Modular Ontology Engineering

Cogan Shimizu
Data Semantics Laboratory, Kansas State University
https://daselab.cs.ksu.edu
http://coganshimizu.com
The Praxis of Knowledge Graphs

- How are they used?
- Development Tools
- (Re)Use
- Useful semantics
- Structural and Usage Patterns
- Automatic knowledge graph construction and population

The Data Semantics Lab
www.daselab.cs.ksu.edu/
How can we develop a highly reusable knowledge graph?
Knowledge Graphs as Resources

Knowledge Graphs are poised to be a significant disruptor in the public and private sectors [1].

- A new “common” resource for information gathering and sharing
- Enables data visualization & discovery
- Crowdsourcing Potential
- Human & Machine Readability
Knowledge Graph (Re)use

Knowledge graphs can be difficult to (re)use without a schema or may be difficult to adapt outside of their original use case.

- Schema is important for knowledge graph (re)use
- Good Schema allows for the (re)use of knowledge graphs outside their original use case [2]
Kansas State University hasPresident Richard Myers. Richard Myers hasBirthdate 03/01/1942. Richard Myers hasEducation Air Command and Staff College. Kansas State University hasStudents 18,171.
A Tiny Knowledge Graph Schema

- Organization
  - hasStudents
  - hasPresident
- Person
  - hasBirthdate
  - hasEducation
- Organization
- date

Number
A Tiny Knowledge Graph Schema
Properties of Good KG Schema

A Knowledge Graph schema should allow for the incorporation of contextual information.

- From the previous example, we can add an extra node that allows us to capture contextual information about the enrollment year.
Properties of Good KG Schema

A Knowledge Graph schema should allow for the incorporation of contextual information.

- From the previous example, we can add an extra node that allows us to capture contextual information about the enrollment year.
Properties of Good KG Schema

A Knowledge Graph schema should allow for the incorporation of contextual information.

- We can do this for the President’s tenure, too.
A (Better) Tiny Knowledge Graph Schema

- **Organization**
  - hasEnrollment
  - providesRole

- **Enrollment**
  - hasEnrollment
  - hasEnrollmentYear
  - number

- **PresidentRole**
  - hasEducation
  - hasStartDate
  - number

- **Person**
  - hasBirthdate
  - date
More Properties of Good KG Schema

A Knowledge Graph schema should be easily adaptable to new or changing use-cases, reduce ambiguity, and be well documented.

- Easy to Adapt and Maintain
- Human Readable Documentation
- Rigorous Definitions
The Role of Formal Logics and Ontology

Formal logic can be used to express a rigorous formulation of human conceptualization in a computationally tractable manner.

- Rigorous and Precise
- Tractable
- W3C Standard: OWL2

Examples of Axioms in Description Logic.

1. $A \subseteq B$
2. $A \cap B \subseteq \bot$
3. $\exists R.T \subseteq A$
4. $\exists R.B \subseteq A$
5. $T \subseteq \forall R.B$
6. $A \subseteq \forall R.B$
7. $A \subseteq R.B$
8. $B \subseteq R^-.A$
9. $T \subseteq \leq 1R.T$
10. $T \subseteq \leq 1R.B$
11. $A \subseteq \leq 1R.T$
12. $A \subseteq \leq 1R.B$
13. $T \subseteq \leq 1R^-T$
14. $T \subseteq \leq 1R^-A$
15. $B \subseteq \leq 1R^-T$
16. $B \subseteq \leq 1R^-A$
17. $A \subseteq \geq 0R.B$
Why Modular Ontologies?

A modularly structured schema allows for rapid adaptability of the knowledge graph, allowing it to evolve in the face of new data or facilitate its reuse [2].

- A bridge between human conceptualization and data
- “Plug and Play” Configuration
Modular Ontology Engineering (briefly)

A methodology for developing highly reusable knowledge graphs that emphasizes a pattern-based approach using graphical representations and systematic axiomatization [2,3,4].

- Use-case driven
- Assumes an “Empirical or Data Reality”
- Ontology Design Patterns
- Create Modules from Patterns
- Modules are the primary component
Schema Diagrams

Informal, but intuitive, graphical representations depicting the relationships between classes in an ontology [5].

- Yellow Boxes := The Key Notion
- Blue Boxes := “Hidden Complexity”
- Arrows := Properties / Relations
- Open Arrows := Subclass Relation

A Schema Diagram for the “Agent Role” ODP
Systematic Axiomatization

By examining each node-edge-node construct in a schema diagram, we can produce a reasonable axiomatization [6].

A node-edge-node construct.

1. $A \subseteq B$
2. $A \cap B \subseteq \perp$
3. $\exists R.T \subseteq A$
4. $\exists R.B \subseteq A$
5. $T \subseteq \forall R.B$
6. $A \subseteq \forall R.B$
7. $A \subseteq R.B$
8. $B \subseteq R^- . A$
9. $T \subseteq \leq 1 R . T$
10. $T \subseteq \leq 1 R . B$
11. $A \subseteq \leq 1 R . T$
12. $A \subseteq \leq 1 R . B$
13. $T \subseteq \leq 1 R^- . T$
14. $T \subseteq \leq 1 R^- . A$
15. $B \subseteq \leq 1 R^- . T$
16. $B \subseteq \leq 1 R^- . A$
17. $A \subseteq \geq 0 R . B$

17 common axioms that may be conveyed by a node-edge-node construct.
Ontology Design Patterns

*Tiny, self-contained ontologies that solve a domain-invariant modeling problem using community identified best practices [7].*

- Agent and TemporalExtent boxes indicate hidden complexity.
- Anything may provide an AgentRole.
- AgentRoles are a sort of property reification that allows us to capture “states of being,” e.g.

A Schema Diagram for the “Agent Role” ODP [8,9].
The Methodology

1. Define use case or scope of use cases.
2. Make competency questions while looking at possible data sources and continue scoping the problem and use-case(s).
3. Identify key notions from the data and the use case and identify which pattern should be used for each. Use “stubs” as necessary.
4. Instantiate these key notions from the pattern templates, then adapt the result as needed, to create modules. Develop the remaining modules from scratch.
5. Systematically add axioms for each module.
6. Assemble the modules and add axioms which involve several modules.
7. Reflect on all entity names and possibly improve them. Check module axioms whether they are still appropriate after putting all modules together.
8. Create OWL files.
CoModIDE

The “Comprehensive Modular ontology Integrated Development Environment” supports a graphical, pattern-based modeling paradigm as a Protégé plugin [10, 11].

1. The Graphical modeling canvas that supports drag and drop actions.

2. The pattern library (MODL) [x]

3. Configuration view

www.comodide.com
Further Supporting Infrastructure

- **OPLa Annotator**
  
  *Quickly and easily add OPLa Annotations to your modules [12].*

- **ROWLTab**
  
  *Create OWL axioms from existential rules [13].*

- **OntoSeer**
  
  *Recommends vocabularies, classes, axioms, and ODPs to reuse [14].*

- **ODPReco**
  
  *Recommends ODPs based on description, CQ’s and axioms [15].*
Examples of Modular Ontologies

- The Enslaved Ontology
  A Domain Ontology for capturing records of the Historic Slave Trade [16].
- A Domain Ontology for Instructions
  An ontology that represents instructions in Cognitive Science tasks [y].
- Recipe Tutorial
  A basic ontology for teaching modular ontology modeling [4].
- Chess Game Tutorial
  A basic ontology for capturing Chess Game Records in PGN [3].
- GeoLink Modular Ontology
  An Ontology from the EarthCube Project that captures oceanographic cruises [18].
PRAXIS- The Semantic Web in Practice: Tools & Pedagogy

- Find effective ways of expanding the expertise and knowledge of:
  - Developers already in industry
  - Domain experts
  - Students

- Improve the Developer Experience (DX)

- Establish the importance of Semantic Web technologies
  - Incorporating them into a technology stack or workflow

- Informal (but refereed) submissions concerning (Academia, Industry) x (Tools, Pedagogy)
  - Single Page abstracts
  - Reports (max 4 pages)
  - Extended Abstracts & Course Material

- Cogan Shimizu
- Rafael Goncalves
- Juan Sequeda
- Ruben Verborgh

www.praxis-workshop.com
Thanks!

Please direct any offline questions to cogan.shimizu@coganshimizu.com
References


[2] Pascal Hitzler, Cogan Shimizu: Modular Ontologies as a Bridge Between Human Conceptualization and Data. ICCS 2018: 3-6


References

References