Knowledge Graphs: Semantics, Machine Learning, and Languages
M. Acosta et al. (Eds.)

doi:10.3233/SSW230005

Towards a Versatile Terminology Service for Empowering FAIR Research Data: Enabling Ontology Discovery, Design, Curation, and Utilization Across Scientific Communities

Philip Strömert[a,1][0000-0002-1595-3213], Vatsal Limbachia[a][0000-0002-9535-6960], Pooya Oladazimi[a][0000-0002-5117-070X], Johannes Hunold[a][0000-0002-4378-6061], and Oliver Koepler[a,1][0000-0003-3385-4232]
[a]*TIB - Leibniz Information Centre for Science and Technology, Hannover, Germany*[04aj4c181]

Abstract. To fully harness the potential of data, the creation of machine-readable data and utilization of the FAIR Data Principles is vital for successful data-driven science. Ontologies serve as the foundation for generating semantically rich, FAIR data that machines can understand, enabling seamless data integration and exchange across scientific disciplines. In this paper, we introduce a versatile Terminology Service that supports various tasks, including discovery, provision, as well as ontology design and curation. This service offers unified access to a vast array of ontologies across scientific disciplines, encouraging their reuse, improvement, and maturation. We present a user-driven service development approach, along with a use case involving a collaborative ontology design process, engaging domain experts, knowledge workers, and ontology engineers. This collaboration incorporates the application and evaluation of the Terminology Service, as well as supplementary tools, workflows, and collaboration models. We demonstrate the feasibility, prerequisites, and ongoing challenges related to developing Terminology Services that address numerous aspects of ontology utilization for producing FAIR, machine-actionable data.

Keywords. FAIR Data, Ontology, Terminology Service, Research Data, Chemistry

1. Introduction

Resources like re3data [1] or FAIRsharing [2] provide comprehensive collections of data repositories, databases, data, and metadata standards and policies. Initiatives like the European Open Science Cloud (EOSC) [3], the German National Research Data Infrastructure (NFDI) [4], and the Physical Sciences Data Infrastructure (PSDI) in the United Kingdom [5] create and provide services and infrastructures to make research data publicly available, adapting the FAIR data principles [6]. The vision of Open

[1] Corresponding Authors: Philip Strömert, philip.stroemert@tib.eu; Oliver Koepler, oliver.koepler@tib.eu.

Science has gained momentum and is being put into practice by more and more scientific communities.

The availability of increasingly large amounts of data enables the exploration of new data-driven, interdisciplinary research questions. These approaches require integration and harmonization of machine-actionable data across disciplinary boundaries addressing Findability, Accessibility, Interoperability, and Reusability. The idea of having machine-actionable data is derived from the JDDCP [7] guideline that refers to making the data readable and modifiable by machines.

Machine-actionable, FAIR research data is achieved by annotation with rich metadata. Furthermore, these metadata themselves also need to be FAIR, meaning metadata used for data annotation needs to be understandable and actionable by both humans and machines. Generic metadata schema like DataCite [8] or Dublin Core [9] cannot express rich discipline-specific descriptions of data. This often results in limiting the annotation to basic metadata such as title, author, date, and format and providing the domain-specific metadata in the form of long free text that decreases data FAIRness due to the lack of semantics [10].

Here, terminologies play an important role in the creation of semantically rich, discipline-specific metadata. They further provide the basis for consensus definitions of entities, thereby ensuring conceptual alignment across domains, even when the nomenclature differs between domains. The use of metadata schemata implementing standardized terminologies promotes interoperability and data integration, as data described through common terminologies can be understood and used across different systems and disciplines.

Terminology services like the Ontology Lookup Service [11], Bioportal, or Linked Open Vocabularies provide access to either general or discipline-specific collections of ontologies, terminologies, or vocabularies. They offer features like browsing, searching, filtering, and downloading ontologies. Terminology services, therefore, play a crucial role in the identification of relevant ontologies in the process of data annotation and the creation of FAIR data. Their relevance is reflected in the integration into various data annotation or data management tools. The CEDAR workbench supports data annotation with domain-specific metadata schema [12]. For the creation of metadata templates, the workbench integrates ontology terms from the Bioportal terminology service [13] which the user can select from. The Dataverse project enables the integration of customized metadata schema which can be populated with terms of linked terminology services [14] [15]. The electronic lab notebook Chemotion utilizes terminologies to annotate data of experiments [16].

With the increasing use of terminologies for data annotation, errors and gaps in terminologies are inevitably uncovered not only by ontology engineers but also by domain experts annotating data. This opens up new application areas for terminology services. What could be more natural than using the services that are used for searching and analyzing ontologies for curation and development as well?

In this paper, we discuss collaborative ontology development workflows and the requirements for a versatile Terminology Service to support these workflows. What are the best approaches and best practices to collaboratively design and curate ontologies in a team of ontology engineers, knowledge workers, and domain experts? What are the requirements for a Terminology Service to support the design and curation workflows of stakeholders with varying levels of expertise? How can a Terminology Service improve ontology development concerning harmonizing and aligning the application and reuse of terminologies across knowledge domains?

In the following section, we provide an overview of related work concerning terminology services and ontology development tools. In the subsequent section, we briefly describe the challenges when using terms from terminologies for data annotation or when reusing terms from existing terminologies in new ones. Section 4 describes our approach to collaborative ontology development and lessons learned in the application of a terminology service for the derived workflows. In section 5 we describe our terminology service and the enhancements we have developed based on the requirements derived from experiences and observations during the ontology development process. Section 6 summarizes and discusses insights and evaluation of user-driven development processes to extend the terminology service as a tool for ontology development and curation.

2. Related Work

Two types of services and tools aim to facilitate ontology access and development: Terminology Services (TS) and ontology development tools. In this section, we present the most prominent open-source tools and services and also discuss deficiencies and gaps in existing solutions concerning envisioning a well-integrated TS for both ontology development and data annotation.

2.1 Terminology Services (TS)

A number of mature terminology services and repositories are publicly available. Terminology registries such as the Basic Register of Thesauri, Ontologies & Classifications (BARTOC) represent the simplest form of such a service [17]. Registries usually list comprehensive metadata about vocabularies, terminologies, and ontologies and link to their original source while Terminology repositories provide access to the terminology data itself, often in combination with access to this data via APIs [18]. Well-used TS frameworks like OntoPortal [19] and Ontology Lookup Service (OLS) [20] provide extended features like searching and browsing within the indexed terminologies, tree views of concepts and properties, and visualization. OntoPortal is an ontology-supported portal architecture developed by the OntoPortal Alliance that can be customized to provide discipline-specific terminology services. It has emerged from BioPortal, which provides access to a collection of biomedical ontologies and terminologies [13]. The Ontology Lookup Service, developed and hosted at the EMBL European Bioinformatics Institute (EBI), provides ontology search and visualization services as well as data access and search through an API. Furthermore, there is Skosmos, an open-source web-based browser and publishing tool specialized for Simple Knowledge Organization System (SKOS) vocabularies [21]. It provides a user interface for browsing and searching the data as well as Linked Data access with APIs that support term-based searches. DBpedia Archivo archives and ontologies on a web-scale to offer access to their different versions over time. It also provides quality metrics [22]. The Linked Open Vocabularies (LOV) portal provides a comprehensive overview of vocabularies in the realm of the Semantic Web and Linked Data, which can also be accessed via a SPARQL endpoint [23].

All terminology services provide rich functionalities, yet they typically encompass ontologies from specific domains or disciplines alone. While this approach proves feasible in many instances where users want to focus on their respective domains, it does pose challenges for those seeking a comprehensive overview of available ontologies across diverse domains for interdisciplinary applications. Both use cases are legitimate and an ideal terminology service should possess the capability to cater to both of these requirements.

2.2 Ontology development tools

Open-source ontology editors like Protégé and its online version WebProtégé [24] or visualization tools like WebVowl [25] can be used to inspect and develop ontologies one at a time. Yet when it comes to researching terms in multiple ontologies or comparing multiple ontologies at once, these tools lack the flexibility a TS provides. At the same time, they are tailored to experienced users and can be quite overwhelming for the novice user, due to the broadness and complexity of the functionalities they provide.

Such development tools are also not well suited for seeing changes between versions of the same ontology directly. To ensure applications will not break when implementing a new version of a reused ontology, it is however essential to enable ontology users to access and keep track of the changes between different ontology versions in a simple manner. This means all ontology versions should be explorable and comparable. If one wants to avoid loading different versions of the same ontology into multiple instances of one's ontology editor, one would usually resort to a TS. Yet, when it comes to providing multiple versions of multiple ontologies, the sheer amount of data and often missing versioning information can pose quite a challenge. Efforts in this direction on the pure ontological level have been made in the DBpedia Archivo project. Integrating version differences into a TS that also allows for browsing the terms of an ontology thus requires either high computational resources or new approaches around displaying changes, like the efforts behind the planned Knowledge Graph Change Language [26].

In addition to ontology archiving and versioning, there have been efforts to facilitate the ontology development and maintenance process. The main focus in facilitating ontology's development process was automation and quality assurance standards. For instance, ROBOT [27] was developed based on the standards of software development to help ontology developers automate common tasks like file conversions, error checking, reasoning, metadata annotation, modularization, and release management. The Ontology Development Kit (ODK) [28] integrates templates, standards, and quality checks with tools like ROBOT or GitHub in a bundled way using Docker to make their use in common development workflows easier.

As mentioned, ontology development tools are aimed at helping developers to have an optimal and precise development process. Also, they help the developer to have a better understanding of ontologies. However, they do not offer powerful ways to explore ontologies as much as TSs. On the other hand, TSs mainly focus on browsing and are not trying to provide further tools regarding the ontology development process. Considering these observations, an ideal TS should provide some tools to facilitate the development process to some extent, when it is used as a trusted main medium through which ontologies are perceived and understood.

3. Prolog - Finding the right words

A major challenge when using terminologies for data annotation or when importing parts of them in new terminologies is that one has to decide which terms to use from which terminology. In order to be able to make an appropriate decision, one must either already know which terminologies and terms are suitable for a specific scientific context and use case, or, more likely, one must be able to gather this knowledge by browsing the available terminologies. A terminology service with a graphical user interface (GUI) is such a resource, which renders terminologies comprehensible for humans, and should therefore be able to provide as much detailed information as possible on the terminologies themselves as well as on the terms they contain. This aspect is especially crucial when there is the need to compare multiple terminologies covering the same or overlapping knowledge domains. Terminology users need to be able to see the semantic differences and similarities as well as interdependencies between terminologies to make informed decisions. An intuitive way of browsing terminologies is thus key for evaluating the scope and use case applicability of terminology and for grasping its overall logical composition. It is important to be able to traverse the term hierarchy easily, similar to how we use file browsers or navigate publications in our document readers. In other cases, we need the ability to sort and filter the terms of terminology, like we are used to from working with spreadsheets. Such GUI functionalities are needed to keep focused when exploring the sometimes very complex semantics of terminology or to get to the desired information quickly in everyday workflows.

4. Ontology Development

To find out which features are required from a TS more concretely in collaborative ontology development workflows, we have analyzed the experiences made by domain and ontology experts in a still ongoing ontology development effort within the NFDI4Chem project [29]. The focus here was to benefit from a user-centered approach where we involve the ontology developers in our TS development process from the early stage to collect fine-grained user stories and requirements.

4.1 Vibrational Spectroscopy Ontology (VIBSO)

As the name implies, the intended domain covered by the Vibrational Spectroscopy Ontology is the discourse around a particular kind of spectroscopy that assays the vibrational modes of molecules and crystals. To be more precise, the ontology is meant to provide a formal representation of the technical terms used by domain experts to describe and share the research data output produced by such assays. It must thus contain terms that refer to the experimental setup, like the used devices, their relevant parts, and attributes, terms that refer to associated processes preceding and following the actual assay, like the preparation of the sample or the data transformation producing interpretable spectral images from raw signals, as well as terms that refer to relevant characteristics of these spectral images, like their dimensionality or analytical significance.

Adhering to best practices in ontology development [30], VIBSO's domain coverage depends heavily on reusing many terms from existing ontologies instead of defining them anew. First and foremost it depends on the Basic Formal Ontology [31] as a common ground for the abstract upper-level classes such as *material entity* or *process* and on the Relation Ontology [32] for commonly used relations. The Ontology for Biomedical Investigations [33] is another important and more concrete dependency of VIBSO, as it provides many general classes and some specific relations within the domain of scientific investigations, such as *assay*, *device,* or *protocol*. Most importantly, VIBSO's core depends on classes from the Chemical Methods Ontology [34], which already defines branches for the main chemical methods of interest - *vibrational spectroscopy* and *Raman spectroscopy* - as well as other general classes needed, such as *spectrum* or *spectrometer*. We are collaborating with the developers and maintainers of CHMO and due to the domain-specific overlap between it and VIBSO, there is the possibility to integrate VIBSO into CHMO in the future. At the moment, however, it seems best to keep the two separated to address the identified gaps and issues regarding VIBSO's scope.

4.1.1 Development Approach

Since most of the ontologies VIBSO depends on are part of the OBO Foundry [35] and to thus ensure interoperability, VIBSO's development also follows the best practices and principles [30] of this community. Furthermore, it is being developed in an iterative, version-controlled way that relies on continuous integration to make sure all changes are properly tracked and that the release files are quality controlled. For the technical implementation of this approach, we use the ODK, ROBOT, Protégé, GitHub, and the NFDI4Chem collection of our TS. The latter plays a central role in the development of VIBSO, as it is used to browse its most current version, to link to its terms in discussions, to search for requested terms in other domain-related ontologies, and to regularly look up the axiomatization patterns and term details of the reused ontologies. With regard to the conceptual aspect of our development approach, we rely on the collaboration with domain experts from chemistry and related scientific fields in which vibrational spectroscopies are being used, as they are the ones who know best what concepts are needed in this domain and how to label and define them. So far, we could benefit from the domain knowledge of scientists from the NFDI4Chem project, the CHARISMA project [36], BASF, and the Scuola Normale Superiore. These domain experts provided an initial list of terms from which a first ontology draft was created. Further term requests have since been filed and are being discussed with the domain experts mostly in regular open online calls and to a much lesser degree in the ontology's source code repository on GitHub.

As part of these development discussions, the domain experts are also asked to provide feedback on the used tools and workflows. This feedback is then used to find out how we can improve such a collaborative ontology development process more generally by enhancing the usability of the TS. The rationale behind this is not to turn the TS into yet another ontology editor that is just simpler to use. We also do not expect the domain experts to contribute to the source code of VIBSO directly, by having to learn the required specialized tooling. We rather want to develop simpler ways to communicate with them about the semantics of specific terms and terminologies. Making it easier in the TS to suggest new terms and changes to their metadata annotations (e.g. labels, definitions, or synonyms) or to discuss terminologies or terms in place, we believe that domain and ontology experts can benefit more from each other's expertise. We thus rather aim at making the TS a better tool to analyze and annotate the broad spectrum of available terminologies.

4.2 Lessons Learned from the VIBSO Development

Within the discussions between ontology engineers and domain experts, we have learned that tools to browse ontologies should be applicable in a simple and uncomplicated manner close to similar tools used by domain experts. Being still rather unfamiliar with the TS and its standard tree view for browsing an ontology, our domain experts requested to also have a tabular view that lists the available VIBSO terms to get a better overview. As a first step, we tried to address this need by switching from a pure Protégé based editing approach to one that uses spreadsheets and a TSV file as an input format for the definition of new VIBSO classes. Although this approach makes it easier for the domain experts to directly see, comment and edit the existing classes or create new ones by using common spreadsheet editors, it often seems to be too much of a hurdle to access and change this TSV via GitHub. In addition, the TSV defines only the classes from VIBSO and not the classes or relations imported from external ontologies. The domain experts thus have no direct way to understand the semantics of the latter. They would have to use the TS to look up why these terms are being used as parent classes or as part of VIBSO's axiomatization in the TSV. Being able to list all of the classes of an ontology in a tabular view directly in the TS avoids these downsides. If such a tabular view could also be used to sort and filter the class list, it would be easier than in the tree view to grasp or analyze certain details more intuitively. Sorting or filtering by term identifier would allow, for example, a more direct differentiation between imported and native terms, which allows one to quickly grasp the magnitude of external dependencies of an ontology. We believe that such a class list view could also gradually improve the acceptance and use of the tree view if the switching between the two views is implemented in a way that enables the user to learn to appreciate their differences in rendering an ontology.

When it comes to discussing term definitions and metadata or ontology-related questions as well as when one wants to request new terms, the most common approach in open-source development is to file issues in the Git-based repositories where the ontologies are maintained. Having to leave the TS to do so means an extra effort for the user. This has caused us to think about how we can use the TS as a medium to simplify such interactions. It would be more user-friendly to be able to at least list such external issues and in the best case read, write, or comment on them directly from within the TS. Like some of VIBSO's domain experts, ontology users who are unfamiliar with such external services would thus not need to learn a second platform to access more context information about an ontology, and to provide valuable feedback to its developers. At the same time also ontology users and developers who are familiar with the required Git workflows could benefit from this in their daily work by not having to switch contexts. Of course, such a feature should at best work with different version control platforms, like GitHub or GitLab. Fortunately, these two have suitable APIs and are very commonly used for open ontology development. So focusing on one of them can be considered already a great step forward.

Another aspect we have identified as a useful enhancement of the TS is the ability to add another layer of context information intended for a special user group. Apart from the metadata annotations provided on terms or the whole ontology, further documentation on design patterns and choices, which helps a user to better understand an ontology, is usually provided in scientific papers or discussions happening on mailing lists and development platforms. Well-curated ontologies contain links and notes to these sources. However, curating such links and notes is quite an effort, as the curators have to decide on their appropriateness to be included in the source code. Many use case or project-specific notes and discussions might thus be excluded or remain buried in the external sources. In addition, when following these links one usually has to leave the TS and thus runs the risk to get sidetracked. Especially when using multiple ontologies in a modular fashion, it can be quite challenging for less experienced users to understand why certain terms have to be imported, as they might seem unnecessary from a use case perspective but are needed from an ontological perspective to remain in line with their axiomatization. Having a functionality in the TS that allows its users to comment on the term and ontology level would be an alternative to sharing in-place insights.

As an example, with a note on the imported OBI term *assay* in VIBSO, which is an important superclass for the needed specializations of vibrational spectroscopy assays, we could provide directly further context to our domain experts about how to interpret the asserted and inferred axiomatization to remain in line with OBI design patterns, when defining new such specialized assays, instead of having to provide this information in the development documentation in other places. By making such a note and its particular context visible also on the same class in other ontologies, other TS users might benefit from such insights as well.

On the other hand, with notes on the ontology level, users can discuss their applicability in certain use cases. To remain in the VIBSO example, it would be quite helpful to add notes on CHMO that communicate to our domain experts the gaps and issues we have identified and need to address in VIBSO and link to their associated issues for further details. With such an additional layer, the CHMO developers as well as others could thus better keep track of our work without having to search multiple source code repositories for the related issues. For ontologies that are not maintained on an open platform that allows file issues, which unfortunately is still quite common, such a TS feature would be even more helpful. We believe that using a TS as such a medium could be a valuable way to keep a better overview of issues that span multiple ontologies and thereby better tackle complex issues of harmonization and mapping.

The experiences made by the ontology experts with the TS in the development of VIBSO and other NFDI4Chem-related ontology work, also lead to improvement suggestions for the TS. One major advantage of the TS for this user group is that for browsing and looking things up quickly in many different ontologies, they do not have to load all of them into Protégé. Another advantage of a TS is the ability to link to individual terms directly, which is an important way to reference these in many different contexts. Doing the lookup tasks in a web-based TS and the development tasks locally in Protégé helps a lot to keep focused, and keeping your focus is very important when working with formal ontologies. Yet, this user group is used to certain, often small but convenient, features that ease their daily workflows. Making such features also available in the TS where it makes sense is preferred. One of such features our ontology experts have identified as useful to port to the TS is the ability to traverse the hierarchy tree quickly with the keyboard navigation keys. Another one is the ability to adjust the size of GUI panes containing the tree view and term details. To find out which other features are also good candidates for porting to the TS, we will have to do further user research.

Table 1: Overview of lesson learned (LL) by observing the VIBSO development process.

Lesson Learned	Description
LL1	Extensive List View for classes.
LL2	Git issue list for ontologies.
LL3	Report issue and Term Request for ontologies.
LL4	Take/Read Notes on ontology/class/property/individual.
LL5	UI improvement such as keyboard navigation for tree views and resizing the tree view area pane.

5. Terminology Service

The development of our terminology service was initially motivated by the need to provide a robust service to search, browse, analyze, and access ontologies, terminologies, or vocabularies for various communities and scenarios like semantic annotation of research data or data generation. The TIB Terminology Service addresses these needs and provides overarching access to ontologies across multiple domains. Introducing the concept of *collections*, we can group ontologies by discipline, domain, or project providing customized views supporting communities not only to identify but also request new terminologies. The NFDI4Chem terminology service [37] is such a discipline-specific view for researchers interested in chemistry ontologies and has been used in VIBSO development. Table 2 summarizes available collections established so far.

Table 2: Collections used to bundle terminologies by domain or projects in the TIB Terminology Service.

Collections	Terminologies	Classes	Properties	Individuals
NFDI4Ing	53	483167	5840	3522
NFDI4Chem	38	147788	5928	26937
CoyPu	8	3767	2955	17289
NFDI4Culture	5	234	785	10
FID Move	9	121501	410	2439
FID BAU Digital	11	12820	1465	14702
FAIR Data Spaces	32	558036	4119	14068

Collections like NFDI4Chem are curated by community-agreed quality criteria [38], [39] applying workflows for suggesting new terminologies. Besides collections, ontologies can also be assigned to subjects like chemistry, physics, or engineering sciences. This classification can be applied to narrow down the ontologies of the TS to work with a specific set that is related to their scientific domain. The Terminology Service periodically checks updates for the indexed terminologies at the original sources to ensure it provides the latest version available.

5.1 Architecture

The TIB Terminology Service was designed following the Frontend-Backend pattern, which positions an ontology lookup service as the data-providing backend for a frontend application developed using the React library. The backend service is built upon the Ontology Lookup Service (OLS) developed by EBI. This architectural structure embodies a tightly integrated design with dedicated modules for ingesting and indexing terminologies. Additionally, it incorporates graph libraries that facilitate the visualization of these terminologies and also defines API methods that enable the presentation of data on the front-end. This architecture allowed us to take advantage of not only the pre-existing API but also the ontology ingestion process, which streamlined our development efforts. However, to ensure the architecture adequately served the unique requirements of the envisioned Terminology Service, we introduced new methods into the adopted system.

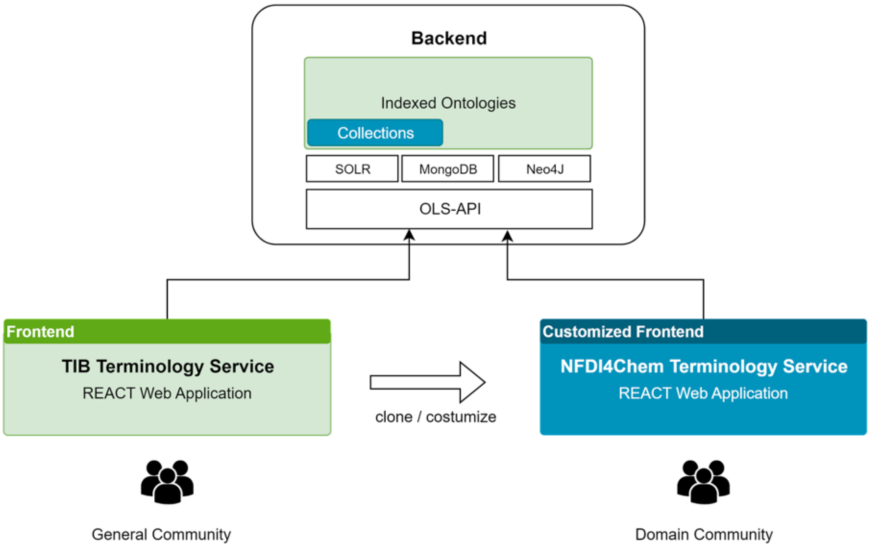

Figure 1: Architecture Terminology Service

The React frontend application addresses the need for flexible, autonomous implementation of features in the user interface, decoupled from the original approach of the OLS-web app.

As the original OLS user interface is embedded into the OLS-web app which is tightly integrated into the OLS backend architecture, serving individual communities would require running multiple OLS backend instances. This is costly for individual projects and hard to maintain.

We, therefore, decided for the TIB Terminology to ingest and index all ontologies in one backend while assigning the ontologies to one or more collections as we have described earlier. These collections are served to the respective communities by the front-end addressing their individual needs and preferences. Usually, collections can be accessed by choosing the filtered view in the search or browsing interface in the TIB TS. If required by a community the collections can be offered as highly customized, individual frontend applications. Such instances are cloned from the original source code of the TIB Terminology Service and are further tailored via configuration settings during the deployment process. This approach led to the development of customized solutions such as the NFDI4Chem Terminology Service. Such instances can not only be customized concerning the collection of terminology but also be enriched with customized components only available for the selected community. These components are centrally maintained in the TIB Terminology Service source code repository and can be activated during the deployment process of the customized front-end.

This architectural design ensures that the TIB Terminology Service is adaptable and responsive to the distinct needs of various user communities, providing them with customized interfaces and functionality while maintaining a centralized, coherent backend.

5.2 User-driven Implementation

The main motivation for implementation was to address the lesson learned and feedback that we observed through the VIBSO development process. As a result, we developed a new feature for each lesson learned in Table 1. Features under development can be accessed on the Terminology Service demo server (https://service.tib.eu/terminology). The source code is available via GitHub [40].

5.2.1 List of Github Issues for an Ontology

As highlighted in Section 4.2, an ideal Terminology Service (TS) should provide the necessary tools for facilitating the development and curation of high-quality, error-free ontologies. Moreover, many ontology developers leverage the GitHub version control system to optimize their development process, aligning it with established software development standards.

In light of this, enabling TS users to interact with GitHub directly through the TS emerged as an essential requirement. This interaction can take both passive and active forms. Passive interaction might involve tracking the discussions surrounding specific ontologies, while active interaction could entail participating in these discussions or even reporting issues.

As a result, we intended to provide a Terminology Service in such a way as to optimize the workflow for both the terminology developer and the end user. To achieve this, we developed a new feature that allows users to access the list of issues for a given ontology repository directly within the Terminology Service.

The feature aids users in tracking the progress of an ontology and being aware of the items in the ontology, which are being worked on. Moreover, it is useful to evaluate the quality of an ontology based on the latest activities, number, and kind of open issues. It can prompt both users and developers to take further steps. Finally, it reduces the mental load associated with context-switching, as users do not have to leave the TS to get to the GitHub issues. Currently, this feature is only available to the ontologies hosted and maintained on GitHub.

5.2.2 Issue Report and Term Request

In addition to monitoring existing repository issues for TS users, we recognized the importance of enabling them to report their issues, a need highlighted by our observations of VIBSO. This feature not only alleviates the need for a context switch between the TS and the target ontology repository (such as GitHub) but also empowers less experienced users of Git-based systems to actively participate in the development process by providing valuable feedback. Furthermore, it offers ontology developers the opportunity to enhance their work quality through an accessible and constructive feedback stream.

To address this need, we introduced a feature that allows TS users to create an issue on the target ontology repository hosted on GitHub. We added a button to the ontology homepage within the TS, which users can utilize to open an issue form, enter the issue content and title, and subsequently submit.

We categorized these issues into two groups: Generic Issues and Term Requests. The distinction was made because certain repositories, such as VIBSO, employ a specific template for filing new term requests. To uphold the integrity of this format, when a user selects the Term Request option, the TS automatically populates the input area with the appropriate template, thus informing users of its existence and encouraging its use.

5.2.3 Notes on Ontologies

As previously discussed, the capability of the Terminology Service (TS) to facilitate users in making or reviewing notes on ontologies was identified as an essential feature during the development of VIBSO. Such notes create an additional layer of context information atop the ontology or ontology terms, thereby benefiting both ontology developers and users. These notes serve as a dynamic feedback system that can either be used in cases where another tracker system is missing or where opening an issue is not desired. At the same time, they foster a deeper understanding of the ontology, by functioning as another avenue for documentation.

In pursuit of these advantages, we have developed a feature within TS that allows users to annotate ontologies with notes. To utilize this feature, a user navigates to the newly introduced "Note" tab in the ontology overview. Here, the user can add a new note, which requires a title, a body of text, and the Internationalized Resource Identifier (IRI) of the target artifact. The ability to specify target artifacts empowers users to apply notes to specific elements such as classes. At present, there are four possible target artifacts: Ontology, Class, Property, and Individual. Additionally, users can view a list of existing notes under the Note tab on the ontology page within TS.

Currently, this feature is in its prototype stage on our demo server. For its first stable release, we intend to enhance its usability by introducing more functionalities. For instance, we will enable users to filter the note list based on the target artifact. Furthermore, we will provide users with the ability to comment on a note about an ontology, thus opening up possibilities for discussion.

5.2.4 List View For Classes

One of the insights we gleaned was the necessity for the class view as a list presentation, in addition to the traditional tree view for ontology terms. We observed that the tree view could occasionally be perplexing for some novice users. Furthermore, certain operations, such as sorting based on the term ID, are not feasible within the tree view of classes.

As a response to these findings, we developed a paginated list view that exists alongside the tree view. This provides users with a comprehensive, tabular perspective of all the classes within an ontology. The list view is highly detailed, incorporating expansive metadata about terms, such as descriptions, relationships, and author comments.

Moreover, we have incorporated a 'jump-to' functionality within the list view. This allows users to quickly navigate to a specific term in the list by searching for its name, thereby enhancing the usability and efficiency of the platform.

5.2.5 UI enhancements

The final aspect of our implementation, based on the lessons learned, involved refining various UI-related features within our front-end application. The most common feedback pertained to the tree view page, where we present the term tree view alongside the term detail table when a term is selected.

The first concern addressed the challenge of navigating the tree solely using mouse clicks. To overcome this, we introduced keyboard arrow key functionality for navigation. Users can now use the right and left arrow keys to expand or collapse a node in the tree, while the up and down arrow keys allow movement between tree nodes.

The second enhancement dealt with the size of the detail table. Feedback suggested it was rather small for terms with extensive metadata. To improve this, we made the view pane resizable. This allows users to adjust the sizes of the tree view and detail table as needed, enhancing the readability of metadata and overall user experience.

6. Discussion

The topic of ontologies, with its abstract nature and philosophical undertones (ontology translates to 'doctrine of being'), often presents a challenging concept for domain experts to grasp. Understanding what terminologies or ontologies are, and why they are essential within their respective research areas, can be a complex process. The realization that a formal (i.e., not grounded in familiar classical/natural language) representation of knowledge (via ontologies) is necessary for machines to capture, comprehend, and interpret knowledge typically comes only after an extensive process of understanding. However, the importance and practicality of terminologies can be quite straightforward to demonstrate, especially in the context of the semantic description of research data through metadata annotation. In such a scenario, the Terminology Service (TS) can serve as an effective gateway tool, providing a swift and simple overview of the available ontologies. This tool can assist domain experts, for example, by providing domain-specific terminology collections that aid in the search and selection of suitable terms for their respective use cases. Further simplifying features, such as a list view, can provide even more support, enhancing the usability and efficiency of the service.

In this work, we employed a user-centered approach to identify the challenges and requirements in the design and development of a Terminology Service aimed at enhancing ontology development. We incorporated our target user group into the TS development process from its inception. In particular, we engaged in close contact with ontology development teams such as VIBSO, ensuring their perspectives and insights were woven into the fabric of the service from the earliest stages.

This collaboration allowed us to gain a deeper understanding of the ontology development process and the requisite features for an effective Terminology Service. We greatly valued the continuous feedback received from the ontology engineers, which proved crucial in shaping our service. Additionally, we endeavored to incorporate data annotators, who are primary users of the ontology, into our development process, thereby ensuring we cater to their needs effectively.

We facilitated numerous events and meetings, including conferences, workshops, and online sessions, with two primary objectives. Firstly, we sought to introduce our TS and demonstrate how it could be utilized for data annotation. Secondly, we aimed to gather feedback and user requirements to improve the quality of the ontologies and their presentation within the TS.

These close interactions with our users were instrumental in helping us conceptualize the ideal TS and identify its essential features. The insights gleaned from these engagements were directly translated into the design and functionality of our service, ensuring it adequately meets user needs and expectations.

Facilitating FAIR data annotation for researchers necessitates the provision of not just any ontologies, but those of high quality. Therefore, an ideal Terminology Service must cater to the needs of ontology developers, thereby enhancing the development process and ensuring quality. Our central finding was the necessity of integrating TS into the entire development process, from inception to conclusion. This implies the TS should address developer needs at various stages, such as artifact browsing and quality check, through discussion, feedback, and notes.

The ability of TS to offer these features offers multiple benefits to ontology developers. Firstly, it facilitates continuous feedback via Git issues and TS notes, ensuring the quality of developed ontologies. Furthermore, having notes and Git issues allows other TS users and less experienced developers to stay current with the latest discussions and topics concerning ontologies. This adds a new layer of context around the target ontology, enhancing its comprehensibility. Moreover, incorporating these ontology development tools into TS reduces context switching between different systems and environments, which could negatively impact a developer's performance.

Finally, we noted that the usability of the TS user interface (UI) is critical for ontology developers. There's a substantial demand for presenting a list view of classes alongside a tree view for an ontology. Given the high volume of terms in many ontologies, it is crucial for developers to swiftly navigate through the tree view using keyboard arrow keys. Additionally, due to the challenge of reading extensive metadata for a term in limited screen space, it is essential to allow TS users to resize different information box panes, thereby improving readability.

7. Conclusion

We have presented a user-centered development of our Terminology Service in the realm of research data annotation, where we actively engaged ontology developers, knowledge engineers, domain experts, and data annotators in the development process from its inception. Their continuous feedback and insights during its usage in an ongoing ontology development process proved instrumental in identifying requirements for the TS.

The added value of the TS by integrating it throughout the ontology development process is a central finding. The inclusion of features such as GitHub issue tracking, term requests, and ontology notes within the TS streamlines the development workflow, reduces context switching, and promotes collaboration among ontology developers and users.

Usability improvements have been another key focus. The introduction of an enhanced list view alongside the traditional tree view provides users with a comprehensive overview of ontology classes, enhancing efficiency and ease of navigation. The incorporation of keyboard arrow functionality and resizable information panes further improves the user interface, allowing for more intuitive interaction.

In conclusion, this work demonstrates the value of the TS for ontology developers and data annotators and proves its potential as an effective tool for ontology design and curation.

The next steps will include enhancing GitHub interaction features and the TS Note system to incorporate ontology curation into the terminology service user interface to establish the TS as an integrated ontology curation tool. We will continue the user-driven development to further support curation workflows from the very beginning. As ontologies continue to play a vital role in knowledge representation and data interoperability, the development and refinement of effective Terminology Services will contribute to advancing scientific understanding and collaboration across diverse domains.

Acknowledgements

We thank all PIs and developers at TIB who contribute to the TIB Terminology Service.

The presented work was conducted as part of the NFDI4Chem project (DFG project no. 441958208). The authors would like to thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for funding and support.

References

[1] "Re3Data," *Registry of Research Data Repositories*. [Online]. Available: https://www.re3data.org. [Accessed: May 30, 2023]

[2] S.-A. Sansone *et al.*, "FAIRsharing as a community approach to standards, repositories and policies," *Nat. Biotechnol.*, vol. 37, no. 4, pp. 358–367, Apr. 2019, doi: 10.1038/s41587-019-0080-8. [Online]. Available: http://dx.doi.org/10.1038/s41587-019-0080-8

[3] "EOSC Portal," *EOSC Portal*, Apr. 14, 2023. [Online]. Available: https://eosc-portal.eu/. [Accessed: May 29, 2023]

[4] N. Hartl, E. Wössner, and Y. Sure-Vetter, "Nationale Forschungsdateninfrastruktur (NFDI)," *Informatik Spektrum*, vol. 44, no. 5, pp. 370–373, Oct. 2021, doi: 10.1007/s00287-021-01392-6. [Online]. Available: https://doi.org/10.1007/s00287-021-01392-6

[5] S. Kanza, C. Willoughby, N. J. Knight, C. L. Bird, J. G. Frey, and S. J. Coles, "Digital research environments: a requirements analysis," *Digital Discovery*, Apr. 2023, doi: 10.1039/D2DD00121G. [Online]. Available: https://pubs.rsc.org/en/content/articlelanding/2023/DD/D2DD00121G. [Accessed: May 29, 2023]

[6] M. D. Wilkinson *et al.*, "The FAIR Guiding Principles for scientific data management and stewardship,"

Sci Data, vol. 3, p. 160018, Mar. 2016, doi: 10.1038/sdata.2016.18. [Online]. Available: http://dx.doi.org/10.1038/sdata.2016.18
[7] M. Martone and Others, “Data citation synthesis group: Joint declaration of data citation principles,” *San Diego CA: Force11. https://www. force11.org/group/joint-declaration-datacitafion-principles-final*, doi: 10.25490/a97f-egyk. [Online]. Available: http://dx.doi.org/10.25490/a97f-egyk
[8] “DataCite Schema,” *DataCite Schema*. [Online]. Available: http://schema.datacite.org/. [Accessed: May 30, 2023]
[9] “DCMI Metadata Terms,” *Dublin Core*. [Online]. Available: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/. [Accessed: May 29, 2023]
[10] S. Kunis, S. Hänsch, C. Schmidt, F. Wong, C. Strambio-De-Castillia, and S. Weidtkamp-Peters, “MDEmic: a metadata annotation tool to facilitate management of FAIR image data in the bioimaging community,” *Nat. Methods*, vol. 18, no. 12, pp. 1416–1417, Dec. 2021, doi: 10.1038/s41592-021-01288-z. [Online]. Available: http://dx.doi.org/10.1038/s41592-021-01288-z
[11] S. Jupp, T. Burdett, C. Leroy, and H. Parkinson, “A new Ontology Lookup Service at EMBL-EBI,” *Workshop on Semantic Web Applications and Tools for Life Sciences*, 2015 [Online]. Available: https://www.semanticscholar.org/paper/b83bfbfc1f2f08e5b88af5ef65ef2a8687ac4112. [Accessed: May 11, 2023]
[12] M. A. Musen *et al.*, “The center for expanded data annotation and retrieval,” *J. Am. Med. Inform. Assoc.*, vol. 22, no. 6, pp. 1148–1152, Nov. 2015, doi: 10.1093/jamia/ocv048. [Online]. Available: http://dx.doi.org/10.1093/jamia/ocv048
[13] N. Noy *et al.*, “BioPortal: A Web Repository for Biomedical Ontologies and Data Resources,” *International Workshop on the Semantic Web*, 2008 [Online]. Available: https://www.semanticscholar.org/paper/cf2202e842500442eaea4c218425c98d599ee481. [Accessed: May 11, 2023]
[14] J. Pleiss, “Standardized data, scalable documentation, sustainable storage – EnzymeML as A basis for FAIR data management in biocatalysis,” *ChemCatChem*, vol. 13, no. 18, pp. 3909–3913, Sep. 2021, doi: 10.1002/cctc.202100822. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/cctc.202100822
[15] *dataverse-external-vocab-support: Scripts and material related to using external vocabulary services with Dataverse*. Github [Online]. Available: https://github.com/gdcc/dataverse-external-vocab-support. [Accessed: May 29, 2023]
[16] P. Tremouilhac *et al.*, “Chemotion ELN: an Open Source electronic lab notebook for chemists in academia,” *J. Cheminform.*, vol. 9, no. 1, p. 54, Sep. 2017, doi: 10.1186/s13321-017-0240-0. [Online]. Available: http://dx.doi.org/10.1186/s13321-017-0240-0
[17] A. Ledl, “Das BAsel Register of Thesauri, Ontologies & Classifications (BARTOC),” presented at the 104. Deutscher Bibliothekartag, Nürnberg, 2015 [Online]. Available: https://opus4.kobv.de/opus4-bib-info/files/1677/Poster_BARTOC_DBT_2015.pdf. [Accessed: May 11, 2023]
[18] J. Voß, J. M. Agne, U. Balakrishnan, and M. Akter, “Terminology Registries and Services,” Nov. 2016, doi: 10.5281/zenodo.166717 [Online]. Available: https://zenodo.org/record/166717
[19] S.-Y. Yang, “OntoPortal: An ontology-supported portal architecture with linguistically enhanced and focused crawler technologies,” *Expert Syst. Appl.*, vol. 36, no. 6, pp. 10148–10157, Aug. 2009, doi: 10.1016/j.eswa.2009.01.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417409000402
[20] S. Jupp, T. Burdett, C. Leroy, and H. E. Parkinson, “A new Ontology Lookup Service at EMBL-EBI,” in *SWAT4LS*, 2015.
[21] “SKOS Simple Knowledge Organization System Reference.” [Online]. Available: https://www.w3.org/TR/skos-reference/. [Accessed: May 11, 2023]
[22] J. Frey, D. Streitmatter, F. Götz, S. Hellmann, and N. Arndt, “DBpedia Archivo: A Web-Scale Interface for Ontology Archiving Under Consumer-Oriented Aspects,” in *Semantic Systems. In the Era of Knowledge Graphs*, 2020, pp. 19–35, doi: 10.1007/978-3-030-59833-4_2 [Online]. Available: http://dx.doi.org/10.1007/978-3-030-59833-4_2
[23] P. Y. Vandenbussche and G. A. Atemezing, “Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web,” *Semant. Pragmat.*, 2017 [Online]. Available: https://content.iospress.com/articles/semantic-web/sw213
[24] M. A. Musen and Protégé Team, “The Protégé Project: A Look Back and a Look Forward,” *AI Matters*, vol. 1, no. 4, pp. 4–12, Jun. 2015, doi: 10.1145/2757001.2757003. [Online]. Available: http://dx.doi.org/10.1145/2757001.2757003
[25] V. Wiens, S. Lohmann, and S. Auer, “WebVOWL Editor: Device-Independent Visual Ontology Modeling,” in *International Semantic Web Conference (P&D/Industry/BlueSky)*, 2018 [Online]. Available: http://ceur-ws.org/Vol-2180/paper-75.pdf
[26] *kgcl: Datamodel for KGCL (Knowledge Graph Change Language)*. Github [Online]. Available:

https://github.com/INCATools/kgcl. [Accessed: May 30, 2023]

[27] R. C. Jackson, J. P. Balhoff, E. Douglass, N. L. Harris, C. J. Mungall, and J. A. Overton, "ROBOT: A Tool for Automating Ontology Workflows," *BMC Bioinformatics*, vol. 20, no. 1, p. 407, Jul. 2019, doi: 10.1186/s12859-019-3002-3. [Online]. Available: http://dx.doi.org/10.1186/s12859-019-3002-3

[28] N. Matentzoglu *et al.*, "Ontology Development Kit: a toolkit for building, maintaining and standardizing biomedical ontologies," *Database* , vol. 2022, Oct. 2022, doi: 10.1093/database/baac087. [Online]. Available: http://dx.doi.org/10.1093/database/baac087

[29] C. Steinbeck *et al.*, "NFDI4Chem - Towards a National Research Data Infrastructure for Chemistry in Germany," *Research Ideas and Outcomes*, vol. 6, p. e55852, Jun. 2020, doi: 10.3897/rio.6.e55852. [Online]. Available: https://riojournal.com/article/55852/

[30] R. C. Jackson *et al.*, "OBO Foundry in 2021: Operationalizing Open Data Principles to Evaluate Ontologies," *bioRxiv*, p. 2021.06.01.446587, Jun. 02, 2021 [Online]. Available: https://www.biorxiv.org/content/10.1101/2021.06.01.446587v1. [Accessed: Sep. 08, 2021]

[31] R. Arp, B. Smith, and D. A. Spear, *Building Ontologies with Basic Formal Ontology*. Cambride, London: MIT Press, 2015.

[32] B. Smith *et al.*, "Relations in biomedical ontologies," *Genome Biol.*, vol. 6, no. 5, p. R46, Apr. 2005, doi: 10.1186/gb-2005-6-5-r46. [Online]. Available: http://dx.doi.org/10.1186/gb-2005-6-5-r46

[33] A. Bandrowski *et al.*, "The Ontology for Biomedical Investigations," *PLoS One*, vol. 11, no. 4, p. e0154556, Apr. 2016, doi: 10.1371/journal.pone.0154556. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0154556

[34] R. S. C. Ontologies, *rsc-ontologies - Overview*. Github [Online]. Available: https://github.com/rsc-ontologies. [Accessed: May 24, 2023]

[35] B. Smith *et al.*, "The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration," *Nat. Biotechnol.*, vol. 25, no. 11, pp. 1251–1255, Nov. 2007, doi: 10.1038/nbt1346. [Online]. Available: http://dx.doi.org/10.1038/nbt1346

[36] "H2020 CHARISMA," *H2020 CHARISMA*. [Online]. Available: https://www.h2020charisma.eu/. [Accessed: May 29, 2023]

[37] "NFDI4Chem Terminology Service," *NFDI4Chem Terminology Service*. [Online]. Available: https://terminology.nfdi4chem.de. [Accessed: May 31, 2023]

[38] P. Strömert, J. Hunold, A. Castro, S. Neumann, and O. Koepler, "Ontologies4Chem: the landscape of ontologies in chemistry," *Pure Appl. Chem.*, Mar. 2022, doi: 10.1515/pac-2021-2007. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/pac-2021-2007/html. [Accessed: May 02, 2022]

[39] P. Strömert, J. Hunold, and O. Koepler, "1st Ontologies4Chem Workshop – Ontologies for chemistry," Sep. 07, 2022. [Online]. Available: https://www.confident-conference.org/index.php/Event:13b31d0e-f398-4bc8-a8c2-54352b1dcc66. [Accessed: Sep. 28, 2022]

[40] *TIB-Terminology-Service-Frontend-2.0: Frontend source code for NFDI4Chem, NFDI4Ing, TIB TS v2.0*. Github [Online]. Available: https://github.com/TIBHannover/TIB-Terminology-Service-Frontend-2.0. [Accessed: May 31, 2023]