

User Experience Benchmarking and Evaluation to Guide the Development of a Semantic Knowledge Graph Exploration Tool

Roberto García ^{a,*}, Juan-Miguel López-Gil ^b and Rosa Gil ^a

^a *Universitat de Lleida, Spain*

E-mails: roberto.garcia@udl.cat, rosamaria.gil@udl.cat

^b *University of the Basque Country, Spain*

E-mail: juanmiguel.lopez@ehu.eus

Abstract. Despite the increasing amount of semantic data available, there is still a lack of adoption of user-facing applications based on semantic technologies, especially those geared towards the exploration of disparate semantic datasets. Benchmarks have already been identified as drivers of the advancement of different domains and till recently there was not a benchmark of semantic data exploration tools. Building on top of the Benchmark for End-User Structured Data User Interfaces (BESDUI), we explore now how it can guide the development of a new tool for semantic knowledge graphs exploration, RhizomerEye. The results at the current stage of development show better results than those for the RhizomerEye predecessor. However, there is the risk of overfitting the tool to the benchmark, overloading the user interface to produce the best benchmark results but producing an unusable UI. To rule this out, an evaluation with real users has been also conducted, using the same dataset and tasks provided by the benchmark, but involving real users to measure the User Experience instead of deriving the UX metrics analytically. Moreover, the evaluation has been complemented with the user satisfaction dimension, unmeasurable by the benchmark. Overall the results are promising, showing comparable results to those of the benchmark, especially for users with knowledge about semantic technologies. On the other hand, the evaluation with real users has made it possible to identify potential RhizomerEye improvements, also taking into account user satisfaction, and ways to better suit BESDUI to be used in evaluations with real users.

Keywords: Semantic data, Benchmark, Exploration, Usability

1. Introduction

Through open data efforts like the Linked Open Data Cloud [1], web pages annotated using schema.org [2] motivated by Search Engine Optimisation benefits or the increasingly popular Knowledge Graphs [3], the amount of semantic data is increasing. However, this has not translated into an increasing adoption of user-facing applications based on semantic technologies [4]. In fact, in most circumstances, this is the desired result. The benefits of semantic technologies should be evident on the server side, while client applications should isolate end-users from the underlying complexities. This is the case for task-specific applications with a user interface tailored to the task at hand.

*Corresponding author. E-mail: roberto.garcia@udl.cat.

However, there are cases where no custom client applications exist to hide the underlying semantic data from end users. This is especially true for users exploring semantic data they want to reuse, especially if it is the combination of multiple datasets [5], or semantic data they are generating and want to inspect to assert its quality. Tech-savvy users can use standards like SPARQL to explore semantic datasets. However, most users are unable to do this, and even developers who are conversant with SPARQL struggle when querying unfamiliar datasets [6]. Consequently, multiple efforts to make semantic data exploration interactive and intuitive are being made [7].

Nevertheless, due to this heterogeneity, it is difficult to compare them, particularly from the user's point of view. Therefore, a reference benchmark was required [8] and our proposal was the Benchmark for End-User Structured Data User Interfaces (BESDUI) [9]. BESDUI showed its usefulness as a benchmark for existing structured data exploration tools, including both semantic and non-semantic data.

Continuing this line of work, we focus now on other areas where BESDUI can make a contribution. First of all, as a way to guide the development of a new semantic data exploration tool, RhizomerEye. In this regard, BESDUI can be easily integrated with the development process because it does not require the involvement of real users. The benchmark is based on a set of predefined tasks and data plus the analysis of the interaction steps required to complete the tasks. This has helped during RhizomerEye development, providing an unbiased and very generic testing framework that also facilitates repeatability.

Additionally, it is important to note that BESDUI does not replace tests with real users, which are still required to measure subjective aspects of the user experience like learnability or user satisfaction [10]. To this end, and also to rule out that basing RhizomerEye development on the benchmark does not result in overfitting or user interface overloading, this paper also contributes an evaluation involving real users using the same data and tasks as the benchmark. From the analysis of the interaction logs provided by all participants in the study, it has been possible to assess all interaction stages based on the KLM model and the real-time required to complete each of the tasks using RhizomerEye, when they were able to accomplish it. Even though the time required to fulfil proposed tasks is almost four times larger than the analytical results for the average of all participants, the results are substantially closer when we limit the comparison to those participants that reported previous knowledge about SPARQL and semantic web platforms, who are the main target group for RhizomerEye.

The rest of this paper is organized as follows. First, we present the related work in Section 2, focusing on that related to user experience benchmarking and evaluation in the context of the Semantic Web. Then, in Section 3, we report on our experience using BESDUI to drive the development of a new semantic data exploration tool, RhizomerEye. In Section 4, we detail the study with real users we have carried out using BESDUI data and tasks, which included the evaluation of RhizomerEye's effectiveness, efficiency, and satisfaction in use. Finally, Sections 5 and 6 present the discussion and the conclusions and future work respectively.

2. Related Work

Given the challenges for user interaction with a growing space of semantic data, there are multiple attempts to develop user-facing applications in a wide range of domains, such as healthcare [11], energy [12] or digital humanities [13]. More generic tools are also being developed [14], ranging from Linked Data browsers [15] to Controlled Natural Language query engines [16] or faceted browsers [17].

However, they are difficult to compare and evaluate due to a lack of well-established practices of User Experience (UX) evaluation tailored to the Semantic Web domain [18]. There are some proposals based on the evaluation with real users, ranging from the exploration and visualization of the ontologies underlying semantic data [19] to semantic data exploration tools [20]. However, comparability is difficult when users need to be involved and it is impossible to guarantee reproducibility at scale.

The most promising tool in this kind of situation is benchmarking, with a long tradition of helping to organize and strengthen research efforts in a particular research area [21]. An example is the Text REtrieval Conference (TREC) benchmarks [22], which have become the de facto standard for evaluating any text document retrieval system. Also, there are success stories in areas related to the Semantic Web like the Ontology Alignment Evaluation Initiative [23].

However, in connection with semantic web and user interaction, there have been just a few efforts based on quite informal criteria, like the Intelligent Exploration of Semantic Data Challenge¹. The Benchmark for End-User Structured Data User Interfaces (BESDUI) [9] is the first one focusing on user interaction with structured data, thus including semantic data. BESDUI is the starting point for the contributions made in this paper. First, regarding the use of this benchmark to guide the development of a new semantic data exploration tool. Second, about evaluating the suitability of the benchmark in comparison to an evaluation with real users, thus not based on the analytical approach BESDUI is based on.

3. Developing a Semantic Data Exploration Tool

This section reports on the development of a new version of Rhizomer [17], an existing semantic data exploration and visualization tool. Rhizomer was developed before BESDUI was available, though it is among the tools that have been evaluated using this benchmark. Rhizomer already showed very good results for the benchmark, but its analysis highlighted many improvement opportunities.

Despite the good results, and taking into account the potential improvements already identified, Rhizomer was based on an aged technological stack and its architecture made it very difficult for others to use it without going through a complicated deployment process. This triggered the development of a new version, called RhizomerEye [24], which was developed from scratch but guided by the same approach that had already shown its usefulness with the existing version.

This approach is motivated by the aim of exploring a semantic dataset without any prior knowledge about its structure or the underlying semantic data and query languages. And it is addressed through the three classical data analysis tasks proposed by Shneiderman [25]: getting an overview of the data, zooming, filtering, and viewing details on demand. Each of these tasks is further detailed in the following subsections, including details about how the BESDUI benchmark influenced the development of the RhizomerEye's features supporting each of these tasks.

The added value of driving the development using the BESDUI benchmark is that the tasks under consideration are just based on the typical information needs for data exploration considered in the Berlin SPARQL Benchmark (BSBM) Explore Use Case². They were conceived without considering the user interaction dimension, just considering typical information needs and the SPARQL queries required to satisfy them. And that is our objective when developing RhizomerEye, to develop a tool capable of satisfying these fundamental information needs when exploring a dataset, and make them possible through a user interface that requires the minimum amount of interaction steps to complete them. This approach is useful during the development process, as detailed in the next subsections. However, this analytical approach ignores important aspects like user interface complexity resulting from overloading it or user satisfaction. To address these shortcomings, we have also conducted an evaluation with real users, as reported in Section 4.

3.1. Overview

This is the first of the data analysis task supported by RhizomerEye. It allows users to get the full picture of the dataset. Rhizomer automatically generates a word cloud to provide an overview of the kinds of things in the dataset. This is the default overview mechanism because it works even for really big datasets like DBpedia, with more than 100 million statements, as the 300 most common classes are displayed.

For a more informative overview that also includes how the main classes relate among them, there is also the option of a network representation. In this case, the 30 most instantiated classes are shown as nodes together with the most frequent properties connecting them as labelled edges. Figure 1 shows an example of the network overview of the BESDUI benchmark dataset based on the BSBM benchmark.

¹IESD Challenge, <https://iesd2015.wordpress.com/iesd-challenge-2015>

²Berlin SPARQL Benchmark (BSBM), Explore Use Case

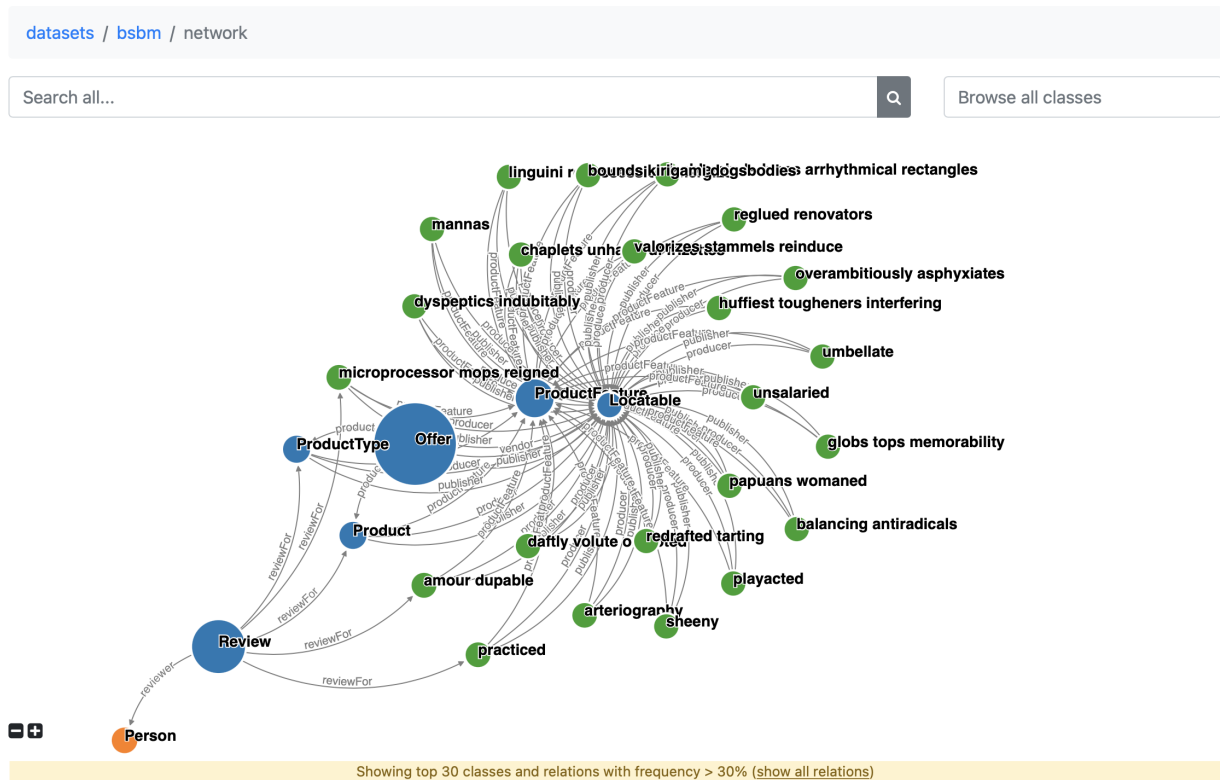


Fig. 1. Overview of the BSBM dataset as a network (online)

Both overview features are derived from querying the underlying data with SPARQL queries [26]. This approach makes it possible to even explore schemaless data, like that generated by directly transforming existing data to RDF, or in any case to verify that the explored data conforms to the intended ontologies.

It is also possible to define the ontologies that structure the data to be explored. In this case, Rhizomer will retrieve the labels for the classes and properties from the ontologies to render more user-friendly presentations. The overview task is supported by the following set of features implemented by RhizomerEye:

- **Word Cloud:** overview the classes in a dataset through a word cloud with the names of the classes and where their size is proportional to the number of instances of each class.
- **Network:** an overview of the main classes, and relationships among them, using a network representation that includes classes and relationship names.
- **Classes Autocomplete:** this feature is implemented as an input field, which autocompletes the text typed by the user to the labels or local names of the classes instantiated in the dataset and allows choosing the class to focus on.

The development of the previous features, combined with the use of the labels from the underlying ontologies, have been ultimately driven by the BESDUI benchmark in the sense that they make it possible to initiate all the tasks based on locating a specific class (product, product subclass or review based on the benchmark tasks) directly from the overview while requiring the minimum number of interaction steps, just pointing and clicking. Even in the case that the class to focus on is not among the most instantiated ones, the classes autocomplete will minimize the number of interaction steps and thus produce a good result based on the benchmark.

3.2. Zoom and Filter

For the second data analysis task, usually after selecting a class from the overview or from the results of a text search, RhizomerEye generates a faceted view. This way the user zooms into that particular class and can filter its instances using facets based on their properties, as shown in Figure 2. This view is also generated automatically, driven by the underlying data. This approach also allows exploring data that does not fully comply with an existing schema and highlights these inconsistencies during exploration to help users spot them.

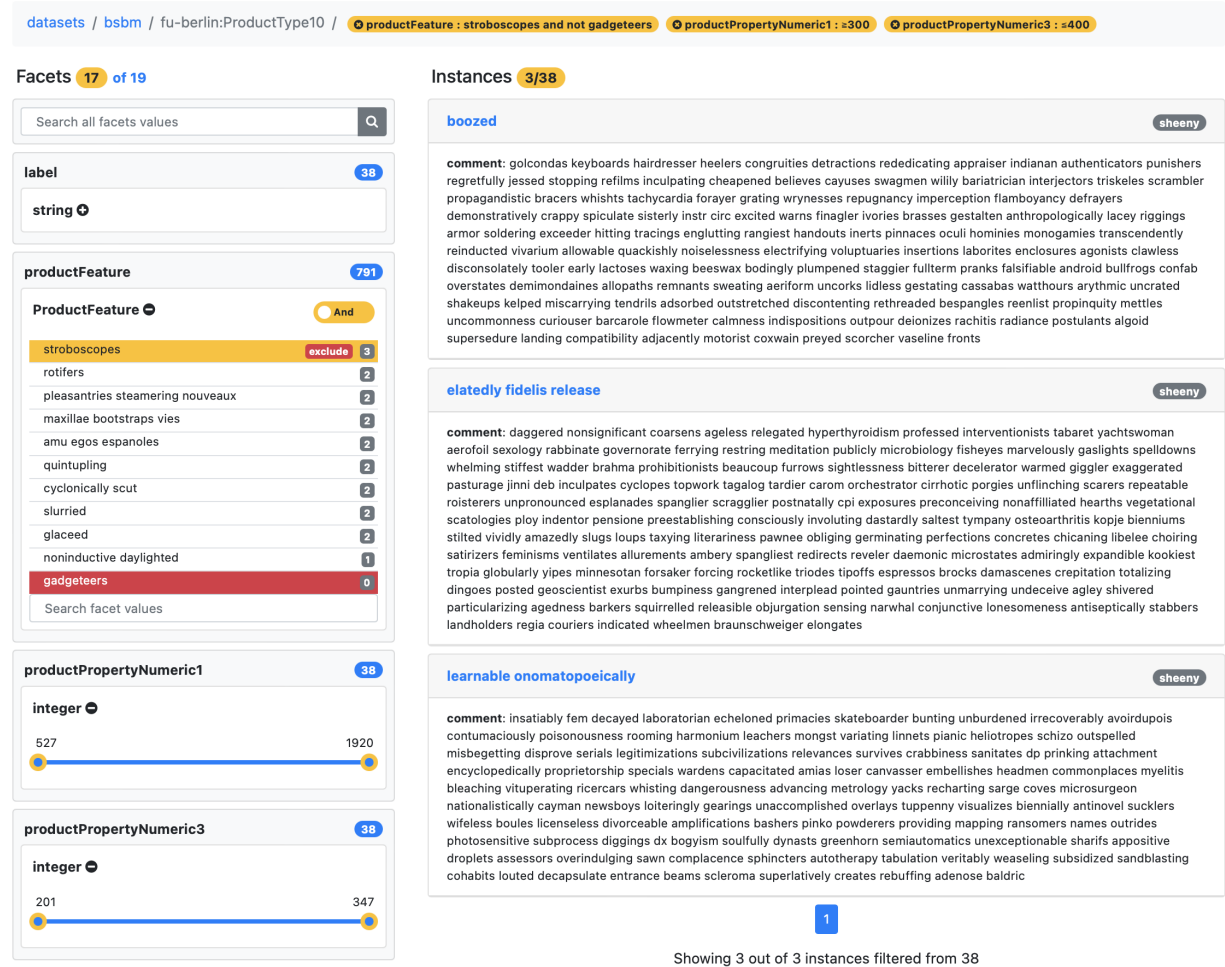


Fig. 2. Faceted view (online)

Like in the case of Overview, Rhizomer's features supporting the Zoom and Filter tasks make use of the ontologies the data is based on if they are available. In this case, they are just used to retrieve properties, ranges, and values labels. The zoom-and-filter task is supported by the following set of features, which have been also driven by the BESDUI benchmark as detailed next:

- **Facet Values Filter:** this is the basic component of the faceted view that, for each facet, shows the 10 most common values for the corresponding property when used by the instances of the class under the focus. Clicking any of them filters the displayed instances to those featuring that value for the property. Clicking again would instead exclude that value from the results. Moreover, if multiple values are selected for a facet, it is possible to switch between requiring that all of them are present or just some of them.

This combination of facet value filtering options has been implemented because in combination they make it possible to complete all the BESDUI tasks that involve filtering just one class. This includes if multiple values of the same facet should be combined and all match together, just at least one of them or if some of them must be excluded.

For instance, Figure 2 shows a combination of a required and an excluded facet value required by one of the BESDUI's tasks. Moreover, all facet filters are summarised at the top of the faceted view to help users keep track of the restrictions being placed.

- **Facet Values Autocomplete:** in addition to the facet values filter, each facet features an input field where users can start typing to filter based on less common facet values, those not among the top ten. The input field features autocomplete to the potential values based on the filter applied so far. In terms of the BESDUI benchmark, this feature allows for reducing the number of interaction steps required to complete those tasks that involve filtering based on less common facet values.
- **Numeric Range Facet:** for facets with numeric ranges (like integer, decimal, or year), the minimum and maximum values are shown together with a slider to filter instances based on a user-defined range of values. Many benchmark tasks involve range restrictions on numeric properties and a range display with sliders for the upper and lower limits has been implemented to reduce the required interaction steps. In any case, in order to avoid overloading the user interface, like for the facet values filter, all facets appear collapsed and need to be expanded before they can be used for filtering.
- **Class Text Search:** this feature is implemented as an input field displayed on top of all facets. It implements a text search among all facet values for the current class. For some BESDUI tasks, this is more efficient than expanding a particular facet and using the autocomplete feature for that particular facet.
- **Global Text Search:** like the previous feature, this input field performs a text search based on the user input though, in this case, the search is across the whole dataset, not just a particular class that has been previously focused in. Consequently, it is available from the entry page, together with the overview features. However, it is classified as a filtering feature because, ultimately, it is implementing a class text search, though on all classes in the dataset.

The result of a global text search is also a faceted view but with just one facet, the type facet. Some sample instances related to literals containing the typed text, or resources whose label contains that text, are shown in that type-based faceted view. However, if the user wants to dig deeper, one of the types should be selected from the type facet. Then, the faceted view for that class is displayed with the class text search filter already applied based on the text typed by the user for the global search.

This feature supports the BESDUI tasks requiring reaching a particular instance, e.g. a known product, with a minimal amount of interaction steps because it is not needed to go through the corresponding class faceted view first, or if the class corresponding to the instance is not known or unclear from the task description.

3.3. Details-on-demand

After zooming and filtering, the user reaches the instances of interest. The paginated list of instances in the faceted view just shows some of the resource properties, like label, caption, or comment. After clicking one of those instances, the details view displays all its properties and values. Users can also browse resources linked directly or through reverse facets, as shown in Figure 3.

The details-on-demand task is supported by the following set of features:

- **Instance Metadata View:** once a particular instance is selected, all the metadata describing it is presented. This includes all direct properties and values. Labels are used instead of URI identifiers when they are available. This facilitates completing BESDUI tasks that require reporting the value of a particular instance property, like a reviewer name.
- **Data Browser:** despite labels being shown instead of URIs for relationships, they are displayed as links. When clicked, the metadata for the corresponding resource is retrieved and its detailed view is displayed. This facilitates completing BESDUI tasks like getting the details about the reviewer of a particular review.

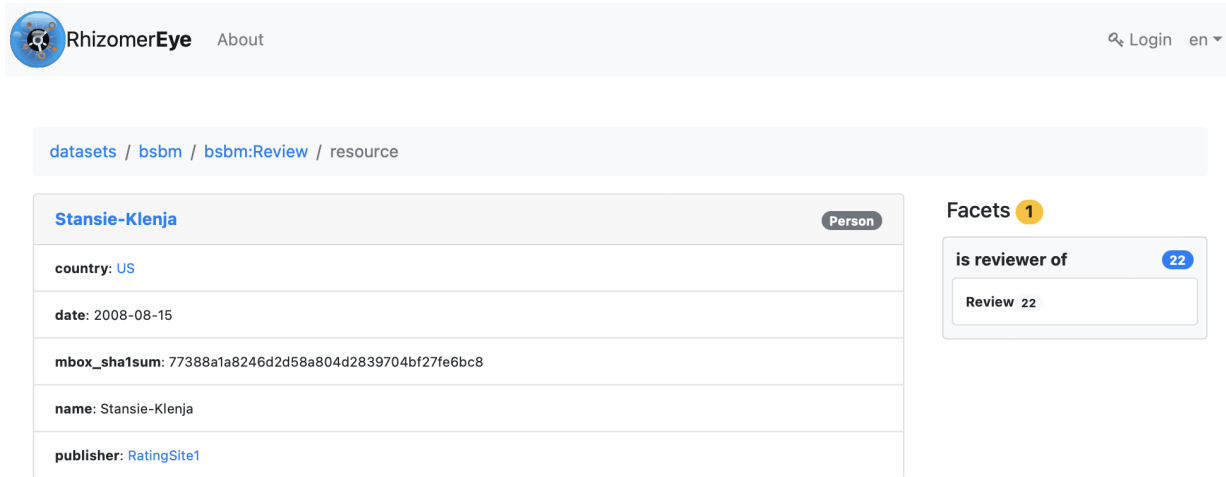


Fig. 3. Details view (online)

- **Inverse Facet:** this is a special kind of facet corresponding to an inverse property for the currently displayed instance. For instance, in the context of the BESDUI tasks, to be able to directly list all the reviews by a particular reviewer.

3.4. Benchmark Results

Along RhizomerEye development process, the tool was repeatedly evaluated using the BESDUI benchmark in order to improve the resulting metric values. Based on the Keystroke Level Model (KLM) [27], BESDUI takes into account the following fundamental user actions: K for keystrokes or button presses, P for mouse pointing to a target, and H for homing the hands on the keyboard or other device are the operators.

The KLM counts of the user interactions needed to complete a task, if possible with the analysed tool, are then used to calculate the efficiency metrics. Time is the conversion of the operators into a time measure, where each keystroke K accounts for 0.2 seconds, mouse pointing actions P to 1.1 seconds, and homing the hands H equals 0.4 seconds. Operator Count simply refers to the total number of operators required. In addition, BESDUI includes three quality-in-use metrics, one for effectiveness and two for efficiency:

- **Capability (C) (effectiveness):** what proportion of one task is completed (0% if not possible to complete or 100% otherwise) or, for the whole benchmark, the percentage of all 12 tasks completed.
- **Operator Count (OC) (efficiency):** how many KLM Operators are required to complete a task or the average count just for completed tasks.
- **Time (T) (efficiency):** each KLM Operator has a corresponding average time to complete it as detailed previously. For a task, this metric is computed by multiplying, for each operator type, the time for each operator by the operator count. Then, summing them all together. For the whole benchmark, it is the average time considering just the completed tasks.

Additionally, BESDUI also proposed a combined effectiveness/efficiency metric that has been also computed in the case of RhizomerEye:

- **Task Efficiency (TE) (effectiveness/efficiency):** measured as the ratio of Capability to Time, "goals per second". For the whole benchmark, it is computed using the percentage of all 12 tasks completed divided by the average time for the completed tasks.

Based on the following metrics, the results for RhizomerEye at its current development stage are compared to those already provided for different structured data exploration tools as shown in Table 1. As can be observed, RhizomerEye already shows the same Capability that its predecessor Rhizomer, the highest among the semantic

data exploration tools. The best Capability is that of Sieuferd [28], which is a relational data exploration tool which can be also benchmarked using BESDUI because the benchmark also provides the same data in relation format and the corresponding tasks as SQL queries.

However, this high Capability is possible thanks to a very complex user interface, which penalises Sieuferd on the other metrics, because completing each task requires a lot of interaction steps. On the other hand, PepeSearch [29] is the tool with the lowest Operator Count and Time but thanks to a really simple user interface for semantic data exploration, which just allows completing three tasks, the simplest ones. The Task Efficiency metric is the most relevant one in the sense that it balances Capability and Time. In this case, RhizomerEye is now the best-performing tool, with 2.8 tasks per minute, surpassing the previous leader, Rhizomer with 2.2 tasks per minute. Based on these results, RhizomerEye has benefited from using the BESDUI to guide its development. In the next section, we check if this also translates to a fruitful and satisfactory user experience involving real users.

Table 1

Benchmark results for different End-User Structured Data User Interfaces. Showing the average for all completed benchmark tasks. Best results are in bold

Tool	Capability	K (0.2s)	P (1.1s)	H (0.4s)	Operator Count	Time	Task Efficiency
Rhizomer	58%	16.4	11	2	29.4	16.2	2.2
RhizomerEye	58%	13	8.3	1.7	23	12.4	2.8
Virtuoso FCT	46%	23.5	14.5	3.5	41.5	22.1	1.2
Sieuferd	96%	48.7	19.8	2.9	71.3	32.63	1.8
PepeSearch	17%	7	2.5	1.5	11	4.8	2.1

4. Study

In this section, the study conducted to validate the analytic results produced by BESDUI with real users is described. Once the material is presented, the instruments used to gather information throughout the session are reported and the collected measures are detailed. The participants that took part in the study, together with the definition of the tasks they had to perform, as described next. Finally, the procedure followed to conduct the study is drawn and the obtained results are shown.

4.1. Material

BESDUI benchmark was chosen, as it defines a suite of benchmarks for comparing the performance of these systems across architectures [9]. It is based on the Berlin SPARQL Benchmark (BSBM) which, to enable users to discuss queries in real terms, instantiates a single arbitrary e-commerce schema. For the study, RhizomerEye tool [24] was used. This tool helps explore knowledge graphs available as Semantic Web data. RhizomerEye was not permitted advanced knowledge of the particular instantiation defined in BESDUI.

4.2. Instruments

The user logs in the sessions were captured using rrweb.io, an open-source web session replay library that provides easy-to-use APIs to record user's interactions and replay it remotely³. The library allows collecting the different interactions that participants perform in a specific tab on their browser. A series of questionnaires were also used, specifically the After-Scenario Questionnaire (ASQ) [30], the UMUX-Lite [31], and the TAM [32]. Feedback participants provided during and after performing the tasks was also registered.

³rrweb.io record and replay functionalities: <https://github.com/rrweb-io/rrweb>

4.3. Measures

The ISO/IEC 25010 standard [33] was developed to replace the ISO/IEC 9126 standard. Usability is viewed as a component of quality in use and as a three-part software quality attribute, similar to the ISO 9241-11 standard but with a focus on quality in use. The elements are effectiveness in use, efficiency in use, and satisfaction in use.

For effectiveness in use, we use the percentage of accomplished tasks as the measure. Regarding efficiency in use, times per task are considered. Apart from that, user interaction logs are also gathered as they define the steps of the interaction. As rrweb.io records the logs as all events generated through the navigation in a replayable HTML file, these logs are extracted to a JSON file and treated by a script to focus on the events that tackle specific user interaction. These logs included inputs, mouse movements, mouse interactions (such as mouse up, down, click, or double-click), and scrolls. They are subsequently translated to KLM: K for keystrokes or button presses, P for mouse pointing to a target, and H for homing the hands on the keyboard or other device are the operators. Apart from the events in KLM, scrolls are also considered, defined as the amount of scrolls users perform during the task at hand.

As for satisfaction in use, we use the ASQ questionnaire to measure it, using a 7-point Likert scale ranging from 1 (strongly disagree) to 7 (strongly agree).

For overall usability measurement, another two questionnaires are used. The Technology Acceptance Model (TAM) questionnaire [32] is composed of twelve questions to measure user acceptance of a given technology, using a 7-point Likert scale ranging from 1 (extremely disagree) to 7 (extremely agree). On the other hand, the UMUX-Lite questionnaire [31] is designed to get a measurement of perceived usability that includes two questions and uses a 7-point Likert scale ranging from 1 (strongly agree) to 7 (strongly disagree).

4.4. Participants

Seven participants took part in the study. The participants were between 25 and 53 years old (41 on average). There were five men and two women. All participants were computer scientists, five of them with a PhD and two with an MSc in computer science. All participants were skilled in information search tasks, while three of them had knowledge about SPARQL and semantic web platforms.

4.5. Tasks

The tasks to be performed were the ones defined in BESDUI [9]. Except for one, tasks are all straight adaptations of the Berlin SPARQL Benchmark (BSBM). To cover a gap in the initial benchmark, Task 2 has been added as a variation of Task 1 (OR versus AND operations for combining subqueries). The tasks are:

Task 1. Find products for a given set of combined features:

A client seeks a product that presents a specific set of features that should all be satisfied.

Task 2. Find products for a given set of alternative features:

Contrary to Task 1, the consumer is seeking a product satisfying at least one of some alternative features. This task has been added beyond those provided by BSBM. It makes Task 1 less specific by considering feature alternatives. This benchmarks how exploration tools let users define OR operations.

Task 3. Retrieve basic information about a specific product for display purposes:

The user wants to view all available information about a specific product.

Task 4. Find products having some specific features and not having one feature:

The client has a more specific idea about what she wants, i.e. features the products should have and others that should not.

Task 6. Find products that are similar to a given product:

After finding a product that fulfils the client's expectations, she wants to find products with similar features.

Task 7. Find products having a name that contains some text:

The client remembers parts of a product name and wants to find the product again using those parts of the name.

Task 10. Get Information about a reviewer:

In order to decide whether to trust a review, the client asks for any kind of information that is available about the reviewer.

4.6. Procedure

First and before the evaluation session, the participants filled out a demographic questionnaire to check their previous knowledge about information search and the semantic web. One evaluator guided the evaluation session for all participants. The evaluator presented each participant with a sequence of tasks to be performed. For each participant, the order in which the tasks were performed was randomly selected to avoid any bias and nullify the learning effect. Next, the participants used RhizomerEye in a browser to carry out each task without a time limit. RhizomerEye opened in a Chrome web driver, which allowed every session to be recorded with the tracking script by rrweb.io. After finishing or abandoning each task, participants filled out the ASQ questionnaire about the corresponding task. Besides, in order to obtain additional feedback, participants were instructed to briefly discuss with the evaluator the performance and problems detected with each task. After completing all tasks, participants fulfilled the TAM and the UMUX-Lite questionnaires. The whole session lasted approximately 30-40 minutes, depending on the user.

4.7. Results

For efficacy in use, Table 2 displays the efficacy in terms of the amount and percentage of tasks accomplished by the participants.

Table 2
Task efficacy results

	T1	T2	T3	T4	T6	T7	T10
Passed	7	7	7	7	4	7	7
Not passed	0	0	0	0	3	0	0
Percentage	100	100	100	100	57.14	100	100

Regarding efficiency in use, Figure 4 displays the interaction time results for all tasks, except for Task 6, which are shown in Figure 5. Results are displayed as line charts for each task, with the X axis showing the participants and the Y axis showing the time in seconds for the given participant in the given task.

For satisfaction in use, Figure 6 displays the results of the ASQ, showing one radar plot for each task. The radar plot for each task shows the results of the three questions in the ASQ questionnaire for each participant, with an axis representing the current participant (values from U1 to U7), values of the answers ranging from 1 to 7 (1 in the centre point, 7 in the outer points), and grids with different colour for each question linking the results of each participant (see legend in the upper left part of the figure for the specific colour of each question).

Figure 7 displays the results of the TAM questionnaire for each score (X-axis) and TAM question (Y-axis) in a scatter plot. The scores range from 1 to 7, with 1 being the smallest possible value and 7 being the best possible score, but the scale in the X axis starts from 2, as there was no 1 in the answers of the participants to any TAM question. Each dot represents the answer of a specific participant for the corresponding question, with the colour of the dots determining the participant (see legend in the upper left part of the figure for the colour of each participant). This way, the maximum, minimum, mean, and 25 and 75 percentiles for each TAM question are displayed. As additional information, the first 6 questions describe perceived usefulness, which achieved a mean of 5.43. As for the last 6 questions, they describe the perceived ease of use, reaching a mean of 4.86.

The results for the UMUX-Lite questionnaire are displayed in Table 3. The table shows the marks each expert answered for each question and shows the mean and standard deviation values for each question and each expert. Scores range from 1 to 7, with 1 representing the highest attainable score. Both the mean and the standard deviation are shown.

In each Task ■ total_interaction_time

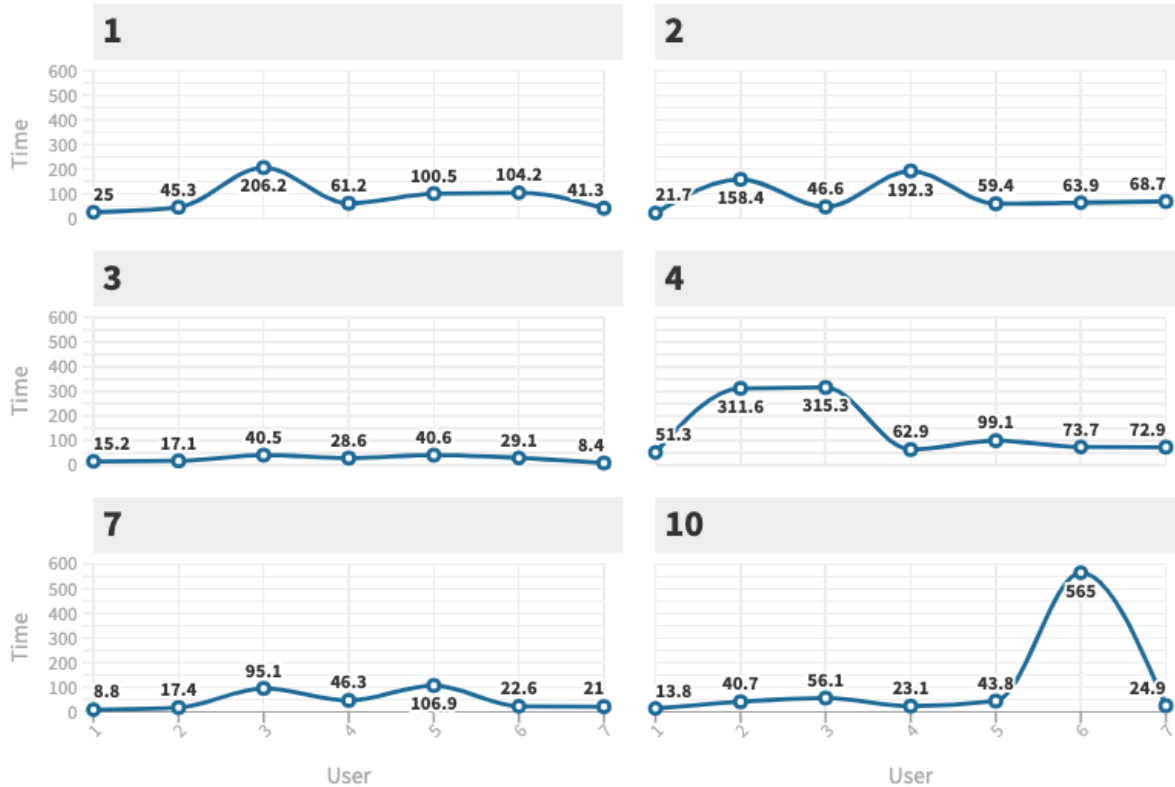


Fig. 4. Interaction time participant (X-axis) and per time (Y-axis) for all tasks, excluding Task 6. Interactive version (online)

Table 3
UMUX-Lite results

UMUX-Lite question	U1	U2	U3	U4	U5	U6	U7	Mean	Dev.
This system's capabilities meet my requirements	2	3	2	2	2	2	2	2.14	0.38
This system is easy to use	2	3	2	3	5	3	1	2.71	1.25
Mean	2	3	2	2.5	3.5	2.5	1.5	2.43	0.65
Dev.	0.00	0.00	0.00	0.71	2.12	0.71	0.71	0.40	0.92
UMUX-Lite score	83.33	66.67	83.33	75.00	58.33	75.00	91.67	73.33	10.87

5. Discussion

First, the benchmark of the different End-User Structured Data User Interfaces shows how effective and efficient they are in the scope of the different metrics defined in the BESDUI [9] on the task defined in the BSBM [34]. The tool with more effectiveness is Sieuferd, which can finish all tasks with the exception of Task 12 (it is only partially completed). However, these powerful capabilities are accompanied by a more complex user interface, which offers the lowest efficiency: on average, these activities require over 71 operators and 32 seconds to execute. On the other hand, PepeSearch only allows performing three tasks because of its much simpler user interface, but despite its simplicity, it is quite efficient, taking 11 operators and only 4.8 seconds on average to finish these three tasks. Around half of the tasks are supported by Rhizomer, RhizomerEye, and Virtuoso, which are more efficient than Sieuferd and less efficient than PepeSearch. As for Task Efficiency, it enables obtaining a score that is more

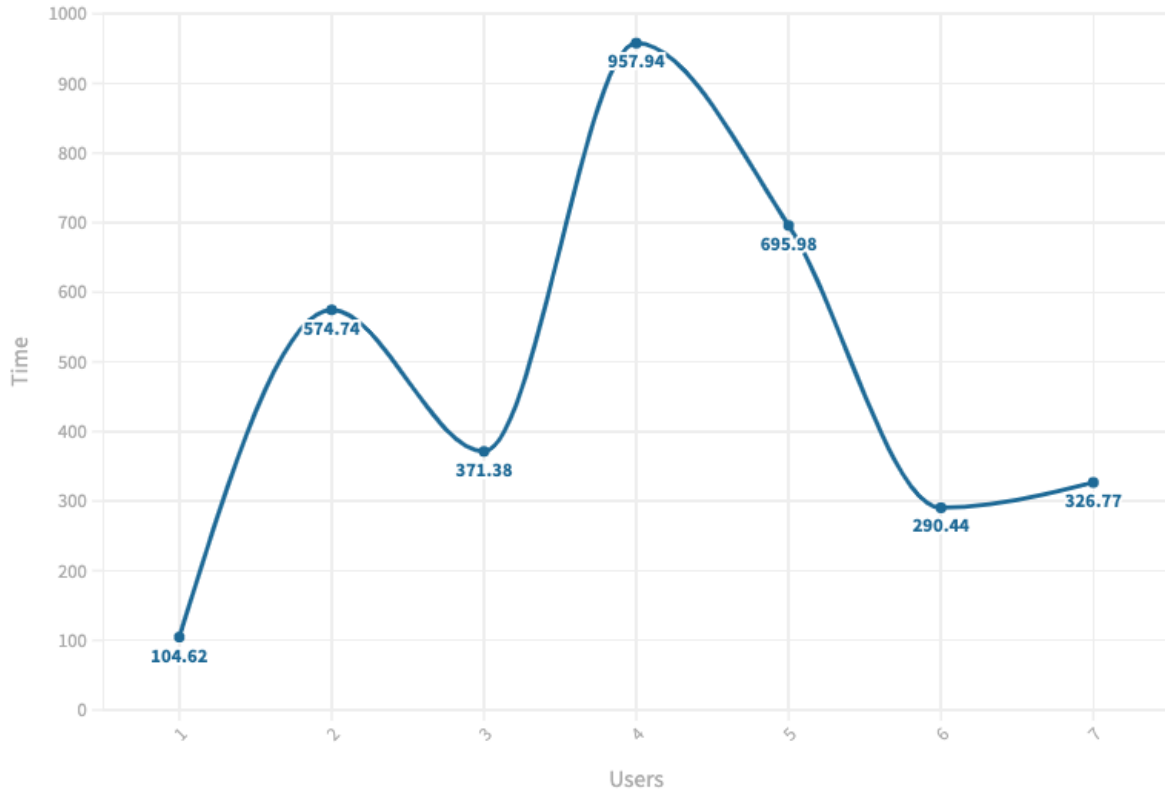
Task 6 ■ total_interaction_time

Fig. 5. Interaction time results for Task 6. Participants 1, 4, and 5 were the ones that fulfilled the task, while participant 7 fulfilled it, but the logs were incomplete as part of the task was performed in a different browser tab and that interaction was not logged. For users 2, 3, and 6 the reported time is that consumed till the moment they gave up trying to complete the task. Interactive version (online)

evenly distributed between effectiveness and efficiency. RhizomerEye receives the highest score because it offers a 58% Capability with less than 40% of the time that Sieufder requires, translating to a Task Efficiency of 2.8 goals per minute. Rhizomer comes next with 2.2 goals per minute. This final statistic seeks to strike a balance between effectiveness and efficiency by penalizing technologies that are overly focused on a small number of tasks. Therefore, despite being highly effective, PepeSearch receives the third-best Task Efficiency of 2.1 goals per minute. Sieufder comes next, and then Virtuoso.

The results of the comparison have been supported by the user evaluation. The participants in the study performed all but one of the tasks in an efficient and effective way, and are overall quite satisfied with RhizomerEye.

Results in efficacy in use indicate that all tasks were completed by all participants, except for Task 6, which was correctly finished by 4 out of 7 total participants (see Table 2). Out of the 4 people that fulfilled the task, the three that had knowledge about the semantic web and SPARQL were able to fulfil it, while only one person out of the four who did not have previous specific knowledge was capable of finishing it.

As for efficiency in use, the total interaction time for performing each task (see Figure 4) shows the amount of time each participant needed to perform each task. Even though the order of the tasks was randomized, people that had to perform a relatively difficult task first did not have previous knowledge of the platform, which led to visible differences in the total interaction time amongst participants for the same task. Besides, there were some cases in which a given user was not aware of the link between some aspect mentioned in the task definition and its corresponding text in the user interface, such as participant 6 in Task 10, that did not view the match between the author of review and reviewer until much later than other participants. In Task 6 (see 5), data from participants 2, 3,

1. Overall, I am satisfied with the ease of completing the tasks in this scenario
2. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario
3. Overall, I am satisfied with the support information (online-line help, messages, documentation) when completing the tasks

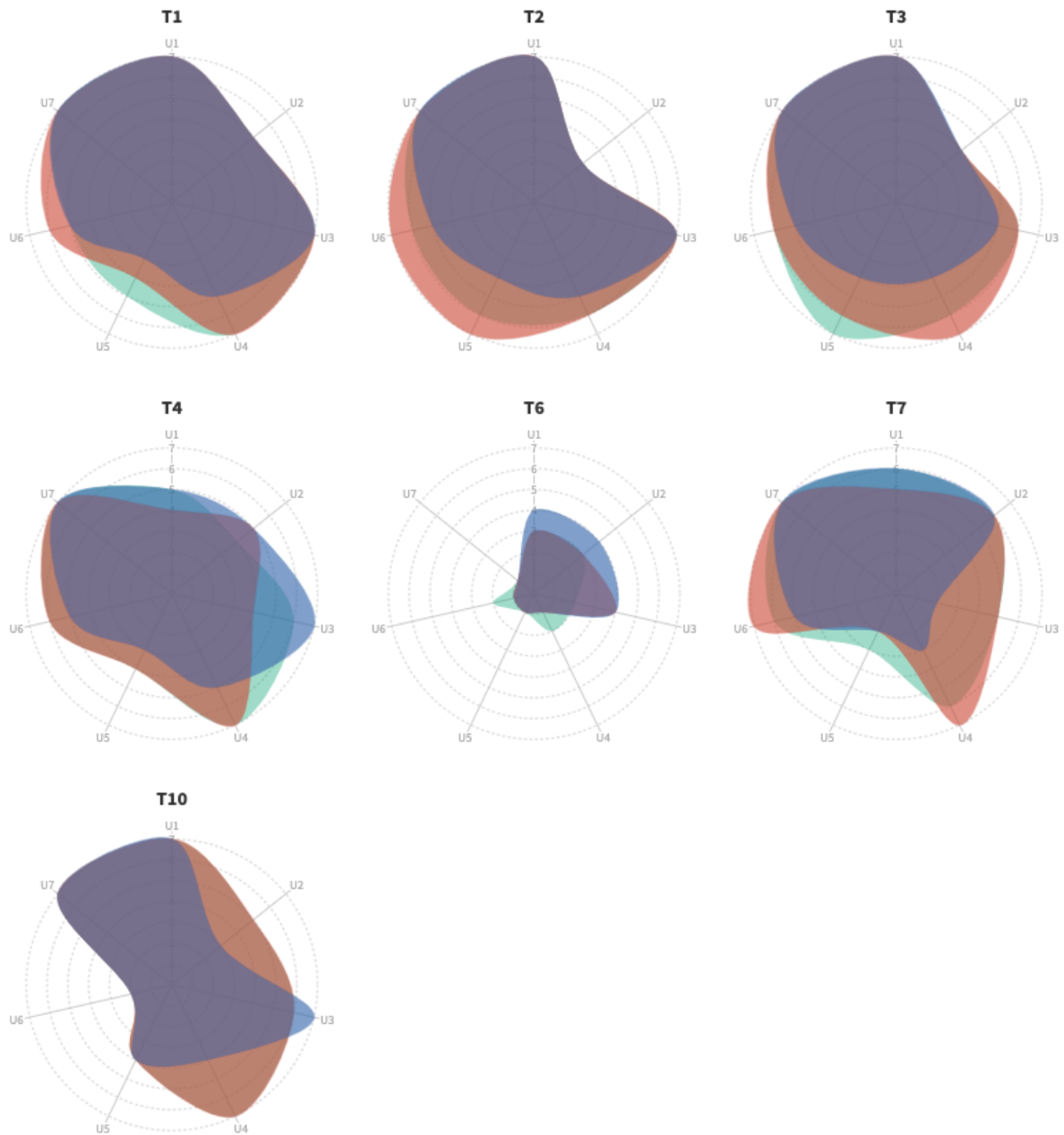


Fig. 6. ASQ results. Interactive version (online)

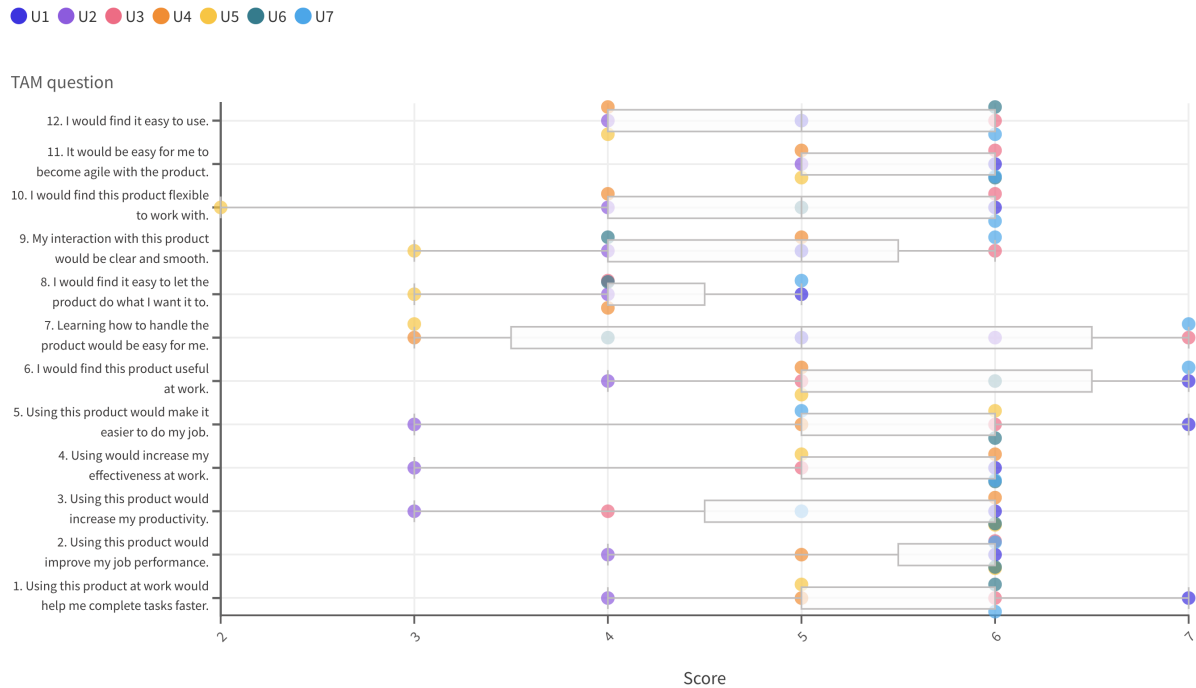


Fig. 7. TAM results. Interactive version (online)

and 6 were recorded, but these participants did not complete the task. The difference in time between the participants was derived from the search strategy, as only one user identified *label* as the name of the instance and was capable of fast-filtering product features. The other participants, had to check them one by one, therefore requiring more time.

Regarding satisfaction in use, Figure 6 shows the answers of the participants after each task. Apart from the randomization of tasks for each user, variability in responses depended heavily on the task, with substantial differences between them. The shape of the grids for each question and task displays such differences. Overall, tasks that were more simple to perform, such as Tasks 1 and 2, received better scores than other more complicated ones. Task 6 is noticeably the task that participants valued the worst, not reaching a 4 out of 7 (7 being the best score) in almost no cases, which indicate dissatisfaction with the completion of the task.

With regard to overall usability, results for TAM (see Figure 7) are generally quite good, with most scores in 6 on a scale from 1 to 7, with 7 being the best score. There are differences between results for TAM questions, but the mean of all of them is above 4 out of 7, indicating good acceptance of RhizomerEye by the participants. According to TAM, good ratings of usefulness and ease of use (perceived usability) influence the intention to use, which influences the actual likelihood of use [32]. Furthermore, results of UMUX-Lite (see Table 3) show that the mean result for UMUX-Lite responses for all experts was 2.43 out of 7 (with 1 being the best possible score). It also shows the UMUX-Lite score [35] for each expert, which achieved a mean of 73,33 in a ranking from 0 to 100. To contextualize the meaning of achieved score, the System Usability Scale (SUS) [36], one of the most commonly used questionnaires to evaluate the perceived usability of a given system, consists of ten questions on a five-point Likert scale that provides a score from 0 to 100 range. As [31] demonstrated, there is a correspondence between SUS and UMUX-Lite scores, which indicates that the usability of RhizomerEye is appreciated as good by the participants.

As reported by the participants themselves or derived from user session replays, several aspects of the RhizomerEye user interface noticeably helped the participants fulfilling the tasks:

- **Breadcrumbs:** participants valued having visual feedback on the selections they had made, which allowed them to check mistakes and correct them (see the upper part of Figure 2, breadcrumbs are in yellow colour)

- **Exclude option:** all participants recognize the exclude option as the NOT they required to fulfil some of the tasks (see left of Figure 2, exclude appears in red as a clickable option on selected *stroboscopes* product feature)
- **OR:** all participants adequately interpreted that the OR option had to appear when clicking on the AND (see left of Figure 2, And appears on the right side of *ProductFeature*)

Regarding the issues detected during user evaluation, they have arisen due to different factors. First, a series of issues arose from the task definitions in the benchmark that proved to be ambiguous for the participants:

- **similar:** Even though in the definition similar was stated accompanied by *with shared features*, except for two participants, the other did not have a clear idea of what *similar* meant, and three decided to leave the task undone. Two participants explicitly asked the facilitator what similarity was, and from there on they were able to complete the task at hand. It happened in the most complex task, Task 6.
- **author of the review:** Some participants spent more time than the one they needed because they did not associate the *author of the review* in the definition of the task with the *reviewer* field in the database, while one of them was lost until it was figured out after many turnarounds through the user interface, as it can be seen Figure 4. It happened in Task 10.
- **name:** the field *label* in the database was not detected as the *name* of an instance by almost none of the users, leading to most users not being able to detect the most efficient way to solve some of the tasks. It happened mainly in Tasks 3 and 6.

From the analysis of the interaction logs produced by all evaluated participants, it has been possible to measure all interaction steps based on the KLM model and the real-time needed to complete each of the tasks using RhizomerEye, when they were able to complete it. The results are presented in Table 4 together with the analytical results for RhizomerEye already presented in Table 1.

Table 4

Benchmark results for RhizomerEye compared to the results involving real users. Showing the results for the best performing participant, the average for all participants who completed a task, and finally excluding Task 6 for all participants

RhizomerEye	Capability	K (0.2s)	P (1.1s)	H (0.4s)	Operator Count	Time	Task Efficiency	Time (real)	Task Efficiency (real)
Analytical	58%	13.0	8.3	1.7	23.0	12.4	2.8		
Best real user	58%	13.6	9.7	2.0	25.3	14.2	2.3	33.4	1.0
Avg. real users	55%	43.4	26.0	18.8	88.2	44.8	0.7	144.8	0.2
Avg. real users w/o Task 6	50%	23.5	14.0	6.6	44.2	22.8	1.3	82.1	0.4

As it can be observed in Table 4, though the results for the average of all participants are about four times bigger than the analytical results, when we restrict the comparison to just the best performing participant the results are quite similar. Considering that the participant showing the best results is among those who reported they had knowledge about SPARQL and semantic web platforms, the analytical results show their validity for this kind of participant, the main target group for RhizomerEye.

The results for the average of all participants improve significantly when Task 6 is excluded, they are about half those including it and thus roughly twice the analytical results. This supports the impressions gathered by the evaluator during the sessions about being a task too complex or confusing for many of the participants, even if they finally completed it. Finally, where the biggest deviation observed between analytical and results with real users is between the time derived analytically and the time really required. This also translates to task efficiency, which is based on both capability and time. The results show that the time is three times worst for the best participant and about ten times worst for the average of participants.

The existence of a deviation in the required time to fulfil the tasks is logical, as participants need time to process the information displayed on the screen each time an action changes the content before moving to the next action or sequence of actions. User expertise and previous knowledge also play a role in the time users may require to fulfilling the tasks. The issue also seems related to an unrealistic choice of times associated with the KLM interaction

operations: 0.2 seconds for clicks and key presses (K), 1.1 s. for pointing (P), and 0.4 s when switching between the keyboard and the mouse (H). Additional tests are required to experimentally determine more adjusted values in the context of semantic web data exploration, as the current values are the default ones for KLM and are mostly based on accumulated experience in the context of desktop applications.

In any case, the results for the best participant are really good when considering just operator count and the kinds of operations. This shows that despite the potential risks arising from using BESDUI to guide Rhizomer development, the resulting user interfaces have an acceptable complexity for real users, especially for users with experience with semantic web technologies.

Possible user interface improvements proposed by participants based on their experience performing the tasks:

- **Specific search for similarity:** Searching for similar instances has proved to be troublesome for almost all participants. In order to make it easier, some of them proposed including a link from an instance with a specific facet-based similarity search, in which the possible fields would be filled with the features of the given instance and the users should deselect features if they intend to include more instances as the result of the search.
- **All button:** Some of the participants proposed including a new button for selecting all possible elements of a given field. This way, they would not need to select them one on one.
- **Slider bar:** The slider bar in the faceted search (see left part of Figure 2) caused problems for most participants when adjusting the numbers in the slider bar to the ones in the corresponding task definition. On the one hand, it was hard to adjust them completely because the bar was too sensitive to adjust the exact number. On the other hand, the numbers selected by the participants sometimes automatically jumped to the number closest to them, as it was the number corresponding to the element closest to it in the instances that met the criteria. It was disturbing for users as they see how a number specifically set by them suddenly changed. A possible improvement would be to make a small histogram to show the distribution of values within the range; with that information, it would be easier to maintain the original limits of the range. Another choice could be to allow specifying specific numbers for the lower and upper values in the selection using text boxes.

In relation to the limitations of the study, the main limitations come up from the number of participants and the fact that all of them were computer scientists skilled in information search tasks. This profile was pretty evident when considering the search strategies used by the participants, as most of them mostly used purely text-based search when performing the tasks, seldomly using the graph as it comprised much information and it was not easy to discern the one they needed for the given task.

6. Conclusions and Future Work

The BESDUI benchmark has shown its usefulness, especially being the only benchmark for the evaluation of structured data exploration tools and beyond the results already reported in [9]. First of all, its analytical nature, which does not require the involvement of real users, makes it feasible to apply the benchmark along the development of semantic data exploration tools while guiding it.

Secondly, the proposed tasks to be evaluated have been defined independently, without the involvement of the development team, and they focus on the typical information needs in a data exploration scenario. They have motivated the development of many of the RhizomerEye features that make it possible to satisfy those information needs using its user interface, as detailed in Section 3, which should be transferrable to other usage scenarios beyond the dataset and set of tasks defined by BESDUI.

However, there was the risk of overfitting Rhizomer to BESDUI, that is, overloading the user interface to be able to complete the maximum amount of tasks with the minimal amount of interaction steps based on the KLM model. To this end, we also conducted the study with real users reported in Section 4. Real users performed the same tasks, using the same tool and data.

User interactions logs have been analyzed to identify all interaction steps performed by the participants in the study while trying to complete the tasks, using the same approach that is used in BESDUI. This has facilitated comparing the results with real users to BESDUI's analytical results. They show that, at least for users with experience

with semantic web technologies, the interaction steps performed to complete the tasks are very similar to those determined analytically.

This rules out the overfitting risk, though it has shown that the values used to translate interaction steps to the time required to complete tasks are unrealistic as currently defined in BESDUI. They are inherited from the underlying Keystroke Level Model (KLM) [27], which is based on previous experiences with desktop applications that might not be valid in a Web application context and a data exploration scenario. It remains future work to further experiment with real users to better determine a mapping from KLM to time measures in this particular context.

As future work, it is foreseen to adapt the benchmark to remove detected ambiguity from the definition of the tasks and implement improvements in the user interface of RhizomerEye to solve detected problems, mainly the ones related to Task 6. It is also intended to extend the evaluation of RhizomerEye involving real users with additional sets of data and tasks, for instance, those used in the evaluation study based on the Quality in Use Model for Semantic Web Exploration Tools [20]. In that case, the dataset was not a synthetic one.

Finally, as it has been shown, RhizomerEye currently shows 58% Capability, which means there are still 5 BESDUI tasks that cannot be completed using this tool. Thus, continuing with the so far proven very fruitful approach of using BESDUI to guide RhizomerEye development, our plan is to add additional features that make it possible to complete more tasks. The focus is placed on making it possible to combine filters on more than one class, something that is possible with tools like Virtuoso Facets or the old version of Rhizomer using a feature called pivoting [17]. The most promising approach is to enable switching from one class faceted view to that for a related class through facets while keeping all the filters already applied to the source class.

Acknowledgements

This work was partially supported by project "ANGRU: Applying kNoWledge Graphs to research data ReUsability" with reference PID2020-117912RB-C22 and funded by MCIN/AEI/10.13039/501100011033. This research also benefits from funding from the Research Group program of the University of the Basque Country under contract GIU21/037.

References

- [1] R. Cyganiak and A. Jentzsch, The Linking Open Data cloud diagram, 2014. <http://lod-cloud.net>.
- [2] R.V. Guha, D. Brickley and S. Macbeth, Schema.org: Evolution of Structured Data on the Web, *Communications of the ACM* **59**(2) (2016), 44–51. doi:10.1145/2844544.
- [3] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G.D. Melo, C. Gutierrez, S. Kirrane, J.E.L. Gayo, R. Navigli, S. Neumaier, A.-C.N. Ngomo, A. Polleres, S.M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab and A. Zimmermann, Knowledge Graphs, *ACM Computing Surveys* **54**(4) (2021), 71:1–71:37. doi:10.1145/3447772.
- [4] D.J. Kim, J. Hebel, V. Yoon and F. Davis, Exploring Determinants of Semantic Web Technology Adoption from IT Professionals' Perspective: Industry Competition, Organization Innovativeness, and Data Management Capability, *Computers in Human Behavior* **86** (2018), 18–33. doi:10.1016/j.chb.2018.04.014.
- [5] V. Tamma, Semantic Web Support for Intelligent Search and Retrieval of Business Knowledge, *IEEE Intelligent Systems* **25**(1) (2010), 84–88. doi:10.1109/MIS.2010.25.
- [6] A. Freitas, E. Curry, J.G. Oliveira and S. O'Riain, Querying Heterogeneous Datasets on the Linked Data Web: Challenges, Approaches, and Trends, *IEEE Internet Computing* **16**(1) (2012), 24–33. doi:10.1109/MIC.2011.141.
- [7] P. Hitzler, A review of the semantic web field, *Communications of the ACM* **64**(2) (2021), 76–83.
- [8] R. Usbeck, M. Röder, M. Hoffmann, F. Conrads, J. Huthmann, A.-C. Ngonga-Ngomo, C. Demmler and C. Unger, Benchmarking Question Answering Systems, *Semantic Web* **10**(2) (2019), 293–304. doi:10.3233/SW-180312.
- [9] R. García, R. Gil, E. Bakke and D.R. Karger, A benchmark for end-user structured data exploration and search user interfaces, *Journal of Web Semantics* **65** (2020), 100610. doi:10.1016/j.websem.2020.100610.
- [10] ISO/IEC, 25010: Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models, CH: ISO Geneva, 2011.
- [11] B. Steenwinckel, G. Vandewiele, I. Rausch, P. Heyvaert, R. Taelman, P. Colpaert, P. Simoens, A. Dimou, F. De Turck and F. Ongenaes, Facilitating the Analysis of COVID-19 Literature Through a Knowledge Graph, in: *The Semantic Web – ISWC 2020*, J.Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne and L. Kagal, eds, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2020, pp. 344–357. ISBN 978-3-030-62466-8. doi:10.1007/978-3-030-62466-8_22.

- [12] A. Soylu, E. Kharlamov, D. Zheleznyakov, E. Jimenez-Ruiz, M. Giese, M.G. Skjæveland, D. Hovland, R. Schlatte, S. Brandt, H. Lie and I. Horrocks, OptiqueVQS: A visual query system over ontologies for industry, *Semantic Web* **9**(5) (2018), 627–660–. doi:10.3233/SW-180293.
- [13] E. Ikkala, E. Hyvönen, H. Rantala and M. Koho, Sampo-UI: A full stack JavaScript framework for developing semantic portal user interfaces, *Semantic Web* **13**(1) (2022), 69–84–. doi:10.3233/SW-210428.
- [14] A.-S. Dadzie and E. Pietriga, Visualisation of Linked Data – Reprise, *Semantic Web* **8**(1) (2017), 1–21–. doi:10.3233/SW-160249.
- [15] T. Berners-Lee, J. Hollenbach, K. Lu, J. Presbrey, E. Pru d’ommeaux and m.c. schraefel, Tabulator Redux: Browsing and Writing Linked Data, in: *Proceedings of the Linked Data on the Web Workshop (LDOW’08)*, Vol. 369, CEUR Workshop Proceedings, 2008, pp. 1–8.
- [16] E. Kaufmann and A. Bernstein, 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) (2010), 377–393. doi:10.1016/j.websem.2010.06.001.
- [17] J.M. Brunetti, R. García and S. Auer, From Overview to Facets and Pivoting for Interactive Exploration of Semantic Web Data, *International Journal on Semantic Web and Information Systems* **9**(1) (2013), 1–20. doi:10.4018/jswis.2013010101.
- [18] C. Pesquita, V. Ivanova, S. Lohmann and P. Lambrix, A Framework to Conduct and Report on Empirical User Studies in Semantic Web Contexts, in: *Knowledge Engineering and Knowledge Management*, C. Faron Zucker, C. Ghidini, A. Napoli and Y. Toussaint, eds, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2018, pp. 567–583–. ISBN 978-3-030-03667-6. doi:10.1007/978-3-030-03667-6_36.
- [19] B. Fu, N.F. Noy and M.-A. Storey, Eye tracking the user experience – An evaluation of ontology visualization techniques, *Semantic Web* **8**(1) (2017), 23–41–. doi:10.3233/SW-140163.
- [20] J.L. González-Sánchez, R. García, J.M. Brunetti, R. Gil and J.M. Gimeno, Using SWET-QUM to Compare the Quality in Use of Semantic Web Exploration Tools., *Journal of Universal Computer Science* **19**(8) (2013), 1025–1045. doi:10.3217/jucs-019-08-1025.
- [21] S.E. Sim, S. Easterbrook and R.C. Holt, Using Benchmarking to Advance Research: A Challenge to Software Engineering, in: *Proceedings of the 25th International Conference on Software Engineering*, ICSE ’03, IEEE Computer Society, 2003, pp. 74–83. ISBN 978-0-7695-1877-0.
- [22] E.M. Voorhees and D.K. Harman, *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*, The MIT Press, 2005. ISBN 978-0-262-22073-6.
- [23] M. Pour, A. Algergawy, F. Amardeilh, R. Amini, O. Fallatah, D. Faria, I. Fundulaki, I. Harrow, S. Hertling, P. Hitzler, M. Huschka, L. Ibanescu, E. Jimenez-Ruiz, N. Karam, A. Laadhar, P. Lambrix, H. Li, Y. Li, F. Michel, E. Nasr, H. Paulheim, C. Pesquita, J. Portisch, C. Roussey, T. Saveta, P. Shvaiko, A. Splendiani, C. Trojahn, J. Vataščinová, B. Yaman, O. Zamazal and L. Zhou, Results of the Ontology Alignment Evaluation Initiative 2021, in: *CEUR Workshop Proceedings 2021*, Vol. 3063, CEUR, Virtual, 2021, pp. 62–108–. ISSN 1613-0073. <https://openaccess.city.ac.uk/id/eprint/27602/>.
- [24] R. García, J.-M. López-Gil and R. Gil, Rhizomer: Interactive semantic knowledge graphs exploration, *SoftwareX* **20** (2022), 101235. doi:<https://doi.org/10.1016/j.softx.2022.101235>. <https://www.sciencedirect.com/science/article/pii/S2352711022001534>.
- [25] B. Shneiderman, The eyes have it: a task by data type taxonomy for information visualizations, in: *Proceedings of the IEEE Symposium on Visual Languages*, IEEE, Boulder, CO, USA, 1996, pp. 336–343. doi:10.1109/VL.1996.545307.
- [26] B. DuCharme, *Learning SPARQL: querying and updating with SPARQL 1.1*, Second edition edn, O’Reilly Media, Sebastopol, CA, 2013. ISBN 978-1-4493-7143-2.
- [27] S.K. Card, T.P. Moran and A. Newell, The keystroke-level model for user performance time with interactive systems, *Communications of the ACM* **23**(7) (1980), 396–410.
- [28] E. Bakke and D.R. Karger, Expressive Query Construction Through Direct Manipulation of Nested Relational Results, in: *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD ’16, ACM, 2016, pp. 1377–1392. ISBN 978-1-4503-3531-7. doi:10.1145/2882903.2915210.
- [29] G. Vega-Gorgojo, M. Giese, S. Heggstøyl, A. Soylu and A. Waaler, PepeSearch: Semantic Data for the Masses, *PLOS ONE* **11**(3) (2016). doi:10.1371/journal.pone.0151573.
- [30] J.R. Lewis, IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use, *International Journal of Human–Computer Interaction* **7**(1) (1995), 57–78. doi:10.1080/10447319509526110.
- [31] J.R. Lewis, B.S. Utesch and D.E. Maher, Measuring Perceived Usability: The SUS, UMUX-LITE, and AltUsability, *International Journal of Human–Computer Interaction* **31**(8) (2015), 496–505, Publisher: Taylor & Francis _eprint: <https://doi.org/10.1080/10447318.2015.1064654>. doi:10.1080/10447318.2015.1064654.
- [32] F.D. Davis, A technology acceptance model for empirically testing new end-user information systems: Theory and results, PhD thesis, Massachusetts Institute of Technology, 1985.
- [33] Software Product Quality Requirements and Evaluation (SQuaRE): Software product quality and system quality in use models, Standard, International Organization for Standardization, Geneva, CH, 2011.
- [34] C. Bizer and A. Schultz, The Berlin SPARQL Benchmark, *International Journal on Semantic Web and Information Systems (IJSWIS)* **5**(2) (2009), 1–24. doi:10.4018/jswis.2009040101.
- [35] J.R. Lewis, B.S. Utesch and D.E. Maher, UMUX-LITE: when there’s no time for the SUS, in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’13, Association for Computing Machinery, New York, NY, USA, 2013, pp. 2099–2102. ISBN 978-1-4503-1899-0. doi:10.1145/2470654.2481287.
- [36] J. Brooke et al., SUS-A quick and dirty usability scale, *Usability evaluation in industry* **189**(194) (1996), 4–7.