Visual Ontology-Based Query Approach for Data Access in Heterogeneous Expertise Environment: Application in PLM Biomedical Imaging

Cong Cuong Pham[1] , Alexandre Durupt[2], Nada Matta[3], Benoit Eynard[4]

[1]Université de Technologie de Compiègne, phamcong@utc.fr
[2]Université de Technologie de Compiègne, alexandre.durupt@utc.fr
[3]Université de Technologie de Troyes, nada.matta@utt.fr
[4]Université de Technologie de Compiègne, benoit.eynard@utc.fr

Corresponding author: Alexandre Durupt, alexandre.durupt@utc.fr

Abstract. The volume and complexity of Biomedical Imaging (BMI) data can be handled by well-known Product Lifecycle Management (PLM) solutions if a research study in this field is modeled as a cyclic process of four phases: study specifications; raw data acquisition; data processing and results publication. However, current PLM systems do not provide easy, flexible and user-adapted data access, especially in the context of heterogeneity expertise environments such as BMI. This paper presents VAQUERO (VisuAlization and QUERy based Ontology), a visual ontology-based data query approach, that aims at providing different kinds of users in the BMI field (common/ external, domain expert and technical users) with easy self-access to their data stored in a PLM Teamcenter system.

Keywords: PLM, Biomedical Imaging, Ontology-based data query, Visual query.
DOI: https://doi.org/10.14733/cadaps.2020.226-248

INTRODUCTION

The volume and complexity of data in the BMI field have increased considerably in recent years. Current research projects can extend to thousands of subjects and involve terabytes of data that are not limited to imaging [1,2]. Initially proposed in industrial enterprises to cope with the increased complexity, heterogeneity and large volume of product data in collaborative engineering, PLM solutions can be used to handle BMI data.

Data query approaches proposed by current PLM systems use query languages (Structured Query Language (SQL) or XML Query (XQuery)). Although these approaches do not require an understanding about where the data is physically stored, users still need to know the logical structure of the data in order to correctly formulate formal queries [1]. As a multidisciplinary field, BMI data is often used by different types of actors with a diversity of backgrounds (including their expertise and technical competence). Access to data by means of query languages has become inappropriate for the BMI community, mainly due to the following two reasons:

- **Lack of technical competence**: The logical data models used in a database may not be related to domain knowledge and thus are difficult to be interpreted by common users or domain experts. In addition, formal query formulation requires an understanding of query language syntaxes and it is very rare that these users in the BMI field will have sufficient skills.

- **Different user perspectives**: Each actor introduces his/her own point of view when adding new data. The same data can be named differently, while data provenances are often not sufficiently recorded. As consequence, finding the right data becomes very difficult due to the data's heterogeneous semantics.

The research posture adopted here is that actors with different backgrounds can access data themselves, without the help of database technicians. To allow self-access, the common sense approach would be to propose user-oriented representations of a database's structure. A visual query approach is therefore a promising route to improve data access. Accommodate visual ontologies should allow for both self-access and the heterogeneity of the expert-actors' competences.

Visual query systems and ontology-based query systems have been proposed separately to enable the data access of end-users. By pointing out the usefulness of having both a visual and ontology-based self-representation of data structure, this paper presents an adapted data query approach called VAQUERO (VisuAlization and QUERy based Ontology). The proposed approach is presented in the context of a real case study at the GIN (Groupe d'Imagerie Neurofonctionnelle of the UMR 5293 CEA CNRS Université de Bordeaux) laboratory, where BMI data handled with PLM solutions are used daily by expert-actors from diverse disciplines and with different competences.

Although the proposed method was built and tested with data in Bio-engineering and in the PLM field, it is general enough and could be used for other purposes including knowledge-based information system and data exchange/ product data management.

The rest of the paper is organized as follows. Section 2 deals with the state of the art on data querying/ontology approaches. Section 3 describes the adapted VAQUERO approach. Section 4 presents the application of this approach in detail using a case study at the GIN Lab. The paper ends in section 5 with some conclusions and proposes future work directions.

STATE OF THE ART: FROM DATA QUERYING TO ONTOLOGY VISUALIZATION

This section presents different approaches aiming to improve the data querying of end-users in Database Management System (DBMS). By analyzing these works, we deduce that ontology and its visual representation is a very promising solution for BMI data access issues in PLM systems, as presented above.

1.1 Traditional data querying approaches

Data access (or data querying) is the process of extracting information from a database to answer a specific question, and is an important function of any information system. Since the launch of the first DBMS, many methodologies have been proposed to facilitate data access:

- **Using keywords** [3]: Keyword-based search looks for results by syntactically comparing stored data with one or more words specified by users. Most PLM systems provide for this kind of search. However, queries can bring too many results, and applying filters can be sub-optimal.
- **Using natural language** [4,5]: Natural language query systems focus on meaning and the natural way humans ask questions; it is not limited to syntactic matching. The common approach is to use a metadata data model like Resource Description Framework (RDF) to convert a database into RDF resources. Users' information needs expressed in natural language are then translated into SPARQL (SPARQL (semantic) Protocol and RDF

Query Language) queries on these RDF resources. The use of natural language in traditional DBMSs is limited due to the cost of transforming data into RDF resources. An alternative approach is to use more recently developed artificial intelligent algorithms (reinforcement learning for example) to translate users' information needs expressed in natural language into formal SQL queries. This direction is obtaining very promising results, but the approach is not yet mature enough to be widely applied.

- **Using direct manipulation languages** [6]: Faceted searching (or faceted navigation) allows users to browse information with a predefined set of categories. Starting with a simple query, users refine their search options with navigation tools. Faceted queries are widely used in eCommerce websites like Amazon or eBay [7,8]. This approach is good for self-access but it appears that only one table of the data model can be queried at a time. Hence, queries with joins are not yet possible. Another direction is offered by Visual Query Systems (VQS). VQSs use visual representations to represent the area of interest and express related queries. These representations collectively inform users about the schema system and data affordances, through a number of sensory variables such as texture, color, size and shape, and allow users to communicate their information needs by interacting with them [9–11] . VQSs can help expert users reduce data structure complexity and thus enhance their data query capacities. However, the diversity of users' profiles is still not being considered. A VQS should provide each user with an appropriate visual representation corresponding to his/her profile. Nevertheless, this approach still requires some computer skills, which means the query formalism is not dynamic and adaptable to the expert-actors' point of view. By providing an explicit and consensus representation of domain knowledge, ontologies could be used to overcome these current issues

1.2 Ontology-Based data access

Defined as "a formal explicit specification of a shared conceptualization" [12–14] , ontology is used mainly in information systems for different purposes such as database integration and natural language processing. Ontology-based data access (OBDA) uses ontologies to provide end-users with a semantic access to databases by means of a three-level architecture [15,16], as illustrated in the Figure 1:

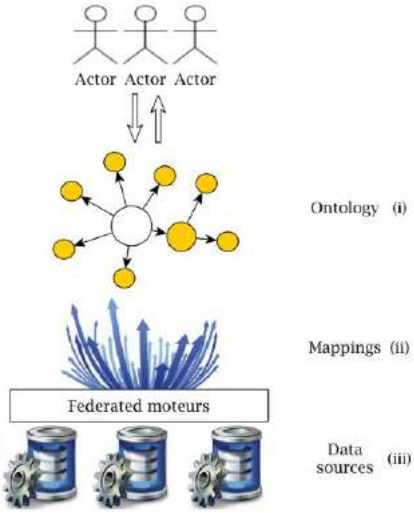


Figure 1: Ontology-based data access [14].

The ontology layer (i) acts as a mediator between the user and the data sources. It uses domain concepts and the relations between them to semantically represent underlying data sources (iii). Since this semantic representation is more familiar to end users than the logical model used to store data, users can use it to express their information needs themselves. Users' semantic queries are translated into formal queries on the data sources using predefined mappings (ii).
OBDA has gained a sizeable amount of theoretical and practical attention from the database community and has become a promising solution for data access problems in heterogeneous expertise environments. However, to date there has been no research on a general implementation methodology that could be reused. Furthermore, the manual construction of a complex query over various ontologies is not easy, and only a few works [17,18] are available to help users to visually construct their queries (see figure 2).

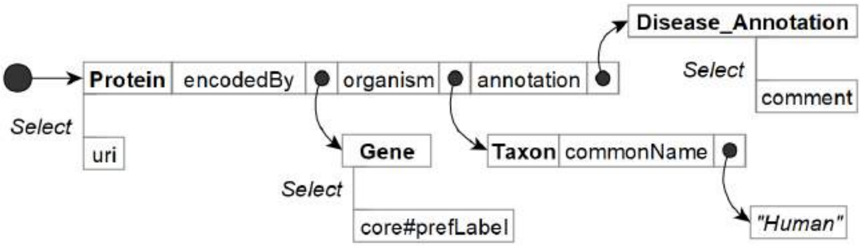


Figure 2: Ontology-based semantic query and its graphical representation, inspired by [19].

vSPARQL is a visual query language containing graphical notations to visually represent SPARQL queries (Smart et al. [15][20]). The query graph essentially contains nodes and arcs. vSPARQL uses color to differentiate between three different types of nodes corresponding to the nature of the variable they represent. Some visualization rules are used to represent various aspects of a SPARQL query, such as "ordering" or "filtering". vSPARQL supports both "SELECT" and "CONSTRUCT" queries. However, some SPARQL features are not supported, such as "ASK", "DESCRIBE", "DISCTINT", "LIMIT", etc.

RDF-GL, a graph-base query language derived from SPARQL, was proposed by Hogenboom et al.[16] to query Resource Description Framework (RDF) data. RDF-GL uses graphical notions like boxes, circles and arrows in different colors to graphically represent semantic queries. RDF-GL is limited in its expressivity capacity compared to SPARQL. It only supports a subset of SPARQL (the "SELECT" requests from among "SELECT", "AST", "CONSTRUCT" and "DESCRIBE"). Some of the SPARQL options of "SELECT" such as "FROM", "REDUCED", etc. are also not supported.
Both of the works presented above, as well as those of Zainab et al [17]and Borsje et al. [18], have the same approach, that is, to use graphical elements to help users visually construct their semantic queries [21,22]. However, none of them profit from the advantages of ontology visualization during the query formulation process. The next sub-section presents the popular ontology visualization techniques that inspired us to contribute our proposition presented in section 3.

1.3 Ontology visualization

Ontology visualization provides necessary supports that facilitate users in creating new ontologies or in exploiting existing ontological resources. An ontology is a hierarchy of concepts enriched with interrelations, axioms, and functions between concepts. Each concept is attached to a list of attributes and a set of instances. Thus, visualization tools must show all of this information in real time and in a manner easily accessible by human operators. Many ontology visualization techniques have been proposed, including Indented List, Link Node and Tree, Zoomable, Space Fill, Focus + Context or 3D Distortion and Landscapes [23].

- **Indented list**: A simple technique that illustrates the hierarchy of concepts using the indent "one-is" inheritance mechanism. This method of structuring concepts in an indented tree makes it easy to interpret concepts and their functions. Users can start with a very compact representation of an ontology's concepts and exploit them gradually by clicking on each node of the indented tree.
- **Node-link graph**: An ontology can be modeled as a set of triples (subject-predicate-object), so it can be represented in a graph where the "predicate" is the label of the arc connecting two nodes "subject" and "object". Several tools or plugins implemented in ontology editors have been developed to provide a graphical visualization of ontologies such as: OntoViz [24], IsaViz [25] SpaceTree [26], OntoTrack[27], etc. These tools use a variety of information visualization notions such as colors, sizes and forms to represent different components of an ontology: concepts, relations, instances, etc. While most of the developed tools are desktop applications, some web-based platforms have been developed recently, WebVOWL is one example [28].
- **Expandable**: This technique represents ontological concepts in different sizes according to their positions in the hierarchical tree of concepts. The concepts closer to the root concept are represented by larger circles. The user can zoom in on a node to enlarge it and to change its display level. Two tools that use this visualization technique are Jambalaya and CropCircles [20][23].
- **Space Filling**: This type of visualization uses the entire space of a screen to visualize ontological concepts. Each concept is represented by a space (a division); its size corresponds to the number of nodes it contains. This technique provides a direct view of the concept structure, but it is not very appropriate for large ontologies.
- **Focus + Context or 3D Distortion**: This type of visualization is based on the notion of distortion of the graphic view in order to combine context and focus. The node in focus is represented in the middle of the visualization and is surrounded by the other nodes which are smaller. In general, this visualization uses a hyperbolic equation and a user concentrates on a specific node to enlarge it. Some examples of this type of visualization are OntoRama [22], MoireGraphs [23], TGVizTab [29], OZONE [30], etc. These visualization techniques provide an overall view of ontological concepts, but they do not give an obvious visualization of the structure of concepts, which becomes difficult to interpret when the ontologies are large.

1.4 Discussion

As described above, several works have been proposed to improve the data access of end-users, including traditional approaches (natural language or visual query systems) and the more recently introduced ontology-based data access approach. Although the latter seems to be one of the most promising solutions for overcoming the issues of data access in the heterogeneous expertise environment [15], there are still a few issues that need to be resolved. First, it lacks a clear and explicit roadmap (step by step) that helps to build an ontology-based data access system from scratch, including: how to construct appropriate application ontologies to use in the intermediate layers, and how to represent them in a clear, intuitive and understandable way. Second, the manual construction of ontological queries is time-consuming and error-prone since it requires the understanding of the SPARQL language's syntax. To resolve the first issue, we describe in section 3.2 a general approach to build application ontologies that is simple to implement and reusable. Concerning the second solution, we combine different visual representation techniques (indented-tree and graph of relations) to make ontologies more understandable. We also introduce a set of visualization rules and graphical notations that help end-users build their ontological queries graphically instead of syntactically. These rules are defined mainly from an analysis of the SPARQL language specifications.

VAQUERO – VISUAL ONTOLOGY-BASED DATA QUERYING APPROACH

1.5 Global methodology, an intermediate layer supporting a heterogeneous expertises environment.

We propose to use a visual representation as an intermediate layer between data sources and users to improve the exploitation (searching and interpreting) of BMI data in PLM Teamcenter [31]. The intermediate layer contains three different layouts corresponding to the three main kinds of users at the GIN laboratory:

- **"Data model"** visualization: At the lowest level of abstraction, this layout is dedicated to advanced/technical researchers who have a fairly solid understanding of data structuring and who are familiar with the business terms of the data model.
- **"Classification**" visualization: Classification is an extension of the data model in the Teamcenter system that helps to classify data into different classes. Each class contains a set of data with common attributes. The use of the classification in the query interface allows a faceted search. Regular users can easily find their most interesting data from one or more classes. Vertical/horizontal relationships can be defined between classes, allowing users to formulate more complex ad-hoc queries on different classes of data. This layout is dedicated to trained actors who frequently manipulate the database.
- **"Ontological"** visualization: This layout is dedicated to actors who are domain experts or common users who know the expertise process and how to use data, but do not have enough technical competence to query the data themselves. Ontologies are used to provide them a semantic view of the underlying data structure. Different ontologies are used for different groups of expert-actors from various disciplines.

In our approach, actors graphically express their information needs on an intermediate layer with an appropriate layout. Users' queries are represented in a graph during the query formulation process. When queries are completed, they are transformed into formal queries thanks to a set of mappings between the data sources and the visual elements.

We propose an integrated "ontological" visualization method (see [32,33]) that contains four main functions:

- **Representation of domain knowledge**: Application ontologies are used to capture and represent domain knowledge in the intermediate layer. The intermediate layer is called the conceptual layer. These ontologies are represented visually.
- **Graphical query construction**: Each group of expert actors (in the same discipline) is provided with an appropriate application ontology and its visualization to formulate their own queries. Users' queries over application ontologies are represented visually in the form of a graph (Q_graph) during the query formulation process with the help of our proposed graphical conventions. Graphical queries (Q_graph) are transformed into SPARQL queries (Q_sparql) using "transformation rules".
- **Query transformation**: SPARQL queries (Q_sparql) over application ontologies are transformed into formal queries (Q_formal) over data thanks to predefined "mappings" (M) between the ontologies and the data sources.
- **Transformation of results**: In addition to a tabular format, the results (R) of formal queries (Q_formal) are also transformed into visual instances (I) to have an optimal user interpretation. In the ideal case, these results should be linked to the application ontologies and displayed on the same conceptual layer.

The mapping approaches are not covered in this article. They were done by hand in consultation with IT experts. For more details see [34]. The next subsection presents our visual ontology proposal for self-access in heterogeneous expertise environments.

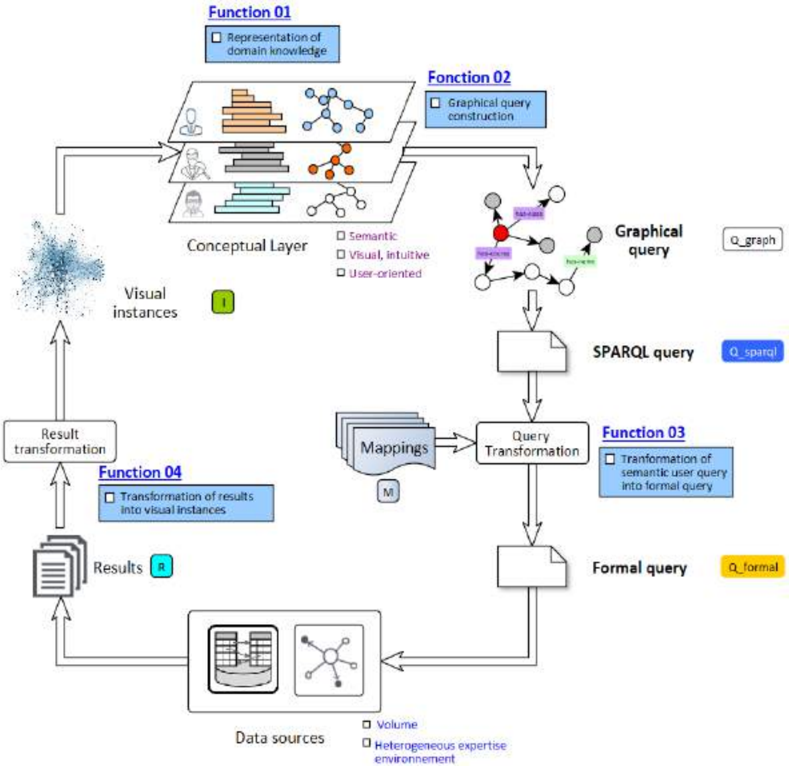


Figure 3: Use of ontological visualization in the conceptual layer enabling semantic data access by expert actors: implementation of different application ontologies and their visualizations adjusted to expert actors' competences.

1.6 Visual ontology proposal for self-access in heterogeneous expertise environments

1.6.1 Visual ontology construction

The construction of an application ontology is often uncertain, tedious and dependent upon the granularity and the size of the ontology. The more detailed and larger the ontology, the more difficult its construction. The construction process could be divided into successive steps where the results of a previous step are validated and adjusted before being used as input for subsequent steps. In addition, the construction process is not straightforward and changes can occur at any stage.

The construction process can be considered as a recursive four-step process (Figure 4):

- **Knowledge Acquisition**: This step aims to identify relevant target concepts and properties (ObjectProperty and DatatypeProperty) that could be used in application ontologies.

- **Knowledge Modeling**: Selects the most appropriate concepts and relationships identified in the knowledge acquisition phase and explicitly describes their uses. The hierarchical relations among concepts are then determined to establish a tree of concepts. New, more specific concepts could then be added to complete the concept tree.
- **Ontology Edition**: The manual expression of an ontology in a formal ontology language is tedious and prone to error. Ontology editors such as Protégé, NeOn, Vitro, etc. should be used. These tools allow to edit and build ontologies in a visual and intuitive interface. The edited ontologies can be exported in different formats such as OWL or RDF/XML.
- **Ontology validation**: The proposed application ontologies need to be tested and validated by daily queries of expert actors. Different test sessions should be tacked in place with different types of expert actors to ensure that proposed ontologies and their visualizations are able to improve the self-data access of each expert actor.

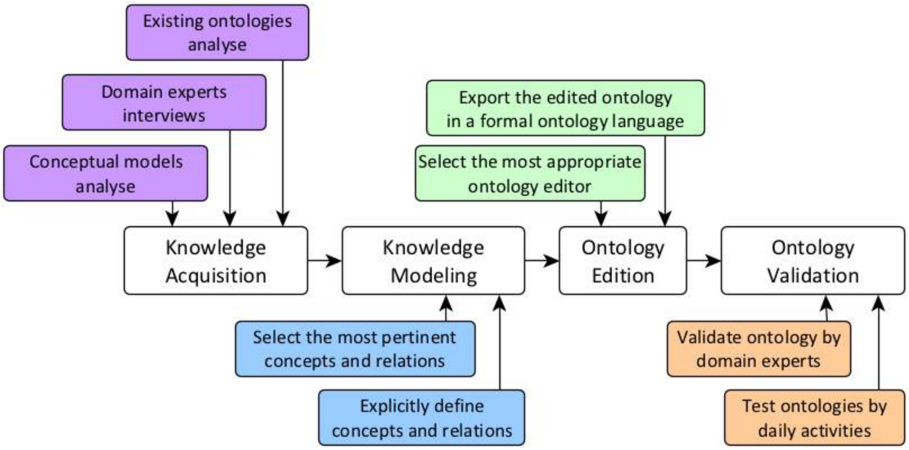

Existing ontologies analyse
Domain experts interviews
Conceptual models analyse
Export the edited ontology in a formal ontology language
Select the most appropriate ontology editor
Knowledge Acquisition
Knowledge Modeling
Ontology Edition
Ontology Validation
Select the most pertinent concepts and relations
Explicitly define concepts and relations
Validate ontology by domain experts
Test ontologies by daily activities

Figure 4: Ontology construction process in four steps.

In the rest of this section, we focus on the "knowledge acquisition" phase, which is most important step and can be accomplished by analyzing logical data models, reusing existing domain ontologies and interviewing domain experts. There is not necessarily an order or priority among these steps. All four should be considered and carried out in order to have a complete and reusable application ontology. The order of accomplishing them can be adapted according to the experience of the user and his/her level of expertise with the database.

Analyzing the data model (IT expert view)

Existing logical data models are valuable for application ontology construction. Predefined entities in the Entity-Relationship (ER) model could be used as a basis for the definition of ontological concepts (column "displayed name" in Table 2). Each entity becomes a class (a concept), while each attribute attached to an entity becomes its property ("DatatypeProperty"). All relations between the entities (strong or weak relation) and their corresponding cardinalities are examined. Each relationship between entities becomes a property object ("ObjectProperty"). Much of the work focusing on the transformation of ER models to OWL ontologies could be reused for this purpose. Documents, reports and articles produced from the day-to-day activities of experts are also an important resource to study. These documents often contain vocabularies and expressions used in the domain, which could be reused to name ontology concepts. Figure 5 illustrates the BMI-LM (BioMedical Imaging Lifecycle Management) data model proposed by Allanic et al. [31] to handle BMI data at the GIN. Business objects can be divided into six different categories corresponding to the six super concepts of application ontologies: "data" (acquisition result, processing unit result, etc.), "tool" (acquisition device, software tool), "source" (subject, study), "protocol" (acquisition definition, processing definition, etc.), "investigator" (agent) and "process" (acquisition, processing).

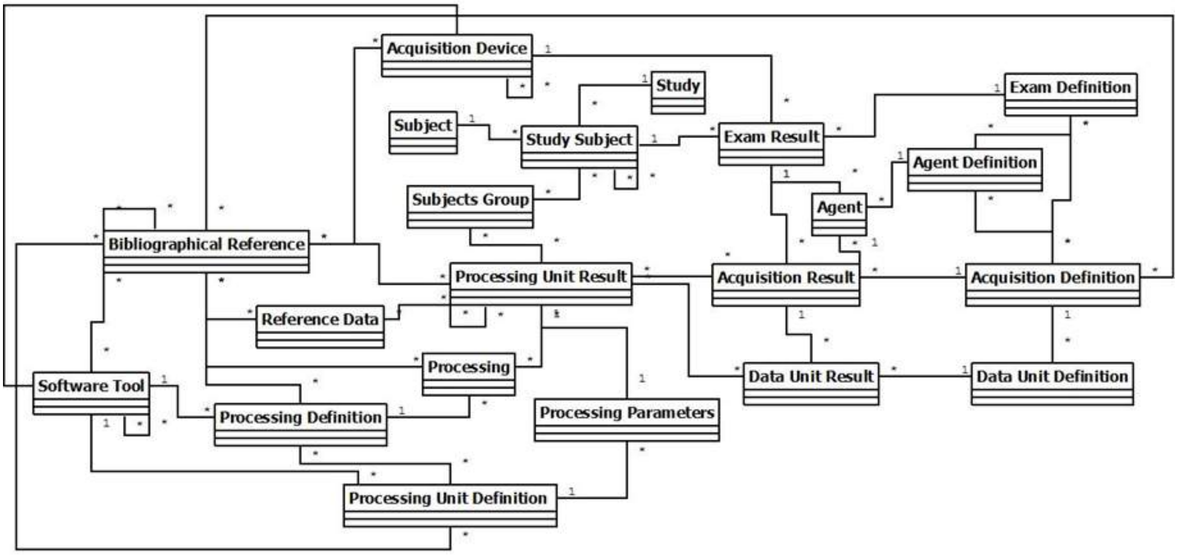


Figure 5: Representation in UML of the BMI-LM data model used at the GIN.

Reusing existing ontologies

The reuse of existing ontologies is necessary to reduce the cost of design and to ensure that the knowledge represented in the constructed ontology is consensual. An application ontology can be constructed from other existing application ontologies.

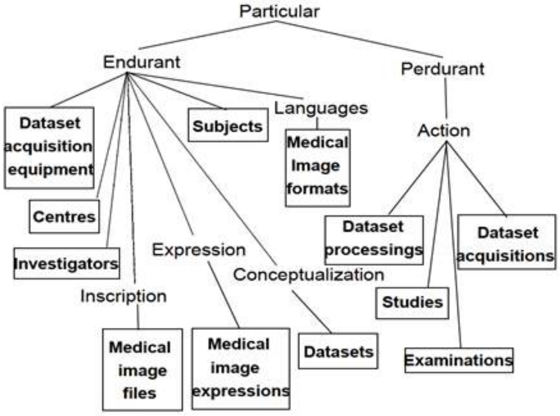


Figure 6: Top concepts of ontoVIP ontology - Project Credible: "Data processing", "data acquisition" and "subject" are reused for the GIN [35].

To identify relevant concepts for application ontologies at the GIN, we studied certain fundamental ontologies (applicable for any domain, such as SUMO, BFO, UFO, DOLCE [36]) and domain ontologies like ontoVip (Figure 6), ontoNeuroLog, QIBO [35]. Some top concepts are reused for the GIN, including dataset processing and dataset acquisition.

Interview to propose application ontologies

Interviews help to understand the processes, users' needs and how data are used by each expert actor. During the interview, we recommend using a knowledge capitalization method like MASK (Method for Analyze and Structuring Knowledge) [37] that provides supported methods and models for capturing and structuring knowledge. Each expert actor is invited to a "common interview" where the approach, purposes and expectations are introduced, and a "private interview" designed to identify the business logic of the domain and how each person uses and manipulates data. Questions that can be proposed during interviews include:

- How is the data named informally, or in other words, what vocabularies are frequently used to call data? What is the definition of each term?
- What is the hierarchy of vocabularies (taxonomy)?
- What are the semantic relationships between the vocabularies used?

Table 1 presents the work positions and manipulated data types of each expert actor who was interviewed during the ontology construction process at the GIN. There are a total of eight expert actors: two assistant engineers who deal with data collection and data processing; one researcher who works with generic data; three researchers working with cognitive and behavioral data; and two researchers who work with anatomical data.

Position	Manipulated data types
Research director	Cognitive data
Group director	Generic data
Researcher	Cognitive neuroimaging, behavioral and functional data
Research engineer	Behavioral data, cohort collection, etc.
Researcher	Anatomical data
Researcher	White fibres, anatomical data
Assistant engineer	Subject Recruitment and Data Collection, MRI
Assistant engineer	Data Processing, MRI, GNB Database Management

Table 1: Summary of scientists who participated in Knowledge Acquisition interviews at the GIN.

During the interviews with expert actors, we also identified the frequently used terms of each interviewer. Many discussions and exchanges contributed to our compilation of a list of consensus terms accepted by all participants. These terms were then used to name the concepts of the application ontology.

CONCEPTS				**Business Object (displayed name in the data model)**	**Business Object (name used in the database)**
rootConcept					
	data				
		acquired-data		Exam Result	GIN4_ExamRes
				Data Unit	GIN4_DataUnit
				Acquisition Results	GIN4_Acquisition
		processed-data		Processing Unit Results	GIN4_ProcessUR
				Processing Results	GIN4_ProcessRes
		reference	reference-bibliography	Bibliography Reference	GIN4_RefBib
			reference-data	Reference Data	GIN4_Template

	investigator				
		acquisition-executive			
		processing-executive			
	process				
		acquisition			
		processing			
	protocol				
		acquisition-protocol		Examination Definition	GIN4_ExamDef
				Acquisition Definition	GIN4_AcqDef
				Data Unit Definition	GIN4_DataUnitDef
		processing protocol		Processing Unit Definition	GIN4_ProcessUD
				Processing Definition	GIN4_ProcessDef
				Processing Parameter	Gin4_ProcessPar
	source				
		study		Study	GIN4_Study
		subject		Subject	GIN4_StudySub
	tool				
		acquisition-tool		Acquisition Device	GIN4_Scanner
		processing-tool		Processing Tool	GIN4_Tool

Table 2: Comparison of the vocabularies used in the BMI-LM data model and in the proposed application ontology.

Table 2 presents the mapping between ontology concepts and business objects in the data model. Thanks to this mapping, when expert actors choose a concept like "acquired-data", they can query "Exam Result", "Data Unit" and "Acquisition Results".

To provide expert actors with an optimal interpretation of application ontologies, we use "indented tree" and "relation graph" from among the ontology visualization techniques presented in section 2.2.

- Expert actors do not only need to consult but also to understand and explore ontology. "Indented tree" is a simple but efficient technique to illustrate the hierarchical structures of ontology. It also provides an easy navigation between concepts that allows users to discover ontological concepts gradually, as well as any "is-a" relationships among them.
- "Relation graph" provides an intuitive visualization of the semantic relationships among concepts. This graph-based representation of ontology (nodes->concepts, edges-> relations) is easily interpreted by expert actors since real world information is also recognized and modeled by the human brain in the same way.

We believe that providing expert actors with both the "indented concept tree" and "relation-graph" techniques in a single interface will help them to maximally exploit ontologies. When users select a concept in the indented tree, all the concepts and relations associated to it are highlighted in the relations graph (as illustrated in Figure 7 with the concept "class").

1.6.2 Graphic convention proposal for self-access

The information needs of expert actors must be specified precisely. To accomplish this, we propose a graph-based query construction approach on the conceptual layer, thanks to indented tree, relations graph and graphical conventions. Users' queries are not formulated syntactically but graphically. The query formulation process becomes the process of constructing a query graph with the help of graphical conventions: nodes represent concepts/ attributes; arcs represent conditions that impose on two nodes, etc. Users' queries are represented graphically in real time during the formulation process. They can be modified by deleting/adding a concept or by changing conditions' operations or attributes' value.

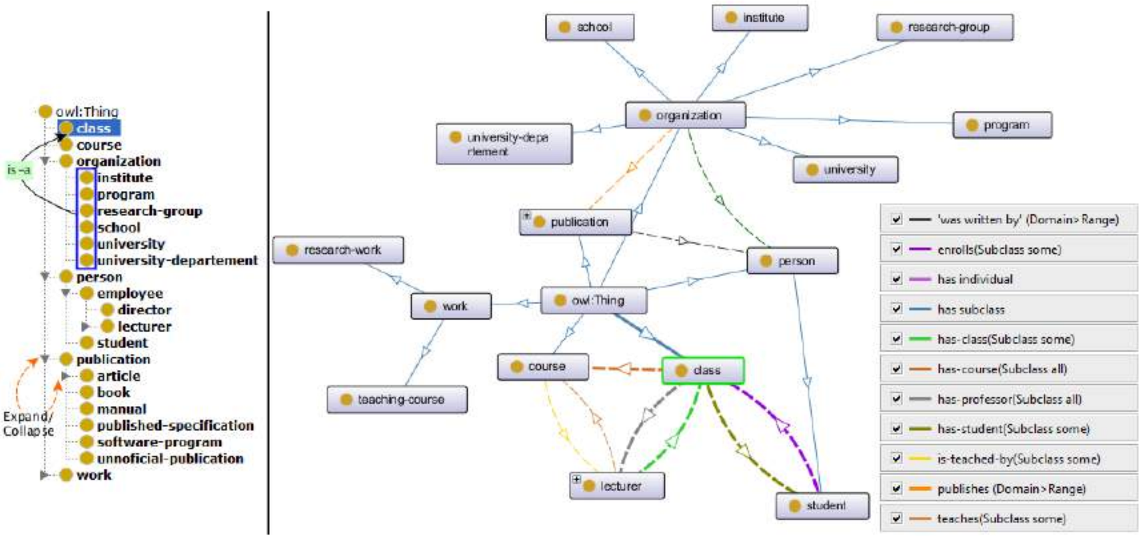


Figure 7: Indented concepts tree and relations graph in a single interface – an optimized method for ontology visualization and exploitation.

New knowledge can be delivered from declared knowledge in application ontologies with the help of inference engines. Therefore, inference engines can be used to enrich users' queries over ontologies before transforming them into formal queries over data sources. However, inference engines should also be implemented together with the interaction and visualization mechanisms and used during the query formulation to help expert actors complete queries corresponding to their information needs. This means relevant concepts and relationships are proposed to expert actors at each query formulation step to enrich and complete their queries. This function becomes especially helpful when they do not have a very clear idea about what they are looking for.

Graphical conventions for users' queries representation

To represent users' queries graphically, we use a set of graphical conventions identified from SPARQL language characters and some studies of Haag et al [28]. and Russell et al. [20]: circles represent nodes, straight lines represent arcs among nodes. These elements are distinguished by shape, size, color and textual sentence (Table 3 and Table 4).

- **Shapes and sizes**:
  - The concepts are represented by circles at the average size.
  - The attributes of a concept ("Datatype") are represented by small circles around the circle representing the concept.
  - The value of an attribute is displayed in a bordered rectangle.
  - The relationships between two concepts ("ObjectProperty"), between a concept and an attribute ("DatatypeProperty"), and the operations between an attribute and its value are represented in unbound rectangles.
- **Colors:**
  - The nodes represented by the concepts are displayed in gray. The nodes represented by the attributes of a concept are displayed in white. The current selected node is displayed in red and at the largest size.

- The rectangles represented by the "ObjectPropery" are displayed in green, the rectangles represented by the "DatatypeProperty" are displayed in purple and the rectangles represented by the operations are displayed in orange.

The graphical conventions presented above are intuitive and easily understood by expert actors, since the way in which queries are represented is more aligned with their mental models of the domain. However, as a consequence, the proposed conventions do not yet cover all aspects of complex queries. Operators like 'OR', 'BETWEEN', 'ORDER BY', etc. cannot yet not be represented directly on the graph. As a solution, in addition to graphical conventions, we can use additional operators (displayed in a pop-up) to complete the query.

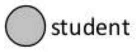
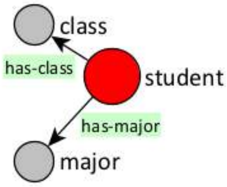
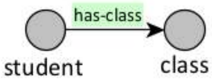
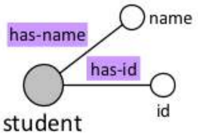
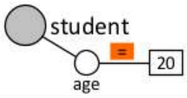
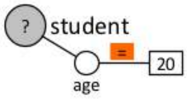
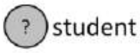
Graphical elements	Description
	The medium-sized gray circle (a node) represents an ontological concept. The name of the concept is displayed next to the circle.
	The current selected node is displayed by a red circle with the largest size. The other nodes remained at their original colors.
	An arc represents a bi-relation between two concepts (nodes). The direction of the arc illustrates the meaning of the relationship. The name of the relationship (ObjectProperty) is displayed in an unblanked green rectangle.
	The attributes of a concept are represented by small white circles displayed around the concept. The DatatypeProperty are displayed in an unbounded purple rectangle.
	The value of an attribute is represented in a rectangle un-bodered orange.
	The interrogation point displayed inside the circle indicates that the values of the corresponding concept will be retrieved in the SELECT query.

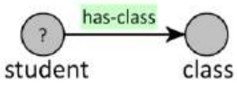
Table 3: Graphical conventions used to represent different components of users' queries.

From users' queries in a graph to SPARQL queries

The graphical query focuses on what the expert actors want to know from the data, rather than how the data is structured in the database. This is also one benefit of SPARQL queries: the initially used RDF data. A graphical query formulated with application ontology can be easily transformed into a formal SPARQL query due to this commonality. Users' queries are then transformed into SPARQL queries thanks to the transformation rules. A SPARQL query is a collection of triples where each triple contains a "subject" connected to an "object" through a "predicate". The query transformation is quite direct: the condition sequences in SPARQL queries are the same as paths among nodes in the users' queries graph. Table 4 shows some examples of query transformation: from the simplest query that queries a single class of data to a query containing conditions on its attributes.

Graphical elements	Description	Corresponding SPARQL query

	Query a class of instances	*SELECT ?std*
		WHERE
		?std rdf:type schl:student.

	Properties are represented by predicates.	*SELECT ?std*
		WHERE
		?std rdf:type schl:student.
		?cl rdf:type schl:class.

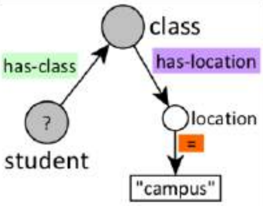
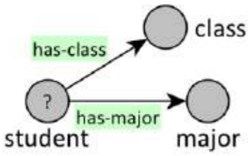
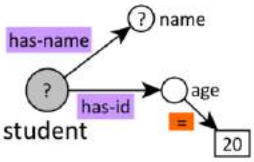
 class has-class has-location ? location student = "campus"	Select instances of one class restricted by conditions on another class. The path to be traced between nodes corresponds to a set of successive triplets.	*SELECT ?std* *WHERE* *?std rdf:type schl:student.* *?cl rdf:type schl:class.* *?lc rdf:datatype schl:location.* *?std schl:has-class ?cl.* *?cl schl:has-location ?lc.* *FILTER (?lc = "campus"^^xsd:string).*
	Select instances of a restricted class by relations with other classes.	*SELECT ?std* *WHERE* *?std rdf:type schl:student.* *?cl rdf:type schl:class.* *?mj rdf:type schl:major.* *?std schl:has-class ?cl;* *schl:has-major ?mj.*
 ? name has-name ? age has-id student = 20	Select instances of a restricted class by conditions on its attributes.	*SELECT ?std, ?name* *WHERE* *?std rdf:type schl:student.* *?name rdf:datatype schl:name* *?age rdf:datatype schl:age* *?std schl:has-name ?name.* *?std shcl:has-age?age.FILTER (?age = 20).*

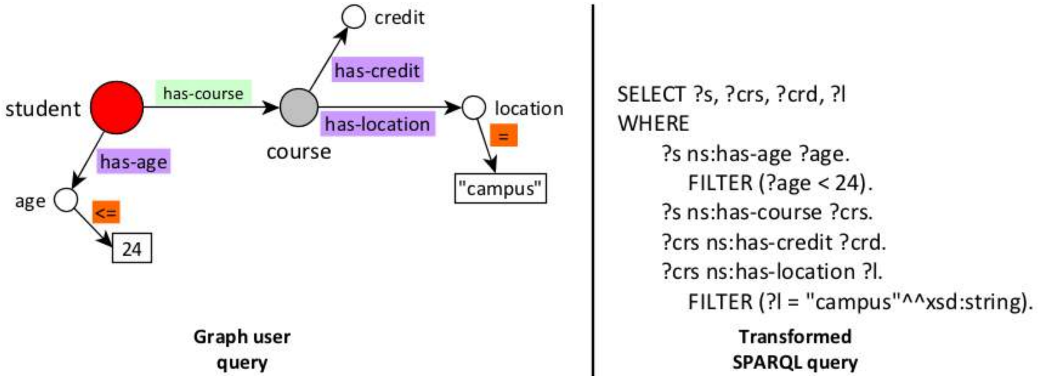
Table 4: Some examples of query transformation: From the user's queries graph to SPARQL queries.

Figure 8 illustrates an example of the SPARQL query and its graphical representation according to the above visualization and transformation rules (Table 4).

From SPARQL queries to formal queries

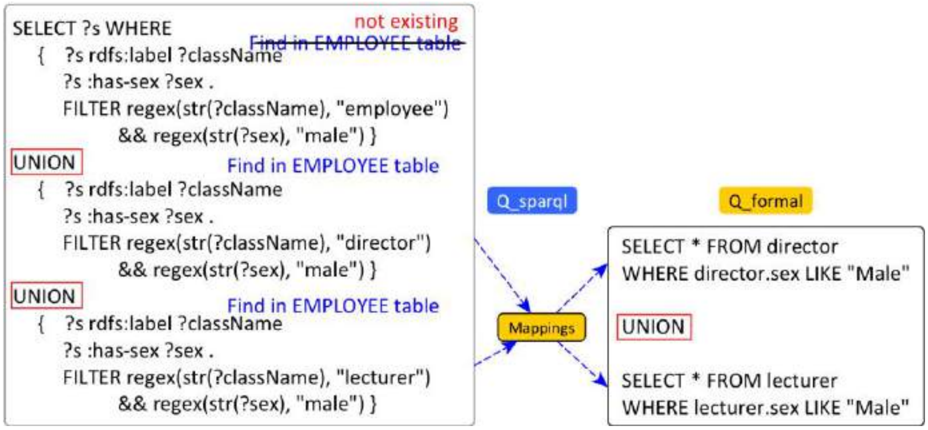
SPARQL queries are transformed into formal queries thanks to mappings. A mapping is an association of two queries: one over ontologies, the other over data sources. In this query transformation step, for each segment of a SPARQL query (Q') we find a set mapping the corresponding query over data sources.

Figure 9 illustrates an example of mapping between two queries, one on the "Director" table of the relational database and one on the ontological concept "Director" in turtle syntax (onTop project and R2RML syntax).



```
SELECT ?s, ?crs, ?crd, ?l
WHERE
      ?s ns:has-age ?age.
         FILTER (?age < 24).
      ?s ns:has-course ?crs.
      ?crs ns:has-credit ?crd.
      ?crs ns:has-location ?l.
         FILTER (?l = "campus"^^xsd:string).
```

Figure 8: An example of a user's query graph and its transformed SPARQL query.



```
SELECT ?s WHERE
   {  ?s rdfs:label ?className
      ?s :has-sex ?sex .
      FILTER regex(str(?className), "employee")
            && regex(str(?sex), "male") }
UNION
   {  ?s rdfs:label ?className
      ?s :has-sex ?sex .
      FILTER regex(str(?className), "director")
            && regex(str(?sex), "male") }
UNION
   {  ?s rdfs:label ?className
      ?s :has-sex ?sex .
      FILTER regex(str(?className), "lecturer")
            && regex(str(?sex), "male") }
```

```
SELECT * FROM director
WHERE director.sex LIKE "Male"

UNION

SELECT * FROM lecturer
WHERE lecturer.sex LIKE "Male"
```

Figure 9: An example of query transformation.

The transformed formal queries over the data sources are then executed by the query evaluation engine of the DBMS. The correlations among results can be identified from the relationships between ontological concepts. Thus, the obtained results can be transformed to the visual instances and then displayed graphically on the query interface. We propose to use data and data visualization techniques that are well developed in order to provide a better graphical and dynamic representation of the results. Navigation and interaction tools should be integrated to help business players interpret and explore data. In the next section, we present a general approach to construct application ontologies to use in the conceptual layer and mappings for querie/result transformation.

Conclusion

The VAQUERO approach is structured to provide all actors with a self-access to data in a heterogeneous expertise environment. To do this, an intermediate layer of integrated adequate visual representations is added between expert actors and their data sources. Each kind of actor uses an appropriate visualization concept corresponding to their technical background to graphically formulate ontological queries. The next section presents an application of the VAQUERO approach at the GIN: VAQUERO system.

VAQUERO QUERY SYSTEM

The VAQUERO query system was developed by following the VAQUERO approach and aims to help all scientists at the GIN to query the Bio-Imaging data managed by PLM solutions.

1.7 Two proposed application ontologies

GIN researchers mainly manipulate two main types of data: imaging data obtained using medical imaging techniques and behavioral/psychological data containing the results of behavioral/ psychological tests. As presented in section 2, the terms used in each proposed ontology should help each type of scientist to recognize the data they are interested in, even if they are named in a specific way in the database. Therefore, the relationships defined between the concepts must represent the semantic relations between the data to which the concepts are connected. As a result of the interviews, we decided to propose two different ontologies: one for researchers who work with imaging data (Figure 10), and another for researchers who work with psychology data (Figure 11). The "imaging data" (MRI – Magnetic Resonance Imaging data for example) and "psychology data" (handedness information for example) are different from each other simply by the nature of the data. These two ontologies are inferred from a more general ontology ("white" part in both figures), which is why they have many concepts in common. All the vocabularies used in these ontologies have been defined though discussions with the GIN scientists to ensure that they are adequate and accessible to the dedicated users. Our approach uses ontology at the application level (application ontology), which is dedicated to a specific sort of dataset. For other datasets, even in the same domain, the proposed application should be adjusted before use.

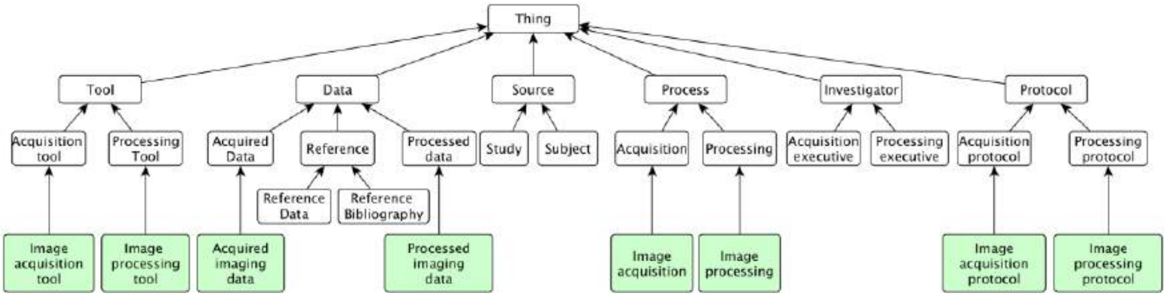


Figure 10: The proposed ontology for researchers working with imaging data.

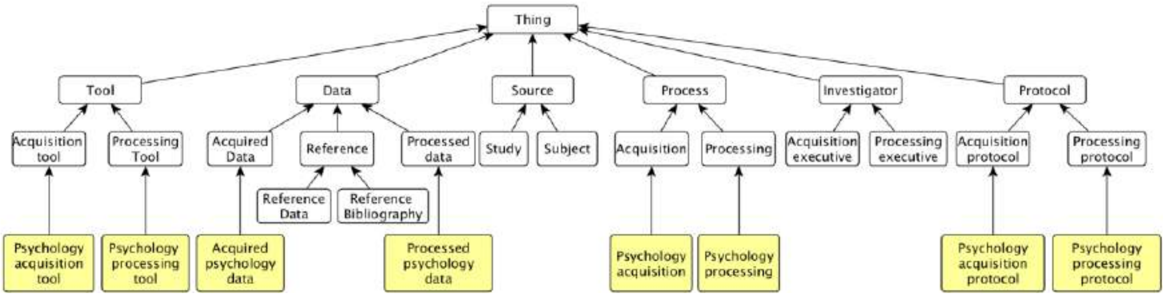


Figure 11: The proposed ontology for researchers working with psychology data.

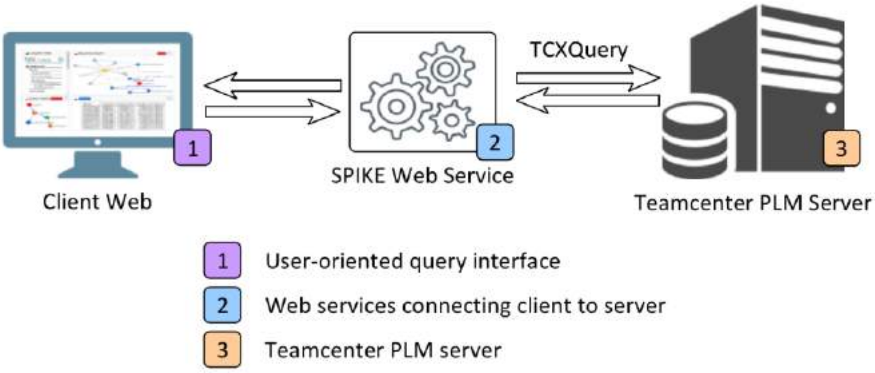


Figure 12: VAQUERO query system infrastructure.

1.8 VAQUERO query system architecture

Figure 12 illustrates the VAQUERO query system with its three main components:

i. The web client is a visual and interactive query interface on which users formulate their data queries.
ii. A web service called SPIKE, which connects the client and the PLM Teamcenter server. Users' queries at the client-side and query's results returned by the PLM Teamcenter server are forwarded bidirectionally through these web services.
iii. The Teamcenter Server PLM: the BMIs are stored in the Teamcenter Database Management System.

The Web Client Interface provides three different layouts for three kinds of users:

i. **"Data model"** *layout*: This layout implements a visualization of BMI-LM (**B**io**M**edical **I**maging **L**ifecyle **M**anagement) data model as it is used by technical users.
ii. **"Classification"** *layout*: Classification is an extension of the BMI-LM data model provided by the PLM Teamcenter DBMS. This layout is dedicated to trained scientists who frequently manipulate the PLM database.
iii. **"Ontology"** *layout*: This layout is dedicated to expert actors – doctors at the GIN. Two ontologies presented in section 1.7 are implemented in this layout.

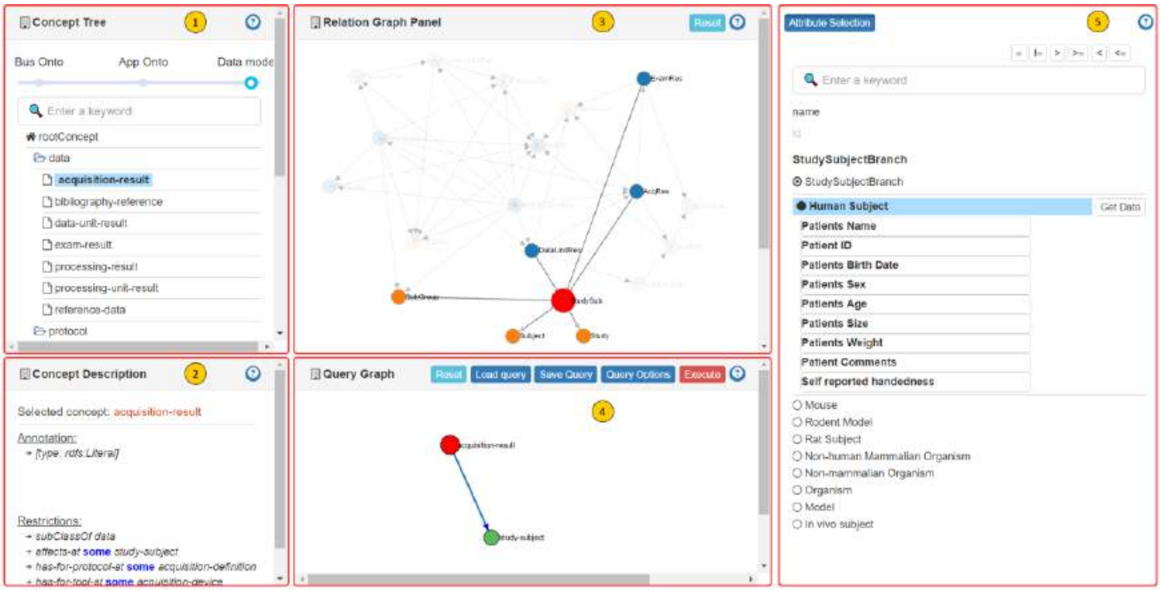
All of the business objects in the data model, the data classification classes or ontology concepts are represented in a navigation tree, while any relationships between them (business objects vs business objects, etc.) are represented in a graphical format which is intuitive and interactive to help users quickly and easily define their queries.

1.9 Query interface

The query interface is web-based, intuitive and interactive. An open-licensed visualization framework is used to provide customers with better visualization of the ontology, classification or data model, such as D3.js. Figure 13 shows a global view of the query interface comprising five areas:

1. **Indented tree**: This area presents an indented tree of objects where an object can be a concept of application ontologies, a class of data or a business object of the BMI-LM data model. Users switch layouts using a cursor at the top of the indented tree.
2. **Description of the current selected object**: This area provides a description of the currently selected object.

3. **Relations graph**: This area graphically illustrates all the relations between objects in the indented tree. When an object is selected in the indented tree, all objects and relations associated to it will be highlighted in this relations graph.
4. **Query graph**: This area shows an image of users' queries in real time.
5. **Query conditions definition**: This area represents attributes of the current selected object in the query graph. To define a query condition, users choose attributes on which conditions are imposed, add operations and set values for the chosen attributes.



Selected concept: acquisition-result

Annotation:
- [type: rdfs:Literal]

Restrictions:
- subClassOf data
- affects-at some study-subject
- has-for-protocol-at some acquisition-definition
- has-for-tool-at some acquisition-device

Figure 13: Implemented query interface in the VAQUERO system.

1.10 Illustrated examples

This section provides real-world examples to demonstrate the feasibility of the VAQUERO system. We present here a query frequently used by scientists working with imaging data.

Search for all "T1-sag" acquisitions made on "left-handed subjects" including "birth data and between 18 and 20 years of age"

On the query interface, this query could be done in three steps:

i. Search all acquisitions of type T1-sag,
ii. Add whose subjects are left-handers,
iii. Add subjects with birth data between 18 and 20 years of age.

For step I, "T1-sag" is an imaging acquisition. To search for all acquisitions of this type, just double click on the "acquisition-imaging-data" concept in the concept tree in order to add it to the graphical query and define the first condition: *acquisition-imaging-data*.**name** = **"T1-sag"** at the right panel.

In step ii, clicking on the "acquisition-imaging-data" node in the graphical query causes the relation graph (panel number 3 in Figure) to highlight it and all related concepts. The user can then select the "subject" concept through the intermediate concept "image-acquisition", add it to the current query and define the condition: *subject*.**handedness** = **"L"** (**L** for *Left-handed* and **R** for *Right-handed*) at the right panel.

Finally, in step iii, the user adds the second condition on the subject concept to complete the initial query: *subject*.**Patients-Birth-Date**> = **"18Y"** and *subject*.**Patients-Birth-Date** <= **"20Y"**.

The query formulation process is illustrated in

Figure 14. This query graph is first transformed into SPARQL query using transformation rules (section 3). The SPARQL query is then transformed into an xQuery query that is executable in the PLM Teamcenter. The results obtained are represented as a graph or as a .csv file (Table 5).

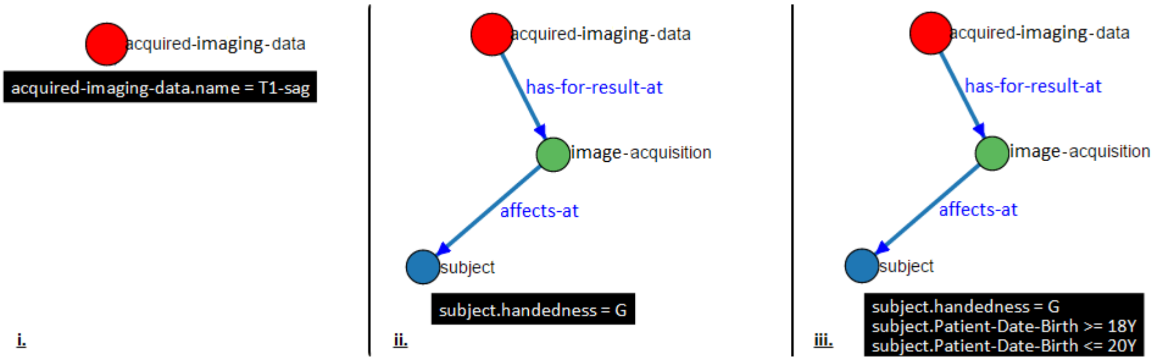


Figure 14: User’s request is formulated step by step.

objectName	Comments	Quality Control	Body Part Examined	Repetition Time	Inversion Time	Flip Angle	Pixel Spacing
ACQima_3D_T1_SAG_ISHARE_V01	20160421	2	BRAIN	2000	880	8	1 ;1
ACQima_3D_T1_SAG_ISHARE_V01		2	BRAIN	2000	880	8	1 ;1
ACQima_3D_T1_SAG_ISHARE_V01	20160427	2	BRAIN	2000	880	8	1 ;1
ACQima_3D_T1_SAG_ISHARE_V01	20160428	2	BRAIN	2000	880	8	1 ;1
ACQima_3D_T1_SAG_ISHARE_V01	20160421	2	BRAIN	2000	880	8	1 ;1
ACQima_3D_T1_SAG_ISHARE_V01		2	BRAIN	2000	880	8	1 ;1
ACQima_3D_T1_SAG_ISHARE_V01	20160422	2	BRAIN	2000	880	8	1 ;1
ACQima_3D_T1_SAG_ISHARE_V01	20160411	2	BRAIN	2000	880	8	1 ;1
ACQima_3D_T1_SAG_ISHARE_V01	20160404	2	BRAIN	2000	880	8	1 ;1
ACQima_3D_T1_SAG_ISHARE_V01		2	BRAIN	2000	880	8	1 ;1

Table 5: Extract of the results of the request 02 in the form of a table .csv.

1.11 Evaluation

We performed a total of three test sessions with the participation of 8t GIN scientists: 1 technician, 3 psychology scientists, 3 imaging scientists and 1 research director. Each participant was asked to select two queries among their frequently used queries and formulate them on VAQUERO. The results show that most of the participants can query needed data in a reasonable time by themselves. For example, the sample query presented above was made in 5 minutes by an imaging scientist after a short training about the query interface and the functions of each component.

1.12 Discussion

In this chapter, we have presented the VAQUERO query system proposed for GIN scientists as an application of the proposed method described in Section 3. The main advantage of the VAQUERO query system is that it enables self-data-access for end-users. Three different layouts are implemented and could be switched easily on the query interface in order to adapt to user's profiles: technical actors, trained actors and expertactors. The interactive mechanisms and graphical conventions used in this web-based application facilitate the construction of users' ontological queries. The query formulation is simple and intuitive enough, even for complicated queries, because of two main factors: First,the user builds queries step by step; she can add or delete any node on the query graph at any moment. The complexity of query graph increases progressively with the number of nodes added and the condition applied on each node. Second, the transformation from graphical query into formal SPARQL query and then SQL query is done automatically and is hidden from the users.

Query\ User	Technician	Psychology Scientist	Imaging Scientist	Research Director
1st query (simple)	2	5	6	8
2nd query (advanced)	3	7	8	10

Table 6: Summary of time required to formulate simple and advanced queries using VAQUERO.

Some points to improve:

- Interactions between different layouts must be developed to ensure a continuation of the query formulation process. In other words, a query being formulated in a layout could be completed and enriched on other layouts.
- An inference engine could be used to help users develop initial queries automatically.
- The query interface should be improved to enhance its usability.

CONCLUSION AND FUTUR WORK

Providing different actors with an easy access to data is difficult. This issue becomes more challenging in a heterogeneous expertise environment, where technical backgrounds are varied and each user has his or her own way to work with data. This paper presented a solution to overcome this issue by using a visual ontology called VAQUERO. The proposed approach is focused on the use of visual representation and ontology; one part for the facilitation of query construction, the other to adapt to the diversity of actors' profiles.

We also present a general but reusable approach for the construction of application ontologies used in the VAQUERO approach. The proposed approach has been tested in the context of PLM for Biomedical Imaging by real use-cases at the GIN laboratory (the VAQUERO query system).

To continue this work, inference mechanisms will be implemented to improve the graphical query formulation. Some techniques of data mining will also be studied to extract inferred information from the query results. We will also concentrate on continued querying, meaning the possibility of further queries on the obtained results.

ORCID

Cong Cuong Pham, https://orcid.org/0000-0001-7521-9945
Alexandre Durupt, https://orcid.org/0000-0001-7668-9703
Benoît Eynard, https://orcid.org/0000-0002-5393-4363

ACKNOWLEDGEMENT

The work presented in this paper was conducted within the ANR (Agence Nationale de la Recherche) funded project BIOMIST (no ANR-13-CORD-0007) for the matic axis no 2 of the Content 2013 Call for Proposal: from content to knowledge and big data. We thank the laboratory

Neurofunctional Imaging Group (GIN-IMN) for sharing data and feedback on the proposed approach in the scope of project BIOMIST.

REFERENCES

[1] Chen, Y.; Wang, W.; Liu, Z.; Xuemin, L.: Keyword search on structured and semi-structured data. SIGMOD'09 Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, 2009, 1005–1010, https://dl.acm.org/citation.cfm?doid=1559845.1559966.

[2] Kaufmann, E.; Bernstein, A.: Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases. Journal of Web Semantics, 8, 2010, 377–393, https://doi.org/10.1016/j.websem.2010.06.001.

[3] Safari, L.; Patrick, J.D.: Restricted natural language based querying of clinical databases. Journal of Biomedical Informatics, 52, 2014, 338–353, https://doi.org/10.1016/j.jbi.2014.07.012.

[4] Waltinger,U.; Tecuci, D.; Olteanu, M.; Mocanu, V.; Sullivan, S.: Natural Language Access to Enterprise Data. AI Magazine; 35(1), 2014, https://doi.org/10.1609/aimag.v35i1.2502 .

[5] Shneiderman, B.: The future of interactive systems and the emergence of direct manipulation. Behaviour & Information Technology,1, 1982, 237–256, https://doi.org/10.1080/01449298208914450.

[6] Ori, B.-Y.; Nadav, G.; Nadav, H.; Lempel, R.; Neumann, A.; Ofek-Koifman, S : Beyond basic faceted search. WSDM '08 Proceedings of the 2008 International Conference on Web Search and Data Mining, 33–44, 2008, https://dl.acm.org/citation.cfm?doid=1341531.1341539.

[7] Catarci, T.: What happened when database researchers met usability. Information Systems;25, 2000, 177–212, https://doi.org/10.1016/S0306-4379(00)00015-6.

[8] Catarci, T.; Costabile, M.F.; Levialdi, S.; Batini, C.: Visual Query Systems for Databases: A Survey. Journal of Visual Langages & Computing, 8, 1997, 215–260, https://doi.org/10.1006/jvlc.1997.0037.

[9] Epstein, R.-G.: The TableTalk query language. Journal of Visual Langages & Computing, 2, 1991,115–141, https://doi.org/10.1016/S1045-926X(05)80026-6.

[10] Gruber, T.-R.: Toward principles for the design of ontologies used for knowledge sharing? International Journal of Human-Computer Studies, 43, 1995, 907–928, https://doi.org/10.1006/ijhc.1995.1081.

[11] Borst, W.-N.; Borst, W.-N.: Construction of Engineering Ontologies for Knowledge Sharing and Reuse, Centre for Telematics and Information Technology (CTIT), University of Twente, Faculty of Electrical Engineering, Mathematics & Computer Science, 1997, https://research.utwente.nl/en/publications/construction-of-engineering-ontologies-for-knowledge-sharing-and-

[12] Studer, R.; Benjamins, V.-R.; Fensel, D.: Knowledge engineering: Principles and methods. Data & Knowledge Engineering, 25, 1998, 161–197, https://doi.org/10.1016/S0169-023X(97)00056-6.

[13] Giese, M.; Calvanese, D.; Haase, P.; Horrocks, I.; Ioannidis ,Y.; Kllapi, H.Koubarakis, M.;Lenzerini, M.; Moller, R.; Ozcep, O.; Rodriguez-Muro, M.;Rosati, R.;Schlatte, R.;Schmidt, M.;Soylu, A.;Waaler, A.: Scalable End-user Access to Big Data. In: Akerkar R, editor. Big Data Computing, CRC Press; 2013, 1–49, http://www.inf.unibz.it/~calvanese/papers-html/book-big-data-2013.html

[14] Calvanese, D.; Liuzzo, P.; Mosca, A.; Remesal, J.; Rezk, M.; Rull, G.: Ontology-based data integration in EPNet: Production and distribution of food during the Roman Empire. Engineering Applications of Artificial Intelligence, 51, 2016, 212–229, https://doi.org/10.1016/j.engappai.2016.01.005.

[15] Smart, P.-R.; Russel, A.; Braines, D.; Kalfoglou, Y.; Bao, J.; Shadbolt, N.-R.: A Visual Approach to Semantic Query Design Using a Web-Based Graphical Query Designer.

International Conference on Knowledge Engineering and Knowledge Management, EKAW 2008 : Knowledge Engineering: Practice and Patterns, Part of the Lecture Notes in Computer Science book series (LNCS, volume 5268), 2008, 275–91, https://link.springer.com/chapter/10.1007%2F978-3-540-87696-0_25

[16] Hogenboom, F.; Milea, V.; Frasincar, F.; Kaymak, U.: RDF-GL: A SPARQL-Based Graphical Query Language for RDF. Part of the Advanced Information and Knowledge Processing book series (AI&KP), 2010, 87–116, https://link.springer.com/chapter/10.1007/978-1-84996-074-8_4

[17] Zainab, S.-S.; Saleem, M.; Mehoomd, Q.; Zehra, D.;Decker, S.; Hasnai, A.: FedViz: A visual interface for SPARQL queries formulation and execution. VOILA : Visualizations and User Interfaces for Ontologies and Linked Data Workshop co-located with ISWC 2015, 2015, https://www.insight-centre.org/content/fedviz-visual-interface-sparql-queries-formulation-and-execution

[18] Borsje, J.; Embregts, H.; Graphical Query Composition and Natural Language Processing in an RDF Visualization Interface. Bachelor Thesis, Supervised by: Flavius Frasincar; 2006, http://webcache.googleusercontent.com/search?q=cache:yZDs2W5vAaYJ:citeseerx.ist.psu.edu/viewdoc/download%3Fdoi%3D10.1.1.137.5037%26rep%3Drep1%26type%3Dpdf+&cd=1&hl=fr&ct=clnk&gl=fr&client=firefox-b-e

[19] Katifori, A.; Halatsis, C.; Lepouras, G.; Vassilakis, C.; Giannopoulou, E.: Ontology visualization methods—a survey. Journal ACM Computing Surveys (CSUR),39, 2007, https://dl.acm.org/citation.cfm?id=1287620.1287621.

[20] Russel, A.; Smart, P.-R.; Braines, D.; Shadbolt, N.-R.: NITELIGHT: A Graphical Tool for Semantic Query Construction. Semantic Web User Interaction Workshop (SWUI 2008), 2008, https://eprints.soton.ac.uk/264975/

[21] Wang, T.-D.; Parsia, B.: CropCircles: Topology Sensitive Visualization of OWL Class Hierarchies. International Semantic Web Conference, ISWC 2006: The Semantic Web - ISWC 2006, Part of the Lecture Springer Link, Notes in Computer Science book series (LNCS, volume 4273), 2006, 695–708, https://link.springer.com/chapter/10.1007/11926078_50

[22] Eklund, P.; Roberts, N.; Green, S.: OntoRama: Browsing RDF ontologies using a hyperbolic-style browser. First International Symposium on Cyber Worlds, 2002. Proceedings, 2002, https://ieeexplore.ieee.org/document/1180907

[23] Jankun-Kelly, T.; Ma, K.-L.: MoireGraphs: radial focus+context visualization and interaction for graphs with visual nodes. IEEE Symposium on Information Visualization 2003 (IEEE Cat. No.03TH8714), 2003, https://ieeexplore.ieee.org/document/1249009.

[24] Sintek, M.: OntoViz 2007. https://protegewiki.stanford.edu/wiki/OntoViz

[25] Pietriga, E.: IsaViz: A Visual Authoring Tool for RDF 2007, https://www.w3.org/2001/11/IsaViz/.

[26] Plaisant, C.; Grosjean, J.; Bederson, B.: SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation. IEEE Symposium on Information Visualization, INFOVIS 2002, https://ieeexplore.ieee.org/document/1173148.

[27] Liebig, T.; Noppens, O.: OntoTrack: A semantic approach for ontology authoring. Journal of Web Semantics;3, 2005, 116–131, https://dl.acm.org/citation.cfm?id=1741319.

[28] Haag, F.; Lohmann, S.; Siek, S.; Ertl, T.: QueryVOWL: A Visual Query Notation for Linked Data. The Semantic Web: ESWC 2015 Satellite Events, 2016, 387–402, https://dl.acm.org/citation.cfm?id=2991147.2991192

[29] Harith, A.: TGVizTab: An ontology visualisation extension for Protégé. Knowledge Capture (K-Cap'03), Workshop on Visualization Information in Knowledge Engineering, Sanibel Island, 2003, http://oro.open.ac.uk/20054/

[30] Bongwon, S.; Bederson, B.: OZONE: A Zoomable Interface for Navigating Ontology, University of Maryland, College Park, MD 20742, 2001, http://www.cs.umd.edu/hcil/ozone

[31] Allanic, M.; Hervé, P.-Y.; Pham, C.-C. ; Lekkal, M.; Durupt, A. ; Brial, T. ; et al.: BIOMIST: A Platform for Biomedical Data Lifecycle Management of Neuroimaging Cohorts. Frontier in

ICT, Computer Image Analysis 2017, https://doi.org/10.3389/fict.2016.00035

[32] Pham, C.-C, Matta, N.;Durupt, A.; Eynard, B.; Joliot, M.; Boutinaud, P.: Using Ontologies to access complex Data : Applications on Bio-Imaging. 4th AI4KM (IFIP WG 12.6 conference) on IJCAI, 2016, https://link.springer.com/chapter/10.1007/978-3-319-92928-6_2

[33] Pham, C.-C.; Matta, N.; Durupt, A.; Eynard, B.; Allanic, M.; Ducellier, G.; et al.: How to share complex data and knowledge: Application in Bio-Imaging. 8th IFAC Conference on Manufacturing Modelling, Management & Control, 2016, https://doi.org/10.1016/j.ifacol.2016.07.589

[34] Chujai, P.; Kerdprasop, N.; Kerdprasop, K.: On Transforming the ER Model to Ontology Using Protégé OWL Tool. International Journal of Computer Theory and Engineering, ,6, 2014,484–489, http://www.ijcte.org/index.php?m=content&c=index&a=show&catid=59&id=1092

[35] Gibaud, B.;Forestier, G.; Benoit-Cattin, H.; Cervenansky, F.; Clarysse, P.; Friboulet, D.; et al. : OntoVIP: An ontology for the annotation of object models used for medical image simulation. Journal of Biomedical Informatics, 52, 2014, 279–292, https://www.sciencedirect.com/science/article/pii/S1532046414001555

[36] Guizzardi, G.: Ontological foundations for structural conceptual models. University of Twente, 2005, https://research.utwente.nl/en/publications/ontological-foundations-for-structural-conceptual-models

[37] Matta, N.; Ermine, J.-L, Aubertin, G.; Trivin, J.-Y.: Knowledge Capitalization with a Knowledge Engineering Approach: The Mask Method. In: Springer; Boston M, editor. Knowledge Management and Organizational Memories, 2002, https://link.springer.com/chapter/10.1007/978-1-4615-0947-9_2