Towards an Interlinked Semantic Wiki Farm

Alexandre Passant[1,2], Philippe Laublet[1]

[1] LaLIC, Université Paris-Sorbonne,
28 rue Serpente,
75006 Paris, France
firstname.lastname@paris4.sorbonne.fr
[2] Electricité de France Recherche et Développement,
1 avenue du Géneral de Gaulle,
92141 Clamart Cedex, France
firstname.lastname@edf.fr

Abstract. This paper details the main concepts and the architecture of UfoWiki, a semantic wiki farm – i.e. a server of wikis – that uses form-based templates to produce ontology-based knowledge. Moreover, the system allows different wikis to share and interlink ontology instance between each other, so that knowledge can be produced by different and distinct communities in a distributed but collaborative way.

Key words: semantic wikis, wiki farm, linked data, ontology population, named graphs, SIOC

1 Introduction

During the last few years, various Web 2.0 services and principles - such as blogging, wikis, social tagging and social networking - gained interest in corporate environments, leveraging tools that people are more and more used to in their personal life to the enterprise [1]. On the other hand, Semantic Web [2] technologies are used in different business information systems to enrich data integration, querying and browsing, thanks to powerful means to represent knowledge like ontologies and standards to model or query data as RDF and SPARQL.

While some consider Web 2.0 and Semantic Web as being opposite concepts with different origins and goals, we believe as others [3] that these two views should - and even must - be combined to offer easy-to-use but powerful services to end-users. Thus, information systems should benefit from usability and social aspects of Web 2.0 and also from data formalisms of the Semantic Web. It will provide to end users means to collaboratively build, maintain and re-use ontology-based data, a task often dedicated to knowledge management experts, especially in organizations.

In this paper we will describe a semantic wiki-farm system, i.e. a wiki server where communities can setup new wiki instances, called UfoWiki – Unifying Forms and Ontologies in a Wiki – that aims to achieve this goal, currently in use at EDF R&D[3]. The paper is organized as follows. First, we will briefly introduce the limits of classical wikis and various implementations of semantic wikis designed to enhance wiki features thanks to semantics. Then we will introduce the features and the architecture of our semantic wiki farm, as well as novelties compared to current semantic wikis systems, especially the way we combine data and meta-data to keep some information about the knowledge created from wiki pages. We will then describe how people can use one wiki to create ontology instances thanks to form-based templates and emphasize on how the system interlinks data from one wiki to another one but also allows to re-use external data. We will then show how created data can be reused to provide advanced features in a single wiki but also for the complete wiki farm.

2 Wikis and Semantic Wikis for Knowledge Management

2.1 Limits of Traditional Wikis

Among the numerous practices and tools that became popular thanks to Web 2.0, wikis offer new and interesting possibilities regarding collaborative knowledge management. Pages versioning, open plus non-hierarchical editing, hyperlinks and back-links provide useful services to gather and build knowledge within communities and business environments or in open environments as the Web.

Nevertheless, traditional wikis suffer from the difficulty for computers to exploit and reuse the knowledge they contain. A reader could learn from a wiki that EDF is a company that produces nuclear energy in France but a software agent will not be able to easily answer queries like *"Is EDF located in France ?"* or *"List all companies known in that wiki"* without natural language processing algorithms. Indeed, wikis deal with documents and not with machine-understandable representations of real-world concepts and objects, as a reader does when browsing or editing a page. So, a wiki will model that *"There are some hyper-links between a page titled EDF, a page titled France and a page titled nuclear energy"*, but will not be able to deduce anything about the nature of those different objects and their relationships, since pages do not carry enough semantics about the knowledge they contain (Fig. 1).

2.2 The Semantic Web and Ontologies for Better Wikis

To bridge this gap between documents and machine-readable knowledge about real world objects, data must be described in a way software agents interpret and understand uniformly in order to reuse it efficiently. Ontologies [4] and the Semantic Web are effective ways to do so, since they provide common data structures, vocabularies and languages for modeling and querying domains of interest and related individuals. During the last few years and since the first SemWiki workshop [6] various semantic wikis prototypes have been built, combining wiki

[3] Electricit de France, aka EDF, is the leading energy company in France, its R&D department involves about 2000 researchers

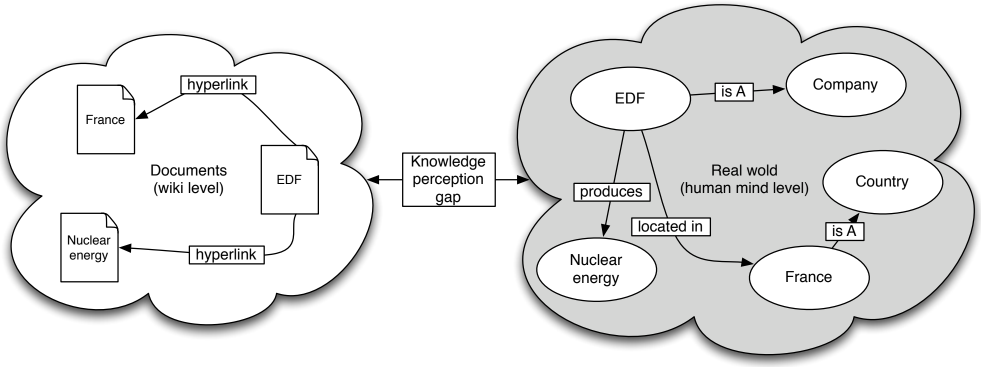


Fig. 1. The gap between documents and real-world knowledge

features and Semantic Web technologies. While tools use different ways to produce this machine-readable data thanks to efforts of their community of users, they all share the common goal of providing value-added services from advanced pages browsing to query answering or even reasoning upon the created dataset.

Systems such as Semantic MediaWiki [7] or SemperWiki [8] require to use a special wiki syntax or to directly embed RDF in order to add semantic annotations to wiki pages. While this is an open approach in the spirit of wiki principles, this can lead to semantic heterogeneity problems since any user can use its own vocabulary to add annotations in a document, making them difficult to re-use. A system like IkeWiki [9] combines plain-text feature of wikis and a dedicated triples-based form interface to help users annotating content by re-using existing ontologies, while OntoWiki [10] can be used as a complete ontology instances editor, with a user-friendly interface that offers different views and browsing and editing interfaces over existing data. Yet, most of those systems require users to have some knowledge about the Semantic Web at a certain time when using it, since they have to deal with namespaces or URIs. This makes the tools difficult to use for people that are not aware of such models, as in business environments where people need to focus on how to use the tools rather than on how he is being build, i.e. benefit from Semantic Web technologies without having to learn them.

In these tools, semantic annotations are mainly used to create and maintain ontology instances and relationships between them, as well as properties, thus providing a real-world and machine-readable representation of the content described inside the pages. They can help to enhance browsing capabilities of the wiki, by suggesting related pages sharing similar instances or listing all pages featuring a certain property as does Semantic MediaWiki. Moreover, new ways to browse the data are available, like in OntoWiki that features map and calendar view of existing data, while some tools provide a back-end RDF store that allows to query data from the whole wiki and embed query results in wiki pages. Finally, some tools also feature inferencing capabilities in order to deduce new knowledge from the current state of the wiki and thus enrich user experience in discovering new knowledge. For example, IkeWIki and OntoWiki can list all instances of a given type taking into account instances of various subclasses. Eventually, it seems important to reference DBpedia [12], a project that aims to represent Wikipedia content in RDF, as well as other semantic wikis, like SweetWiki [11] which does not focus on ontology population but on using semantic web technologies to let users tag their pages and organize those tags, focusing on pages meta-data rather than modeling content of those pages.

3 Modeling a Semantic Wiki Farm

3.1 Main Features of the System

Regarding various aspects of semantic wikis that have been mentioned before, we created UfoWiki, a new semantic wiki farm system - i.e. a wiki server designed to setup and host several wikis - based on the following features, that will be described in the rest of the paper:

- *Ontology-based knowledge representation.* Data created from wiki pages is represented in RDF and is based on a set of ontologies defined by administrators of the wiki in order to avoid semantic heterogeneity problems of data modeling;
- *Usability.* In extent of the previous point and in order to let users easily produce that ontology-based data, we focused on a combination of plain-text and intuitive forms to edit wiki pages, so that users do neither confront to a new syntax or to Semantic Web modeling principles;
- *Interlinking data* While each wiki of the farm acts independent (regarding users that can access it, topics, and modeled knowledge), the system allows different wikis to exchange and interlink their data even if they do not share hyperlinks between each other, thanks to a common knowledge base for the whole system;
- *Modeling both data and meta-data.* While our approach mainly focuses on modeling knowledge contained within wiki pages, we also separately represent the complete wiki server meta-data (wikis, users, pages, tagging actions ...) in RDF, combined with links between those two distinct levels of representation.
- *Immediate reuse of formalized data.* RDF data created among the wikis must be immediately reusable to enhance browsing and querying capabilities of the system, either for a single wiki or the complete farm. Our system uses inline macros, that can provide semantic back-links in the wiki.

3.2 Global architecture

To achieve these goals, our system involves different components. The first part of the architecture consists in a set of ontologies that are used to model RDF data from the wikis, whether it is data about the pages or about their content. For the latter, ontologies must be defined in RDFS or OWL depending on the needs of the knowledge field of the wiki. Regarding the RDF description of wiki pages and user actions, we are using the SIOC ontology [13] and its *type* module[4], a model to describe social media meta-data with unified semantics. We also model tags and tagging actions using the Tag Ontology [14] and the MOAT ontology [15], so that people can give machine-understandable meanings to their tags, especially using URIs of ontology instances created within other wiki pages. Since for all wiki page, data and meta-data are produced within two distinct RDF documents - so that one can export independently each level of representation - we extended the SIOC ontology with a specific property, `embedsKnowledge` in order to formally represent in RDF the link between a wiki page (described in RDF) and the data embedded in it (Fig. 2). This property provides a way to link any instance of `sioc:Item` - and its subclasses - to the URI of a named graph [16], i.e. in practice the URL of a document that contains a set of RDF triples.

Then, the system features its web interface to create wikis, manage wiki forms and browse and edit wiki pages. This interface uses Drupal and is mainly based on a fork of the flexinode module[5] to let wiki owners define their forms. Each form is related to a given class - e.g. people (related to `foaf:Person`) or software project (`doap:Project`) - and each part of the form (a field or a set of fields) can be related to an ontology property and also to a given class, which is used for the autocompletion features of the system. Thus, the editing interface of each wiki combines plain-text and structured parts in order to easily manage the creation of RDF statements according to the ontologies it uses, as we will see on the next section.

The last component of the system is the knowledge base of the wiki farm, storing all created RDF statements thanks to a triple-store, using the 3store[6] API. By storing in real-time all RDF data as well as ontologies in a single place, it offers querying capabilities for the complete data and meta-data of all the wikis, but nevertheless keeps a trace of each statement thanks to its named graphs compliance, so that queries can involve the complete wiki farm data or only data of a given wiki. This store also manages basic inference capabilities (subclasses and subproperties) and supports SPARQL [5] and some SPARUL patterns (SPARQL update[7]) in order to query and update data created from the wiki pages.Moreover, since all wikis of the wiki farm share the same knowledge base, by querying and updating a single RDF store, data can be re-used across wikis. Thus, an ontology instance created in a given wiki can be linked to an ontology instance from another one, even if there is not direct hyperlink between the pages that embeds this knowledge. It allows our system to create knowledge on a distributed way, even between various communities that do not share the

[4] `http://rdfs.org/sioc/types`

[5] `http://drupal.org/modules/flexinode`

[6] `http://threestore.sf.net`

[7] `http://jena.hpl.hp.com/~afs/SPARQL-Update.html`

same wiki but that produce information about the same ontology instances (Fig. 4).

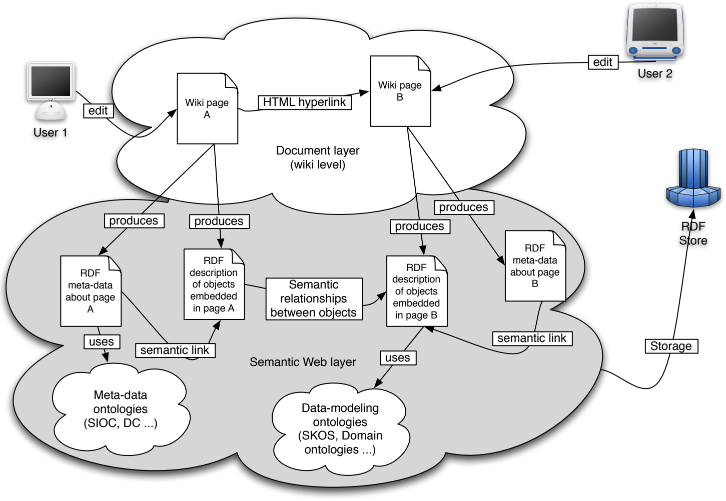

User 1
edit
Wiki page A
HTML hyperlink
Wiki page B
edit
User 2
Document layer (wiki level)
produces
produces
produces
produces
RDF meta-data about page A
RDF description of objects embedded in page A
Semantic relationships between objects
RDF description of objects embedded in page B
RDF meta-data about page B
RDF Store
uses
semantic link
semantic link
Storage
Semantic Web layer
uses
Meta-data ontologies (SIOC, DC ...)
Data-modeling ontologies (SKOS, Domain ontologies ...)

Fig. 2. Architecture of one wiki from the wiki-farm

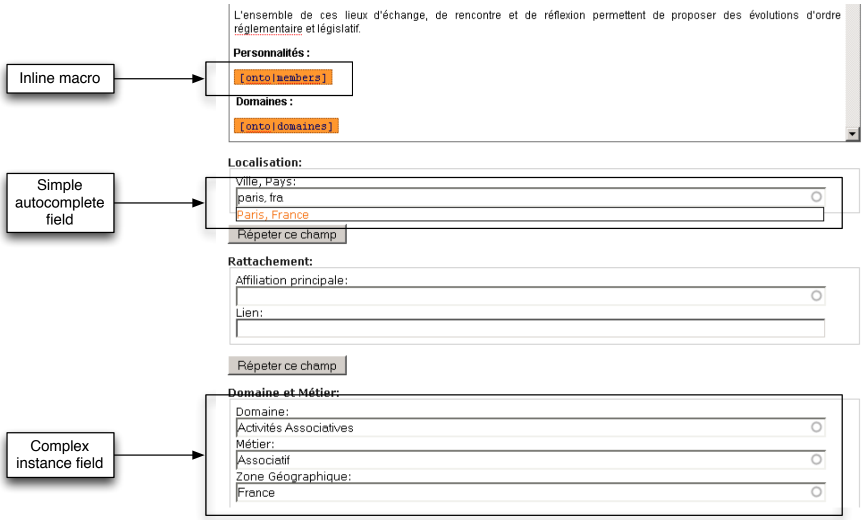
4 Maintaining and interlinking ontology instances between wikis

4.1 Using forms to create and maintain ontology instances

As most semantic wikis, our system automatically creates one main ontology instance for each wiki page, based on the page title. While some wikis do not explicitly assign them a given type and other rely on the page category to define it, our system uses the class assigned to the page form to define it. Regarding definition of properties and relationships of each instance, we use a mix of plain-text and forms in the wiki editing interface, thus separating plain-text content from content to be modeled in RDF, as the Semantic Forms extension[8] for Semantic Wiki or Freebase[9] can do. When creating the page, translation from wiki content to RDF data is then automatically done thanks to the mappings defined by wiki administrators between the form and a set of ontologies. We think that this combination of plain-text and forms to ease the modeling of RDF data (Fig. 3) has numerous advantages:

[8] http://www.mediawiki.org/wiki/Extension:Semantic_Forms
[9] http://www.freebase.com

- First, as fields are defined by the wiki owner for each type of page and so for each class, users know what kind of knowledge is relevant for the wiki regarding a given page and can focus on essential aspects in this context;
- Moreover, as we kept a simple WYSIWYG field for each page, any other relevant information can be added there. It can also help to participate in evolution of the model itself when regular patterns appear, even if the model must be edited manually in this case;
- Users can benefit from autocompletion features, suggesting possible related instances by querying the RDF store with on-the-fly SPARQL queries, thanks to AJAX technologies;
- At last, in our system, this approach allows to create complex relationships and ontology instances inside a single page. While most semantic wikis allow only to create relationships between existing objects, a form part can correspond to a dedicated class in our system, offering better ways to manage complex ontologies population. Moreover, in the page meta-data representation, we distinguish the main instance and the embedded ones, using two subproperties of `sioc:topic` we especially created to achieve this distinction.



L'ensemble de ces lieux d'échange, de rencontre et de réflexion permettent de proposer des évolutions d'ordre réglementaire et législatif.

Personnalités :

[onto|members]

Domaines :

[onto|domaines]

Localisation:

Ville, Pays:
paris, fra ○
Paris, France

Répeter ce champ

Rattachement:

Affiliation principale: ○
Lien:

Répeter ce champ

Domaine et Métier:

Domaine:
Activités Associatives ○
Métier:
Associatif ○
Zone Géographique:
France ○

Fig. 3. Wiki editing interface

While each page corresponds to a given ontology instance, instances are also created for each filled relationship field where a class has been assigned. Then, if one later decides to create a wiki page for these instances, properties will be added to the existing ones. Moreover, when instances are not used anymore in any wiki, i.e. do not have any property, they are automatically removed from the

RDF store to avoid orphan instances. From these aspects, the wiki really acts as a collaborative ontology population tool, beneficing from Web 2.0 features to provide this task. An instance can be created by a user, modified by another, then linked to a third one by another one and even can disappear from the knowledge base if a fourth user edit the page that contains its only reference and removes it.

4.2 Interlinking data between wikis

As we saw in the previous section, our system allows various pages of a given wiki to add information about a single ontology instance. For example, we can create an instance in a wiki page and add a relationship from another instance in a different page than the one that creates it. Yet, our system goes further by allowing two different and disconnected wikis to manage information about the same instance in a distributed way, but keeping the trace of which wiki - and which page - helped to create the information. Thanks to the combination of named graphs and the `embedsKnowledge` property we introduced before, the wiki farm can consider either the whole RDF graph, or subgraphs of RDF statements related to a given wiki only (Fig. 4).

Such a scenario might be useful in some corporate environments, where people do not want to allow anyone to access their wiki, but agree on sharing some expertise and data with others. By exporting only some parts of the wiki page in RDF (i.e. the instances and properties created from some fields of the form page), our model allows the webpage itself to be hidden to not-authorized people while the RDF statements can be exported and become available to a larger community. Moreover, due to our technical architecture that uses SPARQL and SPARUL, the system allows wikis that are distributed on a network (and not from the same wiki farm) to exchange data and interlink it the same way, in case they share a single common RDF database. The system currently do not deal with inconsistency between data from different wikis. We think that this issue should be dedicated to some reasoning engine, that would check inconsistency between produced statements thanks to OWL axioms defined in the ontologies.

Moreover, instead of querying the complete knowledge base, queries can be restricted to data created from a single wiki by using this kind of SPARQL query:

```
select ?page ?title
where {
  graph ?data {
    :EDF ?predicate ?object
  } .
  ?page :embedsKnowledge ?data ;
    rdf:type sioct:WikiArticle ;
    dc:title ?title ;
    sioc:has_container <http://example.org/wiki/6> .
  <http://example.org/wiki/6> a sioct:Wiki .
}
```

This process of combining the two levels of representation can also be used in the autocompletion field, by restricting the autocompletion SPARQL queries to data created from a single wiki, rather than to the whole RDF statements.

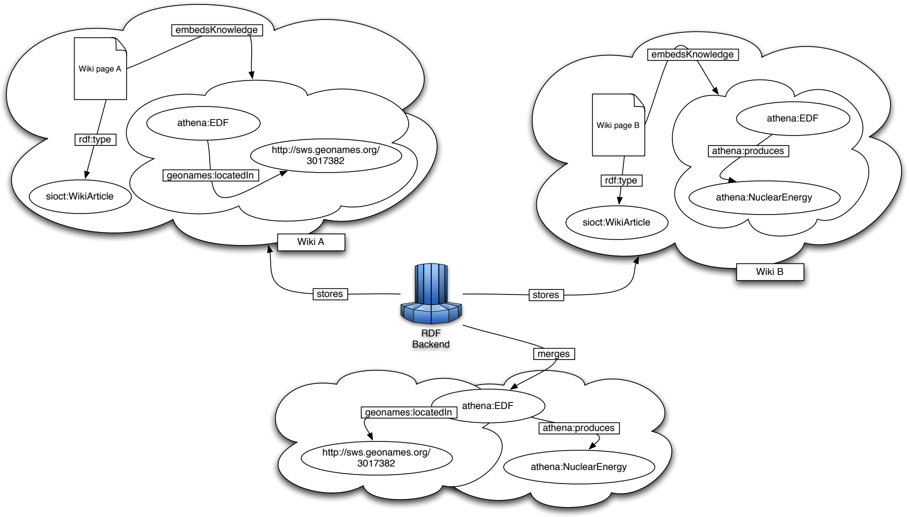


Fig. 4. Interlinking and merging data from different wikis

4.3 Interlinking wiki data with external knowledge

Our system also allows to connect our data to external, publicly available, RDF data. At the moment, a single plug-in is available, to reuse the GeoNames[10] ontology and knowledge base. Each time a form field corresponds to a place and is assigned to the `geonames:Feature` class, the system queries the GeoNames webservice[11] to retrieve the URI of the given instance. Thus, the updated local instances can be linked to external resources, beneficing from a global connection between our data and efforts of communities that help to build such knowledge base. Moreover, we not only link to the URI but also crawl the related RDF file to put in in the wiki knowledge base. Thus, it allow the system to provide geo-location features to end users, without the need for them to type the exact location (i.e. latitude and longitude) of each instance (eg: a people or a company), as they would have done using systems like Semantic MediaWiki and its Semantic Layers extension[12].

[10] `http://www.geonames.org`

[11] `http://ws.geonames.org`

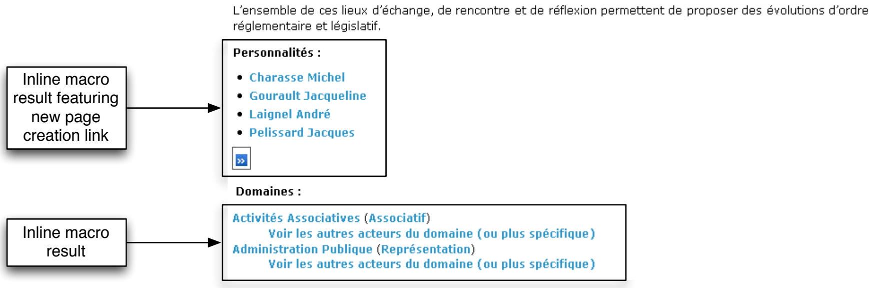
[12] `http://s89238293.onlinehome.us/w/index.php?title=Main_Page`

In the future, we plan to implement new wrappers and linkage systems for other RDF data, especially ways to link to DBpedia extracted knowledge, which can help to provide additional information about instances created within the wikis, and also contribute to the expansion of the Linked Data Web [17]. Regarding this latest point, linking to data from references datasets can help RDF data from our system to be more easily found on the Semantic Web, thanks to lookup services such a Sindice [18] that help to retrieve all resources using and linking to a given URI.

5 Using created data

5.1 Inline macros

The main feature to enhance wiki browsing capabilities in our system is the use of *inline macros*, similar to inline queries of Semantic Mediawiki. Those macros are defined by wiki administrators themselves, using SPARQL and PHP to render the results and are then called by users in wiki pages with simple hooks. Since all data are based on a set of predefined ontologies, queries can be written without having to deal with semantic heterogeneity problems, as people that would have use different property names for the same one, e.g. `isLocatedIn` versus `has_location`. The system then runs the query over the RDF store when the page loads, so that query results are always up-to-date. While queries can be complex, users simply type function names, with some arguments if needed, to use it in wiki pages. For example, `[onto|members]` will be translated in a query that will retrieve all people that are member of the organization described in a wiki page (Fig. 3, Fig. 5). Such queries take inference capabilities of the system into account, so that, for example, if they must list all organizations instances described in the wiki, they will also lists companies or associations if they have been defined as subclasses of the first one in the ontology. Finally, the administrator can decide that the macro will render a link to add new page in the wiki to create an instance of a given type, thus facilitating the process of creating new data.



L'ensemble de ces lieux d'échange, de rencontre et de réflexion permettent de proposer des évolutions d'ordre réglementaire et législatif.

Personnalités :

- Charasse Michel
- Gourault Jacqueline
- Laignel André
- Pelissard Jacques

Domaines :

Activités Associatives (Associatif)
Voir les autres acteurs du domaine (ou plus spécifique)
Administration Publique (Représentation)
Voir les autres acteurs du domaine (ou plus spécifique)

Fig. 5. Browsing an enhanced wiki page

Moreover, macros can take into account the way we combine modeling of data and meta-data in RDF export of wiki pages, so that a wiki can display a list of pages from another wiki for a given query, as the previous SPARQL snippet showed. It allows one wiki to benefit from the effort of another community done in another wiki.

More generally, such queries can be seen as a way to move from classical wiki back-links to semantic back-links, as we bridged the gap between documents and Semantic Web formalized data. While a typical wiki could list thanks to its back-link feature that an organization page has an incoming link from a people page, our system takes advantage of the data formalism to be more specific about the nature of this link, mentioning that this company employs that person, going from the document to the data layer.

5.2 Advanced data view

Finally, those macros can display results according other rendering inter-faces, such as Google Maps, in case the needed geo-location information is available in the RDF store thanks to the integration of the GeoNames lookup service. Thus, while the result is similar to what can be done with the map view of OntoWiki, users do not have to manually enter the coordinates of each instance (e.g. a company) but simply fill a *"City, (State), Country"* field, that will be used to retrieve the appropriate RDF data - including coordinates - from GeoNames an add it in our knowledge base. Here, we clearly see the benefit of using the same model (i.e. the GeoNames ontology) than an existing RDF dataset to include data from external services at zero-cost. The Fig. 6 displays the output map of a macro that retrieves the location of a given association and all of its members from a single wiki interlinked with GeoNames data in a single SPARQL query.

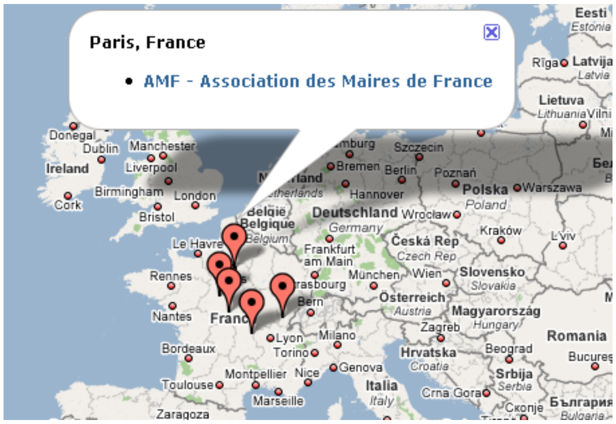


Fig. 6. Map view of the wiki data

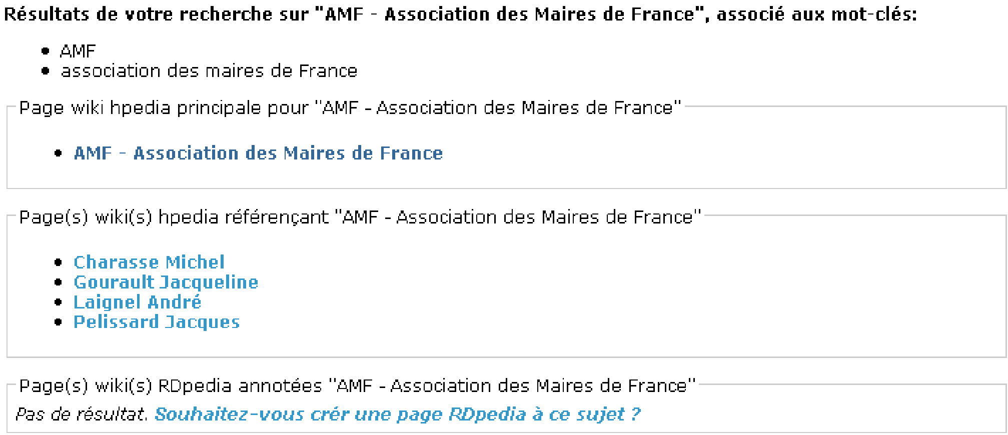
5.3 Semantic search

Another feature of the system is a dedicated semantic search engine, taking into account existing instances described within the wiki (or used in a semantic tagging process) rather than plain-text only when retrieving data. When a user search for a given term in the wiki farm, the system first finds the list of all instances related to this label, using (1) `rdfs:label` that can have be defined thanks to the wiki pages and dedicated forms and (2) the `moat:Tag` instances that contains this term within their label and that are linked to existing instances thanks to a related `moat:Meaning`. Thus, if a user type the search term *"France"*, the system will ask the user if he requires information about *"EDF"* (since it has *"Electricit de France"* as a tag) but also, of course, the *"France"* concept.

Then, the system will list independently:

- All wiki pages - for each wiki, identified by their name - that have this instance as a main topic;
- All wiki pages where the instance is an "alternative" topic (i.e. an instance created within a page);
- All wiki pages *"tagged"* (thanks to MOAT) with this instance.

Thus, it offers various meta-data representation of the wiki.



Résultats de votre recherche sur "AMF - Association des Maires de France", associé aux mot-clés:

- AMF
- association des maires de France

Page wiki hpedia principale pour "AMF - Association des Maires de France"

- **AMF - Association des Maires de France**

Page(s) wiki(s) hpedia référençant "AMF - Association des Maires de France"

- **Charasse Michel**
- **Gourault Jacqueline**
- **Laignel André**
- **Pelissard Jacques**

Page(s) wiki(s) RDpedia annotées "AMF - Association des Maires de France"

Pas de résultat. ***Souhaitez-vous crér une page RDpedia à ce sujet ?***

Fig. 7. Semantic search results example

Moreover, while we do not currently provide user friendly interface to generate new queries or macros, advanced users can run SPARQL queries over the RDF data.

6 Conclusion and future works

In this paper, we described a prototype of wiki that combine structure and Semantic Web modeling capabilities to produce ontology-based and machine-readable data in a collaborative way. We showed how various wikis could be used to model and interlink knowledge about ontology instances in an open and distributed way. We finally showed how such knowledge can be used to enrich functionalities of the wiki. While this system combines some features that already exist in various prototypes, it focuses on usability for end-users, as well as, from the technical side, a way to model and link both data and meta-data, offering capabilities to view different levels of annotation, either from a single wiki or for the complete set of wikis.

The system is currently in use at EDF R&D, where users have created more than 200 instances from various lightweight ontologies. We extensively use the GeoNames integration, making the geo-location feature easy to integrate in order to provide new and interesting ways to browse the wiki content. Inline macros are also useful for end-users since they allow to easily find instances and related wiki pages. For example, we included a macro that lists, in a page dedicated to some company, all other organizations working on the same topics.

Regarding our future works, we will concentrate on adding new value-added functionalities to the wiki for end-users to ease the discovery of relevant information from the set of RDF data, as faceted browsing, as well as interlinking with other existing datasets. We will also focus on how to formalize wiki pages versioning in RDF, in order to see how statements about a given resource can evolve during its lifetime and track more precisely each change of information on a given ontology instance.

Acknowledgements

We would like to thanks the ID-Net team from EDF R&D for their input on the current experiments about our wiki farm.

References

1. McAfee A. P.: Enterprise 2.0: The Dawn of Emergent Collaboration. MIT Sloan Management Review, 47 (3), pp 21-28 (2006).
2. Berners-Lee T., Hendler J., Lassila O.: The Semantic Web. Scientific American, 284 (5), pp 34-43 (2001)
3. Heath T., Motta E.: Ease of interaction plus ease of integration: Combining Web2.0 and the Semantic Web in a reviewing site. Journal of Web Semantics 6 (1) (2008)
4. Gruber T.R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. Formal Ontology in Conceptual Analysis and Knowledge Representation. Guarino N., Poli R. (eds). Kluwer Academic Publishers, Deventer, The Netherlands (1993)

5. Prud'hommeaux E., Seaborne, A.: SPARQL Query Language for RDF. W3C Recommendation 15 January 2008, http://www.w3.org/TR/rdf-sparql-query/ (2008).
6. Schaffert S., Vlkel M., Decker S.: First Workshop on Semantic Wikis: From Wiki to Semantics (SemWiki2006]) at the 3rd Annual European Semantic Web Conference (ESWC), Budva, Montenegro (2006)
7. Krötzsch M., Vrandecic D., Völkel M.: Semantic MediaWiki. Proceedings of the 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, pp 935-942 (2006)
8. Oren E.: SemperWiki: A Semantic Personal Wiki.: Proceedings of Semantic Desktop Workshop at the ISWC2005, Galway, Ireland (2005)
9. Schaffert S.: IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. 1st International Workshop on Semantic Technologies in Collaborative Applications (STICA'06) (2006)
10. Auer S., Dietzold S., Riechert T.: OntoWiki - A Tool for Social, Semantic Collaboration. Proceedings of 5th International Semantic Web Conference 2006 pp 736-749 (2006)
11. Buffa M., Gandon F.L., Sander P., Faron C., Ereteo G.: SweetWiki: a semantic wiki. Journal of Web Semantics 6 (1) (2008)
12. Auer S., Bizer C., Lehmann J., Kobilarov G., Cyganiak R., Ives Z.: DBpedia: A Nucleus for a Web of Open Data. Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007), Busan, South Korea pp 715-728 (2007)
13. Breslin J.G., Harth A., Bojars U., Decker S.: Towards Semantically-Interlinked Online Communities. Proceedings of the 2nd European Semantic Web Conference (ESWC '05), LNCS vol. 3532, pp. 500-514, Heraklion, Greece (2005)
14. Newman R.: Tag Ontology Design. http://www.holygoat.co.uk/projects/tags/ (2005)
15. Passant A., Laublet P.: Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data. Proceedings of the Linked Data on the Web (LDOW2008) workshop at WWW2008, Beijing, China (2008)
16. Carroll J., Bizer C., Hayes P., Stickler P.: Named Graphs, Provenance and Trust. Proceedings The Fourteenth International World Wide Web Conference (WWW2005), Chiba, Japan (2005)
17. Bizer C., Heath T., Ayers D., Raimond Y.: Interlinking Open Data on the Web. Poster, 4th Annual European Semantic Web Conference (ESWC2007), Innsbruck, Austria (2007)
18. Tummarello G., Oren E., Delbru R.: Sindice.com: Weaving the Open Linked Data. Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007), Busan, South Korea 547–560 (2007)