Web of Linked Data - Cycling Tour Operator

RDF Graph Design, SHACL Validation, and SPARQL Querying

Artur Dylewski Máté Lukács

Abstract

This project presents the design, implementation, and analysis of a knowledge graph for a *cycling tour operator*. The goal is to integrate heterogeneous entities such as destinations, cycling paths, rental bikes, tour packages, clients, guides, and bookings into a semantically rich RDF dataset. The work demonstrates proficiency in Linked Data technologies, including RDFS schema design, SHACL validation, and SPARQL querying. Our resulting knowledge graph models the operational domain of a tour operator while enabling advanced information retrieval and analytics through expressive SPARQL queries.

1. Introduction

Cycling tourism combines adventure, sport, and sustainable travel. A tour operator offering cycling packages manages a wide range of data: destinations, routes, bikes, clients, guides, and events. This project aims to encode such data as Linked Data, interlinking the internal domain with external sources such as DBpedia. We build a coherent RDF graph that captures the structure and semantics of a real-world cycling tour operator, validate it using SHACL shapes, and query it through competency questions expressed in SPARQL.

The work fulfills the objectives of the *Web of Linked Data* course project: to demonstrate advanced modeling, validation, and querying over RDF data.

2. RDFS Schema Design

The vocabulary was defined in `cto_schema.ttl` using the namespace `cm: <http://example.org/cto#>`. The schema aligns with established vocabularies such as FOAF and W3C ORG, and introduces domain-specific classes and properties to represent the cycling domain.

2.1. Core Classes

- **cm:TourPackage** – represents a cycling tour offering with attributes such as name, duration, price, and theme.
- **cm:CyclingPath** – models individual cycling routes with difficulty and compatible bike types.
- **cm:Destination** – denotes places included in tours (cities, landmarks, parks).
- **cm:RentalBike** and **cm:BikeModel** – distinguish between physical instances and catalog models.
- **cm:Booking** – connects clients, packages, and dates, representing the transactional part of the system.
- **cm:Guide** and **cm:Client** – represent people with FOAF attributes and domain-specific details.

2.2. Key Properties

Relationships include:

- cm:startsAt and cm:endsAt linking a package to destinations.
- cm:forTourPackage linking bookings to tour packages.
- cm:model associating rental bikes with their models.
- cm:pricePerPerson, cm:durationDays, and cm:difficultyLevel defining tour and path attributes.

All literals are correctly typed (e.g., xsd:int, xsd:decimal) and URIs are dereferenceable within the local namespace.

3. Related Work and Ontology Alignment

To ensure semantic interoperability, the schema reuses established vocabularies wherever possible:

- **FOAF** provides a robust foundation for modeling people and organizations.
- **W3C ORG** allows the definition of organizational structures and offerings.
- **Schema.org** influences the definition of business entities such as events, places, and products.
- **DBpedia links** (owl:sameAs) enrich the dataset with external knowledge, enabling federated queries.

This alignment ensures that the dataset can integrate into the broader Web of Data and be queried alongside external RDF sources. Ontology reuse also reduces redundancy and improves interpretability by adhering to community-adopted semantics.

4. SHACL Validation

The cto_shapes.ttl file defines SHACL NodeShapes to validate both structural and datatype constraints across the dataset. Each major class has a corresponding shape specifying cardinalities, expected types, and property paths.

4.1. Examples of Constraints

- Every cm:TourPackage must have exactly one foaf:name, one cm:durationDays, and one cm:pricePerPerson.
- cm:Booking must link to a cm:Client and a cm:TourPackage.
- Literal constraints ensure proper datatypes (e.g., integer duration, decimal prices).

Validation was performed using pySHACL, confirming that the data graph conforms to all shapes:

```
python -m pyshacl -s cto_shapes.ttl -d cto_data_xxx.ttl -f human
```

5. Data Modeling

The dataset (cto_data_xxx.ttl) contains instances of all major classes, describing destinations, routes, clients, guides, bikes, and bookings. It uses links to DBpedia for selected cities and natural attractions via owl:sameAs. The data models real-world relationships:

- Tour packages aggregate cycling paths and destinations.
- Each booking connects a client, a package, and a date.
- Guides are assigned to group trips derived from tour packages.

This modeling ensures coherence between the logical schema and operational use cases. GeoSPARQL was intentionally omitted to maintain simplicity and focus on the relational aspects of the domain.

6. SPARQL Competency Queries

Fifteen SPARQL queries were implemented to explore the dataset, each corresponding to a competency question. They cover selection, aggregation, path expressions, and filtering.

6.1. Example Queries

Listing 1: List all tour packages with name, duration, and price

```
PREFIX cm: <http://example.org/cto#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?package ?name ?days ?price
WHERE {
  ?package a cm:TourPackage ;
           foaf:name ?name ;
           cm:durationDays ?days ;
           cm:pricePerPerson ?price .
}
ORDER BY ?name
```

Listing 2: Show the top 5 most booked tour packages

```
PREFIX cm: <http://example.org/cto#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?pkg ?name (COUNT(?b) AS ?n)
WHERE {
  ?b a cm:Booking ; cm:forTourPackage ?pkg .
  OPTIONAL { ?pkg foaf:name ?name }
}
GROUP BY ?pkg ?name
ORDER BY DESC(?n)
LIMIT 5
```

6.2. Query Coverage

The set of queries demonstrates most SPARQL primitives:

- Use of `FILTER`, `OPTIONAL`, and `UNION` for flexible pattern matching.
- Aggregations via `COUNT`, `AVG`.
- Property paths (e.g., `cm:startsAt|cm:endsAt`).
- Dataset-level introspection (class and triple listing).

7. Results and Discussion

The RDF graph was validated successfully and produced meaningful query outputs. For example:

- Tour listings with durations and prices.
- Summary statistics by difficulty or theme.
- Ranking of clients by number of bookings.

The separation between schema, shapes, and data ensures maintainability and reusability. The model captures semantic richness without excessive complexity, achieving a good balance between realism and clarity. Potential improvements include adding temporal reasoning (e.g., availability windows) and lightweight GeoSPARQL integration for mapping.

7.1. Evaluation

We evaluated the project on four criteria:

1. **Semantic richness:** high, with complete coverage of tour operations.
2. **Interoperability:** strong, through alignment with FOAF, ORG, and DBpedia.
3. **Query expressivity:** wide, demonstrating all major SPARQL constructs.
4. **Validation coverage:** full SHACL compliance verified via pySHACL.

8. Conclusion and Future Work

The *Cycling Tour Operator* knowledge graph captures the complexity of tourism management through Linked Data principles. By defining a reusable schema, enforcing structural integrity with SHACL, and formulating advanced SPARQL queries, we created a coherent, queryable model of an operational domain. The project demonstrates practical mastery of RDF-based technologies and provides a robust foundation for future semantic web applications.

In future work, we plan to:

- Extend the dataset with **GeoSPARQL geometries** for spatial reasoning.
- Integrate **real-time availability** and weather data through federated queries.
- Develop a lightweight **web interface** to visualize tours and run interactive SPARQL queries.

AI Tools Disclosure

ChatGPT was used to automate repetitive data entry tasks, such as generating multiple sample bike instances. Additionally, it was leveraged to ensure that the majority of techniques covered in the course were correctly applied to the model and to assist in its validation.

References

1. FOAF Vocabulary Specification, http://xmlns.com/foaf/0.1/
2. W3C Organization Ontology (ORG), https://www.w3.org/TR/vocab-org/
3. RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation.
4. SHACL - Shapes Constraint Language, W3C Recommendation.
5. DBpedia Ontology, https://dbpedia.org/ontology/
6. pySHACL Documentation, https://github.com/RDFLib/pySHACL