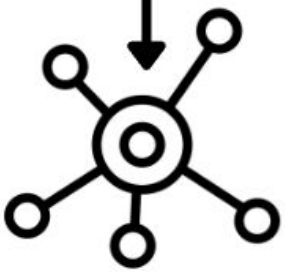
Knowledge Engineering

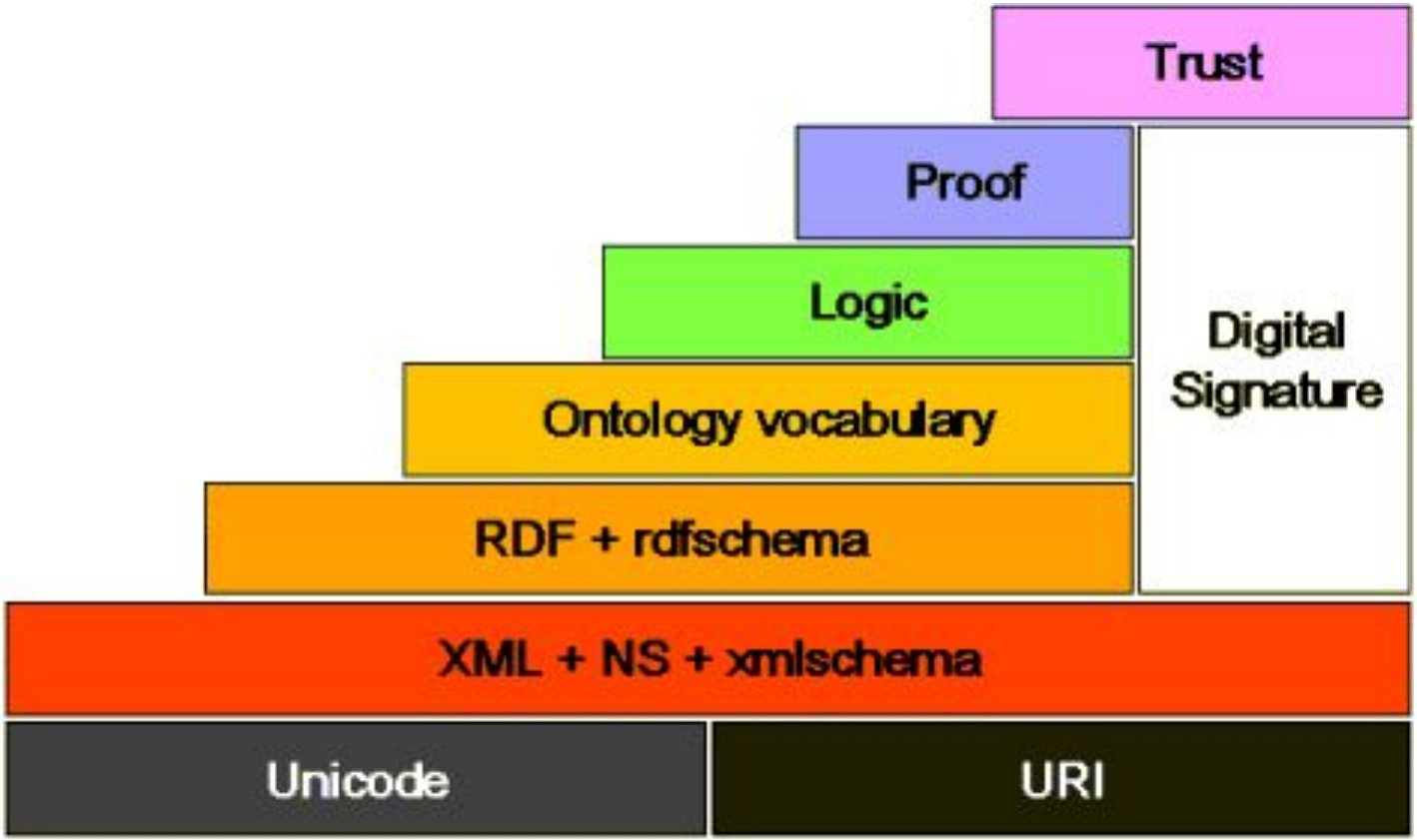
Nicholas Latham, Van Quoc Huy Vo, Chris Nodel

Traditional Web

Semantic Web

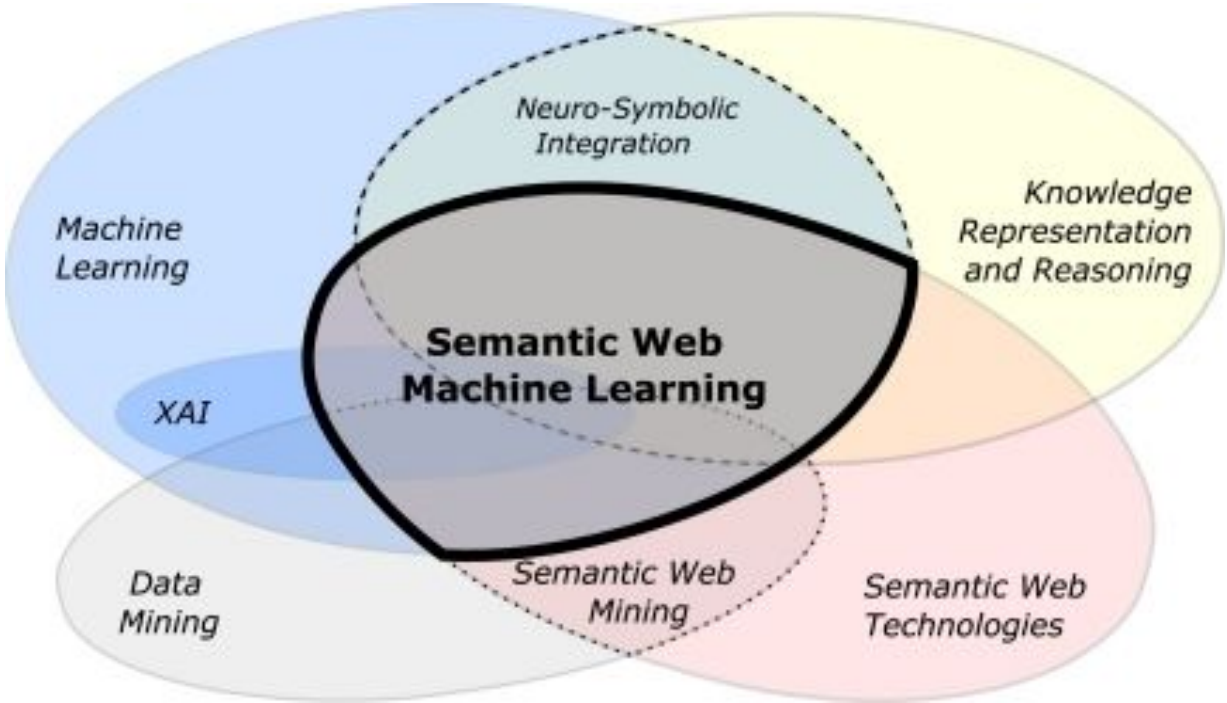


Semantic web architecture



Semantic vs AI

Semantic vs AI



https://dl.acm.org/doi/10.1145/3586163

https://susodigital.com/thoughts/decoding-the-semantic-web-before-ai-does-it-for-you

How to improve your Semantic Web

- Build topic cluster
- Focus on entities, not just Keywords
- Use Schema Markup to Structure Meaning
- Map content to search intent
- Analyse Google’s Knowledge Graph
- Strengthen internal linking with Entity Relationships

Team Questions

- Which facets from this early concept of the Semantic Web are still applicable today, and why?
- Which facets from this early concept have not yet come to fruition or have been abandoned, and why?
- Have there been any more major developments in Semantic Web research since this article was published in 2021?
- How can we reconcile the academic research with industry applications?

Industry-Scale Knowledge Graphs: Lessons and Challenges

By Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jaimie Taylor

Knowledge Graph applications in Industry

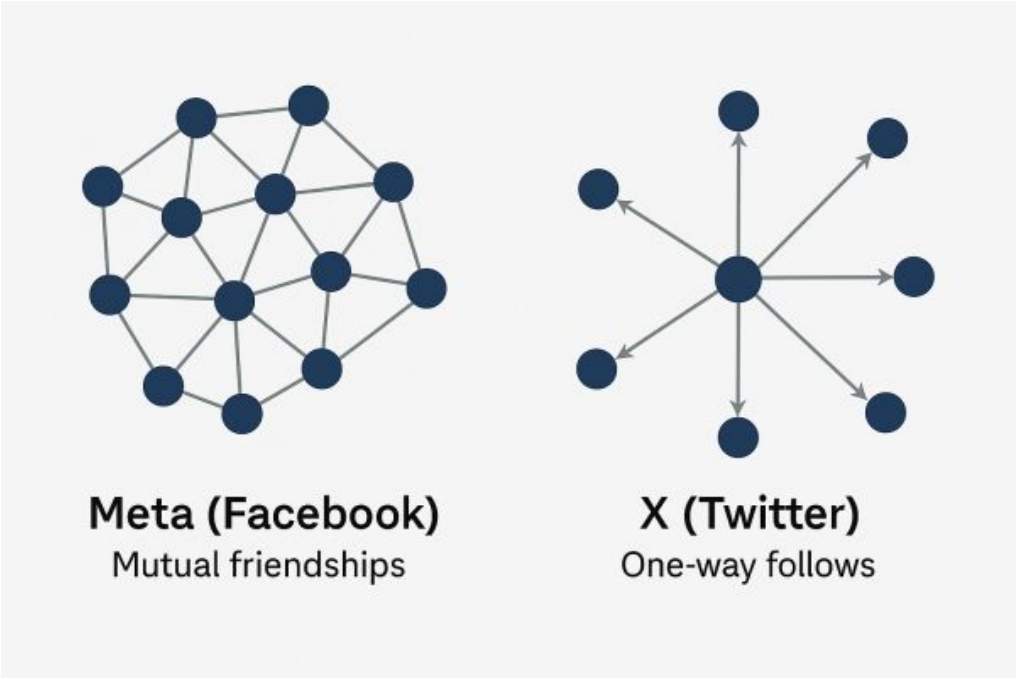
- eBay
- Facebook
- Google Knowledge Graph (KG)
- IBM Watson
- Microsoft

eBay Product Knowledge Graph

- Designed to improve shopping experience
- Inform and suggest items to shoppers
- Help sellers with market research
- Product Research Tool and mobile app
- Independent KG-generation API also exists on Github

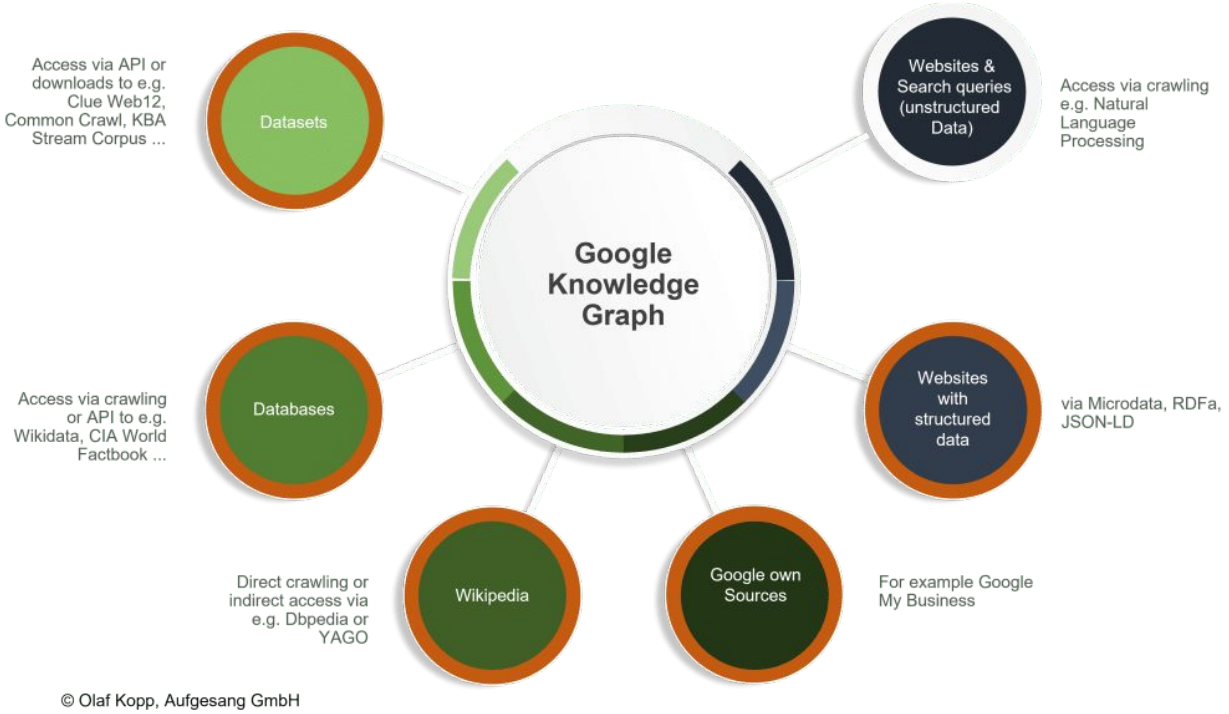
Facebook social graph

- World’s largest social network dataset
- Designed to connect people and their interests
- Rebuilt on a daily basis to always stay up-to-date
- Graph API currently available for public development



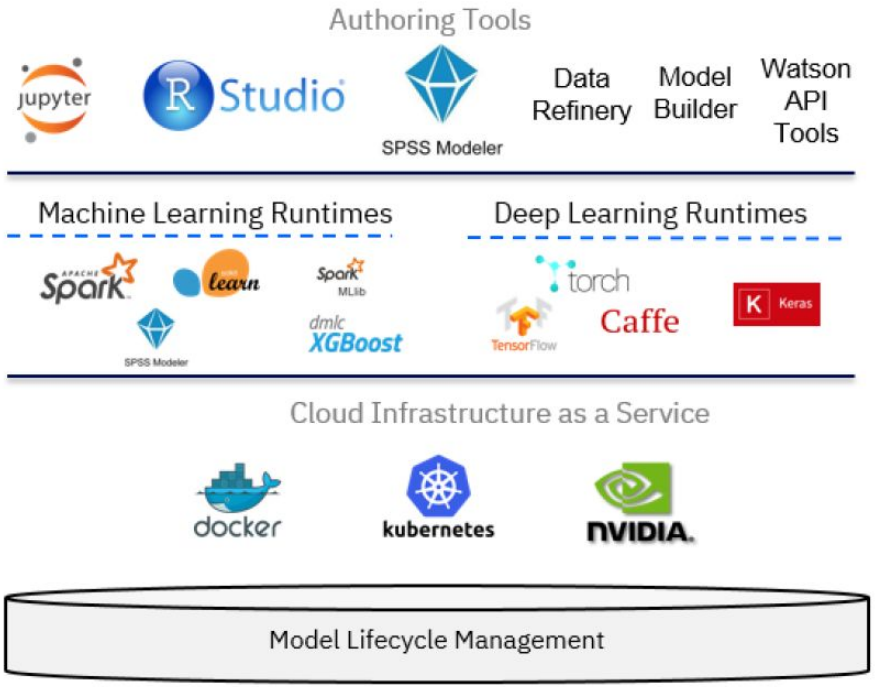
Google Knowledge Graph

- Groundbreaking Knowledge Graph
- Multilayer design
- Designed to improve information from Google searches
- API available for public use



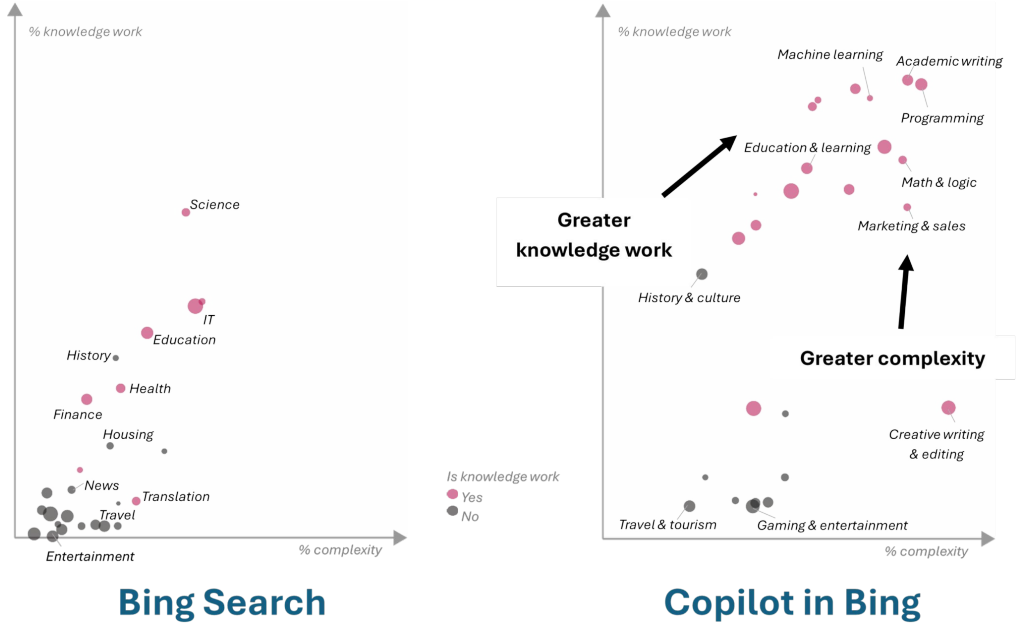
IBM Knowledge Graph Framework

- Designed to work with IBM Watson Discovery Services
- Users can build their own knowledge graph with the framework
- Paid service



Microsoft Bing knowledge graph

- Similar in purpose to Google Knowledge Graph
- Research-driven
- Academic graph
- LinkedIn graph
- Microsoft is currently working on AI solutions
- Copilot has become well-known since this paper was published



Paper #4 Team Questions and Answers

- Is there public access to the underlying axioms of these knowledge graphs?
  - Not openly, but the APIs and some (possibly older) versions of the code are available.
- How did the engineers of these companies ascertain the structure and internal design of their knowledge graph based on their use case?
  - In most cases, there was a defined business purpose, and the engineers worked with that in mind.
- What are some possible combinations of the different companies' knowledge graph concepts?
  - Bing and Google could draw from some of the other graphs. A truly strong connection would require corporate cooperation.
- What notable developments have occurred since this article was published?
  - Since multiple APIs are available, other companies and private users can develop their own KGs more easily.
  - Some of the code is published on Github, and the KGs mentioned in the paper are being updated relatively frequently.

Modular Ontology Modeling

Cogan Shimizu, Karl Hammar, Pascal Hitzler

What is Modular Ontology Modeling

- Modular Ontology Modeling is a methodology for modeling an ontology where instead of a more traditional top down taxonomical approach, it instead uses modules to form a more bottom up construction, taking these more encapsulated structures and putting them together
- Modules in this case refer to portions of the ontology that define key notions and the key elements which define these notions.
- Modules allow instead of a more tree like construction of an ontology to something more akin to a puzzle where pieces can be added

Why Modular Ontology Modeling

Modular Ontology Modeling methodology was developed as a result of a lack of reuse of currently existing ontologies

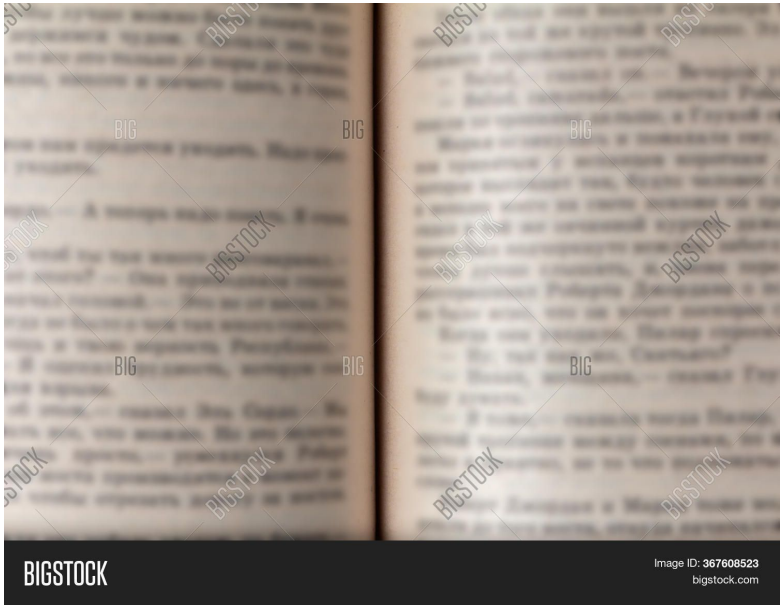
The reasons for this are outlined in four major reasons

Major reasons

1. Differing representational granularity: Representational granularity refers to modeling decisions regarding the level of detail of the ontology
    a. There can be difficulties between the granularity of data and the ontology
    b. Different use cases can call for different levels of granularity within the same ontology
        i. This may mean modifications to the ontologies for these varying use cases but since traditional ontologies are presented as more singular entities making these modifications can be difficult

Major reasons

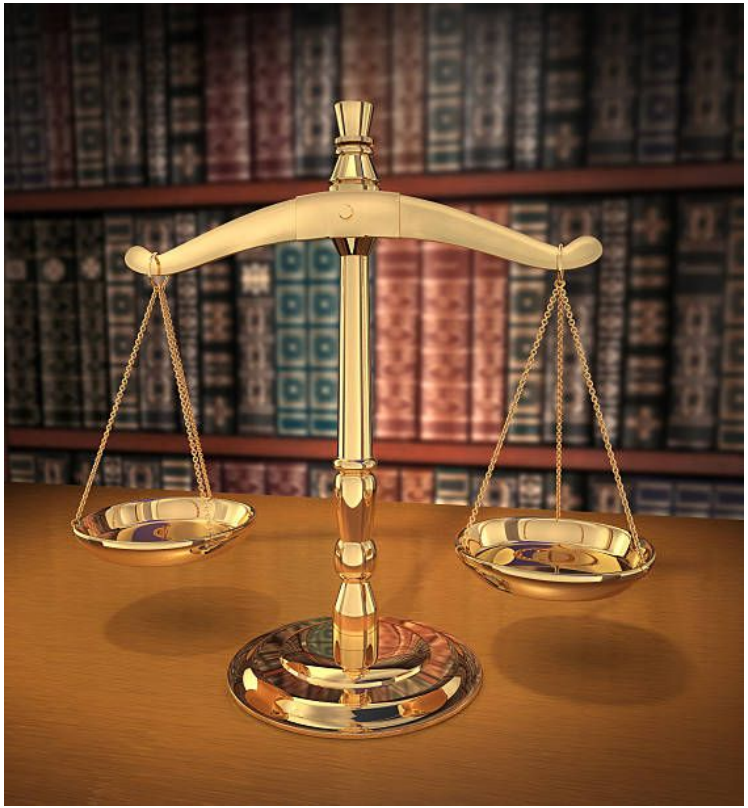
1. Lack of conceptual clarity:
    a. Due to the rather subjective nature of this concept it would be best to describe conceptual clarity as the readability of an ontology or rather how understandable is the design of the ontology for domain experts
    b. This does not just include intuitive naming conventions but also an intuitive structure for the schema
    c. A good ontology should strive for a high conceptual clarity. This is not always the case however.



Major Reasons

Good modeling principles: adherence to good modeling principles should result in easy reusability of ontologies:

- However, current considerations for good modeling principles are not followed by all ontologies
- Not only that but defining the precise minutiae of good modeling principles is still more of an art than a science and does not lend itself to more exacting instruction



Major reasons

Limited re-use support in available tooling:

- In order to reuse an ontology a suitable reuse method must first be selected and implemented
  - However due to the lack of comprehensive tools to support this even with guidance from the authors this process is error prone and time consuming
- Re-use also makes it so your ontology follows the structure and logic of a third party and an evolving ontology may need to resolve arising conflicts between the ontology and re-used resources. Current tooling makes this process non trivial.

How does Modular Ontology Modeling Solve these issues

- Modularity allows the author to create their ontology via a divide and conquer strategy which allows the author to have a more by the concept look at the model rather than strictly by the hierarchy and subclass conceptualization of the taxonomic model.
    - This helps deal with incredibly large ontological models as rather than dealing with the singular structure it can be handled in a more piecewise manner.
- Modules follow certain templates which should make it easier to follow good modeling principles as well as standardizing them.
- Because modules can connect to each other in a multitude of ways it can often lead to better concept clarity

Modular Ontology Modeling Workflow

Describe use cases and gather possible data sources

The use case or problem should be described. This is also where potential data sources (i.e. databases) should be looked at

Gather competency questions

Come up with potential queries related to the use case. This helps to refine the use case and check the adequacy of data

Define Key Notions

Based on your data, use case(s), and competency questions, define key notions/concepts for your ontology

Identify Existing Ontology Design patterns

Based on your key notions look at currently patterns that can be used as a template to start mapping out your key notions

Modular Ontology Modeling Workflow

Create Schema Diagram for modules

Based on your chosen templates you then begin making the Schema for your modules individually. Going by increasing complexity or by centrality of concept helps

Set up documentation and determine axioms

Documentation is important for the reusability of the ontology. A rigorous documentation of the schema and axioms of the modules is a must

Create ontology schema

You can then start putting your modules together to create your ontology schema. Use axioms to bridge the modules together.

Reflect on entity naming and all axioms

Now is the time to check your naming conventions and axioms, to see if it follows propers standards and follow the internal logic of the axioms.

Modular Ontology Modeling Workflow

Create OWL Files

This is the implementation of your ontology into the OWL file. This is the last step as most of the work is related to the conception of the ontology.

Questions

1. How would modularity handle more inherently ambiguous domains, such as say certain medical or biological fields?
2. Any new notable developments from the authors since the paper was published?
3. Are there more advanced ontology development tools than Protégé-2000, and what are they?
4. Are there any more tips for developing ontologies beyond this guide?

Answers

1. This question actually approaches the study of fuzzy ontologies. Fuzzy ontologies use Fuzzy set logic in order to better define more imprecise descriptors. Protege does have a fuzzy logic extension known as Fuzzy protege which can be used to implement fuzzy ontologies [1]
2. I was unable more recent developments from the authors since 2023. However this paper as well as its authors were cited in a 2024 paper using MOMo to more effectively manage the large amounts of data used by Digital Product Passports. [2]
3. The two most current versions of Protege are Web Protege and Protege Desktop [3]
4. There is a MOMO tutorial by Shimizu, Hitzler, and Krisnadhi [4]

references

1. Microsoft Word - IPC 09.doc
2. https://tinyurl.com/y43724zs
3. Protégé
4. 2020-mom-tutorial.pdf