Towards VocBench 3: Pushing Collaborative Development of Thesauri and Ontologies Further Beyond

Armando Stellato[1], Andrea Turbati[1], Manuel Fiorelli[1], Tiziano Lorenzetti[1], Eugeniu Costetchi[2], Christine Laaboudi[2], Willem Van Gemert[2], Johannes Keizer[3]

[1] ART Group, Dept. of Enterprise Engineering
University of Rome Tor Vergata
Via del Politecnico 1, 00133 Rome, Italy
{turbati,fiorelli}@info.uniroma2.it
stellato@uniroma2.it, tiziano.lorenzetti@gmail.com

[2] Publications Office of the European Union
Dissemination and Reuse Directorate, Documentary Management and Metadata Unit
2985 Luxembourg, LUXEMBOURG
{christine.laaboudi,willem.van-gemert}@publications.europa.eu
costezki.eugen@gmail.com

[3] GODAN secretariat, c/o CABI Head Office
Nosworthy Way, Wallingford, Oxfordshire, OX10 8DE, UK
johannes.keizer@gmail.com

Abstract. More than three years have passed since the release of the second edition of VocBench, an open source collaborative web platform for the development of thesauri complying with Semantic Web standards. In these years, a vibrant user community has gathered around the system, consisting of public organizations, companies and independent users looking for open source solutions for maintaining their thesauri, code lists and authority resources. The focus on collaboration, the differentiation of user roles and the workflow management for content validation and publication have been the strengths of the platform, especially for those organizations requiring a centralized and controlled publication environment. Now the time has come to widen the scope of the platform: funded by the ISA^2 programme of the European Commission, VocBench 3 will offer a general-purpose collaborative environment for development of any kind of RDF dataset, improving the editing capabilities of its predecessor, while still maintaining the peculiar aspects that determined its success. In this paper, we review the requirements and the new objectives set for version 3, and then introduce the new characteristics that were implemented for this next iteration of the platform.

Keywords: Collaborative Editing, Ontologies, Thesauri, OWL, SKOS

1 Introduction

In 2008 the group for Agriculture Information Management Standards of the Food and Agriculture Organization of the United Nations (FAO, `http://www.fao.org/`) developed a collaborative platform for collaboratively managing their Agrovoc thesaurus [1].

The so-called "Agrovoc Workbench" soon met the interest of other FAO departments and several other organizations interested in open source solutions for collaborative thesauri development.

Baptized with a new name: "VocBench" – suggesting a more general environment for thesauri management – the platform later had been strongly re-engineered in the context of a collaboration between FAO and the ART group of the University of Rome Tor Vergata (`http://art.uniroma2.it`). The result of this collaboration, VocBench 2 [2] had been rethought as a fully-fledged collaborative platform for thesaurus management, freely available and open-sourced, offering native RDF support for SKOS [3] and SKOS-XL [4] knowledge organization systems [5], while retaining from its original version the focus on multilingualism, collaboration and a structured content validation & publication workflow. The possibility for project administrators to define roles with very specific capabilities and to assign them to different users according to their proficiencies and authorizations, the publication workflow where dedicated users could supervise the work of others and accept their modifications, have been the very strengths of the platform, especially for those organizations requiring a centralized and controlled publication environment. Consistently with this controlled approach, what some other users felt as mostly missing from the system was more freedom on data shaping, longing for unrestricted capabilities for editing data at its very core, as in triple-oriented RDF editing environments.

There is indeed a trade-off in modeling systems between flexibility and control, which is non-trivial to overcome. Controlling the actions that users can perform implies recognizing them as high-level performatives, associated to pre-defined modeling graph patterns and described by rich metadata that can be used to identify them, to authorize their invocation by classes of users, to properly store them in validation stacks, etc… This complex castle is thus usually not meant to be maintained brick-by-brick: the various aspects described above and the dependencies among them are easier to be managed by identifying clear "first-class citizens" (entities recognized by the system and on which operations can be performed) and a pre-defined set of operations (possibly, non-overlapping) on them. The final touch is to implement this higher layer of representation in terms of lower-level aspects of the model/technology (i.e. triples in RDF).

It is thus with a new incarnation of the system, VocBench 3 (or, simply, VB3), that we planned to overcome the above limitations, while widening the original scope of the platform to a general-purpose collaborative environment for development of SKOS thesauri, OWL ontologies and RDF datasets in general. VocBench 3 is funded by Action 1.1 of the ISA^2 Programme of the European Commission for "Interoperability solutions for public administrations, businesses and citizens" (`https://ec.europa.eu/isa2/`). The action is managed by the Publications Office of the European Union (`https://publications.europa.eu/`). VB3 was developed in close collaboration with the ART group of the University of Rome Tor Vergata, the same group that contributed to the development of the second version of the platform. At the time of writing, VocBench 3 has been released to the Publications Office for internal evaluation and acceptance testing, while general availability is planned by September 2017. The homepage of VocBench (`http://vocbench.uniroma2.it/`) already points to the sources of VB3, and it will provide documentation and downloadable artifacts for VB3, when it becomes available to the general public.

In this paper, we review the original requirements of the platform and the new objectives set for version 3, and then introduce the new characteristics that were implemented to meet the goals for this next iteration of the platform. Our aim is thus to highlight the improvements over VB2, while we refer to [2] both for an analysis of related works showing the motivation of the VocBench platform and for architectural insights.

2 Requirements

In this section, we list the foundational stones upon which VocBench 3 was developed, in terms of the requirements to be met. The requirements of VocBench (R1-R7) still hold, but have been reformulated by accounting for the widened scope of the platform and its improved editing capabilities. Other requirements have also been added, following proposals by its developing team and requests by stakeholders presented in dedicated stakeholder meetings.

R1. Multilingualism. Properly characterizing knowledge resources in different natural languages is fundamental. This especially holds for thesauri, due to their use in Information Retrieval, though the overall importance of elaborated lexicalizations is progressively gaining momentum, thanks to data publication initiatives such as the Linguistic Linked Open Data (LLOD: `http://linguistic-lod.org/`) and to models for Ontology-Lexicon interfaces, such as Lemon [6] and its most recent specification OntoLex Lemon, realized in the context of the homonymous W3C community group `https://www.w3.org/community/ontolex/`.

R2. Controlled Collaboration. One of the key features of the system has to be enabling collaboration on a large scale: several, distributed users, have to be able to collaborate remotely on a same project. Opening up to communities is important, though the development of authoritative resources demands for the presence of some control to be exerted over the resource lifecycle: for this reason, users must be granted different access levels, and some of them should be granted the possibility to validate other users' work.

R3. Data Interoperability and Consistency. Interoperability of several resources critically depends on data integrity and conformance to representation standards. However, flexible models such as SKOS translate to underspecified possibilities on the one hand, and formal constraints – beyond the expressiveness of OWL – on the other one. It is thus important that VocBench enforces a consistent use of these models, by preventing the editors from generating invalid data, and by providing "fixing facilities" for spurious data acquired from external sources. Finally, support for alignment to other datasets is also an interoperability must for the Linked Data World.

R4. Software Interoperability/Extensibility. The system should be able to interact with (possibly interchangeable) standard technologies in the RDF/Linked Data world.

R5. Data Scalability. The system must deal with (relatively) large amount of data, still offering a friendly environment. The User Interface must consider this requirement by appropriately subdividing data loading into subsequent requests and implementing dedicated solutions for large results.

R6. Under-the-hood data access/modification. While a friendly UI for content managers/domain experts is important, knowledge engineers need to access raw data beyond the usual front-ends, as well as to benefit from mass editing/refactoring facilities.

R7. Adaptive Context and Ease-of-use. In migrating from the first VocBench to its second version, it was mandatory that different users, ranging from ordinary editors to system administrators, shared an easy and comfortable user experience. The new VB3 should provide an even smoother experience, with very low installation requirements and an as-short-as-possible time-to-use. Whether (and proportionally if) the user is an administrator configuring the system, a project manager configuring a project, a user requesting registration and connection to a given project, or a new user willing to test the system as a desktop tool without settings and configuration hassle, the platform should respond adaptively to their needs.

R8. RDF Languages Support. Differently from both its predecessors, dealing with thesauri only, VB3 has to offer native support for SKOS thesauri, OWL ontologies, and RDF datasets in general.

R9. Maintainability (Architecture and Code Scalability). In special mode, the ability to meet new requirements, cope with changed environments and make future maintenance easier. A weak spot of VB2, in VB3 it is mandatory to be able to add new services, functionalities, plugins etc... without the fabric of the system being altered or too much effort being required in order to align these new elements with all the characteristics of the system, such as validation, history management, roles and capabilities.

R10. Full Editing Capability (RDF Observability and Reachability). Any complex RDF construct should always be inspectable and modifiable by users (providing they have the proper authorization) even in its finer details. While the platform can provide high-level performatives for conveniently creating/modifying complex descriptions of resources according to pre-defined modeling design patterns (i.e. by using RDF graph patterns with variables being instantiated upon usage), the user should never be prevented from inspecting/altering these elements.

R11. Provenance. Actions in VB3 should be handled as first-class citizens themselves, being identified and qualified by proper metadata, including information about which user performed an action, when they did it, which parameters have influenced its performance, etc… Metadata answering to the five "Ws" (with the possible exception of the "why") should provide all information for tracking the origin of an action.

R12. Versioning Support. Besides history and validation mechanisms, providing in-detail reports on the single actions performed by users, it should be possible to take periodic snapshots of the status of a dataset.

R13. Metadata Descriptions. In order for the Semantic Web to fully achieve its vision, linked open data has to speak about itself [7]. This means not only having data modeled according to well-known shared vocabularies, but to be able to grasp meaningful information about a dataset without having to dig into its content.

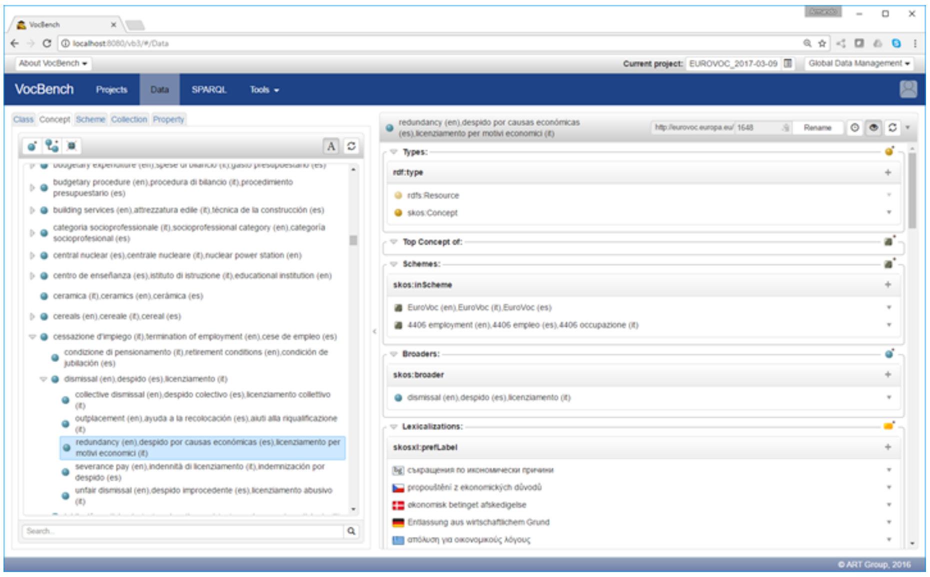
R14. Customizable UI. UI based on ontology analysis are limited by the axiomatic description of the resources they show and of their types, ignoring possible desiderata of the user. VB3 should allow users to represent the information that they want to specify at resource creation, per resource type, so that it will be prompted to the user. Connected technical aspects, such as proper transformation of the user input into serializable RDF content, should also be tackled.

R15. Everything's RDF. Whereas VB2 used a database to store user&project management information, history & validation information; VB3 should follow a more uniform approach, adopting RDF for virtually any information that needs to be stored.

3 Towards VocBench 3

In this section, we discuss the main characteristics of the software that allowed meeting the aforementioned requirements.

User Interface (UI). The UI is the first element that vividly marks the difference between VB3 and its predecessors. The user interface has been rebuilt from scratch, by using different technologies – notably Angular (https://angular.io/) in place of Google Web Toolkit (http://www.gwtproject.org/) – and reorganizing the user experience. The Data view, letting the user explore the project's dataset, is the one that mostly represents these changes. The several tabs of VocBench 2 (which inherited and extended the tab-based model of VocBench 1) that populated the "concept details" panel have been replaced with a single component, called *resource-view* (see Figure 1). The resource-view offers a complete overview of resources of any type, thus remarking another difference with respect to VB1&VB2: there are no first-class citizen resources, such as concepts and (SKOS-XL) labels; in fact, all resources now can be viewed and edited through the resource-view section. The resource-view is a general component that can be specialized depending on the inspected resource: a few sections are shared among all resources, such as *types*, listing the rdf:types for the described resource, *lexicalizations*, listing all the available lexicalizations and *properties*, listing all general properties not addressed by the above *descriptors*, while others are specific to the inspected resource. For instance, the resource-view centered on a concept is composed of the following sections: *types*, *top concepts*, *schemes*, *broaders*, *lexicalizations*, *notes* and the final generic *property* section. While the mapping of these sections to properties of the core modeling vocabularies is trivial (e.g. *types* to rdf:type, *schemes* to skos:inScheme and so on) the sections are however presented with a predicate-object style in order to qualify the predicate, as they might include user-defined or domain-specific subproperties of the above ones. It is worth of note that the general applicability of the resource-view to any resource and the possibility to edit any of their details from it concur to satisfy requirement R10. A special mention goes to the *lexicalizations* section: it represents an abstraction over different kind of properties, and offers specific resolution of their shape, always showing the form of the lexicalization. In VB3, the concept of *lexical model* has been introduced (and separated from the *knowledge model*, e.g. OWL or SKOS) so that, for instance, it is possible to select SKOS-XL (the lexical



budgetary expenditure (en),spese di bilancio (it),gasto presupuestario (es)
budgetary procedure (en),procedura di bilancio (it),procedimiento presupuestario (es)
building services (en),attrezzatura edile (it),técnica de la construcción (es)
categoria socioprofessionale (it),socioprofessional category (en),categoría socioprofesional (es)
central nuclear (es),centrale nucleare (it),nuclear power station (en)
centro de enseñanza (es),istituto di istruzione (it),educational institution (en)
ceramica (it),ceramics (en),cerámica (es)
cereals (en),cereale (it),cereal (es)
cessazione d'impiego (it),termination of employment (en),cese de empleo (es)
condizione di pensionamento (it),retirement conditions (en),condición de jubilación (es)
dismissal (en),despido (es),licenziamento (it)
collective dismissal (en),despido colectivo (es),licenziamento collettivo (it)
outplacement (en),ayuda a la recolocación (es),aiuti alla riqualificazione (it)
redundancy (en),despido por causas económicas (es),licenziamento per motivi economici (it)
severance pay (en),indennità di licenziamento (it),indemnización por despido (es)
unfair dismissal (en),despido improcedente (es),licenziamento abusivo (it)

redundancy (en),despido por causas económicas (es),licenziamento per motivi economici (it)
http://eurovoc.europa.eu/ 1648

Types:
rdf:type
rdfs:Resource
skos:Concept

Top Concept of:

Schemes:
skos:inScheme
EuroVoc (en),EuroVoc (it),EuroVoc (es)
4406 employment (en),4406 empleo (es),4406 occupazione (it)

Broaders:
skos:broader
dismissal (en),despido (es),licenziamento (it)

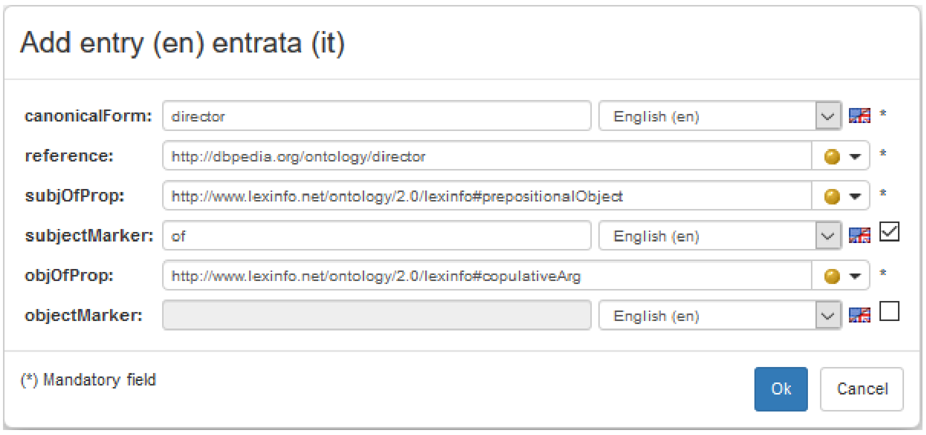
Lexicalizations:
skosxl:prefLabel
bg съкращения по икономически причини
propouštění z ekonomických důvodů
økonomisk betinget afskedigelse
Entlassung aus wirtschaftlichem Grund
απόλυση για οικονομικούς λόγους

Figure 1. VocBench UI showing EuroVoc (`http://eurovoc.europa.eu`)

extension to SKOS allowing for reified labels) as a lexical model for both OWL ontologies and SKOS thesauri. In SKOS-XL, the lexicalizations section would provide dedicated forms for creating reified labels (with the possibility to specify their URI or to having it assigned automatically) and directly show their skosxl:literalform in the object field. The object field is also clickable and it opens a *resource-view* focused on the resource description of the SKOS-XL label. This revised model greatly improves the coverage of requirement R1 by covering not only diverse natural languages, but the different formal languages in which linguistic information can be coded.

The overall data view (Figure 1) is similar in principle to the one of the previous editions, with the resource structures on the left (previously, only the concept tree), and the description of the selected resource on the right. The resource structures is composed of different tabs, depending on the chosen modeling vocabulary. OWL offers two tabs with a class tree & instance list and a property tree respectively, while SKOS adds to them a concept tree, a list of schemes and a collection tree (with the tree showing the containment relation between different SKOS collections). VB1&2 showed concepts through their labels in all of the selected languages for visualization. In VB3, an option allows to toggle between the URIs/qnames of the resources, and the string composed by a *resource-renderer*. Resource renderers provide human-friendly visualizations of resources. Different renderers can be connected to the system as plugins to its dedicated *rendering* extension point. The default renderer behaves in a way similar to VB1&2, showing labels in all of the selected languages for visualization (again, R1).

An important new aspect of the user interface is offered by *Custom Forms*, a flexible data-driven form definition mechanism that we devised for VocBench, allowing users to perform a declarative specification of the key elements that concur to the creation of



Add entry (en) entrata (it)

canonicalForm:	director	English (en)	*
reference:	http://dbpedia.org/ontology/director		*
subjOfProp:	http://www.lexinfo.net/ontology/2.0/lexinfo#prepositionalObject		*
subjectMarker:	of	English (en)	☑
objOfProp:	http://www.lexinfo.net/ontology/2.0/lexinfo#copulativeArg		*
objectMarker:		English (en)	☐

(*) Mandatory field

Figure 2. Custom Form for a relational noun in the W3C OntoLex Lemon model

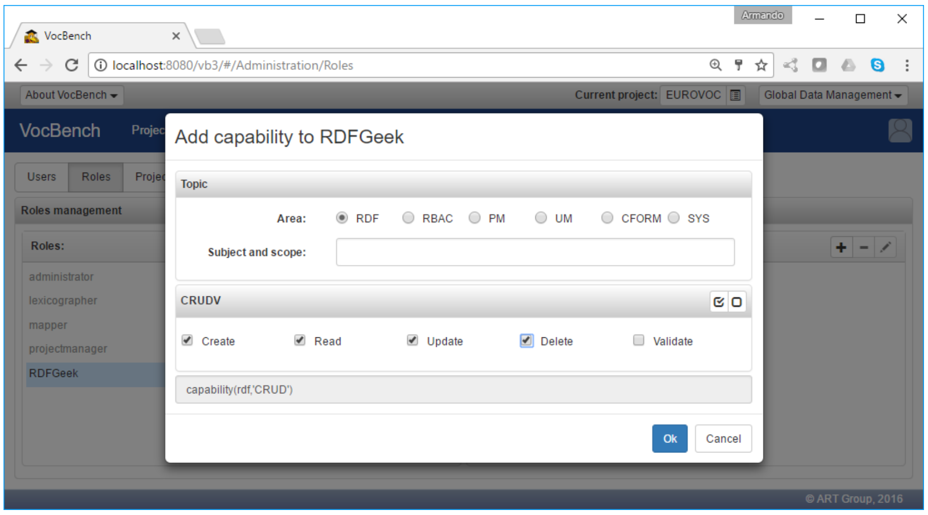
a complex RDF resource (satisfying req. R14). In particular, custom forms rely on the combination of the following four key elements:

- a declaration of the data that is expected to be prompted by the user
- a series of transformations that have to be applied to the prompted data in order to produce valid RDF entities to be stored
- the organization of the produced RDF entities into meaningful graph patterns, instantiating the template of the resource to be created
- the automatic production of a form layout (see Figure 2) based on the above declarations information that is required for "constructing" a new resource

Custom Forms have been described more in details in [8], which analyzed and evaluated their expressive power by applying them to the use case of representing entities for the W3C OntoLex Lemon (`http://www.w3.org/2016/05/ontolex/`) vocabulary.

Another relevant difference in the UI offer lies in the *Project* page: system Administrators (and other users having equivalent authorizations) can inspect projects in all of their details, and easily switch from one to the other, while other users are offered the traditional project list allowing access to only the projects they are registered to. This is particularly convenient for users willing to use VB3 as a desktop tool: in less than 2 minutes, it is possible to start the system for the first time, configure a simple user with default minimal information and administrative rights, log in and seamlessly use the VocBench without the impression of dealing with a cumbersome web juggernaut.

Controlled Collaborative Editing through Role-based Access Control (RBAC). A single installation of VocBench can handle multiple projects, which can also be interlinked for mutual data access (e.g. for purpose of alignment). VocBench promotes the separation of responsibilities through a role-based access control mechanism, checking user privileges for requested functionalities through the role they assume (req. R2). Upon registration, users indicate their personal information, their proficiencies. The proficiencies are obviously the user's declaration, so they do not grant any permission



Add capability to RDFGeek

Topic

Area: [✓ RDF] RBAC PM UM CFORM SYS

Subject and scope:

CRUDV

[✓ Create] [✓ Read] [✓ Update] [✓ Delete] ☐ Validate

capability(rdf,'CRUD')

Ok Cancel

Roles: administrator, lexicographer, mapper, projectmanager, RDFGeek

Figure 3. Editing a capability for a new role in VocBench

per se, but can help *administrators* and *project managers* (users with the role of administering a single project) in selecting users to assign to their project, or trivially by simplifying the assignment of capabilities to them by reusing their declared proficiencies as a template. In VB3, we have completely re-designed the mechanism for roles/capabilities. While VB2 had hard-wired roles with predefined and limited editing possibilities, which do not easily scale-up to possible extensions of the system (req. R9), in VB3 we have defined a simple language for specifying capabilities in terms of *area*, *subjects* and *scopes* E.g. the expression:

auth(rdf(datatypeProperty, taxonomy), 'R')

corresponds to the authorization for being able to read taxonomical information about datatype properties. The 'R' stands for READ, as in the CRUD paradigm, rdf is the *area* of the requested capability while datatypeProperty and taxonomy define the *subject* and *scope* respectively of the capability.

The language is implemented as a series of facts for the Prolog [9] logic programming language. Entailments are guaranteed thanks to rules written in Prolog (which may be extended by users), e.g. the expression: rdf entails any rdf(_) or rdf(_,_) expression, that is any monadic or diadic expression with the rdf predicate (i.e. implying that the simple expression rdf authorizes any operation in the *area* of RDF). The computation of entailments is based on the tuProlog [10] engine.

New roles can be easily created, and existing ones can be modified, through a dedicated *rbac* editing wizard (Figure 3). The default policy recognizes typical roles and their acknowledged responsibilities:

- *Administrator*: the sole inter-project role (i.e. the role exists a-priori from projects). The Administrator has by definition access to all functionalities and configuration options of the system.

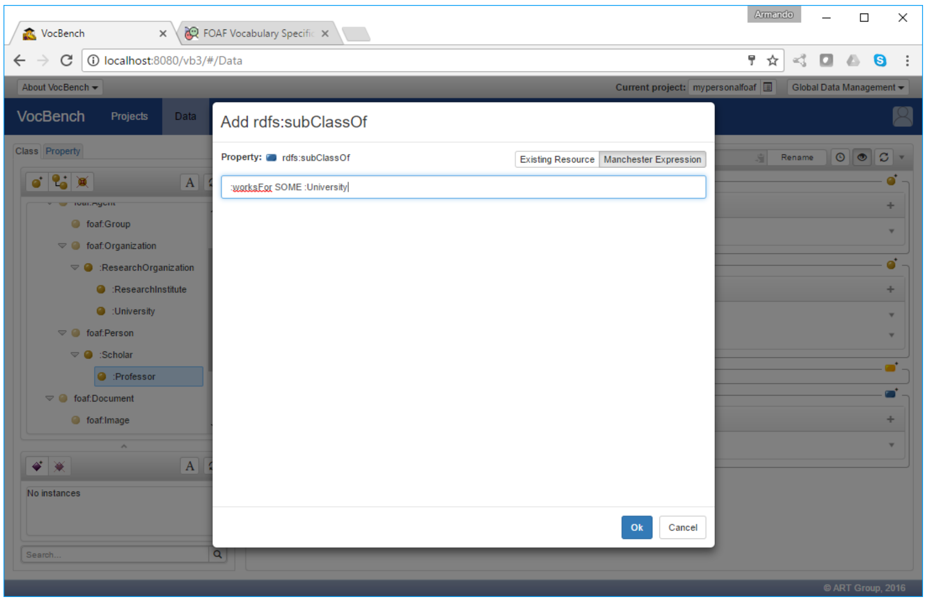
- *Project Managers*: project-local administrators. Inside a project, they can do everything: from data and configuration management to assigning users to the project and granting roles to them. Their boundaries are: other projects and system-level settings and configuration.
- Specific project-local roles: *Ontology editors* (authorized to perform changes at the axiomatic level), *Thesauri Editors* (authorized to work on thesauri without performing OWL editing actions), *Terminologists*/*Lexicographers* (authorized to edit lexicalizations, can be limited to edit only certain languages according to their proficiencies), *Validators* (can perform validation action, see "Formal Workflow Management" section)

Advanced History and Change Tracking mechanism. Both a strength and a weakness in VB2, the Change Tracking mechanism that powered History & Validation was appreciated by most users. However, being based on a pre-defined set of recognized operations, it severely limited system maintainability (req. R9) and the possibility to perform (req. R6) under-the-hood changes (e.g. through changes brought directly through SPARQL) *while* keeping a history which is consistent with the status of the dataset. VB3 abandoned the separated relational DB that held user and history data and implemented, completely in RDF (req. R15), a track-change mechanism working at triple-level and complementing this fine-grained representation with rich metadata (R11) about the invoked action and the context of the invocation. Triples removed/added by each action are reified, grouped around a common resource representing the action that produced the change and stored in a separated (but connected to the project) RDF repository (the *support repository*) together with the actions' metadata. The change-tracking mechanism has been implemented as a new *sail* for the RDF4J framework (`http://rdf4j.org/`). The sail is embedded with the system, but can also be deployed as a pluggable component inside other sail-compliant triple stores (req.R4).

The design of the history and change tracking mechanism in VB3 was guided by a landscape analysis [11], in which we discussed the nature and the representation of change, reviewed some version control systems for RDF, and delved into the challenges posed by validation.

More Powerful yet Streamlined Workflow Management. VB2 had a 5-steps publication workflow, clocked by the property "status" (with values: proposed, validated, published, deprecated and proposed_deprecated) and, redundantly, with information stored in the DB about the status of operations to be validated. Also in VB2, the concepts of *resource* and *action* were mixed up in the validation procedure, with the status of a resource being affected by the validation (e.g. moving from "proposed" to "validated"), while single affected triples had no trace of their validation status if not in the DB tables. This follows from the fact that it is not possible to attach a status to a triple in RDF, if not by reifying the triple. Finally, there is no standard W3C equivalent for the custom "status" property in VocBench, thus reducing this status information about the workflow to something to be removed from the dataset when it gets published.

Benefiting from the new Change Tracking system, we have made things clearer, and easier: there is no "status" property anymore, as the workflow is implicitly expressed



Add rdfs:subClassOf

Property: rdfs:subClassOf

Existing Resource | Manchester Expression

:worksFor SOME :University

Figure 4. Editing axioms for a class in OWL, extending the FOAF ontology

by the validation mechanism coded into graphs. The added/removed triples are stored in the support repository as described in the previous section, while non-reified "pre-views" of them are available in separate graphs (*staging-add-graph* and *staging-delete-graph*) in the main repository, and the system, being aware of this graph/repository organization, presents them appropriately to the user. In this way, the main graph being written implicitly represents stable information, which does not need to be tagged as "validated". The distinction between "validated" and "published" has been removed as most users considered this sort of dual validation as useless. Probably, the original intention, dating back to the first VocBench, was to distinguish which resources had been published in an open version of the dataset from those that – though being validated – had never seen the light out of VocBench. However, this distinction has never been put in place in any known user workflow. Finally, the status of deprecation has been represented through the official owl:deprecated property. The status of "proposed deprecated" is also intuitively represented by the need to validate the action for setting the owl:deprecated property to "true".

Improved and More Complete Support for SKOS. VB2 had already an advanced support for multiple SKOS schemes. We have improved the management by allowing users to select more schemes for browsing the concept tree and by adopting a combination of conventions and editing capabilities for quickly associating the proper schemes to newly created concepts and collections. Support for SKOS collections and ordered collections has been introduced in the system with dedicated UI views and editing facilities.

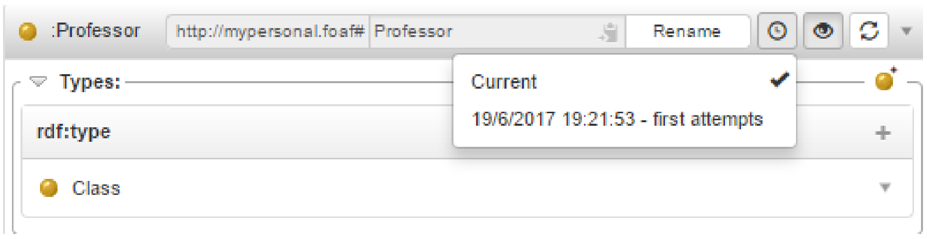

:Professor
http://mypersonal.foaf# Professor
Rename
Types:
rdf:type
Class
Current
19/6/2017 19:21:53 - first attempts

Figure 5. time-traveling across different versions of a resource

OWL Support. VocBench already allowed for importing ontology vocabularies for modeling thesauri. Now VocBench also supports ontology development (requirement R8), with editing of OWL axioms (using the Manchester syntax for both editing and visualization, see Figure 4) and an almost full coverage of OWL2 expressions.

SPARQL Querying and Update. A new SPARQL UI, based on YASGUI [12] has been included in VB3, featuring the same feeding-from-live data mechanism present in VB2. An important improvement over VB2: now changes performed through SPARQL updates can be tracked and, consequently, put under *validation* if the project enables it.

Alignment. VB3 provides the same inter-project alignment support of VB2, allowing users to browse other projects and supporting semi-automatic label-based searches over them in order to provide candidate resources for alignments. In addition to this on-the-fly generation of mappings, VB3 introduces an alignment-validation tool: it is possible to load alignments following the model of the INRIA Alignment API [13], inspect the aligned resources and validate the alignment. Validated alignments can then be projected over standard RDFS/OWL or SKOS properties, depending on the validated relation and the involved entities. E.g. two classes mapped through an inria:EquivRelation will be mapped through the property owl:equivalentClass while two SKOS concepts will be aligned with a skos:exactMatch.

Declarative Service Implementation. An innovation that can be immediately appreciated by developers is the declarative way in which services are implemented: the business logic of the services is represented through a dedicated set of java-annotations that have been specifically developed for the VB framework. Declarations specifying if a service requires read/write access to the data, the capabilities the user must have in order to use them, the pre-requisites on the input parameters, etc. can all be represented in terms of this annotation vocabulary. Therefore, service development becomes an easier task, less prone to errors and the produced code is more readable, as the developer needs only to focus on what the service does.

Versioned datasets and metadata. In VB3, users can create snapshots of a repository and tag them with a version identifier (and other metadata, such as the time of creation of the snapshot). Users can travel across the different points in time identified by these versions, and thus analyze the evolution of browsed resources. The time-travel can be performed both globally, by switching version so that everything in the UI refers to the selected version, and locally, by inspecting different versions of a resource in the resource-view (Figure 5) or different versions of a tree (of classes, concepts, etc...)

Metadata Export. The "metrics" section of VB2 has been replaced with a page for editing and exporting metadata modeled after several existing metadata vocabularies: the Data Catalog Vocabulary (DCAT) [14], the Asset Description Metadata Schema (ADMS) [15], The Vocabulary of Interlinked Datasets (VoID) [16] and the Linguistic Metadata vocabulary (LIME) [17] (a lexical extension to VoID). While DCAT and ADMS mostly deal with static metadata, VoID and LIME offer statistical information about the dataset and its lexical information. The information of VoID and LIME is being computed through a profiler bundled with the LIME API [18]. This metadata build&export functionality is implemented as an extension point of the platform, so that new vocabularies can be dynamically added to the platform. For instance, an application profile for DCAT thought for European public sector data portals (DCAT-AP: `https://joinup.ec.europa.eu/node/145996`) has later been added to the list of exporters, as of the ISA^2 context specifically supporting public administration. Conceptual and lexical metadata will become important in a planned future version of the system, as this information will be exploited to support the setup of automatic alignment processes, in the spirit of [19,20,21].

ICV. A section dedicated to Integrated Constraint Validation (ICV) allows the user to inspect possible anomalies. These include violations of formal constraints (e.g. thesauri constraints on existence and uniqueness of preferred labels, disjointness between taxonomy and relatedness etc…) or problematic (though not necessarily illegal) patterns (e.g. a skos:Concept having a broader concept and being the top concept of a same scheme). Interactive fixes are provided (req. R3) for each discovered integrity break.

Desktop Tool and Collaborative Web Platform. As of requirement R7, the system offers a very lightweight installation (i.e. unzip and click-to-run) which, followed by default configuration options for both system and project creation, make VB3 a good choice for users looking for a simple and easy-to-use desktop tool. Other more complex settings are still possible, satisfying different needs for distributed installation (separation of data servers, UI servers), better performance, etc...

4 Conclusion And Future Work

In the last 3 years, VocBench has addressed the needs of large organizations, companies and independent users needing an open source collaborative environment for editing thesauri, supporting a formalized editorial workflow. A vibrant user community (`http://vocbench.uniroma2.it/support/community.jsf`) has grown in these years around VocBench initially inside various departments of FAO, and later spread across other organizations with analogous needs. Continuous user feedback allowed us to spot bugs and to improve the usability of VocBench.

It is thanks to this community feedback, to the support of the ISA^2 program and to our desire to reach new quality levels that we started this endeavor, by rethinking most of VocBench from scratch, still benefiting from the experiences we had with VB2.

The most important achievement of the new platform lies at its core: a fully-fledged RDF core framework, developed by extending the Semantic Turkey [22] framework (which already powered the original VocBench 2) with functionalities for user management, role-based access control, change tracking and collaboration and by providing it with a new user interface. We hope that this evolution of the system will lay a solid foundation for the realization of a new range of services spacing from knowledge acquisition, evolution and management in the European and worldwide scenario.

Acknowledgments. This work has been funded by the European Commission ISA² programme; the development of VocBench 3 (VB3) is managed by the Publications Office of the EU under the contract 10632 (Infeurope S.A.).

References

1. Caracciolo, C., Stellato, A., Morshed, A., Johannsen, G., Rajbhandari, S., Jaques, Y., Keizer, J.: The AGROVOC Linked Dataset. Semantic Web Journal 4(3), 341–348 (2013)
2. Stellato, A., Rajbhandari, S., Turbati, A., Fiorelli, M., Caracciolo, C., Lorenzetti, T., Keizer, J., Pazienza, M.T.: VocBench: a Web Application for Collaborative Development of Multilingual Thesauri. In : The Semantic Web. Latest Advances and New Domains (Lecture Notes in Computer Science) 9088. Springer International Publishing (2015), pp.38-53
3. World Wide Web Consortium (W3C): SKOS Simple Knowledge Organization System Reference. In: World Wide Web Consortium (W3C). (Accessed August 18, 2009) Available at: http://www.w3.org/TR/skos-reference/
4. World Wide Web Consortium (W3C): SKOS Simple Knowledge Organization System eXtension for Labels (SKOS-XL). In: World Wide Web Consortium (W3C). (Accessed August 18, 2009) Available at: http://www.w3.org/TR/skos-reference/skos-xl.html
5. Hodge, G.: Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files. Council on Library and Information Resources, Washington, DC (April 2000)
6. McCrae, J., Spohr, D., Cimiano, P.: Linking Lexical Resources and Ontologies on the Semantic Web with Lemon. In : The Semantic Web: Research and Applications (Lecture Notes in Computer Science) 6643. Springer Berlin Heidelberg (2011), pp.245-259
7. Jain, P., Hitzler, P., Yeh, P.Z., Verma, K., Sheth, A.P.: Linked Data Is Merely More Data. In : Linked Data Meets Artificial. AAAI Press, Menlo Park (2010), pp.82–86
8. Fiorelli, M., Lorenzetti, T., Pazienza, M.T., Stellato, A.: Assessing VocBench Custom Forms in Supporting Editing of Lemon Datasets. In : Language, Data, and Knowledge (Lecture Notes in Artificial Intelligence) 10318. Springer, Cham (2017), pp.237-252
9. Bratko, I.: Prolog Programming for Artificial Intelligence. Addison Wesley (2001)
10. Denti, E., Omicini, A., Ricci, A.: tuProlog: A Light-Weight Prolog for Internet Applications and Infrastructures. In : Practical Aspects of Declarative Languages (Lecture Notes in Computer Science) 1990. Springer, Berlin, Heidelberg (2001), pp.184-198
11. Fiorelli, M., Pazienza, M.T., Stellato, A., Turbati, A.: Version Control and Change Validation for RDF Datasets. In : Metadata and Semantics Research. 11th Research Conference, MTSR 2017, Tallinn, Estonia, November 28 - December 1, 2017, Proceedings. Springer (2017) (in press).

12. Laurens, R., Rinkea, H.: The YASGUI family of SPARQL clients. Semantic Web 8(3), 373-383 (2017)
13. David, J., Euzenat, J., Scharffe, F., Trojahn dos Santos, C.: The Alignment API 4.0. Semantic Web Journal 2(1), 3-10 (2011)
14. World Wide Web Consortium (W3C): Data Catalog Vocabulary (DCAT). In: World Wide Web Consortium (W3C). (Accessed January 16, 2014) Available at: http://www.w3.org/TR/vocab-dcat/
15. Dekkers, M.: Asset Description Metadata Schema (ADMS). In: World Wide Web Consortium (W3C). (Accessed August 1, 2013) Available at: http://www.w3.org/TR/vocab-adms/
16. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing Linked Datasets with the VoID Vocabulary (W3C Interest Group Note). In: World Wide Web Consortium (W3C). (Accessed March 3, 2011) Available at: http://www.w3.org/TR/void/
17. Fiorelli, M., Stellato, A., Mccrae, J.P., Cimiano, P., Pazienza, M.T.: LIME: the Metadata Module for OntoLex. In : The Semantic Web. Latest Advances and New Domains (Lecture Notes in Computer Science) 9088. Springer International Publishing (2015), pp.321-336
18. Fiorelli, M., Pazienza, M.T., Stellato, A.: An API for OntoLex LIME datasets. In : OntoLex-2017 1st Workshop on the OntoLex Model (co-located with LDK-2017), Galway (2017)
19. Pazienza, M.T., Sguera, S., Stellato, A.: Let's talk about our "being": A linguistic-based ontology framework for coordinating agents. Applied Ontology, special issue on Formal Ontologies for Communicating Agents 2(3-4), 305-332 (December 2007)
20. Pazienza, M.T., Stellato, A.: An Environment for Semi-automatic Annotation of Ontological Knowledge with Linguistic Content. In : The Semantic Web: Research and Applications (Lecture Notes in Computer Science) 4011. Springer (2006), pp.442-456
21. Fiorelli, M., Pazienza, M.T., Stellato, A.: A Meta-data Driven Platform for Semi-automatic Configuration of Ontology Mediators. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S., eds. : Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland (May 2014)
22. Pazienza, M.T., Scarpato, N., Stellato, A., Turbati, A.: Semantic Turkey: A Browser-Integrated Environment for Knowledge Acquisition and Management. Semantic Web Journal 3(3), 279-292 (2012)