Enabling cross-wikis integration by extending the SIOC ontology

Fabrizio Orlandi[1] and Alexandre Passant[2]

[1] Università degli studi di Modena e Reggio Emilia,
Modena, Italy
orlandi.fabrizio.31985@unimore.it
[2] Digital Enterprise Research Institute,
National University of Ireland, Galway
alexandre.passant@deri.org

Abstract. This paper describes how we extended the SIOC ontology to take into account particular aspects of wikis in order to enable integration capabilities between various wiki systems. In particular, we will overview the proposed extensions and detail a webservice providing SIOC data from any MediaWiki instance, as well as related query examples that show how different wikis, designed as independant data silos, can be uniformally queried and interlinked.

Key words: Semantic Web, SIOC, wikis, MediaWiki, Social Semantic Web, Linked Data, DBpedia

1 Introduction

The SIOC Ontology – Semantically-Interlinked Online Communities [4] – is now considered as one of the building blocks of the *"Social Semantic Web"*. More than 50 applications are currently using SIOC[3], either as a common vocabulary to expose their data in RDF, alongside with FOAF for instance, as well as using existing SIOC data, as for instance Yahoo! SearchMonkey. Moreover, the use of SIOC goes further than mainstream Web 2.0 services, from Enterprise 2.0 information integration[4] to Health Care and Life Sciences discourse representation[5].

However, only a few work have been done so far regarding wikis, semantic wikis and the SIOC ontology. While the SIOC Types[6] module already provides the `Wiki` and `WikiArticle` classes that can be used to represent the basic objects manipulated by wikis, some particular features of wikis such as pages versioning and backlinks are not taken into account, neither in the SIOC core nor in its modules. Yet, providing wikis information using SIOC would have several advantages in terms of integration with existing and constantly dynamically-created

[3] `http://sioc-project.org/applications`
[4] `http://www.w3.org/2001/sw/sweo/public/UseCases/EDF/`
[5] `http://esw.w3.org/topic/HCLSIG/SWANSIOC`
[6] `http://rdfs.org/sioc/types`

SIOC data, as well as interlinking with other RDF data for advanced querying purposes. For instance, one will be able to run the same SPARQL query to find latest created items on a MediaWiki instance or on a WordPress weblog. Hence, we recently worked on extending the SIOC ontology for this purposes, as well as providing a SIOC exporter for MediaWiki, potentially creating millions of SIOC-based RDF documents from various popular wiki services.

This paper is organized as follows. First, we will go through an overview of wiki features that are important to consider in such a modeling approach and explain how we took them in consideration in regards of the SIOC ontology and how we extended it based on this analysis. Second, since some wikis already expose their data in a machine-readable form thanks to Semantic Web technologies, we will focus on a state of the art of existing models that achieve the same goal. Then we will detail how we built a webservice that translates any MediaWiki wiki page to RDF using the newly-extended SIOC ontology. We will particularly focus on how this service produces RDF data compliant with the Linked Data principles and how it relates to initiatives such as DBpedia. Then we will show some relevant query examples, from advanced queries in a single wiki to cross-querying capabilities. Finally, we will conclude the paper with an overview of future works on the domain.

2 Using and extending the SIOC ontology for advanced modeling of wiki structure

In this section we spot and explore what we consider being the typical and fundamental features of wikis in terms of structure and social interactions. Typically wikis allow editing of documents and, by definition, allow multiple users to simultaneously contribute to the content; they track history of changes so that pages can be restored to previous modified versions; they include comments or discussion areas; they link to other external sources or within the wiki; they describe categories into hierarchical structures. For each identified feature, we give a brief overview of its goal, and detail how we extended, and generally use, the SIOC Core ontology[7] and its Types module[8] taking each feature into account in our modeling approach.

2.1 Modeling relevant wiki features

Multi-authoring. A fundamental feature of wikis is that multiple users are allowed to modify the same content, enabling some kind of collective intelligence process. In this regard the semantic infrastructure should provide a model to identify users and theirs modifications, marking events with a corresponding timestamp.

This feature can be modeled using the class `sioc:User` as object of the property `sioc:has_creator` that describes a user account in an online community

[7] `http://rdfs.org/sioc/ns`, prefix `sioc` in this document
[8] `http://rdfs.org/sioc/types`, prefix `sioct` in this document

site, and which is a subclass of `foaf:OnlineAccount`. In this way a `foaf:Person` could be linked to several `sioc:User` belonging to different wiki sites. Another way to model the relationships between pages and their authors is to reuse properties from the Dublin Core ontology, i.e. `dc:contributor` (or `dc:creator`) and `dcterms:created`. Yet, these properties do not link to a user URI but to a simple text string, which can be an issue when querying information, especially for cross-querying as we will detail in section 5.

Categories. Wiki pages are generally related to categories, that allow readers to find sets of articles on related topics. Categories can also be organized in a tree-like structure and their semantic model should mantain the original taxonomical structure. In this regard an appropriate solution is provided by the SKOS[9] vocabulary [11], as it offers a way to model hierarchical structures between various categories, represented as instances of `skos:Concept`.

As regards the SIOC ontology, a `sioct:Category` class was already present into the SIOC Types module, allowing only the modeling of a flat set of category names. Hence, we decided to declare this class as a subclass of a `skos:Concept`, giving it the ability to use the wide SKOS ontology capabilities to organize categories into advanced taxonomies. Moreover, thanks to the `sioc:topic` property, one can link any wiki page to such category.

Social tagging. While not all wiki engines support that feature, we believe this is particularly relevant, especially as it offers an open and user-driven classification scheme for wiki pages. The use of tags lead to a non-organised but dynamic organisation process, known as a "folksonomy", rather than the more widely used hierarchical structures.

The properties `sioc:topic` and `dc:subject` can be used to represent tags related to a particular wiki page, either using URIs for these tags (with `sioc:topic`) or simple keywords (`dc:subject`). In addition, vocabularies such as the Tag ontology [12], SCOT [8] or MOAT [13] allow to model tagging as tripartite actions (between a wiki page, a user and a tag) as well as organize tags together or link them to ontology concepts, in order to solve common tagging issues such as ambiguity between tags.

Discussions. Several wikis associate a discussion page to every wiki page, so that each user is able to comment and argue his point-of-view on the topic. On a discussion page, people can discuss about the article subject, or about the way that subject is presented (see the Wikipedia's approach[10]). A first modeling solution could be to simply keep the native wiki text format of the wiki and just semantically link the discussion page to the related article page.

The SIOC's main class responsible for the modeling of a discussion is the `sioc:Forum` class, but there could be other specific classes that are more suitable

---
[9] `http://www.w3.org/2004/02/skos/`
[10] `http://en.wikipedia.org/wiki/Wikipedia:Talk_page_guidelines`

for these discussion purposes, as defined in the Types module. The appropriate class to choose depends also on the type and style of the discussion page. So it has been necessary to identify a proper attribute to link a wiki page to its discussion page. In this regard we decided to add a `sioc:has_discussion` property to the SIOC Core ontology, with domain `sioc:Item` and open range. This choice has been done in order to make this property reusable also in other contexts, for instance linking a simple webpage to a discussion forum. The discussions happening within the related `sioc:Forum` can then be modeled either as wiki-style discussions or threaded ones, and that feature also allows us to re-use advanced SIOC-based argumentative discussion modeling as defined in [10].

Backlinks. Backlinks are an important feature of wikis, as they allow to visualize instantaneously all the incoming links to a website or web page. More precisely they are wiki internal links pointing to a wiki article. It is a very common wiki feature and they may be of significant interest: they indicate who is paying attention to the linked page or topic.

We modeled this feature using the already existing `sioc:links_to` property. This property identifies links extracted from hyperlinks within a SIOC concept and is a subproperty of `dcterms:references`. It is important to remember that this property has to be defined into the RDF description of the original wiki article which links back to the wiki article. Hence, to model for instance that the Wikipedia page about "DERI" features a backlink from the page about "RDF", the following statement would be added into the RDF description of DERI's page.

```
<http://en.wikipedia.org/wiki/
    Resource_Description_Framework> sioc:links_to
 <http://en.wikipedia.org/wiki/
     Digital_Enterprise_Research_Institute> .
```

Listing 1.1. Representing backlinks

Pages versioning. Usually all editable pages on wikis have an associated page history. This history consists of the old versions of the wikitext, as well as a record of the date and time of every edit, the username or IP address of the user who wrote it, and their edit summary. All this is usually accessible through a special *"history"* page which shows time-ordered links to all the revisions. Commonly the latest revision of a wiki page has always the same URL (alias name), meanwhile older versions have further parameters appended to the URL.

The versioning of pages could be modeled in several ways. Taking into account other semantic wikis, that we describe in Section 3, we took inspiration from other existing approaches. Then we defined our own different model because we wanted to keep the pros of each model and we did not find one capable to satisfy completely our needs. An important requirement we take into account is the fast and simple browsing capability that the model should have. For this reason we chose to use transitive properties to express the temporal relation between revisions of a wiki page. The model is displayed in Fig. 1 and all the used properties are now defined in the SIOC Core ontology.

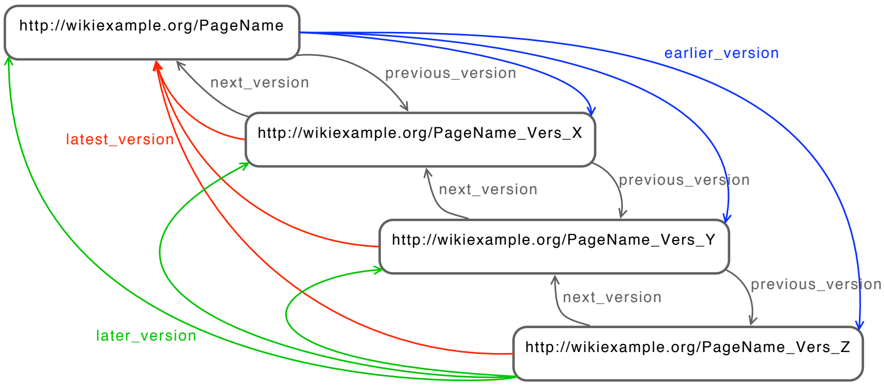


Fig. 1. Pages versioning model with SIOC properties. Please note that, for more clarity, transitive properties `earlier_version` and `later_version` are only displayed for two wiki articles: the latest one and the first one.

The `next/previous_version` properties link only the next/previous revision of a generic `sioc:Item`. Meanwhile `earlier/later_version` are defined as transitive properties and as super-properties respectively of `next/previous_version`. The main advantage of the definition of transitivity and the declaration of super-properties, is that they can be inferred automatically by a reasoner. Hence, using an OWL level reasoning engine, when modeling a `WikiArticle` (or a `sioc:Item` in general), it is only necessary to describe its previous and next revision and the transitive super-properties will be automatically inferred by the system. This can also be convenient during the querying process (described in Section 5): if it is necessary to get all the earlier versions of a wiki page, with transitivity it is sufficient to use the `sioc:earlier_version` transitive property, while in the other case, it has to be implemented a query that recursively "jumps" on each `sioc:previous_version` of the latest wiki article. Another introduced property is the `sioc:latest_version` which points always to the newest revision. Usually it is used in combination with an alias name of the latest version so that it is not necessary to change the referred URI in all the earlier versions as soon as a modification happens. All the wikis we analyzed adopt this solution as it addresses scalability.

2.2 Changes summary

All the changes we made to the SIOC ontology are summarized as follows:

– Defined the `sioct:Category` class as a subclass of `skos:Concept`.

– Added a sioc:has_discussion property, with domain sioc:Item and open range.
– Added a sioc:latest_version property, with sioc:Item as domain and range.
– Added two transitive properties: sioc:earlier_version and sioc:later_version, with sioc:Item as domain and range.
– Defined sioc:later_version as inverse property of sioc:earlier_version.
– Defined sioc:next_version as a subproperty of sioc:later_version.
– Defined sioc:previous_version as a subproperty of sioc:earlier_version.

3 Related work

3.1 Existing models to represent structure of wikis

While our aim is to model wiki features by extending SIOC, several vocabularies have been already proposed to achieve this goal. We will overview some of them in this section, by distinguishing models created with a general purpose and models created for a particular wiki engine but nevertheless available on the Web. In addition, it is worth noticing that we focus here on models and tools defined to represent the structure of wikis and not on the ones that allow domain ontology modeling and population in RDF(S)/OWL from the wiki itself, while these two levels of semantics can be obviously combined.

As regards generic models **WikiOnt**[11] [7] is an ontology for describing and exchanging wiki articles and it aims at integrating Wikipedia (and by extension other MediaWiki-based sites) into the Semantic Web framework, making Wikipedia machine-processable. This OWL ontology uses DublinCore to identify multiple authors of wiki pages as well as the editing date, and provides Article and Category classes. Yet all the other features are currently not modeled by the ontology. **Wiki Interchange Format**[12] (WIF) [16] is a project that allows data exchange between wikis and related tools. Different from other approaches, it also tackles the problem of page content and annotations. WIF defines a subset of XHTML as an over-the-wire format for wiki content exchange. It defines the classes of WikiUser (subclassing foaf:Person) and WikiPage to model pages and authors. It also privides a versioning system thanks to the hasPreviousVersion and hasChangeDate properties. Categories, social tagging, discussions and backlinks are features currently not modeled by the ontology.

In addition to the previous models we considered some particular wiki engines that expose their data in RDF providing an RDF vocabulary for such export. **IkeWiki**[13] [15] aims at creating instance data based on an existing ontology but also at being a tool for creating and editing ontologies. In addition, IkeWiki provides a complete export of the wiki structure using a dedicated OWL ontology. It introduces a User class, subclass of foaf:Person, and uses a hasAuthor

11 http://sw.deri.org/2005/04/wikipedia/wikiont.html
12 http://wif.ontoware.org/2005/04/
13 http://ikewiki.salzburgresearch.at

property, subclass of `foaf:maker`, to associate an author/User with a resource in the wiki. It is also worth noticing that IkeWiki uses SIOC to model discussions, using a `hasDiscussion` property that links any wiki page to an instance of `sioc:Forum`. A wiki article is defined by a `Page` class. Social tagging, backlinks and pages versioning are features currently not modeled by the ontology.
SweetWiki[14] [6] is a semantic wiki based on the CORESE engine. It relies on Web standards for the semantic annotations (RDFa, RDF) and for the ontologies it manipulates (OWL Lite). The SweetWiki ontology manages versions of wiki pages using a `Version` class, defined as a subclass of the main `WikiPage` class. A `pageHasVersion` property links each old version with the latest page represented by the `WikiPage` class. The page version number is declared as an integer number with the `isTheVersionNumber` attribute. To note that SweetWiki offers advanced social tagging features, which are not modeled by other Semantic wikis. Keywords can be collaboratively structured through a lightweight ontology editor and related either to pages, categories (defined using a particular `Category` class) or embedded media content. Sweetwiki also supports backlinks, but not discussion pages at the moment. It also defines its own classes to model authors and wiki pages, respectively using `Person` and `WikiPage`.
Semantic MediaWiki[15] (SMW) [9] uses a particular ontology to represent the semantic data exported from a page by a user, named *SWIVT* – Semantic Wiki Vocabulary and Terminology. This ontology provides a basis for interpreting the semantic data exported by SMW, and it incorporates various elements that are closely related to SMW's metadata model. Yet, while some features such as backlinks or categories are provided by SMW, they are not exposed in the SWIVT model, exporting only a `Wikipage` class.

Finally, regarding the use of SIOC for wikis, we can mention UfoWiki [14], that have been deployed in complement of other SIOC-related data in an Enterprise 2.0 platform, while it does not support versioning in its RDF representation.

3.2 Comparison and positioning of our approach

Based on the previous analysis, we produced a comparison matrix, to underline the pros and cons of each approach. We may conclude that multi-authoring is a feature supported by the ontologies of all the wiki models, and this is because it is an inescapable characteristic of a semantic wiki. On the other hand backlinks and versioning are not modeled by most of the considered wikis. These two features are addressed only by SweetWiki, with the exception of WIF developing a very simple versioning solution. Moreover, some features that are not modeled by these vocabularies could be added by external vocabularies, as for instance social tagging. Finally the most complete model to take into account is SweetWiki, being able to accomplish to every requirement but the discussions.

[14] `http://sweetwiki.inria.fr/sweetwiki`
[15] `http://semantic-mediawiki.org`

×	IkeWiki	SweetWiki	SWIVT	WikiOnt	WIF	SIOC
Multi-authoring	√	√	√	√	√	√
Categories	√	√	√	√	×	√
Social Tagging	×	√	×	×	×	√
Discussions	√	×	×	×	×	√
Backlinks	×	√	×	×	×	√
Versioning	×	√	×	×	√	√

Table 1. Comparing various ontologies to represent wikis structure

4 Exporting SIOC data from MediaWiki

In order to see implications of our extension, our aim was to build an exporter from a popular wiki platform, so that it can expose its data in RDF using our proposed model. Hence we decided to create a web-service application to export any MediaWiki instance. MediaWiki is one of the most popular wiki platforms, hosting all the Wikimedia Fundation wikis (i.e. Wikipedia, Wiktionary, etc.) and propulsing more that 25 millions of wiki articles from different wiki sites[16].

4.1 Principles of the SIOC-MediaWiki webservice

In order to export SIOC data from MediaWiki's wikis we implemented a web-service, written in PHP, that exports a wiki article in RDF with the structure we explained in the previous sections. The webservice is publicly available at `http://ws.sioc-project.org/mediawiki/`.

Fig. 2. Interface of the SIOC-MediaWiki webservice

[16] `http://s23.org/wikistats/largest_html.php`

The MediaWiki exporter is relatively lightweight and built thanks to two PHP classes: the SIOC-Mediawiki exporter itself and the already existing SIOC API[17]. Our approach combines the use of the MediaWiki API as well as the SIOC PHP API – that has been extended based on the previously-detailed ontology changes – to create SIOC data. The exporter class is the part responsible for querying the MediaWiki API and parsing the results, and the SIOC API is responsible for exporting the content in RDF. The script indeed uses the MediaWiki API to get all the information about the article inserted in the form, with the following process (as represented in Fig. 3):

- it automatically discovers the API location (if not detected it is possible to manually specify the API path in the proper text field);
- it connects to the API sending HTTP requests as queries;
- it parses the results of the queries and fills in the proper variables;
- it calls the SIOC API to export in RDF the fetched structural information and outputs the results in RDF/XML serialization.

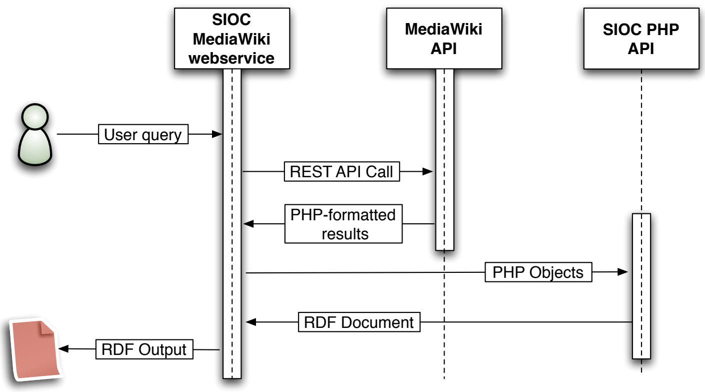


Fig. 3. Activity diagram of the SIOC-MediaWiki webservice

4.2 Following the Linked Data principles

One of our goals with this exporter was not only to create RDF data from any MediaWiki page, but also to easily allow interlinking between various wikis, as well as between wiki data and other RDF data, whatever it is social data modeled with FOAF or SIOC or any other kind of RDF data. Hence, we followed the Linked Data best practices defined in [1], [2] and [3].

Particularly, to offer a better browsing experience and ease the process of crawling SIOC exports of MediaWiki instances, our webservice automatically

[17] `http://wiki.sioc-project.org/index.php/PHPExportAPI`

produces `rdfs:seeAlso` links between wiki pages. Actually, more than a simple link to the wiki page, the exporter provides a link to the related RDF document, as we can see in the following Listing 1.2 related to a particular `sioc:User`. As we can also notice in that example, we distinguish the concept itself (i.e. `User:StefanDecker`) and the related RDF page. These `seeAlso` links are very useful not only to provide link to other related RDF documents, that can be used for instance when browsing data with Tabulator, but also in a crawling perspective. A RDF crawler could easily follow all the `seeAlso` links found on every document and continue to crawl. In this regard, for example, we crawled and exported entire wiki sites just following these links.

```
<sioc:User rdf:about="http://en.wikipedia.org/wiki/User:
    StefanDecker">
 <rdfs:seeAlso rdf:resource="http://ws.sioc-project.org/
     mediawiki/mediawiki.php?wiki=http://en.wikipedia.org
     /wiki/User:StefanDecker"/>
</sioc:User>
```

Listing 1.2. Modeling a user in the MediaWiki exporter

Another interesting feature is the linkage to the corresponding DBpedia[18] resource, if the article belongs to the english Wikipedia. Since DBpedia semantically models the content of a Wikipedia page, this connection is very useful to link semantic data about the content and the structure of a wiki article. DBpedia resource URIs are used in rage of the `foaf:primaryTopic` property, this because it relates a document to the main thing that the document is about.

5 Cross-wikis integration and advanced querying process

In order to evaluate our proposal, we exported and crawled different MediaWiki instances. Four different wikis have been crawled – using for each crawl a single entry point thanks to the use of the `rdfs:seeAlso` links – each one belonging to the same area of interest in order to have a high probability of shared topics and users: *Semanticweb.org*[19], *Protégé Wiki*[20], *RDFa Wiki*[21] and the *ONTOLORE Karlsruhe* wiki[22]. In total, we collected about 1GB of RDF data and loaded it in Sesame [5]. As we needed an higher degree of inference (because of the OWL transitive properties) we installed and configured the reasoning engine OWLIM[23] on the top of it.

[18] `http://www.dbpedia.org`
[19] `http://www.semanticweb.org`
[20] `http://protegewiki.stanford.edu`
[21] `http://rdfa.info/wiki/RDFa_Wiki`
[22] `http://logic.aifb.uni-karlsruhe.de/wiki/ONTOLORE`
[23] `http://www.ontotext.com/owlim/`

5.1 Advanced querying for a single wiki

A first example of advanced querying for a particular wiki is the ability to answer to the following question: "what are the collaborating users that worked alternatively on the same wiki article?". In Listing 1.3 we provide the SPARQL implementation of this query.

```
SELECT DISTINCT ?wikiArt ?Contrib_a ?Contrib_b
WHERE {
 ?x sioc:latest_version ?wikiArt.
 ?wikiArt sioc:earlier_version ?VersA .
 ?VersA sioc:earlier_version ?VersB ;
  dc:contributor ?Contrib_a .
 ?VersB sioc:earlier_version ?VersC ;
  dc:contributor ?Contrib_b .
 ?VersC dc:contributor ?Contrib_a .
 FILTER (?Contrib_a != ?Contrib_b) .
}
```

Listing 1.3. Identifying collaborating users

In Fig. 4 we display a diagram that summarizes the above query, meanwhile in Fig. 5 we show the results we got querying on our SESAME triplestore. As we can see, this query takes advantage of the transitivity of the newly created property `sioc:earlier_version`, since we identify users that worked on earlier versions, and not only immediately on the previous one.

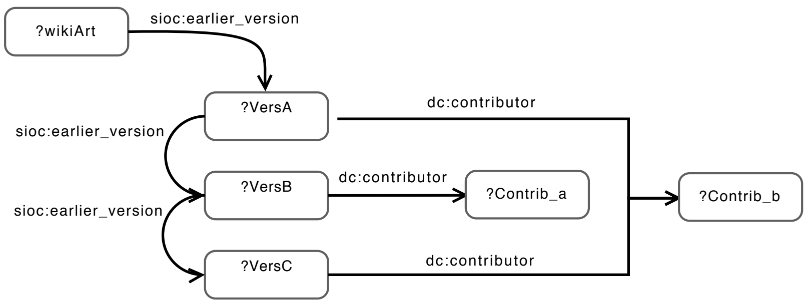
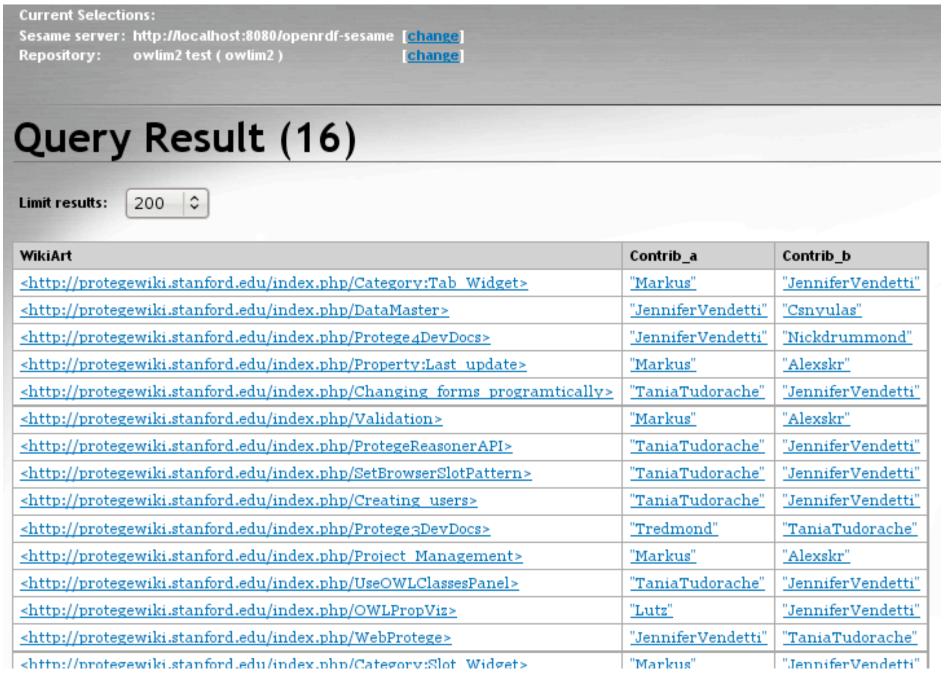


Fig. 4. Identifying collaborating users

The query results provide the article URI and the two usernames in case the first user (`?Contrib_a`) re-edited the article after a modification made by the second user (`?Contrib_b`). It enables you to look for users sharing the same interests and knowledge areas. The query is important expecially in a social semantic context.



WikiArt	Contrib_a	Contrib_b
<http://protegewiki.stanford.edu/index.php/Category:Tab_Widget>	"Markus"	"JenniferVendetti"
<http://protegewiki.stanford.edu/index.php/DataMaster>	"JenniferVendetti"	"Csnyulas"
<http://protegewiki.stanford.edu/index.php/Protege4DevDocs>	"JenniferVendetti"	"Nickdrummond"
<http://protegewiki.stanford.edu/index.php/Property:Last_update>	"Markus"	"Alexskr"
<http://protegewiki.stanford.edu/index.php/Changing_forms_programtically>	"TaniaTudorache"	"JenniferVendetti"
<http://protegewiki.stanford.edu/index.php/Validation>	"Markus"	"Alexskr"
<http://protegewiki.stanford.edu/index.php/ProtegeReasonerAPI>	"TaniaTudorache"	"JenniferVendetti"
<http://protegewiki.stanford.edu/index.php/SetBrowserSlotPattern>	"TaniaTudorache"	"JenniferVendetti"
<http://protegewiki.stanford.edu/index.php/Creating_users>	"TaniaTudorache"	"JenniferVendetti"
<http://protegewiki.stanford.edu/index.php/Protege3DevDocs>	"Tredmond"	"TaniaTudorache"
<http://protegewiki.stanford.edu/index.php/Project_Management>	"Markus"	"Alexskr"
<http://protegewiki.stanford.edu/index.php/UseOWLClassesPanel>	"TaniaTudorache"	"JenniferVendetti"
<http://protegewiki.stanford.edu/index.php/OWLPropViz>	"Lutz"	"JenniferVendetti"
<http://protegewiki.stanford.edu/index.php/WebProtege>	"JenniferVendetti"	"TaniaTudorache"
<http://protegewiki.stanford.edu/index.php/Category:Slot_Widget>	"Markus"	"JenniferVendetti"

Fig. 5. Results of the query for collaborating users on SESAME

5.2 Cross-wiki integration and querying

Another interesting feature of our approach is the ability to do cross-wikis querying, since they are based on the same model. Obviously, one can argue that since all the exported wikis are based on MediaWiki, the same approach could have been used simply with the MediaWiki API. Yet, our proposal has many advantages as it relies on SPARQL instead of a particular API and it provides advanced inference capabilities that the original API does not offer. The following query identifies users involved in different wikis, looking for the same usernames.

```
SELECT DISTINCT ?creator1 ?page1 ?page2 ?wiki1 ?wiki2
WHERE {
 ?page1 sioc:has_container ?wiki1 ;
  dc:contributor ?creator1 .
 ?page2 sioc:has_container ?wiki2 ;
  dc:contributor ?creator2 .
 FILTER (str(?creator1)==str(?creator2)) .
 FILTER (str(?wiki1)!=str(?wiki2)) .
}
```

Listing 1.4. Identifying pages created by a single user in different wikis

While this is a very simple query it requires high computation capabilities when ran through a large number of different wikis. Hence, in Fig. 6 we display a screenshot of the results we get after running the same query between the *Semanticweb.org* wiki and the *Protégé* wiki. Instead of also displaying all the other details, such as the related wiki pages and the two wiki containers, we show only the distinct usernames of the found users.

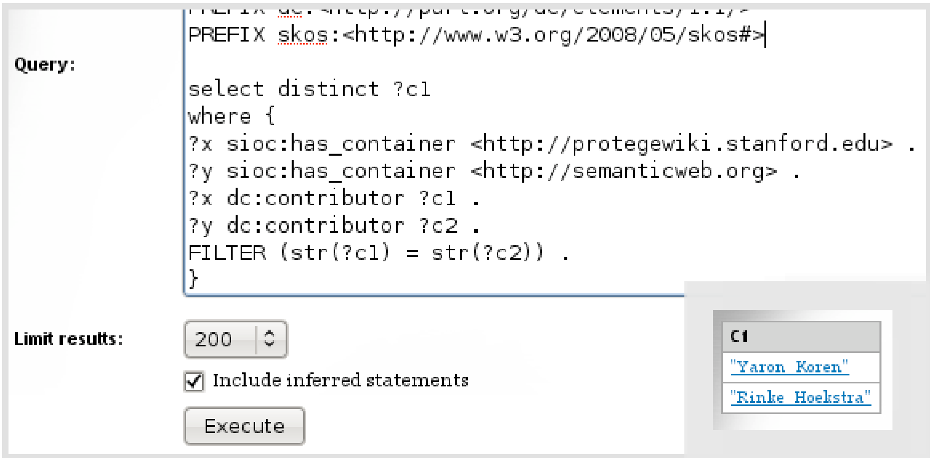


Fig. 6. Query results. Shared users between two wikis.

Yet, as this query relies on a FILTER clause, it will identify common users only if they use the same account name on two different wikis. Moreover, we can imagine that some common account names will be used by different people on different wikis, e.g. JohnSmith. To that extend, we can benefit from the strong ties that exist between FOAF and SIOC and the fact we are modeling a wiki user using the `sioc:User` class. One person can indeed define in his FOAF profile the various wiki accounts he owns, using simple `foaf:holdsAccount` properties. Then, the previous query can be adapted to deal not only with text strings to identify the user, but with their related accounts from the FOAF URI, so that a single query can be used to retrieve all the contributions of a user whatever the wiki used was. Moreover, since the wiki model is based on SIOC, the same query can be used to retrieve wiki pages, blog posts, etc. as follows.

```
SELECT DISTINCT ?content
WHERE {
 <http://example.org/js#me> foaf:holdsAccount ?account .
 ?account rdf:type sioc:User .
 ?content sioc:has_creator ?account .
}
```

Listing 1.5. Cross-sites querying by combining FOAF and SIOC

6 Conclusion

In this paper we presented how the SIOC ontology and lightweight semantics can be used and extended to represent the structure of wikis in an unified way. We first explained our motivations regarding some properties of wikis that we focused on in our modeling process, particularly focusing on a versioning process, and how we can benefit in this case of OWL reasoning capabilities. Then, we described how we designed a webservice to translate any MediaWiki page into SIOC data, following the Linked Data best principles to provide not only isolated RDF, but interlinked data. Finally, we gave some examples regarding how this data could be efficiently used for querying purposes.

While the work done here have been only applied to MediaWiki, further developments may include exporters and plug-in for other platforms to enable better cross-wikis integration. We also consider extending the versioning system defined in SIOC regarding wiki pages to other user-generated content. Moreover the semantic modeling of a wiki article might be improved adding more details about the content of the article itself. One of our goals is also to run cross-queries between our Wikipedia export and DBpedia, for instance to identify which people where the most active on a particular wiki page or topic. Finally, we also believe that this article gives a complete and nice overview regarding how to extend an ontology such as SIOC for particular purposes.

7 Acknowledgements

The work presented in this paper has been funded in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

References

1. Danny Ayers and Max Völkel. Cool URIs for the Semantic Web. W3C Interest Group Note 03 December 2008, World Wide Web Consortium, 2008. `http://www.w3.org/TR/cooluris/`.
2. Tim Berners-Lee. Linked Data. Design issues for the world wide web, World Wide Web Consortium, 2006. `http://www.w3.org/DesignIssues/LinkedData.html`.
3. Chris Bizer, Richard Cyganiak, and Tom Heath. How to Publish Linked Data on the Web. Technical report, 2007. `http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/`.
4. John G. Breslin, Andreas Harth, Uldis Bojārs, and Stefan Decker. Towards Semantically-Interlinked Online Communities. In *Proceedings of the 2nd European Semantic Web Conference (ESWC2005)*, volume 3532 of *Lecture Notes in Computer Science*, pages 500–514. Springer, 2005.
5. Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In *The Semantic Web - ISWC 2002. First International Semantic Web Conference*, volume 2342 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 2002.

6. Michel Buffa, Fabien L. Gandon, Guillaume Ereteo, Peter Sander, and Catherine Faron. SweetWiki: A semantic wiki. *Journal of Web Semantics*, 6(1):84–97, 2008.
7. Andreas Harth, Hannes Gassert, Ina O'Murchu, John G. Breslin, and Stefan Decker. WikiOnt: An Ontology for Describing and Exchanging Wikipedia Articles. In *Proceedings of Wikimania 2005 – The First International Wikimedia Conference*, 2005.
8. Hak Lae Kim, Sung-Kwon Yang, John G. Breslin, and Hong-Gee Kim. Simple algorithms for representing tag frequencies in the scot exporter. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 536–539. IEEE Computer Society, 2007.
9. Markus Krötzsch, Denny Vrandecic, and Max Völkel. Semantic MediaWiki. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, volume 4273 of *Lecture Notes in Computer Science*, pages 935–942. Springer, 2006.
10. Christoph Lange, Uldis Bojārs, Tudor Groza, John G. Breslin, and Siegfried Handschuh. Expressing Argumentative Discussions in Social Media Sites. In *Proceedings of the ISWC2008 Workshop on Social Data on the Web (SDoW2008)*, volume 405 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
11. Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System Reference. W3C Working Draft 29 August 2008, World Wide Web Consortium, 2008. http://www.w3.org/TR/2008/WD-skos-reference-20080829/.
12. Richard Newman, Danny Ayers, and Seth Russell. Tag ontology, December 2005.
13. Alexandre Passant and Philippe Laublet. Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data. In *Proceedings of the WWW2008 Workshop Linked Data on the Web (LDOW2008)*, volume 369 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
14. Alexandre Passant and Philippe Laublet. Towards an Interlinked Semantic Wiki Farm. In *Third Semantic Wiki Workshop – The Wiki Way of Semantics*, volume 360 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
15. Sebastian Schaffert. IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. In *First International Workshop on Semantic Technologies in Collaborative Applications (STICA 06)*, 2006.
16. Max Völkel and Eyal Oren. Towards a Wiki Interchange Format (WIF) - Opening Semantic Wiki Content and Metadata. In *Proceedings of the First Workshop on Semantic Wikis - From Wiki to Semantics (SemWiki-2006)*, volume 206 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006.