

ISO 15926-4 Interactive Visualization Project.

CITS5206 TEAM-18

August 2024

[1 Aim](#)

[2 Base Case](#)

[3 Proposed Solution](#)

[4 Functionality](#)

[5 System Design](#)

[Backend — SPARQL](#)

[Frontend — Streamlit](#)

[Key Features:](#)

[6 Requirements](#)

[**Functional Requirements**](#)

[Data Interaction](#)

[**Non-Functional Requirements**](#)

[Performance](#)

[Maintainability](#)

[Compatibility](#)

1 Aim

The aim of this project is to develop an interactive visualization tool for an equipment reference data library described in the ISO 15926-4 Standard. This tool will allow members of the ISO TC 184 SC4 committee, including Prof. Melinda Hodkiewicz, to efficiently interrogate and visualize hierarchical data relationships through a web-based interface. The tool is intended to be demoed at the ISO meeting in Stavanger, Norway, in late October.

2 Base Case

Currently, members of the ISO TC 184 SC4 committee have to manually search through extensive data records to find relevant information. This process is time-consuming and inefficient, as it involves navigating through large datasets without an intuitive interface to highlight hierarchical relationships.

3 Proposed Solution

The proposed solution is to create a web-based interface that integrates with the data available at the SPARQL endpoint (<https://data.15926.org/sparql>). Using this interface, users will be able to query the database and visualize the hierarchical relations of the equipment reference data. The visualization tool will be developed using Streamlit, but we remain open to alternative frameworks based on project needs and client preferences.

4 Functionality

Our website will consist of the following main pages:

Search and Visualization: Users can enter their queries in a search bar and visualize the hierarchical relations in the data. The page will provide various interactive elements to filter, search, and navigate through the data. Both parent branches and child branches of the hierarchy can be expanded and collapsed for detailed viewing.

5 System Design

There are a number of factors to consider when deciding a stack, shown in the following table:

Factor	Priority	Explanation
Project requirements	Highest	The stack must be capable of fulfilling all requirements.
Continued development	High	The stack must use familiar technologies that are well-known and industry-standard.

Skills and backgrounds	High	Due to the short time frame, picking technologies where all developers are comfortable is important.
Languages' ecosystems	High	The language must have libraries capable of fulfilling the requirements.
Guides and documentation	Medium	The existence of detailed guides and documentation is important, and directly relates to using familiar technology.
Library ease-of-use	Medium	The libraries chosen should have a good developer experience and interface well with one another.
Learning curve	Low	While desirable, this is unavoidable and hard to evaluate as members do not have prior experience which addresses all aspects of the project's requirements.

The project will be built as a web application, organized into a single framework using Streamlit. Streamlit will handle both the frontend interface and backend data interactions with the SPARQL endpoint.

- **Backend:** SPARQL
- **Frontend:** Streamlit
- **Deployment:** Amazon AWS

Backend — SPARQL

The project will use SPARQL to query the equipment reference data stored at the provided endpoint. The SPARQL queries will retrieve data dynamically based on user input, allowing for interactive and responsive data visualization.

Frontend — Streamlit

We propose using Streamlit for the frontend to create an interactive web-based interface. Streamlit allows for rapid development of data applications with Python, integrating seamlessly with SPARQL for backend data retrieval.

Key Features:

- **Search Bar:** Users can enter queries to search the hierarchical data.

- **Data Visualization:** Display hierarchical data with expandable parent and child branches.
- **User Interaction:** Users can filter, search, and navigate through the data interactively.

6 Requirements

Functional Requirements

Data Interaction

Identifier	Name	Description
FR1.1	Query Input	Users should be able to input a hierarchy into the search bar.
FR1.2	Data Retrieval	The system should return relevant information from the SPARQL endpoint based on user queries.
FR1.3	Visualization	The system should provide a visual representation of hierarchical data, with expandable and collapsible parent and child branches.

Non-Functional Requirements

Performance

Identifier	Name	Description
NFR1.1	Defined Time	The system should return query responses within a defined time frame (e.g., within 3 seconds).

Maintainability

Identifier	Name	Description
NFR2.1	Clear Code	The codebase should be well-documented, formatted, and linted to facilitate future updates and maintenance.

Compatibility

Identifier	Name	Description
NFR3.1	SPARQL Integration	The application should interface seamlessly with the SPARQL endpoint for data retrieval.
NFR3.2	Responsive Design	The application design should support both desktop and mobile views.