ForestQB: Enhancing Linked Data Exploration through Graphical and Conversational UIs Integration

OMAR MUSSA, Cardiff University, UK and Saudi Electronic University, Saudi Arabia
OMER RANA, Cardiff University, UK
BENOÎT GOOSSENS, Cardiff University, UK
PABLO OROZCO-TERWENGEL, Cardiff University, UK
CHARITH PERERA, Cardiff University, UK

This paper introduces the Forest Query Builder (ForestQB), an innovative toolkit designed to enhance the exploration and application of observational Linked Data (LD) within the field of wildlife research and conservation. Addressing the challenges faced by non-experts in navigating Resource Description Framework (RDF) triplestores and executing SPARQL queries, ForestQB employs a novel integrated approach. This approach combines a graphical user interface (GUI) with a conversational user interface (CUI), thereby greatly simplifying the process of query formulation and making observational LD accessible to users without expertise in RDF or SPARQL. Developed through insights derived from a comprehensive ethnographic study involving wildlife researchers, ForestQB is specifically designed to improve the accessibility of SPARQL endpoints and facilitate the exploration of observational LD in wildlife research contexts. To evaluate the effectiveness of our approach, we conducted a user experiment. The results of this evaluation affirm that ForestQB is not only efficient and user-friendly but also plays a crucial role in eliminating barriers for users, facilitating the effective use of observational LD in wildlife conservation and extending its benefits to wider domains. (GitHub Link)

CCS Concepts: • **Human-centered computing → User interface toolkits**; • **Information systems → Search interfaces**; *Resource Description Framework (RDF)*; **Information extraction**.

Additional Key Words and Phrases: Linked Data, SPARQL, RDF, Query Builder, Visual Querying

ACM Reference Format:
Omar Mussa, Omer Rana, Benoît Goossens, Pablo Orozco-terWengel, and Charith Perera. 2024. ForestQB: Enhancing Linked Data Exploration through Graphical and Conversational UIs Integration. *ACM J. Comput. Sustain. Soc.* , 1 (June 2024), 33 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Since the advent of the semantic web and linked data (LD) initiatives, there has been a significant expansion of the semantic web community. The use of semantic web technologies offers numerous benefits, including the integration of machine learning and increased accessibility of data. However, despite the growth of the semantic web community, the rate of adoption within non-technical fields remains inadequate [35, 48, 49]. This shortfall is attributed to the

Authors' addresses: Omar Mussa, mussao@cardiff.ac.uk, o.mousa@seu.edu.sa, Cardiff University, Cardiff, UK and Saudi Electronic University, Riyadh, Saudi Arabia; Omer Rana, Cardiff University, Cardiff, UK, ranaof@cardiff.ac.uk; Benoît Goossens, Cardiff University, Cardiff, UK, goossensbr@cardiff.ac.uk; Pablo Orozco-terWengel, Cardiff University, Cardiff, UK, orozco-terwengelpa@cardiff.ac.uk; Charith Perera, Cardiff University, Cardiff, UK, pererac@cardiff.ac.uk.

lay user's need to understand the data nature and how to query it. In terms of querying the Resource Description Framework (RDF) triplestore, SPARQL has been the predominant query language. Yet, it has been shown that using SPARQL demands considerable cognitive effort and poses challenges even for experts, particularly when they lack an understanding of the underlying data structure [4, 33, 61].

The potential of RDF has not been fully harnessed, particularly in specialised fields such as wildlife research, where the ability to access and interpret observational LD is crucial. Wildlife research often involves the collection and analysis of extensive datasets, where observations are derived from a multitude of sensors and sources. These datasets are inherently complex, with a single sensor capable of capturing a variety of measurements, leading to a many-to-one mapping of data points to data sources. For instance, in the context of behavioural ecology, applications frequently encompass a variety of sensors that monitor environmental conditions, track wildlife movements, and gather a plethora of spatial and temporal data [13]. These scenarios are aptly captured by ontologies like the "Sensor, Observation, Sample, and Actuator (SOSA)" [39], which offers a structured framework for representing sensors, observations, and related entities.

Despite the comprehensive nature of such ontologies, the accessibility and exploration of the complex and condensed relationships they represent remain a significant challenge. Traditional interfaces for RDF data exploration and SPARQL querying are not designed with the intricacies of observational data in mind, particularly the many-to-one relationships that are prevalent in wildlife research. This gap in the toolset available to researchers prevents the efficient exploitation of observational LD, which makes the lay user reluctant to adopt such technology. Our research aims to answer the following questions:

RQ1: How can the accessibility of SPARQL endpoints be enhanced to facilitate wildlife researchers in handling observational LD?

RQ2: What are the specific needs and requirements of wildlife researchers in terms of data extraction and exploration from observational LD?

RQ3: Can the integration of a conversational user interface (CUI) with a graphical user interface (GUI) improve the process of query building?

This paper introduces a novel approach tailored to the needs of wildlife researchers, facilitating the accessibility of SPARQL endpoints and addressing the unique challenges posed by observational LD with many-to-one relationships. The research presents three key contributions to the existing body of knowledge:

- We introduce ForestQB, a novel query builder toolkit for wildlife researchers, which uniquely combines a conversational user interface (CUI) with the traditional GUI to enhance and streamline the extraction and exploration of information from observational LD, unlike existing systems that rely solely on GUIs.
- We conducted a comprehensive ethnographic study at the Danau Girang Field Centre (DGFC), to closely investigate wildlife researchers' needs to design the functional requirements of our proposed system.
- We conducted an evaluation to assess the efficiency and effectiveness of the proposed approach, which entailed a comparative analysis of accessing data through the GUI, CUI, and Integrated-UIs.

The paper is structured as follows: Section 2 provides an overview of the existing techniques applicable to accessing LD. This is followed by Section 3, which presents the ethnographic study and the process of gathering requirements. Subsequently, Section 4 offers a detailed overview of the ForestQB system, highlighting both its front-end and back-end components. Thereafter, Section 5 describes the evaluation design and results, with a discussion explaining the evaluation findings. Concluding the paper, Section 6 concisely recaps the study's findings and suggests directions for future research.

2 BACKGROUND AND RELATED WORK

There have been numerous efforts to simplify the linked data accessibility to lay users. A considerable portion of these efforts has focused on the difficulty of formulating SPARQL queries for the non-experts. Consequently, various techniques have been proposed to generate SPARQL queries in a more user-friendly manner. These techniques can be broadly categorised into several groups.

2.1 SPARQL Query Formulation

One approach is to construct SPARQL queries using a graphical user interface (GUI) which is known as SPARQL query builders. These GUIs are generally easier than writing SPARQL manually. However, it is important to consider that these GUIs may cater to different types of users. SPARQL query builders can be categorised as follows:

(1) *Form-Based Query builder* (e.g., Konduit VQB [2], BioGateway App [37], Wikidata Query Service (WQS) [1], SparqlFilterFlow [28, 29], Linked Data Query Wizard (LDQW) [33], ExConQuer [4] and Falcons Explorer [21]) uses traditional form elements to access the data by interacting with the system elements. By leveraging the user's familiarity with conventional forms, form-based QBs are generally designed to be accessible and suitable for a broader range of users. Konduit VQB, BioGateway App, WQS, and SparqlFilterFlow all share a common attribute in that they depend on the visualisation of the RDF triple structure through the employment of pre-populated drop-down lists. Consequently, users are required to select an item and specify filters for the final output. Although both Konduit VQB and BioGateway App demand a certain level of experience in SPARQL, WQS, and SparqlFilterFlow offer more simplified and intuitive designs.
Besides the familiarity with conventional forms, LDQW, ExConQuer, and Falcons Explorer have further enhanced the user experience in querying and exploring linked data by relying on facet search and users' familiarity with spreadsheets. After retrieving the preliminary results, users are given the ability to apply a variety of filters, thereby facilitating the adjustment and customisation of the query to align with their specific preferences and requirements.

(2) *Graph-Based Query builder* (e.g., iSPARQL[2], RDF Explorer [60], ViziQuer [18, 64], QueryVOWL [30], NITELIGHT [50, 51, 55], Graphical Query Language (GQL) [9], SPARQLinG [36] and OptiqueVQS [56]) creates queries by manipulating interface elements that are expressed in some semantic graphical notation. Most of the efforts in creating SPARQL QBs are focused on Graph-based visualisation, as this reflects the most natural behaviour when thinking about LD. The majority of these visualisations are based on the RDF triple pattern, due to the actual structure of the data, as in iSPARQL, NITELIGHT, ViziQuer, RDF Explorer, and OptiqueVQS. Other tools have proposed alternative visualisations in order to increase the abstraction and simplify the design. For example, the GQL tool uses UML, while SPARQLinG has created a custom visualisation called RDF-GL. While the use of the RDF triple may seem natural for Expert-users, greater abstraction seems to encourage Lay-users engagement.

(3) *Natural Language Query builder* (e.g. SPARKLIS [26], onIQ [25], Querix [41], QUICK [62] and NLP-Reduce [40]) is the simplest to use but lacks expressivity in terms of query formulation compared to the previous tools. The

[1]https://query.wikidata.org/
[2]https://virtuoso-catalogue.d4science.org/isparql/

user only needs to type their question in natural language, and the tool seeks to generate a matching query. However, this approach carries two inherent challenges. The first challenge is the users' linguistic variability, affecting the accuracy of generated queries. QUICK attempted to cope with this issue by displaying all of the possible queries for the user to confirm their intent which will increase their satisfaction. The second challenge is the ambiguity that users may experience while attempting to ask questions. SPARKLIS sought to address the ambiguity problem by introducing a controlled sentence formulation experience by restricting the user to dropdown lists to formulate a complete sentence. By limiting the sentence formulation, SPARKLIS has also eliminated linguistic variability.

2.2 Question Answering

A commonly used approach is to use natural language to answer users' queries which are known as *Question Answering Applications* (e.g. CubeQA [34], QA^3 [5], NLQ/A [63], Aqqu [10], BELA [53], and SINA [54]). These applications interpret users' queries as complete sentences and deliver the answers in natural language. This approach is based on natural language processing (NLP) techniques that enable users to express their queries in natural language, which is then translated into a SPARQL query within the system. Therefore, it provides the highest abstraction to the user, which produces high uncertainty regarding the underlying structure of the data and the queries that might be asked. In addition, due to linguistics variability, understanding the user intent is still challenging, affecting the overall user experience.

The Question Answering over Linked Data (QALD) initiative has been at the forefront of advancing research in LD for more than ten years, significantly improving methodologies for extracting a diverse range of question types [24]. These include Factoid, List, Boolean, Temporal, Statistical, and Multilingual Questions. Initially centered on DBpedia, the focus has recently shifted towards Wikidata. Thus, the emphasis of the tools developed through the QALD challenges has primarily been on accurately answering a predefined list of questions, although there has been a gradual shift towards incorporating more user-centric aspects in recent iterations.

2.3 Semantic Search

Another approach used is *Semantic Keyword-Based Search* (e.g. [20, 22, 27, 45, 46]), which is a method for conducting searches based on matching keywords, similar to traditional search engines. This technique leverages the intrinsic structure of semantic web technologies, employing ontologies and RDF data to enhance the search process. Unlike conventional search engines that primarily rely on keyword matching, semantic keyword-based search attempts to understand the context and relationships between terms. For example, some tools have attempted to incorporate semantic relationships, utilising properties such as 'owl:sameAs', to retrieve related entities and expand the scope of the search results. This is particularly useful in linking different but equivalent entities across various datasets, thereby enhancing the comprehensiveness of the search results. However, the functionality of these tools generally remains confined to basic text search operations. While these tools can identify semantically similar entities, they are typically limited in their ability to perform advanced operations like aggregation, filtering based on specific criteria, or interpreting the queries that require an understanding of hierarchical or associative relationships between entities.

2.4 Traditional Browsing

Linked Data Browsers (e.g. Tabulator [11] and Ozone browser [16]) attempted to deliver a similar experience to traditional web browsing by exposing data in a structured manner that is easily navigable. These tools are designed to give the user a general understanding of the data through an entity-centric perspective that emphasises data visualisation and navigation [58]. Typically, these tools lack the capability to perform more targeted queries.

Numerous LD browsers, including notable examples like Humboldt [42], Parallax [38], gFacet [32], Rhizomer [15], and SemFacet [3], have adopted multifaceted categorisation design, commonly referred to as *faceted browsing*. This approach allows users to navigate through datasets by using an array of filters, thereby facilitating targeted access to desired information. These LD Browsers have made impressive progress in improving how data is visualised and navigated. They employ a range of techniques to enable smooth and intuitive exploration of data, with faceted browsing being a key feature. However, while these browsers excel in navigating and displaying LD, they often fall short in their ability to construct detailed queries. This limitation becomes particularly evident when interacting with knowledge graphs that comprise millions of data points (triples). As a result, answering complex queries or extracting specific information from such expansive datasets can become challenging and time-intensive process.

2.5 Insights from Current Literature

Despite the variety in methodologies for accessing LD, within the broader context of usability, both Form-Based and Natural Language-based query builders emerge as the most effective approaches for lay users [43]. The Form-Based method is distinguished by its capacity to offer moderate to high levels of expressiveness through an interface that is familiar in appearance, whereas the Natural Language-based approach is recognised for its intuitive nature and more rapid query formulation capabilities. Conversely, from the perspective of a lay user, the graph-based approach presents an unconventional interface, necessitating an understanding of RDF and the Semantic Web [23, 43]. Additionally, semantic search and LD browsing tools are not adequately equipped to function as tools for query formulation; rather, they are better suited as navigation tools that excel in exploring traditional LD, lacking the facility for crafting customised queries.

Clearly, every tool and visualisation method comes with its own set of strengths and weaknesses. However, a significant gap in the existing literature is the absence of exploration into the potential benefits of integrating multiple methods to mitigate these limitations and improve usability. For instance, no tool under review has integrated a natural language-based model with a form-based approach, a combination that could expedite the process of query formulation [47].

To the best of our knowledge, none of the approaches have considered the specific challenges of accessing observational LD. This type of LD necessitates a shift in focus towards data sources and their associated data, following a Many-to-One perspective where 'many' represents the observations and 'one' refers to the sensor or data source. The existing body of work typically considers simpler query patterns, such as 'Tokyo capital-of Japan', where both entities hold equivalent significance in the query construction. In contrast, observational LD demands a different approach, where the sensor or data source assumes greater importance than individual readings in the context of query formulation. By focusing on these data sources, our approach enables users to intuitively access all related observations, offering a more straightforward method than traditional LD query techniques. Therefore, our proposed method introduces a novel approach that combines expressivity with enhanced usability. It does this by integrating a form-based UI with a conversational UI, both specifically designed to access and explore observational LD. This toolkit design is tailored to meet the needs of wildlife conservation and ecology researchers, addressing this domain's unique challenges and requirements. Although most of the design decisions were intended to satisfy wildlife research expectations, the benefit of this integration is not limited to these types of users [47].

3 DOMAIN EXPLORATION AND REQUIREMENTS GATHERING

Experts in the wildlife domain are anticipated to have limited familiarity with semantic web technologies and often possess minimal technical experience in programming and query languages. This unfamiliarity with the semantic web is not unique to wildlife research but represents a widespread challenge across various disciplines [35, 48, 49]. Consequently, this situation leads to a cautious approach by users towards adopting these technologies due to the significant effort required to learn and comprehend them. Therefore, the aim is to develop a visual query builder that is intuitive enough to construct the requisite queries without the need to use formal query languages to access the data. To address this, a user study was developed with three primary activities. The first includes an ethnographic study that involves being part of the data collection process to gain insight into the domain. The second activity seeks to understand the stakeholders through semi-structured interviews with bioscience researchers. Lastly, the third activity consists of two semi-structured focus group sessions to guide the tool's design process and data visualisation.

3.1 Activity 1: Understanding the Domain

An ethnographic study was carried out to gain an understanding of the nature of wildlife research, which is dissimilar from computer science research. To this end, a visit was made to the Danau Girang Field Centre (DGFC) in Malaysia, in order to observe the research activities and comprehend the requirements and data collection process. During the stay, the activities of both PhD and Bachelor's students were monitored, as well as additional activities related to research assistants and data management within the field centre.

3.1.1 Data Collection. Wildlife research typically requires a considerable amount of time to draw valuable insights from the data. Researchers may spend years monitoring animals and observing them in their natural habitat in order to comprehend a particular behaviour. This data is usually recorded manually on a survey sheet. This includes manual entries for GPS locations, humidity, temperature and time, which are obtained from the appropriate device. GPS collars can be used on a variety of animal sizes, including smaller species like leopard cats. While larger animals can be equipped with GPS collars that transmit data via satellite, allowing for remote tracking, smaller animals are often fitted with GPS collars that do not have satellite connectivity. In these cases, researchers need to be in closer proximity to the animal to download the data directly from the collar. However, it's important to note that only a few animals within a population are usually monitored using GPS collars. Consequently, a predominant portion of wildlife tracking activities is executed through conventional approaches. This process typically entails the direct localisation of animals utilising radio telemetry or the navigation to specific sites guided by GPS coordinates (see Figure 1). During this activity, we monitored the data-gathering process in the following research projects:

- **Pangolin Surveying:** The Sunda pangolin (Manis javanica) is listed as critically endangered on the IUCN RedList [19] and several individuals are closely monitored by wildlife researchers at the centre. As pangolins are small mammals covered in scales it is difficult to tag them with GPS trackers. Therefore, the researchers have opted to install a Very High Frequency (VHF) transmitter, which is a small chip placed on its back that enables tracking of the animals using a Radio Telemetry device. Every morning researchers ascertain the tagged pangolin location and manually register all pertinent details including its GPS location and behaviour.
- **Monkeys Surveying:** This research project was undertaken to monitor and analyse the movements and behaviours of proboscis monkeys (Nasalis larvatus), long-tailed macaques (Macaca fascicularis) and other primate species living in the vicinity of the riverbank. The purpose of the project was to gain further insight into the habits and behaviours of these primates and to monitor their whereabouts in order to better understand the species and their environment.

Fig. 1. A sample of equipment utilised during the data collection process includes, but is not limited to, a Radio Telemetry device, Cameras, a humidity logger, and a temperature sensor.

To gather the necessary survey data and monitor diurnal behavioural variations, the students undertook twice-daily boat trips over a period of one week. The collected data included the GPS location, time, size of the troop, gender of the monkeys, weather conditions, and detailed behavioural notes.

- **Lizards Surveying:** For a period of one week, the students conducted a study in which they monitored the behaviour of lizards along the main trail leading from the river to the field centre, recording their observations every ten feet twice daily. The purpose of this experiment was to determine the differences in the quantity and types of lizards present as well as their behaviour under various weather conditions and locations, such as close to human habitations and places of high noise levels. As part of the data collection, the students collected different data, including quantity, types, time, location, humidity, images, and temperature for each observation. This data was used to measure the impact of environmental variables on the population of lizards.

In addition, there are a number of regular trips that take place along the river, such as the night cruise, which is a boat trip that allows participants to identify various nocturnal species. When a rare animal is distinguished, the staff will snap a photograph and record the GPS location of the identified species. Generally, the data in the field centre were manually written into the survey paper.

3.1.2 Data Transformation and Management. As previously mentioned, the majority of data is initially transcribed by hand onto survey paper before being transferred into digital format, often through the utilisation of spreadsheets by individuals such as students, research assistants, or volunteers. The data acquired is primarily of significant importance to the primary researcher, although there may be instances in which it must be converted into another format for compatibility with alternate software. Images, on the other hand, are maintained in their original format, with their filenames recorded in the spreadsheet.

Subsequently, the data manager undertakes a backup of all data to avert any potential loss of data. Upon completion of data collection for a particular project by the principal investigators, the data manager assumes responsibility for archiving all data to facilitate future accessibility by other researchers. Consequently, other researchers must communicate with the data manager to request relevant data, including data relating to a specific project, location or species. Nonetheless, as the number of records archived continues to increase to billions over the years, it has become a formidable task for the data manager to rapidly filter out and extract data that would be useful to researchers.

The data manager is confronted with an additional obstacle concerning the images from camera traps, which are in the millions and remain unlabelled. Although volunteers and students attempt to categorise the images, there are still numerous images that remain unlabelled and, consequently, are not utilised effectively. This can lead to a failure to identify the whereabouts of endangered species or promptly detect poaching activities, which is of great concern.

3.2 Activity 2: Understanding the Stakeholders

Comprehending the project's users is a crucial undertaking, and as such, we conducted semi-structured interviews with ten participants comprising wildlife researchers, staff, and volunteers associated with the field centre. To be selected, participants were required to have firsthand experience working in DGFC and engaging in wildlife research. All participants, except for the data manager who has a background in Computer Science, hold certification in biosciences. The interviewees' experience levels spanned from one year to over 20 years. The interviews were conducted on a one-to-one basis, either in person during our visit to DGFC or at their respective offices, with the exception of one participant who was interviewed online.

We requested that the DGFC manager nominate participants for our study and distribute invitations containing information regarding the purpose of the study. Upon acceptance of the invitation, a suitable time was scheduled for the interview to take place. The research team presented the participants with the same set of questions, and follow-up inquiries were posed based on the individual's responses. The questions were categorised into four sections to facilitate a comprehensive understanding of the participant's role, data collection methods, data visualisation approaches, and the user's desired outcomes. In the initial section, participants were asked about their involvement at DGFC, the type of data they had previously worked with, the processing techniques employed, and the tools used for data processing. In the second section, the research team questioned participants about their perspective on the current system and identified any drawbacks they had encountered. The third section involved questions relating to data visualisation and the common features of the data being collected. Finally, the fourth section centred on the user's viewpoint regarding data retrieval and the preferred user interface for data presentation.

3.2.1 Data Collection. Participants in the study held varying roles at DGFC, such as Management, Researchers, PhD Students, Research Assistants, Data Managers, and Volunteers. Therefore, each participant had distinct experiences and exposure to diverse data collection methods, which included GPS collars, Camera Traps, Survey data, Drone images, Tree measurements, and weather data. Textual data were mostly transferred into CSV files either through manual entry by students and volunteers or fed from devices such as GPS collars. The data were subsequently formatted or merged manually for analysis or used with different software.

Regarding system weaknesses, the participants highlighted two common issues in dealing with data. Firstly, there was a high likelihood of errors when digitising data, such as during the conversion of survey data to CSV or the labelling of camera trap images. This process also took a considerable amount of time when dealing with large amounts of data. Secondly, there were concerns with the current data archiving methods, as millions of records collected over time were not being utilised optimally, and it was challenging to extract useful data from this vast pool of records.

3.2.2 Data Visualisation. Tables and maps were identified as the most prevalent data presentation methods by the participants. Tables were deemed a natural means of presenting data, while GPS location was deemed essential for most research purposes. Maps were also considered beneficial in easily visualising the location, while tables were deemed useful for data analysis.

During the study, the participants were presented with various interfaces and requested to provide feedback on each one. They were also encouraged to elaborate on features they preferred or disliked. The participants' preferences in data retrieval were investigated as well. The volunteers and data managers expressed a greater interest in simple data visualisation with maps, while the researchers preferred more customised search options.

3.3 Activity 3: Designing the Search Process

The third activity involved brainstorming for stimulating the design process, for which two focus group sessions were conducted, involving nine and six participants, respectively. Participants comprised staff, researchers, and students. The sessions began by providing an explanation of the project's aim and emphasising the need for support from the participants. Attendees were then urged to pose queries and engage in discussions related to the topic of focus during the sessions.

3.3.1 Questions Answering. To gain a comprehensive understanding of the data, the study aimed to identify the progression of a simple query to a more complex one. To accomplish this, participants were requested to note down the questions they sought answers to regarding the data on paper, which were then shared with other participants for discussion. Attendees were urged to evaluate the pertinence of these questions to their respective work. As a result of the participants' diverse backgrounds and the nature of their work in the field, the queries varied in complexity, ranging from straightforward to very intricate. However, several questions were shared among participants, including inquiries about the latest location of a particular animal and its whereabouts at a specific time.

3.3.2 Data Visualisation. During the second phase of the focus group sessions, participants were asked to create visualisations of the data results to identify their preferred method of data presentation. Attendees were encouraged to share their visualisations and explain their perceived usefulness. Furthermore, participants had the freedom not only to introduce their own visualisations into the discussion but also to actively comment on and discuss each other's contributions. In alignment with the preferences articulated in Section 3.2.2, where interviewees demonstrated a fondness for using maps and tables, this inclination was also confirmed during this exercise. Maps and tables emerged as the predominant methods for data presentation, frequently used by participants in this phase of the study. However, a few participants opted for bar or pie charts to facilitate variable comparison.

3.3.3 UI Requirements. In the final phase of the study, the primary aim was to ascertain the most essential features and understand user expectations. While the participants had no prior experience in user interface design, they were encouraged to participate in the design process and express their requirements. To facilitate this, printed user interface designs were shared with the attendees, who were free to borrow any useful features. Some users opted not to refer to these designs and instead relied on their creative skills to generate ideas. Subsequently, each participant presented their interface and explained the included features. Participants were then encouraged to discuss their opinions on other participant interfaces and to identify which of the designs was the most useful to them. Finally, they had to select their preferred features from the printed interfaces and provide reasons for their choices.

Most of the participants' sketches incorporated a map with dropdown lists. One attendee added a year slider to filter results by year, as precise dates were not necessary for their job. Other participants included options such as date pickers and form-like formats. All of the sketches contained either simple or advanced options, with the latter being more relevant to individuals directly involved in research who preferred more specific results.

3.4 Use Case Scenarios

During the previous activities, we gathered several use case scenarios to capture various facets of the dataset and fulfil the needs of the users.

3.4.1 Use Case 1: Explore and Visualise the Results using Maps. An essential task that users may wish to carry out in order to grasp the content of the dataset is exploring the linked data. This use case comprises two distinct activities: (1) selecting sensors at random to obtain a better understanding of the available data, and (2) generating random queries and visualising them on a map without a particular objective in mind. Consequently, the interface must provide visibility to all available sensors and allow users to visualise the outcomes on a map.

3.4.2 Use Case 2: Explore Available Sensors within a Geographic Location. The data collected by wildlife researchers usually contains location data, which is a critical feature. One of the essential use cases identified by all users is exploring sensors within a geographic location. In this scenario, the user can select a specific geolocation and obtain information on all sensors linked to that location. The ability to answer a straightforward question like “What sensors are available at this location?” is essential in the query formulation process. Subsequently, based on the response to the aforementioned question, the user can determine what other questions to ask about the location.

3.4.3 Use Case 3: Explore Available Data within a Geographic Location and Date Range. Responding to user queries will necessitate the extraction of more information from the dataset. The simplest queries from users involve the use of geolocation and date range filters. Users may ask for specific sensor data that falls within a particular location and/or time frame. For instance, a user might inquire, “Where is (Sensor A) located in (Location B) during (Date Range C)?” The query could also be altered to modify the size and order of the results to answer questions such as “What were the ten latest/oldest (order) readings of (Sensor A)?” along with a location and/or date range if desired.

3.4.4 Use Case 4: Narrowing Down the Results by Applying Filters Freely on the Sensor Properties. Alongside ‘Use Case 3’, users may seek to enhance their inquiry by incorporating explicit filters for each observable property associated with the sensors. This will enable the user to customise their query construction to suit their specific needs. If deemed necessary, the user should also have the ability to eliminate undesired observable properties to align with their inquiry. For instance, if a sensor is fixed to an animal to monitor its temperature and speed, the user can formulate a query to extract readings that have high temperatures but no detected movement, which could be indicative of the animal’s sickness.

4 FOREST QUERY BUILDER (ForestQB)

ForestQB represents an integrated toolkit, meticulously designed to streamline the process of query construction for a wide array of users, ranging from novices to experienced individuals. This facilitation is achieved through the integration of a variety of components, encompassing both simplistic and advanced form-based user interfaces, as well as conversational user interfaces. The interface components are populated by sending multiple SPARQL queries to the SPARQL endpoint and reflect the appropriate options. Thus, if the SPARQL endpoint is not responding to the tool request, the tool will fail to construct the correct SPARQL queries. The following subsections will describe each component.

ForestQB design focuses on data that describe sensors and their observations, including spatial queries, reflecting the case for wildlife data. Initially, the query is constructed as a JSON object, which is then sent to a custom parser to convert it into its equivalent SPARQL query. Eventually, the query is sent to the endpoint, and the result is displayed in a tabular format.

Regarding spatial queries, ForestQB supports constructing queries with geof:nearby and geof:within filters. However, geof:within is only supported with enhanced polygons which some SPARQL endpoints might not natively support. ForestQB is available online [3] and includes sample data to test and use with a demonstration video.

4.1 Design Rationale

The design rationale for the user interface was shaped through a comprehensive review of existing literature, complemented by findings from the conducted ethnographic study, particularly highlighted in Activity 3 (refer to section 3.3). In the phase of establishing UI requirements, a predominant preference emerged among participants for a form-based design coupled with an integrated map. This preference was further supported by the analysis of existing form-based tools, which demonstrated their suitability for non-expert users in terms of both expressivity and usability for query formulation. Moreover, an analysis of the proposed questions and the identification of primary relevant use cases (as discussed in section 3.4) led to the decision to incorporate a conversational UI. This inclusion aims to address the shortcomings inherent in solely form-based query interfaces.

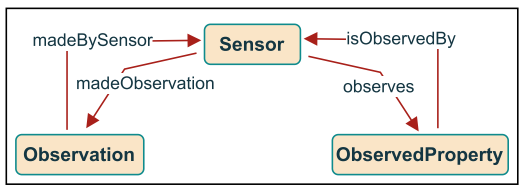


Fig. 2. Demonstrate the relationship between readings (Observation), sensors and the property of the observed data. The sensor is the key entity in accessing the data.

From the data structure perspective, the dataset to be utilised diverges from traditional LD, encompassing sensor and observational data, thereby classifying it as observational LD. This uniqueness necessitates a design focus on the data sources. Thus, instead of asking the user to specify the RDF pattern, the design focuses on the sensor, which is the primary resource to get all related data (see Figure 2). Consequently, the principal design elements include a Sensor List, a Map, a Form-based Filter Panel, and a chatbot-like interface (see Figure 4), which are explained as follows:

(1) Sensors representation: A drop-down list with integrated search functionality. The initial options presented in this list consist of all entities that have been identified as SOSA sensors, which the user can select. Additionally, the user has the option to input any text, which will activate an autocomplete feature similar to those commonly found in search engines, allowing the user to search for entities that are not included in the initial list. Upon selection of at least one of the listed entities, the search will be initiated to retrieve all related entities. Thus, the user has the ability to select a sensor and retrieve all its related entities and readings with a single click, which satisfies Use Case 1.

(2) Map-based query representation: The primary interface that is meant to allow users to efficiently narrow down their search results by specifying certain criteria, such as location and date range filters, that the data must fulfil. In addition, map filters will facilitate the identification of patterns and trends within the data. Thus, this interface will allow the user to answer queries from Use Cases 2 and 3.

[3]https://github.com/i3omar/ForestQB

(3) Form-based query formulation: It is the advanced query representation interface that is meant to allow the user to specify complex queries with multiple criteria to narrow down the search results. This interface consists of a series of input fields, such as text boxes and drop-down menus, that the user can use to specify the characteristics of the data they are looking for. Use Case 4 will be satisfied by this interface.

(4) Query formulation assistant: The conversational UI is implemented to aid the user in generating queries by filling the relevant input fields on behalf of the user or by providing responses in a natural language format. ForestBot allows for the exploration of data in accordance with Use Case 1 and can also be utilised to address inquiries relevant to Use Cases 2 and 3.

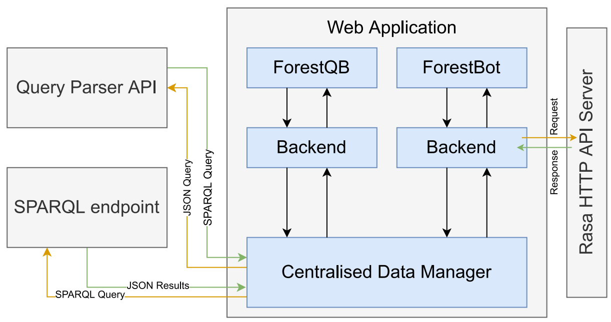


Fig. 3. ForestQB system overview.

4.2 ForestQB Frontend

ForestQB interface is designed to be a combination of a form-based interface and a conversational user interface, as well as featuring map filters. In addition to offering users the flexibility to use either interface or both for constructing queries, ForestQB aims to improve the conversational UI experience by reflecting all of its input on the GUI for verification and for the purpose of building the query incrementally by adding more entities and filters from both interfaces.

The toolkit is a single-page application built using plain JavaScript and the Vue.js framework (vuejs.org). By utilising Vue.js, the application is designed as independent components that serve a variety of functions while sharing the same data through access to a centralised data store implemented using Vuex (vuex.vuejs.org). The centralised data store reflects all user inputs and is used to assemble the JSON query object. Upon the user initiating a search, this object is transmitted as an HTTP request to the query parser, which in turn responds with the matching SPARQL query (see Figure 3).

Generally, ForestQB consist of three main views that are displayed in Figure 4. These views (V) are explained in the following list:

(V1) The first view comprises a Map view with integrated map filters, a date range picker, and a list of entities/sensors with a built-in keyword search function.

(V2) The second view exposes all entities that are linked to the selected sensor from V1 and will remain hidden if no sensor is selected. This view enables the user to add filters for each individual entity.

(V3) The third view is the conversational UI pop-up view, in which the user can express their query using natural language.

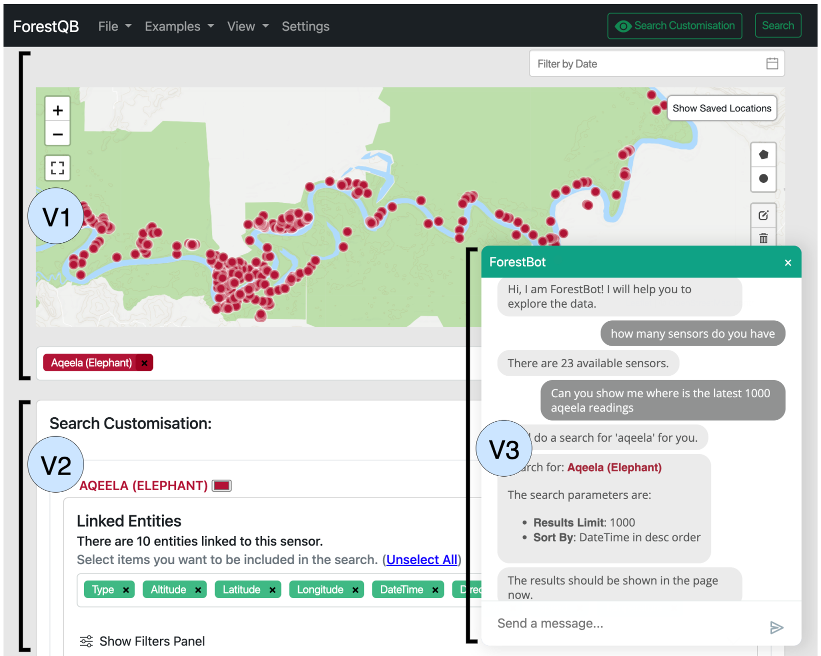


Fig. 4. An illustration of the ForestQB interface, utilising an exemplar query to highlight its three principal views. V3 presents the query in its natural language format, demonstrating the interoperability of the conversational UI to reflect the user's inquiry in V1 and V2. (Demo Link)

V1 and V2 are connected together as basic/advance query formulation views. However, V3 is an independent assistive view that can also populate V1 and V2 based on the user query. An example of an interaction between the user and the interface may involve the following:

(1) From V1, the user selects a map location by drawing a map filter. The tool will then list all related sensors.
(2) The user selects some of the interesting sensors from the sensors list in V1. V2 will expose all entities related to that sensor.
(3) The user can then add a date range filter from the V1 range picker or individually from V2 filters panel.
(4) The user can additionally ask in natural language using the conversational UI in V3 and automatically populate all required filters and entities from V1 and V2.
(5) During the construction of a query, users can continuously refine their search by adding filters and instantly see the updated results. This can be done either by using the search button on the GUI or by making a request through the CUI. This interactive process helps them fine-tune their search until they arrive at the desired outcome.

4.3 ForestQB Conversational UI (ForestBot)

In this study, we employ Rasa, an open-source conversational UI platform, to identify the intent and entities within user queries. The training data in our approach includes a set of potential questions that users may ask, as well as entities that may be present within those queries, which were initially gleaned from the user study. ForestBot comprises two main components: the conversational UI and the backend response handler. The conversational UI, which is powered by Rasa, is responsible for classifying the intent of the query and extracting relevant entities. This component is deployed through the use of Rasa's HTTP API server. On the other hand, the backend response handler is implemented as a JavaScript component that processes Rasa's responses and determines the appropriate actions to take, as well as how to populate relevant interface elements (see Figure 5).

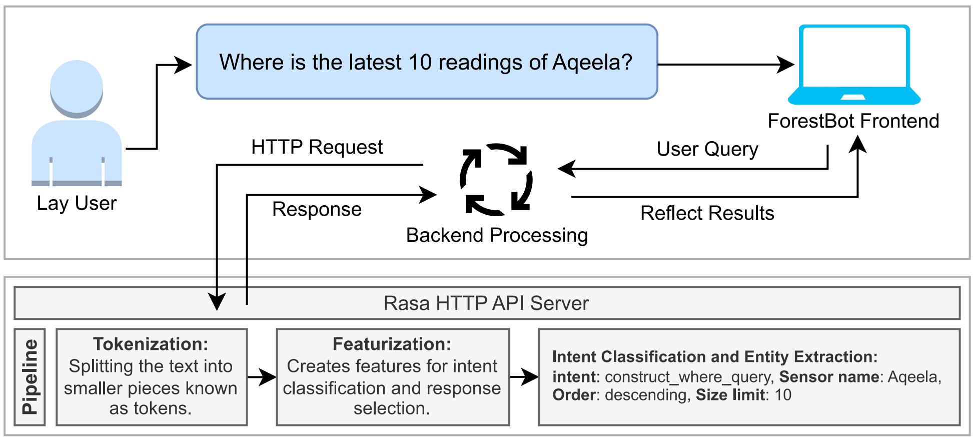


Fig. 5. An overview of the conversational UI backend.

4.3.1 Pipeline. The pipeline employed is composed of various elements, including pre-trained models utilising the spaCy and MITIE libraries. The pipeline generally comprises of Tokenizers, Featurizers, Entity Extractors and Intent Classifiers. These components are specified within the Rasa config.yml file, which is responsible for the configuration of the pipeline. The tokenization process is being performed using the spaCy tokenizser in conjunction with the Sparse featurizer, which is created using the CountVectorizer from sklearn to generate a bag-of-words representation for the user message, intent, and response (see Figure 5). We used the MITIE Text Categorizer for intent classification, which employs a multi-class linear support vector machine (SVM) with a sparse linear kernel. The pipeline, along with its source code, is currently available for public access on GitHub (github.com/i3omar/ForestBot).

4.3.2 Entity Extraction. To extract entities such as names, locations, and dates from user queries, we employed multiple entity extractors, including spaCy, Conditional random field (CRF), and MITIE. Additionally, we used the Rasa RegexEntityExtractor to extract entities from lookup tables in the training data and the EntitiesSynonymMapper to link synonymous entities. To enable users to inquire about relative dates and durations, we incorporated the Duckling Extractor, which requires the use of a duckling server to parse time into a structured format. Subsequently, the system employs SPARQL queries to search the endpoint for all detected resources, thereby confirming their validity and enhancing the understanding of user queries. This approach notably does not depend on specific terminologies or a predetermined set of data, a strategic choice that guarantees the scalability of the approach across diverse datasets. Consequently, the process may initially identify a set of resources, for instance, five, but ultimately confirm the validity of only one. This design intended to ensure high flexibility and prevent the need for retraining in the event of data updates or changes.

Table 1. List of intents with their corresponding example query.

#	Intent Name	Example Query
1	list_all_sensors	List all available sensors.
2	count_all_sensors	How many sensors are available?
3	discover_sensors_within_location	What are the sensors within Area 1 and Area 2?
4	list_observable_info	What is Dara?
5	construct_where_query	What are the 100 most recent Aqeela readings in Area 1 during 2015?
6	construct_when_query	When was the first reading of Dara and Jasper?
7	show_examples	What questions can I ask?

4.3.3 Intents. In any given user communication with the conversational UI, identifying the user's intent, or the purpose they are attempting to convey or achieve (such as expressing a greeting or indicating a specific location), holds significant value to the successfulness of this paradigm. Our model's conversational capabilities include the ability to recognise and respond to chitchatting intents such as 'greet' and 'goodbye' through the direct output of predefined text responses in addition to more sophisticated inquiries stemming from Use Cases 1, 2, and 3. In order to respond to these sophisticated queries, the conversational UI must first identify the intent and extract all relevant entities from the user's question, which necessitates the grouping of similar questions under a single intent. Table 1 presents a list of intents included in the training dataset, accompanied by a corresponding example for each intent. It is important to note, however, that chitchatting intents are not covered in this table. These chitchatting intents comprise nine distinct types, specifically engineered to identify conversational nuances such as greetings and affirmations.

The initial intents, namely 'list_all_sensors' and 'count_all_sensors', serve to facilitate the user's exploration of the data and enable them to identify relevant information to inquire about further. Subsequently, the third intent facilitates the user's ability to inquire about sensors within a specific filter on the map or by allowing the user to directly specify the location's name. Once the user has identified the specific sensors or entities they are interested in, they may pose inquiries in order to retrieve information pertaining to that entity, similar to the inquiry outlined in the fourth intent. The fifth and sixth intents ultimately serve to populate the user interface options to construct the desired query. The last intent provides the user with a list of questions that they can select and utilise as a method of learning by example.

4.4 Scalability of the Approach

As previously mentioned, our approach avoids the utilisation of SPARQL templates or skeleton queries, which typically involve a predefined array of queries with empty slots for input. This traditional approach restricts the user interface's ability to flexibly and effectively manage dynamic and evolving queries. Rather, our strategy relies on the use of the centralised storage to construct a JSON query object, as illustrated in Figure 3. Subsequently, this JSON is translated into its corresponding SPARQL query. This approach enables the incorporation of an arbitrary combination of entities and filters, without requiring users to refer to a specific sequence.

The CUI aligns with this principle, allowing users to seamlessly integrate new resources through the assistant. It also enables the use of the GUI for adding additional entities/filters, while additionally providing the option to return to the CUI for further adjustments to the query. This approach supports a dynamic and non-linear process of query construction, enhancing user interaction and flexibility.

In the context of the CUI's scalability, it is important not to underestimate the model based on its design, which identifies a finite array of intents. The primary objective of this design is to categorise user queries in a manner that aligns generally with the relevant use case. Upon identifying an intent, the CUI engages in a comprehensive search for pertinent entities, deploying multiple SPARQL queries to validate these entities thoroughly and formulate the appropriate query. This strategy is not limited to predefined terms; rather, it adopts an ontology/dataset-independent methodology, enabling effective functioning across various endpoints without necessitating retraining or reconfiguration of the CUI. This aspect underscores the approach's scalability, rendering it compatible with any dataset adhering to the SOSA [39] or a similar ontology.

4.5 Dataset and SPARQL endpoint

The dataset used in this study contains sensitive historical data collected by bioscientists that has been modelled as linked data and is privately available through a SPARQL endpoint. This data populates the Forest Observatory Ontology (FOO)[4], a novel ontology that describes wildlife data generated by sensors. The design of FOO involved a comprehensive analysis of current state-of-the-art ontologies, along with the integration of components from various established sources. The development of FOO was motivated by the goal of constructing the Forest Observatory through the review of Open Data Observatories [31]. The data in the dataset is structured using multiple ontologies, including the SOSA [39] ontology. As a result, ForestQB initially relies on SOSA to access connected sensors and their observable properties, and has the potential to function with other endpoints that also utilise SOSA.

In the process of determining the appropriate approach for sensor data extraction, the tool executes a SPARQL query to determine the underlying ontology. In instances where neither the FOO nor the SOSA ontology is detected, an option is made for users to undertake manual configurations. This feature enables users to specifically tailor the identification parameters for the sensors, thereby ensuring the tool functions correctly with different ontological structures. In light of the sensitive nature of our primary dataset, which prevents its publication, we have created and released a synthetic observational dataset specifically designed to populate the SOSA ontology. This dataset, along with the source code, is currently available for public access on GitHub (github.com/i3omar/ForestRDF).

5 USER EVALUATION

5.1 Experiment Design

We conducted a user study with task-based evaluations. The study aimed to compare the participants' ability to complete tasks using the proposed approach versus using either the GUI alone or the conversational UI alone. The rationale for adopting this approach stems from the absence of available tools specifically tailored to access observational linked data, as outlined in Section 2. The evaluation was implemented in order to gauge the effectiveness and efficiency of the tool, as well as the user satisfaction. Accuracy and completeness of the tasks that the users can accomplish were used to measure the effectiveness [12, 17].

It is an established fact that Information Retrieval (IR) systems are conventionally evaluated based on three metrics: Precision, Recall, and F-Measure. However, from the study context, ForestQB is a data retrieval (DR) system where accuracy is more prioritise over these metrics as irrelevant or missing entities viewed as a total failure [6]. In other words, we have adhered to a rigorous approach in which any data that is deemed inconsequential is treated as a failure. In addition, the amount of time required to complete the task was utilised as a metric for efficiency [12, 17].

[4]https://naeima.github.io/Forest-Observatory-Ontology/

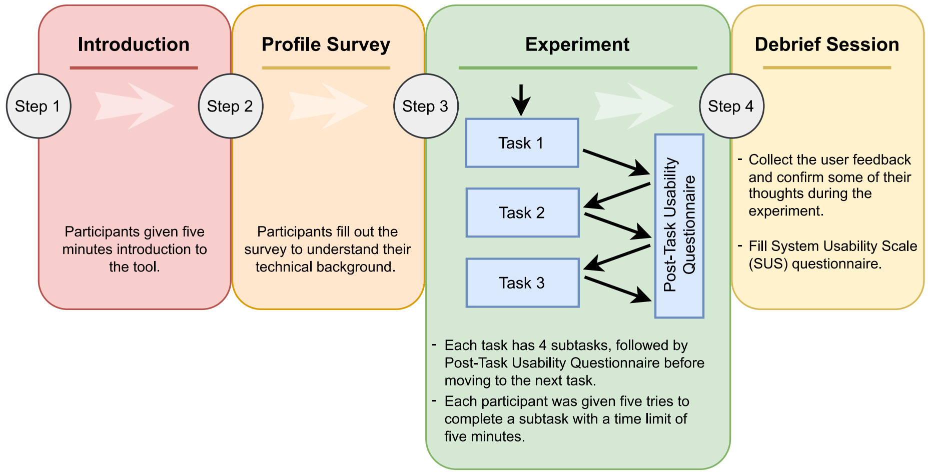


Fig. 6. An overview of the experiment design used in this study.

Each participant was asked to perform a set of tasks in three separate sessions, with each session using a different interface. The three interface options included: 1) a graphical user interface (GUI) only, 2) a conversational user interface (CUI) only, and 3) both interfaces used in parallel. Section 5.2 provides a detailed discussion of the task versions and their respective order. Following each session, the participant was presented with a 7-point Likert scale Post-Task Usability Questionnaire consisting of three questions that were adapted from [44, 52]. The questionnaire was used to assess the participant's satisfaction with ease of use, the time required, and the level of information provided to complete the tasks.

The think-aloud protocol was used, and participants were instructed to express their thoughts by verbally describing their thinking while using the tool. The purpose is to capture challenges the user meet, such as frustration, stress and confusion while using the tool. They were given a 5 minutes introduction to the tool. Then, each participant was asked to fill out the survey to understand their technical background. Each participant was given five tries to complete a task with the right to skip the task, which will be considered a failure to complete a task. A task is deemed unsuccessful if a participant is unable to complete it within a time limit of five minutes.

Finally, a debrief session was conducted at the end to gather the user feedback and confirm some of their thoughts during the experiment. To assess the usability of the user interface, the System Usability Scale (SUS) questionnaire [14], a standardised tool providing a quick and reliable measure of a system's usability, was utilised. This approach has been proven to be highly useful [7, 8]. Accordingly, participants were asked to complete the SUS questionnaire as a quantitative method to evaluate usability. Figure 6 presents a summary of the key steps in the experiment design used in this study.

5.1.1 Pilot Study. Three pilot studies were carried out to assess the evaluation plan and identify any potential challenges in order to ensure the system's preparedness for the experiment. As a result of these pilot tests, several bugs in the tool were detected and fixed, and the interface underwent slight modifications in preparation for the experiment.

5.1.2 Question Validation. To confirm the accuracy and validity of the tasks incorporated in the user experiment, a supplementary trial was conducted with a Subject Matter Expert (SME) who holds over two decades of professional

Table 2. The questions used in the experiments, with variables being highlighted for clarity, an addition not present during the actual study. GUI: graphical user interface, *CUI*: conversational user interface.

	T	Information Need	Used UI
GUI	1.1	What are `Kasih` readings between `2017-05-10` and `2017-05-11`?	
	1.2	Find all sensors, within a location called `area 3`	
	1.3	Where was the `last` `5` `Jasmin` readings within `area 3`?	
	1.4	Is there any `Aqeela` and `Putut` readings in `2012-08-16` where their `speed` was `less than 1`?	
CUI	2.1	Find all sensors, within a location called `area 1`	
	2.2	Find out what is `Ita`	
	2.3	Get Readings of `Ita`, only within the location called `area 1`	
	2.4	Readings of `Ita`, between `2014-07-23` and `2014-07-24` (within the location called `area 1`)	
Integrated-UIs	3.1	Find all sensors, within a location called `area 1`	GUI
	3.2	Find out what is `Sandi`	*CUI*
	3.3	Get `latest` `50` readings of `Sandi` (only within a location called `area 1`)	*CUI*
	3.4	Get the Readings of `Sandi` between `2015-08-11` and `2015-08-12` with `Temperature` `> 32`	GUI

experience and has actively participated in the requirements-gathering process via interviews and focus group. He was allocated version ABC and interacted with the system, following the same usage patterns shared with other participants (versions are discussed in subsection 5.2). The expert confirmed the accuracy of all proposed use cases and provided highly positive feedback. One of the tasks he confirmed was mirroring a real-world situation that they had previously encountered. He said that the tool allowed him to access the data quickly, where they could not easily access it. He reported that the tool facilitated rapid access to data that had previously posed significant challenges to access quickly.

5.2 Tasks Order

In order to prevent bias in evaluating the user interfaces, we created six different versions of the experiment. Each version corresponded to a different order of the three user interfaces, which we denoted as A, B, and C, with A representing the graphical user interface (GUI), B representing the conversational user interface (CUI), and C representing the integrated user interfaces (Integrated-UIs). The permutations of the user interface order used in each version were ABC, BAC, CBA, ACB, BCA, and CAB. All participants were presented with the same set of questions, as shown in Table 2, but the order of the questions was shuffled to match the version order.

5.3 Study Participants

The research involved 18 participants who are affiliated with the School of Biosciences (Cardiff University) and possess a background in ecology. All of the participants have either conducted wildlife research in the past or are currently involved in such research. None of the participants were familiar with the user interface being tested and had not previously participated in its design process. Table 3 summarises the participant profiles, highlighting a range of their years of experience with ecology. It further underscores their lack of familiarity with Semantic Web technologies.

Table 3. This table provides the profile information of the participants alongside their designated experiment version: A for GUI, B for CUI, and C for Integrated-UIs. They were followed by a self-assessment survey of their technical skills, familiarity with similar tools, and familiarity with Semantic Web technologies. A score of 1 indicated complete unfamiliarity, while a score of 5 denoted a high degree of familiarity.

#	Age	Exp	Edu	Major	Experience	Tech. Skills	Similar Tools	Sem. Web
P01	22	ABC	BSc	Biology	1-3 years	2	3	1
P02	21	BAC	BSc	Big Data Biology	1-3 years	3	1	1
P03	24	CBA	MSc	Wildlife Sciences	1-3 years	3	4	1
P04	23	ACB	BSc	Biosciences	1-3 years	2	3	1
P05	22	BCA	BSc	Biosciences	1-3 years	4	4	1
P06	35	CAB	MSc	Molecular Ecology	10+ years	4	4	1
P07	27	ABC	MSc	Conservation Genetics	7-9 years	3	2	1
P08	22	BAC	MSc	Global Ecology	1-3 years	3	2	1
P09	32	CBA	MSc	Biosciences	4-6 years	2	3	1
P10	25	ACB	MSc	Molecular Ecology	4-6 years	2	4	1
P11	26	BCA	BSc	Biological Sciences	4-6 years	2	4	1
P12	26	CAB	MSc	Biosciences	4-6 years	4	3	1
P13	28	ABC	MSc	Conservation Genetics	4-6 years	2	2	1
P14	22	BAC	BSc	Biological Sciences	1-3 years	2	2	1
P15	42	CBA	PhD	Conser. & Biodiversity	10+ years	2	2	1
P16	26	ACB	BSc	Biosciences	4-6 years	2	3	1
P17	22	BCA	BSc	Biological Sciences	1-3 years	2	3	1
P18	26	CAB	MSc	Biosciences	7-9 years	3	3	1

5.4 Quantitative Results

5.4.1 Time on Task. All participants successfully accomplished all tasks within the given timeframe. Notably, the task with the longest duration was completed in approximately three minutes and fourteen seconds, a time considerably less than the designated five-minute limit for each task. Furthermore, it was observed that no participant exceeded the maximum permitted number of attempts, which was set at five. The peak number of attempts recorded for any single task was three. Table 4 summarises the experiment's overall results, including time taken, completion rate, number of attempts and hints.

In order to evaluate the efficiency of interfaces in our study, we compared the average time users spent completing the same task in the initial introduction of the interface versus their subsequent interaction in the final session (see Figure 7). For instance, we analysed the time for the GUI when initially used in version ABC and ACB against its use in the final session in version CBA and BCA. This comparative analysis was undertaken with the primary objective of investigating whether user familiarity with the interface over a short period could potentially lead to improvements in performance. Additionally, we attempted to determine which specific interface demonstrated the most significant enhancement in its learning curve.

As illustrated in Table 5, Integrated-UIs demonstrate superior efficiency in facilitating user learning and task performance. Notably, Integrated-UIs showcased an improvement of over 50% in completion times for Tasks 1, 2, and 4, accompanied by a significant reduction in the variability of users' performance over repeated attempts. This reduction in standard deviation highlights a uniform enhancement in user proficiency, suggesting that Integrated-UIs not only expedite the learning process but also foster a more consistent learning curve across users. Statistical analysis by

Table 4. This table presents a summary of the experiment's overall results, highlighting the overall correct completion rate (CCR), the correct completion rate on the first attempt (1stCCR), the average time taken in seconds, the average number of attempts, and the average number of hints given for each task. It employs colour coding to compare similar tasks across Session 1 (e.g., Task 1.1), Session 2 (e.g., Task 2.1), and Session 3 (e.g., Task 3.1). The *red* colour indicates the lowest performance, **green** signifies the highest, and orange refers to intermediate outcomes. GUI: graphical user interface, *CUI*: conversational user interface, Integ-UIs: integrated user interfaces.

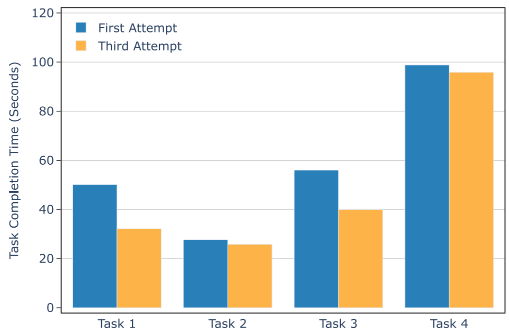
	T	CCR(%)	1stCCR(%)	Avg. time(s)	Avg. # of attempts	Avg. # of hints
GUI	1.1	100	**100**	*47*	**1**	0.3
	1.2	100	**100**	*26*	**1**	*0.2*
	1.3	100	94	*48*	1.1	*0.5*
	1.4	100	83	*101*	1.2	0.7
CUI	2.1	100	**100**	**13**	**1**	**0**
	2.2	100	*94*	**9**	*1.1*	**0**
	2.3	100	*83*	23	*1.2*	**0.1**
	2.4	100	*39*	80	*1.7*	*0.8*
Integ-UIs	3.1	100	**100**	27	**1**	*0.4*
	3.2	100	**100**	**9**	**1**	**0**
	3.3	100	**100**	**22**	**1**	**0.1**
	3.4	100	**100**	**72**	**1**	**0.6**

performing paired t-tests further confirms the effectiveness of Integrated-UIs, with significant p-values indicating noteworthy improvements in Tasks 1 and 4 (p=0.012 and p=0.004, respectively), and a nearly significant result for Task 3 (p=0.055), underscoring a faster achievement of proficiency. The overall evaluation, using Fisher's Combined Probability Test, resulted in a combined p-value of 0.00019 for Integrated-UIs, compared to 0.180 for GUI alone and 0.809 for CUI alone, emphatically supporting the hypothesis that Integrated-UIs markedly enhance learning efficiency and task performance.

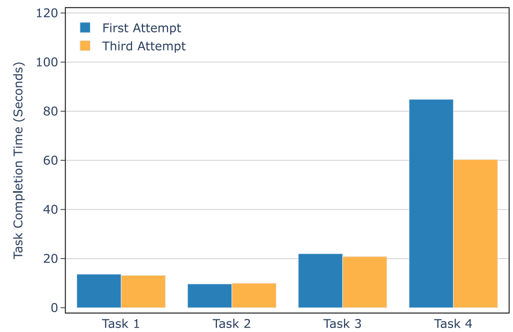
5.4.2 *Usability Questionnaires.* Two metrics were employed to evaluate user satisfaction with the tool. The System Usability Scale (SUS) was applied as the initial measure to assess overall user satisfaction with the tool. Concurrently, the Post-Task Usability Questionnaire was used as a means of comparative analysis across three distinct usage scenarios. The SUS revealed a score of 91.7 for the overall experience, corresponding to an A+ on the percentile ranking and indicating a high level of user satisfaction. Figure 8 depicts the frequency of SUS scores obtained from users. In addition, users gave a score of 9.2 out of 10 when asked about their likelihood of recommending the system to others. In terms of comparing the three usage themes as assessed by the Post-Task Usability Questionnaire, the Integrated-UIs exhibited superior performance in the categories of Time, Information Support, and Easiness.

Interestingly, the GUI and the CUI exhibited comparable results in the Time category, with numerous users encountering difficulties in the final task of the CUI session. In terms of Easiness, the GUI and CUI were closely matched, with a marginal preference leaning towards the GUI interface. However, when evaluating Information Support, the CUI emerged as the least effective among the three interfaces, with numerous users deeming it insufficient compared to the other experiences. Figure 9 summarises the participants' satisfaction ratings of the Post-Task Usability Questionnaire.

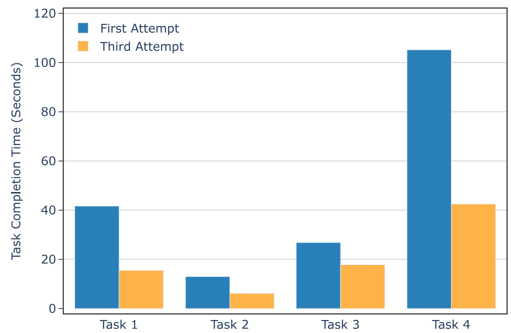
To support our findings, we performed statistical tests using paired t-tests to compare user ratings on Easiness, Time Efficiency, and Information Support across the three sessions (see Table 6). The results suggest that Integrated-UIs are generally rated better than both GUI and CUI in terms of Easiness and Information Support. For Time Efficiency, Integrated-UIs is rated significantly better than CUI, but no significant differences were found between GUI and the



(a) Completion time utilising the GUI.



(b) Completion time utilising the CUI.



(c) Completion time utilising the Integrated-UIs.

Fig. 7. Comparative analysis of task completion time (in seconds) between the initial and final attempts across the three user interfaces.

other two interfaces. This could imply that, overall, users found Integrated-UIs to be the most favourable across the sessions they experienced.

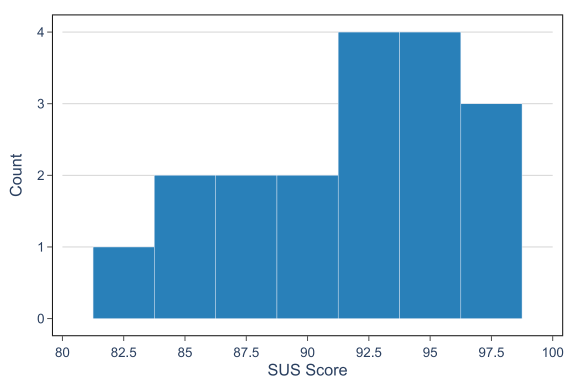


Fig. 8. Distribution of System Usability Scale (SUS) Scores. This figure illustrates the frequency of SUS scores obtained from users.

Table 5. Comparative Analysis of Task Completion Times for Participants Based on UI Sequence: Evaluating the Learning Curve Effect. This table presents average times (sec), standard deviations, and improvement percentages for initial and final attempts at tasks, comparing participants based on the initial vs. final use of a specific UI. Statistical analysis is performed using paired t-tests to assess the significance of improvements, with those statistically significant ($p < 0.05$—or very close to it) marked in **green**. Overall significance is assessed via Fisher's Combined Probability Test. GUI: graphical user interface, *CUI*: conversational user interface.

		Task 1	Task 2	Task 3	Task 4
GUI	Avg. Initial Attempt (s)	50.17	27.67	56	98.83
	Avg. Final Attempt (s)	32.17	25.83	40	95.83
	Improvement (%)	35.88%	6.63%	28.57%	3.04%
	Initial Std. Dev. (σ)	21.77	15.46	9.80	20.92
	Final Std. Dev. (σ)	3.92	15.26	13.15	11.23
	p-value	0.135	0.880	**0.039**	0.723
	Combined p-value	0.180			
CUI	Avg. Initial Attempt (s)	13.67	9.67	22	84.83
	Avg. Final Attempt (s)	13.17	10	20.83	60.33
	Improvement (%)	3.66%	-3.45%	5.3%	28.88%
	Initial Std. Dev. (σ)	4.03	2.66	15.27	57.52
	Final Std. Dev. (σ)	3.87	8.46	6.82	30.68
	p-value	0.837	0.938	0.819	0.164
	Combined p-value	0.809			
Integrated-UIs	Avg. Initial Attempt (s)	41.67	13	26.83	105.17
	Avg. Final Attempt (s)	15.50	6.17	17.83	42.50
	Improvement (%)	62.8%	52.56%	33.54%	59.59%
	Initial Std. Dev. (σ)	15.15	8.44	9.62	30.02
	Final Std. Dev. (σ)	3.89	2.14	6.94	4.85
	p-value	**0.012**	0.107	**0.055**	**0.004**
	Combined p-value	**0.00019**			

Table 6. Results of paired t-tests comparing perceived satisfaction ratings from the Post-Task Usability Questionnaire on Easiness, Time Efficiency, and Information Support across the three sessions (GUI, CUI, Integrated-UIs). The table presents p-values to indicate statistical significance between pairs of interfaces, with values less than 0.05 denoting significant differences. For each comparison where the p-value is **less than 0.05**, the 'Preferred UI' column identifies the interface with the higher mean score as the preferred interface. GUI: graphical user interface, *CUI*: conversational user interface, Integ-UIs: integrated user interfaces.

	Easiness		**Time**		**Information Support**	
UIs	p-value	Preferred UI	p-value	Preferred UI	p-value	Preferred UI
GUI vs. *CUI*	0.302	None	0.430	None	0.601	None
GUI vs. Integ-UIs	**0.028**	Integ-UIs	0.263	None	**0.005**	Integ-UIs
CUI vs. Integ-UIs	**0.005**	Integ-UIs	**0.024**	Integ-UIs	**0.012**	Integ-UIs

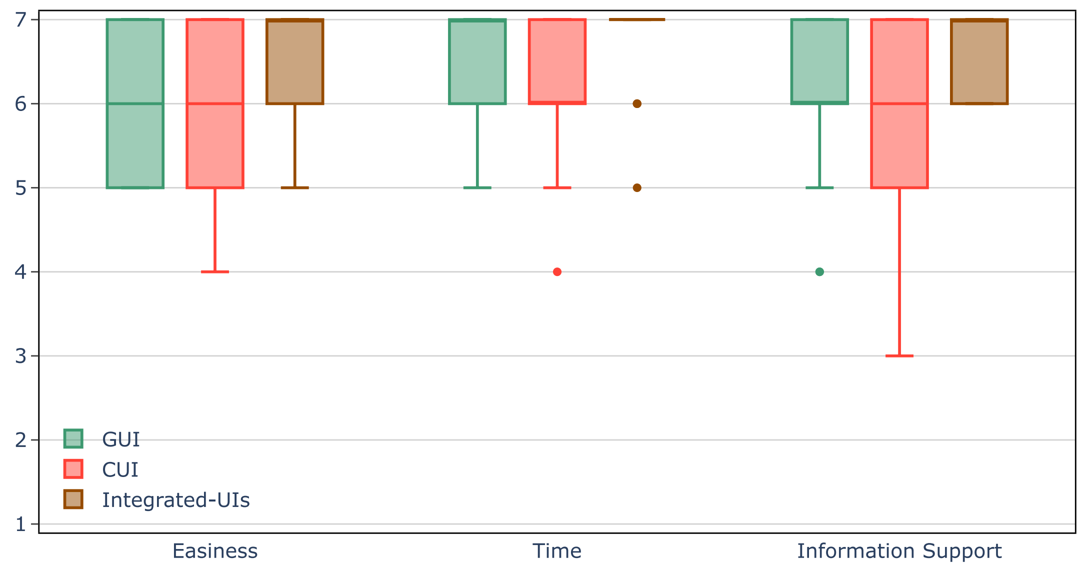


Fig. 9. The perceived satisfaction ratings of the Post-Task Usability Questionnaire (higher is better, 7=extremely satisfied, 1=extremely unsatisfied). The box represents interquartile range (IQR) with the median line indicating the middle value. Whiskers extend to the non-outlier range, while dots represent outliers, indicating data points significantly higher or lower than the rest. The ‘flat’ line in the Time for Integrated-UIs suggests a limited degree of variability, indicating a tight clustering of the majority of scores around a common value.

5.5 Qualitative Feedback

5.5.1 Debrief Session. In the following subsections, we outline the key themes that have been derived from the feedback and explanations provided by the study participants.

Technology Adaption. In this investigation, it was revealed that the participants lacked experience in using programming languages aside from R, which they only used for data analysis. Despite self-assessing their technical abilities as high, they all acknowledged that their use of R was limited to data analytics, with the exception of P02, who held a major in Big Data Biology. P02 indicated that most bioscientists do not frequently use programming languages and that adopting new technologies could be challenging, given the difficulties of using R to execute many of their queries. Although none of the participants had prior knowledge of the semantic web or SPARQL query, they were able to complete all tasks within the given timeframe. All participants reported that they did not need to have an understanding of the data and its underlying technologies in order to extract the data and noted that the tool significantly simplified many of their routine tasks. For instance, P01 mentioned, without prompting, that "It was easy to use compared to a lot of what I have used before." This was also mentioned by P05 and P06.

In contrast, P07 indicated initial confusion with the "Linked Entities" label in the sensor fieldset. This label denotes the entities that are connected to the observations made by the sensor. As the participant explained, the term "entity" is commonly employed to describe individuals rather than the data itself, leading them to suggest replacing it with phrases such as "data entries." This observation underlines the fact that greater levels of abstraction from linked data could potentially enhance user understanding by reducing cognitive load, especially in contexts where terminology may carry different connotations across disciplines.

Collaborative Research. One of the main advantages of using such a tool mentioned by participants (P07, P10, P11, P15, P17) is being able to do better collaborative research. The participants explained that the current datasets are not properly archived, and even public datasets are not queryable and will involve traditional techniques that require downloading the entire dataset. They also show that fast access to the data will provide a faster method for the people in the field to do proper field research. The study participants were encouraged to consider the potential benefits of storing field data as linked data and being granted access to it through the ForestQB interface. Their views were uniformly positive, with the overall feeling that such an approach would address a substantial challenge within their domain.

P07 highlighted collaboration within their field as a "missing link." They explained that tools such as ForestQB could "allow for that collaboration, and it might actually be able to speed the process of research." This opinion was also confirmed by P10, who suggested, "It would probably create a collaborative-like workspace." Likewise, P15 expressed interest in the possibility of conducting a quick search through archived data to identify potentially relevant information, which could contribute to collaborative research efforts between them and the original data gatherers. In contrast, P11 and P17 considered this tool as a means for connecting fieldwork with research endeavours. They believed it would expedite data accessibility within the field for researchers interested in carrying out field investigations in specific locations. P17 argued that this approach could effectively "bridge the gap between on-the-ground conservation efforts, in places like DGFC, and research efforts." This participant outlined that this approach would provide both parties with a straightforward data exploration method.

Productivity. All participants were asked about their experience using the tool and whether the tool changed their productivity. They reported that the tool was able to accomplish tasks similar to those they usually perform; however, it enabled them to do so in seconds rather than hours, as manual work across multiple datasets would require. For instance, in response to a question regarding the comparison between the same tasks being performed using their typical data extraction methods, P12 responded with the following observation: “I can imagine this is like hours and hours.”

As illustrated in Figure 9, the Integrated-UIs appear to receive the highest satisfaction ratings among the evaluated approaches. Participants provided different interpretations to justify this preference, yet in the context of productivity, the consensus pointed towards the convenience of automating tasks through the CUI and the flexibility of customising details with the GUI. For instance, P03 suggested that in a scenario of data exploration, they would be inclined to manipulate the dataset and observe resulting outcomes using the GUI, specifying their potential strategy as “What if I remove this? How it is going to look like.” The participant’s dialogue continued to highlight the perceived limitations of using a chatbot for this particular interaction, asserting, “I don’t think I can probably do that in an easier way when I use the chatbot.” They further indicated that for “a very specific question,” their preference would shift towards the utilisation of the CUI.

Interestingly, without being prompted, both P01 and P05 expressed that their experience with the tool reminded them of many activities they performed during their undergraduate research projects. Furthermore, they believed that the tool would be highly beneficial for undergraduate students as well as post-graduate research projects such as master’s and PhD research. In particular, P05 mentioned that the tool could assist with discovering patterns and determining research questions by exploring the data using the tool’s functionality. In addition, P18 highlighted the issue of human-wildlife conflict, for example, the confrontations between elephants and human communities, affirming its potential to be “the key” to enhancing conservation efforts and minimising these types of conflicts. These observations suggest that the tool has the potential to significantly enhance teaching and research activities for bioscience students and researchers alike.

Gaining Confidence. In the context of comparing the use of conversational UI and GUI alone versus presenting both UIs together, users displayed less anxiety and higher confidence when presented with both UIs. The use of only the conversational UI caused issues related to ambiguity and linguistic variability, and users struggled to frame their sentences correctly when presented with more complex questions. One participant, P02, reported having difficulty performing task T2.4 in the first two attempts and described the experience as “a bit frustrating.” However, P02 and P15 mentioned that if they were using the Integrated-UIs, they would have been able to correct mistakes manually instead of having to retype the entire question. Four participants (P05, P14, P16 and P17) have also expressed satisfaction with the Integrated-UIs and being able to see their typed question reflected on the interface by P05 stating that it “gives me an extra boost of confidence,” while P14 and P16 described it by “reassuring.” Additionally, the participants (P09, P13, P14 and P16) expressed their preference for having the GUI while using the CUI by using the term “to double-check.” P13 stated that they would “want this to double-check what chatbot is coming out of this.” Meanwhile, P09 explained their preference by the need to double-check that “the artificial intelligence is really working,” which would make the CUI experience more trustworthy for the user.

While the conversational UI initially seemed faster and easier to use, the use of GUI became faster once the user became familiar with the location of the necessary interface elements (see Figure 7). Thus, Integrated-UIs resulted in a higher level of confidence and less ambiguity. The given GUI hints were mainly related to the location of the interface elements rather than how to use them. Participants expressed a preference for starting with the conversational UI and then using the GUI to refine their search results further.

Learnability and Usability. In the context of learnability and usability across the three scenarios, all the study participants agreed that the tool did not impose a steep learning curve. Nevertheless, most participants found that integrating the CUI with GUI enhanced the tool’s usability. For instance, P11, who suffers from dyslexia and dyspraxia, described their difficulties in tracking elements on the screen due to their learning disabilities, often requiring an extended duration to comprehend the visual layout. They subsequently expressed their preference for the Integrated-UIs. Their statement, "I just want to ask the question that I need, so that I can at least get a head start or see what's changed or anything like that. It makes it easier for somebody like me," underscored the perceived facilitation the tool offered for their special learning needs.

Six participants (P04, P05, P07, P08, P12, P14) indicated that the presence of the CUI would equip them with a convenient search option, particularly beneficial when they found themselves in a "stuck" situation or encountered a stumbling block, which is also mentioned by P13 by saying "if you're like, not exactly sure." P05 and P14 also contributed to the discussion by suggesting that the chatbot's ability to reflect the user's query directly onto the GUI could assist in understanding the system, thereby serving as a learning resource as well.

5.6 Discussion

When comparing the three methods for accessing the data, it was revealed that the integrated user interface was the preferred choice for all participants. The reason for this preference was that the Integrated-UIs significantly improved the accessibility of the data. This conclusion was drawn by analysing both the user feedback obtained during the debrief session, as well as the time taken by the users to complete the task. In the following discussion, we will delve into the advantages of using ForestQB. This will be preceded by a discussion of the limitations inherent to its use and a look into future improvements and research directions.

5.6.1 Integration to Build User Trust. The use of natural language for formulating queries has consistently demonstrated its simplicity and speed in many application usages [1, 59]. However, the inherent linguistic variability and ambiguity in this method pose challenges, such as validating the accuracy of the produced query and generating more sophisticated, nuanced queries [56, 57, 59, 63]. In contrast, the employment of conventional GUIs offers more consistency across a broader spectrum of users and affords greater expressiveness when formulating queries [43]. Hence, the integration of these two user interfaces enhances the pace of the conventional GUI whilst augmenting user assurance with the CUI.

The feedback gathered from the qualitative user study implies a higher inclination to trust the CUI when they are given the opportunity to validate its output. In addition, presenting users with a GUI that visually demonstrates their queries can enhance their understanding of how to interact with the interface effectively, consequently leading to an increased sense of trust. This heightened assurance is likely attributable to the increased sense of control and user interaction when compared to utilising the CUI in isolation. These observations imply that the integration of these interfaces could be an effective strategy for boosting user assurance and engagement.

5.6.2 Interface Automation to Enhance Speed. All study participants highly appreciated the Form-based query builder featured in the ForestQB, recognising it as the central component of this toolkit. When faced with a binary decision to choose between keeping the GUI or the CUI, a significant majority of the participants (16 out of 18) preferred the GUI. This preference was primarily influenced by the enhanced expressiveness that they perceived while using the GUI, along with the ease of modifying their queries. Conversely, the remaining two participants (P17, P18) who favoured the CUI underscored the speed of query generation as their primary rationale.

Despite this divergence in preferences, there was general agreement among all participants regarding the speedier nature of the CUI in generating queries. As such, a large proportion of participants expressed interest in employing the CUI as a preliminary step for formulating their initial question, viewing this as an automation strategy for query creation. As a result, they proposed using the GUI to fine-tune and modify the initial query to better align with their intended inquiry. Moreover, participants demonstrated interest in utilising the CUI as a support resource in scenarios where they encounter difficulties, providing an additional learning benefit. This dual usage of the CUI promotes example-based learning, thereby enhancing their proficiency in navigating the tool.

In addition, as illustrated in Table 5, for the Integrted-UIs, there appears to be a significant improvement in the learning curve among the participants when comparing their initial attempts with their final attempts. As a result, with the progress of time, Integrated-UIs potentially promote a more rapid learning process and progressively augment user productivity.

5.6.3 Order Optimisation in the Integrated-UIs to Eliminate User Disorientation. Within the context of the experiment in the Integrated-UIs session, participants were obliged to execute a series of four tasks, whereby the initial and final tasks were performed using the GUI, while tasks two and three incorporated the CUI. The intended design of this procedure aimed to provide participants with the opportunity to alternate between both interfaces. However, this systematic approach created a level of confusion amongst several participants. While they expressed a strong preference for the overall concept of interchanging interfaces, the prescribed order of tasks led to some dissatisfaction. Their preference lay in initiating the tasks with one interface and subsequently transitioning to the other, as opposed to the constant alternation necessitated by the experiment's design.

As mentioned in the previous subsection, there was an inclination amongst participants towards commencing with the CUI before transitioning to the GUI. However, participants indicated that any subsequent interchange between the two would be driven primarily by any difficulties encountered while using a particular interface, thus viewing the other as an alternative solution. For instance, if a challenge was encountered within the CUI, such as resource selection, the participant would opt to revert to the GUI in order to perform the task manually.

5.6.4 Limitation of ForestQB.

User Study. The rationale behind our approach in conducting the user study, which included a comparison between the integrated tool version and its distinct GUI and CUI components, is rooted in the lack of publicly accessible tools designed to handle this particular type of data. This absence of comparable tools has presented considerable challenges in validating our methodology through a conventional comparative study. As a demonstration, we employed the time required to complete a task as a measure of progression over time, given the obvious lack of comparability of the methods employed within the same tool. Nevertheless, this measure could clearly indicate a tool's efficiency if compared with another tool under identical task conditions.

Expressiveness. The primary focus in our implementation was to perform the four use cases successfully as outlined in the requirements collection, as explained in section 3.4. Thus, the core capabilities of the tool's expressivity are limited by these predefined use cases, lacking certain operations. For example, ForestQB is limited to creating SELECT queries, with some operations currently not supported, such as the AND operation for intersections. As such, to create a geospatial filter aimed at obtaining data from an intersecting zone between two areas, users are required to draw a polygon within the overlapping region. Despite this limitation, no concerns have been raised thus far in the user feedback. Users have shown a level of satisfaction with the current system as a resource to assist in extracting relevant data for their research. However, it is important to note that expanding the operations and enhancing the expressiveness of the tool could potentially bring about additional benefits to some users.

Ontology-Independent. Considering the nature of our present dataset, which lacks comprehensive annotation, our primary objective has been to develop strategies for extracting relevant information using an approach that is not exclusively dependent on a particular ontology. This ontology-independent methodology is crucial for ensuring that we are not restricted by the need for data to comply with specific ontology configurations, thereby granting us greater flexibility and adaptability in processing diverse datasets.

However, it is worth noting that due to our current focus, the capability to extract data that is deeply embedded in the ontology itself, or that which derives from its inherent inferences, is not currently supported. This limitation primarily results from our intentional decision to prioritise the ontology-independent approach over more rigid, ontology-specific methods. We acknowledge this as a potential area for future development.

The design concept behind the tool was to create an adaptable and robust system capable of interfacing seamlessly with any form of observational data if the tool settings have been properly configured to facilitate this interaction. We have conducted trials using data conforming to SOSA and FOO, as detailed in section 4.5.

Entity Extraction. In the current CUI implementation, we exclusively deploy the model to extract entities using an ontology-independent approach. This procedure implies that following the extraction of entities by the model, the execution of all actions occurs within the CUI component, thereby initiating a separate query to determine the validity of these identified resources. Consequently, this approach has confined us to the execution of only the first three use cases. The reason is that the fourth use case contains an indeterminate count of predicates and filters, which, under existing circumstances, remains challenging to identify without incorporating an ontology. Therefore, attempts to discern this using the prevailing method could detrimentally impact the accuracy of the CUI model and prevent proper implementation.

To overcome this restriction in subsequent developments, we propose an augmentation of the data through appropriate annotations, aligning it with one of two potential approaches. The first approach proposes training the model on a specific ontology, inevitably restricting it to the associated dataset, and any modifications therein would necessitate the complete retraining of the model. Alternatively, the second approach advocates for permitting the user to select or upload the ontology, subsequently providing it as input to the model, thereby aiding in the accurate extraction of resources. Regardless of the chosen approach, each will facilitate the successful integration of the fourth use case into the CUI without compromising the model's precision.

Utilising Large Language Models (LLMs). As previously discussed, we have adopted an ontology-independent approach. While this strategy boasts certain benefits, it is important to acknowledge the current constraints of our entity extraction technique, as mentioned above. Given these limitations, employing an LLM was considered a potential solution to enhance our conversational UI. A notable challenge, however, is that such models are primarily geared towards text generation, which hinders their direct application as the core model in our Integrated-UIs. Specifically, the challenge lies in enabling the LLM to manipulate the GUI inputs, such as the addition or modification of filters. Even with the conversion of the generated text for GUI representation, the propensity of the model to produce hallucinations poses a substantial risk of yielding unanticipated outputs, thereby leading to potential errors or incorrect decisions. This led us to the decision to design our own distinct model and backend functionality aimed at efficiently processing user interactions, populating relevant fields, and formulating accurate queries.

Nonetheless, the potential of integrating an LLM with our approach to assist in extracting more valuable information from the ontology remains a possible solution. When provided with the specific ontology in use, these models may exhibit improved accuracy in identifying relevant entities and can potentially improve query generation by analysing all possible relationships. Still, the tendency of these models to produce unexpected results presents a considerable challenge. Thus, processing these results for UI display requires careful evaluation to prevent unpredictable behaviour.

6 CONCLUSION AND FUTURE WORKS

We have introduced ForestQB, an adaptive query builder tool to support biologists in exploring observational linked data. This instrument integrates GUI and conversational UI, referred to as Integrated-UIs, to enhance data extraction processes. The combination of these two interfaces facilitates the merging of their respective strengths, thereby promoting the overall user experience. To measure the effectiveness of our method, we conducted a user study. The results indicate a marked improvement in user satisfaction regarding our innovative approach. Moreover, Integrated-UIs have demonstrated a reduced learning curve and augmented productivity. Therefore, our research advocates the augmentation of the form-based query builder through the use of conversational UI, a strategy that can significantly enhance user productivity in conjunction with satisfaction. We believe this will promote broader adoption of the semantic web across multiple domains.

In light of the limitations mentioned in subsection 5.6.4, our future research plans will be directed towards exploring suitable techniques for implementing more domain-dependent techniques with the intent to enhance the entity extraction capabilities of the tool. Furthermore, we will be investigating strategies to expand the tool's expressiveness. This enhancement is anticipated to enable the tool to respond to more complex queries, thereby increasing its overall functional capability and effectiveness.

ACKNOWLEDGMENTS

We would like to acknowledge the support we received from Advanced Research Computing at Cardiff (ARCCA), Danau Girang Field Centre (DGFC), GCRF Facilitation Funding, and EPSRC Institutional Sponsorship for International Partnerships at Cardiff University. We would also like to acknowledge the scholarship and continuous support provided by Saudi Electronic University.

REFERENCES

[1] Katrin Affolter, Kurt Stockinger, and Abraham Bernstein. 2019. A comparative survey of recent natural language interfaces for databases. *VLDB Journal* 28, 5 (2019), 793–819. https://doi.org/10.1007/s00778-019-00567-8

[2] Oszkár Ambrus, Knud Möller, and Siegfried Handschuh. 2010. Konduit VQB: A visual query builder for SPARQL on the social semantic desktop. *CEUR Workshop Proceedings* 565 (2010).

[3] Marcelo Arenas, Bernardo Cuenca Grau, Evgeny Kharlamov, Sarunas Marciuska, Dmitriy Zheleznyakov, and Ernesto Jimenez-Ruiz. 2014. SemFacet: Semantic Faceted Search over Yago. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14 Companion)*. Association for Computing Machinery, 123–126. https://doi.org/10.1145/2567948.2577011

[4] Judie Attard, Fabrizio Orlandi, and Sören Auer. 2018. ExConQuer: Lowering barriers to RDF and Linked Data re-use. *Semantic Web* 9, 2 (2018), 241–255. https://doi.org/10.3233/SW-170260

[5] Maurizio Atzori, Giuseppe M. Mazzeo, and Carlo Zaniolo. 2019. QA 3: A natural language approach to question answering over RDF data cubes. *Semantic Web* 10, 3 (2019), 587–604. https://doi.org/10.3233/SW-180328

[6] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*. ACM Press / Addison-Wesley. http://www2.dcc.ufmg.br/livros/irbook/

[7] Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *J. Usability Studies* 4, 3 (2009), 114–123.

[8] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An Empirical Evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction* 24, 6 (2008), 574–594. https://doi.org/10.1080/10447310802205776

[9] Guntis Barzdins, Sergejs Rikacovs, and Martins Zviedris. 2009. Graphical query language as SPARQL frontend. In *Local Proceedings of 13th East-European Conference (ADBIS 2009)*. 93–107.

[10] Hannah Bast and Elmar Haussmann. 2015. More accurate question answering on freebase. *International Conference on Information and Knowledge Management, Proceedings* 19-23-Oct- (2015), 1431–1440. https://doi.org/10.1145/2806416.2806472

[11] Tim Berners-lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. 2006. Tabulator: Exploring and Analyzing linked data on the Semantic Web. In *The 3rd International Semantic Web User Interaction Workshop (SWUI06)*. 16.

[12] NIGEL BEVAN and MILES MACLEOD. 1994. Usability measurement in context. *Behaviour & Information Technology* 13, 1-2 (1994), 132–145. https://doi.org/10.1080/01449299408914592

[13] Allison D Binley, Brandon P M Edwards, Gabriel Dansereau, Elly C Knight, and Iman Momeni-Dehaghi. 2023. Minimizing Data Waste: Conservation in the Big Data Era. *The Bulletin of the Ecological Society of America* 104, 2 (2023), e02056. https://doi.org/10.1002/bes2.2056

[14] John Brooke. 1995. SUS: A quick and dirty usability scale. *Usability Eval. Ind.* 189 (1995).

[15] Josep Brunetti, Roberto García, and Sören Auer. 2013. From Overview to Facets and Pivoting for Interactive Exploration of Semantic Web Data. *International journal on Semantic Web and information systems* 9 (2013), 1–20. https://doi.org/10.4018/jswis.2013010101

[16] Grégoire Burel, Amparo E. Cano, and Vitaveska Lanfranchi. 2009. Ozone browser: Augmenting the web with semantic overlays. *CEUR Workshop Proceedings* 449 (2009), 3–4.

[17] Tiziana Catarci. 2000. What happened when database researchers met usability. *Information Systems* 25, 3 (2000), 177–212. https://doi.org/10.1016/S0306-4379(00)00015-6

[18] Karlis Čerans, Lelde Lace, Mikus Grasmanis, and Julija Ovcinnikova. 2022. A UML-Style Visual Query Environment Over DBPedia. In *Metadata and Semantic Research*. Springer International Publishing, 16–27.

[19] D. Challender, D.H.A. Willcox, E. Panjang, N. Lim, H. Nash, S. Heinrich, and J. Chong. 2019. The IUCN Red List of Threatened Species 2019. https://www.iucnredlist.org/species/12763/123584856

[20] Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. 2020. Dataset search: a survey. *VLDB Journal* 29, 1 (2020), 251–272. https://doi.org/10.1007/s00778-019-00564-x

[21] Gong Cheng, Huiyao Wu, Saisai Gong, Weiyi Ge, and Yuzhong Qu. 2010. Falcons Explorer: Tabular and Relational End-user Programming for the Web of Data. *Semantic Web Challenge 2010* c (2010).

[22] Elizabeth M. Daly, Freddy Lecue, and Veli Bicer. 2013. Westland row why so slow? Fusing social media and linked data sources for understanding real-time traffic conditions. *International Conference on Intelligent User Interfaces, Proceedings IUI* (2013), 203–212. https://doi.org/10.1145/2449396.2449423

[23] Alessio De Santo and Adrian Holzer. 2020. Interacting with Linked Data: A Survey from the SIGCHI Perspective. (2020), 1–12. https://doi.org/10.1145/3334480.3382909

[24] Dennis Diefenbach, Vanessa Lopez, Kamal Singh, and Pierre Maret. 2018. Core techniques of question answering systems over knowledge bases: a survey. *Knowledge and Information Systems* 55, 3 (2018), 529–569. https://doi.org/10.1007/s10115-017-1100-y

[25] Ilie Cristian Dorobăţ and Vlad Posea. 2020. onIQ: An Ontology-Independent Natural Language Interface for Building SPARQL Queries. In *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*. 139–144. https://doi.org/10.1109/ICCP51029.2020.9266272

[26] Sébastien Ferré. 2017. Sparklis: An expressive query builder for SPARQL endpoints with guidance in natural language. *Semantic Web* 8, 3 (2017), 405–418. https://doi.org/10.3233/SW-150208

[27] Kathleen Gregory, Paul Groth, Helena Cousijn, Andrea Scharnhorst, and Sally Wyatt. 2019. Searching Data: A Review of Observational Data Retrieval Practices in Selected Disciplines. *Journal of the Association for Information Science and Technology* 70, 5 (2019), 419–432. https://doi.org/10.1002/asi.24165

[28] Florian Haag, Steffen Lohmann, Steffen Bold, and Thomas Ertl. 2014. Visual SPARQL Querying Based on Extended Filter/Flow Graphs. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*. Association for Computing Machinery, 305–312. https://doi.org/10.1145/2598153.2598185

[29] Florian Haag, Steffen Lohmann, and Thomas Ertl. 2014. SparqlFilterFlow: SPARQL Query Composition for Everyone. In *The Semantic Web: ESWC 2014 Satellite Events*. Springer International Publishing, 362–367. https://doi.org/10.1007/978-3-319-11955-7

[30] Florian Haag, Steffen Lohmann, Stephan Siek, and Thomas Ertl. 2015. QueryVOWL: Visual Composition of SPARQL Queries. In *The Semantic Web: ESWC 2015 Satellite Events*, Vol. 9341. Springer International Publishing, 62–66. https://doi.org/10.1007/978-3-319-25639-9

[31] Naeima Hamed, Omer Rana, Pablo Orozco-terWengel, Benoît Goossens, and Charith Perera. 2021. *Open Data Observatories: A Survey.* Technical Report. Cardiff University. https://orca.cardiff.ac.uk/id/eprint/140048/

[32] Philipp Heim and Jürgen Ziegler. 2011. Faceted Visual Exploration of Semantic Data. In *Human Aspects of Visualization*. Springer Berlin Heidelberg, 58–75.

[33] Patrick Hoefler, Michael Granitzer, Eduardo Veas, and Christin Seifert. 2014. Linked data query wizard: A novel interface for accessing sparql endpoints. *CEUR Workshop Proceedings* 1184, January (2014).

[34] Konrad Höffner, Jens Lehmann, and Ricardo Usbeck. 2016. CubeQA—Question Answering on RDF Data Cubes. In *The Semantic Web – ISWC 2016 (Lecture Notes in Computer Science, Vol. 9981)*. Springer International Publishing, 325–340. https://doi.org/10.1007/978-3-319-46523-4

[35] Aidan Hogan. 2020. The Semantic Web: Two decades on. *Semantic Web* 11 (2020), 169–185. https://doi.org/10.3233/SW-190387

[36] Frederik Hogenboom, Viorel Milea, Flavius Frasincar, and Uzay Kaymak. 2010. RDF-GL: A SPARQL-based graphical query language for RDF. *Advanced Information and Knowledge Processing* 53 (2010), 87–116. https://doi.org/10.1007/978-1-84996-074-8

[37] Stian Holmås, Rafel Riudavets Puig, Marcio Luis Acencio, Vladimir Mironov, and Martin Kuiper. 2020. The cytoscape BioGateway app: Explorative network building from an RDF store. *Bioinformatics* 36, 6 (2020), 1966–1967. https://doi.org/10.1093/bioinformatics/btz835

[38] D Huynh and D Karger. 2009. Parallax and Companion: Set-based Browsing for the Data Web. In *IW3C2*.

[39] Krzysztof Janowicz, Armin Haller, Simon J D Cox, Danh Le Phuoc, and Maxime Lefrançois. 2019. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics* 56 (2019), 1–10. https://doi.org/10.1016/j.websem.2018.06.003

[40] Esther Kaufmann, Abraham Bernstein, and Lorenz Fischer. 2007. NLP-Reduce: A "naïve" but Domain-independent Natural Language Interface for Querying Ontologies. *4th European Semantic Web Conference (ESWC 2007)* (2007), 1–2.

[41] Esther Kaufmann, Abraham Bernstein, and Renato Zumstein. 2006. Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. *5th International Semantic Web Conference (ISWC 2006)* November (2006), 980–981.

[42] Georgi Kobilarov and Ian Dickinson. 2008. Humboldt: Exploring linked data. In *CEUR Workshop Proceedings*, Vol. 369.

[43] Emil Kuric, Javier D. Fernández, and Olha Drozd. 2019. Knowledge Graph Exploration: A Usability Evaluation of Query Builders for Laypeople. In *Semantic Systems. The Power of AI and Knowledge Graphs*, Vol. 11702 LNCS. Springer International Publishing, 326–342. https://doi.org/10.1007/978-3-030-33220-4

[44] James R Lewis. 1991. Psychometric Evaluation of an After-Scenario Questionnaire for Computer Usability Studies: The ASQ. *SIGCHI Bull.* 23, 1 (1991), 78–81. https://doi.org/10.1145/122672.122692

[45] Vanessa Lopez, Spyros Kotoulas, Marco Luca Sbodio, and Raymond Lloyd. 2013. Guided Exploration and Integration of Urban Data. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media (HT '13)*. Association for Computing Machinery, 242–247. https://doi.org/10.1145/2481492.2481524

[46] Vanessa Lopz, Martin Stephenson, Spyros Kotoulas, and Pierpaolo Tommasi. 2014. Finding mr and mrs entity in the city of knowledge. *HT 2014 - Proceedings of the 25th ACM Conference on Hypertext and Social Media* (2014), 261–266. https://doi.org/10.1145/2631775.2631817

[47] Omar Mussa, Omer Rana, Benoît Goossens, Pablo Orozco-terWengel, and Charith Perera. 2023. Making linked-data accessible: A review. (2023). https://orca.cardiff.ac.uk/id/eprint/152125/

[48] Peter Neish. 2015. Linked data: what is it and why should you care? *The Australian Library Journal* 64, 1 (2015), 3–10. https://doi.org/10.1080/00049670.2014.974004

[49] André Rocha and Cássio Prazeres. 2017. LDoW-PaN: Linked Data on the Web—Presentation and Navigation. *ACM Trans. Web* 11, 4 (2017). https://doi.org/10.1145/2983643

[50] Alistair Russell and Paul R. Smart. 2008. NITELIGHT: A graphical editor for SPARQL queries. *CEUR Workshop Proceedings* 401 (2008), 2–3.

[51] Alistair Russell, Paul R. Smart, Dave Braines, and Nigel R. Shadbolt. 2008. NITELIGHT: A graphical tool for semantic query construction. *Proceedings of the Fifth International Workshop on Semantic Web User Interaction (SWUI 2008), collocated with CHI 2008, Florence, Italy, April 5, 2008* 543, November (2008).

[52] Jeff Sauro and Joseph S Dumas. 2009. Comparison of Three One-Question, Post-Task Usability Questionnaires. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. Association for Computing Machinery, 1599–1608. https://doi.org/10.1145/1518701.1518946

[53] Walter Sebastian, Unger Christina, Cimiano Philipp, and Bär Daniel. 2012. Evaluation of a Layered Approach to Question Answering over Linked Data. *The Semantic Web – ISWC 2012* 7650, 00 (2012), 362–374. https://doi.org/10.1007/978-3-642-35173-0

[54] Saeedeh Shekarpour and Sören Auer. 2014. SINA: semantic interpretation of user queries for question answering on interlinked data. *ACM SIGWEB Newsletter* Summer (2014), 1–1. https://doi.org/10.1145/2641730.2641733

[55] Paul R Smart, Alistair Russell, Dave Braines, Yannis Kalfoglou, Jie Bao, and Nigel R Shadbolt. 2008. A Visual Approach to Semantic Query Design Using a Web-Based Graphical Query Designer. In *Knowledge Engineering: Practice and Patterns*. Springer Berlin Heidelberg, 275–291. https://doi.org/10.1007/978-3-540-87696-0

[56] Ahmet Soylu, Evgeny Kharlamov, Dmitriy Zheleznyakov, Ernesto Jimenez-Ruiz, Martin Giese, Martin G. Skjæveland, Dag Hovland, Rudolf Schlatte, Sebastian Brandt, Hallstein Lie, and Ian Horrocks. 2018. OptiqueVQS: A visual query system over ontologies for industry. *Semantic Web* 9, 5 (2018), 627–660. https://doi.org/10.3233/sw-180293

[57] Arjun Srinivasan, Mira Dontcheva, Eytan Adar, and Seth Walker. 2019. Discovering natural language commands in multimodal interfaces. *International Conference on Intelligent User Interfaces, Proceedings IUI* Part F1476 (2019), 661–672. https://doi.org/10.1145/3301275.3302292

[58] Steffen Thoma, Andreas Thalhammer, Andreas Harth, and Rudi Studer. 2019. FUSE: Entity-centric data fusion on linked data. *ACM Trans. Web* 13, 2 (2019). https://doi.org/10.1145/3306128

[59] Christina Unger, André Freitas, and Philipp Cimiano. 2014. An Introduction to Question Answering over Linked Data. In *Reasoning Web. Reasoning on the Web in the Big Data Era: 10th International Summer School 2014, Athens, Greece, September 8-13, 2014. Proceedings*. Springer International Publishing, 100–140. https://doi.org/10.1007/978-3-319-10587-1

[60] Hernán Vargas, Carlos Buil-Aranda, Aidan Hogan, and Claudia Lópes. 2019. RDF Explorer: A Visual SPARQL Query Builder. In *The Semantic Web – ISWC 2019*, Vol. 11778 LNCS. Springer International Publishing, 647–663. https://doi.org/10.1007/978-3-030-30793-6

[61] Paul Warren and Paul Mulholland. 2020. A Comparison of the Cognitive Difficulties Posed by SPARQL Query Constructs. In *Knowledge Engineering and Knowledge Management*, Vol. 12387 LNAI. Springer International Publishing, 3–19. https://doi.org/10.1007/978-3-030-61244-3

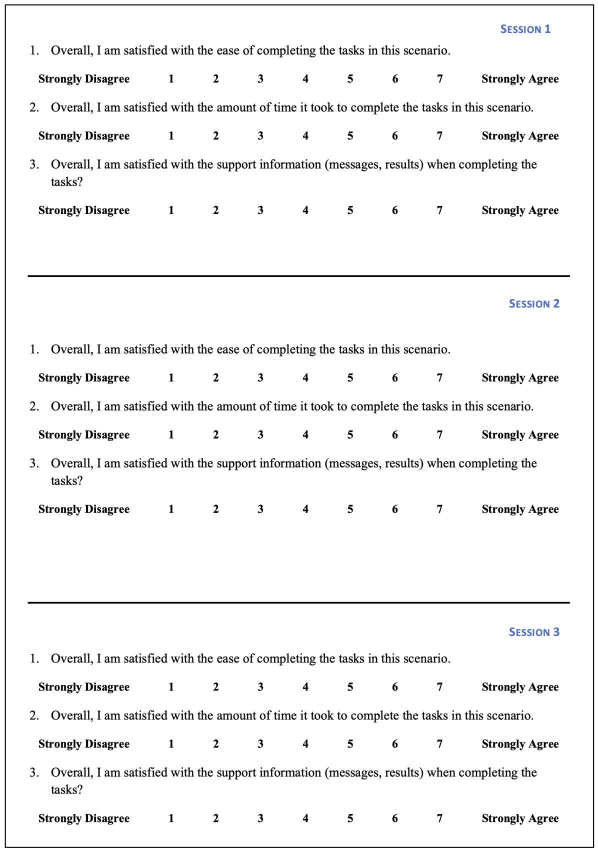
[62] Gideon Zenz, Xuan Zhou, Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. 2009. From keywords to semantic queries—Incremental query construction on the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web* 7, 3 (2009), 166–176. https://doi.org/10.1016/j.websem.2009.07.005

[63] Weiguo Zheng, Hong Cheng, Lei Zou, Jeffrey Xu Yu, and Kangfei Zhao. 2017. Natural language question/answering: Let users talk with the knowledge graph. *International Conference on Information and Knowledge Management, Proceedings* Part F1318 (2017), 217–226. https://doi.org/10.1145/3132847.3132977

[64] Martins Zviedris and Guntis Barzdins. 2011. ViziQuer: A Tool to Explore and Query SPARQL Endpoints. In *The Semanic Web: Research and Applications*. Springer Berlin Heidelberg, 441–445. https://doi.org/10.1007/978-3-642-21064-8

A APPENDIX

Figure 10 presents the form employed in the Post-Task Usability Questionnaire, which is based on a questionnaire consisting of three questions that have been adapted from [44, 52]. These three questions are carefully designed to measure user satisfaction in terms of time, ease of use, and the level of information support. In addition, figure 11 shows the widely recognised System Usability Scale (SUS) Questionnaire, which was employed to assess the usability of ForestQB. Participants were asked to rate a series of 10 statements on a Likert scale that spans from "strongly disagree" to "strongly agree".



SESSION 1

1. Overall, I am satisfied with the ease of completing the tasks in this scenario.

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

2. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario.

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

3. Overall, I am satisfied with the support information (messages, results) when completing the tasks?

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

SESSION 2

1. Overall, I am satisfied with the ease of completing the tasks in this scenario.

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

2. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario.

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

3. Overall, I am satisfied with the support information (messages, results) when completing the tasks?

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

SESSION 3

1. Overall, I am satisfied with the ease of completing the tasks in this scenario.

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

2. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario.

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

3. Overall, I am satisfied with the support information (messages, results) when completing the tasks?

Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree

Fig. 10. Post-Task Usability Questionnaire.

System Usability Scale (SUS)

This is a standard questionnaire that measures the overall usability of a system. Please select the answer that best expresses how you feel about each statement.

	Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
1. I think I would like to use this tool frequently.	☐	☐	☐	☐	☐
2. I found the tool unnecessarily complex.	☐	☐	☐	☐	☐
3. I thought the tool was easy to use.	☐	☐	☐	☐	☐
4. I think that I would need the support of a technical person to be able to use this system.	☐	☐	☐	☐	☐
5. I found the various functions in this tool were well integrated.	☐	☐	☐	☐	☐
6. I thought there was too much inconsistency in this tool.	☐	☐	☐	☐	☐
7. I would imagine that most people would learn to use this tool very quickly.	☐	☐	☐	☐	☐
8. I found the tool very cumbersome to use.	☐	☐	☐	☐	☐
9. I felt very confident using the tool.	☐	☐	☐	☐	☐
10. I needed to learn a lot of things before I could get going with this tool.	☐	☐	☐	☐	☐

How likely are you to recommend this tool to others? (please circle your answer)

Not at all likely 0 1 2 3 4 5 6 7 8 9 10 Extremely likely

Fig. 11. The System Usability Scale (SUS) Questionnaire utilised to evaluate the usability of ForestQB.