 and 2013) *Thesauri and Interoperability with Other Vocabularies*
   ANSI/NISO Z39.19-2005 *Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies* www.niso.org/publications/ansiniso-z3919-2005-r2010
2. Standards for specifications, interoperability, and machine readability:
   Dublin Core, MARC, ZThes, DD 8723-5, SKOS, RDF, RDFS, and OWL

Knowledge Organization System Standards

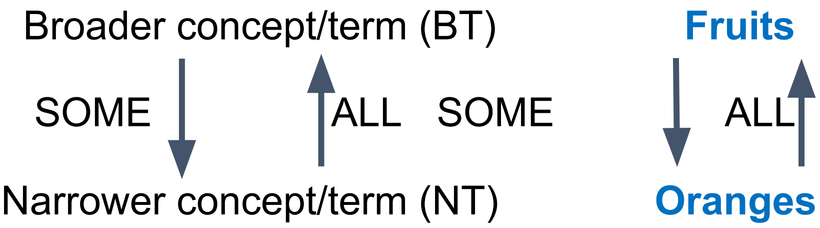
ISO 25964 and ANSI/NISO Z39.19 - Examples from guidelines:

- Concepts are things: nouns or noun phrases
- No duplicates: Concept labels must be unique
- No relationship clashes: A pair of concepts can be either hierarchically or associatively related to each other, but not both.
- No circular relationships: hierarchical relationship logic extends:
    - Concept A is narrower to Concept B, and
    - Concept B is narrower to Concept C,
    - Concept C cannot be narrower to Concept A.

Knowledge Organization System Standards

ISO 25964 and ANSI/NISO Z39.19 - Hierarchical relationship

Reciprocal (bi-directional) relationship, but asymmetrical



Fruits NT (has narrower concept) **Oranges**

Oranges BT (has broader concept) **Fruits**

Three types:

1. Generic – Specific: “is/are a kind of” **Hospitals** NTG **Children's hospitals**
2. Generic – Instance: “is an instance of” **Hospitals** NTI **Massachusetts General Hospital**
3. Whole – Part: “is/are within” **Hospitals** NTP **Emergency rooms**

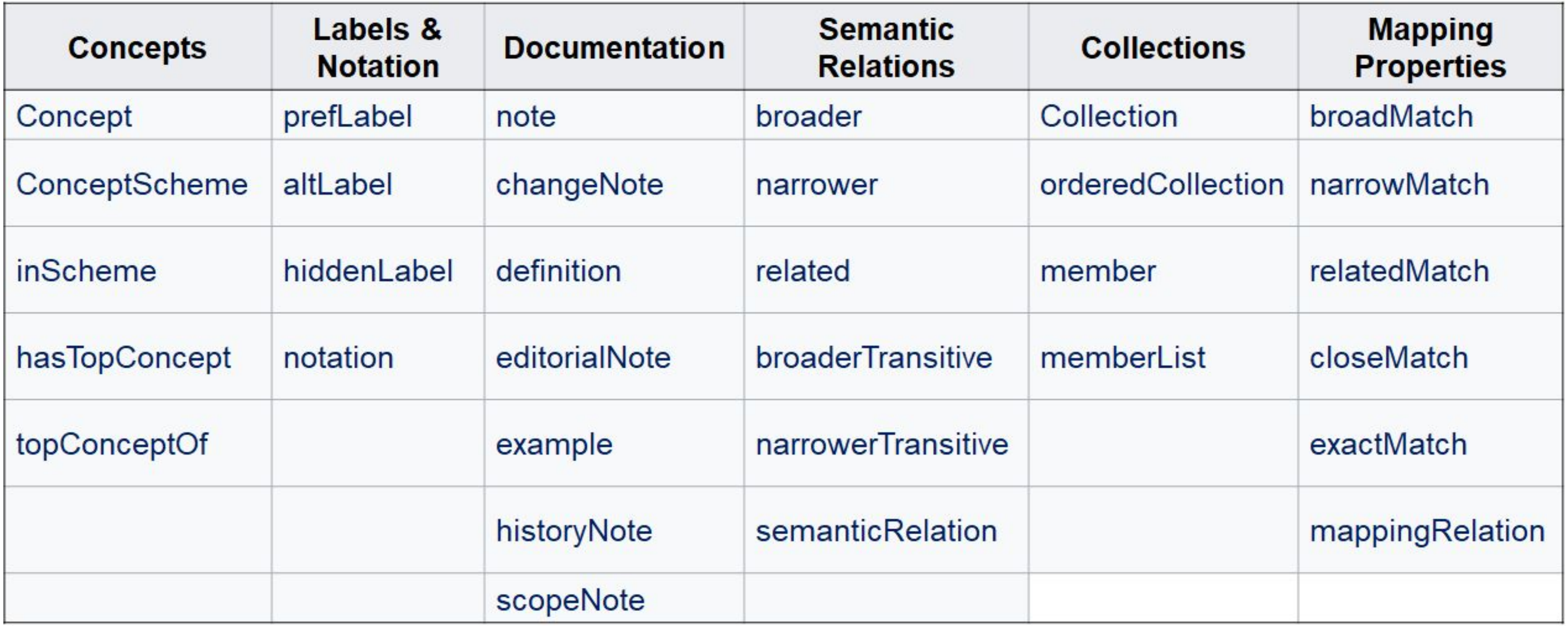
Knowledge Organization System Standards

SKOS (Simple Knowledge Organization System) principles

- A KOS is a group of concepts identified with URIs and grouped into a concept scheme.
- Concepts can be labeled with any number of lexical strings (labels) in any natural language, such as prefLabel and altLabel.
- Concepts can be documented with notes of various types: scope notes, definitions, editorial notes, etc.
- Concepts can be linked to each other using hierarchical and associative semantic relations.
- Concepts can be grouped into collections, which can be labeled and/or ordered.
- Concepts of different concept schemes can be mapped using four basic types of mapping links.

W3C® SKOS

Knowledge Organization System Standards



Concepts	Labels & Notation	Documentation	Semantic Relations	Collections	Mapping Properties
Concept	prefLabel	note	broader	Collection	broadMatch
ConceptScheme	altLabel	changeNote	narrower	orderedCollection	narrowMatch
inScheme	hiddenLabel	definition	related	member	relatedMatch
hasTopConcept	notation	editorialNote	broaderTransitive	memberList	closeMatch
topConceptOf		example	narrowerTransitive		exactMatch
		historyNote	semanticRelation		mappingRelation
		scopeNote			

SKOS Elements

Knowledge Organization System Standards: Ontologies

RDF (Resource Description Framework)

- A World Wide Web (W3C) recommendation https://www.w3.org/TR/rdf11-concepts
- “A standard model for data interchange on the Web”
- Requires the use of URIs to specify things and to specify relations.
- Models all information as subject – predicate – object triples.

RDFS (RDF-Schema)

- A W3C recommendation https://www.w3.org/2001/sw/wiki/RDFS
- Published as part of the RDF Specification Suite Recommendations in 2004
- “A general-purpose language for representing simple RDF vocabularies on the Web”
- Goes beyond RDF to designate classes and properties of RDF resources.

OWL (Web Ontology Language)

- A W3C specification https://www.w3.org/OWL
- “A Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things”
- Based on RDF and RDFS. OWL is W3Cs attempt to extend RDFS.

Knowledge Organization Systems: Summary

Taxonomies	Thesauri	Ontologies
• All concepts belong to a limited number of top-concept hierarchies or facets	• All concepts have relationships, but “hierarchies” may be as few as 2 concepts.	• All concepts have relationships, but not necessarily hierarchical, rather semantic.
• Loosely follow ANSI/NISO and ISO guidelines for organizing concepts.	• Strictly follow ANSI/NISO or ISO for organizing concepts.	• Organizational principles are stateable in the ontology.
• SKOS is the recommended specifications standard.	• SKOS is the recommended specifications standard.	• OWL and RDFS are recommended standards.
• Supports classification, categorization, concept organization.	• Supports concept scoping, disambiguation, relationships with similar concepts.	• Support modeling and understanding of a domain.
• Approach is usually a top-down navigation through concepts.	• Approach is concept-centered and what concepts are related.	• Approach emphasizes entities and their interrelations.
• Especially serving end-users when browsing.	• Especially serving indexers indexing, researching searching.	• Especially serving knowledge modeling, knowledge graphs, reasoning

3.
Ontologies

Types of ontologies and differing approaches to ontology design

Components of an OWL ontology

Entities – subjects or objects of properties (domains and ranges)

- Classes
  - Named sets of concepts that share characteristics and relations
  - May group subclasses or individuals (instances of the class)
  - In SKOS: Concept schemes, Top concepts of a scheme, Concepts (usually with narrower concepts)
- Individuals
  - Members or instances of a class.
  - In SKOS: Concepts (that are named entities or without narrower concepts)

Properties – predicates about individuals (instances)

- Object properties
  - Relations between individuals
  - May be directed (single direction), symmetric, or with an inverse (different in each direction)
  - In SKOS: Relationships
- Datatype properties
  - Attributes or characteristics of individuals
  - The object of a datatype property is a *value*.
  - In SKOS: Attributes

Literals – values of attributes, with just a *lexical form* and a *datatype.*

https://www.w3.org/TR/2012/REC-owl2-primer-20121211/

Ontology Types

Upper or foundation ontologies (top-level ontology, upper model)

- A generic, standard framework to serve as a high-level model for a domain ontology, taxonomy, or other KOS
- Examples: gist, Basic Formal Ontology (BFO), Suggested Upper Merged Ontology (SUMO), General Formal Ontology (GFO)

Domain or ontologies

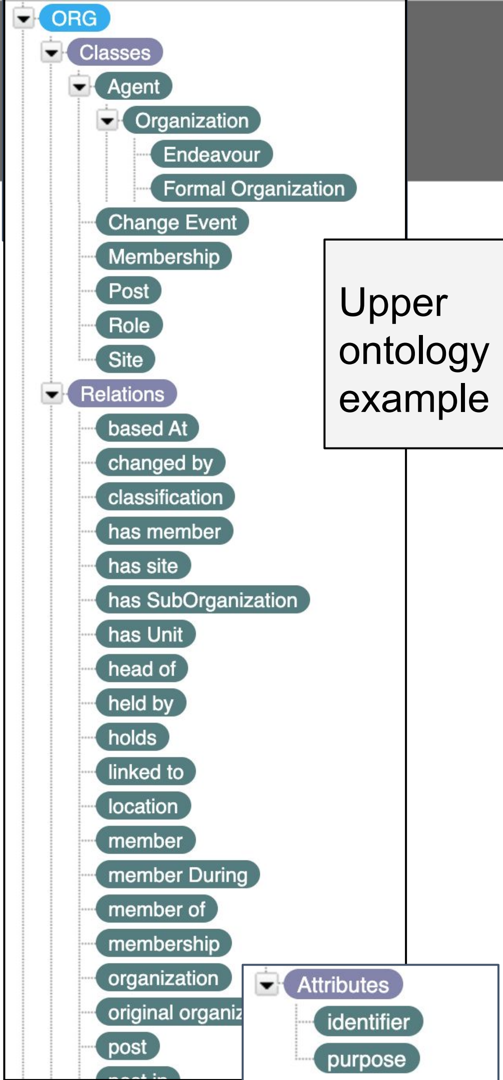
- Concepts and relations belong to a specific subject domain
- Examples: Systems Biology Ontology, Gene Ontology, BBC Ontology, Financial Industry Business Ontology (FIBO)

Actually, a continuum of how generic or specific an ontology may be.

Ontology Types

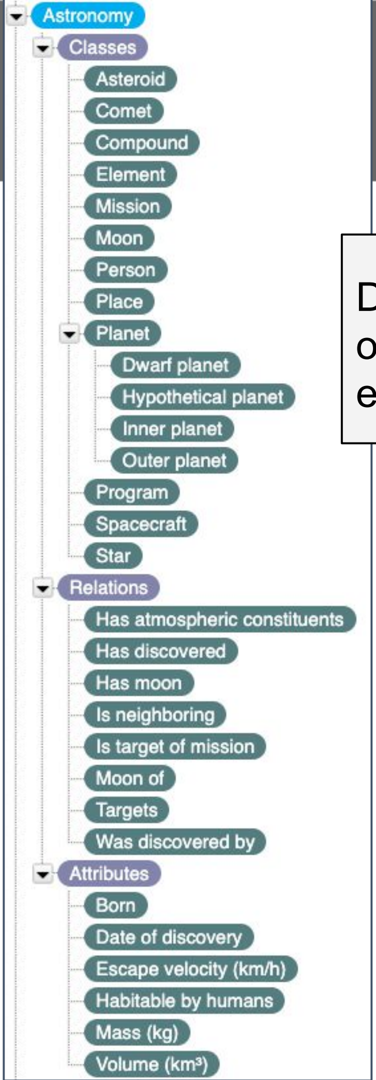
- An upper ontology is not always smaller than a domain ontology, but it is more generic.

- An upper ontology is intended to be extended to define a domain ontology.
- A domain ontology may also be extended or applied to more specific instances.



- ORG
  - Classes
    - Agent
      - Organization
        - Endeavour
        - Formal Organization
    - Change Event
    - Membership
    - Post
    - Role
    - Site
  - Relations
    - based At
    - changed by
    - classification
    - has member
    - has site
    - has SubOrganization
    - has Unit
    - head of
    - held by
    - holds
    - linked to
    - location
    - member
    - member During
    - member of
    - membership
    - organization
    - original organiz
    - post
  - Attributes
    - identifier
    - purpose

Upper ontology example



- Astronomy
  - Classes
    - Asteroid
    - Comet
    - Compound
    - Element
    - Mission
    - Moon
    - Person
    - Place
    - Planet
      - Dwarf planet
      - Hypothetical planet
      - Inner planet
      - Outer planet
    - Program
    - Spacecraft
    - Star
  - Relations
    - Has atmospheric constituents
    - Has discovered
    - Has moon
    - Is neighboring
    - Is target of mission
    - Moon of
    - Targets
    - Was discovered by
  - Attributes
    - Born
    - Date of discovery
    - Escape velocity (km/h)
    - Habitable by humans
    - Mass (kg)
    - Volume (km³)

Domain ontology example

Screenshots from Poolparty

Ontology Types

- “Ontology” may refer to
  - a generic model (upper or domain) or
  - a combination of a taxonomy with a semantic ontology layer.

- If an ontology is not a semantic layer to taxonomies, then it likely needs to contain specificity within it.
- If an ontology is a semantic layer overlaying and linked to taxonomies, then it need not be as large, detailed and specific, even if it’s a domain ontology.

Ontology Types

Components of all ontologies (upper and domain):

Classes
Relations (Object properties)
Attributes (Data properties)

Additional component in certain domain ontologies:

Instances (Individuals)

Questions regarding instances

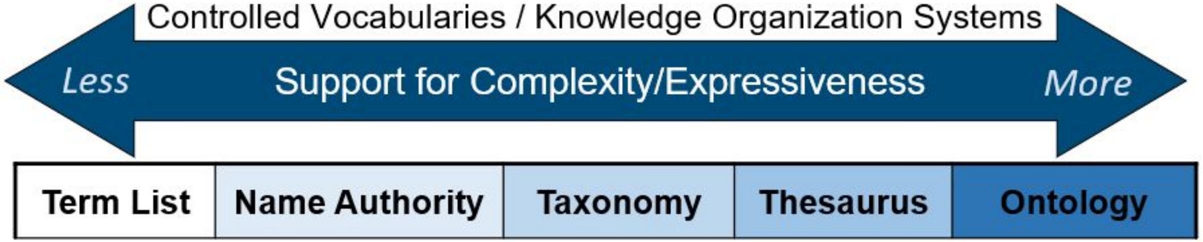
- Are instances the most specific entity (any type), or only unique named entities?
- Are instances data?
- "An ontology need not include any individuals, but one of the general purposes of an ontology is to provide a means of classifying individuals, even if those individuals are not explicitly part of the ontology." - *Ontology Components, Wikipedia* *https://en.wikipedia.org/wiki/Ontology_components*

➢ If instances are not explicitly part of the ontology, then they may be in a linked name authority or taxonomy.

Ontology Approaches

Two approaches to developing domain ontologies:

1. The ontology comprises all entities: classes, subclasses, and instances.

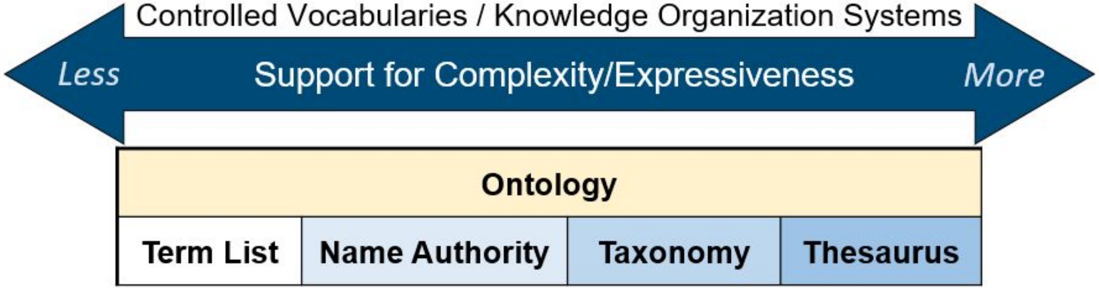


Controlled Vocabularies / Knowledge Organization Systems

Less Support for Complexity/Expressiveness *More*

Term List	Name Authority	Taxonomy	Thesaurus	Ontology

2. The ontology comprises only the classes needed to describe the characteristics of the model.



Controlled Vocabularies / Knowledge Organization Systems

Less Support for Complexity/Expressiveness *More*

Ontology			
Term List	Name Authority	Taxonomy	Thesaurus

Ontology Approaches

Two approaches to domain ontologies:

1. Single knowledge organization system:
   - The ontology comprises all entities: classes, subclasses, and instances.
   - The ontology is as detailed as any taxonomy or thesaurus, but has the addition of semantic relations and attributes.
   - Classes have multiple levels of of subclasses.
   - Has individuals for unique named entity instances.
   - Modeled in dedicated ontology software.

2. Combination of a taxonomy with an ontology model:
   - The ontology comprises classes and subclasses to the extent needed to describe the generic characteristics of the model.
   - The ontology does not include all possible levels of hierarchy, nor any instances.
   - The ontology is a model that is applied as a semantic layer to a taxonomy or multiple taxonomies and/or thesaurus and name authorities.
   - More the hierarchy resides in the SKOS taxonomy.
   - Modeled in combination taxonomy/ontology software, such as PoolParty.

Differing definitions of what is ***in scope*** of the ontology or merely ***linked*** to the ontology.

Ontology Approaches

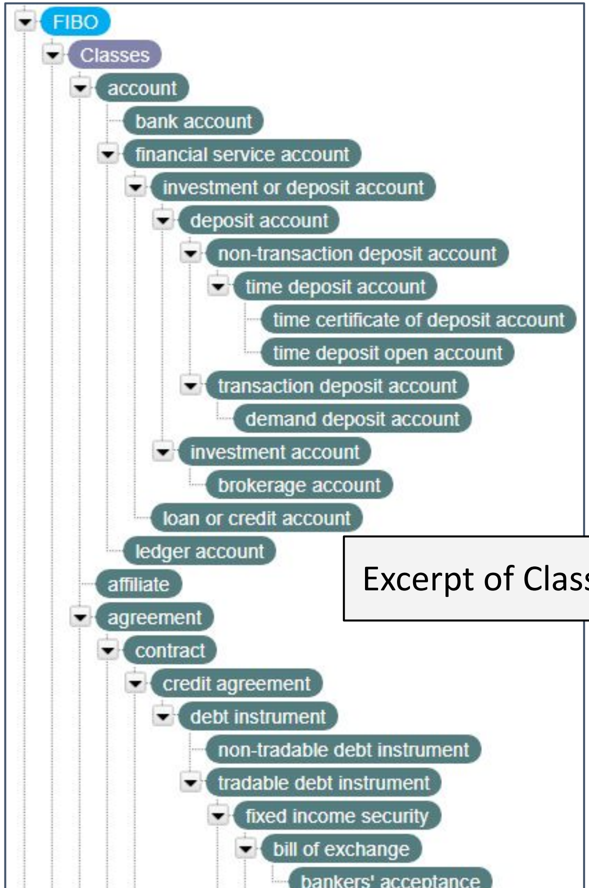
1. Single knowledge organization system

May become extensive.

Domain ontology example:

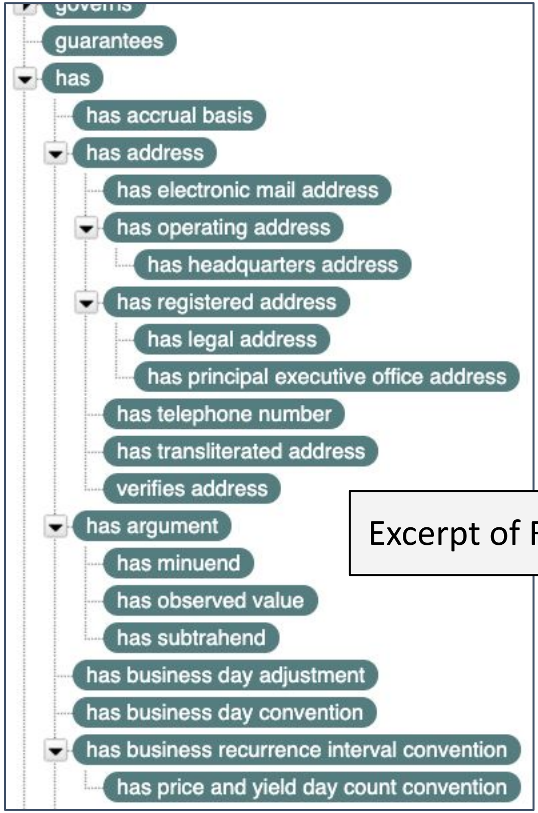
Financial Industry Business Ontology (FIBO)

Classes - 1384
Relations - 522
Attributes - 157



- FIBO
  - Classes
    - account
      - bank account
      - financial service account
        - investment or deposit account
          - deposit account
            - non-transaction deposit account
              - time deposit account
                - time certificate of deposit account
                - time deposit open account
            - transaction deposit account
              - demand deposit account
          - investment account
            - brokerage account
        - loan or credit account
      - ledger account
    - affiliate
    - agreement
      - contract
        - credit agreement
          - debt instrument
            - non-tradable debt instrument
            - tradable debt instrument
              - fixed income security
                - bill of exchange
                  - bankers' acceptance

Excerpt of Classes



- governs
- guarantees
- has
  - has accrual basis
  - has address
    - has electronic mail address
    - has operating address
      - has headquarters address
    - has registered address
      - has legal address
      - has principal executive office address
    - has telephone number
    - has transliterated address
    - verifies address
  - has argument
    - has minuend
    - has observed value
    - has subtrahend
  - has business day adjustment
  - has business day convention
  - has business recurrence interval convention
    - has price and yield day count convention

Excerpt of Relations

Screenshots from Poolparty

Ontology Approaches

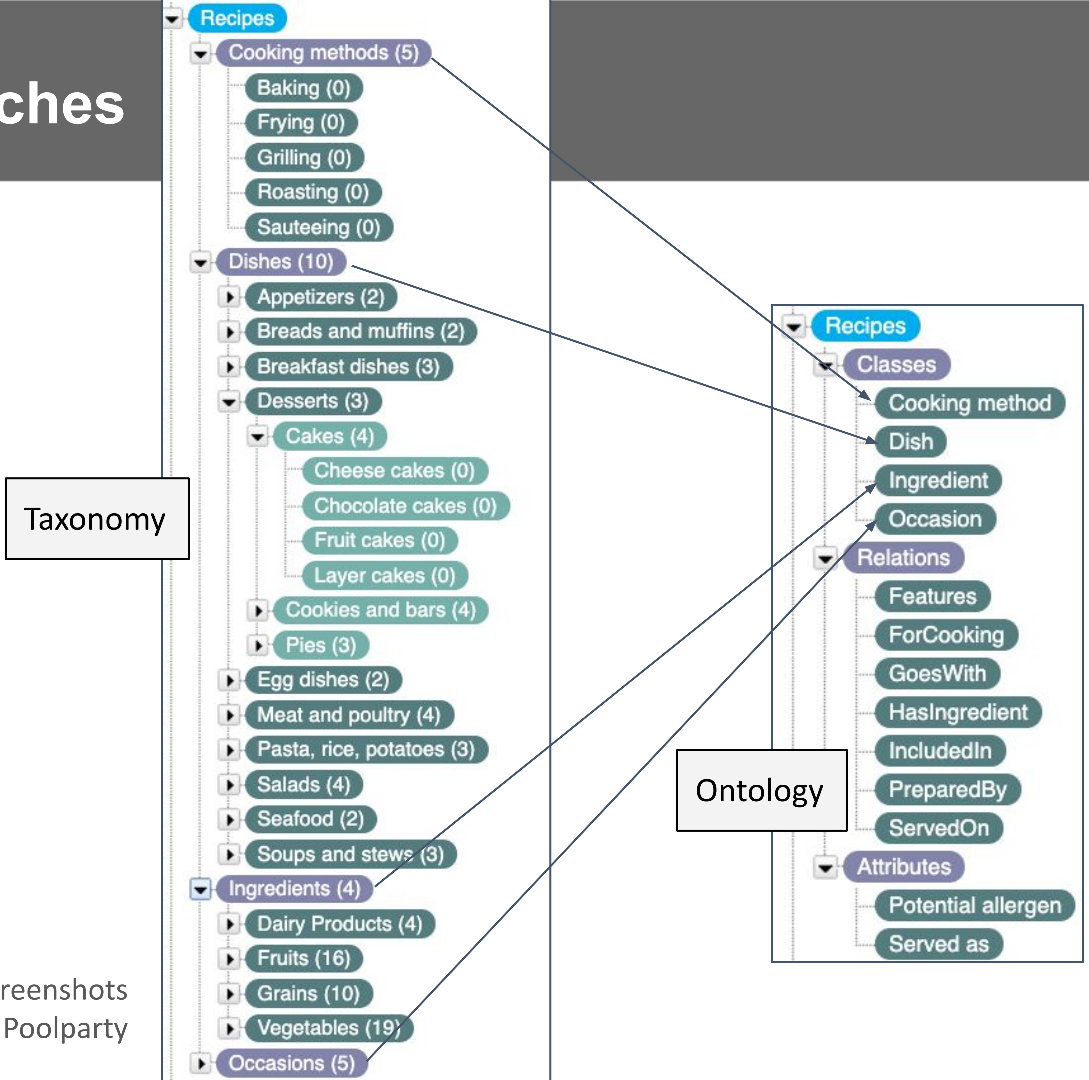
2. Taxonomy + ontology layer

The ontology tends to be smaller and simpler.

Taxonomy may be based on SKOS, whereas ontology is based on OWL.

Domain ontology example:

Classes - 4
Relations - 6
Attributes - 2



Taxonomy

- Recipes
  - Cooking methods (5)
    - Baking (0)
    - Frying (0)
    - Grilling (0)
    - Roasting (0)
    - Sauteeing (0)
  - Dishes (10)
    - Appetizers (2)
    - Breads and muffins (2)
    - Breakfast dishes (3)
    - Desserts (3)
      - Cakes (4)
        - Cheese cakes (0)
        - Chocolate cakes (0)
        - Fruit cakes (0)
        - Layer cakes (0)
      - Cookies and bars (4)
      - Pies (3)
    - Egg dishes (2)
    - Meat and poultry (4)
    - Pasta, rice, potatoes (3)
    - Salads (4)
    - Seafood (2)
    - Soups and stews (3)
  - Ingredients (4)
    - Dairy Products (4)
    - Fruits (16)
    - Grains (10)
    - Vegetables (19)
  - Occasions (5)

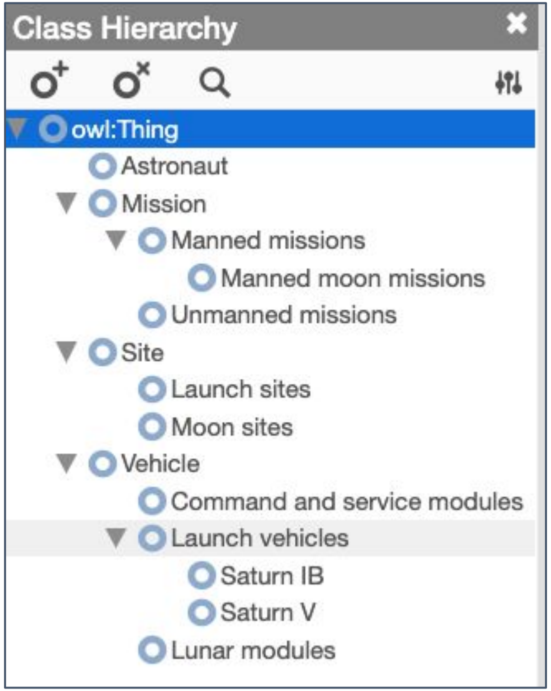
Ontology

- Recipes
  - Classes
    - Cooking method
    - Dish
    - Ingredient
    - Occasion
  - Relations
    - Features
    - ForCooking
    - GoesWith
    - HasIngredient
    - IncludedIn
    - PreparedBy
    - ServedOn
  - Attributes
    - Potential allergen
    - Served as

Screenshots from Poolparty

DataCentric Architecture Forum

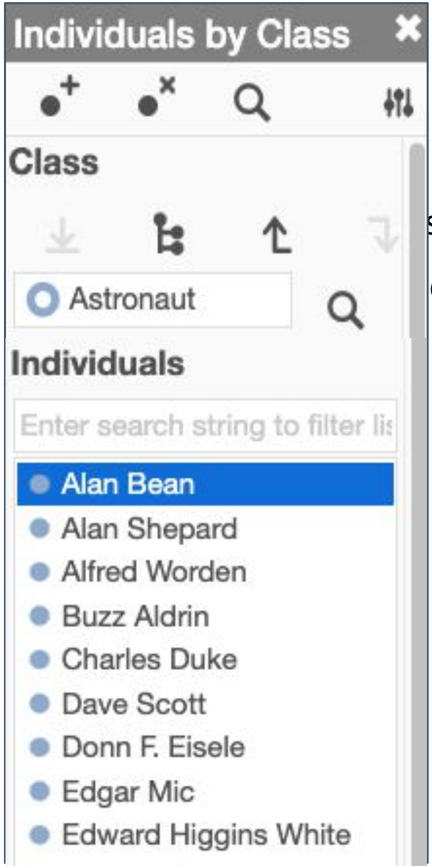
Ontology Approaches



Class Hierarchy

- owl:Thing
  - Astronaut
  - Mission
    - Manned missions
      - Manned moon missions
    - Unmanned missions
  - Site
    - Launch sites
    - Moon sites
  - Vehicle
    - Command and service modules
    - Launch vehicles
      - Saturn IB
      - Saturn V
    - Lunar modules

Fully expanded class hierarchy



Individuals by Class

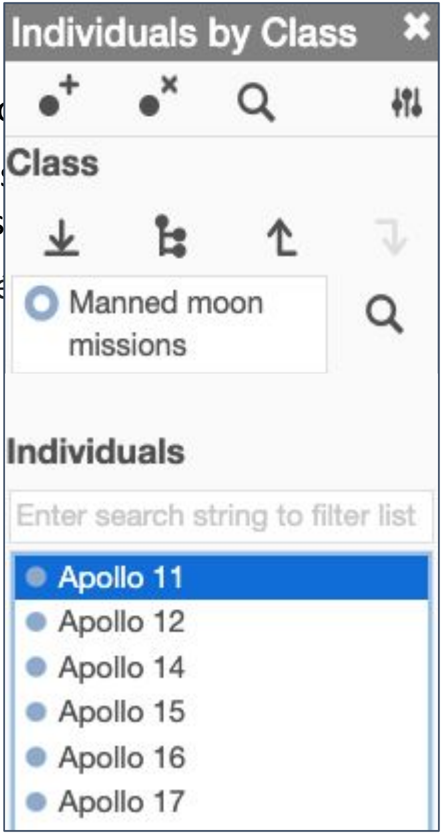
Class

Astronaut

Individuals

Enter search string to filter lis

- Alan Bean
- Alan Shepard
- Alfred Worden
- Buzz Aldrin
- Charles Duke
- Dave Scott
- Donn F. Eisele
- Edgar Mic
- Edward Higgins White



Individuals by Class

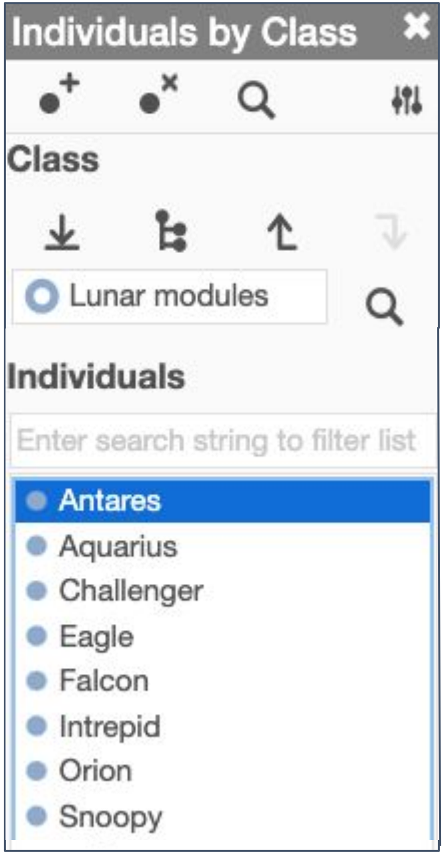
Class

Manned moon missions

Individuals

Enter search string to filter list

- Apollo 11
- Apollo 12
- Apollo 14
- Apollo 15
- Apollo 16
- Apollo 17



Individuals by Class

Class

Lunar modules

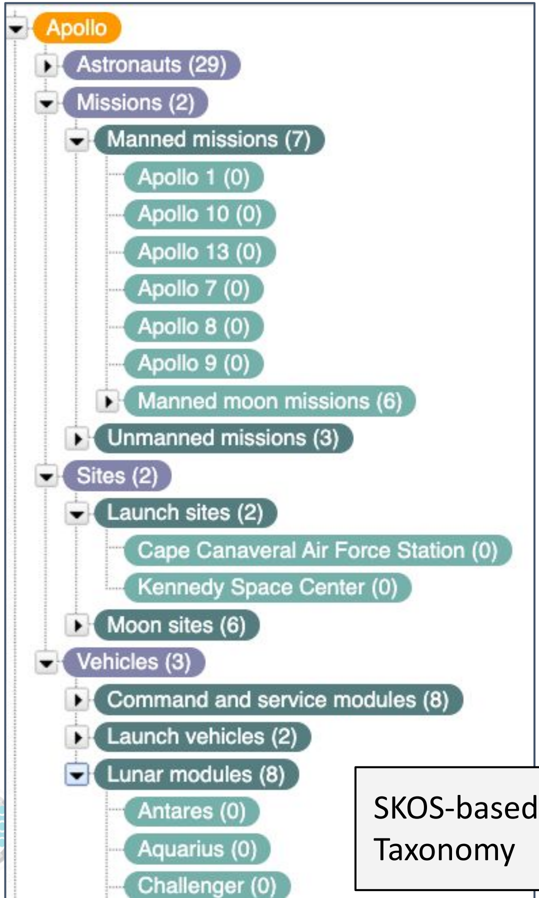
Individuals

Enter search string to filter list

- Antares
- Aquarius
- Challenger
- Eagle
- Falcon
- Intrepid
- Orion
- Snoopy

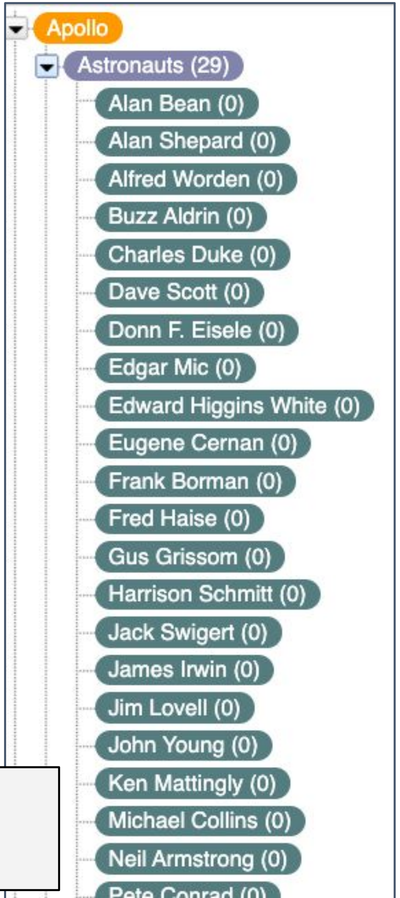
Screenshots from WebProtégé

Ontology Approaches



- Apollo
  - Astronauts (29)
  - Missions (2)
    - Manned missions (7)
      - Apollo 1 (0)
      - Apollo 10 (0)
      - Apollo 13 (0)
      - Apollo 7 (0)
      - Apollo 8 (0)
      - Apollo 9 (0)
      - Manned moon missions (6)
    - Unmanned missions (3)
  - Sites (2)
    - Launch sites (2)
      - Cape Canaveral Air Force Station (0)
      - Kennedy Space Center (0)
    - Moon sites (6)
  - Vehicles (3)
    - Command and service modules (8)
    - Launch vehicles (2)
    - Lunar modules (8)
      - Antares (0)
      - Aquarius (0)
      - Challenger (0)

SKOS-based Taxonomy



- Apollo
  - Astronauts (29)
    - Alan Bean (0)
    - Alan Shepard (0)
    - Alfred Worden (0)
    - Buzz Aldrin (0)
    - Charles Duke (0)
    - Dave Scott (0)
    - Donn F. Eisele (0)
    - Edgar Mic (0)
    - Edward Higgins White (0)
    - Eugene Cernan (0)
    - Frank Borman (0)
    - Fred Haise (0)
    - Gus Grissom (0)
    - Harrison Schmitt (0)
    - Jack Swigert (0)
    - James Irwin (0)
    - Jim Lovell (0)
    - John Young (0)
    - Ken Mattingly (0)
    - Michael Collins (0)
    - Neil Armstrong (0)
    - Pete Conrad (0)

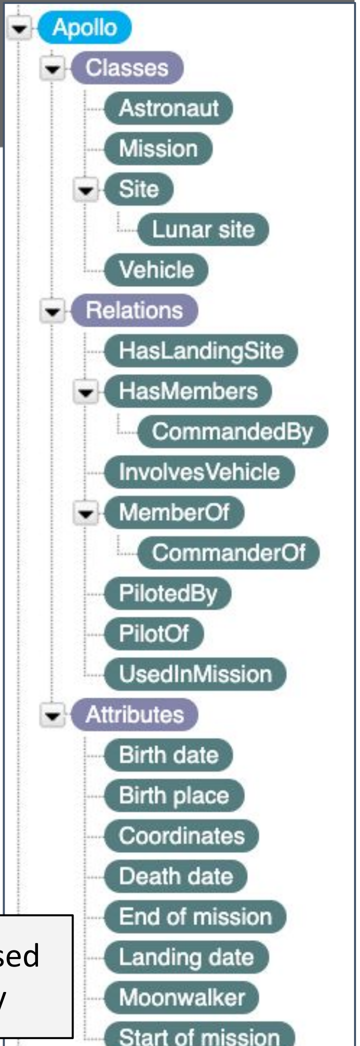
2. Taxonomy + ontology layer

All concepts (whether class-like or individuals) are maintained in the same hierarchy.

OWL classes, relations, and attributes are managed separately.

Example from PoolParty

Screenshots from Poolparty



- Apollo
  - Classes
    - Astronaut
    - Mission
    - Site
      - Lunar site
    - Vehicle
  - Relations
    - HasLandingSite
    - HasMembers
      - CommandedBy
    - InvolvesVehicle
    - MemberOf
      - CommanderOf
    - PilotedBy
    - PilotOf
    - UsedInMission
  - Attributes
    - Birth date
    - Birth place
    - Coordinates
    - Death date
    - End of mission
    - Landing date
    - Moonwalker
    - Start of mission

OWL-based Ontology

Ontology Approaches

Benefits of combining a high-level ontology as a semantic layer with a taxonomy

- Makes use of existing taxonomies, even multiple taxonomies
- Easier to model the ontology
  - Existing taxonomies provide a basis for knowledge modeling
  - No need to distinguish between sub-classes and individuals
- Supports expert specialization
  - Domain experts develop and maintain name authorities (instance entities)
  - Domain experts and/or taxonomists develop and maintain taxonomies
  - Ontologists develop and maintain the ontology
- More flexible and adaptable
  - The taxonomy changes more frequently than does the ontology
  - Taxonomies can easily be added
- Different purposes are served
  - The ontology is for modeling, reasoning, and analysis
  - The taxonomy is for tagging and information/data retrieval

Ontology Approaches

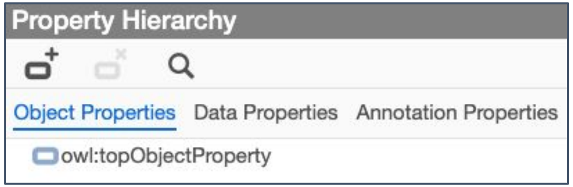
Not recommended approach to building ontologies:
Importing taxonomies into an OWL-based ontology or dedicated ontology tool

Why not?

- Taxonomy hierarchies get converted to class-subclass hierarchies.
- The class-subclass hierarchy in ontologies is of the hierarchical type generic-specific ("is a kind of") only.
- Taxonomies may contain other types of hierarchies: whole-part and generic-instance, but they are not indicated as such.
- Importing taxonomies into ontologies will incorrectly treat...
  - whole-part taxonomy relations as class-subclass relations
  - generic-instance taxonomy relations as class-subclass relations, not class-instance affiliations

Ontology Approaches

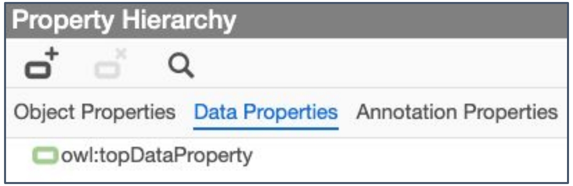
Problem from importing a SKOS taxonomy .ttls file into Protege



Property Hierarchy

Object Properties | Data Properties | Annotation Properties

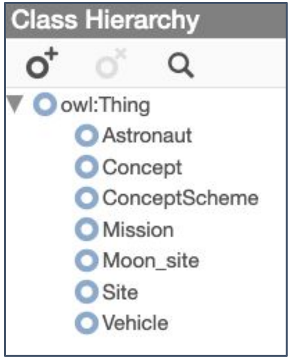
owl:topObjectProperty



Property Hierarchy

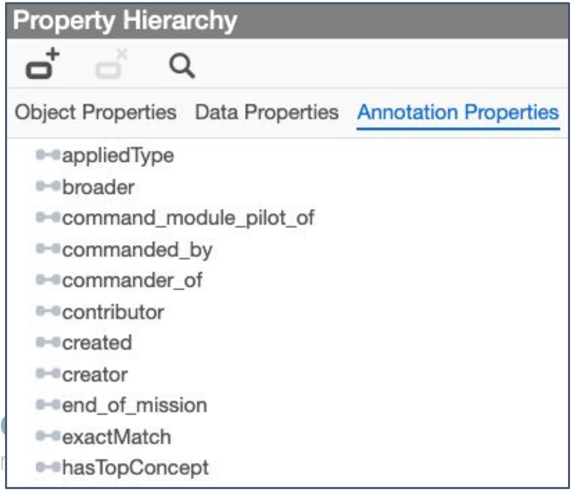
Object Properties | Data Properties | Annotation Properties

owl:topDataProperty



Class Hierarchy

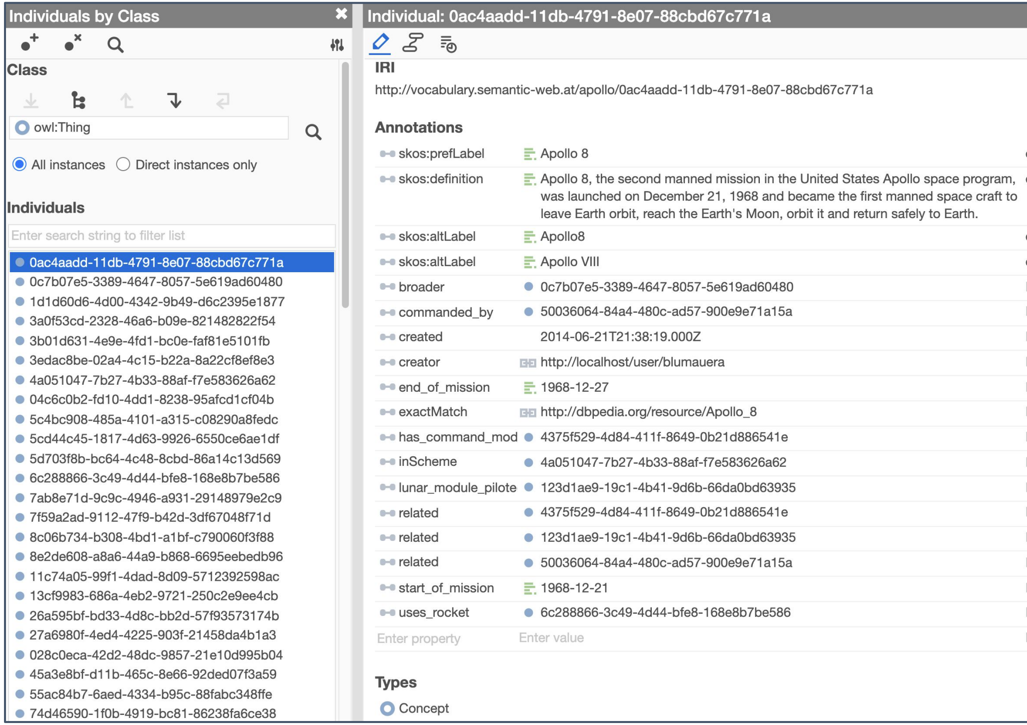
- owl:Thing
  - Astronaut
  - Concept
  - ConceptScheme
  - Mission
  - Moon_site
  - Site
  - Vehicle



Property Hierarchy

Object Properties | Data Properties | Annotation Properties

- appliedType
- broader
- command_module_pilot_of
- commanded_by
- commander_of
- contributor
- created
- creator
- end_of_mission
- exactMatch
- hasTopConcept



Individuals by Class

Class

○ owl:Thing

● All instances ○ Direct instances only

Individuals

Enter search string to filter list

- 0ac4aadd-11db-4791-8e07-88cbd67c771a
- 0c7b07e5-3389-4647-8057-5e619ad60480
- 1d1d60d6-4d00-4342-9b49-d6c2395e1877
- 3a0f53cd-2328-46a6-b09e-821482822f54
- 3b01d631-4e9e-4fd1-bc0e-faf81e5101fb
- 3edac8be-02a4-4c15-b22a-8a22cf8ef8e3
- 4a051047-7b27-4b33-88af-f7e583626a62
- 04c6c0b2-fd10-4dd1-8238-95afcd1cf04b
- 5c4bc908-485a-4101-a315-c08290a8fedc
- 5cd44c45-1817-4d63-9926-6550ce6ae1df
- 5d703f8b-bc64-4c48-8cbd-86a14c13d569
- 6c288866-3c49-4d44-bfe8-168e8b7be586
- 7ab8e71d-9c9c-4946-a931-29148979e2c9
- 7f59a2ad-9112-47f9-b42d-3df67048f71d
- 8c06b734-b308-4bd1-a1bf-c790060f3f88
- 8e2de608-a8a6-44a9-b868-6695eebedb96
- 11c74a05-99f1-4dad-8d09-5712392598ac
- 13cf9983-686a-4eb2-9721-250c2e9ee4cb
- 26a595bf-bd33-4d8c-bb2d-57f93573174b
- 27a6980f-4ed4-4225-903f-21458da4b1a3
- 028c0eca-42d2-48dc-9857-21e10d995b04
- 45a3e8bf-d11b-465c-8e66-92ded07f3a59
- 55ac84b7-6aed-4334-b95c-88fabc348ffe
- 74d46590-1f0b-4919-bc81-86238fa6ce38

Individual: 0ac4aadd-11db-4791-8e07-88cbd67c771a

IRI

http://vocabulary.semantic-web.at/apollo/0ac4aadd-11db-4791-8e07-88cbd67c771a

Annotations

skos:prefLabel	Apollo 8
skos:definition	Apollo 8, the second manned mission in the United States Apollo space program, was launched on December 21, 1968 and became the first manned space craft to leave Earth orbit, reach the Earth's Moon, orbit it and return safely to Earth.
skos:altLabel	Apollo8
skos:altLabel	Apollo VIII
broader	0c7b07e5-3389-4647-8057-5e619ad60480
commanded_by	50036064-84a4-480c-ad57-900e9e71a15a
created	2014-06-21T21:38:19.000Z
creator	http://localhost/user/blumauera
end_of_mission	1968-12-27
exactMatch	http://dbpedia.org/resource/Apollo_8
has_command_mod	4375f529-4d84-411f-8649-0b21d886541e
inScheme	4a051047-7b27-4b33-88af-f7e583626a62
lunar_module_pilote	123d1ae9-19c1-4b41-9d6b-66da0bd63935
related	4375f529-4d84-411f-8649-0b21d886541e
related	123d1ae9-19c1-4b41-9d6b-66da0bd63935
related	50036064-84a4-480c-ad57-900e9e71a15a
start_of_mission	1968-12-21
uses_rocket	6c288866-3c49-4d44-bfe8-168e8b7be586
Enter property	Enter value

Types

○ Concept

All properties are grouped as “Annotation” properties.

Display of individuals is by URI ID, not by name.

4. Ontologies Applied to Taxonomies

Re-using ontologies, selecting from ontologies, creating custom schemes, and applying them to existing taxonomies

Reusing Ontologies

Why reuse ontologies?

- Many published taxonomies exist.
- Many ontologies are free and listed in directories.
- Many ontologies are intended for reuse - all upper ontologies and many domain ontologies.
- The idea of ontology reuse often assumes application of a generic ontology to specific instances within custom taxonomies (exception: detailed biomedical ontologies).
- Reuse of an ontology need not be complete, but can be selective of parts of an ontology, as applicable.
- Reuse of ontologies or parts of ontologies can be from internal proprietary ontologies, and not just published ontologies.

Reusing Ontologies

Tips for reusing ontologies

- Domain ontologies are created for a specific domain context, which likely does not match yours exactly > Be selective
- Use existing ontologies as a starting point, to select from and add to
- You may select parts of more than one ontology, to create a new custom ontology as a mashup
- A tool, such as PoolParty, that supports creating a “custom scheme” from selected parts of of ontologies is a good option

Applying Ontologies to Taxonomies

Extending SKOS concepts to be part of an ontology

- Ontology class labels correspond/match the SKOS concept scheme or concept labels to which they will be applied
  - The ontology “layer” is not an upper hierarchical layer, but an overlay to the higher levels of the SKOS project.
  - Tip: consider using singular for ontology class names and plural for SKOS concept names
- There is no dilemma in determining if an entity is an individual/instance or a class, since the ontology layer comprises only classes.

Applying Ontologies to Taxonomies

Extending SKOS concepts to be part of an ontology

- An ontology or custom scheme is applied/linked to the taxonomy project or to a specific concept scheme.
- Classes are applied to the levels of concept schemes, top concepts, or broad-level concepts.
- Class properties (relations and attribute types) are then inherited by all narrower concepts.
- Relevant relations and attributes are available for all concepts in the taxonomy, based on their class assigned to their broader concept or concept scheme.
- Instantiating a relations between a pair of specific concepts or adding values to attributes must still be done manually, as it would have to be done in a detailed ontology.

Applying Ontologies to Taxonomies

Demo

of applying an ontology to a taxonomy

in a taxonomy/ontology tool

PoolParty

5.
Knowledge Graphs

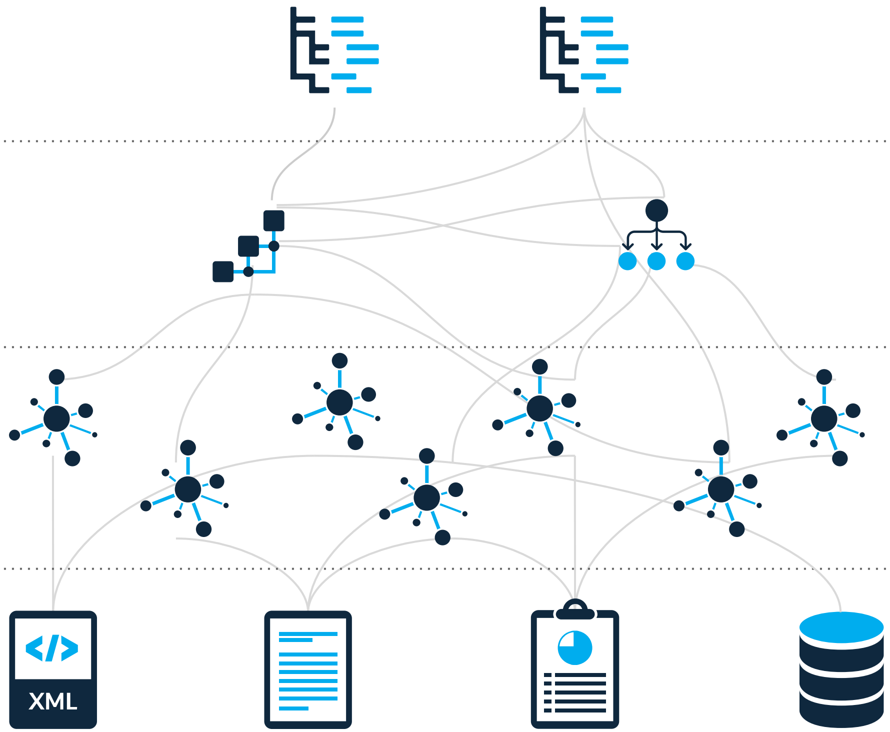
Building a knowledge graph

Creating a Knowledge Graph

What is a knowledge graph?

- “A knowledge graph is a model of a knowledge domain”
- A knowledge graph represents **unified information across an organization,** enriched with context and semantics that are meaningful across information silos.

Knowledge graph and applications



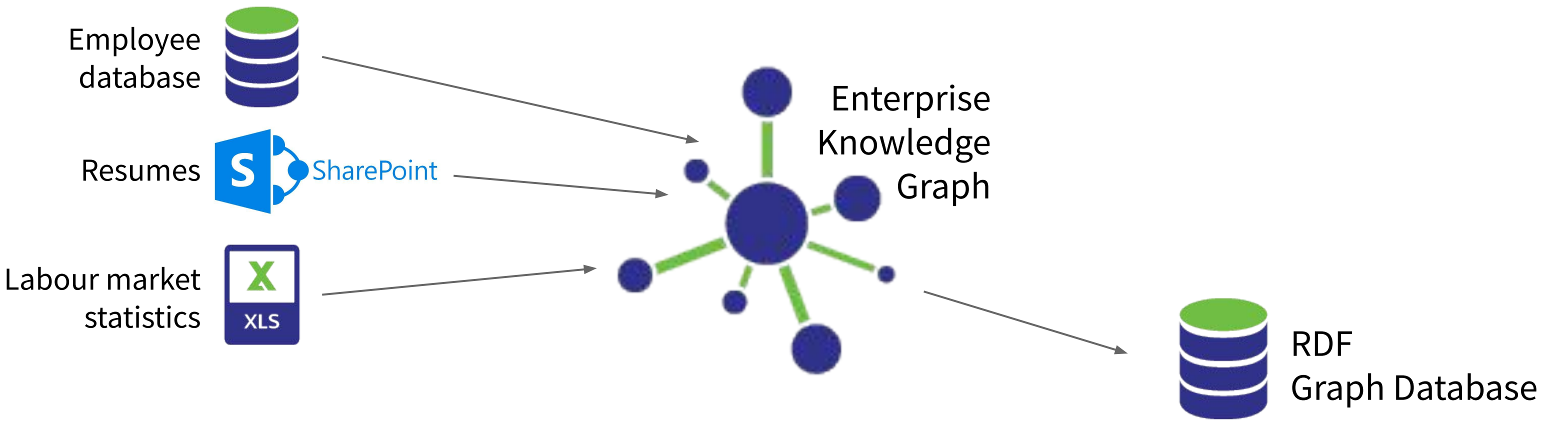
Applications (augmented AI)

Ontology and controlled metadata

(Virtual) Data Layer

Content & Data Layer

Knowledge graphs for data integration and analytics



- Metadata enrichment, linked data, text mining, entity-centric search, agile reporting

Building a knowledge graph

Data-centric approach

- Understand the data and purposes of the knowledge graph
  - Enlist domain experts
- Try to create a model that is **as simple as possible, but no simpler.**
  - A model should be independent of a specific application
  - But only the necessary parts of the domain should be modelled.

Example: Elevator specifications

Different perspectives

- Functional: Number of floors, dimensions, service capacity
- Engineering: Dimensions, power train, electrical, speed
- Purchasing: Manufacturer, price, maintenance options
- Regulatory: Standards certifications, requirements

Start small

➢ Only the relevant perspective(s) should be modeled

➢ Only relevant parts of a perspective should be modeled

Ontology + vocabularies = knowledge model

Division of labor:

- Model controlled vocabularies in SKOS
  - Taxonomies
  - Term lists
  - Name authorities
- Model **entity types, attributes** and semantic (non-hierarchical) **relationships** in OWL

Modelling a domain: Step by step

Goals

Reduce complexity. Make immediate progress

Steps

1. **Identify controlled vocabularies.** Model in SKOS
    - Example: Cooking methods, ingredients, etc.
2. Identify major Entities and Relationships
    - Traditional “Entity-Relationship (ER) modeling”
3. Translate E-R model to OWL
4. Add essential and task-relevant properties to the entities

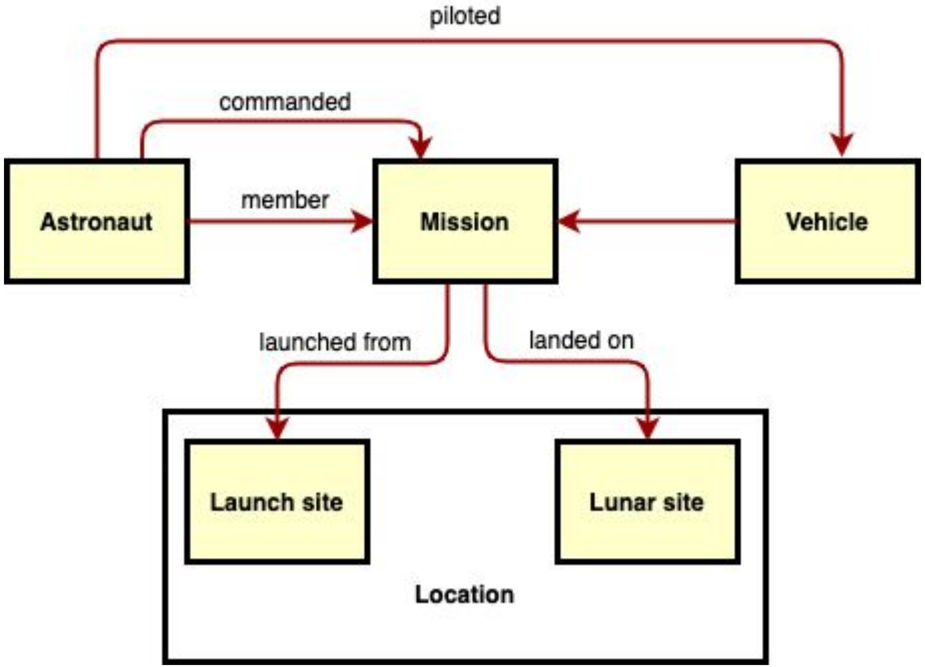
Example: Apollo space program

- 17 missions to space
- 6 moon landings
- Duration: 1961-1972
- Dramatis personae:
  - Launch vehicle, command module, landing module
  - Astronauts
  - Launch and landing sites

Example: Identify controlled vocabularies

- Astronauts
- Launch vehicles
- Lunar locations
- etc.

Example: Entity-Relationship model



Example: Express in OWL

Classes

- Astronaut
- Mission
- Location
    - Lunar site
    - Launch site
- Vehicle

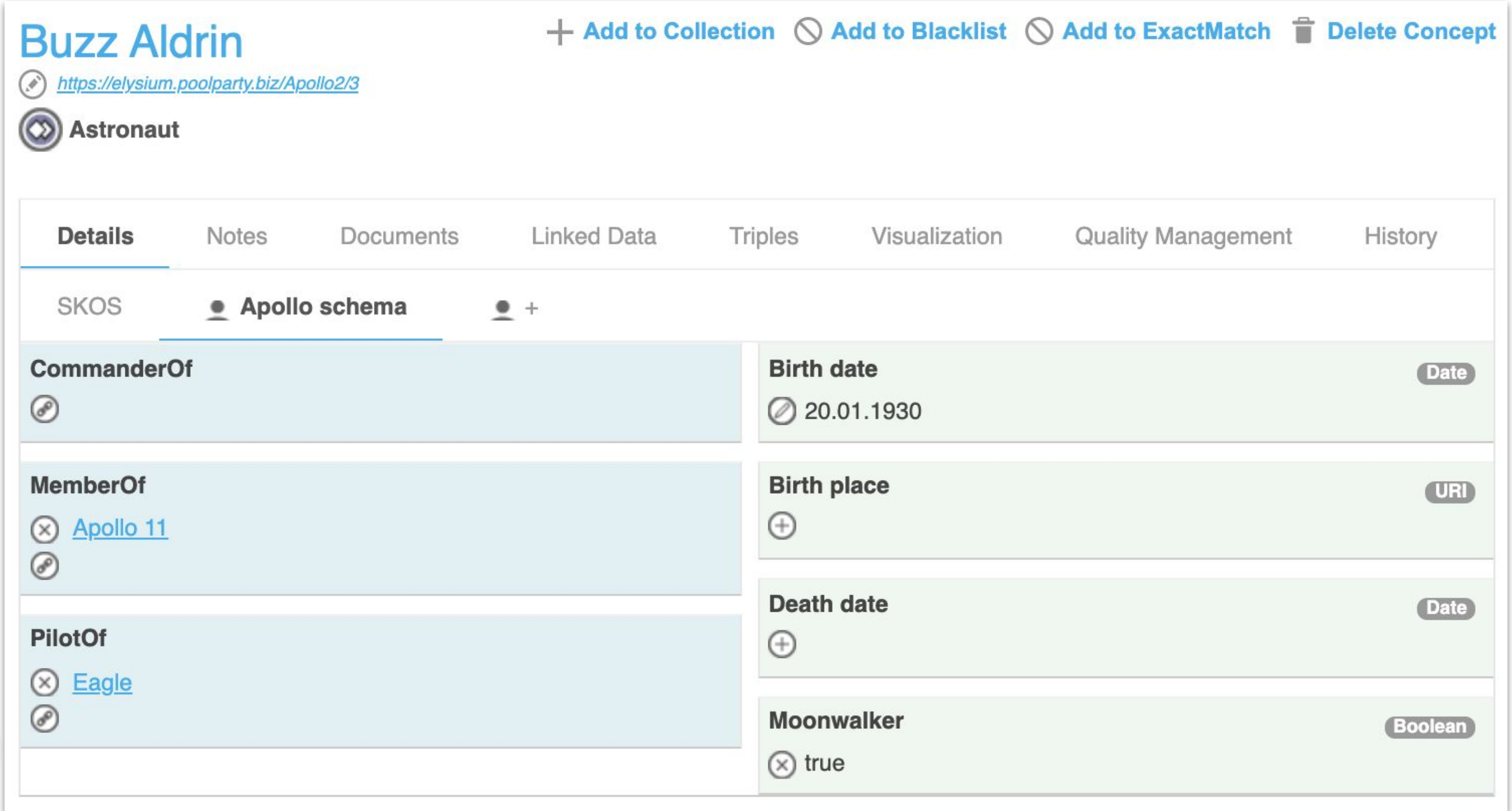
Relationships

Astronaut

- member of Mission
- piloted Vehicle

Attributes

Example: Ontology as extended data model



Buzz Aldrin

https://elysium.poolparty.biz/Apollo2/3

Astronaut

CommanderOf

MemberOf
Apollo 11

PilotOf
Eagle

Birth date (Date)
20.01.1930

Birth place (URI)

Death date (Date)

Moonwalker (Boolean)
true

6.
Ontology Modeling Tips

DataCentric Architecture Forum

Tips in ontology modeling

Ontology Design Tips

- Reuse **Relations** between more than one pair of Classes.

  Example: Relation inverse pair: requires / needed for is used between:

  Class: Job role and Class: Certification

  and

  Class: Job role and Class: Skill

➢ Consider whether classes that share a relation are subclasses of a common class

- Reuse **Attributes** among more than one Class.

  Example: Attribute: phone number

  Class: Person

  and

  Class: Company

Ontology Design Tips

Determining whether a property should be an Attribute or a new Class + Relation

For example:

- Class: Business name
- How to manage property of “location”?

  a. Attribute: Location: text field in which to entry the address or geo-location coordinates
     OR

  b. Class: Location to correspond to a term list or a hierarchical taxonomy concept scheme of countries, states/provinces, and cities
     And semantic relation: locatedIn (and possibly inverse: locationOf)

- Consider: Is there a use case for looking up Business Names by Location?

Ontology Design Tips

Keep it simple and high-level

Start with creating one candidate class per SKOS concept scheme

- Create semantic relations between pairs of classes
- Create attributes sufficient for business needs, not for everything possible.
  - ➢ Not all classes need attributes, especially generic topical classes
- Remove candidate classes that end up being neither the domain nor range of any property (relation or attribute).

Identify use cases for linking other categories of concepts: to find X by Y.

- Create additional classes and relations for these uses cases.
- Determine if any of these can be established as subclasses of existing classes, if they share all relation types and attribute types.

Conclusions

“Ontology” can have multiple meanings

1. A fully built-out, semantically rich knowledge organization system (like a taxonomy or thesaurus), with the addition of semantic relationships and attributes, managed by the assignment of classes.
2. A somewhat generic knowledge model, comprising entities and properties, intended to be applied to or extended by means of linking to taxonomies (and possibly name authorities)
3. The *combination* of the generic knowledge model and the linked taxonomy(s) and name authorities.
4. A semantic model built on OWL

Questions / Contact

Alexis Dimitriadis
Data and Knowledge Engineer
Semantic Web Company GmbH
Neubaugasse 1, Top 8
1070 Vienna
Austria
+43-1-4021235
alexis.dimitriadis@semantic-web.com
www.linkedin.com/in/alexisdimitriadis

Heather Hedden
Data and Knowledge Engineer
Semantic Web Company Inc.
One Boston Place, Suite 2600
Boston, MA 02108
USA
(857) 400-0183
heather.hedden@semantic-web.com
www.linkedin.com/in/hedden

Semantic Web Company www.semantic-web.com

PoolParty software www.poolparty.biz