

Modeling, Managing, Exposing, and Linking Ontologies with a Wiki-based Tool

Alessio Bosca¹, Matteo Casu¹, Mauro Dragoni², Andi Rexha³

¹ Celi s.r.l., Via S. Quintino 31, I-10131, Torino, Italy

² FBK-IRST, Trento, Italy

³ Know-Center Graz, Graz, Austria

alessio.bosca—casu@celi.it dragoni@fbk.eu arexha@know-center.at

Abstract

In the last decade, the need of having effective and useful tools for the creation and the management of linguistic resources significantly increased. One of the main reasons is the necessity of building linguistic resources (LRs) that, besides the goal of expressing effectively the domain that users want to model, may be exploited in several ways. In this paper we present a wiki-based collaborative tool for modeling ontologies, and more in general any kind of linguistic resources, called MoKi. This tool has been customized in the context of an EU-funded project for addressing three important aspects of LRs modeling: (i) the exposure of the created LRs, (ii) for providing features for linking the created resources to external ones, and (iii) for producing multilingual LRs in a safe manner.

Keywords: Ontology Authoring, Collaborative Modeling, Multilinguality

1. Introduction

The construction of effective LRs is one of the most important aspects for improving the quality of information systems like ones that widely perform natural language processing and information retrieval activities. Recently, the wide usage of LRs raised the necessity of building such artifacts in a way that grants their quality and the possibility of easily exploiting them. Here, we want to focus on four important challenges concerning the creation and maintenance of LRs.

Collaboration It is nowadays well-established that crafting LRs has become a teamwork activity, as it requires a range of knowledge and skills hardly findable all together in a single person. For this reason, collaborative aspects in LRs modeling, in particular concerning ontologies, have been investigated, and several works to support and enhance collaboration in this context have been presented (see e.g. (Palma et al., 2011; Dimitrova et al., 2008)). The requirements and features that have emerged from these studies highlighted the need to support collaboration in an articulated way: from supporting the collaboration between who understands the domain to be represented (the domain expert) and who has proper expertise in LRs modeling (the knowledge engineer), in order to support communication, discussion, and decision making between (geographically) distributed teams of contributors. Supporting collaboration requires enabling the awareness of the user on the evolution of the modeling artifacts, favoring the coordination of the modeling effort within the team, as well as fostering the communication of the modeling choices and decisions made among the modeling actors.

Multilinguality With the recent rapid diffusion over the international computer networks of world wide distributed document bases, the issue of multilinguality is becoming increasingly relevant. So far, research and development activities have been concentrated on monolingual environments and, in the large majority of cases, the default language has been English. Although English admittedly tends to play a predominant role in international communica-

tions, the diversity of the world's languages and cultures gives rise to an enormous wealth of knowledge and ideas. A clear example of this scenario is represented by users throughout the world that, independently of their native tongue, want to access to the massive volumes of information available over the networks and, in particular, over the World Wide Web. Together with the growth of these requests, the LRs community has started exploring the possibility to use multilinguality for increasing the expressiveness of the artifacts, which are more and more used for enhancing the information associated to the available resources. A recent example, which witnesses the importance of multilinguality in the field of ontology engineering, is provided by the Monnet Project ¹, which targets the problem of multilingual information access at the semantic level (McCrae et al., 2011).

Exposure The growth of complex and distributed platforms, especially when Web Services are widely used, required the necessity of making the built LRs available through the network in order to permit their usage. Such a feature includes the necessity of exposing all information in several standard formats that may also evolve through the time; therefore, an effective exposure service is necessary when there is an intensive usage of LRs in very dynamic contexts.

Linking The last aspect concerns the necessity of linking the modeled LRs with the existing ones. With the grown of interest in creating semantic artifacts for representing knowledge, the possibility of defining effective links between different artifacts permits to spread knowledge not only for enriching the representation of the knowledge itself, but also for breaking language barriers in accessing contents. One of the most general benefit of having mapping facilities is the possibility of quickly enrich original ("local") artifacts with information coming from external artifacts, without the need of redefining knowledge in the "local" one.

¹<http://www.monnet-project.eu>

This way, updates on the linked resources are instantly available to the “local” ones by avoiding the publish of deprecated information. A survey offering different perspectives concerning the linking problem may be found in (Euzenat and Shvaiko, 2007)(Bellahsene et al., 2011).

In the lights of these aspects, we may state that the building of multilingual LRs is a complex activity that requires to tackle a significant number of problems.

In this paper, we present an existing wiki-based tool, called MoKi, that have been customized in order to address the challenges described above. The tool has been originally built by targeting the modeling of ontologies, as we will describe later in this paper; however, all the principles and the described features may be generalized in the context of building any kind of LRs.

This paper is structured as follows: Section 2. presents an overview of the most recent active projects concerning the modeling and editing of LRs. Section 3. presents the collaborative tool for creating and maintaining ontologies and thesauri, and the customizations that have been implemented for addressing the specific challenges described above. Section 4. discuss how the produced tool advances the state of the art; while, Section 5. concludes the paper.

2. Ontology Editors Tools

The set of tools available today for modeling LRs is very wide and they provide different sets of functionalities based on their purposes. In this Section, we will provide a survey of the most important tools available for knowledge modeling.

*Amine*² is a rather comprehensive, open source platform for the development of intelligent and multi-agent systems written in Java. It is composed by 4 different modules which one of them is related to knowledge modeling. Such “Ontology Layer” offers the possibility to create, edit, and query an ontology. The tool itself is a standalone platform; therefore, no connection with and exposure to external services as well as collaborative environment are provided.

The Apelon DTS (Distributed Terminology System)³ is an integrated set of open source components that provides comprehensive terminology services in distributed application environments. DTS supports national and international data standards, which are a necessary foundation for comparable and interoperable health information, as well as local vocabularies. Typical applications for DTS include clinical data entry, administrative review, problem-list and code-set management, guideline creation, decision support and information retrieval. Though not strictly an ontology management system, Apelon DTS has plug-ins that provide visualization of concept graphs and related functionality that make it close to a complete solution.

DOME (DERI Ontology Management Environment (DOME))⁴ is developed by the Ontology Management Working Group (OMWG). Their goal is to create a suite

for the efficient and effective management of ontologies that provides an integral solution for the overall problem. The main inspiring principles are simplicity, completeness, and reuse. Such a suite will comprise tools support for editing, browsing, versioning, evolution, mapping, and merging of ontologies. The tool is available in the form of freely combinable Eclipse Plug-ins and it makes use of an ontology-neutral API that will provide access to different ontology repositories and reasoning engines. Actually, such suite supports the development of code for managing ontologies, but it provides only a very light support for non-expert people.

*FlexViz*⁵ is a Flex-based, Protg-like client-side ontology creation, management and viewing tool. It allows users to create and to browse a single ontology where the concepts are represented by nodes and the relationships between concepts (e.g. “Is_A” and “Part_Of”) are represented as arcs. The main features of the tool include node and arc type filtering, searching, many different graph layouts, customizable node and arc labels, customizable node and arc tooltips, and the possibility of navigating the history of each concept.

*Knoodl*⁶ facilitates community-oriented development of OWL based ontologies and RDF knowledge bases. It also serves as a semantic technology platform, offering a Java service-based interface or a SPARQL-based interface so that communities can build their own semantic applications using their ontologies and knowledge bases. With respect to almost all the other tools, Knoodl supports the editing ontologies remotely through an API interface in order to foster the collaboration between people. Moreover, a template mechanism is provided in order to support the knowledge modeling for people with non-engineering expertise.

*The NeOn toolkit*⁷ is investigating the entire development and evolution lifecycle of networked ontologies that enable complex, semantic applications. They pay special attention to integrating NeOn research into work practices. Therefore, the methodology, toolkit and infrastructure have been intertwined with their deployment and testing from the early phases of the project. NeOn uses a case-centered methodology, which means that the research results are applied to the real-world cases involving partners from industry and public bodies. The main features of the tool include the management of the dynamics and the evolution of ontologies in an open, networked environment; the support for collaborative development of networked ontologies; the possibility of using contexts for developing, sharing, adapting and maintaining networked ontologies and an improved human-ontology interaction (i.e. making it easier for users with different levels of expertise and experience to browse and make sense of ontologies). However, it is important to highlight that the idea of collaboration is not intended as many people that work at the same time on the same ontologies, but as the possibility of connecting different on-

²<http://amine-platform.sourceforge.net/>

³<http://apelon-dts.sourceforge.net/index.html>

⁴<http://dome.sourceforge.net/>

⁵<http://www.thechiselgroup.org/flexviz>

⁶<http://knoodl.com/ui/home.html>

⁷http://neon-toolkit.org/wiki/Main_Page

tologies developed by different people through an internal linking mechanism.

Protégé⁸ is a free, open source visual ontology editor and knowledge-base framework. The Protégé platform supports two main ways of modeling ontologies via the Protégé-Frames and Protégé-OWL editors. Protégé ontologies can be exported into a variety of formats including RDF(S), OWL, and XML Schema. There are a large number of third-party plugins that extends the platform's functionality that may be found in the Protégé Plugin Library. The support for the collaboration is provided through the Collaborative Protégé plug-in extension. Such a plug-in, supports the collaborative ontology editing as well as annotation of both ontology components and ontology changes. In addition to the common ontology editing operations, it enables annotation of both ontology components and ontology changes. It supports the searching and filtering of user annotations, also known as notes, based on different criteria. Two different collaborative modes are supported:

- the multi-user mode: allows multiple clients to edit simultaneously the same ontology hosted on a Protégé server. All changes made by one client are immediately visible by other clients. This mode is also referred to as client-server mode, or concurrent mode and requires a client-server setup. This mode is based on the implementation of the multi-user Protégé and is the preferred mode in which Collaborative Protégé should be run;
- the standalone mode: allows multiple users to access the same ontology in succession. The ontology can be stored on a shared network drive and all clients will access the same project files. However, simultaneous access is not possible. This mode is also referred to as the consecutive mode.

TopBraid Composer⁹ is an enterprise-class modeling environment for developing Semantic Web ontologies and building semantic applications. Fully compliant with W3C standards, Composer offers comprehensive support for developing, managing and testing configurations of knowledge models and their instance knowledge bases. It is based on the Eclipse IDE and it supports only a standalone usage of the tool. Basically, it supports only the creation of ontologies in local and it provides an editor for defining SPARQL queries.

Anzo¹⁰ includes an (RDFS and OWL-based) ontology editor that can be used directly within Excel. In addition to that, Anzo includes the capability to automatically generate an ontology from existing spreadsheet data, which is very useful for quick bootstrapping of an ontology. Moreover, it is possible to publish ontologies in Linked Open Data format.

Hozo¹¹ is an ontology visualization and development tool

⁸<http://protege.stanford.edu/>

⁹http://www.topquadrant.com/products/TB_Composer.html

¹⁰http://www.cambridgesemantics.com/products/anzo_for_excel

¹¹<http://www.hozo.jp/ckc07demo/>

that brings version control constructs to group ontology development. Modeled ontologies are managed on the server and shared by all users, therefore, collaboration between users is supported.

Lexaurus Editor¹² is for off-line creation and editing of vocabularies, taxonomies and thesauri. It supports import and export in Zthes and SKOS XML formats, and allows hierarchical/poly-hierarchical structures to be loaded for editing, or even multiple vocabularies to be loaded simultaneously, so that terms from one taxonomy can be reused in another. The tool provide a graphical support for managing multilingual ontologies and tool interface.

Model Futures OWL Editor¹³ combines simple OWL tools, featuring UML (XMI), ErWin, thesaurus and imports. The editor is tree-based and has a "navigator" tool for traversing property and class-instance relationships. It can import XMI (the interchange format for UML) and Thesaurus Descriptor (BT-NT XML), and EXPRESS XML files.

OntoTrack¹⁴ is a browsing and editing ontology authoring tool for OWL Lite. It combines a sophisticated graphical layout with mouse enabled editing features optimized for efficient navigation and manipulation of large ontologies.

PoolParty¹⁵ is a triple store-based thesaurus management environment which uses SKOS and it provides an interesting text analysis and extraction feature for tag recommendations. Moreover, there is a PoolParty Web service that enables a Zthes thesaurus in XML format to be uploaded and converted to SKOS (via skos:Concepts). Finally, it provides a Linked Data service for exposing the modeled artifacts.

TemaTres¹⁶ is a Web application to manage controlled vocabularies, taxonomies and thesaurus with a simple collaborative environment. It is easy to use, thanks to the implementation of only basic functionalities, and it permits to expose data in various formats and through a SPARQL endpoint.

ThManager¹⁷ is a tool for creating and visualizing SKOS RDF vocabularies. ThManager facilitates the management of thesauri and other types of controlled vocabularies, such as taxonomies or classification schemes by providing the basic facilities for modeling them.

Vitro¹⁸ is a general-purpose web-based ontology and instance editor with customizable public browsing. Vitro is a Java web application can be executed on a public collaborative environment in order to allow (i) the creation or load of ontologies in OWL format; (ii) the editing of instances and relationships; (iii) the building of a public web site for publishing data; and (iv) the possibility of searching data

¹²<http://www.k-int.com/products/lexauruseditor>

¹³<http://www.modelfutures.com/owl>

¹⁴<http://www.informatik.uni-ulm.de/ki/ontotrack/>

¹⁵<http://poolparty.punkt.at/>

¹⁶<http://sourceforge.net/projects/tematres/>

¹⁷<http://thmanager.sourceforge.net/>

¹⁸<http://vitro.mannlib.cornell.edu/>

through the integration of the Solr search engine.

VocBench¹⁹ is a web-based, multilingual, editing and workflow tool that manages thesauri, authority lists and glossaries using SKOS-XL. Designed to meet the needs of semantic web and linked data environments, VocBench provides tools and functionalities that facilitate both collaborative editing and multilingual terminology. It also includes administration and group management features that permit flexible roles for maintenance, validation and publication. The tool is designed to allow external developers to add functionality via plug-ins without disturbing the core; with Protégé, these are the only tools that foresee an expanding mechanism of their functionalities through plug-ins, or others. Finally, it provides a native for SKOS for the management of multiple triple-stores.

In the next Section, we will present the collaborative tool presented in this paper, while the discussion and its comparison with respect to the ones described above will be provided in Section 4..

3. The MoKi Tool

MoKi²⁰ is a collaborative MediaWiki-based (Wikimedia Foundation, 2012) tool for modeling ontological and procedural knowledge in an integrated manner²¹. MoKi is grounded on three main pillars, which we briefly illustrate with the help of Figure 1:

- each basic entity of the ontology (i.e., concepts, object and datatype properties, and individuals) is associated to a wiki page. For instance, the concept Mountain in Figure 1 is associated to a wiki page which contains its description;
- each wiki page describes an entity by means of both unstructured (e.g., free text, images) and structured (e.g. OWL axioms) content;
- a multi-mode access to the page content is provided to support easy usage by users with different skills and competencies.

The multi-mode access is the key feature that permits the support of collaboration between different types of users. Figure 1 shows the three access modes, implemented in MoKi, for accessing the unstructured and structured content of the wiki page:

- the unstructured access mode allows the user to edit/view the unstructured part of a MoKi page. Editing/viewing occurs in the standard MediaWiki way;
- the fully-structured access mode allows the user to edit/view the structured part of a MoKi page using the full OWL 2 expressiveness²² and is meant to be

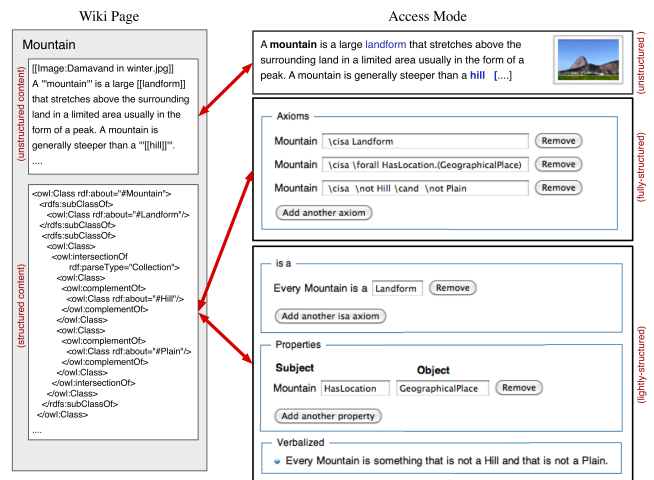


Figure 1: A page and the access modes in MoKi

used by knowledge engineers to author the formal statements describing the entity associated to the wiki page;

- the lightly-structured access mode enables users to edit/view the content of the structured part of the MoKi page in a simplified way. This access mode, consists of a form meant to be used by domain experts, and contains statements that correspond to all the axioms in the fully-structured access mode. In the upper part the user can view and edit simple statements which can be easily converted to/from OWL statements. An example is the uppermost statement “Every Mountain is a Landform” in the lightly-structured access mode of Figure 1. The bottom part of the form provides a verbal description (automatically obtained via the OWL 2 Verbalizer (Kaljurand and Fuchs, 2007)) of those OWL statements which cannot be intuitively translated/edited as simple statements in the upper part of the page. The purpose of this verbal description is to give the domain expert a flavor of the complex statements that the knowledge engineer has formalized. If doubtful about some of the statements, the domain expert can mark them and ask for a clarification using e.g., the Discussion mechanism.

A comprehensive description of core version of MoKi is presented in (Ghidini et al., 2012).

3.1. Supporting the Collaborative Modeling of Multilingual Ontologies with MoKi

In this subsection, we describe the main customizations implemented in the MoKi tool for providing support to the collaborative multilingual management and exposure of ontologies and, more in general, LRs.

Collaborative Editing Facilities. Given the complexity of managing linguistic resources, each contribution often needs to be checked and agreed upon by a community of experts. This is especially true when linguistic resources are used to represent terminological standards which need to be carefully discussed and evaluated. To support this collaborative activity, we foreseen the usage of the wiki-style

¹⁹<http://vocbench.uniroma2.it/>

²⁰<http://moki.fbk.eu>

²¹Though MoKi allows to model both ontological and procedural knowledge, here we will limit our description only to the features for building ontologies.

²²We adopt the syntax of *latex2owl*: <https://dkm.fbk.eu/index.php/Latex2owl>

[new][edit][reply] Translations -- Admin 13:18, 8 February 2012 (UTC)

What do you think about my Spanish translation?

Figure 2: Example of started discussion.

features of MoKi, expanded with the possibility of assigning specific tasks of ontology entity evolution to specific experts who need to monitor, check, and approve the suggested change. This customization promotes the management of the changes carried out on the ontology (both at domain and linguistic layer) by providing the facilities necessary to manage the ontology entity life-cycle.

These facilities may be split in two different sets of features. The first group may be considered as a monitor of the activities performed on each entity page. When changes are committed, approval requests are created (Figure 3). They contain the identification of the expert in charge of approving the change, the date in which the change has been performed, and a natural language description of the change. Moreover, a mechanism for managing the approvals and for maintaining the history of all approval requests for each entity is provided. The second set of features contains the facilities for managing the discussions associated with each entity page. A user interface for creating the discussions has been implemented together with a notification procedure that alerts users when new topics/replies, related to the discussions they are following, are posted (Figure 2). Therefore, MoKi provides a set of MediaWiki-based collaborative editing functionalities, such as:

- *Discussions*: to discuss about challenging issues related to the ontology modeling. It is possible to discuss on single ontology entities or on (a part of) the whole model. Comments in the discussion pages are organized in threads, with details on the user and date/time associated to each comment;
- *Watchlist*: to monitor interesting ontology entities. Any change performed on monitored ontology entities is notified (with messages and email alerts) to the user;
- *Notifications*: to inform users about ontology changes that are relevant for them. E-mail or message notifications are automatically sent in case changes to pages in the users' watchlist occur. Users can also send specific notifications, soliciting a confirmation or revision on some aspects of the ontology from particular users;
- *History and Revision*: to track changes and comments added on a specific ontology entity.

Multilinguality Management Features. MoKi is equipped with a set of features for enabling both manual and automatic translation of labels and descriptions associated to each entity defined by users into ontologies and vocabularies that they are modeling. Manual translations may be provided with a custom interface, while automatic suggestions are available through a connector that invokes

Pending approvals:

#	Request date	User	Group Manager	Description	Sign
1	01/03/2013	Admin	k_engineer	The following parent have been removed: Method.	Sign
2	01/03/2013	Admin	k_engineer	The following parent have been added: FarmingMethod.	Sign
3	01/03/2013	Admin	k_engineer	The following parent have been added: Method.	Sign
4	01/03/2013	Admin	t_admin	The Spanish translation of the concept description has been changed.	Sign
5	01/03/2013	Admin	k_engineer	The following properties have been added: ISAffectedBy (range: PestControlTechnique).	Sign

Figure 3: Approval box.

Figure 4: Quick translation box for editing entities translations

external translation services. Concerning automatic translation, the component sends the request to APIs exposed by third-party translation services (for instance, Microsoft Bing²³) and, after the retrieval of the result, the representation of the entity is updated with the returned information. Further details, about the translation services currently supported by MoKi can be found in (Dragoni et al., 2011).

Also the export functionality, which enables the dump of the linguistic artifact in several semantic web formats (OWL, SKOS, etc.), has been revisited in the light of multilinguality, by adding the possibility to choose the export languages, among the available ones.

For facilitating the work of language experts, we have implemented the possibility of comparing side-by-side two lists of translations. This way, the language expert in charge of revising the translations, avoiding to navigate among the entity pages, is able to speed-up the revision process. Figure 5 shows such a view, by presenting the list of concepts in the English and Italian translations. At the right of each element of the table, it is placed a link allowing to invoke a quick translation box (as shown in Figure 4) that gives the opportunity to quickly modify information without opening the entity page. Finally, in the last column, it is placed a flag indicating that changes have been performed on that concept, and a revision/approval is requested.

Linked Open Data Service. In order to permit the exposure of the ontology artifact to third-party services, MoKi has been equipped with a service that exposes entity information by using Linked Open Data formats. Such a service offers the possibility to perform operations on the ontology remotely; examples of available remote operations are the

²³<http://www.bing.com/translator>

List all Concepts

Number of concepts in the Domain Model: 62

Concept	Description	Concept translation	Description translations
Activity	A type of action performed by an agent in general sense	attività	
agricultural method	Practices used to enhance crop and livestock health and prevent weed, pest or disease problems without the use of chemical substances	agrario metodo	le pratiche vegetali e animali usati per promuovere la salute e la prevenzione delle malattie, parassiti e infestanti problemi senza l'uso di sostanze chimiche
European agricultural method	Agricultural techniques used in Europe.	metodo agricolo europeo	le tecniche agricole utilizzate in Europa.
animal origin processed product	Any product of animal origin canned, cooked, frozen, concentrated, pickled or otherwise prepared to assure its preservation in transport, distribution and storage, but does not include the final cooking or preparation of a food product for use as a meal or part of a meal such as may be done by restaurants, catering companies or similar establishments where	animale sorgente processad prodotto	

Figure 5: View for comparing entities translations

retrieval of the entire ontology, or part of it, or the possibility to edit the ontology e.g., by adding a new translated label. The service provides a RESTful interface for receiving the requests, while the results are exposed by using the SKOS language ²⁴.

This customization has been implemented for providing an exposure feature that permits the linking between MoKi and external third-party tools that want to exploit the modeled artifacts. The interface of the service allows to retrieve ontology data by sending POST or GET requests to the service. The current version of the service is able to understand four different parameters:

- “*method*”: the name of the method that the client wants to invoke (see (Dragoni, 2014) for the list of the available methods);
- “*concept*”: the name of the concept for which the client wants to retrieve information;
- “*langid*”: the identification code of the language used for filtering the results;
- “*updatetimestamp*”: used when the client wants to retrieve only the list of the concepts updated after a certain timestamp.

Entity Linking Component. Linking the local ontology to external ones provides a big benefit for extending the local knowledge and also for navigating through external resources. In order to supply these benefits, MoKi has been extended with an Ontology Linking component, which offers to experts a set of suggestions for candidate matching concepts, ranked by their confidence. This component has been designed as an iterative process implemented in two steps:

- *Syntactic search*: similar concepts, based on their label content and description, are searched. This phase uses a syntactic match, based on label similarity algorithms, enriched with semantic information (like synonyms). The results are computed starting from a set of pre-indexed resources generally expressed in linked open data formats.

²⁴<http://www.w3.org/2004/02/skos/>

- *Feedback re-rank*: suggestions are re-ranked according to feedbacks provided by the users. This phase uses previous accepted mappings to boost and improve the results of the *Syntactic search*

The first step is used in some of the state of the art approaches like in (William W Cohen, 2003). Semantic relationships may not be exploited from the word similarity, but multilinguality in this case, raises chances of syntactic similarity.

The other advantage of this component is the feedback from the user. Such feedbacks allow to have a partial alignment of the ontology and orient the search for similar concepts to a certain subset of concepts. This is very important to avoid false positive results of the *Syntactic search* step. It also gives room for the implementation of further machine learning techniques allowing the improvement of the *Syntactic search* by learning weights of the terms occurring in the labels and descriptions.

Practically, from the concept page it is possible to invoke an external service that receives as input a bean containing all information related to the concept that has to be mapped (in case of multilingual representation of the concept all labels and all descriptions are used), and that returns a list of candidate concepts that can be mapped. The user has only to select which concept, if any, is a good mapping to the local one.

Figure 6 shows how the list of candidate mappings appears when a new request is sent.

Concept IRI	Score
http://aims.fao.org/aos/agrovoc/c_37978 (Learning)	0.31948918
http://aims.fao.org/aos/agrovoc/c_49835 (ontology learning)	0.18927069
http://aims.fao.org/aos/agrovoc/c_32638 (Animal learning)	0.15439323
http://aims.fao.org/aos/agrovoc/c_49834 (machine learning)	0.10363734
http://aims.fao.org/aos/agrovoc/c_12120 (Learning capacity)	0.080269895
http://aims.fao.org/aos/agrovoc/c_9000055 (Experiential learning)	0.049552567
http://aims.fao.org/aos/agrovoc/c_37466 (mode of action)	0.048363775
http://aims.fao.org/aos/agrovoc/c_7579 (Systemic action)	0.029819423
http://aims.fao.org/aos/agrovoc/c_50149 (Action plans)	0.013072614
http://aims.fao.org/aos/agrovoc/c_543 (Apprenticeship)	0.012131344

Figure 6: Candidate mappings.

4. Remarks

In this Section, we discuss the features of MoKi with respect to the state of the art in order to highlight which were the improvements carried out in the presented version with respect to the real-world knowledge management environment.

In particular, we focused on how MoKi addressed challenges described in Section 1. and how they have been addressed by the other tools. In particular, the facilities that we have considered are the following:

- support to multilingual modeling of the ontology; concerning this point, we do not mean only the possibility of writing axioms about multilinguality (for example, the `<rdfs:label>` tag in OWL); but we analyzed if a complete support for facilitating the exploitation of machine translation services for modeling the domain in a multilingual way has been provided;

- support for collaboration between users; as collaboration, we also consider the facilities for non-formal experts people;
- general ontology management; the support for managing multiple ontologies in the same environment;
- ontology mapping; the support for linking the modeled ontologies with other within the same environment or through external services;
- ontology suggestion; the support of automatic enrichment of ontologies starting from the analysis of text sources;
- ontology exposure; if ontologies are exposed through API interface or Web Services;
- ontology evolution; how the versioning of modeled ontologies is managed and how these information are exposed to third-party services that may be exploit them for further activities.

If we analyze the features of all other tools, we may notice that the only two aspects widely supported are the general ontology management and the ontology exposure. About the first one, even it should be foregone by all ontology management tool, some of them only provide simple mechanisms for editing single ontologies or controlled vocabularies without providing the support for managing more complex scenario like the one in which multiple ontologies that have to be mapped each other. Instead, concerning knowledge exposure, most of the tools provide effective module for publishing modeled data both in a natural language or in a structured way. This aspect is crucial nowadays by considering the importance of connecting different services to the Linked Open Data cloud.

Concerning ontology evolution, some of the comprehensive ontology tools support the versioning of the ontologies and the tracking of their changes; however, it is not specified if these information are exposed or not. In (Dragoni et al., 2011), we explained why it is important also to expose these information. Briefly, if third-party tools have such information available, they may exploit them for activities like tagging and retrieval of resources that are compatible, from a “temporal” point of view, with old versions of the ontologies.

Another aspect that has been taken into account by some tools is the possibility of managing the multilinguality of the ontologies as well as the definition of mappings between concepts. However, while the axiomatization of such information is simple to implement from a technical point of view; it is different the fact of providing a mechanism that is able to support the possibility of plugging different external services providing such facilities (i.e. machine translation and ontology mapping). Indeed, the possibility of plugging different services allows to equip the modeling tools with facilities that may be more effective with respect to the domain that users want to model. Let’s consider as example the agricultural domain: the use of domain-adapted services, i.e. for translating concept labels and descriptions, as well as, the suggestion of candidate mappings with concepts defined in external ontologies, allows

to avoid issues concerning ambiguities and incorrect words in translations, as well as, to improve the effectiveness of the suggested mappings.

Finally, two kind of features that, in general, are not present in almost all tools, or that have been investigated just preliminarily, is the collaboration between users, and the possibility of analyzing external textual sources for retrieving suggestions about possible enrichment of the ontology. Concerning collaboration, we have seen in Section 2., that this aspect has been faced by providing the supporting of working remotely for modeling the same ontology (or vocabulary). Unfortunately, this is not enough for providing a full support for collaborative modeling. Indeed, as “collaboration”, we do not mean just the possibility of modeling something by users geographically distributed, but from users with different expertise. Such users need facilities that allow them to model knowledge without having experience in writing axioms or in formalizing entity relationships. The same is for the ontology suggestion task, where only one tool (i.e. PoolParty) provides the possibility of analysis text for extracting terms that may be exploited as candidate suggestions for enriching the ontology. Indeed, even this feature has been thought for tagging resources, and not for suggesting terms, this is the only facilities approaching the suggestion task.

By performing a “skimming” of the tools presented in the previous section, with respect to the challenges focused in this paper, we identified that the only tools that may be compared with the MoKi are four: *Neon*, *VocBench*, *Protégé*, and *Knoodl*. While the first two, *Neon* and *VocBench*, are the ones more oriented for supporting the management of multilinguality in ontologies; while their support for collaboration is quite limited. On the contrary, *Protégé* and *Knoodl* provide some step ahead with respect to the previous ones concerning collaboration, while their support for multilinguality is restricted to the description of the labels. Finally, an interesting aspect is given by the possibility, in *VocBench* and in *Protégé*, of implementing plugins for extending the core functionalities. This open environment gives the room for the potential implementation of connectors with external services.

MoKi has been evaluated in two different context:

- the usability and the usefulness of the tool has been measured in a real-world use case within the Organic.Lingua EU project. The results and the lessons learned may be found in (Dragoni et al., 2013).
- the quality of the multilingual ontologies modeled in a collaborative way by different experts supported by the tool facilities have been injected in a Cross-Language Information Retrieval system in order to evaluate the impact of such ontologies on systems effectiveness. The results have been presented and discussed in (Dragoni, 2014).

Therefore, we may conclude that the MoKi version presented in this paper advances the general state of the art of the modeling knowledge tools. In particular, the features that significantly highlight the delta with respect to the other tools are the support for the collaboration between

users with different expertise and the implementation of components supporting the plug of different external services for several ontology management tasks like domain-based translations, ontology mappings, and suggestion services for ontology enrichment.

On the contrary, from a critical point of view, the main lack of MoKi is related to its usability that sometimes is quite complicated due to the difficulty of merging all management workflows implemented with the tool. This is the point will be mainly taken into account for the general improvement of the tool.

5. Conclusions

In this paper we presented a tool for modeling multilingual ontologies, but usable also for building any kind of LRs, in a collaborative way. We briefly discussed the challenges concerning the modeling tasks in the collaborative environment, as well as the problematics related to the multilinguality, the exposure, and the linking of the created artifacts. We shown a state of the art of knowledge modeling, and we presented a collaborative modeling wiki-based tool customized in order to provide a set of features able to address the presented challenges.

6. References

- Bellahsene, Zohra, Bonifati, Angela, and Rahm, Erhard, editors. (2011). *Schema Matching and Mapping*. Springer.
- Dimitrova, Vania, Denaux, Ronald, Hart, Glen, Dolbear, Catherine, Holt, Ian, and Cohn, Anthony G. (2008). Involving domain experts in authoring owl ontologies. In *Proceedings of the 7th Int. Semantic Web Conference (ISWC 2008)*, volume 5318/2010 of *LNCS*, pages 1–16. Springer Berlin / Heidelberg.
- Dragoni, Mauro, Ghidini, Chiara, Stoitsis, Giannis, Sicilia, Miguel Angel, and Sanchez-Alonso, Salvador. (2011). Recommendations for revising existing ontologies and schemas. Organic.Lingua EU Project, Deliverable D3.1.1.
- Dragoni, Mauro, Francescomarino, Chiara Di, Ghidini, Chiara, Clemente, Julia, and Alonso, Salvador Sánchez. (2013). Guiding the evolution of a multilingual ontology in a concrete setting. In Cimiano, Philipp, Corcho, Óscar, Presutti, Valentina, Hollink, Laura, and Rudolph, Sebastian, editors, *ESWC*, volume 7882 of *Lecture Notes in Computer Science*, pages 608–622. Springer.
- Dragoni, Mauro. (2014). Revised (final) organic.lingua moki. Organic.Lingua EU Project, Deliverable D3.2.1.3.
- Euzenat, Jérôme and Shvaiko, Pavel. (2007). *Ontology matching*. Springer.
- Ghidini, Chiara, Rospocher, Marco, and Serafini, Luciano. (2012). Conceptual modeling in wikis: a reference architecture and a tool. In *eKNOW2012, Valencia, Spain*, pages 128–135.
- Kaljurand, Kaarel and Fuchs, Norbert E. (2007). Verbalizing owl in attempto controlled english. In *Proceedings of Third International Workshop on OWL: Experiences and Directions, Innsbruck, Austria (6th–7th June 2007)*, volume 258.
- Wikimedia Foundation. (2012). Mediawiki. <http://www.mediawiki.org>.
- McCrae, J., Spohr, D., and Cimiano, P. (2011). Linking lexical resources and ontologies on the semantic web with lemon. In *ESWC (1)*, volume 6643 of *Lecture Notes in Computer Science*, pages 245–259. Springer.
- Palma, Raúl, Corcho, Oscar, Gómez-Pérez, Asunción, and Haase, Peter. (2011). A holistic approach to collaborative ontology development based on change management. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(3).
- William W Cohen, Pradeep Ravikumar, Stephen E Fienberg et al. (2003). A comparison of string distance metrics for name-matching tasks. In *PWorkshop on Information Integration on the Web (IIWeb-03)*, volume 47, 2003, volume 47, pages 73–78.