Designing Realistic Defeasible Ontologies

Victoria Chama

University of Cape Town

Abstract

Ontologies play a crucial role in knowledge representation and automated reasoning on the web, defining concepts and relationships within domains. However, the ontology community lacks effective tools for expressing defeasibility, while the reasoning community struggles with realistic defeasible ontologies. This research aims to bridge this gap by providing practical ontologies with defeasible features across various domains and designing a methodology for creating them. By employing a preferential approach to non-monotonic reasoning and leveraging existing Protégé plugins, the expected outcomes include defeasible ontologies for evaluating defeasible reasoners and a reusable methodology for developing such ontologies.

Research Objectives

The study investigates the following objectives:

- **Objective 1: Design and implement realistic defeasible ontologies** in four domains, incorporating multiple levels of exceptionality.
- **Objective 2: Evaluate ontology engineering tools:** (such as Protégé with DIP plugin) for their ability to model and reason with defeasible knowledge.
- **Objective 3: Develop a domain-independent methodology:** for constructing defeasible ontologies, based on engineering best practices.

Defeasible Ontologies

Currently, most defeasible reasoning implementations and evaluations have favoured the automated generation of defeasible ontologies [1] which in practice may not capture the intricate relationships, exceptions, and context-specific information of real-world domains.

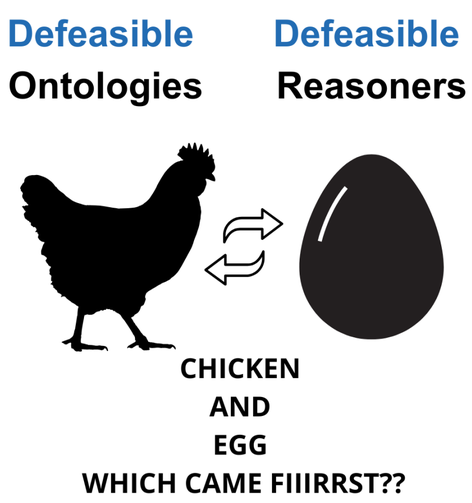

Figure 1. Defeasible Ontologies vs Defeasible Reasoning Implementations

Motivation

While automated generation of defeasible ontologies offers a convenient way to create test cases for evaluating defeasible reasoning algorithms, it also comes with certain disadvantages:

1. **Lack of Realism and Artificial Scenarios:** First, these synthetic testbeds (simplified examples) dont reflect how rules and exceptions actually interact in practice
2. **Efficiency:** Second, they give a false sense of confidence reasoners may appear efficient, but their performance hasnt been tested against realistic, domain-specific knowledge.
3. **Usefulness:** Third, do the reasoners provide appropriate real-world responses to queries posed to well-constructed, complex, realistic, defeasible ontologies.

Defeasible Ontology Domains

Design Approach

Multiple domains will be consider for the design of a defeasible ontology to establish design patterns.

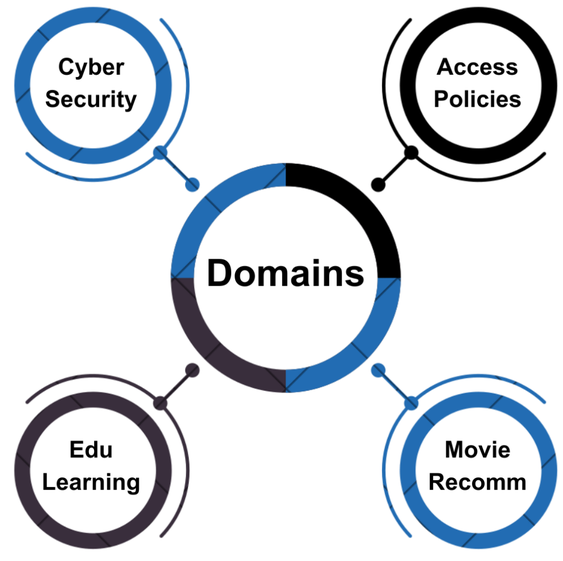

Figure 2. Domains to be implemented

Conclusions

1. **Bridges the gap** between defeasible reasoning theory and real-world application.
2. **Empowers developers** with tools and methodologies to handle exceptions in dynamic domains.

Key Goal:

- Break the “chicken and egg” cycle in ontology and reasoner development
- Enable robust evaluation of defeasible reasoning tools

Paves the way for smarter, adaptable AI systems that mirror human-like reasoning.

Expected Contributions

- **Theoretical Contribution**. A formal methodology for defeasible ontology design.
- **Practical Impact**.
  - **Domain-Specific Ontologies:** Ready-to-use resource for testing defeasible reasoners.
  - **Empirical Benchmarking of Reasoners:** Ontologies serve as empirical benchmarks to evaluate the correctness, usefulness, and performance of defeasible reasoners
  - **Tool Compatibility and Validation:** Our work also evaluates tool supportusing Protégé plugins like DIP - to see whether current technologies are ready for defeasible logic in practice

References

[1] Giovanni Casini, Thomas Meyer, Kody Moodley, Uli Sattler, and Ivan Varzinczak. Introducing defeasibility into owl ontologies. In *The Semantic Web-ISWC 2015: 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II 14*, pages 409–426. Springer, 2015.

[2] Rizwan Iqbal, Masrah Azrifah Azmi Murad, Aida Mustapha, Nurfadhlina Mohd Sharef, et al. An analysis of ontology engineering methodologies: A literature review. *Research journal of applied sciences, engineering and technology*, 6(16):2993–3000, 2013.

[3] Gabriele Sacco, Loris Bozzato, and Oliver Kutz. Know your exceptions: Towards an ontology of exceptions in knowledge representation. *arXiv preprint arXiv:2403.00685*, 2024.

Supervisor: Prof. Tommie Meyer