An Analysis Framework for Ontology Querying Tools

Fajar J. Ekaputra Estefanía Serral Dietmar Winkler Stefan Biffl

Vienna University of Technology, Christian Doppler Laboratory CDL-Flex
Favoritenstrasse 9-11/188, AT 1040 Vienna, Austria
+43/(0)1/58801-18810

<firstname>.<lastname>@tuwien.ac.at

ABSTRACT
While knowledge querying is a key capability of ontologies, the query language recommended by W3C, SPARQL, is not easy to use for some user types, e.g., casual users and domain experts. To improve this drawback, user-friendly Ontology Query Tools (OQTs) have been introduced. However, there is, to our knowledge, no comprehensive framework for researchers and practitioners to compare the capabilities of the wide range of available OQTs. In this paper we introduce, based on a systematic literature review, a framework that allows researchers and practitioners to classify and compare OQTs regarding their capabilities and their support for relevant user types and scenarios. We evaluate the framework based on a real-world use case. Major result of the evaluation was that the framework was found useful and usable by users from the target audience to identify the most suitable OQTs for their context.

Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information Search and Retrieval – *query formulation, information filtering.*

H.5.2 [**Information Systems**]: User Interfaces – *evaluation/methodology, interaction styles.*

General Terms
Measurement, **Human Factors**

Keywords
User Interaction, Ontology Querying, Ontology Tools, Analysis Framework

1. Introduction
Ontologies have become the backbone of mission-critical systems, such as automation systems [1] or medical and genomic systems [2]. Ontologies also serve as foundation of the semantic web technology [14], which provides a large and interlinked network of information fragments coming from distributed data publishers all around the world. The rapid growth of semantic data representation to a staggering 31.5 billion triples for users of linked data[1] creates a high demand to make the ontology technology easier to use. In the context of ontology querying, SPARQL[2] is the language most used by ontology experts, as it is a precise and effective ontology querying language recommended by W3C. However, some user groups have difficulties in using SPARQL to query ontologies [3][7].

Ontology users can be classified in the following categories or groups [3]: *lay users or casual users* are mainstream users, who do not need to understand semantic web technologies in depth; *domain experts*, who have a deep knowledge in the domain but only limited expertise in semantic web technologies; and *technical users or ontology experts*, who are experts in semantic web technologies. While ontology experts can directly use SPARQL for querying ontologies, lay users and domain experts typically have a steep learning curve to use the formal language effectively and efficiently [3][7]. Therefore, lay users and domain experts need tool support to query data with little knowledge about formal ontology languages or ontology data structures.

This challenge has received recently notable attention [3][4][8] from the semantic web research communities and numerous *Ontology Query Tools* (OQT) have been proposed to bridge the gap between informal user query goals and formal query languages. Common approaches are: (a) to provide an OQT as a stand-alone tool, e.g., Querix [6], Nitelight [9] and MashQL [4]; or (b) to embed an OQT as a part of a comprehensive ontology visualization tool, e.g., Protégé[3] and Tabulator[4]. Thus, a major issue is to evaluate to what extent the provided tools support expected user requirements and scenarios.

While several survey papers (e.g., [3], [5], and [8]) reported on classification and comparison frameworks for ontology visualization tools, these frameworks do not address the problem of classifying OQTs for specific user groups and do not address key capabilities such as the detail of formal query capabilities or the support for result visualization.

This paper introduces a comprehensive OQT analysis framework to enable OQT classification and comparison regarding their capabilities and support for the intended user groups. Using this framework, researchers and practitioners can evaluate OQT tools and select the tool that fits best to a given usage context. We provide an evaluation concept based on real-world use cases and present results and lessons learned from a pilot application of the OQT analysis framework.

The remainder of the paper is structured as follows. Section 2 summarizes related work; Section 3 motivates the research issues

I-SEMANTICS 2013, 9th Int. Conf. on Semantic Systems, Sept. 4-6, 2013,Graz, Austria

[1] http://lod-cloud.net/state/#domains

[2] http://www.w3.org/TR/rdf-sparql-query/

[3] http://protege.stanford.edu/

[4] http://www.w3.org/2005/ajar/tab

and the research approach; Section 4 summarizes the Systematic Literature Review process and result. Section 5 explains the user classification and UI design guidelines. Section 6 explains the criteria analysis and classification. Section 7 introduces the framework design and the framework process guideline. Section 8 reports on the feasibility evaluation and the evaluation result. Section 9 discusses the evaluation results, and finally, Section 10 concludes and suggests ideas for further work.

2. Related Work

This section summarizes related work on semantic web technology surveys and semantic web comparison frameworks.

In the area of semantic web technology surveys Katifori *et al.* provided one of the earliest surveys on ontology visualization tools [5]. They categorize the tools within two dimensions: ontology visualization type and the employed space dimension. The authors also propose several elements, e.g., overview windows and zooming function to be displayed within ontology visualization tools and point out the capabilities such as keyword-search function, evolution and time visualization within the ontology. Recently, Dadzie *et al.* provided a comprehensive discussion of Linked Data (LD) browsers and suggested two sets of guidelines: a general guideline for analytic activity and knowledge discovery for a complex and large data set and a guideline for LD consumption requirements [3]. These surveys provide a number of attributes, e.g., faceted search support and visual representation to classify ontology-related tools, but mainly focus on ontology visualization tools, not on OQTs.

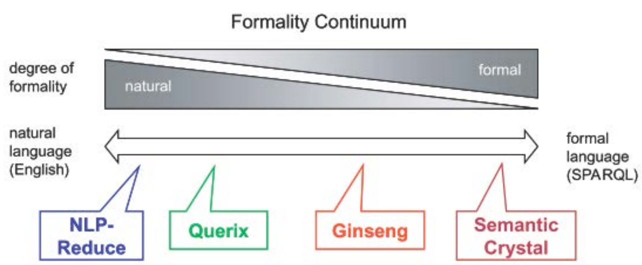

Figure 1: OQTs positioned in the Formality Continuum [7].

Regarding semantic web frameworks, Kaufmann *et al.* propose an interesting classification framework for OQT user interfaces (OQT-UIs), the *Habitability Hypothesis* [7]. This hypothesis suggests that the best solution for lay users for querying ontologies lies in the middle of a Formality Continuum between natural language and formal language (see Figure 1). The authors evaluate their hypothesis by developing four OQTs within the Formality Continuum[5] and they proved it through user evaluation. These four OQTs are NLP-Reduce, Querix, Ginseng, and Semantic Crystal. *NLP-Reduce*, a naïve Natural Language (NL) OQT, that supports keywords and full natural language querying. *Querix* is a limited NL-OQT, which requires the query to start with certain words and end with a question mark or full stop. *Ginseng* is a guided input NL-OQT that provides predetermined, controlled, and menu-based words/sentences akin to English. *Semantic Crystal* is a graphical OQT with a formal and restrictive intermediary notation.

Jarrar *et al.* [4] contributed another important classification framework for ontology query formulation tools. This work takes into account some querying aspects, e.g., query formulation user interface types; however, they only focus on query formulation and do not evaluate important OQT aspects such as OQT support for formal query capabilities and visual ontology notation. Although these semantic web frameworks define and analyze relevant aspects of semantic web tools, none of them provides a comprehensive analysis framework for OQTs with respect to individual requirements and scenarios of different user groups.

3. Research Issues and Approach

The goal of this paper is to provide researchers and practitioners with a framework that allows them to classify, compare, and find the appropriate OQT for their individual user requirements and scenarios. From this goal, we derived the following two research issues: (a) design of an OQT comparison framework, and (b) feasibility study of the designed OQT analysis framework.

RI-1: Design of a flexible OQT analysis framework to rate OQT candidates according to user requirements. The goal of this research issue is to provide a comprehensive OQT analysis framework to help stakeholders to evaluate them and find the best option for a given context (i.e., use case and user groups). We build on the semantic web technologies surveys and framework [3][5][7] to provide an approach focused on the querying aspects instead of visualization or tools in general. We also analyzed stakeholder types and needs regarding query interfaces.

The framework has to address two groups of OQT criteria to classify OQTs, i.e., functional and non-functional criteria. Furthermore, we also propose a scoring system based on these criteria to enable researchers to specify their OQTs requirements in the analysis and selection process.

RI-2: Feasibility study of the OQT analysis framework. The goal of this research issue is to evaluate the feasibility of the presented approach regarding the OQT analysis framework, i.e., collect initial evidence on the usability and usefulness of the framework for the target user groups. We conduct an evaluation based on the Systematic Literature Review (see Section 5) to test the feasibility of the approach and to identify improvement options for better addressing individual stakeholder needs.

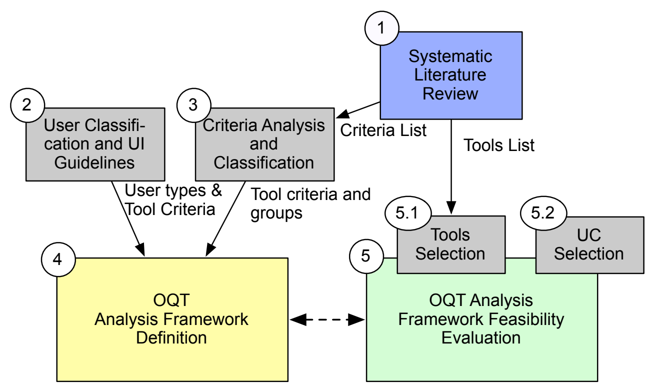

Figure 2 Research Process of the Analysis Framework.

This research approach followed in this paper adapts and extends the one proposed by Poston *et al.* for software testing tools [17]. The approach consists of the following steps (see Figure 2): (1) Systematic Literature Review on ontology query tools and user interface design; (2) classification of users types for the framework; (3) criteria analysis and classification; (4) design of the framework by defining its components and the process for using the framework; and (5) feasibility evaluation of the framework with real-world use cases.

[5] https://files.ifi.uzh.ch/ddis/oldweb/ddis/research/talking-to-the-semantic-web/

4. Systematic Literature Review

Systematic Literature Review (SLR) is a method to identify, evaluate, and interpret in a pre-defined scope all relevant research documents to address a particular research question, topic area, or a phenomenon of interest [15]. We conducted an adapted version of the SLR process as we focused on the content of the key papers (see process steps below).

1. Define goal of the SLR. The goal of the SLR is (a) to collect candidates for relevant OQT classification criteria for an OQT analysis framework and (b) to identify the reports on OQTs, which are candidates for the tool evaluation. This SLR is an initial work, and will be further extended in the future work.

2. Scope of the SLR to identify and select primary studies. The scope of SLR sources for the primary studies should cover the conference proceedings and journals most relevant for the OQT scientific community. Therefore, we included the proceedings of six conferences[6] and four journals[7] that can be considered as most relevant in the semantic web field. The SLR covers 1204 papers from the last 5 years, i.e., papers published in 2008 to 2013 and fitting the following keywords: *query OR sparql OR user OR interface OR visual OR interactive OR graphic*. We are searching for primary studies in the selected conferences and journals manually from DBLP[8] and the official website of conferences and journals in February 2013, which resulted in 78 papers.

3. Inclusion and exclusion criteria. In order to select the most relevant papers from the results of the previous step, we used the following inclusion and exclusion criteria. *Inclusion Criteria***:** Full papers that provide an approach or survey or classification framework regarding ontology querying tools and/or relevant ontology tools (e.g., ontology visualization tools and ontology query user interface). *Exclusion Criteria***:** Papers that do not introduce tools or interfaces related to ontology querying and/or relevant ontology tools. The process of applying these inclusion and exclusion criteria to the full papers reduced the number of papers finally to 24 papers[9], which were used for the extraction of candidate for criteria and tools (see Sections 7 and 8).

5. User Classification and UI Guidelines

This section explains the process of user classification and user interface guidelines, which provide user types and OQT criteria that are important inputs to the framework.

User Classification. Dadzie *et al.* [3] introduced two primary types of users: lay users and ontology experts. They also suggested the domain expert only as a sub-type of the lay user. In contrast to [3], we model domain experts as a distinct third user type (see Table 1), since domain experts have a deeper knowledge of the domain than lay users in general. We decide against distinguishing ontology experts who know and do not know the domain, since they have the same ability to do querying in expert level. Therefore, we base the framework on three user types.

Table 1 User Types.

	Novice Semantic Web knowledge	**Expert Semantic Web knowledge**
Novice domain knowledge	Lay user	Ontology expert
Expert domain knowledge	Domain expert	

Lay users (LU) have limited domain knowledge and semantic web technologies. They typically do not need to perform complex querying tasks, and their queries are limited only to simple data retrieval. This user group will likely require an OQT closer to NLI in the Formality Continuum to address their lack of knowledge in formal ontology query language.

Domain experts (DE) have a limited knowledge of semantic web technologies but a very good understanding of data structure and content in their domain. Therefore, they may need to do complex querying, ontology validation, or even ontology modification and updating. This group of user will likely need an OQT that supports more complex queries, while hiding the complexity of the formal ontology query language.

Ontology experts (OE) have a good understanding of semantic web technologies. These users should be capable to user RDF as data format, and also to interpret ontology models. They should not have any problem in using formal language for their tasks. Hence, an OQT closer to the formal language would allow them forming more complex queries. One example of OQT that explicitly designed for ontology experts is Nitelight [9].

User interface guidelines can provide input to group a large number of specific criteria to make an evaluation framework easier to understand and handle. There is a general guideline to design user interfaces for information visualization [16] consisting of three parts: *Overview*, *zooms and filter*, and *details-on-demand*. The guideline also provides the following tasks at the highest level of abstraction: overview, zoom, filter, details-on-demand, relate, history, and extract. This guideline line provides for the framework the criteria group *Query Visualization*.

Shneiderman proposed a framework for querying task UI guidelines [10], which is structured into six groups: 1) *Functions*, i.e., operations that users perform on the data; 2) *Tasks*, i.e., part of functions that have to be done to be able to operate a function; 3) *Interaction Modes* are facilities to access the data system; 4) *Retrieval Response Types* refer to the response produced by data system after querying; 5) *Query Features* are provided by query language; and 6) *User Types* categories. From this UI framework we extract two criteria groups for the framework: *Query Features* and *Interaction Modes*.

6. Criteria Analysis and Classification

This section provides the criteria analysis and classifications for the framework based on the SLR results and User Interface guidelines. We divide the criteria into two groups: 1) Functional criteria, which are directly related to the user and could have different levels of importance for different user types; and 2) Non-functional criteria, which contain several criteria unrelated to certain type of users, but relevant for specific UCs, e.g. license. While license have nothing to do with the targeted user type, several use cases could use it to filter out non-compatible OQTs.

Criteria from the SLR results. From the SLR results, we extracted the following functional criteria: *(1) Formal query support:* formal querying support, *(2) User clarification UI:*

[6] Extended Semantic Web Conference, International Semantic Web Conference, International Conference on Computational Semantics, International Conference on Semantic Systems (i-Semantic), International Conference on Knowledge Engineering and Knowledge Management, and Web Reasoning and Rule Systems Conference Series

[7] Journal of Web Semantics, Semantic Web Journal, Journal of Data Semantics, and International Journal On Semantic Web and Information Systems

[8] http://www.informatik.uni-trier.de/~ley/db/

[9] http://juang.me/node/6

allows the user to refine her queries; *(3) Ontology source customization:* allows the user to configure the ontology querying sources; and *(4) Query result visualization:* categorizes the query result visualization types available. See details on the criteria in Table 2.

In addition to the functional capabilities, we also identified the following *non-functional criteria* to rate the suitability of OQTs for specific use cases: (1) formal ontology query language, (2) license type and (3) availability. More detail about these non-functional criteria will be provided in Section 7.1.

Criteria groups from UI Design Guidelines. Based on the UI guidelines reported in Section 5, we derived the following criteria groups for the analysis framework: *(1) Query features*, which is adapted to SPARQL features, (2) *Natural Language UI and Graphical Querying UI*, which is the UI type of OQT and (3) *Query result visualization*, which derived from the general UI guidelines. These will be explained further in the Section 7.1.

7. Design of the Analysis Framework

Based on the results from the analysis steps reported in Sections 4, 5, 6, and also related work from Section 2, we identified the OQT criteria to classify requirements for the target user groups.

7.1 OQT Criteria and Groups

This section explains in detail both the functional criteria and the non-functional criteria identified for our framework. For each functional criterion, we also provide the measurement scale to enable scoring for tool comparison and selection.

Functional Criteria. Table 2 shows the criteria relevant for the user groups, sub-criteria, and their measurement.

Table 2 Functional Criteria

Criteria (sub-criteria)		**Definition & Reference**	**Measurement Scale**
Formal Query Support		Capability to directly use formal query from the tools [3][20]. Support may vary between OQTs, depending on what query language is supported, e.g., SPARQL, SeRQL	Yes/No
Interaction Modes / UI	NL-UI	Availability of natural language user interface for querying ontology [4][6][7]. This interface type usually takes form in a keyword search.	Yes/No
	Graphical-UI	Availability of graphical user interface for querying ontology [4][6][7]. There are various implementations of the UI, e.g., form-based search, query formulation mash-up.	Yes/No
	User Clarification	Availability of system-suggested refinement user interface of user queries [20][27], e.g., Google auto-complete feature,	Yes/No
Query Features	Select	SELECT feature of the query language [10][9][13], i.e., extract raw values from the ontology, which results in tables.	Yes/No
	Ask	ASK feature of the query language [10][9][13], i.e., confirm whether a fact in ontology is true or not; ask for a Boolean query.	Yes/No
	Describe	DESCRIBE feature of the query language [10][9][13], i.e., to provide a description about specific information within the ontology.	Yes/No
	Construct	CONSTRUCT feature of the query language [10][9][13], i.e., create new information based on the information available in the ontology.	Yes/No
	Insert	INSERT feature of the query language [10], i.e., insert new information into the ontology.	Yes/No
	Update	UPDATE feature of the query language [10], i.e., update information within the ontology	Yes/No
	Delete	DELETE feature of the query language [10], i.e., delete information from the ontology.	Yes/No
	Solution sequences and modifier	Availability of solution sequencing and modifier features in the query language (i.e., Order by, Projection, Distinct, Offset, and Limit), to modify the query according to the user [10][20][24].	0-2 (1=part, 2=full)
Ontology Source Configuration	Ontology Source Customization	Ability to change the source of ontology that the user queries from. [24][27]	Yes/No
	Multiple Source	Allow users to define queries from multiple sources, i.e., to let the user combine several sources to get the result from his/her single query. [19][20][22]	Yes/No
	Query Concatenation	Feature that allow users to define queries from his/her previous query result [16]	Yes/No
Query Result Visualization	Plain Visualization	Result visualization in triples or other raw format for ontology [16][9][22]	Yes/No
	Text Visualization	Textual representation of query result [16][19][21]	Yes/No
	Graphic Visualization	Graphical representation of query result [16][20]	Yes/No
	Result Faceted or filtering	Ability to filter or clarify the result using some of the results attribute, e.g., sort by time, filter out result before year 1970 [16][18][19]	Yes/No

Non-Functional Criteria. Table 3 shows non-functional criteria that can be used by stakeholders as informal decision support but are currently not rated in the framework.

Table 3 Non-Functional Criteria.

Criteria	Definition & Reference
Formal ontology query languages	Formal languages supported by the OQT, e.g., SPARQL and SeRQL, [6][9][13].
License Type	License type of the OQT, e.g. Apache, GPL, LGPL, MIT, commercial [24][18].
Availability	Source, demo, and documentation available for the tools. [4][19][20][22]
Ontology-Engine	Engine / library used for executing the queries in the OQT, e.g., Apache Jena[10], Sesame[11], OWL API[12], etc. [9][24][27]
R&D Period	OQT Research period and the current status. [4][7]
Independency	Type of the OQT, whether as stand-alone tool or add-on to a complex tool. [9][24]
Technology	The implementation technologies (e.g., library, programming language) used within the OQT, both for front-end and back-end. [7][9][13]
Intermediary Notation	Availability of intermediary notation for querying and its explanation. [4][9][13]

7.2 Guidelines for Using the Framework

This paper introduces an OQT analysis framework to enable OQT classification and comparison regarding their capabilities and support for the intended user groups, which are supported by the scoring system and usage steps.

Scoring system. We have defined the measurement scale for each functional criterion in Table 2. The values collected from this scale will be called a ***Criteria Score***. If a tool supports the criteria feature, its score sums one point.

The second part of the scoring system is called the ***User-Group Score***. This score reflects the needs for each criterion for each user groups and can be obtained, for instance, by performing a survey on user groups' needs. An example will be provided in the feasibility evaluation explained in Section 8..

The last part of the scoring system is called ***Scenario-Specific Score*** (see Table 5), which is the multiplication of Criteria Score and User-Group Score for a specific scenario (e.g., SemaPlorer has 57% score for lay users, which means it has features that satisfy 57% of lay users needs for our scenario). These scores, used to compare and analyze the OQTs (especially their functional criteria), are calculated by applying the following formula, where: *i* represents each criterion, *CS* represents the criteria score, *UGS* represents the user group score for each criteria, and *SSC* reflects the final tool rating:

$$SSC = \frac{\sum_{i=1}^{n} CS(i).UGS(i)}{\sum_{i=1}^{n} CS_{max}(i).UGS_{max}(i)}$$

Analysis, Rating, and Comparison. The first two steps for using the framework are: (1) to select the OQTs that users want to analyze; and (2) to define the use case and target user types for the scenario in which the OQT is needed. The optional third step is the definition of User-Group Score. While this framework provides User-Group Score for each criterion per user type based on our feasibility evaluation use case (see also Table 4), if needed, users can also customize the score to suit better their requirements. We provide our steps to define the user-group score in our feasibility evaluation in Section 8.

After these steps are finished, users can proceed to use the analysis framework as follows: (1) OQT Filtering Process, filter and analyze OQTs using the framework's non-functional criteria, and (2) OQT Scoring and Analysis, functional criteria scoring and for further analysis and classification (See Figure 3).

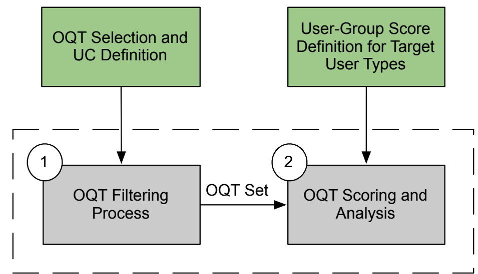

Figure 3 Steps of the OQT Analysis Framework.

8. Feasibility Evaluation

After defining the OQT analysis framework and its guidelines, the next step of the research process is to conduct a feasibility evaluation to collect initial evidence on the usefulness and usability of the framework. Population sample of this evaluation were seven friendly researchers at TU Wien with different levels of knowledge on semantic web technologies and on the domain (multidisciplinary systems engineering) as proxies for target users of the framework. These researchers aged 25 – 48 and had at least a Master Degree in computer science.

Ontology Query Tool Selection. The SLR results provided the following list of OQTs for the evaluation:

1) **QuestIO** [23] (2008), a natural language interface for accessing structured information that is domain independent and uses Sesame as ontology library.

2) **DSpace-Semantic-Search** [24] (2011), a structured semantic query UI that helps construct and submit entailment-based queries and uses OWL-API as ontology library.

3) **Nitelight** [9] (2008), a web-based graphical tool for semantic query construction that is based on the W3C SPARQL specification and uses Jena as ontology library.

4) **SWIP** [13] (2012) is short for Semantic Web Interface Using Patterns situated in the Semantic Web framework; the paper does not provide information about the ontology library used.

5) **AutoSPARQL** [27] (2011), a User Interface that implements an active learning approach on top of the QTL algorithm and uses Jena as ontology library.

6) **Visor** [25] (2011), ontology data browser that focuses on the multi-pivot approach; the paper does not provide information about the ontology library used.

7) **Quick** [26] (2010), a novel system for that helps users to construct semantic queries in a given domain and uses Sesame as ontology library.

8) **FREyA** [18] (2010) an NLI OQT that combines syntactic parsing with the knowledge encoded in ontologies domain and uses Sesame as ontology library.

[10] http://jena.apache.org
[11] http://www.openrdf.org/
[12] http://owlapi.sourceforge.net/

9) **PowerAqua** [22] (2011), an ontology-based question answering system that uses Sesame and Virtuoso as ontology engines.

10) **Q2Semantic** [21] (2008), a lightweight keywords interface to semantic search. This OQT and uses Sesame as ontology library.

11) **SemaPlorer** [20] (2009), an interactive semantic exploration of data and media based on federated cloud infrastructure and uses Sesame as ontology library.

12) **VisiNav** [19] (2009), a system based on interaction model designed that explores easy search and navigation of large amounts of web data from multiple sources; the paper does not provide information about the ontology library used.

Table 4 Criteria Score by User Group (UGS).

Criterion (sub-criterion)		LU	DE
Plain Query Support		**1.5**	**3.5**
Interaction Modes / UI	NL-UI	5.0	4.5
	Graphical-UI	3.0	3.0
	User Clarification	**4.0**	**2.0**
Query Features	Select	**1.5**	**5.0**
	Ask	3.0	4.5
	Describe	2.0	3.5
	Construct	**2.0**	**4.0**
	Insert	**3.0**	**5.0**
	Update	**3.0**	**5.0**
	Delete	3.0	4.0
	Solution sequences & modifier	3.5	5.0
Ontology Source Configuration	Ontology Source Customization	2.5	4.0
	Multiple Source	**2.0**	**4.5**
	Query Concatenation	3.5	4.5
Query Result Visualization	Plain Visualization	2.5	2.0
	Textual Visualization	4.5	5.0
	Graphical Visualization	3.5	4.5
	Result Faceted / filtering	3.5	5.0

Context and Population Sample. The context for the use case is the Engineering Knowledge Base (EKB) [12], an ontology-based framework for data integration in multidisciplinary engineering systems. For achieving data integration in the EKB, ontology querying is an essential requirement. However, some EKB users have limited knowledge in semantic web technologies (i.e., domain expert or lay users in our user type) are reluctant to use SPARQL for data querying [3][7] and, therefore, a good audience for a framework to find an appropriate OQT.

Within the feasibility evaluation, seven selected EKB users used the framework to get suggestions regarding the best OQT solution for their use. They also provided feedback whether the suggested OQT were actually good enough for their needs.

User Group Score Definition. The participating users were classified to user groups according to a Likert scale from 1 (no knowledge) to 5 (very good knowledge) to rate the participant background knowledge in semantic web technologies and the domain multidisciplinary engineering, resulted in two lay users, two domain experts, and three ontology experts.

The participants rated each functional criterion (see Section 5) on a Likert scale from 1 (not important) to 5 (very important). From the collected data the score for user group was derived (see Table 4, notable score differences are printed in bold). We focused on key OQT users: lay users and domain experts.

Evaluation Process and Result. The evaluation process follows the steps described in Section 7.2.

(1) **OQT Filtering.** While in some contexts it may be necessary to filter out OQTs not compatible to the use cases, the EKB use case does not limit the OQT selection. Hence, all twelve OQTs provided from previous step are rated.

(2) **OQT Scoring and Analysis.** Within this step we calculate the criteria and scenario-specific score for each OQT based on our survey within CDL community and analysis of the OQTs.

Table 5 shows the scenario-specific scoring chart from evaluating the 12 OQTs with the evaluation framework. Two tools were found to be notably stronger than the others: *SemaPlorer* with 55% points and *Visor* as a close second with 50%. Then there is a group of tools with scores between 29% and 35% points. The lowest scoring tools achieve just 20% points or less.

Table 5 Criteria and Scenario-Specific Scoring table (SSC).

OQT	Criteria (C)	C Rank	LU	LU Rank	DE	DE Rank
SemaPlorer	55%	1	57%	1	55%	1
Visor	50%	2	53%	2	49%	2
NITELIGHT	35%	3	29%	8	32%	4
AutoSPARQL	30%	4	35%	4	31%	5
FREyA	30%	4	37%	3	31%	5
VisiNav	30%	4	33%	5	34%	3
Dspace	29%	7	30%	7	31%	5
Q2Semantic	29%	7	33%	5	28%	8
PowerAqua	20%	9	18%	11	19%	10
Quick	20%	9	23%	9	17%	11
SWIP	20%	9	22%	10	23%	9
QuestIO	10%	12	11%	12	11%	12

For lay users, the first and second place do not differ with the criteria score, with *SemaPlorer* and *Visor* lead the scoring charts with 57% and 53% points respectively. The tool, which scores third place in the criteria score rating, Nitelight, falls back strongly in the ranking for lay users. Our analysis suggests that this fall is caused by the unavailability of features that have high lay user-group scores, e.g., NL-UI and textual result visualization.

FREyA and AutoSPARQL have better ranks for lay users compared to their ranks for domain experts. Similar to Nitelight both lack features for domain experts, e.g., the availability of ask query and solution sequences and modifier. As the result, their rank drops lower than other OQTs that provide such features, i.e. VisiNav and Nitelight.

Table 6 shows the complete result from OQT criteria score. Most tools provide a NLI UI, the Select feature, and text visualization. However, no tool provides Insert, Update, and Delete features or query concatenation. Only few tools provide plain query support, ask, describe, and construct query features, and visualization types other than text visualization.

Table 6 Overview on OQT analysis framework feasibility evaluation results.

Framework Criteria		QuestIO	Dspace-Search	NITE-LIGHT	SWIP	AutoSPARQL	Visor	Quick	FREyA	Power-Aqua	Q2-Semantic	Sema-Plorer	VisiNav
Functional Criteria													
Plain Query Support		No	Partial	No	No	No	No	No	No	No	Partial	Yes	No
Query-UI	**NLI-UI**	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Graphical-UI	No	No	Yes	No	No	Yes	Yes	No	No	No	Yes	N/A
	User Involvement UI	No	No	No	No	Yes	Yes	Yes	Yes	No	Yes	Yes	N/A
Query-Features	**Select**	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Ask	No	No	Yes	Yes	No	Yes	No	No	No	No	No	No
	Describe	No	No	Yes	No	No	Yes	No	No	No	No	No	No
	Construct	No	No	Yes	No	No	Yes	No	No	No	No	Yes	No
	Insert	N/A	No	No	No	No	No	No	No	No	No	No	N/A
	Update	N/A	No	No	No	No	No	No	No	No	No	No	N/A
	Delete	N/A	No	No	No	No	No	No	No	No	No	No	N/A
	Solution Sequences & Modifier	N/A	Partial	Partial	Partial	Partial	N/A	N/A	N/A	N/A	N/A	Yes	N/A
Ontology Source	**Ontology Source Customization**	N/A	Yes	N/A	No	Yes	N/A	N/A	N/A	N/A	N/A	No	N/A
	Multiple Ontology Source Support	N/A	No	No	No	No	N/A	N/A	N/A	Yes	N/A	Yes	Yes
	Query Concatenate	N/A	No	No	No	No	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Result Visualization	**Plain Visualization**	N/A	No	Yes	N/A	No	No	N/A	No	Yes	No	No	No
	Text Visualization	N/A	Yes	No	N/A	Yes	Yes	N/A	Yes	Yes	Yes	Yes	Yes
	Graphic	N/A	No	No	N/A	No	Yes	N/A	Yes	No	Yes	Yes	Yes
	Faceted result / filter	N/A	No	No	N/A	N/A	Yes	N/A	Yes	No	No	Yes	Yes
Non-Functional Criteria													
Formal ontology query languages		SeRQL, SPARQL	SPARQL	SPARQL	SPARQL	SPARQL	SPARQL	SPARQL	SPARQL	SeRQL, SPARQL	SPARQL	SPARQL	N/A
License Type		N/A	Apache 2.0	N/A	N/A	N/A	N/A	N/A	Open source	N/A	N/A	N/A	N/A
Availability		N/A	Source code	N/A	N/A	Source code	Video	N/A	Source code	Demo	N/A	Source code	Video
Ontology-Engine		Sesame	OWL API	Jena	N/A	Jena	N/A	Sesame	Sesame	Sesame, Virtuoso	N/A	Sesame	N/A
R&D Period		2008	2011 - Now	2007-2008	2010 - Now	2011-Now	2011-Now	2010	2012	2011-Now	2008	2009	2009
Independency		Stand-alone	Part of System	Stand-Alone	Stand-alone	Stand-alone	Stand-alone	Stand-alone	Stand-alone	Stand-alone	Stand-alone	Part of System	Stand-alone
Technology		Java	OWL2, Maven	Java-Script	Java, Supple Parser	Java-script, Java	Html5, JQuery, Django	Java	Java, Lucene	Java	Ajax	Lucene Sail	N/A
Intermediary Notation		N/A	N/A	VQL	Pivot Query	N/A	Visual Notation	N/A	N/A	N/A	N/A	N/A	N/A

*) N/A means that there is currently no clear information about the value of criteria for the tool

9. Discussion

The experience of creating the OQT analysis framework and its feasibility evaluation warrant a discussion of lessons learned on strengths and limitations we found regarding the research issues.

RI-1: Design of an OQT analysis framework. The structure of the analysis framework for OQTs in steps (including user classification, criteria analysis based on systematic literature review results, and UI guidelines, and the definition of analysis framework) helped to ensure a systematic repeatable approach. The framework presents five main functional criteria groups and provides several sub-criteria to support the researchers and practitioners in analyzing OQT for their specific target user groups and UCs. The strengths of the frameworks lie on the ability to provide flexible analysis on different user groups. As a user of the framework, it is possible to customize the user-group score of each. As the reported capabilities of the tools did not follow a clear pattern in the evaluation, the user-defined scoring seems to be favorable compared to uniform scoring.

RI-2: Feasibility of the OQT analysis framework. EKB users, who analyzed the result, pointed out that the OQTs scores provide a useful insight for their OQT selection, even with the limited scope of the initial evaluation. One of the strong points of the framework was found to be the ease of use, which facilitates a relatively comfortable learning curve for using the framework. This is a promising result, which encourages extending the scope of the research, e.g., consult a wider scope of research reports in the systematic literature review, discuss with a wider population sample the importance of non-functional criteria for their work with OQTs.

A limitation we found was the unavailability of data on some criteria for several tools (see cells with "N/A" in Table 6), which make providing a comprehensive report more challenging.

Threats to validity. As any empirical study the feasibility study presented in this paper has to deal with threats to validity. The sample population of the feasibility study was a relatively small group of researchers from one research institution, which introduces a bias into the evaluation. Therefore, more research is needed with an extended sample population to strengthen external validity of the empirical research.

10. Conclusion and Further Work

In this paper we presented an OQT analysis framework to support user- and user-group-specific OQT rating based on the results of a systematic literature review [15]. We provided an evaluation concept and conducted an initial feasibility evaluation based on a real-world use case. Major result of the feasibility evaluation was that the framework was found useful and usable by users in the target audience to identify the most suitable OQTs for their key scenarios.

From the feasibility evaluation we found some notable facts. First, most OQTs in the analysis provide a natural language UI. While the implementation varies, even sophisticated graphical querying tools also provide keyword search for the user. Second, the absence of modification query supports in all OQTs that we have selected is a limitation that contrasts user group needs. While the typical OQT aims for lay users that typically do not need modification query support, we should not ignore that the domain experts would expect to use this kind of query (see also Table 5).

As the results of the initial empirical feasibility study are promising, we plan to develop an OQT recommender system [28] to simplify the use of the framework and accumulate knowledge regarding OQTs. We found several advanced criteria from the SLR study that we did not include in the framework criteria, but consider to add in the future, e.g., spatial and temporal dimension of result visualization [11][20] and the possibility to add more OQT UI types [4]. We also plan to expand our SLR in scope of sources (i.e., more conferences, journals, and digital libraries) and time span to ensure strong coverage of OQTs and user needs.

Acknowledgements

This work has been supported by the Christian Doppler Forschungsgesellschaft, Österreichischer Austauschdienst (OeAD) and the BMWFJ, Austria.

References

[1] T. Moser, S. Biffl, W. D. Sunindyo, and D. Winkler, “Integrating Production Automation Expert Knowledge Across Engineering Domains,” in *IJDST*, vol. 2, no. 3, pp. 88–103, 2011.

[2] Feigenbaum, Lee, Ivan Herman, Tonya Hongsermeier, Eric Neumann, and Susie Stephens. "The semantic web in action." *Scientific American Magazine* 297, no. 6 (2007): 90-97.

[3] Dadzie, A. S., & Rowe, M. (2011). Approaches to visualising Linked Data: A survey. *Semantic Web*, *2*(2), 89–124.

[4] Jarrar, M., & Dikaiakos, M. D. (2012). A Query Formulation Language for the Data Web. *IEEE Transactions on Knowledge and Data Engineering*, *24*(5), 783–798.

[5] Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., & Giannopoulou, E. (2007). Ontology visualization methods—a survey. *ACM Computing Surveys (CSUR)*, *39*(4), 10.

[6] Kaufmann, E., & Bernstein, A. (2007). How useful are natural language interfaces to the semantic web for casual end-users? *The Semantic Web*, 281–294.

[7] Kaufmann, E., & Bernstein, A. (2011). Evaluating the Usability of Natural Language Query Languages and Interfaces to Semantic Web Knowledge Bases. *Web Semantics: Science, Services and Agents on the World Wide Web*, *8*(4).

[8] Ramakrishnan, S., & Vijayan, A. (2012). A study on development of cognitive support features in recent ontology visualization tools. *Artificial Intelligence Review*, 1–29.

[9] Smart, P., Russell, A., Braines, D., Kalfoglou, Y., Bao, J., & Shadbolt, N. (2008). A visual approach to semantic query design using a web-based graphical query designer. *Knowledge Engineering: Practice and Patterns*, 275–291.

[10] B. Shneiderman, “Improving the human factors aspect of database interactions,” *ACM Transactions on Database Systems*, vol. 3, no. 4, pp. 417–439, Dec. 1978.

[11] S. Batsakis and E. Petrakis, “SOWL: spatio-temporal representation, reasoning and querying over the semantic web,” in *I-SEMANTICS*, 2010, p. 15.

[12] W. D. Sunindyo, T. Moser, D. Winkler, and S. Biffl, “Foundations for Event-Based Process Analysis in Heterogeneous Software Engineering Environments,” in *SEAA 2010*, pp. 313–322.

[13] C. Pradel, “Allowing End Users to Query Graph-Based Knowledge Bases,” *Knowledge Engineering and Knowledge Management*, pp. 8–15, 2012.

[14] Hitzler, Pascal, Markus Krotzsch, and Sebastian Rudolph. *Foundations of semantic web technologies*. Chapman and Hall/CRC, 2011.

[15] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering (version 2.3). Technical report, Keele University and University of Durham, 2007.

[16] Shneiderman, B. (1996, September). The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings. IEEE Symposium on* (pp. 336-343). IEEE.

[17] Poston R.M., Sexton M.P.: *Evaluating and selecting testing tools*, IEEE Software (9/3), 1992.

[18] D. Damljanovic, M. Agatonovic, and H. Cunningham, “Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction,” *The Semantic Web: Research and Applications*, pp. 106–120, 2010.

[19] A. Harth, “VisiNav: A system for visual search and navigation on web data,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 8, no. 4, pp. 348–354, 2010.

[20] S. Schenk, C. Saathoff, S. Staab, and A. Scherp, “SemaPlorer—interactive semantic exploration of data and media based on a federated cloud infrastructure,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 4, pp. 298–304, 2009.

[21] H. Wang, K. Zhang, Q. Liu, T. Tran, and Y. Yu, “Q2semantic: A lightweight keyword interface to semantic search,” *The Semantic Web: Research and Applications*, pp. 584–598, 2008.

[22] V. Lopez, M. Fernández, E. Motta, and N. Stieler, “PowerAqua: Supporting users in querying and exploring the Semantic Web,” *Semantic Web*, vol. 3, no. 3, pp. 249–265, 2012.

[23] V. Tablan, D. Damljanovic, and K. Bontcheva, “A natural language query interface to structured information,” *The Semantic Web: Research and Applications*, pp. 361–375, 2008.

[24] D. Koutsomitropoulos, R. Borillo Domenech, and G. Solomou, “A structured semantic query interface for reasoning-based search and retrieval,” *The Semantic Web: Research and Applications*, pp. 17–31, 2011.

[25] I. Popov, M. Schraefel, W. Hall, and N. Shadbolt, “Connecting the dots: a multi-pivot approach to data exploration,” *The Semantic Web–ISWC 2011*, pp. 553–568, 2011.

[26] G. Zenz, X. Zhou, E. Minack, W. Siberski, and W. Nejdl, “From keywords to semantic queries—Incremental query construction on the Semantic Web,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 166–176, 2009.

[27] J. Lehmann and L. Bühmann, “AutoSPARQL: Let users query your knowledge base,” *The Semantic Web: Research and Applications*, pp. 63–79, 201

[28] Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." *Knowledge and Data Engineering, IEEE Transactions* on 17.6 (2005): 734-749.