

# Assistive Query Building for Semantic Data <sup>\*</sup>

Vangelis Kritsotakis<sup>1</sup>, Yannis Roussakis<sup>1</sup>, Theodore Patkos<sup>1</sup>, and Maria Theodoridou<sup>1</sup>

Institute of Computer Science, FORTH, Heraklion, Greece  
{vkrits,rousakis,patkos,maria}@ics.forth.gr

**Abstract.** We present A-QuB, a platform that facilitates the exploration, discovery and management of semantic metadata. A-QuB incorporates a multitude of features on top of an intuitive and user friendly environment, in order for both novice and expert users to create and execute complex queries. The platform is agnostic of the underlying conceptual model and configurable on different aspects.

## 1 Metadata Querying for Interdisciplinary Research

The publishing of structured and semantically enriched data is changing traditional models of conducting business and research. Modern science in particular is becoming more collaborative and multidisciplinary, taking advantage of the plethora of data being produced by groups with diverse scientific backgrounds. So called Virtual Research Environments (VREs) aim to promote this scheme, overcoming physical or semantic barriers, and facilitating researchers from diverse fields to exchange data and resources, decoupling science from ICT.

For such an interoperability to be achieved, various challenges are faced from the technical, semantic and legal standpoint. One major goal is the generation of high level ontologies with rich metadata that are easily explored by researchers. In this paper, we present the Assistive Query Building (A-QuB, pronounced “/ei/ cube”) platform<sup>1</sup>, a fully functional platform that facilitates the exploration, discovery and management of semantic metadata for both novice and expert users who wish to execute complex queries. A-QuB provides an intuitive, user friendly environment that is highly configurable, making it appropriate for various domains, still being agnostic of the underlying conceptual model.

Similar existing tools can be classified broadly into three general groups. Query building tools for the Semantic Web, such as FedViz [2] and SPARQLing [1], that are generic systems and suitable for both experts and non-experts in SPARQL. Such tools though are not always appropriate for the purposes of a VRE, as they cannot exploit the features of the underlying conceptual model. For instance, they lack provision of active guidance to users, which is important when exploring, rather than just querying, metadata.

---

<sup>\*</sup> This work was carried out within the H2020 VRE4EIC project (No 676247).

<sup>1</sup> A-QuB website (username/password: demo): <https://tinyurl.com/y6ujmsdt>

At the other extreme, one can find custom UIs developed in the context of VREs, such as EPOS, iMarine, Bluebridge, Lifewatch Greece, and others. Despite their capacity to exploit the underlying models, such systems are often too tightly coupled to the domain or schema of interest, hindering portability and restricting the opportunities for collaboration among disciplines beyond the ones originally designed for. It is also not uncommon for such systems to sacrifice expressive features of SPARQL for simplicity in design. Finally, researchers can also rely on traditional SPARQL endpoints provided by widely used triplestores, such as Blazegraph, GraphDB etc, which are usually suited for more expert users.

A-QuB does not only aim at offering (another) simple query interface for providing access to the underlying metadata, hiding the complexity of writing expressive SPARQL queries. Although the look'n'feel resembles similar query building systems with the goal of reducing the learning curve for the novice user, the features incorporated are based on a thorough analysis of the requirements of existing VREs and Research Infrastructures (RIs) (see [3]), offering a repertoire of solutions for various beneficiaries. Overall, the goal is to support both

- **Query writing for the expert**, exploiting the capacity of the underlying conceptual models and gathering best practices from similar systems,
- **Discovery of metadata (exploratory search) for the novice user**, to help researchers search across domains and data that they are not familiar with, guiding them in the process of creating a query, thus promoting interoperability and cooperation among research communities.

## 2 Main Features of A-QuB

The current A-QuB implementation offers an experimental metadata catalogue containing metadata from affiliated RIs. Imagine a user searching for the projects involving a researcher named Jeremy, whose last name cannot be recalled. The only information is that Jeremy was member of a health organization located around central Europe between 2010 and 2015, and that the metadata for these projects can be found in the EPOS and EKT VREs. Apparently, apart from its simplicity for demonstration purposes, this query is rather vague, calling for some assistance by the system, in order not to overwhelm the user with results.

One key characteristic of A-QuB is that it dynamically executes sub-queries and presents partial results on-the-fly, while the user builds her query. The goal is twofold. On the one hand, it helps exploration by presenting results and allowing the user to limit the scope of the query. For instance, once the user finds the Jeremy she is looking for while adding filters, she can select the appropriate instance, without having to re-execute the query from scratch. On the other hand, it helps avoid meaningless queries that return no results. This is achieved by transparently altering the options available during the query building process, eliminating choices that can lead to an empty search space.

A-CuB offers a combination of different ways for searching relevant metadata within the same query. It enables the user to compose queries using *keyword search* (a person named Jeremy), *entity-based search* (an entity “Person” with

a relation “is member of” on entity “Organization”), *time range-queries*, if supported by the metadata (the “is member of” relation is restricted between 2010 and 2015), *filter-based search* (the filter “a project has contact person Jeremy” and the filter “a person is member of an organization unit” are nested conjunctive filters), or *geo-spatial queries* (the user can interact with a map, in order to select a region in central Europe or give coordinates).

Filter-based search occurs based on an intuitive interaction model (Figure 1(a)) and may involve conjunctive and disjunctive nested queries. As excessive use of nested filters may lead to difficult to comprehend queries, there is an option to limit the depth and/or width of filters.

Geo-spatial queries are constructed with the help of an interactive map (Figure 1(b)), offering most of the features expected, such as searching by toponyms or geographical regions, selecting specific instances, etc. Note that in most existing systems, this is a standalone feature, whereas in A-CuB it is incorporated as a simple partial step.

The platform also gives the user the ability to store queries for later use. This is a handy option, enabling the retrieval of complex queries, to be used as templates for constructing competency queries with multiple filters or for making minor adaptations to complex exploration tasks.

Every step performed while constructing the query (i.e. every filter applied, instance selected, search-term used or region set, etc.), is analyzed by the system to construct SPARQL building blocks. These blocks are merged appropriately together, creating the final QUERY that is executed.

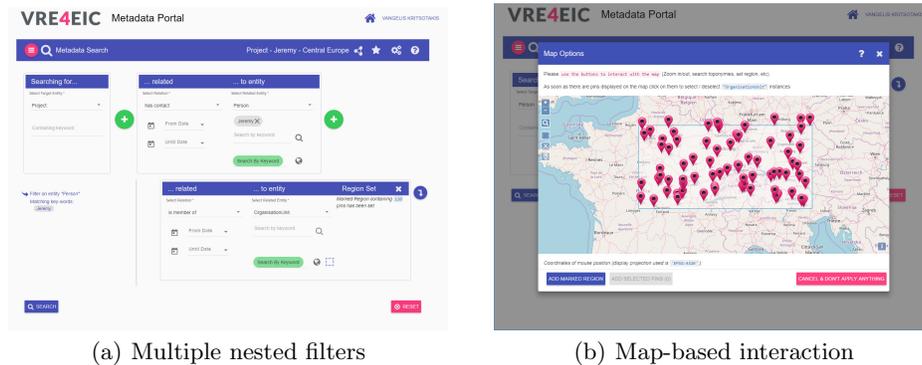
After executing a query, the results can be examined through an internal resolver, allowing simple navigation from any instance of the targeted entity to any of the instances of the related entities. In that way, end users can even continue their discovery after the results are retrieved. For instance, in the running example, the user can explore more information about Jeremy from the returned results, in order to make sure this is the person originally intended.

Finally, the platform provides a number of peripheral functionalities that complement its operation. A multi-factor authentication mechanism is used for granting access to the end users, while Role-Based Access Control ensures that users actions are properly regulated. Importing metadata is also supported, coupling the process with the acquisition of relevant provenance information for facilitating trust and versioning. Also, the platform is configurable on many different levels, in order to enhance flexibility, robustness and simplicity.

## 2.1 Implementation

A-QuB is a modern web platform, implemented in Spring Boot at the back-end, and the MVC AngularJS framework at the front-end. It is executed from the command line as a standalone Maven application, which embeds a Jetty server container (by default) and does not require any prior installed software, which makes it completely portable.

The design of the system is based on the Service-Oriented Architecture. The back-end, which is responsible for the management of Semantic (RDF) Data has



(a) Multiple nested filters

(b) Map-based interaction

**Fig. 1.** Different means of constructing complex queries

been implemented using a RESTful API on top of the open source version of Virtuoso Universal Server. Any other triple store providing the basic operations (e.g., query, insert etc.) can be used as well; an initial version operated on the Blazegraph triple store. Finally, a lightweight and portable H2 Database Engine is used for storing configuration options and saved queries.

### 3 Current and Future Work

As part of the H2020 VRE4EIC project, A-QuB currently operates using the CERIF<sup>2</sup> standard conceptual model, enabling interoperability among diverse research infrastructures. A series of task-based evaluation sessions have been conducted in the course of the project in several events with satisfactory results, as reported in the project's deliverables<sup>3</sup>

Our future implementation tasks involve the addition of features, such as the ability to export results in various formats and the linking of data with external sources. A more ambitious objective is to seamlessly feed resources retrieved by querying into external tools, e.g., workflow engines, data visualizers etc.

### References

1. Dastgheib, S., McSkimming, D., Natarajan, K., Kochut, K.: Sparqling: A graphical interface for sparql. In: International Semantic Web Conference'15 (2015)
2. e Zainab, S.S., Hasnain, A., Saleem, M., Mehmood, Q., Zehra, D., Decker, S.: Fedviz: A visual interface for sparql queries formulation and execution. In: ISWC (2015)
3. Zuiderwijk, A.: Analysing open data in virtual research environments: New collaboration opportunities to improve policy making. *Int. J. Electron. Gov. Res.* **13**(4), 76–92 (2017)

<sup>2</sup> <https://www.eurocris.org/cerif/main-features-cerif>

<sup>3</sup> <https://www.vre4eic.eu/publications/public-deliverables>