Assignment 2: Knowledge Graph Schema Design

Knowledge Graphs with Large Language Models
MSc in AI and Data Science, AUEB (2025–2026)

Evangelos Rafail Kalemkeridis

Introduction

This report presents the design and testing of the knowledge graph schema for the **DeFi Protocol Explorer**, which was specified in Assignment 1. The goal of this assignment is to create a semantically consistent knowledge graph schema that supports a set of competency questions through executable SPARQL queries. The schema and data were implemented in RDF/OWL and tested using Protégé.

Task 1: Schema Design

The ontology was created in RDF/OWL (Turtle syntax). It models the essential entities and relationships of the decentralized finance (DeFi) ecosystem.

Main Classes

- **Protocol**: A DeFi application (e.g., Uniswap, Aave, Lido).
- **Blockchain**: A blockchain network or L2 where protocols are deployed (e.g., Ethereum, Arbitrum).
- **Category**: Describes the type of a protocol (DEX, Lending, Liquid Staking).
- **Token**: A cryptocurrency token supported or used by protocols.
- **Organization**: Entities such as audit firms, VCs, or teams.
- **Person**: Individuals related to protocol creation or governance.

Object Properties

- `:deployedOn` — links a protocol to the blockchain(s) it operates on.
- `:belongsToCategory` — associates a protocol with its DeFi category.

- `:supportsToken` — connects a protocol to supported tokens.
- `:auditedBy`, `:backedBy`, `:partnersWith` — represent relations with organizations.

Datatype Properties

- `:tvlUSD` (decimal) — protocol's Total Value Locked.
- `:launchDate` (date) — launch date of the protocol.
- `:sourceURL`, `:lastUpdated` — provenance metadata.

Dummy Data

Sample RDF data was created for testing and includes instances of four protocols (Uniswap, Aave, Lido, SushiSwap), three blockchains (Ethereum, Arbitrum, Polygon), and several tokens and organizations. The data can be opened directly in Protégé with the namespace `http://example.org/defi#`.

Task 2: Schema Testing with SPARQL Queries

The schema was validated by transforming the competency questions from Assignment 1 into SPARQL queries and executing them against the dummy data.

Example Queries

- **Q1:** What is the TVL of Aave?
  ```
  SELECT ?tvl WHERE { ?p rdfs:label "Aave"; :tvlUSD ?tvl. }
  ```
- **Q2:** On which blockchain is Uniswap deployed?
  ```
  SELECT ?chainName WHERE { ?p rdfs:label "Uniswap"; :deployedOn ?c; rdfs:label
  ?chainName. }
  ```
- **Q6:** Which protocols are deployed on both Ethereum and Arbitrum?
  ```
  SELECT ?pName WHERE { ?p :deployedOn :Ethereum, :Arbitrum; rdfs:label
  ?pName. }
  ```
- **Q7:** Which protocols share the same audit firm as SushiSwap?
  ```
  SELECT ?otherName WHERE { ?sushi rdfs:label "SushiSwap"; :auditedBy ?a.
  ?other :auditedBy ?a; rdfs:label ?otherName. FILTER(?other != ?sushi)}
  ```
- **Q10:** Which protocols belong to the Liquid Staking category?
  ```
  SELECT ?pName WHERE { ?p :belongsToCategory :LiquidStaking; rdfs:label
  ?pName. }
  ```

- **Q11:** Which blockchain hosts the most DeFi protocols?
  `SELECT ?chain (COUNT(?p) AS ?count) WHERE { ?p :deployedOn ?chain. } GROUP BY ?chain ORDER BY DESC(?count)`

All queries execute successfully in Protégé's SPARQL tab using the provided dummy data.

Design Decisions

- RDF/OWL was selected for semantic richness and compatibility with Protégé and SPARQL.
- A simple namespace (`http://example.org/defi#`) ensures readability and no conflicts.
- The schema separates **object properties** (relationships) from **datatype properties** (attributes) to maintain clarity.
- Each class and property has human-readable `rdfs:label` and `rdfs:comment` annotations to aid GraphRAG grounding.
- Provenance attributes (`:sourceURL`, `:lastUpdated`) support future explainability and reduce LLM hallucinations.

LLM Usage

Large Language Models (LLMs) were used to accelerate schema design and query drafting, but their output was critically reviewed and adapted. Example prompts used:

- "Design an RDF/OWL ontology for a DeFi knowledge graph with Protocol, Blockchain, Token, and Organization classes. Include example Turtle syntax and SPARQL queries."
- "Suggest SPARQL queries for retrieving protocol TVL, category, and blockchain deployment based on the following ontology."

Conclusion

The designed RDF/OWL schema effectively represents key concepts and relations in the DeFi ecosystem. All competency questions are answerable through SPARQL queries over the dummy data, confirming the schema's validity and expressiveness. The model can be easily expanded with real-world data from APIs such as DeFiLlama or CoinGecko, and it provides a strong foundation for GraphRAG-based QA systems.

Evangelos Rafail Kalemkeridis
MSc in AI and Data Science, AUEB
November 2025