Evaluating the Knowledge Graph Editor of the Virtual Record Treasury of Ireland

Alex Randles[1*], Lucy McKenna[1], Lynn Kilgallon[2], Beyza Yaman[1], Peter Crooks[2] and Declan O'Sullivan[1]

[1] *ADAPT Centre for Digital Content, Trinity College Dublin, Ireland*
[2] *Department of History, Trinity College Dublin, Ireland*

Abstract
The Virtual Record Treasury of Ireland (VRTI) is a digital reconstruction of historical records damaged in the Irish Civil War. The reconstructed items are represented as a Knowledge Graph (KG). Since many subject matter experts (e.g. historians), lack the knowledge engineering skills to interact with and improve the KG, a web-based application was developed. The developed application facilitates easy searching, editing, and creation of resources within the VRTI-KG, eliminating the need for the user to know complex SPARQL queries or SHACL constraints. The approach uses data validation involving SHACL constraints to detect inaccurate information in edits. The approach is highly configurable, allowing synchronization with data model changes within the VRTI, and potentially directly deployed for other KGs in other topic areas in the future. In addition, a user evaluation involving 9 experts assessed user satisfaction, understanding, accuracy, and efficiency in the use of their editor. Results showed that most participants were satisfied and could efficiently complete tasks, though some areas for improvement were identified. The evaluation's methodology and insights could benefit other researchers in their design of similar linked data application and enhancing the interaction with respective end users.

Keywords
Digital Humanities, KG Search and Edit, KG Interface, User Testing

1. Introduction

The **Virtual Record Treasury of Ireland (VRTI)**[1] [1–4] is the result of a seven-year programme of State-funded research hosted at Trinity College Dublin. The VRTI digitally recreates the archival collections which were lost in the destruction of the Public Record Office of Ireland (PROI) at the opening of the Irish Civil War in June 1922 [1–3]. The PROI was Ireland's state archive, which was located within the Four Courts complex in Dublin, on the

VOILA 2024: The 9th International Workshop on the Visualization and Interaction for Ontologies, Linked Data and Knowledge Graphs co-located with the 23rd International Semantic Web Conference (ISWC 2024), Baltimore, USA, November 11-15, 2024.
* Corresponding author.
alex.randles@adaptcentre.ie (A. Randles); lucy.mckenna@adaptcentre.ie (L. McKenna); kilgall@tcd.ie (L. Kilgallon); beyza.yaman@adaptcentre.ie (B. Yaman); pcrooks@tcd.ie (P. Crooks); declan.osullivan@adaptcentre.ie (D. O'Sullivan)
0000-0001-6231-3801 (A. Randles); 0000-0002-6035-7656 (L. McKenna); 0000-0002-3075-8571 (L. Kilgallon); 0000-0003-2130-0312 (B. Yaman); 0000-0001-6782-044X (P. Crooks); 0000-0003-1090-3548 (D. O'Sullivan)

[1] https://virtualtreasury.ie/

banks of the River Liffey and was established in 1867 to bring together all of Ireland's official records and state papers - the oldest dating back more than 700 years - into one central archive. Rising tensions between the National Army forces of the new Irish Free State and Anti-Treaty republican saw the occupation of the Four Courts in Dublin in Easter 1922, and tensions erupted into open conflict when the Battle of the Four Courts commenced on 28 June 1922. On the third day of the war, on 30 June, an explosion shattered the eastern wall of the Record Treasury of the PROI [1–3]. The ensuing fire spread to the paper and parchment records held in the PROI, destroying almost all of the records it contained, and scattering the ashes of seven centuries of Ireland's historical records across the city. A century later, efforts began to reconstruct virtually Ireland's lost archive with the creation of the Beyond 2022 project and saw the launch of the VRTI as a digital resource on 30 June 2022.

A key component of the VRTI's digital offering is the **Knowledge Graph (KG)** for Irish History. It was decided to use a Knowledge Graph to represent the information to support the integration of heterogeneous data and facilitate interlinking and reasoning of that data. The VRTI-KG [3] contains knowledge on noteworthy People, Places, Roles, Organisations, and Interests and their interconnections, from Ireland's lost history, which was discovered from examination of the recreated archives. The information in the **VRTI-KG** was gathered from both subject matter experts (historians reviewing artefacts) as well as automated techniques (named entity recognition in digitized documents). An interdisciplinary approach to the data curation using KG technologies has been followed [3]. Named graphs are used to represent different interpretations of historical representations. From the outset, the project has been committed to using open source and open standards-based technologies. Consequently, the VRTI-KG employs the World Wide Web Consortium (W3C) Resource Description Framework (RDF) [5] to implement the KG. The Beyond 2022 ontology[2] was created to model domain specific concepts and relationships in the KG. The ontology was constructed by extending the CIDOC-CRM [6] ontology. CIDOC-CRM was designed to document cultural heritage and museums and provides a sufficient base for the representation of the KG and associated named graphs. Various tools for knowledge engineers are available[3].

SPARQL [7] is a W3C recommendation RDF query language which allows RDF data to be retrieved and changed. However, creating queries requires a high level of relevant technical knowledge which limits the straightforward consumption of the data. Furthermore, manual creation of queries increases workload due to the time-consuming process. Moreover, updating the data to ensure the quality of represented knowledge requires complex update queries to be created which limits the ingestion of relevant information by subject matter experts.

Evaluating the usability of an approach provides a method to promote collaboration between computer scientists who are developing tools for subject-matter experts [8]. Subject-matter experts in this context are researchers involved in digital humanities. It is important to validate approaches with respective end users to determine whether a sufficient level of usability is provided by the application. In addition, usability testing allows for requirements to be identified, validated and refined. A benefit of using standardized metrics [9] in an evaluation is that it provides a measurement for clear communication and comparison of scientific data.

---

[2] https://ont.virtualtreasury.ie/ontology/

[3] https://virtualtreasury.ie/knowledge-graph

This paper presents an initial user evaluation of the **VRTI-KG Editor**, a web application designed to enable subject-matter experts, to access and refine pertinent data (including geospatial information) within a KG. The editor was designed as a configurable framework allowing the synchronization of the editing views with changes in the data model. In addition, it is hoped the configurability of the editor will allow it to be used within other KG projects. The participants in the evaluation were asked to complete a number of tasks which mimic expected user interaction by subject matter experts. A number of metrics were used to measure the user satisfaction with and user understanding of the application, together with metrics to measure accuracy and efficiency in use of the editor. The qualitative data collected through open comments was analysed to identify common patterns which influence usability. The findings of the evaluation will be used in the next phase to inform an iteration of refinement which is hoped to further improve usability of the editor.

This paper is structured as follows: Section 2 presents background information and related work relevant to the design and implementation of the VRTI-KG Editor. Section 3 provides an overview of the VRTI-KG Editor itself. Section 4 presents the structure of the user evaluation completed on the editor and findings discovered from the results. Section 5 outlines future work and concludes the paper.

2. Related Work

The related work discussed in this section includes approaches designed to support users in exploring and investigating RDF data using a user interface, in particular visualisations.

A survey [10] in 2023 was completed of 28 tools which were designed for the visualization and exploration of KGs. The tools were tagged for each respective reference context: digital libraries or generic databases and were classified according to the type of interaction paradigm used, the type of information displayed and the strategy used to reduce the displayed information. The interfaces were categorised using four categories which include 1) relates to the interaction paradigm used, which can be a tabular visualization, node-link visualization or visual query composition 2) relates to the type of information to be displayed 3) category relates to the strategies used to reduce the displayed information: navigational visualization, incremental visualization or summarized visualization and 4) relates to the interfaces that deals with digital library contents. The survey concludes that few of the existing interfaces combine different interaction paradigms. Information reduction strategies are present in most tools and are crucial for exploring complex KGs. However, the challenge of support for users with limited relevant technical background still exists.

Sampo-UI [11] is a tool designed to allow users to create configurable GUIs for respective KGs. The tool is based on the "SAMPO" model which aims to provide diverse perspectives of RDF data stored in respective graphs. The tool includes faceted search which allows searching through data analytics represented in tabular format and visualisations including geospatial. The tool has been adopted most often by the cultural heritage domain. While the tool is used by a large number of users, several limitations were identified. For example, Sampo-UI has limited support for the geospatial mapping of polygons required to represent area boundaries. In addition, named graphs are currently not supported, which is a requirement for certain KGs, such as the VRTI-KG.

VisKonnect [12] is an approach designed to present visual connections between historical figures and potential interlinks between them. The interlinking of these figures focuses on related events such as sport tournaments, award ceremonies, or summit meetings. VisKonnect models data using the EventKG, an event-centric KG that covers relations between events and persons and allows inferring of required information. The approach provides three visualisations which include event timeline, an event map, and a relationship graph. In addition, a chat interface is used to query the data using natural language to allow users to easily answer questions. The multiple visualisations demonstrate how the can aid searching by users. In particular, the event map inspired the navigation on the geospatial maps in the VRTI-KG editor.

LodView [13] is a tool designed to allow browsing of the RDF resources stored in a dataset using SPARQL [7] queries. A HTML representation of the links of each resource is shown to users when a URI is selected. The representation includes a table which lists the properties and objects associated with the selected resource. The table allows users to follow links in the table to other related resources. The tool includes a configuration file to customize elements such as the SPARQL endpoint, namespace of resources and styling of the interface. The configuration file provided inspiration for use of one in the editor. However, LodView does not support searching for resources other than directly inputting a specific URI and editing of data.

WissKI[4] (Wissenschaftliche Kommunikationsinfrastruktur) is an interface tool designed for managing scholarly KGs. The tool is integrated with an existing tool named Drupal[5] which is a web content management system. It leverages the structure of scholarly ontologies, such as CIDOC CRM [6] to create the data model used by the tool to facilitate editing. WissKI's Pathbuilder supports the configuration of the tool for a specific KG. However, WissKI does not include a data creation process that completes data validation steps to prevent poor quality data being inserted into the KG. In addition, the tool requires a high-level of knowledge of the RDF ontology terms, which inhibits non-technical users. The editor we propose includes a data creation pipeline (Section 3.2) to facilitate straightforward detection of potential quality issues. In addition, the information presented on the interface (Section 3.1) is simplified to hide the underlying complexities of the RDF terms to support non-technical user's.

Approaches surveyed provided useful insights which helped to inform design decisions for the VRTI-KG editor. In particular, Sampo-UI [9] provided inspiration as it uses different search paradigms: free-text search, faceted, geospatial, and temporal similar to the proposed editor. However, it was discovered that few undertook a formal user evaluation.

3. VRTI-KG Editor

The VRTI-KG Editor[6] [4] was designed to allow subject matter experts with no relevant KG technical background to edit and insert data into the VRTI-KG. The user requirements for the editor were gathered by conducting several interviews with digital humanities researchers. An iterative process of prototyping was completed where the requirements were defined, validated and refined based on the feedback of the interviews. The web-based editor allows users to search the graph to find desired resources and displays the current data related to the selected resource.

---

[4] https://wiss-ki.eu/
[5] https://www.drupal.org/
[6] Screencast of editor at https://drive.google.com/file/d/1NI2_0hs4g4Mo5ZInVlOQrbsyZvsGLGYN

The graph editor was implemented using several Python libraries[7]. Flask was used to create the web application and provides a comprehensive and customizable library for constructing web applications and allows diverse Python libraries to be used. In the editor, a number of such Python libraries are used. The SPARQLWrapper library facilitates the execution of SPARQL [7] queries on remote endpoints. Folium is used to visualize geospatial data on geographical maps. Queries are manipulated using string formatting methods available in Python. PySHACL is used to execute Shapes Constraint Language (SHACL) [14] constraints. SHACL is a W3C recommendation which defines the SHACL Shapes Constraint Language, a language for validating RDF graphs against a set of conditions. These conditions are provided as shapes and other constructs expressed in the form of an RDF graph. In this context, RDF graphs are named "shapes graphs" in SHACL and the RDF graphs that are validated against specified shapes are called "data graphs".

3.1. Design and Implementation

This section illustrates the use of the VRTI-KG editor web application through the searching and editing of a person entity resource. The selected person used for illustration is a prominent Irish writer named "Samuel Beckett"[8]. The SPARQL queries created by the editor during the use case are available[9]. First, the person editing option is selected. A person can be retrieved by name, gender, ID, birth/death date, birth/death place, among other using the page shown in **Figure 1**.

In our example, the string "Samuel Beckett" was entered into the text area associated with the **full name search filter**. Thereafter, the string is inserted into a FILTER keyword condition in a SPARQL [7] query `(FILTER (CONTAINS(LCASE(STR(?Name)), LCASE('Samuel')) && CONTAINS(LCASE(STR(?Name)), LCASE('Beckett')`)). The name is split into multiple words to ensure that matches are found regardless of the ordering of the input. The results of the query are shown in a tabular format which allows users to easily retrieve the respective resource. In addition, the table can be searched (**A** – **Figure 1**) when there are larger query results retrieved by the initial search. As can be seen in Figure 1, only one resource is returned for the completed search in our example. Thus, the user then selects the edit button (**B**) for Beckett which results in redirection to the page detailing associated data related to Beckett. Another more specific query is used to retrieve all information in the graph related to Beckett including direct and inverse relations.

[7] https://github.com/alex-randles/Editor-Evaluation/blob/main/libraries.pdf
[8] https://kb.virtualtreasury.ie/person/Beckett_Samuel-Barclay_c20_dib_a0533
[9] https://github.com/alex-randles/Editor-Evaluation/blob/main/sample-queries.pdf

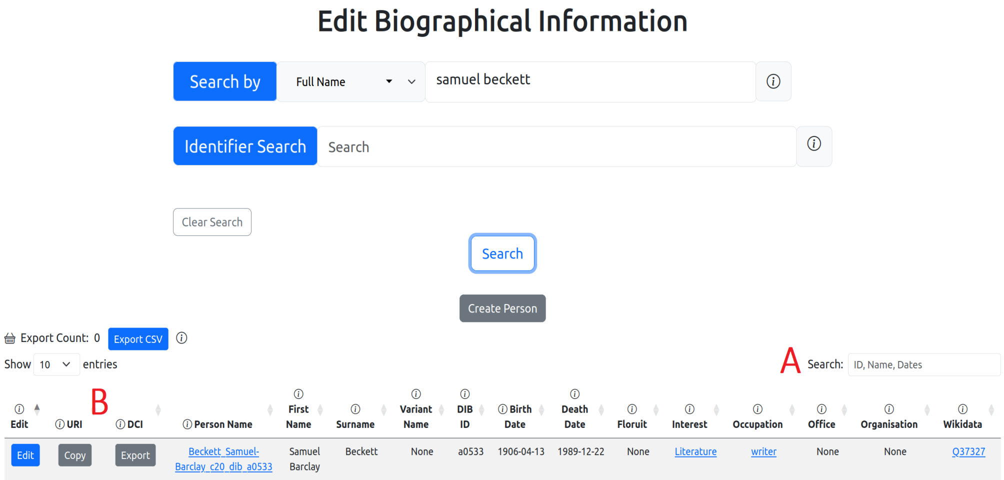

Figure 1: Screenshot of Search Results of **VRTI-KG Editor** for "Samuel Beckett"

The retrieved information is displayed in a tabular format as shown in **Figure 2**.

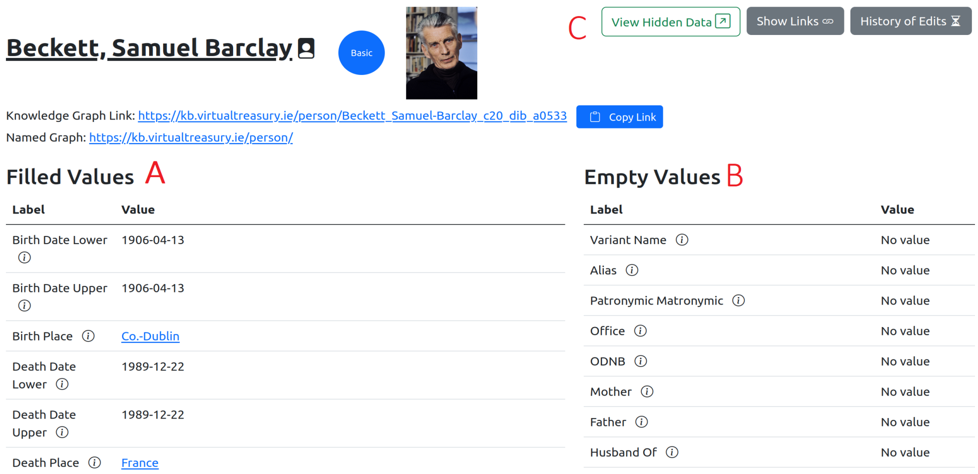

Figure 2: Screenshot of **VRTI-KG Editor** describing "Samuel Beckett"

All URIs returned from the query results are formatted in a more human readable format which involves stripping to the end of the URI. In addition, they are converted into internal links allowing users to further traverse the graph using the application itself. "Filled Values" (**A** – **Figure 2**) relate to attributes which have values already associated, while "Empty Values" (**B**) relate to attributes which do not contain values as yet. The "View Hidden Data" button (**C**) allows users to view data stored in a private experimental graph. The hidden data includes information which is currently being curated in order to improve quality for publication. The data is retrieved using a SPARQL query executed on a separate namespace graph and displayed in tabular format. The "Show Links" button relates to other associated links, such as those to other data stored in the VRTI. The links are categorised as internal (VRTI link) and external (outside VRTI). The "Copy Link" button allows users to easily copy and paste the resource URI into their desired location. The "History of Edits" button redirects to a page which displays provenance of changes which have been executed on the respective resource. The provenance includes the name of the user, time of edit, number of edits and details of the edited values. The provenance data is initially stored in a database which is uplifted using R2RML (RDB to RDF Mapping Language) [15] into a private provenance sub-graph in the VRTI-KG. Thereafter, users can select to edit the resource which results in redirection to the view presented in **Figure 3**.

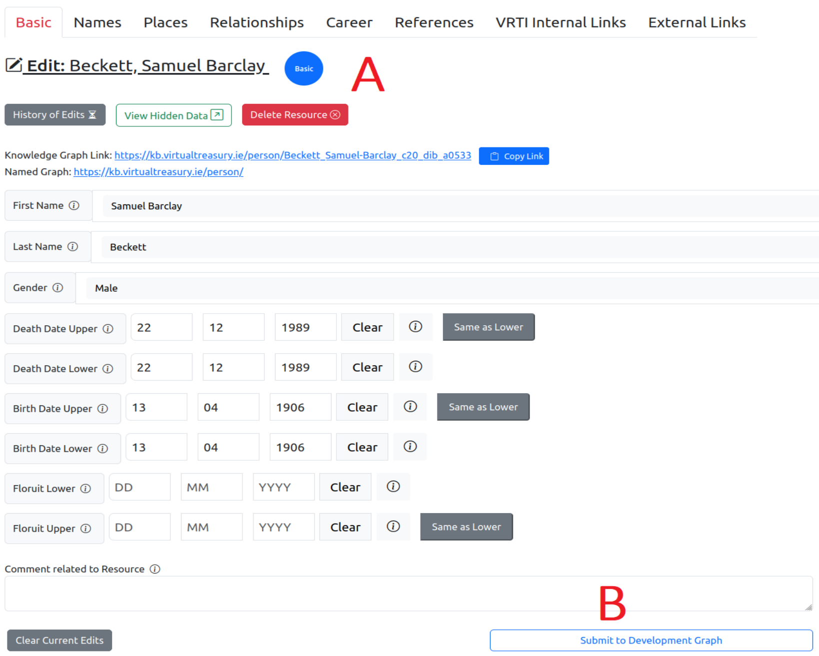

Figure 3: Screenshot of Edit Page for "Samuel Beckett" in **VRTI-KG Editor**

The headings shown (**A** – **Figure 3**) at the top of the page are designed to group attributes based on similarities in order to improve navigation for the user. For instance, "Places" relate to the birth and death of a person, while "Relationships" relate to family and marriages. As can be seen in the example, Beckett has a basic level of completeness (represented in blue) and additional information could be added. Once the necessary changes are completed, validated (see earlier) and submitted (**B**), the updates will be propagated into the VRTI-KG and in the view presented in **Figure 2**.

3.1.1. Configurability of Editor

The editor includes several configuration files represented in JSON format, making it versatile for diverse KG deployments. These files are responsible for customising the editing views (see **Figure 3**) including the dropdowns, styling and SPARQL query creation. **Listing** 1 presents an extract of a configuration file for the occupation of a person in the VRTI-KG.

The extract shows the occupation of a person ("Occupation"). The property ("property_uri") used to link an occupation resource to a person. The column ("column_names") of the CSV file where an occupation name is inserted when a new resource is created. The type ("type_uri") of the resource to be shown in the dropdown options when an occupation is being edited. The name of the tab ("tab_name") where occupations are shown on the editing page. The structure of the SPARQL query ("inverse_relation") and whether an occupation can be removed ("removable") from a person. Some attributes such as a gender can only be changed and not removed.

```
"Occupation": {
    "property_uri": [crm.P107_has_current_or_former_member],
    "column_names": ["STATUSOCCUPATION"],
    "type_uri": vrti.Occupation,
    "removable": True,
    "inverse_relation": True,
    "tab_name": "Career",
},
```

Listing 1: Extract of a configuration file

The file contains similar configurations for the attributes related to people, places, organisations and offices in the VRTI-KG. The configuration can be easily changed when the data model changes, which will automatically propagate the changes into the interface of the editor. Styling can also be changed from the configuration files. Bootstrap[10] CSS classes are used for configurable styling. For instance, the maintainer wants to change the colour of the blue buttons (**Figure 1**) to another colour, such as red. They would find the class for the colour red which is "danger" and replace it in the configuration which will automatically update the interface.

3.2. Creation of New Resources

The previous method to ingest data into the VRTI-KG described in [3] involves the creation of resources using only CSV file upload. The graph editor method is designed to supplement that process by providing an intuitive interface to subject-matter experts, and which completes quality checks to ensure that incomplete data is not inserted into the VRTI-KG. New resources can be created and inserted into the KG using the editor interface alone. The process involves users interacting with a form on the interface. The form allows users to input respective values using various types (e.g. dropdown and free text) of input. The form includes validation which helps to prevent incorrect data being created. For instance, regular expressions are used to ensure that Wikidata links match the expected format (e.g. URI starts with respective host name). The resulting RDF graphs are compared against a set of conditions to capture any other quality issues. In addition, attributes of created resources are compared with existing resources to ensure that no duplicates exist. The steps in the creation pipeline[11] can be summarized as follows:

User Input. First, users input information into a form displayed on the application[12]. The form includes a combination of free text input and selection from restricted dropdown menus. Validation is completed on the input to help prevent incorrect data being inserted. The validation provides visual cues and popups to indicate incorrect values. The form once valid and submitted will result in the generation of temporary source data represented as relational data. The relational data is uplifted into RDF format using R2RML. The respective R2RML

[10] https://getbootstrap.com/docs/5.3/getting-started/introduction/
[11] https://github.com/alex-randles/Editor-Evaluation/blob/main/creation-pipeline.png
[12] https://github.com/alex-randles/Editor-Evaluation/blob/main/creation-form.png

mapping is retrieved and executed which results in an RDF file containing mapped data from the input of users.

SHACL Validation. SHACL [14] shapes have been created for the entities created by the graph editor in order to ensure that resulting resources are sufficient quality. Graphs which do not satisfy these conditions result in debugging information being shown to users and the creation discontinued. The information shown is derived by querying (using SPARQL [7]) the SHACL quality report defined in RDF format using the SHACL Validation Report Vocabulary.

Duplicate Detection. Duplicate detection involves querying existing data in the graph to ensure that no other resources exist which represent the same entity. For instance, attributes such as first name, surname and century are compared to existing data when a person entity is created. The comparison is completed using SPARQL [7] query templates where respective values from the user's input are inserted. A listing of potential duplicates (if any found) is presented to the users. The listing includes other details (such as the associated dates) related to each duplicate to allow them to filter the results. In addition, each duplicate contains an internal link to allow them to instead edit a discovered duplicate.

Finally, the generated RDF data is inserted into the VRTI-KG when the SHACL shapes conform to the data graph and the users confirm that no duplicates exist. A SPARQL Load query is used to insert the generated data into the graph. Thereafter, the updates are propagated into the view of the interface to allow users to search and edit the created resource(s).

4. Evaluation of VRTI-KG Editor

The hypothesis for this study was: *The application provides a sufficient level of **understanding**, **accuracy**, **efficiency** and **satisfaction** during the experiment.*

4.1. Methodology

A user evaluation[13] was conducted to investigate the hypothesis. The participants were provided with a list of tasks to complete using the editor. Then, they were asked to fill out questionnaires detailing their user experience during the experiment. In addition, separate namespaces were setup in the Blazegraph triplestore[14] to allow each participant to independently edit a copy of the VRTI-KG. Each namespace was analyzed after the study to calculate the accuracy of each participant's edits. The following four instruments were used during the experiment to measure user's interaction with the editor.

Post Study Usability Questionnaire (PSSUQ). The PSSUQ [9] is a standardized questionnaire, which was designed by IBM in 1995 to determine the level of **satisfaction** provided by a software system. The questionnaire includes a Likert scale from 1 (best case) to 7 (worst case) to rate the level of satisfaction. In addition, an open-comment section is included with each question. The questionnaire includes four sub-scales which relate to different aspects of the interaction. The four sub-scales are system usefulness, information quality, interface quality and overall. It was decided to use the PSSUQ rather than the System Usability Scale (SUS) [16] due to the extensive psychometric evaluation that has completed with it. In addition,

[13] https://github.com/alex-randles/Editor-Evaluation
[14] https://github.com/blazegraph/database/wiki/REST_API

it includes open-comment sections to capture diverse feedback, which allows participants to elaborate on their user experience.

Understanding Questionnaire. The understanding questionnaire[15] was designed to measure the user's **understanding** of information provided by the editor during the course of the experiment. The questions included a combination of free text and multiple choice. Each question was accompanied by an optional open comment section to allow participants to elaborate further on their answer. The questions asked for information related to different search methods which included entering the URI of a resource (Q1-2) and selecting a search filter and entering an accompanying value (Q3-7) For instance, task 1 asked participants to complete a search using the URI of a person and to examine the search results. Then, they were asked to provide the birth date of the person in Q1.

Time of Completion. The participants were asked to self-report the total time it took them to complete the experiment. The recorded times were used to measure the **efficiency** of the interaction.

Accuracy of Edits made. SPARQL [7] ASK queries[16] were posed to each participant's associated namespace graph. Each query embodied the change to the graph that would be expected as a result of the correct execution of a task. The ASK queries were used to return a Boolean (true or false) indicating if each edit was completed successfully or not.

4.2. Experiment Execution

The experiment was approved by the TCD Research Ethics Committee. This section discusses participants and tasks they were asked to complete.

Participants. The participants consisted of 9 subject-matter expert historians. The participants were recruited based on an internal meeting with the lead VRTI historian which discussed who satisfied the inclusion/exclusion criteria. Participants signed an informed consent form. The experiment was completed asynchronously with each participant provided with login details and a link to the task sheet and questionnaires. Thereafter, participants could access the editor online whenever they decided to complete the experiment. Participants were not provided with any assistance during the completion of the experiment. In addition, they were not provided with any training for the application prior to the experiment.

Task Sheet. A focus group was conducted with researchers in computer science and digital humanities to determine what tasks should be completed during the experiment. The resulting task sheet[17] included 8 tasks with the final task consisting of 6 sub-tasks. The tasks were designed to mimic expected user interaction of historians with the editor, including search, edit and creation of resources in the VRTI-KG. Tasks 1-2 involved searching for people using a provided URI. Tasks 3-5 involved searching for people using different search filters, such as their area of interest. Tasks 6 and 7 involved changing and adding attributes to an existing person. Task 8 included sub-tasks which involved creating a new person with several attributes, including name, occupation and associated dates and places.

[15] https://github.com/alex-randles/Editor-Evaluation/blob/main/understanding.pdf
[16] https://github.com/alex-randles/Editor-Evaluation/blob/main/experiment-queries.pdf
[17] https://github.com/alex-randles/Editor-Evaluation/blob/main/task-sheet.pdf

4.3. PSSUQ Results

Figure 4 presents a boxplot of the results of the PSSUQ questions and four sub-scales (Section 4.1), which was used to measure user **satisfaction** with the editor.

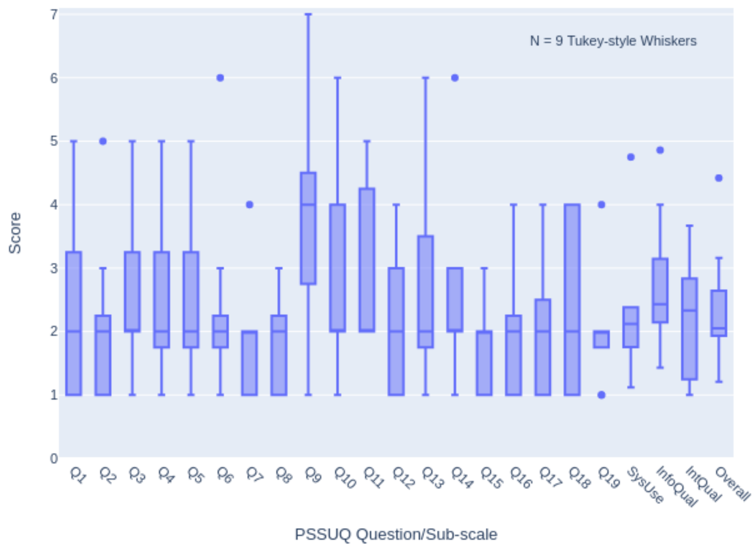

Figure 4: Scores of PSSUQ Questions and Sub-scales

The rectangle of a boxplot represents 50% of the data points. The position of the line in the rectangle represents the distribution of the data points. The line indicates if the data is normally distributed (centre), positive skew (left) and negative skew (right). Outliers are represented by the points outside of the rectangle. All sub-scales scored between 11.3% and 25.2% better than their respective "acceptable thresholds" [9], which indicates overall sufficient satisfaction with the application by users. System usefulness (**SysUse**) scored best with 25.2% better than the normal PSSUQ threshold. Interface quality (**IntQual**) an **Overall** scored similar to their thresholds with 14% and 17.9%, respectively. Information quality (**InfoQual**) scored worst with only 11.3% better than its threshold, which indicates the information provided by the editor should be improved. An example comment which relate to information quality: "Maybe more hover info would be useful for people less familiar with the KG.". Adding more descriptive names to buttons and hover text, and softening technical terminology are hoped to improve the information quality score in the next iteration of the application. In addition, limiting the initial information to prevent overwhelming the users could improve the information quality score, as a participant stated "I think the page for editing an entity has too much when you first land on it". However, comments such as "clean, clear, uncluttered, and no ambiguity", "I found the system to be very easy to use and navigate" and "Yes, the design is clean and consistent" are supportive of an overall conclusion of sufficient user satisfaction.

4.3.1. Understanding Results

Figure 5 presents a bar chart of the scores of how many participants answered correctly each question in the **understanding** questionnaire, used to measure the understanding of each user of the information being presented on the editor interface. 7 questions were included in this questionnaire.

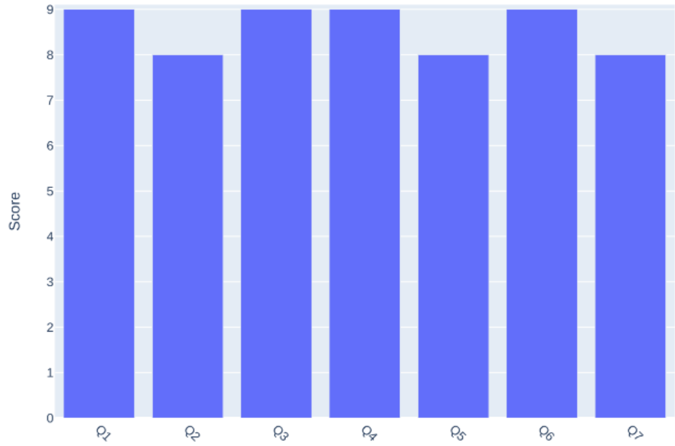

Figure 5: Correct answers in the understanding questionnaire

Mean score for all questions was 94%. 4 out of 7 questions scored 100% correct, which indicates overall sufficient understanding of the information provided by the editor to users. In addition, comments such as "Navigation makes sense and I could find what I needed", "Yes - the design and text fields provided made it easy to navigate and find the information I needed" and "very clear in the novel search" support this observation. The worst scoring questions (Q2 and Q5) related to the occupation of a person shown in the search results table. 2 participants provided the area of interest rather than occupation, which is the column beside in the results table. In addition, 2 participants when answering Q5 provided the URI for the person rather than the named graph as stated in the question. Both are located near each other, which may have caused confusion.

4.3.2. Edit Accuracy Results

Figure 6 presents a bar chart of the **accuracy** (number of correct edits) determined by executing the ASK queries (see section 4.2) upon each participant's namespace graph which resulted from the edits. Task 6 and 7 asked the participants to edit an existing person, while Task 8 asked them to create a new person with distinct attributes (subtasks a-h).

The mean score for all of the edit tasks was 7 (77%) correct edits, indicating that the accuracy of edits by participants was overall sufficient. The edits of existing people (T6 and T7) scored best with a mean of 8 (88%), while the creation of a new person scored worse. However, T8a-T8e scored similar with a mean of 8 (88%) correct edits. These sub-tasks asked participants to add associated names, dates and gender of a person. T8(f)-T8(h) scored worse with a mean of 6 (66%). These sub-tasks asked them to add the occupation, birth and death place. A participant stated that there was no option to insert these attributes and included a comment, "During the last task there was no option to enter the final three pieces of information given.". The inputs for these attributes were only visible when the user selected the respective tab on the page. The task sheet stated, "by navigating through the tabs at the top of the page", however, the task could have explicitly stated which tabs to use for each attribute. 2 participants submitted their edits to the graph before adding all attributes and the second attempt was blocked as the editor does not allow a URI to be created twice (unless deleted first using the editor). A participant stated "I accidentally created the record before I had completed the instructions, and then could not add the final piece of data as I got a 'Duplicate entity' error message.”. This issue could be resolved in future iterations of the editor by adding an explicit warning message to state the URI already exists in the KG and allowing them to recreate in exceptional cases.

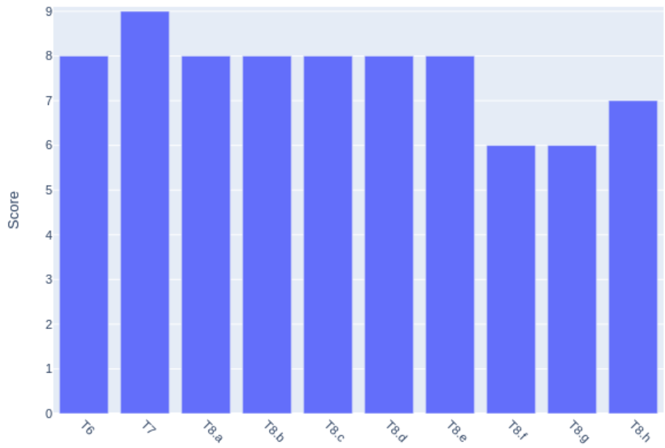

Figure 6: Accuracy of edits completed on KG

4.4. Completion Times Results

Figure 7 presents a boxplot of the overall times for completion of the experiment (self reported by each participant), used as a proxy measure efficiency of user interaction with the editor. The mean time for completion of the experiment was 18 minutes. The minimum time was 10 minutes and maximum time was 25 minutes. The standard deviation was 4.8 which indicates the times were spread around the mean and that efficiency of interaction was not equal for all participants. Some of the Comments which supported sufficient efficiency include, “It was straightforward to learn” and “It was pretty efficient”. The outliers could be as a result of certain participants investigating other aspects of the interface which were not directly involved in the experiment. Furthermore, some participants may have had previous experience using semantic web technologies which resulted in improved efficiency. Moreover, the amount of buttons could have decreased efficiency for certain participants with a participant stating, “The pages are a bit overwhelming with button options”. The Spearman’s Rank-Order coefficient test [17] was used to measure the correlation between efficiency and the other metrics. Spearman’s is a nonparametric test designed to measure correlation between variables and is less sensitive to outliers compared to similar tests. A p-value of 0.05 was applied to the test to indicate a statistically signification correlation. The p-value for efficiency and PSSUQ is -0.033 which is not statistically significant. However, the p-value for time and accuracy is -0.757, which is statistically significant. This value indicates as time decreases, accuracy increases. Thus, participants who completed the experiment faster had better accuracy. This correlation indicates that participants who struggled with the interaction of the application completed worse edits.

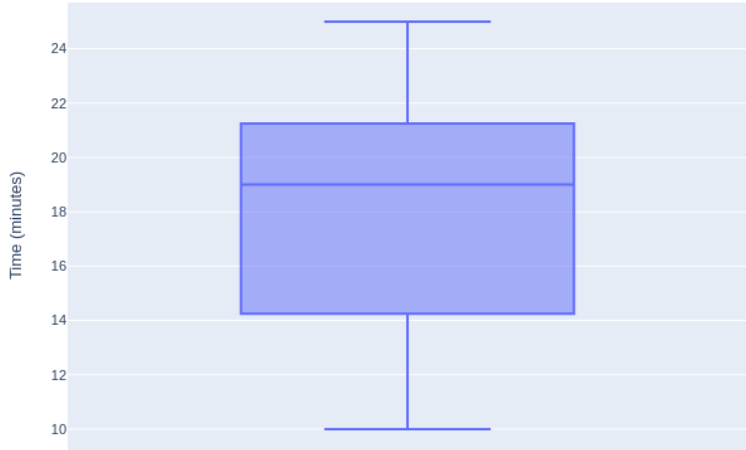

Figure 7: Total time to complete the experiment

4.4.1. Key findings from results

The following key findings have been drawn from the analysis of the experiment results.

Too many buttons. 2 of the participants stated that the information on pages was overwhelming. Comments such as "The pages are a bit overwhelming with button options" and "I think the page for editing an entity has too much when you first land on it" related to this aspect. While the inclusion of additional buttons was hoped to improve functionality, it resulted in confusion during general interaction. It was decided to review buttons and add groupings of existing buttons into dropdown menus in hopes of not overwhelming users.

Re-creation of Resources. 2 participants submitted their created person to the graph before adding all stated attributes with one stating "I accidentally created the record before I had completed the instructions, and then could not add the final piece of data as I got a 'Duplicate entity' error message.". 1 of these participants was the same as mentioned in the last point, stating the page was overwhelming. As this could happen in future, we will consider providing an explicit warning and allowing them to recreate resources in exceptional cases. However, users with more experience would be likely to know that they can access the edit page for the created resource and make changes on that page.

Improvement on existing approach. The previous method to view resources in the VRTI-KG involved dereferencing specific URIs using LodView [13] (Section 2). A participant stated "A big improvement on the LodView" which indicates the editor is possibly an improvement upon LodView for viewing data for the VRTI use case.

Improvement with Experience. 5 of the participants stated that they feel with more experience and guidance the application would become straightforward to use with them stating "Lots of options, easy to navigate, but will be more comfortable once used more frequently.", "I think that some direction would be needed to train users initially but I can easily see myself becoming proficient using this system", "More hand-holding might be useful for initial use" and "The editor looks good and will be straightforward to use with guidance and greater familiarity.".

5. Conclusion

In summary, sufficient **satisfaction** was observed as the scores of each sub-scale of the PSSUQ scored between 11.3% and 25.2% better than their acceptable research thresholds [9]. Sufficient user **understanding** was supported with a mean of 94% correct answers for all questions. Sufficient **efficiency** of interaction with the editor was supported as the mean time was 18 minutes and all participants completed all tasks between 10 and 25 minutes, which involved multiple searches, edits and creation of a new person. Sufficient **accuracy** of the edits was supported as the mean was 77% correct edits. In addition, Spearman's correlation test [17] indicates that as efficiency improves so does accuracy. It is expected that the improvements outlined in Section 4.4.1 could improve these measurements in future versions of the editor. Publishing the implementation will allow other research projects with KGs to configure and use it to search and edit their data. Finally, it is hoped that the evaluation methodology can provide useful insights to researchers when validating similar approaches.

Acknowledgements

Virtual Record Treasury of Ireland (VRTI) is funded by the Government of Ireland, through the Department of Tourism, Culture, Arts, Gaeltacht, Sport and Media, under the Project Ireland 2040 framework. The project is also partially supported by the ADAPT Centre for Digital Content Technology under the SFI Research Centres Programme (Grant 13/RC/2106_P2).

References

[1] B. Yaman, L. McKenna, A. Randles, L. Kilgallon, P. Crooks, D. O'Sullivan, Digital Prosopography Information in Virtual Record Treasury of Ireland Knowledge Graph, in: Proceedings of the 1st International Workshop of Semantic Digital Humanities (SemDH) Co-Located with the 21st Extended Semantic Web Conference, 2024. https://ceur-ws.org/Vol-3724/paper2.pdf.

[2] P. Crooks, Z. Reid, C. Wallace, The Virtual Record Treasury of Ireland: A century of Recovery from the 1922 Four Courts Blaze - and Beyond, Hist Irel 30 (2022) 38–41.

[3] C. Debruyne, G. Munnelly, L. Kilgallon, D. O'Sullivan, P. Crooks, Creating a Knowledge Graph for Ireland's Lost History: Knowledge Engineering and Curation in the Beyond 2022 Project, J. Comput. Cult. Herit. 15 (2022). https://doi.org/10.1145/3474829.

[4] G. Goerz, M. Oischinger, B. Schiemann, An implementation of the CIDOC conceptual reference model (4.2. 4) in OWL-DL, in: Proceedings of the 2008 Annual Conference of CIDOC-The Digital Curation of Cultural Heritage, 2008.

[5] S. Harris, A. Seaborne, E. Prud'hommeaux, SPARQL 1.1 Query Language, World Wide Web Consortium (W3C) Recommendation 21 (2013) 778. https://www.w3.org/TR/sparql11-query/ (accessed April 1, 2023).

[6] E. Ikkala, E. Hyvönen, H. Rantala, M. Koho, Sampo-UI: A full stack JavaScript framework for developing semantic portal user interfaces, Semant Web 13 (2022) 69–84.

[7] A. Randles, D. O'Sullivan, R2 [RML]-ChatGPT Framework, in: Fifth International Workshop on Knowledge Graph Construction@ ESWC2024, 2024: pp. 35–50.

[8] A. Randles, D. O'Sullivan, P. Crooks, L. Kilgallon, L. McKenna, B. Yaman, Facilitating Search of the Virtual Record Treasury of Ireland Knowledge Graph using ChatGPT, in: Proceedings of Poster and Demo Track and Workshop Track of the 20th International Conference on Semantic Systems (SEMANTiCS), 2024.

[9] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, Y. Tang, A brief overview of ChatGPT: The history, status quo and potential future development, IEEE/CAA Journal of Automatica Sinica 10 (2023) 1122–1136.

[10] J.R. Lewis, Psychometric Evaluation of the PSSUQ Using Data from Five Years of Usability Studies, Int J Hum Comput Interact 14 (2002) 463–488. https://doi.org/10.1080/10447318.2002.9669130.

[11] C. Spearman, The proof and measurement of association between two things., Am J Psychol 15 (1904) 72–101. https://doi.org/10.2307/1412159.

[12] E. Bernasconi, C. Miguel, M. Mecella, Linked Data interfaces: a survey, in: 19th Conference on Information and Research Science Connecting to Digital and Library Science, 2023: pp. 1–16.

[13] I. Heibi, S. Peroni, D. Shotton, OSCAR: a customisable tool for free-text search over SPARQL endpoints, in: Semantics, Analytics, Visualization: 3rd International Workshop, SAVE-SD 2017, Perth, Australia, April 3, 2017, and 4th International Workshop, SAVE-SD 2018, Lyon, France, April 24, 2018, Revised Selected Papers 3, 2018: pp. 121–137.

[14] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: International Semantic Web Conference, 2007: pp. 722–735.