The Wikibase Approach to the Enslaved.Org Hub Knowledge Graph

Cogan Shimizu[1](✉), Pascal Hitzler[2], Seila Gonzalez-Estrecha[3], Jeff Goeke-Smith[3], Dean Rehberger[3], Catherine Foley[3], and Alicia Sheill[3]

[1] Wright State University, Dayton, Ohio, USA
cogan.shimizu@wright.edu
[2] Kansas State University, Manhattan, USA
[3] Michigan State University, East Lansing, USA

Abstract. Many methodologies and platforms for creating, deploying, and defining the manner of knowledge graphs are available. For this paper, we single out the platform, Wikibase. Using Wikibase comes with many advantages: out-of-the-box software for de-referencing, a convenient user interface, a consistent way to track and record provenance and lineage, and the ability to execute SPARQL queries against an RDF representation of the knowledge graph. However, the provenance mechanism and the exact nature of the structure of the Wikibase representation can complicate developing a principled schema for knowledge graphs, as well as the approach to the materialization of the data for upload to the platform. In this paper, we detail the methodology used to design, implement, and deploy the Enslaved.Org Hub, a nationally recognized knowledge graph for documenting the peoples of the historical slave trade.

1 Introduction

There are many methodologies and platforms for creating, deploying, and defining the manner of knowledge graphs now available, which emphasize different characteristics or use-cases for a particular knowledge graph (KG). Of particular interest, are *community-driven knowledge graphs* (CKGs). That is, a KG that accepts community data from community sources (i.e., data that comes from outside the original development team) and, in general, have a focus on modeling and presenting provenance and lineage of the constituent data. With the growth of the use of KGs, some communities and larger constituencies are being left behind because many of the human-machine interfaces for KGs require more advance technical skills. To address this, the platform, Wikibase, can be deployed to mitigate issues of access for less technically practiced community members.

Using the Wikibase platform has many advantages: an out of the box software for de-referencing, a convenient user interface for the less technically practiced, a consistent way to track and record provenance and lineage of data, and the option

T. R. Payne et al. (Eds.): ISWC 2023, LNCS 14266, pp. 419–434, 2023.
https://doi.org/10.1007/978-3-031-47243-5_23

to execute SPARQL queries against an RDF representation of the knowledge graph. However, the provenance mechanism and the exact nature of the structure of the Wikibase representation can complicate developing a principled schema for knowledge graphs, as well as the approach to the materialization of the data for upload to the platform.

Indeed, other institutions, such as the European Union (EU), have also come to the conclusion that utilizing Wikibase can serve as the foundation for a community-driven knowledge graph. For instance, the EU Knowledge Graph[1] [6] is deployed on a Wikibase installation, as is the Disability Wiki that serves as an metadata knowledge graph for community documents [3]. There is also growing interest in a methodology (e.g., [1]) for deploying to Wikibase, but the process is, as of yet, nascent. Furthermore, the Wikibase platform does offer up data modeling problems for the accurate deployment of ontologies, which still need to be addressed.

This paper reports on the lessons learned using Wikibase while developing the Enslaved.Org Hub,[2] which is a very visible[3] community-driven knowledge graph for documenting the stories of peoples of the historical slave trade, with a focus on North America and the Caribbean [8,9].

Concretely, this paper describes the overall approach that was taken for developing and deploying a CKG to Wikibase, using the Enslaved.Org Hub knowledge graph as a case study. The purpose of this paper is to **(a)** report on the approach taken for the Enslaved.Org Hub knowledge graph design and deployment and **(b)** demonstrate how traditional ontology engineering can be applied to deploying to the Wikibase platform. Our approach consists of three top-level steps, as follows. We provide a finer grained detailing of the steps in Sect. 3.

1. **Develop the schema:** details how the Modular Ontology Modeling (MOMo) [13] methodology can be used to create a reusable and extendable schema for a CKG that takes into account both learned lessons while developing the Enslaved.Org Hub, and reports on recent work that can improve the process.
2. **Populate the KG:** encapsulates exactly how raw data is translated into a format that can be ingested by the Wikibase platform, a process that includes the strategies taken for both de-duplication and validation of the data.
3. **Deploy the KG:** includes both the processes taken to make the data FAIR [15], and the documentation strategy.

In Sect. 2, we introduce the Enslaved.Org Hub mission and motivate the use of both Wikibase and a knowledge graph for tackling the described obstacles. In Sect. 3, we provide additional detail regarding each of the sub-steps which compose the overall approach. Finally, in Sect. 4, we conclude with future work.

[1] https://linkedopendata.eu/wiki/The_EU_Knowledge_Graph.
[2] https://enslaved.org/.
[3] It made U.S. national news upon launch, e.g. [9].

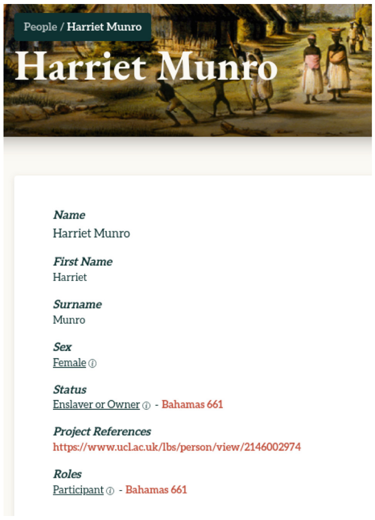
2 Case Study: The Enslaved.Org Hub

The scourge of African enslavement was fundamental to the making of Europe, Africa, the Americas, and Middle East and parts of the Asian subcontinent. The enduring legacies of black bondage shape the moral questions of humanity in our times. We have seen in the past decade a growth in interest in the subject in film, on television, and in historical fiction. Historians have spilled much ink writing monographs aimed primarily at other scholars. At the same time, however, it is a worthy goal to expand the production of scholarly output and to bring what historians do to the general public.

Recently, there has been a significant shift in perceptions about what we can know about Enslaved Africans, their descendants, and those who asserted ownership over them throughout the world. As a result, a growing number of collections of scanned original manuscript documents, digitized material culture, and databases, that organize and make sense of records of enslavement, are free and readily accessible for scholarly and public consumption. Although this data is available through individual data silos, this proliferation of different projects and databases presents scholars, students, and the interested public with a number of challenges:

- Most of these databases focus on the individuals of the slave trade, but data is often limited to the focus of the project. Further, the task of disambiguating (or merging) individuals across multiple datasets is nearly impossible given the current, silo-ed nature of all databases about slavery and the enslaved;
- There is no central, universally recognized clearinghouse for slave data. As such, it is difficult to find projects and databases;
- Individual projects and databases are isolated, preventing federated and cross project searching, browsing, and quantitative analysis;
- There are no best practices for digital data creation collectively agreed upon by the scholarly community;
- Important data is often lost or remain locked away in scholars' files, completely inaccessible to other scholars, students, descent communities, and the general public;
- Project participants rarely get scholarly credit for the work that goes into creating and releasing digital data;
- and Humanists have little incentive to deposit datasets.

To address these challenges, the Enslaved.Org project, has pioneered a new model for humanities scholarship. Enslaved.Org brings together programmers, project managers, archivists, librarians, and historians in a collective endeavor and, over the years, with an expanding consortium of contributors. This collaborative approach challenges humanists to broaden their thinking about the production of knowledge; the sharing, as opposed to guarding, of research materials; and the benefits of collaboration. In sum, the model of Enslaved.Org disrupts conventions of humanities scholarship in much the way it attempts to disrupt – for the better – historical perspectives on slavery and the individual lives of those enslaved.



People / Harriet Munro

Harriet Munro

Name
Harriet Munro

First Name
Harriet

Surname
Munro

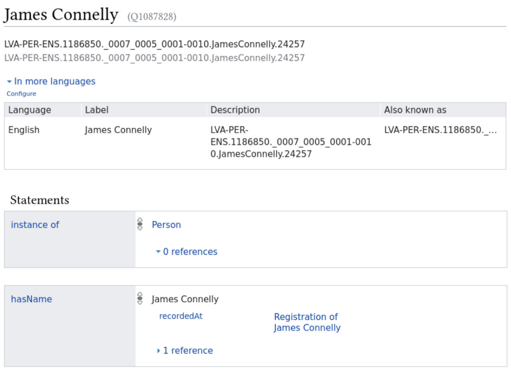
Sex
Female ⓘ

Status
Enslaver or Owner ⓘ - Bahamas 661

Project References
https://www.ucl.ac.uk/lbs/person/view/2146002974

Roles
Participant ⓘ - Bahamas 661

(a) This view shows a *knowledge box*, which provides a friendly user interface for displaying the data about a particular entity (Harriet Munro) from the Enslaved.Org Hub knowledge graph.



James Connelly (Q1087828)

LVA-PER-ENS.1186850._0007_0005_0001-0010.JamesConnelly.24257
LVA-PER-ENS.1186850._0007_0005_0001-0010.JamesConnelly.24257

In more languages
Configure

Language	Label	Description	Also known as
English	James Connelly	LVA-PER-ENS.1186850._0007_0005_0001-0010.JamesConnelly.24257	LVA-PER-ENS.1186850._...

Statements

instance of	Person 0 references
hasName	James Connelly recordedAt: Registration of James Connelly 1 reference

(b) This view shows the standard Wikibase view of the data about an entity (James Connelly), which provides a more granular detail (including provenance and lineage) for each assertion at a slight cost to readability and usability.

Fig. 1. Snapshots of the user interfaces for the Enslaved.Org Hub knowledge graph.

The technical goal of Enslaved.Org established the Enslaved.Org Hub,[4] a website that provides one-stop querying and inspection capabilities for integrated historic data on the slave trade, originating from a diverse set of data sources and contributors, thereby allowing students, researchers and the general public to search over numerous databases to understand and reconstruct the lives of individuals who were part of the historical slave trade (see Fig. 1 for screenshots of interfaces provided by the Hub). To address the underlying data integration issues, Enslaved.Org opted to follow the state of the art by establishing a knowledge graph, expressed in RDF, with an underlying schema in form of an OWL ontology, called the Enslaved.Org Ontology; the modeling approach and its core concepts are detailed in [14].

The Enslaved.Org Ontology expresses metadata record types and core fields that the Enslaved.Org research team identified as frequently occurring in historic slave trade data projects. This paper focuses on conveying the development process used to make the raw data accessible on the Wikibase platform and navigable against a schema.

[4] http://enslaved.org/.

Availability

The Enslaved.Org Hub can be found and explored at https://enslaved.org/. Documentation for Enslaved.Org metadata, ontology, and controlled vocabularies is available at https://docs.enslaved.org/. The source code is available at https://github.com/matrix-msu/Enslaved-Hub, shared under the GPL 3.0.[5] The data itself is shared under the CC BY-NC-SA 4.0 license.[6]

Uptake and Usage

Enslaved.Org has had significant public uptake [9], as well as the voluntary submission of additional historical databases.[7] Additionally, in the last year, the platform has $\tilde{3}$.5K unique monthly users and, over that time period, has had 334K page views over 53K visits. In the last six months, the dataset has been downloaded 54 times for large scale analytical purposes (by users outside of the team).

3 The Approach

This section comprises a total of seven sub-steps, aligned with the three steps of the approach, as depicted in Fig. 2 (middle and right): developing and transferring the schema (Sect. 3.1); materializing, validating, and resolving co-references in the knowledge graph (Sect. 3.2); and deploying the knowledge graph to the Wikibase platform (Sect. 3.3), as well as engaging bespoke interfaces for data discoverability, navigability, and visualization.

3.1 Developing the Schema

This top-level step in our approach comprises two distinct sub-steps: develop a schema for the use-case according to best practices and adjust the schema as necessary to fit to the semantics or technological idiosyncrasies of where the knowledge graph will be deployed.

[5] https://opensource.org/licenses/GPL-3.0/.
[6] https://creativecommons.org/licenses/by-nc-sa/4.0/.
[7] https://institute.enslaved.org/schedule/.

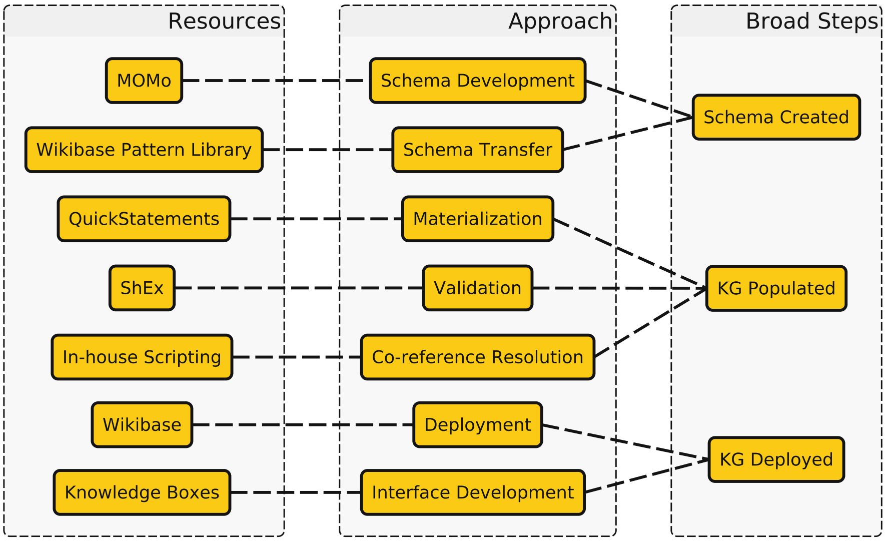


Fig. 2. This diagram displays the resources used (left) to implement our approach (middle) for creating and deploying the Enslaved.Org Hub knowledge graph. These steps are broadly categorized (right) for simplicity.

Schema Development with Best Practices The development of the Enslaved.Org Ontology – which serves as the schema for the Enslaved.Org Hub Knowledge Graph – was originally accomplished by executing the Modular Ontology Modeling (MOMo) methodology [13]. The steps of this nine-step process is shown in Fig. 3. MOMo was designed to create reusable and extendable schemas. These characteristics are driven by its modular nature, which in turn, leverage underlying best-practices encapsulated within re-used ontology design patterns. The process taken for developing the schema is in deeper detail described in [14], and it is essentially a further development of the eXtreme Design Methodology [4] emphasizing modularity, ontology design pattern *libraries*, use of certain types of schema diagrams, and a systematic approach to axiomatization in OWL.

For the context of our discussion herein, it is important to emphasize that MOMo has been designed to create ontologies capable of capturing complex relationships between ontology entities in a flexible manner akin to human expert conceptualizations, without overemphasizing formal logical aspects. One of the results of this is that MOMo often naturally leads to ontologies with plenty of reification (and sometimes even reification involving other reification). This would usually make a large ontology harder to understand, however in MOMo this is balanced out by the highly modular structure which, in a divide-and-conquer fashion, compartmentalizes the ontology into much easier to understand modules each of which focuses on a single key term that is part of the human expert's way of conceptualizing the application domain. Central and systematic

1. Identify and scope the use-case;
2. Identify key notions from the use-case and data sources;
3. Make or collect competency questions;
4. Match the key notions to patterns;
5. Instantiate the patterns into modules;
6. Systematically axiomatize the modules;
7. Assemble the modules and fill inter-module gaps, as necessary;
8. Review the final product for consistency; and
9. Produce the artifacts (e.g., documentation and serialization).

Fig. 3. This listing shows the nine steps that comprise the Modular Ontology Modeling (MOMo) [13] methodology.

use of simplified schema diagrams throughout the collaborative (with domain and data experts) development process furthermore provide a means to bridge (inter)disciplinary gaps without burdening domain experts with technical ontology engineering details.

Transferring the Schema. On the outset of the project, it was unknown exactly which platform would be used to deploy the knowledge graph. There were several options discussed, such as the use of a triplestore (e.g., Apache Jena Fuseki[8]), a property graph (e.g., Neo4j[9]), and Wikibase). It was our (naïve) thought that it would be relatively straightforward to map the ontology produced from following MOMo into a new version, as necessary, based on the future platform. That is, by possibly providing mappings between the patterns and modules that composed the resultant ontology to whichever new form was needed.

In the end, for the reasons stated above, Wikibase was chosen. This choice, while having many positives regarding its data management, transparency, and navigability support, was – unfortunately – not nearly as straightforward when mapping a traditionally designed ontology into the underlying Wikibase model [16]. As such, we point to two recent works [7,12] which report on improved processes for mapping traditional ontological formalisms into the Wikibase structure, where we seek to side-step the issue by providing ontological primitives (i.e., patterns) that are directly mappable into Wikibase, but still retaining their ability represent other common ontological formalisms that are present in other (non-Wikibase) ontologies and knowledge graph schemas.

[8] https://jena.apache.org/documentation/fuseki2/.
[9] https://neo4j.com/.

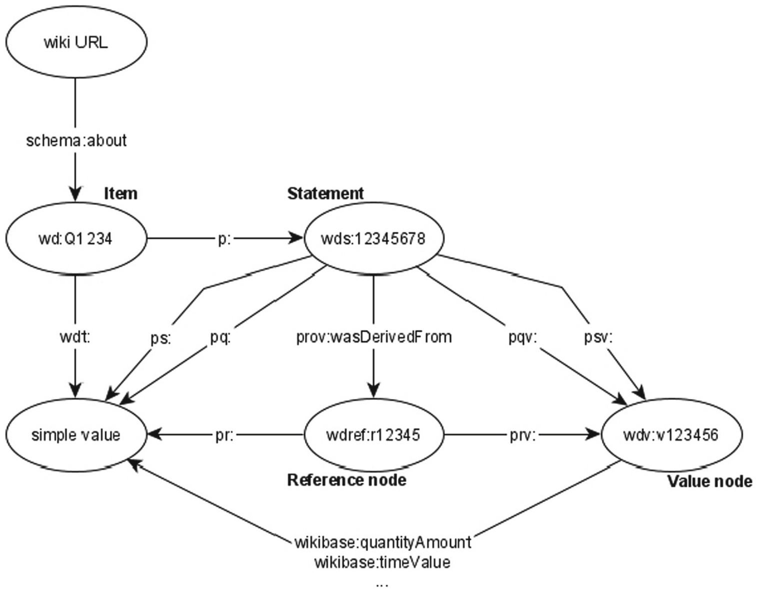


Fig. 4. This diagram shows the Wikibase RDF structure. Essentially all assertional triples are reified as "Statements," where all qualifications and provenance (references) on the assertion are linked as context to the Statement node. The different namespaces originate from Wikibase, and they are used to differentiate the role of a predicate in the reification process (i.e., turning a fact into a Statement) without necessitating changing the name of the predicate.

In particular, the Wikibase RDF structure is shown in Fig. 4. Essentially, Wikibase reifies all assertional triples into Statements, which can then have *qualifiers* and *references* attached to them, which provide contextual information about the statement. For example, a *qualifier* might be the temporal extent of the fact (i.e., when the statement was valid) and a *reference* might indicate from whence the data was derived (e.g., a publication). This sort of structure is easily modeled in labeled property graphs or the forthcoming RDF*, but is a bit less straightforward in traditional RDF graphs. Figure 5 shows how the patterns developed in [7,12] can still result in a coherent (graphical) depiction of the ontology or schema (see [12] for a detailed discussion).

By starting with these patterns (as laid out in detail in [12]), it is possible to develop a modular ontology in the spirit of the MOMo methodology, that is then seamlessly translateable into the Wikibase RDF structure. Thus, the next steps of the Wikibase process become streamlined, and some validation tasks become less onerous.

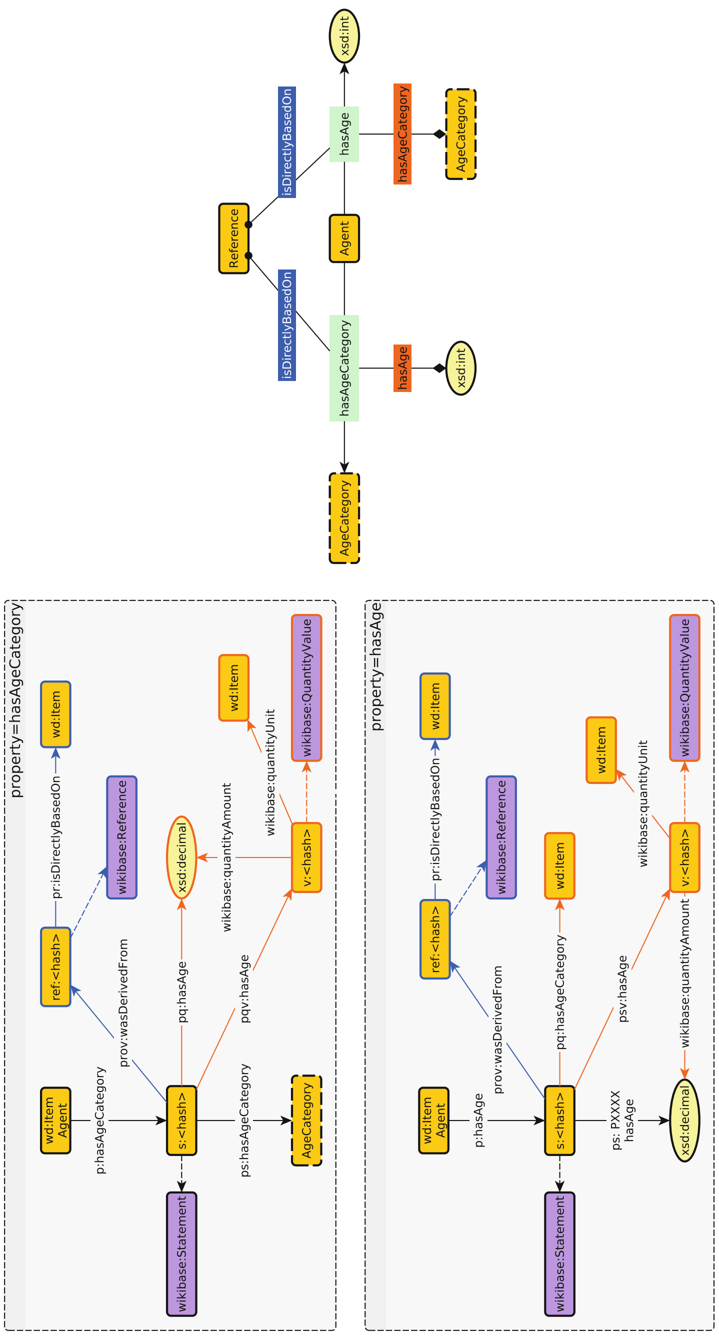


Fig. 5. This diagram shows three schema diagrams, which indicate how to model a record of a person's age according to the underlying Wikibase structure (on the left). The intent is to use only the right diagram, with the Wikibase model implicit. See [12] for elaborations.

3.2 Implementing the Knowledge Graph

In many ways, the materialization of the Enslaved.Org Hub knowledge graph is a classic Extract, Transform, Load (ETL) operation. Two pieces of software were primarily used to perform these steps: OpenRefine and QuickStatements. While we do not recommend the exclusive use of these two pieces of software, they do come with very useful functionality that aids in the development of a CKG.

OpenRefine[10] provides powerful mechanisms for cleaning up dirty or noisy data. It also provides analytical tools that can help understand the nature of ingested (or *refined*, even) data. This tool is the core piece of software for performing the Extract and Transform steps.

QuickStatements[11] is an open-source Wikibase software tool that performs ingest operations from a browser. It also provides an interface for editing the items that are to be ingested. To ingest into Wikibase, the data must be transformed into statements to be digested by QuickStatements. To this end, OpenRefine can output triples in the QuickStatements format. This tool is primarily used for the Load aspect of the ETL pipeline; it and extensions for usability, are discussed in the next section.

The Extract and Transform steps of the general pipeline occur over three discrete steps in this methodology: reconciliation,[12] materialization, and validation.

Reconciliation is a broad term for co-reference resolution and semantic harmonization. It is common, especially in the context of knowledge graphs which are community-driven (e.g., expect new data – and possibly schemas – to be contributed over time from outside the original development team) to hold some data fields as controlled, in order to prevent explosion of similar, but distinct terms from appearing in the knowledge graph.[13] Essentially, this step will take data elements from the ingested data, and based on specified column headers, attempt to map it to an identifier within a controlled vocabulary. OpenRefine offers the ability to connect a reconciliation service between the ingested data and a particular Wikibase installation. For instance, for the column *Occupation*, an occupation such as *Carpenter* is matched using a fuzzy search to an identifier of the form Q###. This Q### is the unique identifier for the item *Carpenter*

[10] https://openrefine.org/.

[11] https://github.com/magnusmanske/quickstatements.

[12] The terms reconciliation, de-duplication, and co-reference resolution tend to be used interchangeably within our field, but may have slightly different semantics in other disciplines (e.g., linguistics).

[13] This can, of course, backfire. For example, the oft-used example of *partOf* in Wikidata is extremely overloaded having many "flavors" of usage. However, the use of controlled vocabularies as a type can be used to neatly cluster terms to make querying easier (e.g., Carpenter and Woodworker can both be easily found with the same query).

in the connected Wikibase installation, and it will carry all the content associated with the term.

Materialization is the formulation of triples (i.e., text of the form *subject predicate object*, where each element of the triple is a URI). After all the vocabularies are reconciled, the schema of the data model can be uploaded using OpenRefine's Wikibase extension.[14] Within the context of OpenRefine, the schema is a list of edits that a prospective user would take to add the entry manually. As a result, this is merely a mapping of tabular data to Wikibase fields. The exact steps to fill out this form would be directly informed by the schema, as developed in Sect. 3.1. The process of transforming ontological axioms into Wikibase structures is an open question. This is partially informed by [7,12], but a principled approach has not yet been developed. For the Enslaved.Org Hub, the process was done manually by experts with an intuition for the state of the data (rather than the optimistic view an ontology would take). After the materialization process has completed, the extension provides the option to download all the statements.

Validation is a straightforward check that the data has been materialized correctly, as well as checking for holes or gaps in the data. The recommended method for doing so is via *data shapes*.[15] The W3C recommends the use of SHACL [11], but this is not required. For example, against the Enslaved.Org Ontology using ShEx [2],the Enslaved.Org Project formulated four shape expressions (one for people, one for events, one for places, and one for sources). The validation itself was accomplished via the JavaScript ShEx library.[16]

To accelerate the process, validation should be part of a CI/CD (continuous integration/continuous development) pipeline. Based on the size of the KG, ample time should be budgeted for this iterative step. Neither SHACL (via e.g., pySHACL[17] nor ShEx run quickly. Validating the Enslaved.Org KG required an overnight execution. Fixes, unfortunately, require manual intervention; and thus QuickStatements (described further in the next section) can be used to fix outstanding errors before the data is validated against the shape expressions again.

3.3 Deploying the Knowledge Graph

For this step of the methodology, we define *deploying* to mean less the configuration of the Wikibase installation, and more both the ingestion of data into Wikibase, and the considerations (within the context of the Enslaved.Org Project's context) in managing the installation over the (C)KG life cycle.

[14] https://openrefine.org/docs/technical-reference/wikibase/architecture.

[15] Essentially, validation occurs by checking the data under the closed world assumption, rather than the open world assumption.

[16] https://github.com/shexjs/shex.js.

[17] https://github.com/RDFLib/pySHACL.

As the choice of using Wikibase was largely motivated by community (data) accessibility, it is also important that updating and maintaining the CKG are approachable and accessible tasks. For the Enslaved.Org Project (i.e., Digital Humanities), it is very common to encounter people who are not technically trained. For example, Wikibase provides a Python library, Pywikibot,[18]for ingesting data, however this is only useful to those who have the requisite training.

As such, the ingestion tool of choice is another provided by Wikibase, QuickStatements, as introduced in the previous section. For the Enslaved.Org Project, however, it was necessary to improve the QuickStatements tool, and resulted in the Enslaved.Org QuickStatements.[19] In particular, some performance issues were encountered when thousands of records (items) were ingested. The tool had excessive execution time and relied heavily on the browser. Thus, if a new addition into a Wikibase installation was slow, then editing existing records was even slower because of the nature of the operation. Enslaved.Org QuickStatements tries both to fix some of these performance issues and to enhance the user experience (UX) so non-programmers feel comfortable utilizing the tool.

Briefly, some of the key improvements to Quickstatement are

- removed reliance on the browser for execution: all ingestion and editing happens server side;
- improved concurrency: parallel execution with 40 bots to ingest hundreds or thousands of records within minutes;
- restructured the database: one table per batch is used instead of having all the batches in one monolithic table, allowing access to individual batches more quickly;
- added pagination to the batches page;
- improved the general UX;
- pruned unnecessary third-party and external libraries;
- added file ingestion for statements instead of browser-only ingestion;
- ameliorated error reporting;
- and modified the tool so it relies on a different API call to edit items as a whole: a new feature that is critical for including a large number of sources and/or qualifiers per statement.

The systems environment does include additional challenges when running a Wikibase at beyond a small test environment for development. While foreseen challenges included normal concerns over maintenance, backup, and restoration, the key concerns come with the growth of the community data. Substantial growth in the items stored in Wikibase was expected. Yet, less anticipated was that Mediawiki,[20] and thus Wikibase, retain the history of every page, which in the context of Wikibase is every Item or Property. Preserving the edit history of the data we are presenting is an important function for the community, yet,

[18] https://pypi.org/project/pywikibot/.

[19] https://github.com/matrix-msu/QuickStatements.

[20] Mediawiki is the foundation to all Wikimedia software, including Wikipedia and Wikidata.

simultaneously, it is crucial to minimize the total number of edits on the data set, both for semantic reasons (edits imply something changed, and the reason for the change should be recorded), and for space reasons (every edit potentially is a full copy of the Item that must be stored). To manage this, a method was developed to create a full running second copy of the database, starting at a given point in time, for testing edits and additions. In doing so, the iterative process of testing and of building the additions and changes to a data set can be attempted and refined without adding edits to the primary repository.

Another key challenge involved in running a Wikibase is that a project like Enslaved.Org regularly requires reprocessing data – if not the whole data set, then a substantial fraction of it. Wikibase, as extension of Mediawiki, is optimized for single page viewing as the primary interactive mechanism (see Fig. 1b). Thus, free text search, which effectively requires an index of all text in the system, requires updating and maintaining that index as an ongoing process, often requiring regular sweeps of every item in the system. At the same time, for graph databases, the system often maintains a constant stream of requests for the latest changes. With these system heavy processes, if they get too far behind, the system will stop doing interactive updates and requests, and the systems operator must reload the data from scratch, from a bulk export operation. These system problems match the sort that David Rosenthal outlines in his discussion of using cloud systems for community archival work: [5] "Traditionally, access to archived content has been on an item-by-item basis. ... Each scholar accessing the collection will access a substantial proportion of it in a short time". To help alleviate this tax on systems, Enslaved.Org provides monthly dumps of all content as a downloadable dataset, as well as the same dumps that are used to load the free text and graph search tools. It is also encouraged to use these dumps for analysis.

Developing Bespoke Visualizations. How a knowledge graph is used, especially a *community-driven* knowledge graph, is greatly influenced by the interests, expertise, and backgrounds of those expected to interact with it. In particular and as stated in Sect. 2, the end-users might range from elementary school students to (amateur) genealogists to historians. Thus, we expect a wide range of socio-technical competencies and interests. As such, the Enslaved.Org Hub supports many modes of interaction: dumps of all content (supra), a traditional Wikibase view (Fig. 1b), and a knowledge box view (Fig. 1a). While the provision of the first two of these services comes easily with the use of the Wikibase platform, the latter requires additional software and UI/UX development.

The exact description of the development is outside of the scope of the paper; we, instead, mention it here as an important consideration in the deployment of any knowledge graph, but in particular those that are expected to engage with non-technical users or laypeople.

4 Conclusion

In this paper, we have outlined our approach for developing a community-driven knowledge graph that will be deployed on the Wikibase platform, as demonstrated by the Enslaved.Org Hub, which is publicly available online at https://enslaved.org/.
The approach taken comprises:

1. develop the knowledge graph schema (i.e., the ontology) according to best practices;
2. adjust the schema to the underlying semantics and technology stack which will deploy the knowledge graph;
3. de-duplicate (or resolve co-references), materialize, and validate the contents of the knowledge graph; and
4. deploy the knowledge graph, including any bespoke tools for discovery, navigation, and visualization.

Step 1 is motivated and described in detail in [14]. For Step 2, we provide updates to that process by including recent work on mapping traditional ontological structures into Wikibase much more seamlessly. Step 3 was described via its smaller tasks, and Step 4 was discusses lessons learned through the deployment of the Enslaved.Org Hub.

Finally, while this paper reports the development and deployment of the Enslaved.Org Hub, the steps taken are largely generalizable and will be helpful to those in the wider community that are interested in creating, deploying, and maintaining a knowledge graph on Wikibase.

Future Work

We have identified two particularly important next steps, which will improve the outcomes of taking our approach.

– Ontology to Shapes: This methodology does not specifically subscribe to a particular methodology for creating shape expressions from an ontology. While the process taken for the Enslaved.Org project was manual, a generalized methodology for interpreting ontological axioms into shape expressions would be useful. In particular, we will also examine the WShex [10] for applicability in this case.
– Configurable reconciliation: this methodology recommended the use of the fuzzy search functionality of OpenRefine, as it is conveniently packaged within the ingestion software. However, co-reference resolution is a relatively open field. Focusing efforts on de-duplication methods within the tighter bounds of mapping to a controlled vocabulary may be easier.

Supplemental Material Statement: Throughout the paper, references are given to enslaved.org sources, which are publicly available and relevant to assessing the paper's contributions. In particular, please see the Availability paragraph at the end of Sect. 2.

Acknowledgement. This work was supported by The Andrew W. Mellon Foundation through the Enslaved.Org Hub project.

References

1. Alemayehu, S.A., et al.: Methodology for creating a community corpus using a Wikibase knowledge graph. In: Villazón-Terrazas, B., Ortiz-Rodriguez, F., Tiwari, S., Sicilia, M.A., Martín-Moncunill, D. (eds.) KGSWC 2022. CCIS, pp. 285–297. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-21422-6_21
2. Baker, T., Prud'hommeaux, E. (eds.): Shape Expressions (ShEx) 2.1 Primer. Final Community Group Report 09 October 2019 (2019). https://shex.io/shex-primer/index.html
3. Bisen, K.S., et al.: Wikibase as an infrastructure for community documents: The example of the Disability Wiki platform. In: Simsek, U., Chaves-Fraga, D., Pellegrini, T., Vahdat, S. (eds.) Proceedings of Poster and Demo Track and Workshop Track of the 18th International Conference on Semantic Systems co-located with 18th International Conference on Semantic Systems (SEMANTiCS 2022), Vienna, Austria, September 13th to 15th, 2022. CEUR Workshop Proceedings, vol. 3235. CEUR-WS.org (2022). https://ceur-ws.org/Vol-3235/paper14.pdf
4. Blomqvist, E., Hammar, K., Presutti, V.: Engineering ontologies with patterns - the eXtreme Design methodology. In: Hitzler, P., Gangemi, A., Janowicz, K., Krisnadhi, A., Presutti, V. (eds.) Ontology Engineering with Ontology Design Patterns - Foundations and Applications, Studies on the Semantic Web, vol. 25, pp. 23–50. IOS Press (2016). https://doi.org/10.3233/978-1-61499-676-7-23
5. Cloud for preservation. https://blog.dshr.org/2019/02/cloud-for-preservation.html
6. Diefenbach, D., Wilde, M.D., Alipio, S.: Wikibase as an infrastructure for knowledge graphs: the EU knowledge graph. In: Hotho, A., et al. (eds.) ISWC 2021. LNCS, vol. 12922, pp. 631–647. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88361-4_37
7. Eells, A., Shimizu, C., Zhou, L., Hitzler, P., Estrecha, S.G., Rehberger, D.: Aligning patterns to the Wikibase model. In: Hammar, K., Shimizu, C., McGinty, H.K., Asprino, L., Carriero, V.A. (eds.) WOP 2021, Workshop on Ontology Design and Patterns 2021. Proceedings of the 12th Workshop on Ontology Design and Patterns (WOP 2021) co-located with the 20th International Semantic Web Conference (ISWC 2021). Online, October 24, 2021. CEUR Workshop Proceedings, vol. Vol-3011. CEUR.org (2021)
8. Computer science professor, postdoc launch online database on history of slavery. https://cacm.acm.org/news/249167-computer-science-professor-postdoc-launch-online-database-on-history-of-slavery/fulltext?mobile=false
9. A massive new effort to name millions sold into bondage during the transatlantic slave trade. https://www.washingtonpost.com/history/2020/12/01/slavery-database-family-genealogy/
10. Gayo, J.E.L.: WShEx: a language to describe and validate Wikibase entities. CoRR abs/2208.02697 (2022). https://doi.org/10.48550/arXiv.2208.02697
11. Knublauch, H., Kontokostas, D. (eds.): Shapes Constraint Language (SHACL). W3C Recommendation 20 July 2017 (2017). https://www.w3.org/TR/shacl/
12. Shimizu, C., et al.: Ontology design facilitating Wikibase integration - and a worked example for historical data. CoRR abs/2205.14032 (2022). https://doi.org/10.48550/arXiv.2205.14032

13. Shimizu, C., Hammar, K., Hitzler, P.: Modular ontology modeling. Semantic Web **14**(3), 459–489 (2023). https://doi.org/10.3233/SW-222886
14. Shimizu, C., et al.: The enslaved ontology: peoples of the historic slave trade. J. Web Semant. **63**, 100567 (2020). https://doi.org/10.1016/j.websem.2020.100567
15. Wilkinson, M.D., Dumontier, M., et al.: The FAIR guiding principles for scientific data management and stewardship. Sci. Data **3** (2016). https://doi.org/10.1038/sdata.2016.18
16. Zhou, L., et al.: The Enslaved dataset: A real-world complex ontology alignment benchmark using Wikibase. In: d'Aquin, M., Dietze, S., Hauff, C., Curry, E., Cudré-Mauroux, P. (eds.) CIKM 2020: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19–23, 2020, pp. 3197–3204. ACM (2020). https://doi.org/10.1145/3340531.3412768