Semantic Representation of Provenance in Wikipedia

Fabrizio Orlandi
Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
fabrizio.orlandi@deri.org

Pierre-Antoine Champin
LIRIS, Université de Lyon, CNRS, UMR5205
Université Claude Bernard Lyon 1, F-69622
Villeurbanne, France
pchampin@liris.cnrs.fr

Alexandre Passant
Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
alexandre.passant@deri.org

***Abstract*—Wikis are often considered as being a wide source of information. However, identifying provenance information about their content is crucial, whether it is for computing trust in public wiki pages or to identify experts in corporate wikis. In this paper, we address this issue by providing a lightweight ontology for provenance management in wikis, based on the W7 model. Furthermore, we showcase the use of our model in a framework that computes provenance information in Wikipedia, also using DBpedia to compute provenance and contribution information per category, and not only per page.**

I. Introduction

From public encyclopedia to corporate knowledge management tools, wikis are often considered as being a wide source of information. Yet, since wikis generally offer an open publishing process where everyone can contribute, identifying provenance information in their pages is an important requirement. In particular this information can be used to identify trust values for pages or pages fragments [2] as well as for identifying experts based on the number of contributions [9] and other criteria such as the users' social graphs [10] etc. By providing this information as RDF [6], provenance metadata becomes more transparent and offers new opportunities for the previous use-cases, as well as letting people link to provenance information from other sources, and personalizing trust metrics based on the trust they have to a person regarding a particular topic [5].

This paper describes three of our contributions to address this issue and make provenance information in MediaWiki-powered wikis [1] available on the Semantic Web:

1) a lightweight ontology to represent provenance information in wikis, based on the W7 theory [13] and using SIOC and its extensions;
2) a software architecture to extract and model provenance information about Wikipedia pages and categories, using the aforementioned ontology;
3) a user-interface to make this information openly available on the Web, both to human and software agents and directly within Wikipedia pages.

In the next section, we discuss some related work in the realm of provenance management on the Semantic Web. Then, we give some background information regarding SIOC and various extensions used in our work. In Section IV, we present the W7 theory and the lightweight ontology we have built to represent it in RDFS. We then describe our software architecture and how we compute provenance information in Wikipedia and finally present the user-interface to access this information, before concluding the paper.

II. Related Work

The representation and extraction of provenance information is not a recent research topic. Many studies have been conducted for representing provenance of data [15], but few of them have been focused on integrating provenance information into the Web of data [6]. Providing this information as RDF would make provenance meta-data more transparent and interlinked with other sources, and it would also offer new scenarios on evaluating trust and data quality on the top of it. In this regard a W3C Provenance Incubator Group[2] has been recently established. The mission of the group is to "provide a state-of-the art understanding and develop a roadmap in the area of provenance for Semantic Web technologies, development, and possible standardization". Requirements for provenance on the Web[3], as well as several use cases and technical requirements have been provided by the working group. A comprehensive analysis of approaches and methodologies for publishing and consuming provenance metadata on the Web is exposed in [7].

Another research topic relevant to our work is the evaluation of trust and data quality in wikis. Recent studies proposed several different algorithms for wikis that would automatically calculate users' contributions and evaluate their quantity and quality in order to study the authors' behavior, produce trust measures of the articles and find experts. WikiTrust [2] is a project aimed at measuring the quality of author contributions on Wikipedia. They developed a tool that computes the origin and author of every word on a wiki page, as well as "a measure of text trust that indicates the extent with which text has been revised"[4]. On the same topic other researchers tried

This work is funded by the Science Foundation Ireland under grant number SFI/08/CE/I1380 (Líon 2) and by an IRCSET scholarship.

[1] MediaWiki is the wiki engine that powers Wikipedia – www.mediawiki.org

[2] established in September 2009. http://www.w3.org/2005/Incubator/prov/

[3] http://www.w3.org/2005/Incubator/prov/wiki/User_Requirements

[4] WikiTrust: http://wikitrust.soe.ucsc.edu/

to solve the problem of evaluating articles' quality, not only examining quantitatively the users' history [9], but also using social network analysis techniques [10].

From our perspective, there is a need of publishing provenance information as Linked Data from websites hosting a wide source of information (such as Wikipedia). Yet, most of the work on provenance of data is, either not focused on integrating the information generated on the Web of data, or mainly based on provenance for resource descriptions or already structured data. On the other hand, the interesting work done so far on analyzing trust and quality on wikis does not take into account the importance of making the information extracted available on the Web of data.

III. Background

A. Using SIOC for wiki modelling

The SIOC Ontology — Semantically-Interlinked Online Communities [1] — provides a model for representing online communities and their contributions[5]. It is mainly centered around the concepts of *users*, *items* and *containers*, so it can be used to model content created by a particular user on several platforms, enabling a distributed perspective to the management of User-Generated Content on the Web. In particular, the atomic elements of the Web applications described by SIOC are called `Items`. They are grouped in `Containers`, that can themselves be contained in other `Containers`. Finally, every `Container` belongs to a `Space`. As an example, a `Site` (subclass of `Space`) may contain a number of `Wikis` (subclass of `Container`) and every `Wiki` contains a set of `WikiArticles` (subclass of `Item`) generated by `UserAccounts`. For more details about SIOC, we invite the reader to consult the W3C Member Submission [1] and its online specification[6].

While the SIOC Types module provides several subclasses of `Container` and `Item`, including `Wiki` and `WikiArticle`, some characteristics of wikis required further modelling. Hence, in our previous work [11] we extended the SIOC Ontology to take into account such characteristics (*e.g.* multi-authoring, versioning, etc.). Then, some tools to generate and consume data from wikis using our model have also been developed [12].

B. The SIOC Actions module

While SIOC represents the state of a community at a given time, SIOC-actions [4] can be used to represent their dynamics, *i.e.* how they evolve. Hence, SIOC provides a *document-centric* view of online communities and SIOC-actions focuses on an *action-centric* view. More precisely, the evolution of an online community is represented as a set of *actions*, performed by a user (`sioc:UserAccount`), at some time, and impacting a number of objects (`sioc:Item`). SIOC-actions provides an extensible hierarchy of properties for representing the effect of an *action* on its *items*, such as `creates`, `modifies`, `uses`, etc. Besides the SIOC ontology, SIOC-actions relies on the vocabulary for Linking Open Descriptions of Events (LODE)[7]. The core of the module is the `Action` class (subclass of `event:Event` from the Event Ontology) which is a timestamped event involving an agent (*e.g.* a `UserAccount`) and a number of digital artifacts (*e.g.* `Items`). For more details about SIOC Actions and its implementation see the following Sec. IV.

IV. Representing the W7 model using RDFS/OWL

The W7 model is an ontological model created to describe the semantics of data provenance [13]. It is a conceptual model and to the best of our knowledge a RDFS/OWL representation of this model has not been implemented yet. Hence we will focus on an implementation of this model for the specific context of wikis. As a comparison, in [14] the authors use the example of Wikipedia to illustrate theoretically how their proposed W7 model can capture domain or application specific provenance.

The W7 model is based on the Bunge's Ontology [3], furthermore it is built on the concept of tracking the history of the events affecting the status of things during their life cycle. In this particular case we consider the data life cycle. The Bunge's ontology, developed in 1977, is considered as one of the main sources of constructs to model real systems and information systems. Since the Bunge's work is a theoretical work, there has been some effort from the scientific community to translate his work into machine readable ontologies[8].

The W7 model represents data provenance using seven fundamental elements or interrogative words: *what*, *when*, *where*, *how*, *who*, *which*, and *why*. It has been purposely built with general and extensible principles, hence it is possible to capture provenance semantics for data in different domains. We refer to [13] for a detailed description of the mappings between W7 and Bunge's models, and in Table I we provide a summary of the W7 elements (as in [14]).
Looking at the structure of the W7 model it is clear the motivation why we chose the SIOC Actions module as core of our model. Most of the concepts in the Actions module are the same as in the W7 model. Furthermore wikis are community sites and the Actions module has been implemented to represent dynamic, action-centric views of online communities.

In the following sections we give a detailed description of how we answered each of these seven questions.

A. What

The *What* element represents an event that affected data during its life cycle. It is a change of state and the core of the model. In this regard, there are three main events affecting data: *creation*, *modification* and *deletion*. In the context of wikis, each of them can appear: users can (1) add new sentences (or characters), (2) remove sequences of characters, or (3) modify characters by removing and then adding content

[5] http://sioc-project.org

[6] http://rdfs.org/sioc/spec/

[7] LODE Ontology specification — http://linkedevents.org/ontology/

[8] Evermann J. provides an OWL description of the Bunge's ontology at: http://homepages.mcs.vuw.ac.nz/~jevermann/Bunge/v5/index.html

Provenance element	Construct in Bunge's ontology	Definition
What	Event	An event (i.e. change of state) that happens to data during its life time
How	Action	An action leading to the events. An event may occur, when it is acted upon by another thing, which is often a human or a software agent
When	Time	Time or more accurately the duration of an event
Where	Space	Locations associated with an event
Who	Agent	Agents including persons or organizations involved in an event
Which	Agent	Instruments or software programs used in the event
Why	-	Reasons that explain why an event occurred

TABLE I
DEFINITION OF THE 7 WS BY RAM S. AND LIU J.

in the same position of the article. In addition, in systems like Wikipedia, some other specific events can affect the data on the wiki, for example "quality assessment" or "change in access rights" of an article [14]; however, they can be expressed with the three broader types defined above.

Since (1) wikis commonly provide a versioning mechanism for their content and (2) every action on a wiki article leads to the generation of a new article revision, the core event describing our *What* element is the creation of an article version. In particular we model this creation, and the related modification of the latest version (*i.e.* the permalink), using the SIOC-Actions model as shown in Listing 1.

```
<http://example.com/action?title=Dublin_Core#380106133>
    sioca:creates <http://en.wikipedia.org/w/index.php?
         title=Dublin_Core&oldid=380106133>;
    sioca:modifies <http://en.wikipedia.org/wiki/
         Dublin_Core>;
    a sioca:Action.
```

Listing 1. Representing the "What" element

As we can see from the example above expressed in Turtle syntax, we have a `sioca:Action` identified by the URI ⟨http://example.com/action?title=Dublin_Core# 380106133⟩ that leads to the creation of a revision of the main wiki article about "Dublin Core". The creation of a new revision was originated by a modification (`sioca:modifies`) of the main Wikipedia article ⟨http://en.wikipedia.org/wiki/ Dublin_Core⟩. Details about the type of event are exposed in the next section about the *How* element, where we identify the type of action involved in the event creation.

B. *How*

The *How* element in W7 is an equivalent to the *Action* element from Bunge's ontology, and describes the action leading to an event. In wikis, the possible actions leading to an event (*i.e.* the creation of a new revision) are all the edits applied to a specific article revision. By analyzing the *diff* between two subsequent revisions of a page, we can identify the type of action involved in the creation of the newer revision. In particular we focus on modelling the following types of edits: *Insertion*, *Update* and *Deletion* of both *Sentence*s and *Reference*s. With the term *Sentence* here we refer to every sequence of characters that does not include a reference or a link to another source, and with *Reference* we refer to every action that involves a link or a so-called Wikipedia *reference*. As discussed in [14], another type of edit would be a *Revert*, or an undo of the effects of one or more edits previously happening. However, in Wikipedia, a revert does not restore a previous version of the article, but creates a new version with content similar to the one from an earlier selected version. In this regard, we decided to model a revert as all the other edits, and not as a particular pattern. The distinction between a revert and other types of action can be yet identified, with an acceptable level of precision, by looking at the user comment entered when doing the revert, since most users add a related revert comment [9].

Going further, and to represent provenance data for the action involved in each wiki edit, we modelled the *diffs* appearing between pages. To model the differences calculated between subsequent revisions we created a lightweight Diff ontology, inspired by the Changeset vocabulary[10]. Yet, instead of describing changes to RDF statements, our model aims at describing changes to plain text documents. It provides a main class, the `diff:Diff` class, and six subclasses: `SentenceUpdate`, `SentenceInsertion`, `SentenceDeletion` and `ReferenceUpdate`, `ReferenceInsertion`, `ReferenceDeletion`, based on the previous *How* patterns.

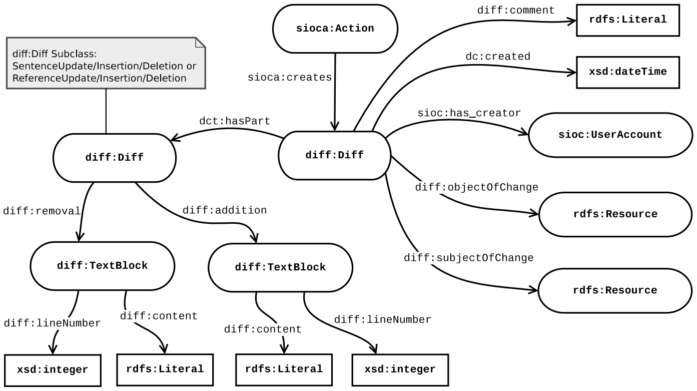


Fig. 1. Modeling differences in plain text documents with the *Diff* vocabulary

The main `Diff` class represents all information about the change between two versions of a wiki page (see Fig. 1). The `Diff`'s properties `subjectOfChange` and `objectOfChange` point respectively to the version changed by this *diff* and to the newly created version. Details about the time and the creator of the change are provided respectively by `dc:created` and `sioc:has_creator`. Moreover, the comment about the change is provided by the `diff:comment` property with range `rdfs:Literal`. In

[9] Note that we could also compare the n-1 and n+1 version of each page to identify if a change is a revert

[10] The Changeset schema: http://purl.org/vocab/changeset/schema#

Figure 1 we also display a `Diff` class linking to another `Diff` class. The latter represents one of the six `Diff` subclasses described earlier in this section. Since a single *diff* between two versions can be composed by several atomic changes (or *"sub-diffs"*), a `Diff` class can then point to several subclasses using the `dc:hasPart` property. Each `Diff` subclass can have maximum one `TextBlock` removed and one added: if it has both, then the type of change is an *Update*, otherwise the type would be an *Insertion* or a *Deletion*.

The `TextBlock` class is part of the Diff ontology and represents a sequence of characters added or removed in a specific position of a plain text document. It exposes the content itself of this sequence of characters (`content`) and a pointer to its position inside the document (`lineNumber`). It is important to precise that usually the document content is organized in sets of lines, as in wiki articles, but this class is generic enough to be reusable with other types of text organization. To note also that each of the six subclasses of the `Diff` class inherit the properties defined for the parent class, but unfortunately this is not displayed in Figure 1 for space reasons.

With the model presented it is possible to address an important requirement for provenance: the reproducibility of a process. Starting from an older revision of a wiki article, just following the *diffs* between the newer revisions and the `TextBlocks` added or removed, it is possible to reconstruct the latest version of the article. This approach goes a step further than just storing the different data versions: it provides details of the entire process involved in the data life cycle.

C. When

The *When* element in W7 is equivalent to the *Time* element from Bunge's ontology, and obviously refers to the time an event occurs, which is recorded in every wiki platform for page edits. As depicted in Figure 1, each `Diff` class is linked to the timestamp of the event using the `dc:created` property. The same timestamp is also linked to each `Diff` subclass using the same property (not shown in Fig. 1 for space reasons). The time of the event is modelled with more detail in the `Action` element as shown in the following Listing 2 [11].

```
<http://example.com/action?title=Dublin_Core#380106133>
   dc:created "2010-08-21T06:36:17Z"^^<http://www.w3.org
        /2001/XMLSchema#dateTime>;
   lode:atTime [
     a time:Instant;
     time:inXSDDateTime "2010-08-21T06:36:17Z"^^<http://
          www.w3.org/2001/XMLSchema#dateTime>.
   ];
   a sioca:Action.
```

Listing 2. Representing the "When" element in Turtle syntax

In this context we consider actions to be instantaneous. As in [4] we track the instant that an action is taking effect on a wiki (*i.e.* when a wiki page is saved). Usually, this creation time is represented using `dc:created`. Another option, provided by the LODE ontology, uses the `lode:atTime` property to link to a class representing a time interval or an instant.

D. Where

The *Where* element represents the online "Space" or the location associated with an event. In wikis, and in particular in Wikipedia, this is one of the most controversial elements of the W7 model. If the location of an article update might be considered as the location of the user when updating the content, then this information on Wikipedia is not completely provided or accurate. Indeed we can extract this information only from the IP address of the anonymous users but not from all the Wikipedia users. To note that is possible to link a `sioc:UserAccount` (*e.g.* ⟨http://en.wikipedia.org/wiki/User:96.245.230.136⟩) to the related IP address using the SIOC `ip_address` property.

E. Who

The *Who* element describes an agent involved in an event, therefore it includes a person or an organization. On a wiki it represents the editor of a page, and it can be either a registered user or an anonymous user. A registered user might also have different roles in the Wikipedia site and, on this basis, different permissions are granted to its account. With this work we are only interested in keeping track of the user account involved in each event, and not also in the role on the wiki. Users are modelled with the `sioc:UserAccount` class and linked to each `sioca:Action`, `sioct:WikiArticle` and `diff:Diff` with the property `sioc:has_creator`. A `sioc:UserAccount` represents a user account, in an online community site, owned by a physical person or a group or an organization (*i.e.* a `foaf:Agent`). Hence a physical person, represented by a `foaf:Person` subclass of `foaf:Agent`, can be linked to several `sioc:UserAccount`.

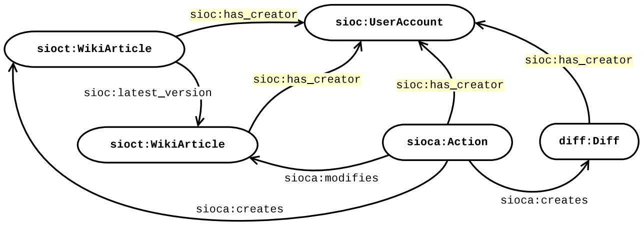


Fig. 2. Modeling the *Who* element with `sioc:UserAccount`

F. Which

The *Which* element represents the programs or the instruments used in the event. In our particular case it is the software used in editing the event, which might be a bot or the wiki software used by the editor. Since there is not a direct and precise way to identify whether the edit has been made by a human or a bot, our model does not make this distinction. A naive method could be to look at the username and check if it contains the "bot" string.

G. Why

The *Why* element represents the reasons behind the event occurrence. On Wikipedia it is defined by the justifications for a change inserted by a user in the "comment" field. This is

[11] For all the namespaces see: http://prefix.cc

not a mandatory field for the user when editing a wiki page but the Wikipedia guidelines recommend to fill-in this text field. We model the comment left by the user with a property `diff:comment` linking the `diff:Diff` class to the related `rdfs:Literal`.

V. Application using provenance data from Wikipedia

A. Collecting the data from the Web

In order to validate and test our modelling solution for provenance on wikis and in particular from the Wikipedia website, we collected data from the English Wikipedia and the DBpedia service. The DBpedia project[12] since it extracts and publishes structured information from the English Wikipedia, is considered as its RDF export. Collecting data not only from Wikipedia but also from the DBpedia source has an important advantage: it directly provides us structured data modelled with popular standard lightweight ontologies in RDF. We use the DBpedia data especially for the categories that hierarchically structure the articles on Wikipedia. We ran our experiment collecting a portion of the Wikipedia articles, and in particular the articles belonging to the whole hierarchy under a given category. By doing this we could limit our dataset only to articles strongly related with each other, and collect a user community with the same interest in common.

A PHP script has been developed to extract all the articles belonging to a category and all its subcategories, and for each article all its revision history. More in detail, this program:

- Executes a SPARQL[13] query over the DBpedia endpoint to get the categories hierarchy;
- Stores the categories hierarchy (modelled with the SKOS[14] vocabulary) in a local triplestore;
- Queries again the DBpedia endpoint to get all the articles belonging to the categories collected;
- For all the articles collected it generates (and stores locally) RDF data using the SIOC-MediaWiki exporter[15];
- Using the `sioc:previous_version` property it exports RDF for all the previous revisions of each article.

It is clear the advantage of using DBpedia in this process since we collected structured data just executing two lightweight SPARQL queries.

A second PHP script has been developed to extract detailed provenance information from the articles collected with the previous step. This script calculates the *diff* function between consecutive versions of the articles, and retrieves more related information from the Wikipedia API. The data retrieved from the API is composed by all the information needed for the creation of the model described in the previous section. Therefore information about the editor, the timestamp, the comment and the ID of the versions are identified. Moreover the algorithm is not only capable of extracting the *diff* function, but also to compute the type of change for each of the differences identified. This allows us to mark each change with one of the *Sentence* or *Reference Insertion/Update/Deletion* subclasses of the `diff:Diff` class. Finally the script generates RDF data with the model described before and inserts it in the local triplestore. In order to test our application we ran the data extraction algorithm starting from the category "*Semantic Web*" on the English Wikipedia, and we generated data for all the 166 wiki articles belonging to this category and its subcategories recursively. As we can see, using Semantic Web technologies, we have the advantage of having a single and standard language to query wiki and provenance data together, while developers that need to query original systems have to learn a new API for each new system we want to query.

B. A Firefox plug-in for provenance from Wikipedia

In order to show the potential of the data collected and the data model created, we built an application to show some interesting statistics extracted from provenance information of the analyzed articles. The application displays a table directly on the top of each Wikipedia article exposing some information about the most active users on the article and their edits. In particular this has been developed using a Greasemonkey[16] script: a Mozilla Firefox extension that allows users to install scripts that make on-the-fly changes to HTML web page content. This script is developed in JavaScript language and is now compatible with other popular Web browsers. The structure of the application is then composed by the following elements: 1) **The triplestore** containing the data collected and exposing a SPARQL endpoint for querying the data; 2) **A PHP script**, used as an interface between the Greasemonkey script and the triplestore; 3) **A Greasemonkey script**, which retrieves the URL of the Wikipedia loaded page, sends the request to the PHP script and then displays the returned HTML data on the Wikipedia page. The PHP script in this application is important because it is responsible for executing the SPARQL queries on the triplestore. Furthermore it retrieves the results and creates the HTML code to embed on the Wikipedia page. A screenshot of the result of the process is displayed in Figure 3.

The tables displayed in Figure 3 appear only on the top of the Wikipedia articles and categories that we analyzed with the method described in Section V-A. A different type of table is showed when the page visited is a category page. In Figure 3 on the top table, we can see the top six users who did the biggest number of edits on the article. For each of these users we then compute: (1) their total number of edits on the page; (2) their percentage of "ownership" on the page (or better, the percentage of their edits compared to all the edits done on the article); (3) their number of lines added on the article; (4) their number of lines removed on the article; (5) their total number of lines added and removed on all the articles belonging to the category *"Semantic Web"*. With the other use-case, when the user visits a Wikipedia category page, we display different

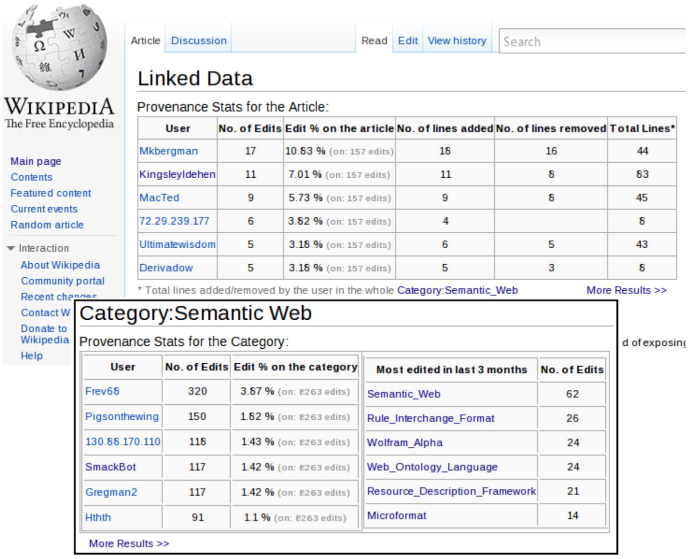
[12] http://dbpedia.org

[13] Query Language for RDF: http://www.w3.org/TR/rdf-sparql-query/

[14] SKOS Reference: http://www.w3.org/TR/skos-reference/

[15] http://ws.sioc-project.org/mediawiki/

[16] http://www.greasespot.net/



Linked Data

Provenance Stats for the Article:

User	No. of Edits	Edit % on the article	No. of lines added	No. of lines removed	Total Lines*
Mkbergman	17	10.83 % (on: 157 edits)	18	16	44
KingsleyIdehen	11	7.01 % (on: 157 edits)	11	8	83
MacTed	9	5.73 % (on: 157 edits)	9	8	45
72.29.239.177	6	3.82 % (on: 157 edits)	4		8
Ultimatewisdom	5	3.18 % (on: 157 edits)	6	5	43
Derivadow	5	3.18 % (on: 157 edits)	5	3	8

* Total lines added/removed by the user in the whole Category:Semantic_Web   More Results >>

Category:Semantic Web

Provenance Stats for the Category:

User	No. of Edits	Edit % on the category
Frev68	320	3.87 % (on: 8263 edits)
Pigsonthewing	150	1.82 % (on: 8263 edits)
130.88.170.110	118	1.43 % (on: 8263 edits)
SmackBot	117	1.42 % (on: 8263 edits)
Gregman2	117	1.42 % (on: 8263 edits)
Hthth	91	1.1 % (on: 8263 edits)

Most edited in last 3 months	No. of Edits
Semantic_Web	62
Rule_Interchange_Format	26
Wolfram_Alpha	24
Web_Ontology_Language	24
Resource_Description_Framework	21
Microformat	14

More Results >>

Fig. 3. A screenshot of the application on the "Linked_Data" page and the table from the Category "Semantic_Web" page

types of information but using the same method. See the table on the bottom in Figure 3. Browsing a wiki category page, the application shows a list of the users with the biggest number of edits on the articles of the whole category (and related subcategories). It also shows the related percentages of their edits compared to the total edits on the category. The second table on the right exposes a list of the most edited articles in the category during the last three months. To note also that at the bottom of each table there is a link pointing to a page where a longer list of results will be displayed.

At the moment the PHP script developed is available at http://vmuss06.deri.ie/WikiProvenance/index.php. Just using this script is possible to have the same information displayed using the Greasemonkey script and also to have the RDF descriptions of the page requested. In order to represent these statistical information in RDF, we use SCOVO, the Statistical Core Vocabulary [8]. It relies on the concept of *Item* and *dimensions* to represent statistical information. In our context, the item is one piece of statistical information (*e.g. user "X" edited 10 lines on page "Y"*), and various items are involved in the description: (1) the type of information that we want to represent (number of edits, percentage, lines added and removed etc.); (2) the page or the category impacted; (3) the user involved. Hence, we created four instances of `scv:Dimension` to represent the first dimension, and relied then simply on the `scv:dimension` property for the other ones. As an example, the following snippet represents that the user *KingsleyIdehen* made 11 edits on the *SIOC* page.

```
ex:123 a scovo:Item ;
   rdf:value 11 ;
   scv:dimension :Edits ;
   scv:dimension <http://wikipedia.org/wiki/SIOC>;
   scv:dimension <http://wikipedia.org/wiki/User:
        KingsleyIdehen>.
```

Listing 3. Representing the number of edits by a user with SCOVO

VI. Conclusion and Future Work

The goal of this paper was to provide a solution for representing and managing provenance of data from Wikipedia (and other wikis) using Semantic Web technologies. To solve this problem we provided: a specific lightweight ontology for provenance in wikis, based on the W7 model; a framework for the extraction of provenance data from Wikipedia; an application for accessing the generated data in a meaningful way and exposing it to the Web of data. We showed that the W7 model is a good choice for modelling provenance information in general and in wikis but, because of its high abstraction level, it has to be refined using for instance other specific lightweight ontologies. In our case this has been done using SIOC and the Actions module. Future developments will include a refinement of the proposed model and a subsequent alignment with other general-purpose ontologies for representing provenance as Linked Data (*e.g.* the *Open Provenance Model*). We also plan to improve and extend the potentialities of our application offering more features, and providing a wider range of data with an architecture that automatically updates the data as soon as it changes on Wikipedia.

References

[1] SIOC Core Ontology Specification. W3C Member Submission 12 June 2007, World Wide Web Consortium, 2007. http://www.w3.org/Submission/sioc-spec/.

[2] B.T. Adler, L. de Alfaro, I. Pye, and Vishwanath Raman. Measuring author contributions to the wikipedia. In *Proceedings of WikiSym '08*. ACM, 2008.

[3] Mario Bunge. *Treatise on Basic Philosophy: Ontology I: The Furniture of the World*. Riedel, Boston, 1977.

[4] P.A. Champin and A. Passant. SIOC in Action - Representing the Dynamics of Online Communities. In *Proceedings of the 6th International Conference on Semantic Systems (I-SEMANTICS 2010)*. ACM, 2010.

[5] J. Golbeck, B. Parsia, and J. Hendler. Trust networks on the semantic web. *Cooperative Information Agents VII*, pages 238–249, 2003.

[6] Olaf Hartig. Provenance information in the web of data. In *2nd Workshop on Linked Data on the Web (LDOW 2009) at WWW*, 2009.

[7] Olaf Hartig and Jun Zhao. Publishing and Consuming Provenance Metadata on the Web of Linked Data. In *Proceedings of 3rd Int. Provenance and Annotation Workshop*, 2010.

[8] M Hausenblas, W Halb, Y Raimond, L Feigenbaum, and D Ayers. SCOVO: Using statistics on the Web of data. In *Semantic Web in Use Track of the 6th European Semantic Web Conference (ESWC2009)*, 2009.

[9] B Hoisl, W Aigner, and S Miksch. Social Rewarding in Wiki Systems–Motivating the Community. In *Proceedings of the 2nd international conference on Online communities and social computing*, pages 362–371. Springer-Verlag, 2007.

[10] NT Korfiatis, M Poulos, and G Bokos. Evaluating authoritative sources using social networks: an insight from Wikipedia. *Online Information Review*, 2006.

[11] Fabrizio Orlandi and Alexandre Passant. Enabling cross-wikis integration by extending the SIOC ontology. In *4th Semantic Wiki Workshop (SemWiki 2009)*. CEUR-WS, 2009.

[12] Fabrizio Orlandi and Alexandre Passant. Semantic Search on Heterogeneous Wiki Systems. In *International Symposium on Wikis (WikiSym2010)*. ACM, 2010.

[13] Sudha Ram and Jun Liu. *Understanding the semantics of data provenance to support active conceptual modeling*, pages 17–29. Springer Berlin / Heidelberg, lncs edition, 2007.

[14] Sudha Ram and Jun Liu. A New Perspective on Semantics of Data Provenance. In *First International Workshop on the role of Semantic Web in Provenance Management (SWPM 2009)*, 2009.

[15] Y.L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance techniques. *Computer Science Department, Indiana University, Bloomington IN*, 47405, 2005.