Final Project Report

Topic: Designing a SHACL Shape Graph for Validating Movie Entities
Student: Matěj Volf
Course: Graph Data and Knowledge

GitHub repository: https://github.com/volfmatejj/SHACL-Movies

Introduction

The goal of this project was to design and test a SHACL shape graph for validating RDF data in the movie domain. When working with open knowledge graphs such as DBpedia, data quality is often inconsistent. Some movie records miss important attributes, while others contain values in unexpected formats or lack semantic typing.

To address this issue, the project focuses on defining validation rules using Shapes Constraint Language (SHACL). These rules are applied to entities of type dbo:Film to detect missing information, incorrect data types, and incomplete semantic descriptions. The resulting shape graph is designed as a general validation layer that can be reused for any RDF dataset based on the DBpedia ontology.

The written report represents a descriptive part of the project. All technical artifacts created during the project, including SHACL shapes, SPARQL queries, and test datasets, are stored in a public GitHub repository. This repository serves as a complete archive of the implementation, allowing for the verification of results and reuse of the validation logic. Specifically, the repository contains the shape definitions (shapes_movies_en.ttl), the synthetic datasets used for testing (data_synthetic_valid.ttl, data_synthetic_errors.ttl), and the SPARQL query used to extract real-world data (SPARQL Query), along with the extracted data itself (movies_en.ttl).

Domain Analysis

The DBpedia ontology (dbo) was selected as the target vocabulary for this project. As a first step, I analyzed which properties are most associated with movie entities and divided them into core and advanced attributes based on their importance.

Core Attributes:

- rdfs:label – the title of the movie
- dbo:director – a link to the director of the movie
- dbo:runtime – the duration of the movie in seconds
- dbo:releaseDate – the release date of the movie

Advanced Attributes:

- dbo:starring – links to actors appearing in the movie
- dbo:budget – the production budget
- dbo:genre – the genre of the movie

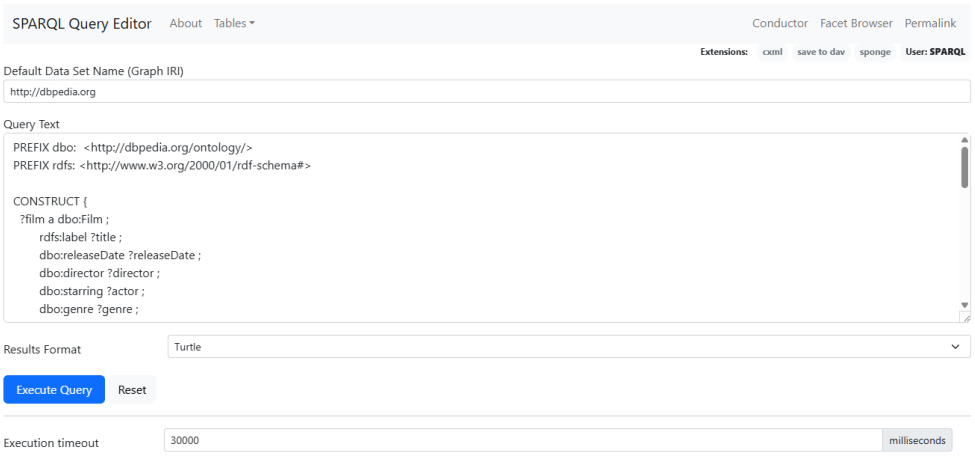
Movie data was extracted from DBpedia using a SPARQL CONSTRUCT query to create a local RDF dataset for testing and validation purposes.

Tools and Environment

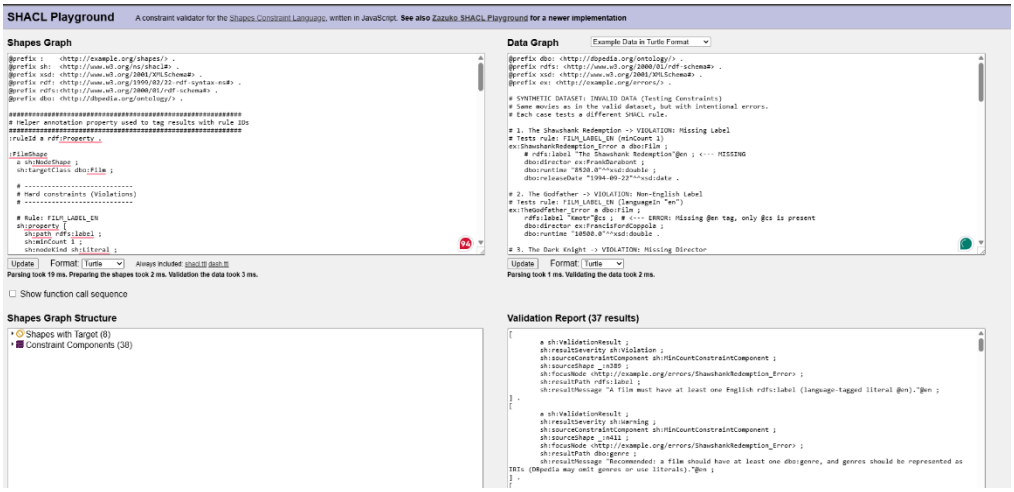
Several tools were used during the development and verification of the project. The DBpedia SPARQL endpoint available at https://dbpedia.org/sparql was used to explore the ontology and extract movie data. The SHACL shapes were designed and iteratively tested using the SHACL Playground (https://shacl.org/playground/), which allows interactive validation and inspection of validation reports.

All final experiments were performed locally using prepared RDF datasets and the validation results were analyzed to verify the correctness and usefulness of the defined constraints.

The DBpedia SPARQL endpoint



SHACL Playground



SHACL Implementation

The validation logic is implemented in a SHACL file (shapes_movies_en.ttl) containing a single NodeShape targeting the class dbo:Film. The constraints are divided into two levels of strictness using the sh:severity property.

Level 1: Core Constraints (Violations)

These constraints define the minimum metadata requirements for a movie entity. If any of these rules fail, the record is considered structurally invalid.

- **Title Check:** Every movie must have at least one English title (rdfs:label with language tag @en).
- **Director Check:** Every movie must have at least one director, and the value must be an IRI.
- **Runtime Check:** Every movie must have a runtime value. The value must be numeric and fall within a realistic range between 60 and 18,000 seconds (1 minute to 5 hours).
- **Release Date Check:** The release date must not be multi-valued. If present, it must be typed as xsd:date or xsd:dateTime.

Level 2: Advanced Constraints (Warnings)

These constraints focus on data completeness and semantic quality. Violations at this level do not invalidate the record but highlight potential data quality issues.

- **Type Consistency:** Directors and actors are recommended to be explicitly typed as dbo:Person. This rule is implemented as a warning because DBpedia data often omits type assertions.
- **Budget Validation:** A movie is recommended to have a budget value. If present, the budget should be represented as literal and should be single-valued.
- **Genre Check:** A movie is recommended to have at least one genre, represented as an IRI.

Verification and Results

To evaluate the correctness of the designed shapes, a two-phase validation process was performed.

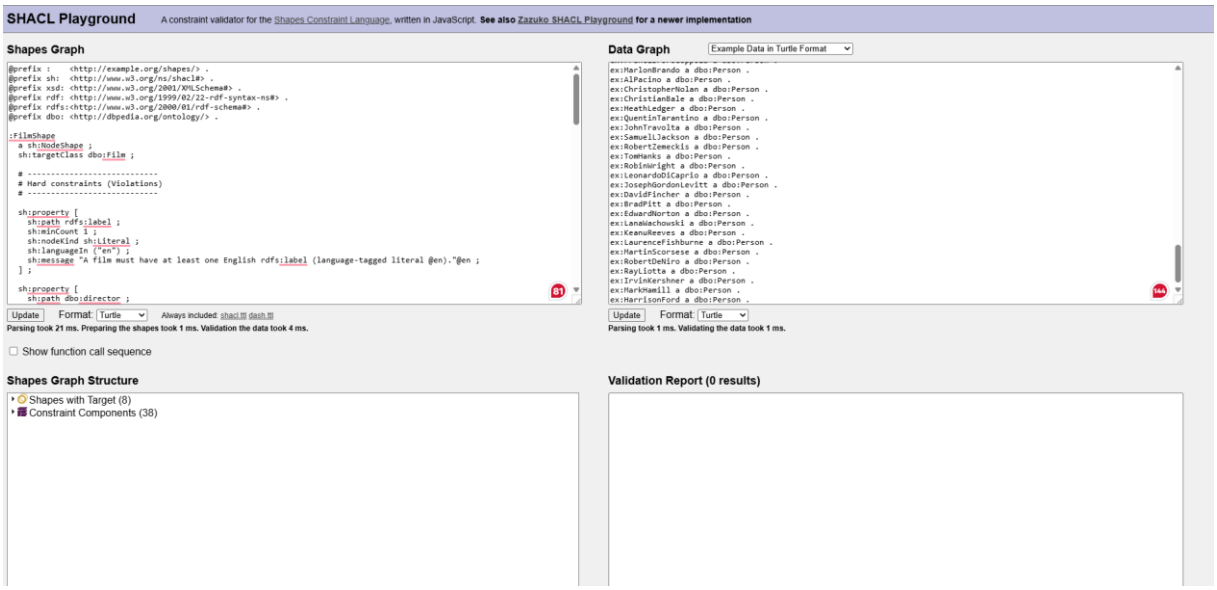
Phase A: Synthetic Data Testing

Two artificial datasets were created to test the validation rules in a controlled environment:

- **Valid Dataset (data_synthetic_valid.ttl):** Contains ten correctly structured movies (e.g., *The Shawshank Redemption*, *The Godfather*) with complete metadata.
- **Error Dataset (data_synthetic_errors.ttl)**: Contains twelve intentionally incorrect movie records designed to trigger specific rules (e.g., missing title, runtime represented as text).

Results:

- The valid dataset passed validation with 0 Violations and 0 Warnings, confirming that correct data is accepted.



- The error dataset triggered exactly the expected validation results. For example, Forrest Gump (missing runtime) produced a Violation, while Titanic (budget as text) produced a Warning. The detailed validation report can be found in the file Validation_Synthetic_Errors.txt

Both datasets are available in the GitHub repository.

Phase B: Real-World Data Testing

In the second phase, a real dataset (movies_en.ttl) was extracted from DBpedia and validated against the SHACL shape graph. The SPARQL query and the resulting dataset are included in the GitHub repository to ensure reproducibility.

Key findings:

- **Missing Core Data (Violations):** Two movies (*Fantaghirò 5* and *La taverna della libertà*) failed validation because they lacked the mandatory dbo:runtime property. This confirms that the shape graph correctly identifies incomplete records.

- **Missing Type Information (Warnings):** A frequent warning concerned missing dbo:Person typing for directors and actors. This occurred because the extracted dataset contained entity URIs without their corresponding type assertions. This behavior is typical for open linked data and demonstrates that the warnings provide meaningful feedback rather than false errors.

  These constraints were intentionally modeled as warnings rather than violations. The goal was not to enforce strict typing, but to illustrate potential semantic incompleteness in the data. If the dataset were enriched with additional type information, these warnings could be eliminated or the corresponding constraints removed without affecting the core validation logic.

- **Data Sparsity:** Many movies triggered warnings for missing dbo:budget and dbo:genre, showing that non-essential metadata is often incomplete in DBpedia.

Conclusion

This project demonstrates how SHACL can be used as an effective tool for automated data quality validation. The designed shape graph clearly distinguishes between critical structural errors and softer data quality issues.

The validation results show that while DBpedia provides a large amount of movie data, its completeness and consistency vary significantly. The SHACL validation report acts as a simple quality audit, highlighting records that require correction (e.g., missing runtimes) or enrichment (e.g., missing budgets or genres). The developed shapes are reusable and can be applied to validate other RDF datasets based on the DBpedia ontology.

The designed SHACL shapes are intentionally extensible and can be easily expanded with additional constraints, such as audience ratings, box office revenue, or production countries. This makes the shape graph suitable not only for this project, but also as a reusable validation template for other datasets based on the DBpedia ontology.