iKNOW- A Knowledge Graph Management Platform for the Biodiversity Domain

Samira Babalou[1,2,*,†], Erik Kleinsteuber[1], Badr El Haouni[1], Franziska Zander[1,2], David Schellenberger Costa[2], Jens Kattge[2,3] and Birgitta König-Ries[1,2]

[1]Institute for Computer Science, Friedrich Schiller University Jena, Germany
[2]German Center for Integrative Biodiversity Research (iDiv), Halle-Jena-Leipzig
[3]Max Planck Institute for Biogeochemistry, Jena, Germany

Abstract

We present iKNOW, a platform for building and managing biodiversity Knowledge Graphs (KG), currently under development. We show the architecture of iKNOW and look at the planned workflow of the KG creation process.

Keywords

Semantic Web, Knowledge Graph Platforms, Biodiversity

1. Introduction & Literature Review

In the biodiversity domain, the potential benefits of Knowledge Graphs (KGs) have been recognized for quite some while [1] and first graphs on specific sub topics exist. Still, uptake is disappointingly slow. Furthermore, most of the already proposed KGs in this area focus on data from natural history collections, only [2, 3, 4]. In particular, KGs leveraging the wealth of tabular data available in the biodiversity domain are still lacking. We believe, that – as in other domains – one major roadblock to wider adoption is the large effort and high semantic web expertise still needed to create and manage KGs. Addressing this problem, in our ongoing project, iKNOW [5], we aim to build a semantic-based toolbox for KG generation in the biodiversity domain. iKNOW focuses on the (semi-) automatic, reproducible transformation of tabular biodiversity data into RDF statements.

So far, in biodiversity as in many other domains, the few existing KGs have been created largely manually in one-off efforts. While over the last few years several KG platforms have been proposed, none meets the requirements of the biodiversity domain for both generic (e.g., ingest of data in different formats, provenance management) and discipline-specific functionality (e.g., resolution of species names). If the potential for KGs is to be leveraged for this important

ISWC-Posters-Demos-Industry 2022 (International Semantic Web Conference (ISWC) 2022: Posters, Demos, and Industry Tracks)

*Corresponding author.
samira.babalou@uni.jena.de (S. Babalou)
0000-0002-4203-1329 (S. Babalou); 0000-0001-8388-4929 (E. Kleinsteuber); 0000-0001-5685-5137 (B. E. Haouni); 0000-0001-6892-7046 (F. Zander); 0000-0003-1747-1506 (D. S. Costa); 0000-0002-1022-8469 (J. Kattge); 0000-0002-2382-9722 (B. König-Ries)

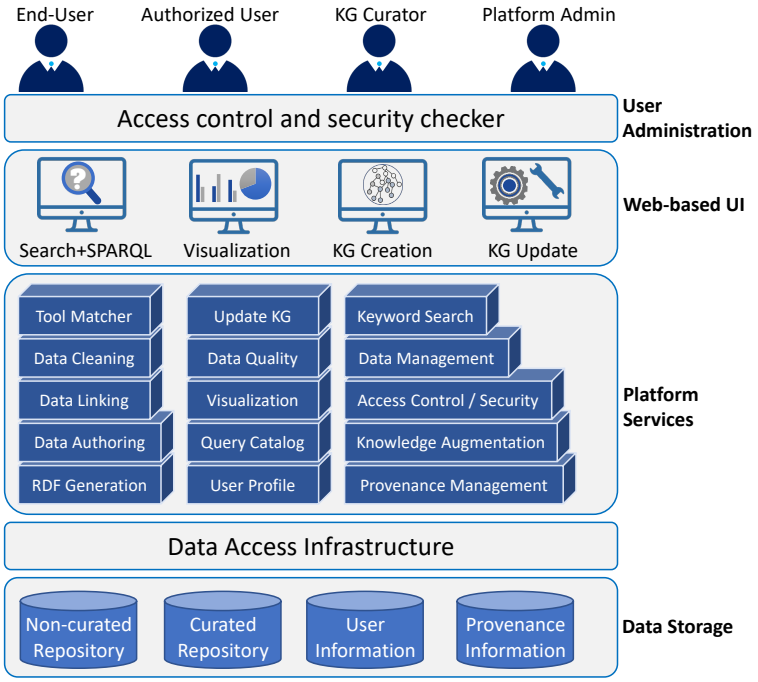

Figure 1: Architecture of iKNOW.

domain, it is our conviction, that a KG management platform providing both generic and discipline-specific functionality is needed for developing, maintaining, and using KGs. Such a platform can reduce the barriers for non-semantic web experts to use and finally benefit from KGs to explore new exciting findings. In this paper, we show the architecture of iKNOW along with its planned functionalities.

2. iKNOW: The Proposed Platform

The iKNOW project is a joined effort by computer scientists and domain experts from the German Centre for Integrative Biodiversity Research (iDiv) (www.idiv.de). The work benefits from the wealth of well-curated data sources and expert knowledge on their creation, cleaning, and harmonization available at iDiv. Thus, for now, iKNOW focuses on the (semi-)automatic, reproducible transformation of tabular biodiversity data into RDF statements. It also includes provenance tracking to ensure reproducibility and update ability. Further, options for visualization, search, and query are planned. Once established, this platform will be open-source and available to the biodiversity community. Thus, it can significantly contribute to making biodiversity data widely available, easily discoverable, and integrable. In this section, we present shortly the architecture and workflow of KG generation at iKNOW.

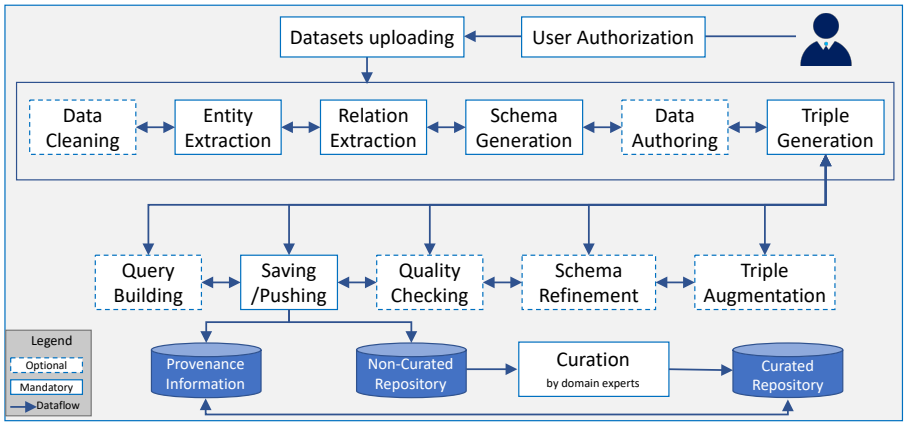

Figure 2: Workflow in the KG Creation Scenario at iKNOW.

2.1. iKNOW Architecture

Figure ?? shows the planned architecture of iKNOW in five layers: (i) In the **User Administration** layer, access level and security will be controlled. Authorized users can generate or update the KG. All end-users can search and visualize the KG. The platform's admin can add new tools or functionalities and approve the user registration. The KG curator curates the recent changes on the KG. (ii) The **Web-based UI** layer shows different scenarios for KG management: building a KG, updating the KG, visualizing the KG's triples, and keyword and SPARQL search. (iii) The **Platform Services** layer provides a set of required services for the KG management functionalities. (iv) On the **Data Access Infrastructure** layer the communication of services and data storage is managed. (v) At the bottom level of the iKNOW platform, the **Data Storage** layer contains the graph database repository, provenance information, and user information management.

2.2. Workflow in the KG Creation Scenario

Figure 2 shows the planned iKNOW workflow for the KG creation scenario. The workflow shows the data flow between the steps towards KG generation. Not all steps are mandatory; some optional processes in each step can add further value to the KG based on the user's needs. For every uploaded dataset, we build a sub-KG. It will be the subgraph of the main KG in iKNOW. In the first step, users go through the authentication process. The verified users can upload their datasets. If required, the *Data Cleaning* process will take place. We plan to offer different tools for this step, which users can select and adjust based on their needs. In the *Entity Extraction* step, we map the entities of the dataset to the corresponding concepts in the real world (which build instances of sub-KGs). This mapping is the basis for interlinking entities with external KGs like Wikidata or domain-specific ones. Each mapped entity is a node in the KG. For this process, we will embedded different tools at iKNOW, in which users can select the desired tool along with the desired external KGs. In the *Relation Extraction* step, the relations between the KG's nodes will be extracted via the user-selected tool. Note that in the entity and relation extraction

steps, the tools return the extracted entities and relations to the user. Through our GUI, the user can edit them (*Data Authoring* step). Each column from the relational dataset refers to a category in the world. We consider the types of the column as classes in the KG. Along with the extracted relations in the previous step, the schema of this sub-KG will be created in the *Schema Generation* step. In the *Triple Generation* step, (subject, predicate, object)-triples based on the extracted information from the previous steps will be created. Nodes in the KG are subjects and objects, and relationships are predicates. The triples are generated for classes and instances in the sub-KG.

After these processes, the generated sub-KG can be used directly. However, one can take further steps such as: *Triple Augmentation* (generate new triples and extra relations to ease KG completion), *Schema Refinement* (refine the schema, e.g., via logical reasoning for the KG completion and correctness), *Quality Checking* (check the quality of the generated sub-KG), and *Query Building* (create customized SPARQL queries for the generated sub-KG). In the *Pushing* step of our platform, the generated KGs are saved first at a temporal repository (shown by "non-curated repository" in Figure 2). After a manual data curation by domain experts in the *Curation* step, the KG will be published in the main repository of our platform. With this step, we aim to increase the trust and correctness of the information on the KG.

All information regarding the user-selected tools with parameters and settings along with the initial dataset and intermediate results will be saved in every step of our platform. With the help of this, users can redo the previous steps (which shows by arrows in both directions). Moreover, this enables us to track the provenance of created sub-KG. In each step mentioned above, we plan to have a tool-recommendation service to help the user select the right tool for every process. For that, we will consider different parameters, such as the characteristics of the dataset and tools.

3. Implementation

The iKNOW platform is currently under development (https://planthub.idiv.de/iknow) and is distributed under an open-source license in github.com/fusion-jena/iKNOW. The Python web framework Django (www.djangoproject.com) is used for the backend with a PostgreSQL (www.postgresql.org/) database to maintain users, services, tools, datasets, and the KG generation parameters in the iKNOW platform (used in provenance tracking). We use the compiler Svelte (https://svelte.dev/) with SvelteKit as a framework for building web applications to create a user-friendly web interface. For security, maintenance, and provenance reasons, all tools from external providers used within the workflow will be executed in a sandbox using Docker (www.docker.com/). For managing the triplestore, we are using the graph database Blazegraph (https://blazegraph.com/). Any sub-KG created by an end-user, first, will be placed at the non-curated triplestore. After curation by domain experts, the new sub-KG will be added to the curated triplestore. The curated triplestore also serves as the base for SPARQL queries and the keyword search via search engine Elasticsearch (www.elastic.co/elasticsearch/).

iKNOW is a modular platform, which increases the flexibility of our platform and allows adding new tools. Our ultimate goal is to provide a large set of tool choices for the end-user. Although only a few tools are embedded so far, we plan to add more tools for each functionality

in the platform. Then users have a variety of choices with respect to different needs and use cases. Our open-source code and modular designs of our platform make both the front and backend of our platform easily extendable. We encourage users (new developers) to use or extend our reusable UI components to speed up their development.

Acknowledgments

The work described in this paper is conducted in the iKNOW Flexpool project of iDiv, the German Centre for Integrative Biodiversity Research, funded by DFG (Project number 202548816). We thank our colleague Sven Thiel for comments on the manuscript and the iKNOW PIs Helge Bruelheide, Christine Römermann and Christian Wirth for their insights into biodiversity research and data integration needs.

References

[1] J. Sachs, R. Page, et al., Training and hackathon on building biodiversity knowledge graphs, Research Ideas and Outcomes 5 (2019) e36152.
[2] R. D. Page, Ozymandias: a biodiversity knowledge graph, PeerJ (2019).
[3] L. Penev, M. Dimitrova, et al., Openbiodiv: a knowledge graph for literature-extracted linked open data in biodiversity science, Publications (2019).
[4] M. Stocker, T. Heger, et al., Skg4eosc-scholarly knowledge graphs for eosc: Establishing a backbone of knowledge graphs for fair scholarly information in eosc, Research Ideas and Outcomes 8 (2022) e83789.
[5] S. Babalou, D. Schellenberger Costa, et al., Towards a semantic toolbox for reproducible knowledge graph generation in the biodiversity domain-how to make the most out of biodiversity data, INFORMATIK 2021 (2021).