**Università degli Studi di Modena e Reggio Emilia**

**Facoltà di Ingegneria - Sede di Modena**

**Corso di Laurea Specialistica in Ingegneria Informatica**

# Using and extending the SIOC ontology for a fine-grained wiki modeling

## Estensione ed uso dell'ontologia SIOC per una modellazione dettagliata dei wiki

**Relatore:**
Chiar.mo
Prof.Sonia Bergamaschi

**Candidato:**
Fabrizio Orlandi

**Correlatore:**
Dott.Alexandre Passant

Anno Accademico 2007/2008

*A due persone speciali*

*che avrebbero tanto voluto essere qui*

# *Ringraziamenti*

*Desidero ringraziare la Prof. Sonia Bergamaschi, per l'aiuto fornitomi e per l'entusiasmante opportunità offerta di un'esperienza di ricerca al DERI in Irlanda.*
*I would like to thank Alexandre Passant who gave me a lot of help and feedback with his experience and knowledge during the months spent in DERI.*
*Non potrei mai dimenticare di ringraziare tutti gli amici e compagni di studio con cui ho condiviso gioie e dolori in questi anni di Università.*
*Ma soprattutto un enorme ringraziamento va alla mia famiglia per il suo costante sostegno durante tutto il percorso di studi.*

*KEYWORDS*
*Semantic Web*
*Web2.0*
*SIOC ontology*
*Wiki*
*Social Web*

# Contents

# List of Figures

# List of Tables

# Introduzione

Il presente lavoro di tesi è stato condotto sotto la supervisione della Prof.Sonia Bergamaschi presso l'Università di Modena e Reggio Emilia e del Dott.Alexandre Passant del Digital Enterprise Research Institute (DERI)[1] situato a Galway in Irlanda. Presso questo istituto sono stati effettuati quattro mesi di tirocinio, e parte dei risultati sono stati presentati con un articolo, effettuato in collaborazione con Alexandre Passant, sottomesso al quarto Workshop sui Semantic Wiki [SemWiki-2009] che si tiene presso il 6° European Semantic Web Conference (ESWC) a Creta, Grecia (1 giugno 2009). L'articolo è ancora in attesa di accettazione da parte della commissione della conferenza.

In questa tesi verrà proposta un'estensione dell'ontologia SIOC che permetta la capacità di interconnettere e modellare semanticamente i wiki. Inoltre verranno presentate particolari applicazioni software che consentono di esportare un qualsiasi wiki, sviluppato utilizzando il framework MediaWiki[2], in dati in formato SIOC. La modellazione semantica dei wiki e la loro esportazione è stata effettuata seguendo i principi del Linked Data, ovvero con la possibilità di utilizzo dei dati da parte di qualsiasi applicazione nell'ambito del Semantic Web. Vengono inoltre proposti esempi e casi d'uso su come questi dati potrebbero essere utilizzati in modo efficiente per operazioni di interrogazione complesse.

[1] `http://www.deri.org/`

[2] `http://www.mediawiki.org/`

L'ontologia SIOC, Semantically-Interlinked Online Communities[12], è ora considerata come uno degli elementi costitutivi del *"Social Semantic Web"*. Pubblicata come W3C Member Submission, attualmente vanta un utilizzo su più di 50 applicazioni[3], venendo utilizzata sia come ontologia comune per esporre i dati in RDF, così come col progetto FOAF[4], sia sfruttando direttamente i dati SIOC già esistenti sul Web, come nel caso di Yahoo! SearchMonkey[5]. L'implementazione del progetto SIOC va oltre i principali servizi Web 2.0, spazia infatti da ambiti di integrazione delle informazioni Enterprise 2.0[6], a rappresentazioni della conoscenza in sistemi di Health Care e Life Sciences[7].

Tuttavia, poco lavoro è stato effettuato finora per quanto riguarda i wiki, i wiki semantici e l'ontologia SIOC. Mentre in SIOC è già previsto un modulo chiamato *Types*[8] contenente le classi `Wiki` e `WikiArticle` che possono essere usate per rappresentare i tipi di oggetti che costituiscono i wiki. Alcune caratteristiche particolari come ad esempio le versioni di una pagina o i link entranti non vengono prese in considerazione, né nell'ontologia SIOC *Core* principale, né nei suoi moduli.

Eppure fornendo informazioni sui wiki utilizzando SIOC porterebbe a molti vantaggi, in termini di integrazione con gli attuali esistenti dati SIOC, costantemente creati in modo dinamico dalla comunità e dalle applicazioni Web, nonché per l'interconnessione con altri dati RDF a scopo di interrogazione avanzata. Ad esempio sarà possibile eseguire la stessa query di ricerca per trovare corrispondenze su più oggetti creati su un wiki o su un blog o su un forum. Pertanto col presente lavoro di tesi si è mirato ad estendere l'ontologia SIOC proprio per

[3] `http://sioc-project.org/applications/`

[4] `http://foaf-project.org/`

[5] `http://developer.yahoo.com/searchmonkey/`

[6] `http://www.w3.org/2001/sw/sweo/public/UseCases/EDF/`

[7] `http://esw.w3.org/topic/HCLSIG/SWANSIOC`

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

questo scopo, così come a fornire una esportatore SIOC per i wiki MediaWiki, portando così potenzialmente alla creazione di milioni di dati in formato RDF basati su SIOC e provenienti da svariati e popolari servizi Web, come ad esempio Wikipedia.

Il presente lavoro di tesi verrà organizzato nel seguente modo. Saranno innanzitutto individuate due parti principali: la prima parte di studio e ricerca dello stato dell'arte, e la seconda parte di sperimentazione e illustrazione del lavoro implementativo effettuato.

Nel primo capitolo verrà effettuata una panoramica generale dell'ambito in cui si colloca questa tesi, illustrando i concetti di Web Semantico, Web 2.0 e comunità online. Così come verrà fornita una descrizione dettagliata del progetto SIOC, l'ontologia e il suo utilizzo, e dei wiki come strumento fondamentale di creazione collaborativa di conoscenza.

A partire dal secondo capitolo invece, i wiki verranno approfonditi ulteriormente, andando a focalizzare l'attenzione sulle caratteristiche fondamentali di questi servizi Web. In particolare su quelle funzionalità ritenute più importanti da considerare in un approccio di modellazione semantica strutturale.

Nel terzo capitolo verrà trattata un'esplorazione dello stato dell'arte dei wiki semantici. In particolare sul modo in cui alcuni disponibili wiki semantici attualmente modellano una tipica struttura di wiki. Nonostante l'obiettivo prefisso è quello di modellare le caratteristiche dei wiki estendendo SIOC, diversi vocabolari sono già stati proposti per raggiungere questo obiettivo, quindi alcuni di loro verranno studiati e presentati analizzandone i pregi ed i difetti. Da sottolineare inoltre che questo studio non mira a descrivere lo stato dell'arte dei wiki semantici, quanto piuttosto si focalizza sull'ontologia che ne definisce tutte le caratteristiche strutturali fondamentali.

Il quarto capitolo, il primo della seconda parte, descrive come l'ontologia

SIOC è stata estesa in modo da tener conto degli aspetti particolari dei wiki evidenziati nei capitoli precedenti. L'estensione è stata progettata in modo da tenere conto in particolare degli obiettivi di interconnessione tra piattaforme di wiki differenti, e da consentire il massimo riutilizzo del modello da parte delle più disparate applicazioni in ambito Semantic Web.

Le applicazioni software implementate verranno esaminate nel quinto capitolo. Saranno infatti descritti gli esportatori creati che si occupano di tradurre ogni pagina wiki, di tipo MediaWiki, in documenti RDF che utilizzano il nuovo modello esteso dell'ontologia SIOC. Due applicazioni sono state implementate, una è un webservice in linguaggio PHP, e l'altra è invece di tipo standalone scritta in linguaggio Java. Solamente il webservice verrà però descritto nel dettaglio in quanto l'altra applicazione implementa stessi principi e metodi seguiti dalla prima. L'analisi si incentrerà in particolare sul modo in cui questo servizio produce dati RDF compatibili con i principi di "Linked Data", e su come si può fare riferimento ad altre iniziative importanti come quella di DBpedia[9].

Nel sesto ed ultimo capitolo, al fine di valutare l'estensione dell'ontologia SIOC e il modello proposto, dopo la creazione del webservice esportatore "SIOC-MediaWiki", verranno dettagliate le prove di interrogazione avanzate effettuate sui dati RDF prodotti. Sono stati infatti raccolti dati da quattro wiki MediaWiki differenti attraverso l'utilizzo di uno speciale crawler e del webservice sviluppato. L'obiettivo è quello di raccogliere grandi quantità di dati da collocare all'interno di un database "triple store" RDF, in modo tale da poterli interrogare utilizzando il linguaggio SPARQL[10] e provare alcune utili e avanzate query. Quest'ultime verranno mostrate con alcuni esempi di interrogazioni effettuate su un unico wiki, e tra wiki differenti.

Infine, in conclusione della tesi, verrà presentata una panoramica dei possibili

[9] http://dbpedia.org/

[10] http://www.w3.org/TR/rdf-sparql-query/

lavori di sviluppo futuro riguardo a questo particolare dominio.

# Introduction

The research subsumed in this thesis has been conducted under the supervision of Prof.Sonia Bergamaschi from the University of Modena and Reggio Emilia, Italy and of Dott.Alexandre Passant from the Digital Enterprise Research Institute (DERI)[11] of Galway, Ireland. A four months internship to DERI Galway has been done during this thesis, and part of the results have been submitted as a full paper, together with Alexandre Passant, for the Fourth Workshop on Semantic Wikis [SemWiki2009] at the 6th Annual European Semantic Web Conference (ESWC) in Crete, Greece (June 1, 2009). The acceptance of the paper for the conference is still pending.

In this thesis I propose an extension to the SIOC Ontology defined to provide the ability to model and interlink wikis. Furthermore applications to export any MediaWiki instance into SIOC data, following the Linked Data principles, have been developed together with examples regarding how this data could be efficiently used for querying purposes.

The SIOC Ontology – Semantically-Interlinked Online Communities [12] – is now considered as one of the building blocks of the *"Social Semantic Web"*. Published as a W3C Member Submission, currently more than 50 applications are using SIOC[12], either as a common vocabulary to expose their data in RDF,

[11] http://www.deri.org/

[12] http://sioc-project.org/applications

alongside with FOAF for instance, as well as using existing SIOC data, as for instance Yahoo! SearchMonkey[13]. Moreover, the use of SIOC goes further than mainstream Web 2.0 services, from Enterprise 2.0 information integration[14] to Health Care and Life Sciences discourse representation[15].

However, only a few work have been done so far regarding wikis, semantic wikis and the SIOC ontology. While the SIOC Types[16] 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 SIOC data, as well as interlinking with other RDF data for advanced querying purposes. For instance, one will be able to run the same query to find latest created items on a MediaWiki instance or on a WordPress weblog. Hence, I 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 thesis is organized as follows. First, I will go through an overview of wiki features that are important to consider in such a modeling approach and explain how I took them in consideration in regards of the SIOC ontology and how I extended it based on this analysis. Then, since some wikis already expose their data in a machine-readable form thanks to Semantic Web technologies, I will focus on a state of the art of existing models that achieve the same goal. Then it will be detailed how I built a webservice that translates any MediaWiki wiki page to RDF using the newly-extended SIOC ontology. I will particularly focus on how this

[13] `http://developer.yahoo.com/searchmonkey/`

[14] `http://www.w3.org/2001/sw/sweo/public/UseCases/EDF/`

[15] `http://esw.w3.org/topic/HCLSIG/SWANSIOC`

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

service produces RDF data compliant with the Linked Data principles and how it relates to initiatives such as DBpedia. Subsequently I will show some relevant query examples, from advanced queries in a single wiki, to cross-querying capabilities. Finally, the conclusion of the thesis will be with an overview of possible future works on the domain.

# Part I

# State of the Art

# Chapter 1

# Background

## 1.1 Web 2.0 and Social Media

### 1.1.1 Web 2.0

Although Web 2.0 is currently a very popular term, it is difficult to give its precise definition. Briefly, Web 2.0 is the Web where people meet, collaborate and share anything that is popular by using social software applications. In this regard it could be appropriate to see what web users say about Web 2.0, and how they define it. Here is the definition taken from the "Web 2.0" article on Wikipedia[1]: "The term "Web 2.0" refers to a perceived second generation of web development and design, that aims to facilitate communication, secure information sharing, interoperability, and collaboration on the World Wide Web. Web 2.0 concepts have led to the development and evolution of web-based communities, hosted services, and applications; such as social-networking sites, video-sharing sites, wikis, blogs, and folksonomies."

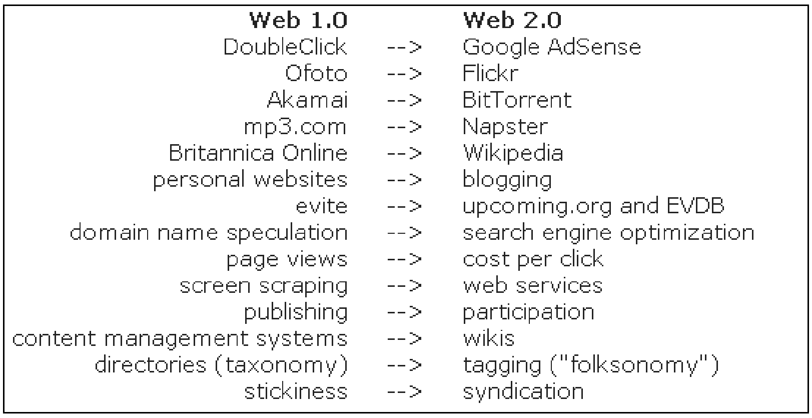
This definition is exactly an example of how the web community can cooperate

[1] `http://en.wikipedia.org/wiki/Web_2.0` - March 2009

using the web instruments provided by this evolution of the Web.

The term refers to second generation of Internet-based services: blogs, wikis, communication tools and platforms like del.icio.us[2], Flickr[3], Skype[4], Wikipedia[5], last.fm[6], Technorati[7]. Web 2.0 applications derive from new techniques such as rich internet applications (RIA), Asynchronous JavaScript and XML (AJAX), semantically valid Extensible HyperText Markup Language (XHTML), Cascading Style Sheets (CSS), Syndication and aggregation of data in RSS or Atom, clean and meaningful URLs. A user of Web 2.0 must feel as if he/she used traditional desktop applications to share anything with the community.

In accordance with Tim O'Reilly [23], the meaning of Web 2.0 can be presented by contrasting the traditional Web with new Web 2.0, as in Figure 1.1[8].



| **Web 1.0** | | **Web 2.0** |
|---|---|---|
| DoubleClick | --> | Google AdSense |
| Ofoto | --> | Flickr |
| Akamai | --> | BitTorrent |
| mp3.com | --> | Napster |
| Britannica Online | --> | Wikipedia |
| personal websites | --> | blogging |
| evite | --> | upcoming.org and EVDB |
| domain name speculation | --> | search engine optimization |
| page views | --> | cost per click |
| screen scraping | --> | web services |
| publishing | --> | participation |
| content management systems | --> | wikis |
| directories (taxonomy) | --> | tagging ("folksonomy") |
| stickiness | --> | syndication |

Figure 1.1: From Web 1.0 to Web 2.0

[2] http://del.icio.us/

[3] http://www.flickr.com/

[4] http://www.skype.com/

[5] http://www.wikipedia.org/

[6] http://www.last.fm/

[7] http://www.technorati.com/

[8] From "What Is Web 2.0, Design Patterns and Business Models for the Next Generation of Software" by Tim O'Reilly (09/30/2005) http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

### 1.1.2 Online Communities

Online Communities are group of people that primarily interact via several different types of communication media (e.g. mobile phones, Internet, email, social network service on the Web, newsletter, etc.). The main reason why a user belongs to social network is the desire to share and meet others with a similar domain of interests. Collaboration is a good way of reaching information and knowledge.

A social network service focuses on building online communities of people who share interests and/or activities, or who are interested in exploring the interests and activities of others. Most social network services are web based and provide a variety of ways for users to interact, such as e-mail and instant messaging services. Social networking has encouraged new ways to communicate and share information. Social networking websites are being used regularly by millions of people, and it now seems that social networking will be an enduring part of everyday life.[9]

Communication can be divided to three modes, which is classified on the basis of the techniques used: one-to-one (e.g. emails, instant messaging, etc.); one-to-many (e.g. web pages, blogs, etc.); many-to-many (e.g. forum, wikis, etc.).
Networks have diverse sizes. In a small, tight one, there are few people who form a kind of a private area. However, there can also be a lot of participants with loose connections (weak ties). From the collaboration point of view, the latter mode is more valuable as it is more probable to introduce new ideas. Hence, it is better to have connections with other networks than with only one. However, unlimited access to information exchange can involve some risk; there is a possibility that a social network is flooded with unneeded information. To avoid that, or at least to limit the possibility of reaching poor data, rating and annotating shared resources

[9] http://en.wikipedia.org/w/index.php?title=Social_network_service&oldid=278365464

were introduced.

### 1.1.3 Wikis

The first wiki system was developed by Ward Cunningham in 1994, under the name WikiWikiWeb [21]. In Hawaiian the word "wikiwiki" means "quick". The original definition of a wiki describes it as "The simplest online database that could possibly work" [27]. Wikis are web sites that can be collaboratively edited by anyone. Pages are written in a simple syntax so that even novice users can easily edit pages. The syntax consists of simple tags for creating links to other Wikipages and textual markups such as lists and headings. The user interface of most Wikis consists of two modes: in reading mode, the user is presented normal webpages that can contain pictures, links, textual markup, etc. In editing mode, the user is presented an editing box displaying the Wiki syntax of the page (containing the text including the markup tags). During editing, the user can request a preview of the page, which is then rendered by the server and returned to the user.

Many Wiki engines exist for anyone who wants to setup a Wiki, most of these engines are open-source. Many sites run a Wiki as a community venue, enabling users to discuss and write on topics. For example, many open-source projects have a documentation Wiki, where users can collaboratively add documentation about the project. The burden of editing is thus shared over the whole community, while still allowing anybody to quickly find relevant documentation (which is harder in e.g. a forum or bulletin board). Popular Wikis such as Wikipedia[10] can grow very fast, since interested visitors can edit and create pages at will.

Wikis are inherently server-based which has the advantage of user platform independence (users only need a browser) but the disadvantage of requiring a round-trip for all changes. For example, after each edit the page is committed

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

back to the server, and the newly rendered page is returned to the user. Although the usability of most Wikis is good compared to other web applications (being simple and fast), the need for server round-trips is a disadvantage compared to desktop applications.

A problem with large Wikis is finding relevant information. Since almost all the information in current Wikis is textual, the only possibility to locate relevant information is a full-text search on some keywords. The only semantics of pages lies in the links between pages. Indeed almost all Wiki engines generate navigational benefits from these links: one can see all pages linking to the current one, and go to these related pages. But this navigation through related pages is quite limited, and does not address the need for more intelligent information retrieval.
A description of fundamental wiki features will be provided in Chapter 2.

## 1.2 The Semantic Web

The Semantic Web is the ongoing evolution of the Web into a more powerful and more reusable infrastructure for information sharing and knowledge management. The current Web is a publishing platform and indeed allows us to connect with arbitrary information sources across all physical and technical boundaries. But the Web is merely a publishing infrastructure of documents and links; very little consideration is given to the content or meaning of the documents or to the meaning of the links. As a consequence, the Web serves as an excellent giant document repository and, as a communication platform, enables the provision of online services, but knowledge reuse is limited because no uniform standard is available to express the meaning or intended usage of pieces of online information.

### 1.2.1 Overview and goals of the Semantic Web

The Semantic Web [7] is a web of information that is more understandable and more usable by machines than the current Web. It can be regarded as an extension of the existing Web, whose information is mostly human-readable. Although the current Web also has some machine-usable structure such as head and body of documents, levels of heading elements, classes of `<div>` elements[11], this structure has coarse granularity and little agreed-upon meaning. The Semantic Web allows for finer granularity of machine-readable information and offers mechanisms to reuse meaning.

The Semantic Web can also be considered similar to a large online database, containing structured information that can be queried. But in contrast to traditional databases, the information can be heterogeneous: it does not conform to one single schema; the information can be contradicting: not all facts need to be consistent; the information can be incomplete: not all facts need to be known; and resources have global identifiers allowing interlinked statements to form a global "Web of Semantic Data".

#### Resource Description Framework, RDF

The fundamental data-model of the Semantic Web is the Resource Description Framework[12] (RDF) [17]. RDF is a language for asserting statements about arbitrary identifiable resources. The use of global identifiers (URIs)[4] allows statements from different sources to interlink, ultimately forming a hypergraph of statements. RDF is a formal language in the sense that a syntax, grammar, and model-theoretic semantics are defined [2]. It is a W3C standard and it was designed to be read and processed by machines, not to be displayed to people.

[11] <div> is an HTML tag which expresses a block-level logical division.

[12] `http://www.w3.org/RDF/`

On account of RDF, programs or automated scripts (crawlers) can efficiently search, discover, collect and process information from the Web. RDF is based on statement concepts. In a statement, there is a subject, a predicate and an object; altogether they are called a triple (a statement). A collection of RDF statements produces a directed graph in which arrows point from subjects to objects and texts on arrows are predicates. If we consider for example the folowing sentence: "Modena is a city", it can be represented by an RDF statement that has the following structure: there is a subject (resource) *Modena*, a predicate (property) *is a*, an object (value) *city*. Supposing all three parts are attributed with an URI, all with the namespace `http://example.com`, the above statement can be illustrated by a graph showed on Fig 1.2.

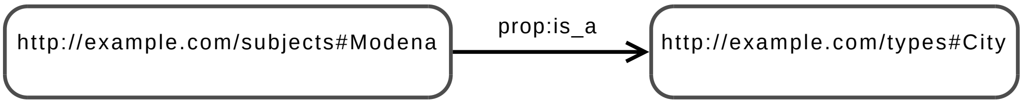

http://example.com/subjects#Modena
prop:is_a
http://example.com/types#City

Figure 1.2: RDF statement.

Besides the graph, RDF/XML representation can be used to show triples and relationships between them. To note that RDF/XML is only one of the multiple possible serializations for RDF data. As in the following Listing 1.1.

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:prop="http://example.com/property#">
   <rdf:Description rdf:about="http://example.com/subjects#Modena">
     <prop:is_a rdf:resource="http://example.com/types#City">
   </rdf:Description>
</rdf:RDF>
```

Listing 1.1: RDF/XML representation

The semantics described with RDF provide a formal meaning to a set of statements through an interpretation function into the domain of discourse. But this interpretation function is relatively straightforward and explicit: the semantics of RDF prescribe relatively few inferences to be made from given statements – there is only little implicit information in statements. RDF can thus be seen as a language for statements without specifying the meaning of these statements.

Such computer-usable meaning can be achieved by defining a vocabulary (a set of terms) for RDF and by specifying what should be done when such a term is encountered. Currently, two such vocabularies have been agreed upon and standardized: RDF Schema (RDFS) and Web Ontology Language (OWL).

RDF Schema (RDFS)[13], allows the expression of a schema-level information such as class membership, sub-class hierarchies, class attributes (properties), and sub-property hierarchies [3]. RDFS allows simple schema information, but its expressiveness is limited.

The Web Ontology Language (OWL)[14] therefore extends RDFS (although the two are formally not completely layered) and provides terms with additional expressiveness and meaning[1]. OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes. OWL is composed of three sub-languages: OWL Lite, OWL DL and OWL Full, as it will be described with more detail later in the next section "Ontologies", and we will refer mainly on the first one.

### SPARQL

SPARQL[15] is a query language for RDF and it can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed

[13] http://www.w3.org/TR/rdf-schema/

[14] http://www.w3.org/2004/OWL/

[15] http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/

as RDF via middleware. Its name is a recursive acronym that stands for SPARQL Protocol and RDF Query Language. It is standardized by the SPARQL Working Group[16] (was RDF Data Access Working Group) of the W3C (World Wide Web Consortium) and currently it is an official W3C Recommendation. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. It also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs.

An example, the following simple SPARQL query returns "people who were born in Berlin before 1900, ordered by name".

```
PREFIX abc: <http://example.com/exampleOntology#>

SELECT ?name ?birth WHERE {
   ?person abc:birthPlace <http://example.com/resource/Berlin> .
   ?person abc:birth ?birth .
    FILTER (?birth < "1900-01-01"^^xsd:date) .
   ?person abc:name ?name .
}
ORDER BY ?name
```

Listing 1.2: SPARQL query example

In the example variables are indicated by a "?", and bindings for `?name` and `?birth` will be returned ordered by name. The SPARQL `FILTER` clause restrict solutions to those for which the filter expression evaluates to TRUE, in this case if the `?birth` variable has a date value minor than "1900-01-01". The SPARQL query processor will search for sets of triples that match these four triple patterns, binding the variables in the query to the corresponding parts of each triple. To make queries concise, SPARQL allows the definition of prefixes and base URIs in a fashion similar to the Turtle[17] RDF syntax. In the query example above, the

[16] http://www.w3.org/2009/sparql/wiki/Main_Page
[17] http://www.dajobe.org/2004/01/turtle/

prefix "abc" stands for "`http://example.com/exampleOntology#`".

### 1.2.2 Ontologies

The term ontology has its origin in philosophy, and has been applied in many different ways. In computer science and information science, an ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. A widely cited web page and paper [15], associated with the effort to define this term, is credited with a deliberate definition of ontology as a technical term in computer science. The paper defines ontology as an "explicit specification of a conceptualization," which is, "the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold among them" [Tom Gruber et al., 1995]. While the terms specification and conceptualization have caused much debate, the essential points of this definition of ontology are:

- An ontology defines (specifies) the concepts, relationships, and other distinctions that are relevant for modeling a domain.

- The specification takes the form of the definitions of representational vocabulary (classes, relations, and so forth), which provide meanings for the vocabulary and formal constraints on its coherent use.

In the technology stack of the Semantic Web standards, ontologies are called out as an explicit layer. There are now standard languages and a variety of commercial and open source tools for creating and working with ontologies. Although RDF and RDF Schema are helpful in expressing simple statements, they lack when used in more complex cases. That is why Web Ontology Language (OWL) was developed.

OWL is a markup language for publishing and sharing data using ontologies on the Internet. It consists of three sub-languages: OWL Lite, OWL DL and OWL Full. Each sub-language encapsulates the former ones. It is mainly the level of restrictions, which distinguishes them. An OWL ontology contains a description of classes, properties and their instances. Also, it allows us to define cardinality constraints on properties, specifying transitivity and uniqueness. In general, an ontology represents a domain and objects within that domain. It is a form of knowledge representation of such domain.

The following points summarize a list of some of the most popular ontologies for specific domains of interests.

- people and social networks: Friend Of A Friend (FOAF)[18]
- online discussions: Semantically-Interlinked Online Communities (SIOC)[19]
- documents: Dublin Core (DC)[20]
- thesauri, taxonomies and subject-heading systems: Simple Knowledge Organization System (SKOS)[21]

## 1.3 Combining Web 2.0 and the Semantic Web

It is quite common to view the Web2.0 and the Semantic Web as mutually exclusive and competing paths to the Web of the future. The two approaches are in fact complementary, and both face challenges the other can solve, such as how to integrate Web2.0 data on a Web scale, and how to enable users to create semantically rich annotations. Web2.0 provides several applications producing and reusing user-generated content, supporting social and collaborative interaction on

[18] http://www.foaf-project.org/

[19] http://www.sioc-project.org

[20] http://dublincore.org/

[21] http://www.w3.org/2004/02/skos/

the Web, and providing engaging user interactions. The Semantic Web vision relies on data published in machine-readable formats, given formal semantics through the use of shared ontologies, and interlinked on a Web scale. By making Web data more open to processing by machines, the Semantic Web fundamentally aims to bring tangible benefits to users.

Starting from Web2.0 applications producing vast amounts of user-generated content, such as wiki entries, tagged photos, and links joining people in social network, the Semantic Web offers a platform on which publishing data in RDF lowers the barriers to its reuse by others.

In the following two paragraphs Semantic Wikis first, then the SIOC project, will be briefly described. They are two different and important examples of the meeting points between Web2.0 and the Semantic Web.

### 1.3.1 Semantic Wiki

A semantic wiki is a wiki with an "underlying model of the knowledge described in its pages"[22]. A semantic wiki extends a wiki by "semantic technologies" like RDF and OWL. The main idea is to make the inherent structure of a wiki – given by the strong linking between pages – accessible to machines (agents, services) beyond mere navigation. This is generally done by annotating existing navigational links with symbols that describe their meaning. Such annotations are useful for many purposes, e.g. enhanced presentation by displaying contextual information, enhanced navigation by giving easy access to relevant related information, and enhanced "semantic" search that respects the context in addition to the content. Semantic wikis exist in many different flavors (e.g. Semantic MediaWiki[23],

[22] as stated in the "Semantic wiki" article from Wikipedia: http://en.wikipedia.org/w/index.php?title=Semantic_wiki&oldid=274954057 (March 2009)

[23] http://semantic-mediawiki.org/

IkeWiki[24] , SweetWiki[25]). While for some of them the page content is still in the foreground and the annotations just optional "added value", others require annotations as mandatory.

Semantic wikis may even improve daily life for humans, as stated in the documentation of Semantic MediaWiki [18], some useful use cases:

- Lists (e.g. the twenty largest metropolitan areas in Spain) could be generated by querying the knowledge base instead of maintaining them by hand.
- Searches will become more powerful.
- Less categories will be needed; as with lists, many categories can be inferred from the knowledge base.
- Different language versions (e.g. of Wikipedia) will become more consistent by sharing common data.
- External semantic-web-aware applications can re-use the knowledge from the wiki.

Precursors of semantic wiki engines provided means for storing page-related metadata and tags and searching them. Most of today's semantic wikis utilize formal languages known from the semantic web, such as RDF or OWL ontologies. Where a regular wiki enables users to describe resources in natural language, a semantic wiki enables users to additionally describe resources in a formal language. Their capabilities range from annotating pages and links with metadata to representing ontological concepts. Actually, a sophisticated semantic wiki should offer many levels of knowledge representation.
Several attempts to classify the different approaches of semantic wikis have been

[24] http://ikewiki.salzburgresearch.at

[25] http://sweetwiki.inria.fr/sweetwiki

made, the latest can be found on the Semanticweb.org website[26], but they have many differences especially as regards the data model, the method of annotation, navigation, and the import/export to and from the Semantic Web.

### 1.3.2 SIOC

SIOC (Semantically-Interlinked Online Communities)[27] aims to enable the integration of online community information [8]. It allows the description of information contained within online community sites (blogs, forums, wikis, etc.) and by doing so make it possible to connect these sites together, forming a Social Web of Data. SIOC provides a Semantic Web ontology for representing rich data from the Social Web in RDF. It has recently achieved significant adoption through its usage in a variety of commercial and open-source software applications, and is commonly used in conjunction with the FOAF vocabulary for expressing personal profile and social networking information. By becoming a standard way for expressing user-generated content from such sites, SIOC enables new kinds of usage scenarios for online community site data, and allows innovative semantic applications to be built on top of the existing Social Web. The SIOC ontology was recently published as a W3C Member Submission, which was submitted by 16 organizations[28].

The SIOC ontology is divided in the main SIOC Core ontology, and in three modules: Access, Types and Services. The main classes and properties of the SIOC Core ontology are illustrated in the following Figure 1.3.

An example about a very basic document describing a blog entry, taken from the SIOC Core Ontology Specification[29], is displayed in the following Listing 1.3.

[26] `http://semanticweb.org/wiki/Semantic_Wiki_State_Of_The_Art`

[27] `http://sioc-project.org`

[28] `http://www.w3.org/Submission/2007/02/`

[29] `http://rdfs.org/sioc/spec/`

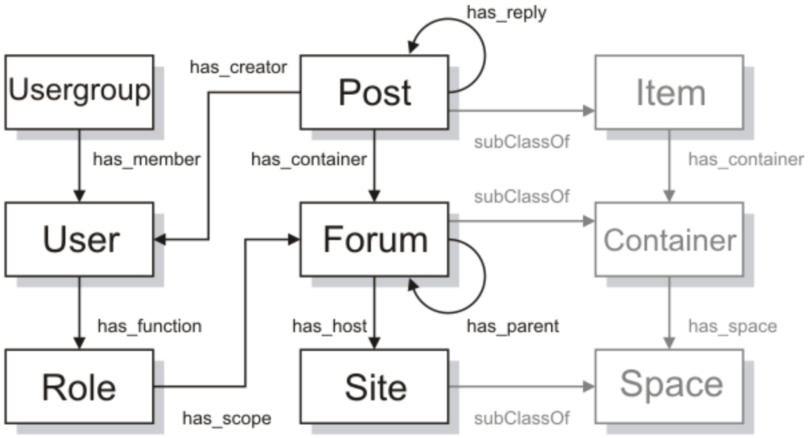


Figure 1.3: Main SIOC Core classes and properties

```
<sioc:Post rdf:about="http://johnbreslin.com/blog/2006/09/07/creating-
    connections-between-discussion-clouds-with-sioc/">
   <dcterms:title>Creating connections between discussion clouds with SIOC</
       dcterms:title>
   <dcterms:created>2006-09-07T09:33:30Z</dcterms:created>
   <sioc:has_container rdf:resource="http://johnbreslin.com/blog/index.php?
       sioc_type=site#weblog"/>
   <sioc:has_creator>
       <sioc:User rdf:about="http://johnbreslin.com/blog/author/cloud/" rdfs
           :label="Cloud">
           <rdfs:seeAlso rdf:resource="http://johnbreslin.com/blog/index.php
               ?sioc_type=user&amp;sioc_id=1"/>
       </sioc:User>
   </sioc:has_creator>
   <sioc:content>SIOC provides a unified vocabulary for content and
       interaction description: a semantic layer that can co-exist with
       existing discussion platforms.</sioc:content>
   <sioc:topic rdfs:label="Semantic Web" rdf:resource="http://johnbreslin.
       com/blog/category/semantic-web/"/>
   <sioc:topic rdfs:label="Blogs" rdf:resource="http://johnbreslin.com/blog/
       category/blogs/"/>
   <sioc:has_reply>
       <sioc:Post rdf:about="http://johnbreslin.com/blog/2006/09/07/creating
           -connections-between-discussion-clouds-with-sioc/#comment
           -123928">
           <rdfs:seeAlso rdf:resource="http://johnbreslin.com/blog/index.php
               ?sioc_type=comment&amp;sioc_id=123928"/>
```

```
      </sioc:Post>
   </sioc:has_reply>
</sioc:Post>
```

Listing 1.3: Describing a blog entry with SIOC

The brief example illustrated introduces the basics of SIOC. In other words, it says:

- There is a post titled "Creating connections between discussion clouds with SIOC" created at 09:33:30 on 2006-09-07 written by a user "Cloud" on topics "Blogs" and "Semantic Web" with contents described in sioc:content.
- More information about its author can be found at

  `http://johnbreslin.com/blog/index.php?sioc_type=user&sioc_id=1`
- The post has a reply and detailed SIOC information about this reply can be found at

  `http://johnbreslin.com/blog/index.php?sioc_type=comment&sioc_id=123928`

This simple example uses only two classes of SIOC objects: `sioc:Post` and `sioc:User`. There are other classes in SIOC used to describe more information about users, sites, communities and other objects. Further details about the Core ontology, and a full definition of these classes and related properties, can be found in the namespace located at: `http://rdfs.org/sioc/ns`.
SIOC modules are used to extend the available terms and to avoid making the SIOC Core Ontology too complex and unreadable.

- SIOC Access module[30] contains classes and properties that allow to express information about access rights such as users' permissions and status of content Items.

[30] `http://rdfs.org/sioc/access`

- SIOC Types Module[31] includes some of the SIOC Core Ontology multiple sub-classes for different types of Containers and Posts, such as: Wiki, WikiArticle, Weblog, BlogPost, etc.

- SIOC Services Module[32] provides a simple way to tell others about a web service (it should not be confused with web service definitions that define the details of a web service). A sioc:Service allows us to indicate that a web service is associated with (located on) a sioc:Site or a part of it.

Currently more than 50 applications are using SIOC[33], either as a common vocabulary to expose their data in RDF, alongside with FOAF for instance, as well as using existing SIOC data. By installing relevant SIOC export plugins, online community sites can generate linked data and start forming a critical mass of RDF data about user-created content [11]. Other tools allow users to browse SIOC data or to translate existing data, such as mailing list archives, to SIOC.

A simple and effective way to use and link to/from SIOC data is to interlink SIOC with other vocabularies such as FOAF and SKOS. By doing so it is possible to make online community data, described in SIOC, a more integrated part of the Web of Data. Common practice to facilitate linking to SIOC should be: the linking to social media sites and their user accounts on these sites by owners of FOAF profiles; then SIOC exporters can be optimized to make SIOC data easier to discover; finally Semantic Web indexing and lookup services can find and provide access to SIOC data [11]. A summary about the concepts of linking SIOC with other ontologies such as FOAF and SKOS, and about linking to SIOC data (especially the way to identify a user with his online accounts), is illustrated in Figure 1.4[34].

[31] http://rdfs.org/sioc/types

[32] http://rdfs.org/sioc/services

[33] http://sioc-project.org/applications

[34] image taken from: http://sioc-project.org/node/158

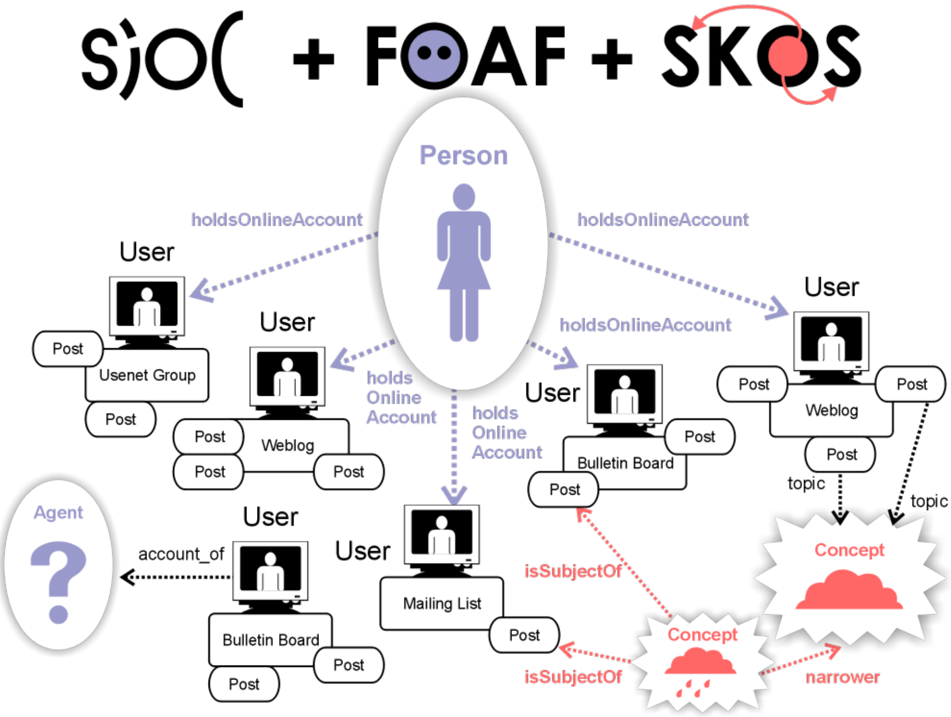


Figure 1.4: Interlinking SIOC, FOAF and SKOS.

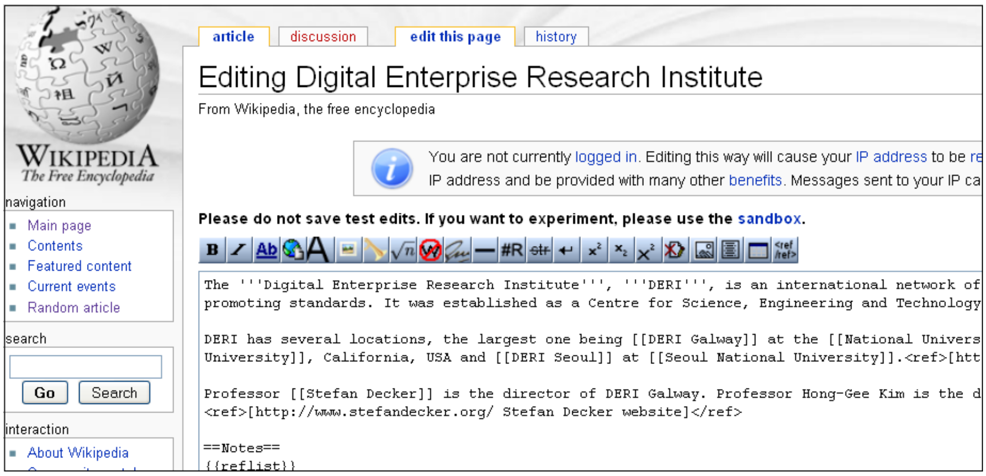
# Chapter 2

# Common Structural Features of Wikis

In this chapter will be considered 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 commenting or discussion areas; they link to other external sources or within the wiki; they describe categories into particular structures. For each identified feature a brief overview of its goal is given. In this study either semantic or traditional wikis have been considered, these characteristics are common in both of them.

## 2.1 Multi-authoring

An inescapable feature of wikis is that multiple users are allowed to modify the same content, enabling some kind of collective intelligence process and collaboration. In this regard the wiki infrastructure should provide a model to identify users and theirs modifications, marking events with a corresponding timestamp. The authoring interface of wikis is what makes wikis a quite easy system to use as it avoids the necessity of technical knowledge in order to edit a page. This feature is strongly linked to the pages versioning function, because as soon as a user contributes to an article editing it, a new version of the article is created. Fig 2.1 illustrates an example of a typical editing interface showed on the Wikipedia.



Editing Digital Enterprise Research Institute

From Wikipedia, the free encyclopedia

You are not currently logged in. Editing this way will cause your IP address to be re
IP address and be provided with many other benefits. Messages sent to your IP ca

**Please do not save test edits. If you want to experiment, please use the sandbox.**

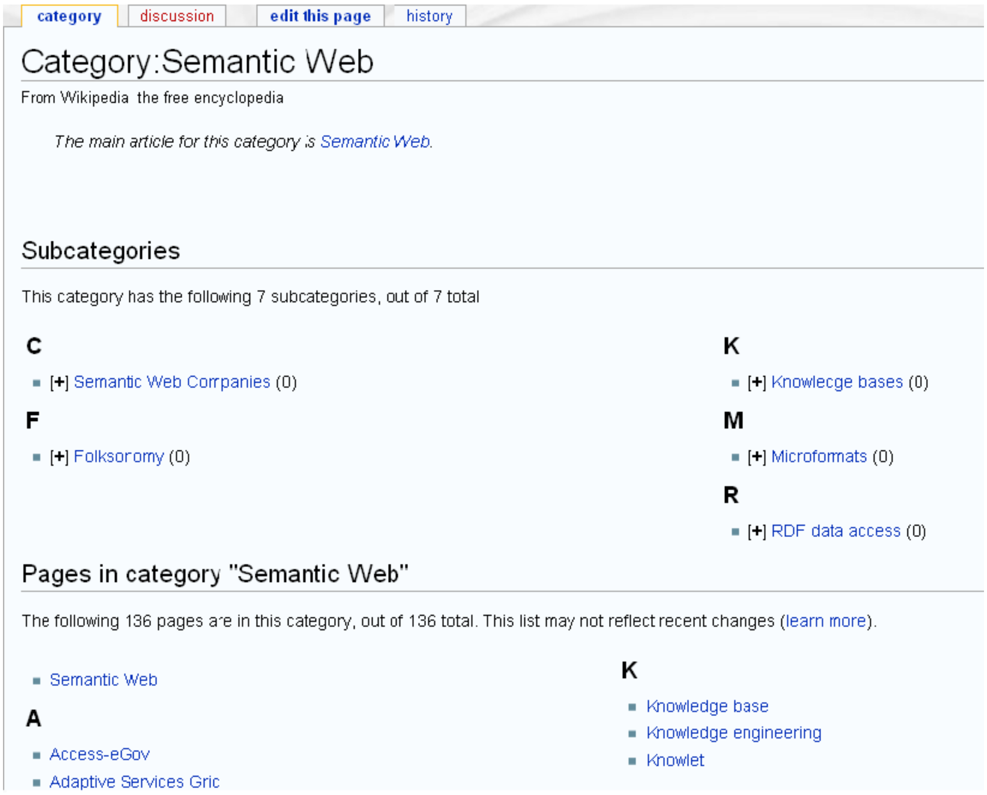
```
The '''Digital Enterprise Research Institute''', '''DERI''', is an international network of
promoting standards. It was established as a Centre for Science, Engineering and Technology

DERI has several locations, the largest one being [[DERI Galway]] at the [[National Univers
University]], California, USA and [[DERI Seoul]] at [[Seoul National University]].<ref>[htt

Professor [[Stefan Decker]] is the director of DERI Galway. Professor Hong-Gee Kim is the d
<ref>[http://www.stefandecker.org/ Stefan Decker website]</ref>

==Notes==
{{reflist}}
```

Figure 2.1: Editing a page on Wikipedia

## 2.2 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 maintain the original taxonomy structure. Categories provide automatic indexes that are useful as tables of contents. Categories are also used as universal "tags" for articles, describing that the article belongs to a certain group of articles. When adding a category to an article, the name of the category is arbitrary but, of course, users are encouraged to try to use categories that already exist instead of creating new ones.

For example, on the Wikipedia, the "Digital Enterprise Research Institute" article belongs to the categories: Scientific organizations, Semantic Web and Scientific organization stubs. Each category can be browsed as in the example displayed in Fig 2.2.



Category:Semantic Web

From Wikipedia the free encyclopedia

*The main article for this category is Semantic Web.*

Subcategories

This category has the following 7 subcategories, out of 7 total

C

- [+] Semantic Web Companies (0)

F

- [+] Folksonomy (0)

K

- [+] Knowledge bases (0)

M

- [+] Microformats (0)

R

- [+] RDF data access (0)

Pages in category "Semantic Web"

The following 136 pages are in this category, out of 136 total. This list may not reflect recent changes (learn more).

- Semantic Web

A

- Access-eGov
- Adaptive Services Gric

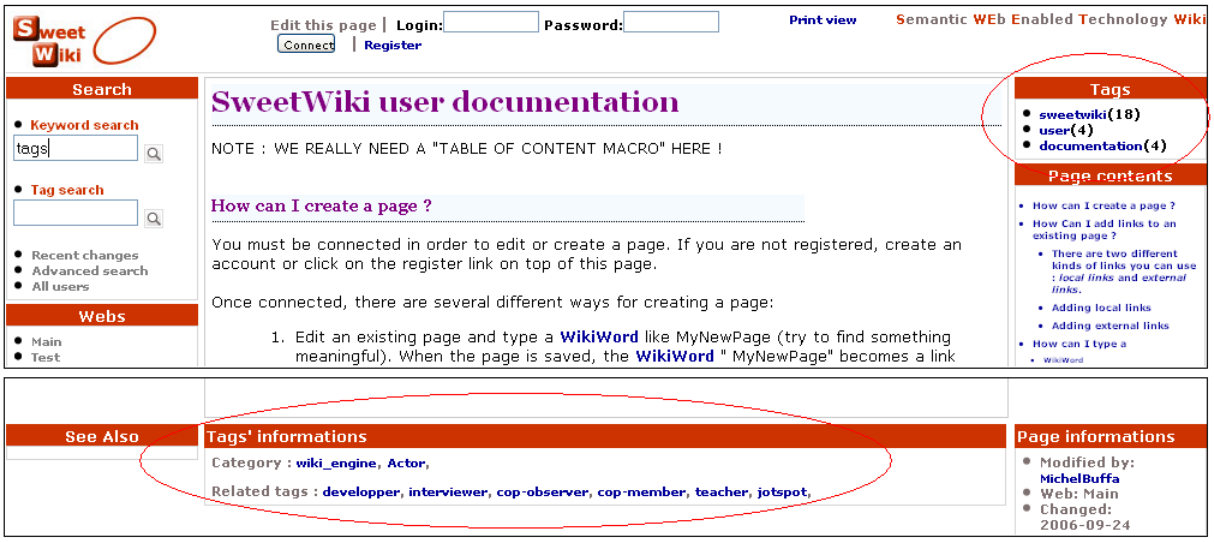
K

- Knowledge base
- Knowledge engineering
- Knowlet

Figure 2.2: Browsing Categories on Wikipedia

## 2.3 Social Tagging

This process is an unstructured (or better, free structured) approach to classification with users assigning their own labels to a subject. 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 tag lead to a non-organised but dynamic organisation process, known as a "folksonomy", rather than the hierarchical structures that a taxonomy or other formal classification system usually provides. SweetWiki for example, as detailed in Chapter 3, is one of the few semantic wikis that implements this functionality. A SweetWiki page showing tags related to the article is illustrated in Fig 2.3.



SweetWiki user documentation

NOTE : WE REALLY NEED A "TABLE OF CONTENT MACRO" HERE !

How can I create a page ?

You must be connected in order to edit or create a page. If you are not registered, create an account or click on the register link on top of this page.

Once connected, there are several different ways for creating a page:

1. Edit an existing page and type a WikiWord like MyNewPage (try to find something meaningful). When the page is saved, the WikiWord " MyNewPage" becomes a link

Tags' informations

Category : wiki_engine, Actor,

Related tags : developper, interviewer, cop-observer, cop-member, teacher, jotspot,

Figure 2.3: Social Tagging on SweetWiki

## 2.4 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. In this regard interesting is the explanation of the Wikipedia's approach described in the guidelines[1]. In wikis usually talk pages are the same as normal wiki articles, but with a different URL or namespace.

[1] http://en.wikipedia.org/wiki/Wikipedia:Talk_page_guidelines

## 2.5 Backlinks

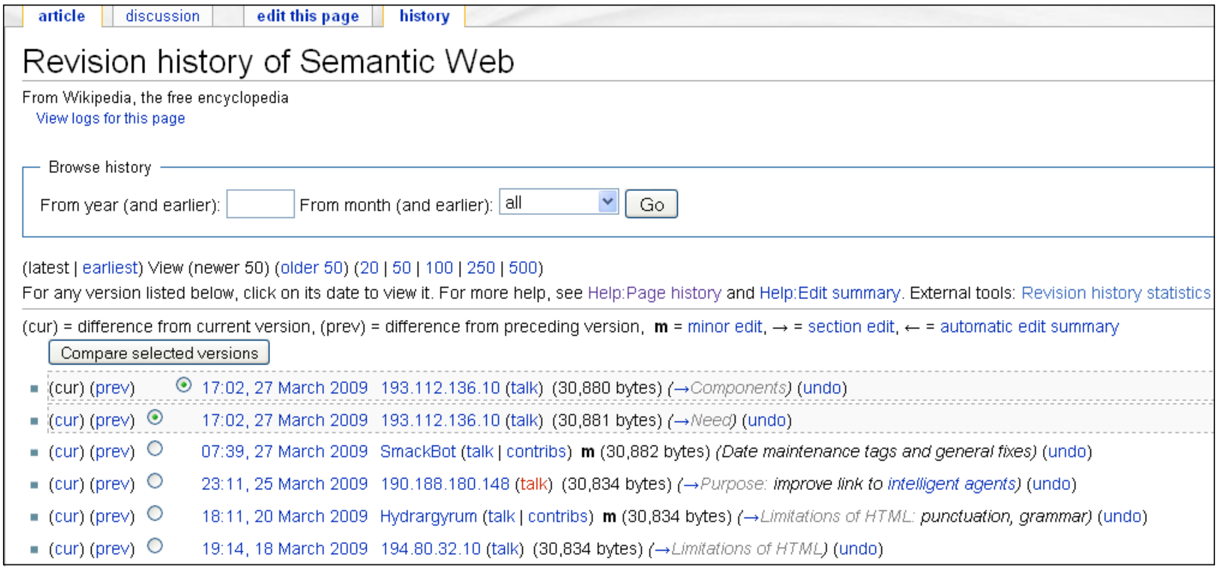
Backlinks are an important feature of wikis, as they allow to visualize instantaneously all the incoming links to a website or web page. More precisely they are wiki internal links pointing to a wiki article. It is a very common wiki feature and usually a list of backlinks related to a page is provided by clicking on a link called *"What links here"*. Backlinks of a webpage may be of significant interest: they indicate who is paying attention to it. It is also important to notice that backlinks are not an inverse relation of internal links.

## 2.6 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.

Normally with this functionality it is possible to browse all changes made to the page, to view a specific version, to compare an old version with the current version or with its predecessor or with another specific revision. A very well detailed example is provided by the Wikipedia help guide[2], as in Fig 2.4.

[2] http://en.wikipedia.org/wiki/Help:Page_history



Revision history of Semantic Web

From Wikipedia, the free encyclopedia
View logs for this page

Browse history

From year (and earlier): From month (and earlier): all

(latest | earliest) View (newer 50) (older 50) (20 | 50 | 100 | 250 | 500)

For any version listed below, click on its date to view it. For more help, see Help:Page history and Help:Edit summary. External tools: Revision history statistics

(cur) = difference from current version, (prev) = difference from preceding version, **m** = minor edit, → = section edit, ← = automatic edit summary

- (cur) (prev) ● 17:02, 27 March 2009 193.112.136.10 (talk) (30,880 bytes) *(→Components)* (undo)
- (cur) (prev) ● 17:02, 27 March 2009 193.112.136.10 (talk) (30,881 bytes) *(→Need)* (undo)
- (cur) (prev) ○ 07:39, 27 March 2009 SmackBot (talk | contribs) **m** (30,882 bytes) *(Date maintenance tags and general fixes)* (undo)
- (cur) (prev) ○ 23:11, 25 March 2009 190.188.180.148 (talk) (30,834 bytes) *(→Purpose: improve link to intelligent agents)* (undo)
- (cur) (prev) ○ 18:11, 20 March 2009 Hydrargyrum (talk | contribs) **m** (30,834 bytes) *(→Limitations of HTML: punctuation, grammar)* (undo)
- (cur) (prev) ○ 19:14, 18 March 2009 194.80.32.10 (talk) (30,834 bytes) *(→Limitations of HTML)* (undo)

Figure 2.4: Pages Versioning on Wikipedia

# Chapter 3

# Modeling Structural Features of Wikis: *State of the Art*

In this chapter it will be explored the way some available Semantic wikis currently implement a typical wiki's structure. While the aim was to model wiki features by extending SIOC, several vocabularies have been already proposed to achieve this goal. Some of them in the following sections, by distinguishing models created with general purposes and models created for a particular wiki engine but not available on the Web. Hence it is important to remark that this study does not aim at describing the state of the art in Semantic wikis, it is better focused on the ontology model that defines all the fundamental structural features required by a wiki. My work started considering a list of available "state of the art" Semantic wikis retrieved from the `SemanticWeb.org`[1] website. Then I made a selection of only three Semantic wikis with an ontology model and a practical implementation as well (*SweetWiki*, *IkeWiki* and *Semantic MediaWiki*), and three theoretical ontology models designed more generically, independently of semantic wikis applications (*Wiki Interchange Format*, *WikiOnt* and *SIOC*).

[1] `http://www.semanticweb.org`

In each section of this chapter it will also be introduced how the semantic wiki models considered before may implement a generic wiki page about the *Digital Enterprise Research Centre (DERI)*. As regards pages coming from a wiki model with a practical implementation, it is possible to state that they are very close to the actual implementation of a generic page, rather than wiki pages based on models such as *WIF*, *WikiOnt* and *SIOC*, which are mainly invented and based only on the provided specifications.

## 3.1 SweetWiki and its related model

SweetWiki[2] [14], developed by the Edelweiss team located at INRIA Sophia Antipolis-Méditerranée, is a semantic wiki based on the CORESE engine[3]. CORESE is an RDF engine, based on Conceptual Graphs (CG), that enables the processing of RDFS, OWL Lite and RDF statements relying on a CG formalism. It can perform SPARQL Queries and run rules over the RDF graph.

SweetWiki relies on web standards for the wiki page format (XHTML), the macros included in pages (JSPX/XML tags), the semantic annotations (RDFa, RDF), and the ontologies it manipulates (OWL Lite). It improves access to information with faceted navigation, enhanced search tools and awareness capabilities, acquaintance networks identification, etc. It comes with an embedded ontology editor and a reasoning engine, and it uses SPARQL as query language. Another interesting point is that SweetWiki allows metadata to be extracted and exploited by other applications.

The wiki makes use of two ontologies: one for the wiki structure and one for the content, and what distinguishes SweetWiki is that many concepts are annotated on the same page, using RDFa. As regards the wiki structure ontology, all the classes

[2] http://sweetwiki.inria.fr/sweetwiki

[3] http://www-sop.inria.fr/edelweiss/software/corese/

and properties are listed in Table 3.1, and they have been directly retrieved from the namespace located at `http://sweetwiki.inria.fr/ontology#`.

| **SweetWiki** | **Ontology** |
|---|---|
| CLASSES: | PROPERTIES: |
| Document | isChildCategoryOf |
| WikiPage | isRelatedToCategory |
| Image | backwardLink |
| Video | forwardLink |
| Calendar | createLink |
| File | hasForKeyWord |
| Web | noWikiWord |
| Person | includeDocument |
| Category | isTheVersionNumber |
| KeyWord | isInterestedBy |
| RSS | pageHasFeed |
| Version | hasForWeb |
| NewConcept | seeAlso |
|  | author |
|  | camelWord |
|  | name |
|  | description |
|  | modification |
|  | pageHasVersion |

Table 3.1: Classes and properties of the SweetWiki ontology

In order to evaluate the way SweetWiki models the wiki structure with the ontology, it has been taken into consideration the most important wiki features spotted in the previous chapter and described the implementation of each one with a brief description of the modeling solution.

- **Multi-authoring:** in order to accomplish to this feature SweetWiki uses a class called `Person` to identify a human being and it links a `Person` with a `WikiPage` using the `author` property. Every single change is modeled with the `modification` property that states the date of the last update.
- **Categories:** it is provided a `Category` class that allows users to divide concepts into categories. These categories could be identified as in a father-child relationship using the `isChildCategoryOf` property, or a keyword could be included into a category with the `isRelatedToCategory` property.
- **Social tagging:** it is one of the most important features offered by SweetWiki considering that a few Semantic wikis model this feature. In particular this could be done using the class `KeyWord` (and `NewConcept` for the new keywords not yet attached to the domain ontology). Keywords could be linked to a document in general, or a wiki page in particular, using the property `hasForKeyWord`, and they could also be linked to a `Category` using the property `isRelatedToCategory`.
- **Discussions:** this feature is currently not modeled.
- **Backlinks:** using the `backwardLink` property is possible to include all the pages who contains link to this `WikiPage`.
- **Pages versioning:** it is possible to trace each version of a `WikiPage` using the `Version` class that is declared as a `WikiPage` subclass. Then the

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.

It is now possible to consider an example of a wiki source page generated by SweetWiki. The following page code is in XHTML with metadata embedded using the RDFa syntax, exactly as SweetWiki does. I retrieved a code example directly from a SweetWiki page source located at the demo website [4], and then I tried to adapt the content to a different page. The code posted regards only the main-central part of the wiki page, all the other surrounding structures of a page are pointless in this context.

```
<html xmlns:WIKI="http://sweetwiki.inria.fr/ontology#" xmlns="http://www.w3.
    org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <meta property="WIKI:name" content="DigitalEnterpriseResearchInstitute"/>
  <link rel="WIKI:author" href="http://sweetwiki.inria.fr/user#UserName"/>
  <meta property="WIKI:description" content="Digital Enterprise Research
      Institute (DERI)"/>
  <meta property="WIKI:modification" content="2008-11-12T15:55:46"/>
  <meta property="WIKI:isTheVersionNumber" content="7"/>
  <meta property="WIKI:hasForWeb" content="Main"/>
  <meta property="WIKI:forwardLink" content="Semantic_Web"/>
  <link rel="WIKI:backwardLink" href="/sweetwiki/data/Main/SIOC.jsp"/>
  <link rel="WIKI:hasForKeyWord" href="http://sweetwiki.inria.fr/folksonomy/
      sweetwiki#semanticweb"/>
  <link rel="WIKI:hasForKeyWord" href="http://sweetwiki.inria.fr/folksonomy/
      sweetwiki#research"/>
  <link rel="WIKI:hasForKeyWord" href="http://sweetwiki.inria.fr/folksonomy/
      sweetwiki#ireland"/>
</head>
<body>
  <h1>Digital Enterprise Research Institute</h1>
  <p>
```

[4] http://argentera.inria.fr/sweetwiki/

```
  The <b>Digital Enterprise Research Institute</b>, <b>DERI</b>, is an
      international network of independent research organisations aiming at
      developing <a href="/sweetwiki/data/Main/Semantic_Web.jsp" title="
      Semantic Web">Semantic Web</a> technologies and promoting standards.
      It was established ...
  </p>
  <table xmlns:sparql="http://www.w3.org/2005/sparql-results#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:cos="http://www.inria.fr/acacia/corese#"
  class="plain">
</body>
</html>
```

Listing 3.1: A SweetWiki page source about DERI

As we can see from the Listing 3.1 at the top of the page the SweetWiki ontology namespace is declared, and the prefix WIKI is used for each element. Then, inside the <head> tag, some of the used classes and properties are further detailed, expressing relationships (semantic) to the current page. Indeed here is possible to find the properties described before included into <meta> and <link> tags. The subject of all this properties is the wiki page itself. Finally in the <body> section it is displayed the main text of the article, always in XHTML, and a further declaration of other common namespaces.

## 3.2 Ikewiki and its related model

IkeWiki[5] is a Java-based semantic wiki engine developed by Salzburg Research[25]. Despite the IkeWiki project development has stopped, and currently developers are working on a successor called KiWi[6], I took into consideration the IkeWiki last release: 2.0 (Mar 5 2008). IkeWiki offers collaborative knowledge engineer-

[5] http://ikewiki.salzburgresearch.at

[6] http://www.kiwi-project.eu

ing, an easy to use and interactive user interface, and the support for different levels of formalisation ranging from informal texts to formal ontologies. It aims at creating instance data based on an existing ontology but also at being a tool for creating and editing ontologies.

IkeWiki makes use of the recommended Semantic Web technologies RDF and OWL and it is mainly developed in Java. It incorporates several external ontologies so that it is possible to insert in every page a large variety of metadata properties. From a structural-ontological point of view the model provided is more content focused than the SweetWiki's one. Indeed some of the IkeWiki features available (for example "backlinks" and "pages versioning") are not modeled into the ontology. The IkeWiki ontology consists of classes and properties which are divided by type. There are *Object properties*, *Annotation properties* and *Datatype properties*, as listed in Table 3.2.

| **IkeWiki Ontology** | |
|---|---|
| CLASSES | Object PROPERTIES |
| Page | instanceHasType |
| NavMenuItem | hasRole |
| MultimediaObject | hasAuthor |
| Image | NavigationalLink |
| Video | ImageLink |
| Audio | IncludeLink |
| MP3Audio | TemplateLink |
| FlashVideo | Attachment |
| Document | hasDiscussion |
| PDFDocument | Annotation PROPERTIES |

| | |
|---|---|
| MSWordDocument | article-xml |
| User | article-structured |
| Role | mime-type |
| Permission | Datatype PROPERTIES |
| Template | NavMenuPosition |
| | NavMenuLabel |

Table 3.2: Classes and properties of the IkeWiki Ontology

The OWL ontology specification has been retrieved from the `ikewiki.owl` file located at `./WEB-INF/ontologies/base/` in the main software package of the version 2.0. The following points summarize the IkeWiki implementation of each wiki feature previously spotted.

- **Multi-authoring:** this feature is supported by the `User` class that is a `foaf:Person` subclass. Each user could be linked to a `Role` or a `Permission`. The `hasAuthor` property associates an author (or a User) with a resource in the wiki, and it is a subclass of the property `foaf:maker`. In fact an important property is not currently modeled in the ontology: the contributions; in my opinion, when a user modifies a page, but he is not the author, it is important to specify that he is only a contributor. This property has a strong link with the versioning model provided because of the way each modification is stored giving the ability to identify every single change as a contribution.
- **Categories:** in IkeWiki there is not a precise model that agrees with this function. In fact the only way to create a taxonomy is to annotate a resource type associated to the page, and this could be done by including concepts

and structures from other external ontologies like SKOS or DC or more specific ones. For example it is possible to associate a page about Bilberries to a `skos:Concept` page type and/or a `bio:Species` page type (because the Blueberry is a species in the biology domain).

- **Social tagging:** this operation is currently not modeled.
- **Discussions:** to accomplish to this requirement IkeWiki simply uses a link to an external ontology, in particular to the SIOC ontology. A property called `hasDiscussion` points to a `sioc:Forum`, so that a relation from a resource to a SIOC discussion forum could be established.
- **Backlinks:** this feature is currently not modeled.
- **Pages versioning:** this feature is currently not modeled.

In order to understand more clearly the modeling solutions described, it is possible to consider an example of a wiki source page generated by IkeWiki. The following page code is made in N3 notation syntax and describes the underlying semantics of a page about DERI. This source code has been created from the modification of another IkeWiki page, extracted directly from the "export" function provided by the IkeWiki interface.

```
<http://www.example.com/Digital_Enterprise_Research_Institute>
  a  <http://www.w3.org/2000/01/rdf-schema#Resource> , <http://www.w3.org
      /2004/02/skos/core#Concept> ;
  <http://www.w3.org/2000/01/rdf-schema#comment>
    "Digital Enterprise Research Institute (DERI)" ;
  <http://ikewiki.srfg.at/base/article-structured>
    """The ’’’Digital Enterprise Research Institute’’’, DERI, is an
        international network of independent research organisations aiming
        at developing [[Semantic_Web | Semantic Web]] technologies and
        ...""" ;
  <http://ikewiki.srfg.at/base/article-xml>
    "<wif:page xmlns:wif=\"http://ikewiki.srfg.at/syntax/1.0/core\">
    <title xmlns=\"http://ikewiki.srfg.at/syntax/1.0/core\">
        Digital_Enterprise_Research_Institute</title>
    [...]
    <wif:paragraph>The <wif:strong>Digital Enterprise Research Institute,
        DERI</wif:strong>, is an international network of independent
        research organisations aiming at developing <wif:intlink target=\"
        Semantic_Web\">Semantic Web</wif:intlink> technologies and ...
    </wif:page>" ;
  <http://ikewiki.srfg.at/base/hasAuthor> <http://ikewiki.srfg.at/user/
      Administrator> ;
  <http://ikewiki.srfg.at/base/hasDiscussion>
    <http://ikewiki.srfg.at/discussion/Forum-11xy11xy11> ;
  <http://purl.org/dc/elements/1.1/creator> "Administrator" ;
  <http://purl.org/dc/elements/1.1/date> "2008-11-12" ;
  <http://purl.org/dc/elements/1.1/language> "en" ;
  <http://purl.org/dc/elements/1.1/title> "Digital Enterprise Research
      Institute" ;
  [...]
```

Listing 3.2: DERI page source example in IkeWiki

The above Listing 3.2 describes the information of a wiki page detailed before in this section, with the exception of forward/back-links considered later in Listing 3.3. The subject is always identified by the <http://www.example.com /Digital_Enterprise_Research_Institute> URI, and this is declared as a rdfs:Resource and a skos:Concept. So the listing above describes the concept DERI that could be linked to a wiki page. The content of the article is modeled in two ways: the first in typical wiki structured language as object of an article-structured property, and the second in XML format (Wiki Interchange Format, or WIF, compliant) as object of an article-xml property. All the other used properties are straightforward to understand, to note that several Dublin Core properties have been used to decribe the document characteristics. Furthermore the hasDiscussion property links to a sioc:Forum and the hasAuthor property is sub-class of foaf:maker.

The following Listing 3.3 shows the implementation of forward/back-links between the *DERI* article and the *National University of Ireland, Galway* article.

```
<http://www.example.com/Digital_Enterprise_Research_Institute>
  [...]
  <http://www.w3.org/2000/01/rdf-schema#seeAlso>
    <http://www.example.com/National_University_of_Ireland,_Galway> ;
  <http://ikewiki.srfg.at/base/NavigationalLink>
    <http://www.example.com/National_University_of_Ireland,_Galway> ;
  <http://www.w3.org/2004/02/skos/core#related>
    <http://www.example.com/National_University_of_Ireland,_Galway> ;
  <http://www.w3.org/2004/02/skos/core#semanticRelation>
    <http://www.example.com/National_University_of_Ireland,_Galway> .

<http://www.example.com/National_University_of_Ireland,_Galway>
  <http://www.w3.org/2000/01/rdf-schema#seeAlso>
    <http://www.example.com/Digital_Enterprise_Research_Institute> ;
  <http://ikewiki.srfg.at/base/NavigationalLink>
    <http://www.example.com/Digital_Enterprise_Research_Institute> ;
  <http://www.w3.org/2004/02/skos/core#related>
    <http://www.example.com/Digital_Enterprise_Research_Institute> ;
  <http://www.w3.org/2004/02/skos/core#semanticRelation>
    <http://www.example.com/Digital_Enterprise_Research_Institute> .
```

Listing 3.3: Interlinking wiki articles in IkeWiki

The document displayed in Listing 3.3 starts defining the *National University of Ireland, Galway* resource as a page linked by the *DERI* resource, and as an object semantically related to the *DERI* subject (using `rdfs:seeAlso`, `skos:related` and `skos:semanticRelation`). The same for the *DERI* resource, for which the *National University of Ireland, Galway* is declared as a backlink.

## 3.3 Semantic MediaWiki and its related model

Semantic MediaWiki[7][19], or briefly SMW, is an extension to MediaWiki, enabling it to become a Semantic wiki. MediaWiki[8] "is a free software wiki package written in PHP, originally for use on Wikipedia. It is now used by several other projects of the non-profit *Wikimedia Foundation* and by many other wikis". Semantic MediaWiki is developed by an extended group of developers worldwide (the SMW Project), significant developments have contributed by Institut AIFB Universität Karlsruhe. A fundamental feature provided is the enhancement of WikiML[9] annotations to allow users to include relations and properties to wiki pages. SMW conceptualizes pages as concepts and stores the semantic data in Mediawiki's MySQL database, but it can also be exported. SMW's content is grounded in OWL DL, and this data can also be retrieved via SMW's web interface for OWL export. Several typical Semantic wiki features are provided through *Special Pages* hiding the underlying structure and the part of ontology used. These special pages are built-in query forms without user-edited content that use `Special:` as a namespace prefix.

SMW uses a particular ontology to represent the semantic data exported from a page by a user, and it is called *SWIVT* (Semantic Wiki Vocabulary and Terminology)[10]. 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 its most basic concepts, such as Wikipage or Subject, are generic enough to apply to many semantic wikis, and thus to allow simple reuse or mapping between different systems.

[7] `http://semantic-mediawiki.org/`

[8] `http://www.mediawiki.org`

[9] WikiMl (for Wiki Markup Language) is a non XML markup language similar to the formats used by WikiWikiWeb systems for user entry.

[10] `http://semantic-mediawiki.org/swivt/`

In SMW annotation syntax is most relevant (and most visible) to wiki editors, but it is a small part of the overall SMW system. The underlying conceptual framework, based on properties and types is rather more relevant. Different namespaces are used to distinguish the semantic roles that wiki pages may play: they can be *individual elements* (the majority of the pages, describing elements of the domain of interest), *categories* (used to classify individual elements, and also to create sub-categories), *properties* (relationships between two pages or a page and a data value), and *types* (used to distinguish different kinds of properties).
The following analysis is mainly based on the SWIVT ontology (as displayed in Table 3.3), even though this doesn't reflect the effective underlying ontology, actually properties listed in the table are properties to define the ontology derived from the wiki page, not the structure itself. In this regard a comment on the way SMW may implement each feature is added.

| **SWIVT Ontology** | | |
|---|---|---|
| CLASSES | PROPERTIES | INSTANCES |
| BuiltinType | allowedValue | EmailType |
| Container | baseProperty | GeoType |
| ConversionFactor | conversionFactor | NumberType |
| CustomType | creationDate | PageType |
| NAryType | displayUnit | StringType |
| Subject | factor | TextType |
| Type | mainUnit | TemperatureType |
| UnitType | modifier | TimeType |
| Wikipage | object | URIType |
| | page | |
| | sourceSite | |

| | | |
|---|---|---|
| | type<br>unit<br>value | |

Table 3.3: Classes and properties of the SWIVT ontology

- **Multi-authoring:** the SWIVT ontology doesn't provide a "User-related" class, each user is linked to a wiki page called User:Username. Through a special page (Special:Contributions) is possible to browse all the contributions made on that article.
- **Categories:** not modeled by the SWIVT ontology, but one of the semantic roles that wiki pages may play is a Category. Each category is mapped into a wiki page called Category:*CategName*.
- **Social tagging:** feature not provided.
- **Discussions:** not modeled by the SWIVT ontology, each discussion is linked to a Talk:PageTitle page.
- **Backlinks:** not modeled by the SWIVT ontology, they are accessible only through a "special page".
- **Pages versioning:** not modeled by the ontology.

In order to understand more clearly the modeling solutions described, it is possible to consider an example of a wiki source page generated by Semantic MediaWiki. The following page code, expressed in RDF, describes the underlying semantics of a page about DERI. This source code has been created from the modification of another SMW page, extracted directly from the "export" function provided by SMW itself.

```
<rdf:RDF>
<!-- Ontology header -->
  <owl:Ontology rdf:about="">
    <swivt:creationDate rdf:datatype="http://www.w3.org/2001/XMLSchema#
        dateTime">2008-11-13T17:41:51+01:00</swivt:creationDate>
    <owl:imports rdf:resource="http://semantic-mediawiki.org/swivt/1.0"/>
  </owl:Ontology>
<!-- exported page data -->
  <swivt:Subject rdf:about="http://sandbox.semantic-mediawiki.org/wiki/
      Special:URIResolver/Digital_Enterprise_Research_Institute">
    <rdfs:label>Digital Enterprise Research Institute</rdfs:label>
    <swivt:page rdf:resource="http://sandbox.semantic-mediawiki.org/wiki/
        Digital_Enterprise_Research_Institute"/>
    <rdfs:isDefinedBy rdf:resource="http://sandbox.semantic-mediawiki.org/
        wiki/Special:ExportRDF/Digital_Enterprise_Research_Institute"/>
    <rdf:type rdf:resource="http://sandbox.semantic-mediawiki.org/wiki/
        Special:URIResolver/Category-3AScientific_Organizations"/>
  </swivt:Subject>
  <swivt:Subject rdf:about="http://sandbox.semantic-mediawiki.org/wiki/
      Special:URIResolver/User-3AUserName">
    <rdfs:label>User:UserName</rdfs:label>
    <swivt:page rdf:resource="http://sandbox.semantic-mediawiki.org/wiki/
        User:UserName"/>
    <rdfs:isDefinedBy rdf:resource="http://sandbox.semantic-mediawiki.org/
        wiki/Special:ExportRDF/User:UserName"/>
    <property:Foaf-3Ainterested rdf:resource="http://sandbox.semantic-
        mediawiki.org/wiki/Special:URIResolver/
        Digital_Enterprise_Research_Institute"/>
  </swivt:Subject>
  <swivt:Subject rdf:about="http://sandbox.semantic-mediawiki.org/wiki/
      Special:URIResolver/Semantic_Web">
    <rdfs:label>Semantic Web</rdfs:label>
    <swivt:page rdf:resource="http://sandbox.semantic-mediawiki.org/wiki/
        Semantic_Web"/>
    <rdfs:isDefinedBy rdf:resource="http://sandbox.semantic-mediawiki.org/
        wiki/Special:ExportRDF/Semantic_Web"/>
    <property:Has_topic rdf:resource="http://sandbox.semantic-mediawiki.org/
        wiki/Special:URIResolver/Digital_Enterprise_Research_Institute"/>
  </swivt:Subject>
  <swivt:Subject rdf:about="http://sandbox.semantic-mediawiki.org/wiki/
      Special:URIResolver/DERI">
    <rdfs:label>DERI</rdfs:label>
    <swivt:page rdf:resource="http://sandbox.semantic-mediawiki.org/wiki/
        DERI"/>
    <rdfs:isDefinedBy rdf:resource="http://sandbox.semantic-mediawiki.org/
        wiki/Special:ExportRDF/DERI"/>
    <property:Instance_of rdf:resource="http://sandbox.semantic-mediawiki.
        org/wiki/Special:URIResolver/Digital_Enterprise_Research_Institute
        "/>
  </swivt:Subject>
[...]
```

Listing 3.4: DERI page source example in SMW, 1st part

The Listing 3.4 above is related to the first part of the RDF page generated. It includes the ontology definition (inside the `owl:Ontology` class description) and the main subjects of the page. There are four `swivt:Subject` defining the relationships between `Digital_Enterprise_Research_Intitute` and `DERI`, `Semantic Web`, `User:UserName`. These resources use three special *Object properties* to define the relationships described in the second part of the source code example, in the following Listing 3.5.

```
[...]
<owl:ObjectProperty rdf:about="http://sandbox.semantic-mediawiki.org/wiki/
    Special:URIResolver/Property-3AInstance_of">
  <rdfs:label>Instance of</rdfs:label>
  <swivt:page rdf:resource="http://sandbox.semantic-mediawiki.org/wiki/
      Property:Instance_of"/>
  <rdfs:isDefinedBy rdf:resource="http://sandbox.semantic-mediawiki.org/wiki
      /Special:ExportRDF/Property:Instance_of"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://sandbox.semantic-mediawiki.org/wiki/
    Special:URIResolver/Property-3AHas_topic">
  <rdfs:label>Has topic</rdfs:label>
  <swivt:page rdf:resource="http://sandbox.semantic-mediawiki.org/wiki/
      Property:Has_topic"/>
  <rdfs:isDefinedBy rdf:resource="http://sandbox.semantic-mediawiki.org/wiki
      /Special:ExportRDF/Property:Has_topic"/>
</owl:ObjectProperty>
[...]
```

Listing 3.5: DERI page source example in SMW, 2nd part: definition of Object properties

The three *Object Properties* described in the listing above are `Instance_of`, `Has_topic` and `Foaf:interested`. These properties are used in the first part of the example (see Listing 3.4) and have been defined by SMW, or by SMW's users, to extend the ontology and the set of properties available within the wiki. Each *object property* is expressed with the `property:` prefix, and is linked to its RDF description using the `Special:ExportRDF` functionality.

```
  <owl:AnnotationProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#
      page">
   <rdfs:isDefinedBy rdf:resource="http://semantic-mediawiki.org/swivt/1.0"/>
  </owl:AnnotationProperty>
  <owl:AnnotationProperty rdf:about="http://semantic-mediawiki.org/swivt/1.0#
      creationDate">
   <rdfs:isDefinedBy rdf:resource="http://semantic-mediawiki.org/swivt/1.0"/>
  </owl:AnnotationProperty>
  <owl:Class rdf:about="http://semantic-mediawiki.org/swivt/1.0#Subject">
   <rdfs:isDefinedBy rdf:resource="http://semantic-mediawiki.org/swivt/1.0"/>
  </owl:Class>
</rdf:RDF>
```

Listing 3.6: DERI page source example in SMW, 3rd part: References to the SWIVT Ontology

The above Listing 3.6 describes the last part of the RDF export of a page. It explains all the ontology references used above within the RDF description, precisely where the prefix `swivt:` has been used instead of the complete ontology path.

After this analysis it is possible to conclude that SMW is not the best example of ontological-structural modeling to take into consideration regarding the initial study purposes. Anyway it has been chosen to keep it in the comparison because of its importance and its advanced level of development in the general Semantic wiki landscape.

## 3.4 Wiki Interchange Format

Wiki Interchange Format (WIF)[11] is a project that allows data exchange between wikis and related tools. Different from other approaches, it also tackle the problem of page content and annotations[26]. WIF defines a subset of XHTML as an over-the-wire format for wiki content exchange. A WIF API is supposed to extract WIF from a wiki and be able to insert WIF in the desired wiki syntax back into the wiki.

The WIF project is recently evolved in the Structured Text Interchange Format (STIF)[12], i.e. a subset of XHTML, which aims more generally to exchange structured text between different applications, not only wikis. Anyway, regarding the main purposes of this thesis, I considered more interesting the evaluation of the ontology model[13] proposed by the WIF team, a model that aims to be as general as possible. Classes and properties of this ontology are listed in Table 3.4.

| **WIF Ontology** |
|---|
| CLASSES<br>WikiUser<br>Page |
| PROPERTIES<br>hasUserPage |

[11] http://semanticweb.org/wiki/Wiki_Interchange_Format

[12] http://semanticweb.org/wiki/Structured_Text_Interchange_Format

[13] http://wif.ontoware.org/2005/04/

| |
|---|
| hasMimeType |
| hasName |
| hasSize |
| hasAuthor |
| hasPreviousVersion |
| hasSubpage |
| hasChangeDate |

Table 3.4: Classes and properties of the WIF ontology

- **Multi-authoring:** this feature is supported by the `WikiUser` class (subclass of `foaf:Person`) that identifies an author of the wiki, and that could be linked to a wiki page (simply: `Page`) with the `hasAuthor` property. Each user may be linked to a personal Page with the `hasUserPage` attribute, and every single modification could be marked with `hasChangeDate` that describes the change date.
- **Categories:** this feature is not modeled.
- **Social tagging:** this feature is not modeled.
- **Discussions:** this feature is not modeled.
- **Backlinks:** this feature is not modeled.
- **Pages versioning:** a simple mechanism is provided for the versioning of the pages; a wiki page that has an older version should have a `hasPreviousVersion` property that points to another wiki page, and a `hasChangeDate` attribute that describes the change date.

An example of a wiki source page has been generated according to the WIF ontology specifications. The following page code is in XHTML format. This page describes, in a markup language, the content of the wiki page as it is visible to a web browser. The semantics of the page are linked with the `<link>` tag pointing to the `Digital_Enterprise_Research_Institute.rdf` file.

```
 <?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/
    xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
  <title>Digital Enterprise Research Institute</title>
  <meta http-equiv="Content-type" content=’text/html; charset="utf-8"’ />
  <link rel="alternate" type="application/rdf+xml" title="RDF Version" href
      ="Digital_Enterprise_Research_Institute.rdf" />
 </head>
 <body>
  <h1 class="firstHeading">Digital Enterprise Research Institute</h1>
  <p>The <b>Digital Enterprise Research Institute</b>, <b>DERI</b>, is an
      international network of independent research organisations aiming at
  developing... </p>
 </body>
</html>
```

Listing 3.7: A WIF XHTML page source about DERI

As we can see from the Listing 3.7 this is a simple XHTML page with the RDF content linked. The attached RDF file is displayed in the following Listing 3.8.

```
<!--file: Digital_Enterprise_Research_Institute.rdf-->

<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:foaf="http://xmlns.com/foaf/0.1/"
 xmlns:wif="http://wif.ontoware.org/2005/04#">

  <wif:Page rdf:about="http://www.wikiname.com/wiki/
      Digital_Enterprise_Research_Institute">
    <wif:hasName>DigitalEnterpriseResearchInstitute</wif:hasName>
    <wif:hasAuthor>
      <wif:WikiUser>
        <foaf:Person>
          <foaf:name>UserName</foaf:name>
        </foaf:Person>
        <wif:hasUserPage>
          <wif:Page rdf:about="http://www.wikiname.com/wiki/User:UserName"/>
        </wif:hasUserPage>
      </wif:WikiUser>
    </wif:hasAuthor>
    <wif:hasPreviousVersion>
      <wif:Page rdf:about="http://www.wikiname.com/wiki/
          Digital_Enterprise_Research_Institute_oldVers"/>
    </wif:hasPreviousVersion>
    <wif:hasChangeDate>2008-11-12T02:02:02</wif:hasChangeDate>
  </wif:Page>
</rdf:RDF>
```

Listing 3.8: A WIF RDF page source about DERI

The RDF showed above is a very simple example of how the WIF ontology could be used to model a generic wiki page. The namespaces included are only RDFS, FOAF, and of course WIF. The information expressed are about, in order, the page title, the article author, the previous revision and the date of creation of the current revision.

## 3.5 WikiOnt

WikiOnt[14] is an ontology for describing and exchanging wiki articles[16]. It has been proposed by Andreas Harth and Hannes Gassert from DERI Galway - Ireland in 2005, and it aims at integrating Wikipedia (and by extension other MediaWiki-based sites) into the Semantic Web framework, making Wikipedia machine-processable and -understandable. The WikiOnt ontology is a representation of Wikipedia and

[14] http://sw.deri.org/2005/04/wikipedia/wikiont.html

allows software programs to query the dataset and reuse the data. It has been made a conscious design choice to keep the ontology as compact as possible so that users new to the Semantic Web are able to quickly grasp the ontology and are able to reuse the data in own applications. The ontology comprises of classes and properties some of them reused from already existing ontologies such as Dublin Core and SKOS: they have been defined, in OWL, using the `wiki:` namespace at `http://sw.deri.org/2005/04/wikipedia/wikiont.owl`. Classes and properties in this ontology are summarized in the following Table 3.5.

| **WikiOnt** | **Ontology** |
|---|---|
| CLASSES | PROPERTIES |
| Article | contentType |
| Category | height |
| Image | width |
| Stub | contributor |
| | creator |
| | date |
| | title |
| | num_views |
| | narrower |
| | Image |
| | link |
| | internalLink |
| | externalLink |
| | redirect |
| | text |

Table 3.5: Classes and properties of the WikiOnt Ontology

- **Multi-authoring:** this feature is supported using the Dublin Core ontology, the included properties are: `dc:creator` regarding the person primarily responsible for making the resource, `dc:contributor` regarding the person responsible for making contributions to the resource, `dc:date` to associate a point or period of time with an event in the life-cycle of the page.
- **Categories:** a specific class, `Category`, has been created in order to define a category in the wiki's category graph, and it is a subclass of the main `Article` class that describes a wiki page.
- **Social tagging:** feature not modeled by the ontology.
- **Discussions:** feature not modeled by the ontology.
- **Backlinks:** feature not modeled by the ontology.
- **Pages versioning:** feature not modeled by the ontology.

In order to understand more clearly the modeling solutions described, it has been considered an example of a wiki source page generated according to the WikiOnt ontology specifications. The following page code is in XHTML format. This page describes, in a markup language, the content of the wiki page as it is visible to a web browser. The semantics of the page are linked with the `<link>` tag pointing to the `Digital_Enterprise_Research_Institute.n3` file.

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/
    xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```

```
  <head>
  <title>Digital Enterprise Research Institute</title>
  <meta http-equiv="Content-type" content=’text/html; charset="utf-8"’ />
  <link rel="alternate" type="text/n3" title="N3 Version" href="
      Digital_Enterprise_Research_Institute.n3" />
  </head>
  <body>
  <h1 class="firstHeading">Digital Enterprise Research Institute</h1>
  <p>The <b>Digital Enterprise Research Institute</b>, <b>DERI</b>, is an
      international network of independent research organisations aiming at
      developing...
  </p>
  </body>
</html>
```

Listing 3.9: A WikiOnt XHTML page source about DERI

As we can see from the Listing 3.9 this is a simple XHTML page with the RDF content linked. The attached RDF file is displayed in the following Listing 3.10. The RDF content is expressed in *Notation3* (N3) language, the compact and readable alternative to RDF’s XML syntax.

```
<!--file: Digital_Enterprise_Research_Institute.n3-->

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix wiki: <http://sw.deri.org/2005/04/wikipedia/wikiont.owl#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

<http://www.wikiname.com/wiki/Digital_Enterprise_Research_Institute>
  rdf:type wiki:Article ;
  dc:title "Digital Enterprise Research Institute" ;
  dc:creator "UserName" ;
  dc:date "2008-11-12T00:20:29-05:00" ;
  skos:subject <http://www.wikiname.com/wiki/Scientific_Organizations> ;
  wiki:image <http://www.wikiname.com/imgs/DERI_Logo.jpg> ;
  wiki:externalLink <http://www.deri.org/> ;
  wiki:internalLink <http://www.wikiname.com/wiki/Semantic_Web> ;
  wiki:text "This article is about the Digital Enterprise Research Institute
      , DERI..." .
```

```
<http://www.wikiname.com/imgs/DERI_Logo.jpg> wiki:contentType "image/jpg" .
```

Listing 3.10: A WikiOnt RDF (N3) page source about DERI

The RDF showed above is a very simple example of how the WikiOnt ontology can be used to model a generic wiki page. The namespaces included with the `@prefix` instructions are `RDF`, Dublin Core (`DC`), `SKOS`, and of course `WIKI` for the WikiOnt ontology. The information expressed are modeled in the same way described in the previous part of this section. After all the typical document details (i.e. type, title, creator and date) it is important to note the expression of a concept related to the DERI subject: `Scientific Organizations` is a `skos:Concept` related to the topic. Subsequently, internal and external links, embedded images and the main article text, are described using the WikiOnt ontology properties.

## 3.6 SIOC

The SIOC initiative has been already described in the first chapter. In this section it has been considered only the wiki-related aspects of the ontology in order to understand its possibilities and limits before my extension proposal. This proposal has been done after this analysis and after the comparison between SIOC and all the other models exposed before in this chapter.

SIOC has a very large set of properties and classes, and it is commonly used in conjunction with the FOAF vocabulary, for expressing personal profile and social networking information, together with the Dublin Core vocabulary, for expressing documents details. The SIOC OWL ontology is divided in modules, the two most important are: the Core ontology[15], and the Types module[16]. Seeing that the

[15] `http://rdfs.org/sioc/ns`

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

vocabulary is more extensive than the previous ones and that it is not only concentrated on wikis, it has been decided to list only the necessary classes and properties as summarized in Table 3.6. In the table it has been included, not only the SIOC objects considered, but also other useful external properties from different vocabularies.

| **SIOC Core Ontology** | **SIOC Types Ontology** |
|---|---|
| CLASSES:<br>Forum<br>User<br>Usergroup | CLASSES:<br>Wiki<br>WikiArticle<br>Category |
| PROPERTIES:<br>topic<br>has_creator<br>has_container<br>has_discussion<br>previous_version<br>next_version<br>links_to | |

| **External ontologies** |
|---|
| dc:creator |
| dc:contributor |
| dc:date |
| dc:subject |
| foaf:OnlineAccount |

Table 3.6: Wiki-related classes and properties of the SIOC Ontology

Classes and properties listed above could be used to model wikis in the way described in the following points. For each wiki feature a possible implementation model is exposed, focusing also on the needed improvement.

- **Multi-authoring:** this feature may be modeled using the class `sioc:User` that describes a user account in an online community site and it is a `foaf:OnlineAccount` subclass. Then it is possible to reuse properties from the Dublin Core ontology as WikiOnt does, i.e. `dc:creator`, `dc:contributor` and `dc:date`.

- **Categories:** there is a `Category` class in the SIOC Types module, so this is the best candidate to accomplish to this feature. This class should be used as object of the `sioc:topic` property. Unfortunately taxonomic structures are not modeled by the ontology, so that it is not possible to define a category in a wiki's category graph. This `Category` class might be subclass of a `skos:Concept` to be able to use the SKOS vocabulary capabilities.

- **Social tagging:** thanks to a triangulation between `sioc:topic`, `dc:subject` and an external tagging ontology such as the *Tag ontology* [17] or SCOT[18] and MOAT[19], it is possible to link tags to ontology concepts and provide them meaning.

[17] `http://www.holygoat.co.uk/owl/redwood/0.1/tags/`

[18] `http://scot-project.org/`

[19] `http://moat-project.org/`

- **Discussions:** the SIOC main class responsible for the modeling of discussions is the `sioc:Forum` class, but there could be more specific Forum subclasses that are more suitable for the discussion purposes. It is only necessary to identify a proper attribute capable of linking a wiki page to a `sioc:Forum`.

- **Backlinks:** feature not explicitly modeled by the ontology, but a `sioc:links_to` property could be used in this case.

- **Pages versioning:** feature not completely modeled by the ontology. Actually there are properties such as `sioc:previous_version` and `sioc:next_version` which can express a sequential relation between revisions. This should be combined with a Dublin Core property such as `dcterms:created` (or `dcterms:modified`) to express the date/time of creation (or modification) of the page version. For a fine-grained modeling there are several other ways to accomplish to the versioning problem but this will be examined later, in the next chapters, where the extension proposals will be exposed.

In order to understand more clearly the modeling solutions described, it has been considered an example of a wiki source page generated according to the SIOC ontology specifications. The following page code is in RDF language.

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:dcterms="http://purl.org/dc/terms/"
 xmlns:sioc="http://rdfs.org/sioc/ns#">
  <rdf:Description rdf:about="http://www.wikiname.com/wiki/
      Digital_Enterprise_Research_Institute">
    <rdf:type rdf:resource="http://rdfs.org/sioc/types#WikiArticle"/>
    <dc:title>Digital Enterprise Research Institute</dc:title>
    <dc:creator>UserName</dc:creator>
    <dcterms:created>2008-10-03 22:50:32</dcterms:created>
    <sioc:has_container rdf:resource="http://www.wikiname.com/"/>
    <sioc:description>Wiki page about Digital Enterprise Research Institute,
         DERI</sioc:description>
    <sioc:has_creator rdf:resource="http://www.wikiname.com/user/UserName"/>
    <sioc:links_to rdf:resource="http://www.deri.org"/>
    <rdfs:label>Digital_Enterprise_Research_Institute</rdfs:label>
    <sioc:topic rdf:resource="http://example.org/skos/concepts#SemanticWeb
        "/>
    <sioc:topic rdf:resource="http://example.org/skos/concepts#
        ScientificOrganization"/>
    <rdfs:seeAlso rdf:resource="http://www.wikiname.com/wiki/Semantic_Web"/>
    <rdfs:seeAlso rdf:resource="http://www.wikiname.com/wiki/
        Scientific_Organization"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.wikiname.com/">
    <sioc:container_of rdf:resource="http://www.wikiname.com/wiki/
        Digital_Enterprise_Research_Institute"/>
    <rdf:type rdf:resource="http://rdfs.org/sioc/types#Wiki"/>
    <rdfs:label>WikiName</rdfs:label>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.wikiname.com/user/UserName">
    <rdf:type rdf:resource="http://rdfs.org/sioc/ns#User"/>
    <rdfs:label>UserName</rdfs:label>
  </rdf:Description>
</rdf:RDF>
```

Listing 3.11: A RDF page source about DERI with SIOC data

As we can see from the Listing 3.11 the RDF showed above is a very simple example of how the SIOC ontology can be used to model a generic wiki page. The other external vocabularies included are `RDF/S` and Dublin Core (`DC` and `DCTERMS`). The information expressed are modeled in the same way described in the previous part of this section.

After all the typical document details (i.e. type, title, creator and date) some characteristics are important to note. The wiki `http://www.wikiname.com` is defined as the `sioc:Container` (and the `sioct:Wiki`) of the `WikiArticle` about *DERI*. The article's author is both a `sioc:User` and a `dc:creator`, so to simultaneously have the expression of the user's URI and his literal "UserName". Related concepts to the main subject are expressed as objects of the `sioc:topic` property, and the corresponding page URIs are linked with `rdfs:seeAlso` properties (providing further useful RDF data).

Finally, regarding the use of SIOC for wikis, we can mention UfoWiki [24], that have been deployed in complement or other SIOC-related data in an Enterprise 2.0 platform, while it does not support versioning in its RDF representation.

## 3.7 Comparison of existing approaches

Based on the previous analysis, it has been produced a comparison matrix, to underline the pros and cons of each approach. The following Table 3.7 includes one SIOC column, that considers the SIOC Ontology before the proposed extension. A table considering also the the improvements aimed with the extension is illustrated in the next chapter. The column exposed here agrees with the details described in the previous Paragraph 3.6 about SIOC.

| | Multi-auth. | Categ. | Social tag | Discuss | Backlink | Version. |
|---|---|---|---|---|---|---|
| Sweetwiki | **yes** | **yes** | **yes** | no | **yes** | **yes** |
| IkeWiki | **yes** | **yes*** | no | **yes*** | no | no |
| S.M.W. | **yes** | **yes** | no | no | no | no |
| WIF | **yes** | no | no | no | no | **yes** |
| WikiOnt | **yes*** | **yes** | no | no | no | no |
| SIOC | **yes*** | no | **yes*** | **yes** | no | no |

*(yes* = yes, using external ontologies)*

Table 3.7: Comparing various ontologies to represent wikis structure

A *"yes*"* mark has been put in the table where a feature is reached using an external ontology. As described before (to summarize):
– IkeWiki uses DC, SKOS, FOAF, and SIOC for Categories, Social Tagging and Discussions;
– WikiOnt uses DC for Multi-authoring;
– SIOC uses DC, FOAF and Tag Ontology for Multi-authoring and Tagging.

After the comparative analysis it is possible to 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 pages 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.

Therefore the most complete model to take into account is SweetWiki, being able to accomplish to every requirement but the discussions.
Moreover, it is important to remark that some features that are not modeled by these vocabularies could be added by external vocabularies, for instance social tagging.

# Part II

# Our Contribution

# Chapter 4

# Extending the SIOC ontology

This chapter describes how the SIOC ontology has been extended to take into account particular aspects of wikis in order to enable integration capabilities between various wiki systems. In particular, the proposed extensions and the chosen modeling solutions will be overviewed.

## 4.1 Motivations

The SIOC Ontology – Semantically-Interlinked Online Communities [12] – is now considered as one of the building blocks of the *"Social Semantic Web"*. More than 50 applications are currently using SIOC[1], 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[2]. Moreover, the use of SIOC goes further than popular and mainstream Web 2.0 services, from Enterprise 2.0 information integration[3] to Health Care and Life Sciences discourse

[1] http://sioc-project.org/applications

[2] http://developer.yahoo.com/searchmonkey/

[3] http://www.w3.org/2001/sw/sweo/public/UseCases/EDF/

representation[4].

Wikis are one of the most important examples of collaborative and social platforms, and while the SIOC Types[5] 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 ontology nor in its modules. Yet, providing wikis information using SIOC would have several advantages in terms of integration with existing and constantly dynamically-created 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 than on a WordPress blog[6]. Hence, I 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.

## 4.2 Modeling 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.

This feature could be modeled in several ways. Taking into account other se-

[4] `http://esw.w3.org/topic/HCLSIG/SWANSIOC`

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

[6] `http://wordpress.org/`

mantic wikis, described in Chapter 3, it has been taken inspiration from all the different existing models. Then I defined my own different model to keep the pros of each model, also because I did not find one capable to satisfy completely my needs. An important requirement taken into account is the fast and simple browsing capability that the model should have (i.e. using proper applications to browse it as a complete graph it should be easy to use). For this reason it has been chosen to use transitive properties to express the temporal relation between revisions of a wiki page, but this will be better examined later in this section.

### 4.2.1 Comparison of different approaches

In order to define an effective versioning model I compared existing models and proposed several variations on these approaches. The examined solutions are described and compared in this section.

The first very simple model is displayed in Fig 4.1. I have called this approach "Versioning with numbers" and, as the name suggests, it uses sequential numbers to identify the temporal relations and the order between revisions.

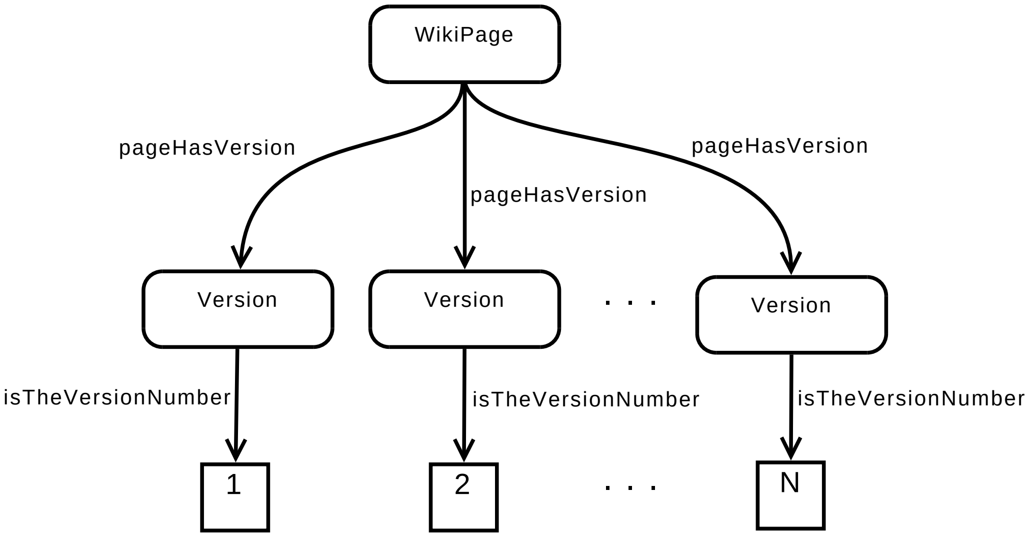


Figure 4.1: Versioning model with numbers.

This approach to pages versioning is very similar to the SweetWiki one, and it is based on the main concept that it is possible to know if a version is newer or older than another one just by comparing the version number. The more the number is greater the more the version is newer. This solution is very simple and easy to use, but it needs on the background a specific human knowledge (i.e. if the number increases then the version is more recent), so it does not agree completely with the Semantic Web principles. For instance with this modeling solution you cannot do inference easily by using included OWL axioms such as transitivity while reasoning over numbers should imply the use of a dedicated rule language. The following model resolves this particular issue.

The second model considered is even simpler than the previous one, and it is displayed in Fig 4.2. It provides only the *previous* and *next* properties, so that each version has a pointer to the previous/next one.

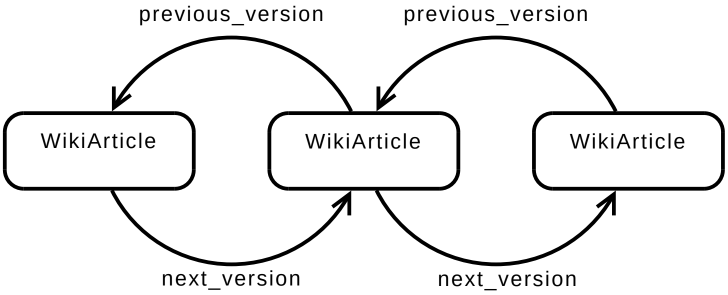


Figure 4.2: "Previous/Next" versioning model.

In regard to this modeling solution it is obvious to say that the biggest limit is the complexity, not of the model itself, but of the browsing and querying capabilities it provides. For instance, for the common use-case such as requesting the latest version of the article starting from an older revision, the only solution is to look for the newest version "jumping" with queries on each version temporally located between the older one and the newest one. This issue could be overcome with the introduction of the following versioning model, as in Fig 4.3.

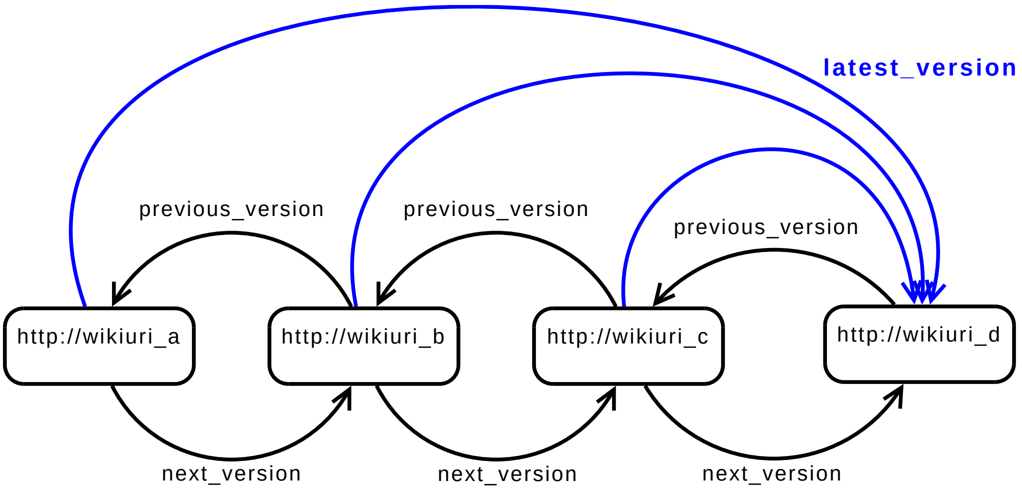


Figure 4.3: "Previous/Next with Latest" versioning model.

The model I called "Previous/Next with Latest" is just a little improvement of the previous examined model. It adds a `latest_version` property which points always to the newest revision of the `WikiArticle`. The introduction of this property is a simple but very useful improvement, since the most common use-case is to request the latest version of a page.
The emerging issue with the introduction of this capability is what has been called the *scalability* problem. In fact if the latest page version is always identified with a different URI, it is necessary to face the issue of updating all the earlier versions pointing to the latest one as soon as a new revision is created. Especially with a large number of revisions, as in the wiki case, it could affect considerably the scalability of the system and its performances. The solution of this problem is the use of an *alias* name/URI for the latest version, as common in most of the wikis. Hence it is possible to use the same URI for the newest revision (e.g. `http://wiki.org/ArticleName`) and for all the old revisions a distinct resource identifier (e.g. `http://wiki.org/ArticleName_versXYZ`). By doing so it is no more necessary to change the `latest_version` pointers, as soon as a

new version is created, because they link always to the same URI.

A drawback in the general "Previous/Next" model is still the browsing capability offered. Because, if the *latest version* use-case is resolved, still it has the limitation for a request of a particular version of the article starting from one another that is temporally distant. The only way to resolve this query is, again, to look for the needed version "jumping" with queries on each version temporally located between the starting one and the matching one. In other words it is necessary to repeat the same query on every version in between, looking for the previous/next version and verifying if it satisfies the request. The solution is given by using *transitive properties*.

OWL (the W3C Web Ontology Language)[7] provides this logical characteristic of properties, as stated in the official reference specifications[5]. According to the specifications: "When one defines a property P to be a transitive property, this means that if a pair (x,y) is an instance of P, and the pair (y,z) is also instance of P, then we can infer the pair (x,z) is also an instance of P.
Syntactically, a property is defined as being transitive by making it an instance of the built-in OWL class `owl:TransitiveProperty`, which is defined as a subclass of `owl:ObjectProperty`.
Typical examples of transitive properties are properties representing certain part-whole relations. For example, we might want to say that the `subRegionOf` property between regions is transitive:

```
<owl:TransitiveProperty rdf:ID="subRegionOf">
  <rdfs:domain rdf:resource="#Region"/>
  <rdfs:range  rdf:resource="#Region"/>
</owl:TransitiveProperty>
```

Listing 4.1: OWL transitive property

From this an OWL reasoner should be able to derive that if `ChiantiClassico`,

[7] `http://www.w3.org/TR/owl-ref/`

Tuscany and Italy are regions, and ChiantiClassico is a subregion of Tuscany, and Tuscany is a subregion of Italy, then ChiantiClassico is also a subregion of Italy.

Note that because owl:TransitiveProperty is a subclass of owl:ObjectProperty, the following syntactic variant is equivalent to the example above:

```
<owl:ObjectProperty rdf:ID="subRegionOf">
  <rdf:type rdf:resource="&owl;TransitiveProperty"/>
  <rdfs:domain rdf:resource="#Region"/>
  <rdfs:range  rdf:resource="#Region"/>
</owl:ObjectProperty>
```

Listing 4.2: OWL transitive property

To summarize referring to the versioning context, by introducing transitivity on previous/next_version properties, it is possible to a reasoner to infer all the previous/next versions of a wiki article simply querying one article version. The model based on this concept is the one that has been chosen to implement on SIOC, and it is explained with more details in the next section 4.2.2 **Our model**, see also Fig 4.5.

The last modeling approach considered is a "Event-based" versioning solution. It is different from the other models exposed before because it is focused on the events that create or modify the wiki versions. In Fig 4.4 a diagram of the model is displayed.

It is based on the assumption that a WikiArticle has been created by a user, so every page is linked with a created property to the information on the CreationEvent such as author, date/time, etc. Furthermore, the same WikiArticle could have been modified by someone, giving origin to a new version of this article. At the same time the new originated article is associated to a CreationEvent. The ModificationEvent has related information properties as well as the creation one. When a particular version is cancelled a

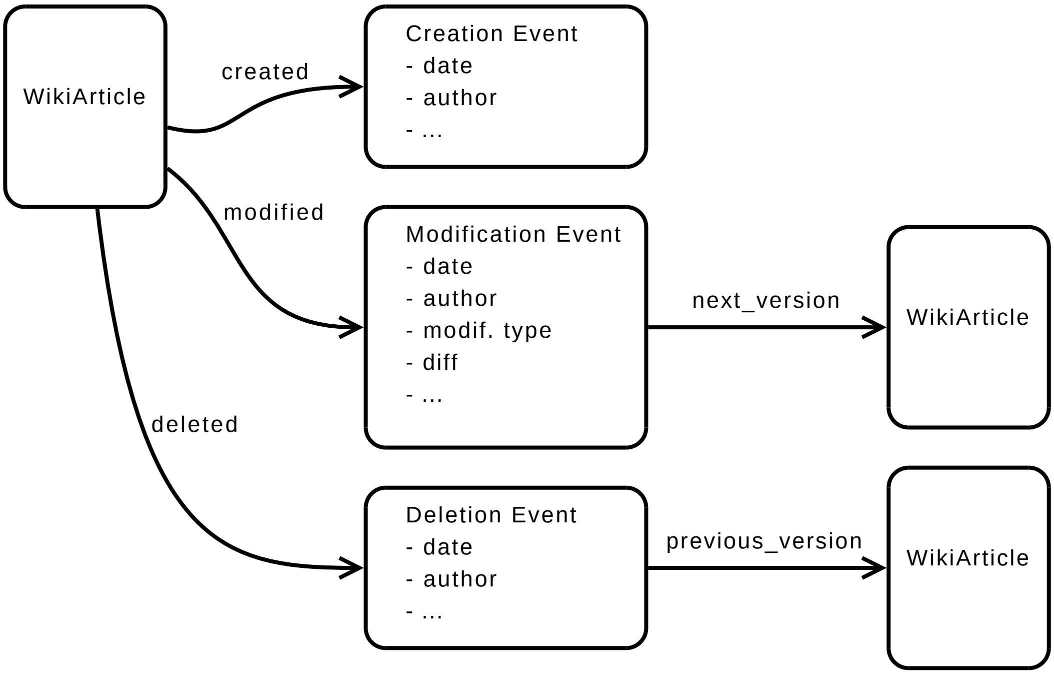


Figure 4.4: "Event-based" versioning model.

DeletionEvent is associated with the corresponding deleted property, and the cancelled version now should point to his previous version if it exists.

After an overview on the considered versioning approaches, it has been decided to make a comparison in order to select the most appropriate one. The first considered aspects were: *scalability*, *Semantic Web philosophy* compliance and *structure simplicity*. As detailed before in each model description, the results of this first comparison are showed in the following Table 4.1.

Another comparison matrix is displayed in Table 4.2, and it evaluates the models regarding their *browsing speed* and *querying complexity*. Each evaluation characteristic is divided in two cases, depending on the type of query. Two different queries has been defined, or two different use-cases, to underline more precisely the pros and cons of each versioning approach. The two selected queries are:

| | Scalability | SemWeb philosophy | Simple Structure |
|---|---|---|---|
| w/ Numbers | $+$ | $-$ | $+/-$ |
| Previous/Next | $+$ | $+$ | $+$ |
| Prev/Next + Latest + No Alias | $-$ | $+$ | $+$ |
| Prev/Next + Latest + Alias | $+$ | $+$ | $+$ |
| Prev/Next+Latest+Alias+Trans. | $+$ | $+$ | $+/-$ |
| Event based | $+$ | $+/-$ | $-$ |

Table 4.1: Versioning models comparison

- *Query 1:* Select the latest version of a `WikiArticle` starting from an older one.
- *Query 2:* Select all the earlier (older) versions until a specified date.

| | Browsing speed | | Querying complexity | |
|---|---|---|---|---|
| | Query 1 | Query 2 | Query 1 | Query 2 |
| w/ Numbers | + | + | + | + |
| Previous/Next | - | - | +/- | +/- |
| Prev/Next + Latest + No Alias | + | - | + | +/- |
| Prev/Next + Latest + Alias | + | - | + | +/- |
| Prev/Next + Latest + Alias + Transitivity | + | + | + | + |
| Event based | - | - | - | - |

Table 4.2: Versioning models comparison

Analyzing the two comparison matrixes it is possible to state that, as regards browsing and querying performances, the two models performing positively under all the points-of-view are: *"Versioning with numbers"* and *"Previous/Next with Latest and Alias and Transitivity"*. Then comparing these two models in the previous table we note that the *"Versioning with numbers"* one loses points in the "Semantic Web philosophy" feature. Hence the best model to take into account is the *"Previous/Next with Latest and Alias and Transitivity"*, which is the one implemented on SIOC and that will be detailed in next section.

### 4.2.2 Our model

A diagram explaining the model which has been chosen to implement is displayed in Fig. 4.5 and all the used properties are now defined in the SIOC Core ontology. It is necessary to state that the showed schema in Fig. 4.5 is an example of what can be done with the model, not a diagram of the complete model itself, but it has been created to clarify how the model works.

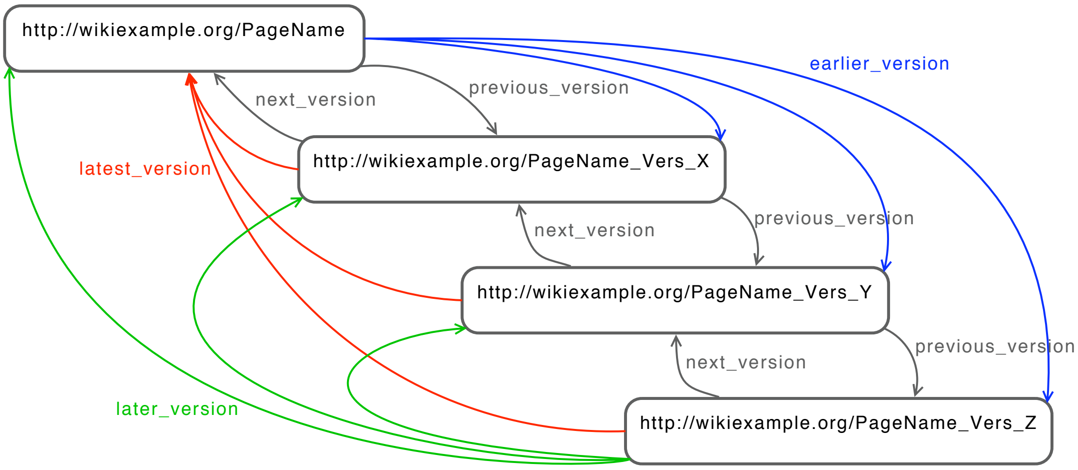

http://wikiexample.org/PageName
earlier_version
next_version
previous_version
latest_version
http://wikiexample.org/PageName_Vers_X
next_version
previous_version
http://wikiexample.org/PageName_Vers_Y
previous_version
next_version
later_version
http://wikiexample.org/PageName_Vers_Z

Figure 4.5: Pages versioning model with SIOC properties.

In the diagram above please note that transitive properties `earlier_version` and `later_version` are only displayed for two wiki articles: the latest one and the first one, and not for all the intermediate versions as it should be.

The `previous/next_version` property links only the previous/next revision of a `sioc:Item`. Meanwhile `earlier/later_version` are transitive properties and super-properties of `previous/next_version`. The main advantage of the definition of the two transitive properties as super-properties of the already existing `next/previous_version` is that they can be inferred automatically by a reasoner. Hence, modeling a wiki article (or a `sioc:Item` in general), it is only necessary to describe the previous and the next revision. This can also be convenient during the querying process (as described in Chapter 6): 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.

To summarize, my work in the versioning area has been to add two transitive properties: `sioc:earlier_version` and `sioc:later_version`, with `sioc:Item` as domain and range; then to define them as super-properties of `sioc:previous_version` and `sioc:next_version` respectively. The OWL definition of these properties is as follows.

```
<owl:TransitiveProperty rdf:about="http://rdfs.org/sioc/ns#earlier_version">
  <rdfs:comment xml:lang="en">Links to a previous (older) revision of this
      Item or Post.</rdfs:comment>
  <rdfs:domain rdf:resource="http://rdfs.org/sioc/ns#Item"/>
  <rdfs:isDefinedBy rdf:resource="http://rdfs.org/sioc/ns#"/>
  <rdfs:label xml:lang="en">earlier_version</rdfs:label>
  <rdfs:range rdf:resource="http://rdfs.org/sioc/ns#Item"/>
  <owl:inverseOf rdf:resource="http://rdfs.org/sioc/ns#later_version"/>
</owl:TransitiveProperty>
[...]
<owl:TransitiveProperty rdf:about="http://rdfs.org/sioc/ns#later_version">
  <rdfs:comment xml:lang="en">Links to a later (newer) revision of this Item
       or Post.</rdfs:comment>
  <rdfs:domain rdf:resource="http://rdfs.org/sioc/ns#Item"/>
  <rdfs:isDefinedBy rdf:resource="http://rdfs.org/sioc/ns#"/>
  <rdfs:label xml:lang="en">later_version</rdfs:label>
  <rdfs:range rdf:resource="http://rdfs.org/sioc/ns#Item"/>
  <owl:inverseOf rdf:resource="http://rdfs.org/sioc/ns#earlier_version"/>
</owl:TransitiveProperty>
```

Listing 4.3: SIOC earlier/later_version properties

Then it has been added the sub-property instructions to the `sioc:previous/next_version` properties, adding the following lines:

```
<owl:ObjectProperty rdf:about="http://rdfs.org/sioc/ns#previous_version">
   [...]
   <rdfs:subPropertyOf rdf:resource="http://rdfs.org/sioc/ns#earlier_version
       "/>
</owl:ObjectProperty>
[...]
<owl:ObjectProperty rdf:about="http://rdfs.org/sioc/ns#next_version">
   [...]
   <rdfs:subPropertyOf rdf:resource="http://rdfs.org/sioc/ns#later_version
       "/>
</owl:ObjectProperty>
```

Listing 4.4: SIOC previous/next_version properties

Furthermore it has been introduced another new property: the `sioc:latest_version` attribute, having a `sioc:Item` as domain and range as well as all the other versioning properties. The OWL code is the same as in the following lines:

```
<owl:ObjectProperty rdf:about="http://rdfs.org/sioc/ns#latest_version">
  <rdfs:comment xml:lang="en">Links to the latest revision of this Item or
      Post.</rdfs:comment>
  <rdfs:domain rdf:resource="http://rdfs.org/sioc/ns#Item"/>
  <rdfs:isDefinedBy rdf:resource="http://rdfs.org/sioc/ns#"/>
  <rdfs:label xml:lang="en">latest_version</rdfs:label>
  <rdfs:range rdf:resource="http://rdfs.org/sioc/ns#Item"/>
</owl:ObjectProperty>
```

Listing 4.5: SIOC latest_version property

A summary of the extension changes is described at the end of this chapter.

## 4.3 Representing 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 maintain the original taxonomy structure. In this regard an appropriate solution is provided by the SKOS[8] vocabulary [22], as it provides a way to model hierarchical structures between various categories, represented as instances of `skos:Concept`.

In regards of 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, it has been 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. The OWL declaration of this class is as follows:

```
<owl:Class rdf:about="http://rdfs.org/sioc/types#Category">
  <rdfs:label xml:lang="en">Category</rdfs:label>
  <rdfs:comment xml:lang="en">
    Category is used on the object of sioc:topic to indicate that this
        resource is a category on a site.
  </rdfs:comment>
  <rdfs:isDefinedBy rdf:resource="http://rdfs.org/sioc/types#"/>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2008/05/skos#Concept"/>
</owl:Class>
```

Listing 4.6: SIOC Types Category class

Moreover it is important to note that thanks to the `sioc:topic` property, it is possible to link any wiki page to such category.

[8] http://www.w3.org/2004/02/skos/

### 4.3.1 SKOS

SKOS (Simple Knowledge Organization System)[9], currently developed within the W3C framework, is a model for expressing knowledge organization systems. In particular for the representation of thesauri, classification schemes, taxonomies, subject-heading systems, terminologies, glossaries, or any other type of structured controlled vocabulary. The SKOS data model is formally defined in the specifications[10] as an OWL Full ontology, and SKOS data are expressed as RDF triples. It is designed as a modular and extensible family of languages, in a way that its use and implementation should be as simple as possible.

SKOS is divided into three main components:

- **SKOS Core**, defines the classes and properties sufficient to represent the common features found for example in a standard thesaurus.
- **SKOS Mapping**, defines a vocabulary to express matching (exact or fuzzy) of concepts from one concept scheme to another.
- **SKOS Extensions**, provides ways to declare relationships between concepts with more specific semantics than the simple "broader-narrower" relationship, such as class-instance or partitive relationships.

The SKOS data model views a knowledge organization system as a concept scheme comprising a set of concepts. These SKOS concept schemes and SKOS concepts are identified by URIs, enabling anyone to refer to them unambiguously from any context, and making them a part of the World Wide Web. SKOS concepts can be labeled with any number of lexical strings, in any given natural language. One of these labels in any given language can be indicated as the "preferred" label for that language, the others as "alternate" labels, and they may also be "hidden".

[9] http://www.w3.org/2004/02/skos/

[10] http://www.w3.org/TR/2009/CR-skos-reference-20090317/

Let's see now some example on how to use the SKOS Core ontology. These examples are taken from the SKOS Reference specification[11]. Examples of RDF graphs are given using the Turtle RDF language. All examples assume that they are preceded by the following prefix and URI base directives:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@base <http://example.org/ns/> .
```

The class `skos:Concept` is the class of SKOS concepts. A SKOS concept can be viewed as an idea or notion; a unit of thought. However, what constitutes a "unit of thought" is subjective, and this definition is meant to be suggestive, rather than restrictive. For example, in the graph below, `<MyConcept>` is an instance of `skos:Concept` (`skos:Concept` is also an instance of `owl:Class`).

```
<MyConcept> rdf:type skos:Concept .
```

A SKOS concept scheme can be viewed as an aggregation of one or more SKOS concepts. Semantic relationships (links) between those concepts may also be viewed as part of a concept scheme. The graph below describes a concept scheme with two SKOS concepts, one of which is a top-level concept in that scheme.

```
<MyScheme> rdf:type skos:ConceptScheme ;
  skos:hasTopConcept <MyConcept> .

<MyConcept> skos:topConceptOf <MyScheme> .

<AnotherConcept> skos:inScheme <MyScheme> .
```

The property `skos:hasTopConcept` is, by convention, used to link a concept scheme to the SKOS concept(s) which are topmost in the hierarchical relations for that scheme.

[11] http://www.w3.org/TR/2009/CR-skos-reference-20090317/

As stated before, a lexical label is a string of characters in a given natural language. SKOS enables a distinction to be made between the "preferred", "alternate" and "hidden" lexical labels for any given resource. The "preferred" and "alternate" labels are useful when generating or creating human-readable representations of a knowledge organization system. While a "hidden" label is a lexical label for a resource, where it is preferable for a character string to be accessible to applications performing text-based indexing and search operations, but not to be visible otherwise. For example[12]:

```
ex:animals rdf:type skos:Concept;
   skos:prefLabel "animaux"@fr;
   skos:altLabel "bêtes"@fr;
   skos:hiddenLabel "betes"@fr.
```

The last SKOS characteristic to be analyzed is semantic relations, because they play a crucial role for defining concepts. The meaning of a concept is defined not just by the natural-language words in its labels but also by its links to other concepts in the vocabulary. SKOS supplies three standard properties: `skos:broader` and `skos:narrower` which enable the representation of hierarchical links (such as the relationship between one genre and its more specific species), and `skos:related` which enables the representation of associative (non-hierarchical) links (such as the relationship between one type of event and a category of entities which typically participate in it). To assert that one concept is broader in meaning (i.e. more general) than another, the `skos:broader` property is used, the `skos:narrower` property is used to assert the inverse. As in the following example using all the three properties:

```
ex:animals rdf:type skos:Concept;
   skos:prefLabel "animals"@en;
   skos:narrower ex:mammals.
ex:mammals rdf:type skos:Concept;
```

[12] next examples are taken from the SKOS Primer: `http://www.w3.org/TR/2009/WD-skos-primer-20090317/`

```
   skos:prefLabel "mammals"@en;
   skos:broader ex:animals.

ex:birds rdf:type skos:Concept;
   skos:prefLabel "birds"@en;
   skos:related ex:ornithology.
ex:ornithology rdf:type skos:Concept;
   skos:prefLabel "ornithology"@en.
```

## 4.4 Integrating discussion pages

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[13]). 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 more specific Forum subclasses that are more suitable 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 is necessary to identify a proper attribute capable of linking a wiki page to its discussion page. In this regard it has been decided to add a `sioc:has_discussion` property to the SIOC Core ontology, with domain `sioc:Item` and open range.

```
<owl:ObjectProperty rdf:about="http://rdfs.org/sioc/ns#has_discussion">
  <rdfs:comment xml:lang="en">The discussion that is related to this Item.</
      rdfs:comment>
  <rdfs:domain rdf:resource="http://rdfs.org/sioc/ns#Item"/>
  <rdfs:isDefinedBy rdf:resource="http://rdfs.org/sioc/ns#"/>
```

[13] `http://en.wikipedia.org/wiki/Wikipedia:Talk_page_guidelines`

```
  <rdfs:label xml:lang="en">has_discussion</rdfs:label>
</owl:ObjectProperty>
```

Listing 4.7: SIOC has_discussion property

This choice has been done in order to make this property reusable also in other contexts and not only in the wiki's one, 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 discussion or threaded ones, and that features also allows us to re-use advanced SIOC-based argumentative discussion modeling as defined in [20].

## 4.5 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 usually a list of backlinks related to a page is provided by clicking on a link called *"What links here"*. Backlinks of a webpage may be of significant interest: they indicate who is paying attention to that page. It is also important to notice that backlinks have not an inverse relation with internal links.

This feature may be modeled using the already existing property `sioc:links_to`. This property identifies links extracted from hyperlinks within a SIOC concept and it is a `dcterms:references` sub-property. It is very 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, as detailed later:

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

Listing 4.8: Representing backlinks

## 4.6 Changes summary and comparison

To summarize all the changes made to the SIOC Core ontology and the SIOC Types module, a brief list of the proposed extensions has been created, 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; useful to define the structure of the entire revisions chain.
- 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`.

Furthermore, based on the previous analysis made in Chapter 3, it has been produced a comparison matrix, to underline the pros and cons of each semantic wiki approach examined, and to show how powerful could be the extension proposal made. The following Table 4.3 includes two SIOC columns, the first that considers the SIOC Ontology before the proposed extension, and the second one considering also the improvements aimed with the extension. The former agrees with the details described in the previous Paragraph 3.6 about SIOC. The latter with the extension described in this chapter.

| | Multi-auth. | Categ. | Social tag | Discuss | Backlink | Version. |
|---|---|---|---|---|---|---|
| Sweetwiki | **yes** | **yes** | **yes** | no | **yes** | **yes** |
| IkeWiki | **yes** | **yes*** | no | **yes*** | no | no |
| S.M.W. | **yes** | **yes** | no | no | no | no |
| WIF | **yes** | no | no | no | no | **yes** |
| WikiOnt | **yes*** | **yes** | no | no | no | no |
| SIOC | **yes*** | no | **yes*** | **yes** | no | no |
| SIOC+ext. | **yes** | **yes** | **yes** | **yes** | **yes** | **yes** |

*(yes* = yes, using external ontologies)*

Table 4.3: Comparing various ontologies to represent wikis structure

A *"yes*"* mark has been put in the table where a feature is reached using an external ontology. As described before (to summarize):

– IkeWiki uses DC, SKOS, FOAF, and SIOC for Categories, Social Tagging and Discussions;

– WikiOnt uses DC for Multi-authoring;

– SIOC uses DC, FOAF and Tag Ontology for Multi-authoring and Tagging.

As shown in the table the extension proposal made accomplishes with all the requirements spotted and explored in the previous sections.

# Chapter 5

# Applications

In order to see implications of the proposed extension, my aim was to build an exporter from a popular wiki platform, so that it can expose its data in RDF using SIOC. Then, it has been decided to create two types of applications, a web-service and a standalone application, to export any MediaWiki instance. MediaWiki[1] is one of the most popular wiki platform, hosting all the Wikimedia Fundation wikis (i.e. Wikipedia, Wiktionnary, etc.) and propulsing more that 25 millions wiki articles from different platforms[2]. So a Java application and a PHP webservice have been developed, both to export SIOC data from MediaWiki's wikis.

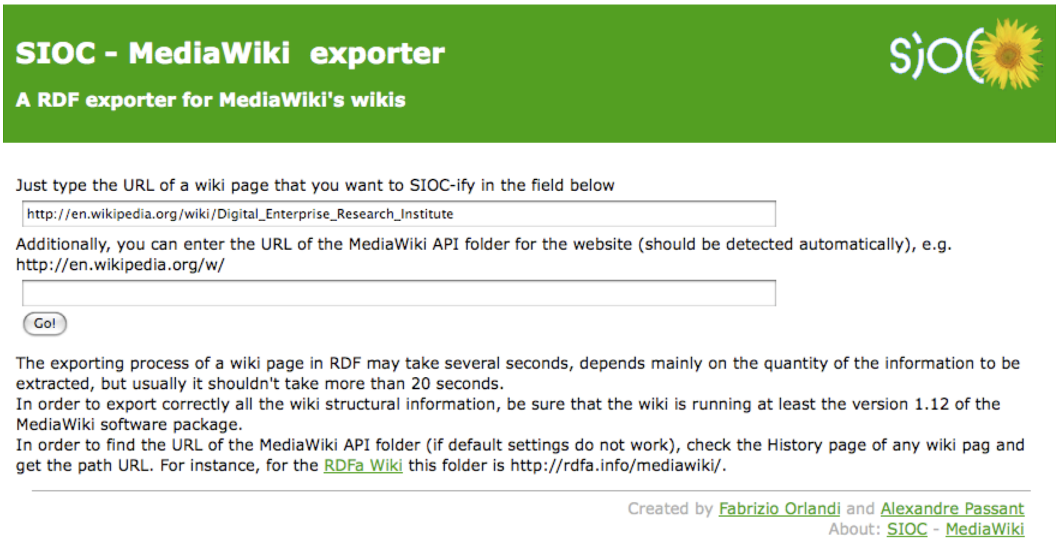
## 5.1 PHP Exporter

### 5.1.1 MediaWiki webservice

In order to export SIOC data from MediaWiki's wikis it has been implemented a webservice, written in PHP, that exports a wiki article in RDF with the structure explained in previous sections. The webservice is publicly available at `http:`

[1] `http://www.mediawiki.org`

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



SIOC - MediaWiki exporter

A RDF exporter for MediaWiki's wikis

Just type the URL of a wiki page that you want to SIOC-ify in the field below

http://en.wikipedia.org/wiki/Digital_Enterprise_Research_Institute

Additionally, you can enter the URL of the MediaWiki API folder for the website (should be detected automatically), e.g. http://en.wikipedia.org/w/

The exporting process of a wiki page in RDF may take several seconds, depends mainly on the quantity of the information to be extracted, but usually it shouldn't take more than 20 seconds.
In order to export correctly all the wiki structural information, be sure that the wiki is running at least the version 1.12 of the MediaWiki software package.
In order to find the URL of the MediaWiki API folder (if default settings do not work), check the History page of any wiki pag and get the path URL. For instance, for the RDFa Wiki this folder is http://rdfa.info/mediawiki/.

Created by Fabrizio Orlandi and Alexandre Passant
About: SIOC - MediaWiki

Figure 5.1: Interface of the SIOC MediaWiki webservice

`//ws.sioc-project.org/mediawiki/`.

The MediaWiki exporter is relatively lightweight and built thanks to two PHP classes: the SIOC Mediawiki exporter itself and the already existing SIOC API[3] (Application Programming Interface), that I improved in order to take the new characteristics of the model into account. My approach combines the use of the MediaWiki API as well as the SIOC PHP API – that was 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 (represented in Fig. 5.2):

- 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;

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

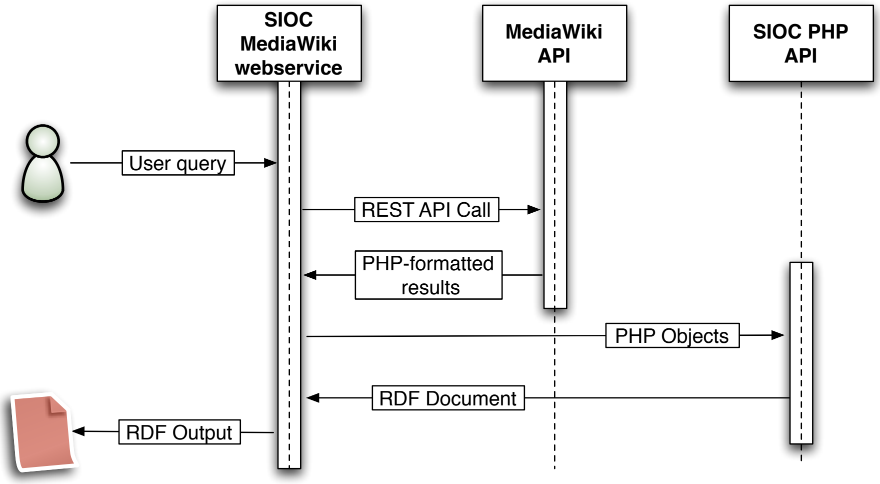


Figure 5.2: Activity diagram of the SIOC MediaWiki webservice

- it parses the results of the queries and fills in the proper variables;

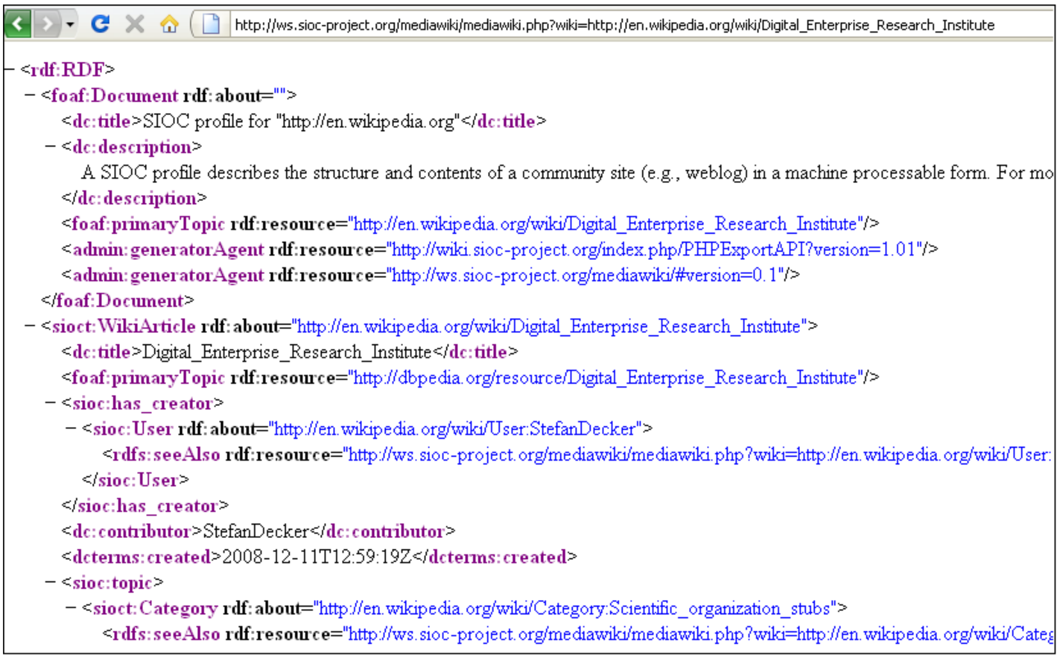
- it calls the SIOC API to export in RDF the fetched structural information and outputs the results in RDF/XML serialization, using the appropriate content-type.

The first steps in the process involve the use of the MediaWiki API, a brief description of the API is provided in the following section.

The semantic data extracted is displayed by the application simply sending to the browser the RDF text created, then the web browser in use can choose the way to display it. Normally, without any plugins installed, the output should look like in Fig 5.3.

It is possible to have a better browsing experience installing plugins or using special purpose web-services, for example: the *SIOC Browser*, that allows people to browse and receive additional information from SIOC data sources or data stores [10]; or using generic RDF tools, such as Disco[4], the OpenLink

[4] http://sites.wiwiss.fu-berlin.de/suhl/bizer/ng4j/disco/



http://ws.sioc-project.org/mediawiki/mediawiki.php?wiki=http://en.wikipedia.org/wiki/Digital_Enterprise_Research_Institute

```
- <rdf:RDF>
  - <foaf:Document rdf:about="">
      <dc:title>SIOC profile for "http://en.wikipedia.org"</dc:title>
    - <dc:description>
        A SIOC profile describes the structure and contents of a community site (e.g., weblog) in a machine processable form. For mo
      </dc:description>
      <foaf:primaryTopic rdf:resource="http://en.wikipedia.org/wiki/Digital_Enterprise_Research_Institute"/>
      <admin:generatorAgent rdf:resource="http://wiki.sioc-project.org/index.php/PHPExportAPI?version=1.01"/>
      <admin:generatorAgent rdf:resource="http://ws.sioc-project.org/mediawiki/#version=0.1"/>
    </foaf:Document>
  - <sioct:WikiArticle rdf:about="http://en.wikipedia.org/wiki/Digital_Enterprise_Research_Institute">
      <dc:title>Digital_Enterprise_Research_Institute</dc:title>
      <foaf:primaryTopic rdf:resource="http://dbpedia.org/resource/Digital_Enterprise_Research_Institute"/>
    - <sioc:has_creator>
      - <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:
        </sioc:User>
      </sioc:has_creator>
      <dc:contributor>StefanDecker</dc:contributor>
      <dcterms:created>2008-12-11T12:59:19Z</dcterms:created>
    - <sioc:topic>
      - <sioct:Category rdf:about="http://en.wikipedia.org/wiki/Category:Scientific_organization_stubs">
          <rdfs:seeAlso rdf:resource="http://ws.sioc-project.org/mediawiki/mediawiki.php?wiki=http://en.wikipedia.org/wiki/Categ
```

Figure 5.3: Screenshot of the SIOC-MediaWiki webservice output

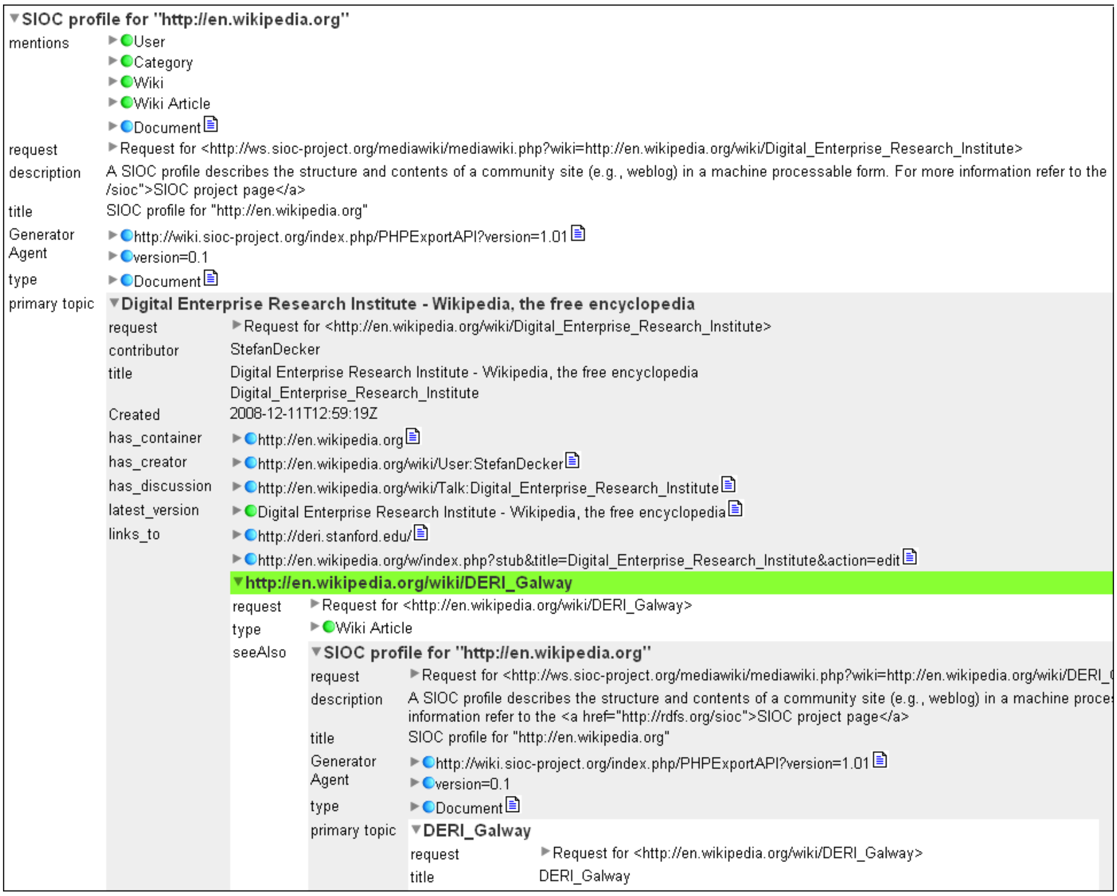
RDF Browser[5], Tabulator[6], Timeline[7] or Zitgist[8], directly using SIOC data in RDF/XML or by translating it into a specific data type.

For instance, using the Tabulator web service, the RDF export of the DERI page from Wikipedia, is visualized as in Fig 5.4, with an interactive view of classes and properties and the possibility to browse all the `seeAlso` links simply clicking on each "expand" button next to the link.

### 5.1.2 MediaWiki API

An important part of the entire exportation process is the correct use of the MediaWiki's API. In this regard the API documentation provided by MediaWiki developers has been studied, it is publicly available at the following address:

[5] `http://demo.openlinksw.com/DAV/JS/rdfbrowser/index.html`
[6] `http://www.w3.org/2005/ajar/tab`
[7] `http://simile.mit.edu/timeline/`
[8] `http://browser.zitgist.com:8890/`



```
▼SIOC profile for "http://en.wikipedia.org"
mentions       ▶User
               ▶Category
               ▶Wiki
               ▶Wiki Article
               ▶Document
request        ▶Request for <http://ws.sioc-project.org/mediawiki/mediawiki.php?wiki=http://en.wikipedia.org/wiki/Digital_Enterprise_Research_Institute>
description    A SIOC profile describes the structure and contents of a community site (e.g., weblog) in a machine processable form. For more information refer to the
               /sioc">SIOC project page</a>
title          SIOC profile for "http://en.wikipedia.org"
Generator      ▶http://wiki.sioc-project.org/index.php/PHPExportAPI?version=1.01
Agent          ▶version=0.1
type           ▶Document
primary topic  ▼Digital Enterprise Research Institute - Wikipedia, the free encyclopedia
               request         ▶Request for <http://en.wikipedia.org/wiki/Digital_Enterprise_Research_Institute>
               contributor     StefanDecker
               title           Digital Enterprise Research Institute - Wikipedia, the free encyclopedia
                               Digital_Enterprise_Research_Institute
               Created         2008-12-11T12:59:19Z
               has_container   ▶http://en.wikipedia.org
               has_creator     ▶http://en.wikipedia.org/wiki/User:StefanDecker
               has_discussion  ▶http://en.wikipedia.org/wiki/Talk:Digital_Enterprise_Research_Institute
               latest_version  ▶Digital Enterprise Research Institute - Wikipedia, the free encyclopedia
               links_to        ▶http://deri.stanford.edu/
                               ▶http://en.wikipedia.org/w/index.php?stub&title=Digital_Enterprise_Research_Institute&action=edit
                               ▼http://en.wikipedia.org/wiki/DERI_Galway
                               request   ▶Request for <http://en.wikipedia.org/wiki/DERI_Galway>
                               type      ▶Wiki Article
                               seeAlso   ▼SIOC profile for "http://en.wikipedia.org"
                                         request        ▶Request for <http://ws.sioc-project.org/mediawiki/mediawiki.php?wiki=http://en.wikipedia.org/wiki/DERI_
                                         description    A SIOC profile describes the structure and contents of a community site (e.g., weblog) in a machine proce
                                                        information refer to the <a href="http://rdfs.org/sioc">SIOC project page</a>
                                         title          SIOC profile for "http://en.wikipedia.org"
                                         Generator      ▶http://wiki.sioc-project.org/index.php/PHPExportAPI?version=1.01
                                         Agent          ▶version=0.1
                                         type           ▶Document
                                         primary topic  ▼DERI_Galway
                                                        request    ▶Request for <http://en.wikipedia.org/wiki/DERI_Galway>
                                                        title      DERI_Galway
```

Figure 5.4: The SIOC-MediaWiki webservice using Tabulator

`http://www.mediawiki.org/wiki/API`.

The goal of this API (Application Programming Interface) is to provide direct, high-level access to the data contained in the MediaWiki databases. Client programs can use the API to login, get data, and post changes. The API takes its input through parameters in the query string, then its structure is divided in modules. Every module has its own set of parameters, which is listed in the documentation. The main examined and used module is the Query (or `action=query`) module. It allows for retrieving all sorts of data, and it has many submodules (called query modules), each with a different function. There are three types of query modules:

- *Meta information* about the wiki and the logged-in user;
- *Properties* of pages;
- *Lists* of pages that match certain criteria.

A typical example of a query to the API is illustrated in the following Listing 5.1, where it is first described the query then the obtained results in XML format.

```
URL encoded query:

/api.php?action=query&titles=Albert%20Einstein&prop=info&format=xmlfm

results:

<?xml version="1.0" encoding="utf-8"?>
<api>
  <query>
    <pages>
      <page pageid="736" ns="0" title="Albert Einstein" touched="2007-07-06
          T04:37:30Z" lastrevid="142335140" counter="4698" length="86906" />
    </pages>
  </query>
</api>
```

Listing 5.1: A query to the MediaWiki API

This query requests for information about the wiki page with title "Albert Einstein". The results provide some general information about the article such as date of creation, wiki ID, title, etc. They are displayed in XML format because the `format=xmlfm` parameter has been explicitly used. MediaWiki API supports a number of generic output formats (XML,PHP,JSON,...), and some domain specific formats (RSS, ...) for special modules. Since my application is developed in PHP, I decided to get the output in PHP format.

The properties used to get the content about a particular web page or about the wiki itself can be summarized with the following points.

- *info*: used to get basic page information.

  By default, the following properties will be listed and have been used in my application: page ID, namespace, title, last touched timestamp, last revision ID, whether the page is a redirect.

- *revisions*: returns revisions for a given page, or the latest revision for each of several pages.

  The following properties have been used in the application: revision ID, timestamp (the date and time the revision was made), the user who made the revision, the maximum number of revisions to return, revision ID to start listing from, revision ID to stop listing at, direction to list in.

- *categories*: gets a list of all categories used on the provided pages.

  The following properties have been used in the application: page ID, wiki page title, namespace, category title.

- *links*: gets a list of all wiki internal links on the provided pages.

  The following properties have been used in the application: page ID, page title, namespace, link title.

- *extlinks*: gets a list of all external links on the provided pages.

  The following properties have been used in the application: page ID, page title, namespace, link title.

- *backlinks*: lists pages that link to a given page, similar to the `Special:Whatlinkshere` function provided by the MediaWiki software. By default they are ordered by linking page title.

  The following properties have been used in the application: page ID, page title, namespace, link title.

- *parse*: returns parsed wikitext following the specified parameters.

  The following properties have been used: wikitext to parse, wikipage title, properties to get: HTML text, categories, links, external links, revision ID.

All these properties and parameters have been used to get all the needed information to be able to semantically model wiki articles. The results are parsed by the application using the PHP's `unserialize()` function, since the output data format is serialized, with line breaks added for readability.
Once all the data from all the queries is retrieved and parsed, and properly stored into variables, it is delivered to the SIOC PHP API that produces RDF structured data following the principles explained in the previous chapters. A detailed description of the RDF model generated is illustrated in the following paragraph and in Section 5.1.4 "The RDF model generated".

### 5.1.3 Following the Linked Data principles

One of the 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, I followed the Linked Data best practices defined in [4], [6] and [9].

Particularly, to offer a better browsing experience and ease the process of crawling SIOC export of MediaWiki instances, the webservice automatically 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 it is possible to see in the following example related to a particular `sioc:User` (Listing 5.2). It is also possible to notice in the example that there is a distinction between the concept itself (i.e. `User:StefanDecker`) and the related RDF page. At the moment, the exporter returns RDF/XML description, but it is also possible to consider using content-negotiation to redirect, for instance, to the HTML page regarding that particular user. These `seeAlso` links are very useful not only to provide links 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, entire wiki sites have been crawled and exported 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 5.2: Modeling a user in the MediaWiki exporter

A very interesting future development, in a Linking Open Data perspective, might be the association between this `OnlineAccount` and the `foaf:Person` holder of the account. And this is possible with the `foaf:holdsAccount` property. Adding this feature it becomes possible to interlink precisely all the user accounts on different wikis belonging to the same person and then, for example, to know what are the contributions made by the same persons on different wikis, what are their interest areas, etc. At the moment it is only possible to use the literal username, but this method is not so accurate.

Another interesting feature is the linkage to the corresponding DBpedia[9] resource, if the article belongs to the english version of the 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.

[9] `http://www.dbpedia.org`

It is very important to note that a rdfs:seeAlso link has been introduced for every URI that describes another wiki page, so that it can be exported as well.

### 5.1.4 The RDF model generated

The following snippet of RDF code has been generated by the exporter from the Wikipedia page regarding DERI[10].

It has been decided to use dc:title, dcterms:created and dc:contributor to model the document with the Dublin Core ontology. The choice of using dcterms:created to identify the date of creation of this particular revision, instead of dcterms:modified, has been made because the URI of this WikiArticle refers to a single revision and a revision could not be modified but created. The author of the version is a dc:contributor, and his username is expressed as a literal. It is also exported the user account URI, used by the author in this wiki, as a sioc:User (subclass of foaf:OnlineAccount).

All these information, categories, internal and external links, discussion pages and wiki containers are modeled in the following way.

```
<sioct:WikiArticle rdf:about="http://en.wikipedia.org/wiki/
    Digital_Enterprise_Research_Institute">
 <dc:title>Digital_Enterprise_Research_Institute</dc:title>
 <foaf:primaryTopic rdf:resource="http://dbpedia.org/resource/
     Digital_Enterprise_Research_Institute"/>
 [...]
 <dc:contributor>StefanDecker</dc:contributor>
 <dcterms:created>2008-12-11T12:59:19Z</dcterms:created>
 <sioc:topic>
   <sioct:Category rdf:about="http://en.wikipedia.org/wiki/Category:
       Scientific_organizations">
   [...]

 <sioc:links_to>
```

[10] http://en.wikipedia.org/wiki/Digital_Enterprise_Research_Institute

```
   <sioct:WikiArticle rdf:about="http://en.wikipedia.org/wiki/Semantic_Web">
   [...]

 <sioc:has_discussion>
   <sioct:WikiArticle rdf:about="http://en.wikipedia.org/wiki/Talk:
       Digital_Enterprise_Research_Institute">
   [...]

 <sioc:has_container>
   <sioct:Wiki rdf:about="http://en.wikipedia.org"/>
 [...]
</sioct:WikiArticle>
```

The listing above has been obviously reduced for displaying purposes, every resource has a `seeAlso` link associated (as illustrated in Listing 5.2 for the description of a `sioc:User`) and of course a closing tag.
As illustrated, and defined in the previous section, MediaWiki's categories are mapped into the `sioct:Category` class through the `sioc:topic` property. Internal links can be exported as well, following the `rdfs:seeAlso` links. The same for the discussion pages, marked by MediaWiki with the "Talk:" prefix[11] and defined in range of the `sioc:has_discussion` property. The `sioct:Wiki` container, identifying the wiki site holding the `sioct:WikiArticle`, is espressed with the `sioc:has_container` property.
Pages versioning, as described in the Section 4 "Extending the SIOC ontology", is modeled in RDF as in the following example.

```
<sioc:previous_version>
  <sioct:WikiArticle rdf:about="http://en.wikipedia.org/w/index.php%3Ftitle
      %3DDigital_Enterprise_Research_Institute%26oldid%3D246494912">
  [...]

<sioc:latest_version>
  <sioct:WikiArticle rdf:about="http://en.wikipedia.org/wiki/
      Digital_Enterprise_Research_Institute">
  [...]
```

[11] `http://en.wikipedia.org/wiki/Wikipedia:Talk_page`

Listing 5.3: Modeling versioning in the MediaWiki exporter

To be noted that the URI of a previous version is typically marked by MediaWiki with the "oldid" parameter at the end. Furthermore this is an export example of the latest version of the "Digital_Enterprise_Research_Institute" article, so a newer sioc:next_version is not expected. The webservice exports only the previous/next_version properties, and not the earlier/later_version ones, because the transitive super-properties can be inferred by a reasoner.

The last interesting feature of the RDF model exported is the modeling solution for redirection pages. MediaWiki pages may have several different names/titles associated to the same article through a redirect, so that for example if you look for the following URL: http://en.wikipedia.org/wiki/DERI, you will be automatically redirected to the main article: http://en.wikipedia.org/wiki/Digital_Enterprise_Research_Institute. Redirections has been modeled as in the following example:

```
<sioct:WikiArticle rdf:about="http://en.wikipedia.org/wiki/DERI">
  [...]
  <owl:sameAs rdf:resource="http://en.wikipedia.org/wiki/
      Digital_Enterprise_Research_Institute"/>
  <rdfs:seeAlso rdf:resource="http://ws.sioc-project.org/mediawiki/mediawiki
      .php?wiki=http://en.wikipedia.org/wiki/
      Digital_Enterprise_Research_Institute"/>
  [...]
```

Listing 5.4: Modeling redirections in the MediaWiki exporter

The owl:sameAs property indicates that two URI references actually refer to the same thing: the individuals have the same "identity" (i.e., in the previous example, properties of the "DERI" page are the same on the "Digital Enterprise Research Institute" page). The rdfs:seeAlso link provides further details about the subject pointing to the RDF export of the main page.

### 5.1.5 Generic wiki PHP API

In this section an overview on the SIOC PHP Export API, that has been extended, will be given. The PHP Export API provides an easy way for developers to create SIOC exporters, as it maps SIOC Classes to PHP objects, with simple functions to export the created data. In this way it enables developers to create SIOC export tools without the need to get into technical details about how information is represented in RDF/XML as they are operating at the level of SIOC concepts instead. Thus, developers only have to deal with extracting content from their databases and then passing it to the API that will render SIOC data.

The API consists of a PHP file, called `sioc_inc.php`, mainly constituted by two main classes: the `SIOCExporter` class, which is the exporter itself, and the `SIOCObject` class, a generic SIOC Class from which all real SIOC classes inherits (e.g. `SIOCPost`, `SIOCWikiArticle`, etc.). The basic exporter usage is to create a `SIOCExporter` with the corresponding class, to set its parameters, to add it some SIOC Classes instances and to export data in RDF. The second fundamental class is `SIOCObject`, a virtual class that is not designed to be instantiated, but is a base to other SIOC / PHP Classes such as Site, Forum, Post, etc.

To be able to model wikis and wiki pages the SIOC API has been extended, following the principles exposed in the previous sections. The classes `SIOCWikiArticle`, `SIOCWiki` and `SIOCCategory` have been created, each extending the `SIOCObject` class. All these classes are used by the PHP MediaWiki exporter to write RDF data on output starting from the extracted data.

The PHP source code of the API is freely accessible through the SIOC Wiki page about the API[12] or it can be downloaded from SVN[13].

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

[13] `http://sw.deri.org/svn/sw/2005/08/sioc/phpapi/`

# Chapter 6

# Query examples

In order to evaluate my proposal, after the creation of the SIOC-MediaWiki exporter, it has been crawled and exported four different MediaWiki instances, using the exporter and a crawler able to follow the `rdfs:seeAlso` links. The aim was to collect a large amount of data and to put it on a RDF store, so that it was possible to query this data with SPARQL and test some useful and advanced interrogations. Four different wikis have been crawled, each one belonging to the same area of interest in order to have a high probability of shared topics and users: *Semanticweb.org*[1], *Protégé Wiki*[2], *RDFa Wiki*[3] and the *ONTOLORE Karlsruhe* wiki[4]. For each wiki site it has been used a single entry point, thanks to the `seeAlso` links, and a crawler capable to follow these properties.
In total it has been collected about 1GB of RDF data that has been inserted into a RDF triple store. SESAME[5] [13] has been used as triple store, but since an higher degree of inference was needed (because of the OWL transitive properties)

[1] `http://www.semanticweb.org`
[2] `http://protegewiki.stanford.edu`
[3] `http://rdfa.info/wiki/RDFa_Wiki`
[4] `http://logic.aifb.uni-karlsruhe.de/wiki/ONTOLORE`
[5] `http://www.openrdf.org/`

it has been installed and configured the OWL reasoning engine OWLIM[6] on top of SESAME.
The SPARQL queries implemented are mainly focused on an inter-wikis browsing experience, because one of the most important purposes of this project is the interconnection between wiki communities. Furthermore it has been thought about particular use-cases which cannot be implemented without the use of semantics.

## 6.1 Advanced querying for a single wiki

First of all, some queries that implement the same common features that all the wikis offer have been developed. An example is the following SPARQL query:

```
SELECT DISTINCT ?version ?creator ?date
WHERE {
 <http://semanticweb.org/wiki/Wikipedia> sioc:
    earlier_version ?version.
 ?version dcterms:created ?date ;
  sioc:has_creator ?creator .
 FILTER (xsd:dateTime(?date) > xsd:dateTime("2008-01-01
    T00:00:00Z")).
 FILTER (xsd:dateTime(?date) < xsd:dateTime("2008-01-31
    T00:00:00Z")).
} ORDER BY DESC(?date)
```

Listing 6.1: Querying earlier versions of a wiki page

It displays the revisions history of the "*Wikipedia*" article, created in January 2008, with creator name and creation date, ordered by descending date. Please note that to get all the previous versions of an article we just use the inferred

[6] http://www.ontotext.com/owlim/

transitive property sioc:earlier_version, so it is not needed to "jump" with a query from each version to its sioc:previous_version.

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?".

```
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 6.2: Identifying collaborating users

In Fig. 6.1 a diagram that summarizes the above query is displayed, meanwhile in Fig. 6.2 I show the results I got querying on SESAME. As we can see, this query takes advantage of the transitivity of the newly created property sioc:earlier_version, since identifying users worked on earlier version, and not necessary immediately on the previous one.

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 the same knowledge areas. It is an important query especially in a social semantic context.

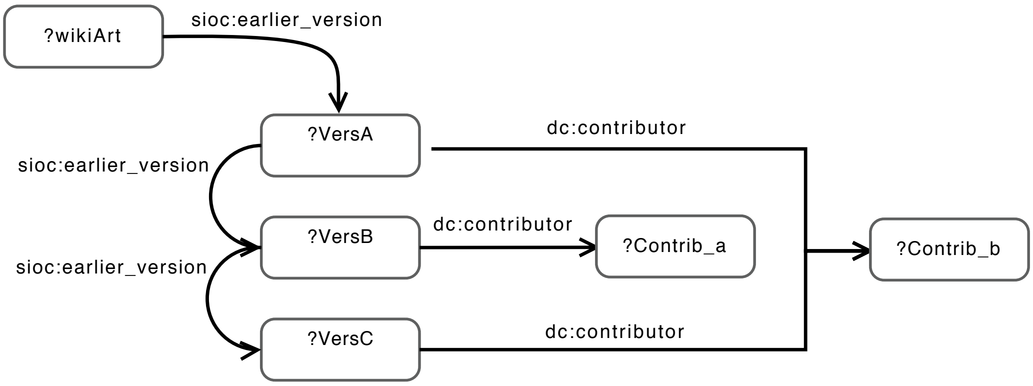
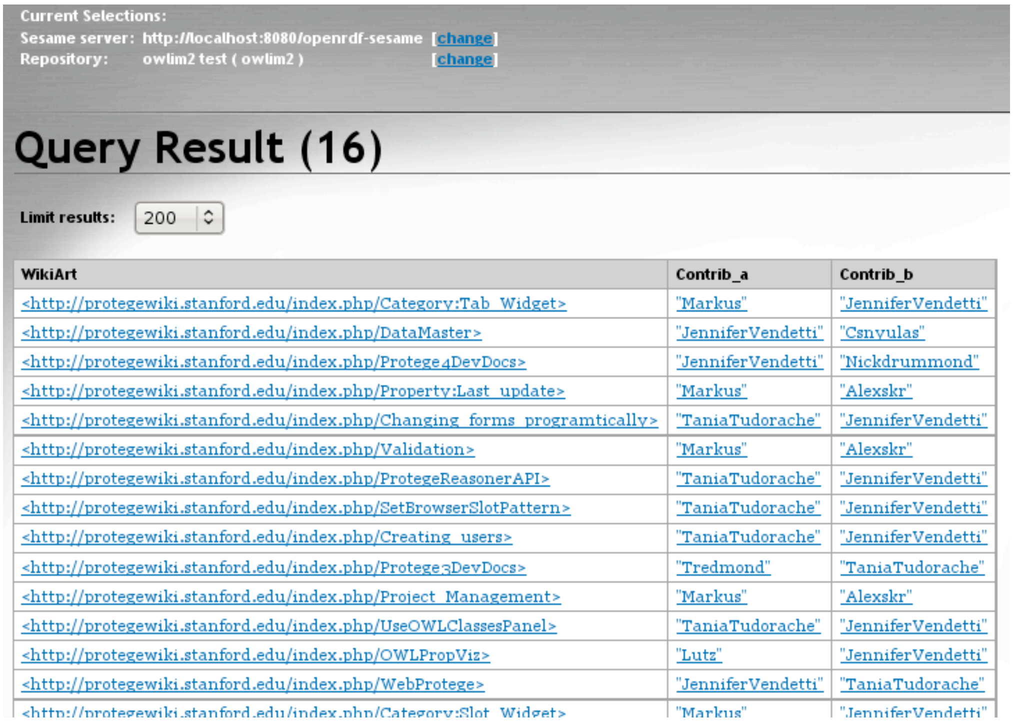


Figure 6.1: Identifying collaborating users


Current Selections:
Sesame server: http://localhost:8080/openrdf-sesame [change]
Repository: owlim2 test ( owlim2 ) [change]

Query Result (16)

Limit results: 200

| 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" |

Figure 6.2: Results of the query for collaborating users on SESAME

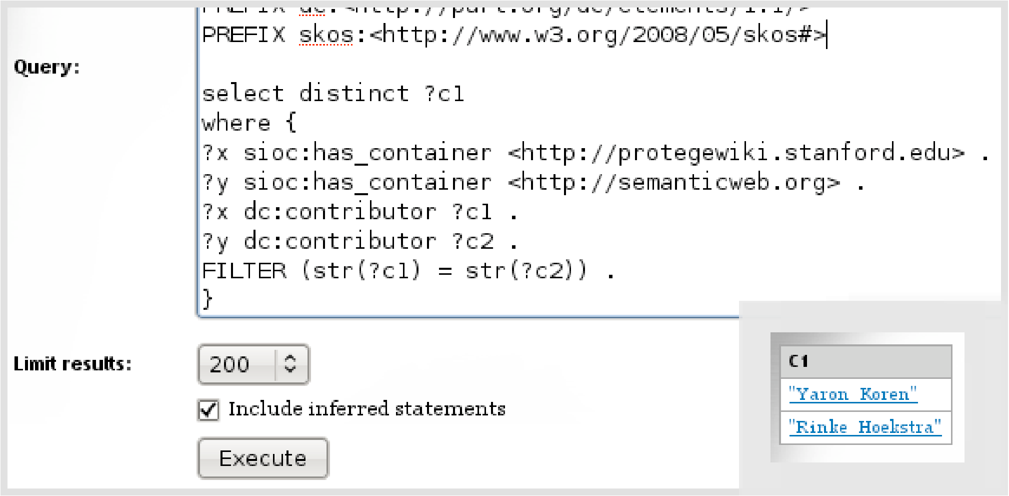
## 6.2 Cross-wiki integration and querying

Another interesting feature of the described 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, my proposal has many advantages, as relying on SPARQL instead of a particular API, as well as providing advanced inference capabilities that the original API does not provide. The following query identifies users that are involved in different wikis, based on the similarity of their usernames.

```
SELECT DISTINCT ?creator ?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 6.3: Identifying pages created by a single user in different wikis

While this is a very simple queryit requires high computation capabilities when ran through a large number of different wikis. Hence, in Fig. 6.3 a screenshot of the results got after running the same query between the *Semanticweb.org* wiki and the *Protégé* wiki is displayed. Instead of also displaying all the other details, such as the related wiki pages and the two wiki containers, it has been shown only the distinct usernames of the found users.



Query:

```
PREFIX dc:<http://purl.org/dc/elements/1.1/>
PREFIX skos:<http://www.w3.org/2008/05/skos#>

select distinct ?c1
where {
?x sioc:has_container <http://protegewiki.stanford.edu> .
?y sioc:has_container <http://semanticweb.org> .
?x dc:contributor ?c1 .
?y dc:contributor ?c2 .
FILTER (str(?c1) = str(?c2)) .
}
```

Limit results: 200

☑ Include inferred statements

| C1 |
|---|
| "Yaron Koren" |
| "Rinke Hoekstra" |

Figure 6.3: 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, it is possible to benefit from the strong ties that exist between FOAF and SIOC and the fact that the modeling of a wiki user has been done 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/person#me> foaf:holdsAccount ?
    account .
```

```
  ?account rdf:type sioc:User .
  ?content sioc:has_creator ?account .
}
```

Listing 6.4: Cross-querying by combining FOAF and SIOC

Web sites using SIOC collectively create the "SIOC-o-sphere", and by doing so it is possible to interrogate different containers with the same query and interlink blogs, forums, wikis, etc.

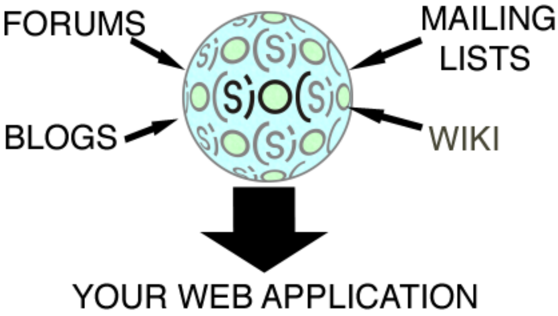


Figure 6.4: Interconnected websites on the SIOC-o-Sphere.

# Conclusions

In this thesis, it has been presented how the SIOC ontology and lightweight semantics can be used and extended to represent the structure of wikis in unified ways. First it has been explained the motivation regarding some properties of wikis on which I focused on in the modeling process, particularly concentrating on a versioning process, and how it is possible to benefit in this case of OWL reasoning capabilities. Then, it has been described how I 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, some examples regarding how this data could be efficiently used for querying purposes have been provided.

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. It has been also considered 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 the important goals is also to run cross-queries between this Wikipedia export and DBpedia, for instance to identify which people where the most active on a particular wiki page or topic, reasoning also on the content of wiki pages. Finally, this thesis gives a complete and nice overview regarding how to extend an ontology such as SIOC for particular purposes.

# Bibliography

[1] OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004 , World Wide Web Consortium, 2004. http://www.w3.org/TR/owl-features/.

[2] RDF Semantics. W3C Recommendation 10 February 2004, World Wide Web Consortium, 2004. http://www.w3.org/TR/rdf-mt/.

[3] RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004, World Wide Web Consortium, 2004. http://www.w3.org/TR/rdf-schema/.

[4] 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/.

[5] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL Web Ontology Language Reference. W3C Recommendation 10 February 2004, World Wide Web Consortium, 2004. http://www.w3.org/TR/2004/REC-owl-ref-20040210/.

[6] Tim Berners-Lee. Linked Data. Design issues for the world wide web, World Wide Web Consortium, 2006. `http://www.w3.org/DesignIssues/LinkedData.html`.

[7] Tim Berners-Lee, James A. Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[8] Diego Berrueta, Dan Brickley, Stefan Decker, Sergio Fernández, Christoph Görn, Andreas Harth, Tom Heath, Kingsley Idehen, Kjetil Kjernsmo, Alistair Miles, Alexandre Passant, Axel Polleres, Luis Polo, Editors: Uldis Bojārs Michael Sintek, and John G. Breslin. SIOC Core Ontology Specification. W3C Member Submission 12 June 2007, World Wide Web Consortium, 2007. `http://www.w3.org/Submission/sioc-spec/`.

[9] 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/`.

[10] Uldis Bojārs, John G. Breslin, and Alexandre Passant. SIOC Browser – Towards a Richer Blog Browsing Experience. In *Proceedings of the 4th Blogtalk Conference (Blogtalk Reloaded)*. Books on demand, 2006.

[11] Uldis Bojārs, Alexandre Passant, Richard Cyganiak, and John G. Breslin. Weaving sioc into the web of linked data. In *Proceedings of the WWW2008 Workshop Linked Data on the Web (LDOW2008)*, volume 369 of *CEUR Workshop Proceedings*. ceur-ws.org, 2008.

[12] 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.

[13] 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.

[14] 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.

[15] Thomas. R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies*, 43(5–6):907–928, 1995.

[16] 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.

[17] Graham Klyne and Jeremy J. Carroll. Resource Description Framework (RDF): Concepts and abstract syntax. W3C Recommendation 10 February 2004, World Wide Web Consortium, 2004. `http://www.w3.org/TR/rdf-concepts/`.

[18] Markus Krötzsch and Vrandečić. Semantic MediaWiki documentation: Help:User_Manual, 2009. `http://semantic-mediawiki.org/wiki/Help:User_manual`.

[19] Markus Krötzsch, Denny Vrandečić, 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.

[20] Christoph Lange, Uldis Bojars, 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.

[21] Bo Leuf and Ward Cunningham. *The Wiki Way: Collaboration and Sharing on the Internet*. Addison-Wesley Professional, 2001.

[22] 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/.

[23] Tim O'Reilly. O'Reilly Network: What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software, 30 September 2005. http://www.oreillynet.com/lpt/a/6228.

[24] 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.

[25] Sebastian Schaffert. IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. In *First International Workshop on Semantic Technologies in Collaborative Applications (STICA 06)*, 2006.

[26] 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.

[27] Cunningham Ward and Leuf Bo. What is wiki, 2002. `http://www.wiki.org/wiki.cgi?WhatIsWiki`. accessed March 2009.