2025 Semantic Web - Postgraduate Students

Title: Extending the 3T Trigger Language with Native SPARQL Integration

1. Project Background

Triggers are a well-established mechanism in relational databases for enforcing business rules, automating workflows, and maintaining data integrity. However, RDF triple stores lack first-class trigger support. The 3T system addresses this gap by providing a triple-store-agnostic trigger engine with a custom interpreted language.

While the current 3T language supports imperative logic, it does not natively support SPARQL, limiting its expressiveness. This project extends the 3T trigger language to seamlessly integrate native SPARQL queries and updates within trigger bodies.

2. Project Objectives

- Extend the 3T trigger language with native SPARQL support.
- Implement safe execution of SPARQL queries and updates using Apache Jena.
- Prevent SPARQL injection and unsafe query construction.
- Evaluate the performance impact of SPARQL-enabled triggers.
- Demonstrate the extended system using realistic trigger scenarios.

3. Scope and Assumptions

The project builds on the existing 3T codebase and uses Apache Jena as the RDF access layer. Recursive triggers and SPARQL optimization are out of scope unless explicitly approved.

4. Deliverables

Deliverable 1: Language Specification Update

Design and document language extensions supporting SPARQL UPDATE, SELECT, and ASK within trigger bodies. Provide updated grammar definitions and example trigger scripts.

Deliverable 2: Implementation

Extend the 3T parser and interpreter to support SPARQL execution. Integrate Apache Jena for query and update processing, including variable binding and result mapping. (SEMANTIC WEB PROJECT)

Deliverable 3: Safety and Security

Implement parameterized SPARQL execution to prevent injection attacks. Ensure correct typing of literals and IRIs and controlled dataset access.

Deliverable 4: Performance Evaluation

Evaluate the overhead of SPARQL-enabled triggers compared to native operations and measure update throughput with increasing numbers of triggers. Present results using tables and plots.

Deliverable 5: Demonstration Scenario

Develop a demo showing SPARQL-based trigger conditions and SPARQL UPDATE actions with clear before-and-after knowledge graph states.

5. Evaluation Criteria

- Correctness and completeness of the language extension
- Robustness and clarity of the implementation
- Security of SPARQL execution
- Quality of performance evaluation
- Clarity of documentation and demo effectiveness

6. Expected Learning Outcomes

The student will gain experience in language design, RDF and SPARQL processing, database systems engineering, experimental evaluation, and research-quality software development.