SemSLATES: Improving Enterprise 2.0 Information Systems using Semantic Web Technologies

Alexandre Passant[1], Philippe Laublet[2], John G. Breslin[1,3], Stefan Decker[1]
[1] Digital Enterprise Research Institute
National University of Ireland, Galway, Ireland
E-mail: firstname.lastname@deri.org
[2] LaLIC, Université Paris-Sorbonne
28 rue Serpente, 75006 Paris, France
E-mail: philippe.laublet@paris-sorbonne.fr
[3] School of Engineering and Informatics
National University of Ireland, Galway, Ireland
E-mail: john.breslin@nuigalway.ie

***Abstract*—While the use of Web 2.0 tools and principles in organizations — a practice commonly known as Enterprise 2.0 — helps knowledge workers to collaboratively build and exchange information more easily, it introduces several issues in terms of efficiently integrating and retrieving this information. In this paper, we describe how Semantic Web technologies can be efficiently deployed to solve these issues and to enhance such ecosystems. We detail the SemSLATES methodology, a middleware architecture for Enterprise 2.0 that combines Social Web principles and Semantic Web technologies in a novel and innovative way for the benefit of end users. Since the work presented here has been applied in an industrial context, we emphasize our motivations and the benefits of the approach through a complete and real-world case study.**

I. Introduction

During recent years, the Web has shifted from a read-only Web to a read-write Web, also known as the Social Web or Web 2.0 [30], in which people voluntarily provide and share content collaboratively. Not only are these Web 2.0 paradigms being used for leisure-oriented applications (such as photo sharing or social networks), but they can also be useful for more professionally-oriented applications, from discussions in scientific communities (such as Nature Networks[1]) to knowledge management in enterprises. To that extent, Enterprise 2.0 [28] is a recent trend describing the next generation of information management and collaboration tools being used in organizational contexts, similar to how Web 2.0 is often being used to describe a second-generation of Web-based communities and hosted services. It is considered as *"the use of emergent social software platforms within companies, or between companies and their partners or customers"*[2]. Interestingly, a major characteristic of Enterprise 2.0 is not its technical aspect, but its underlying philosophy of sharing voluntary information in an open way, sometimes without necessarily being directly rewarded (as in wikis, in which users' contributions are all merged together). Hence, the Enterprise 2.0 philosophy is different from that of existing technologies that have been used so far for collaborative work in organizations (with tools such as BSCW[3] involving workflows, rights management, etc.) and sometimes this can lead to difficulties in terms of its adoption.

When defining Enterprise 2.0, McAfee [28] introduces the SLATES acronym, identifying six main features that such information systems should cover: *Search*, *Links*, *Authoring*, *Tags*, *Extension* and *Signals*. For example, blogs and wikis can ease the *Authoring* process and definition of *Links*, with easy-to-use user interfaces for publishing content; folksonomies can be used to add *Tags*, *i.e.* free-text keywords, to content; while microblogging and RSS feeds can be used for *Extension* and *Signals* purposes, the former being an efficient way to provide real-time query answering in organizations. Furthermore, most of these tools provide plain-text *Search* functionalities.

However, while such ecosystems provide new ways to collaboratively build and share information in enterprises, they raise various issues. While some of these issues are sociological as we have mentioned before (the related philosophy often breaks the usual internal workflows of information management, thus requiring new social paradigms such as combining bottom-up and top-down strategies [4]), we will focus on the technical issues in this paper. Moreover, since such ecosystems are gaining more and more interest — as shown by recent studies such as one by Forrester Research that predicts that the Enterprise 2.0 solutions market will be $4.6 billion by 2013[5] — or another by Gartner that identified that many more *social computing platforms* will be considered for adoption by companies during the next ten years[6] — we believe that there is a real need to solve these issues, and that Semantic Web technologies can be a key enabler for such augmented

[1] http://network.nature.com
[2] http://blog.hbs.edu/faculty/amcafee/index.php/faculty_amcafee_v3/enterprise_20_version_20/
[3] http://bscw.de/
[4] http://strange.corante.com/2006/03/05/an-adoption-strategy-for-social-software-in-enterprise
[5] http://www.forrester.com/Research/Document/Excerpt/0,7211,43850,00.html
[6] http://gartner.com/it/page.jsp?id=739613

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8369
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8369

Enterprise 2.0 systems.

The rest of this paper is structured as follows. In the next section, we detail three main problems with such information systems regarding data management and retrieval that motivated our work and that we encountered in an Enterprise 2.0 information system at EDF R&D[7]. Then, we will introduce some of the basics of Semantic Web technologies and detail our SemSLATES proposal, which provides a social semantic middleware layer on the top of such platforms, that aims to solve the previously-described issues. We then will describe what are its needs in terms of data models, semantic annotations for existing content, and ontology population, and how we propose to achieve it with a user-driven approach. Then, the second part of this paper describes a case study of the SemSLATES approach, describing the advantages of our proposal through a concrete and detailed implementation report. Here, we provide details of how we improved existing applications and built new ones to enhance information retrieval and user experience in an organizational context. We also consider the uptake and evaluation of the proposed architecture, as well as positioning our work regarding state of the art. Finally, we conclude the paper with an overview of future work and some more thoughts about Enterprise 2.0 and the Semantic Web.

II. Issues with Enterprise 2.0 information systems

A. *Information fragmentation and heterogeneity of data formats*

Information sharing and social networking in organizations is generally object-centric: people publish and browse information about particular *objects* such as projects, research topics, customers, etc. While some Enterprise 2.0 information systems are provided using dedicated suites, such as IBM Lotus Connections[8], they generally consist of an aggregation of services fragmented over a company network. Indeed, the heterogeneity of people and topics in organizations often leads to different ways to share information and hence to different applications being deployed: some people may only need an RSS reader, other will require a wiki or a blog, etc. Moreover, these services might be installed at different periods of time which make them even more heterogeneous (some software architectures may becomes obsolete and are consequently replaced by new ones, etc.).

As a consequence of services fragmentation, knowledge about a particular object can be spread between various sources in the company network: deliverables of a project can be drafted on a wiki but the latest project news may have been blogged and commented about on another platform, while RSS feeds may also contain valuable information regarding the project partners. Consequently, knowledge workers must query different sources of information in Enterprise 2.0 systems to get information about a particular topic. Most importantly, they must know that these sources exist in order to be able to reach them. Furthermore, different applications imply different APIs and data formats. Hence, information integration is a costly task for developers. While this is not a new issue *per se*, Enterprise 2.0 strengthens it since it allows open and uncontrolled publishing of information by end users, thus providing more and more distributed and heterogeneous data within a company.

B. *Knowledge integration and re-use*

Wikis are used in many organizations as a way to collaboratively build open and evolving knowledge bases, in areas ranging from project management to software development. For example, in our use case, more than 80 wikis and about 4500 pages were created on a single wiki server[9].

Yet, while much valuable information is contained in wikis, software agents cannot easily exploit and reuse this knowledge. A reader could learn from a wiki that EDF is a company that produces nuclear energy in France but an application will not be able to easily answer queries such as *"Is EDF located in France?"* or *"List all the companies referenced in this wiki"* without using complex natural language processing algorithms. The main reason is that wikis simply deal with documents and hyperlinks and not with machine-understandable representations of real-world concepts, as understood by readers when they are browsing or editing a page. A wiki engine will indeed store the fact that *"There are some hyperlinks between a page titled EDF, a page titled France and a page titled nuclear energy"*, but it will not be able to deduce anything about the nature of those different objects and their relationships, since the pages do not carry enough semantics about the knowledge that they contain. Hence, there is a gap between the documents and their interpretation. Consequently, a user must parse and read all the pages from a wiki to get an answer to his query, which can be a time-consuming task.

C. *Tagging and information retrieval*

Tagging is a practice well-known on Web 2.0 websites and consists of the attachment of multiple free-text keywords or "tags" as metadata to created content. Tags are often used as a means of categorizing similar content from various users for later retrieval and browsing. One important feature of tagging is its collaborative aspect, since tags can be shared between people, and are often used to retrieve and browse documents produced by other users. The collection of these tagging actions and keywords created by many users is generally known as a folksonomy [39].

The limits of free-tagging approaches are mainly due to tag ambiguity and heterogeneity as well as a lack of organization between tags [27]. Consequently, while tagging can be a time-saving method for end users who are categorizing content when publishing it, it becomes costly when trying to retrieve relevant information. For example, since tag-based search engines are plain-text only (*i.e.* allowing one to look for

[7]The work described in this paper, *i.e.* issues, solutions and the related evaluation, took part at Électricité de France Research and Development — http://rd.edf.fr, the major electricity company in France — during Ph.D. studies. More details can be found in [31].

[8]http://www-01.ibm.com/software/lotus/products/connections/

[9]This does not take into account wikis and pages from alternative wiki servers that may exist within the company.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8369
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8369

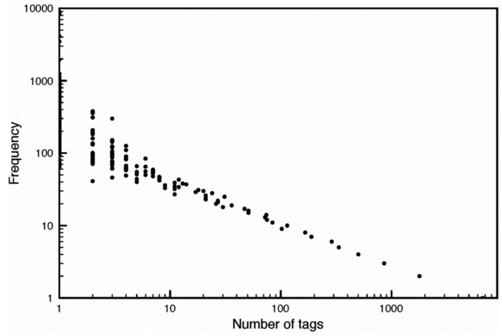

Fig. 1. Tags distributions within our corporate blogging platform

items tagged exactly with the given tag), someone looking for items tagged *"social_software"* will neither get those tagged *"socialsoftware"* (spelling variant) nor *"wiki"* (more specific tags), and they will not be able to find *"logiciel_sociaux"* either (linguistic variation). While clustering approaches can be used in some cases [19], an analysis of our organizational folksonomy raised other interesting issues regarding that topic.

First, as Fig. 1 emphasizes, most of the tags used in our blogging platform were used only a few times. In a total of 12,257 tags – used within 21,614 blog posts – it appeared that more than 68% were used twice or less, while only 10% were used more than ten times. As [22] has reported, tag clustering may not be adapted for this kind of distribution, unless they are combined with other techniques such as, for instance, taking into account the underlying tagged information. This is also a complex issue if dealing with non plain-text documents, such as PowerPoint files or diagrams that can be exchanged in corporate blogging platforms.

In addition, another lesson learned from our analysis is that users tag differently depending on their level of expertise and that these differences in tagging behaviors also raises several issues when retrieving content. For instance, we identified that experts in solar energies used tags such as *"TF"*[10], while non-experts would use generic ones like *"solaire"*[11]. This relates to the different *basic levels* of knowledge that people have regarding particular domains [38], as also raised by [18] when analyzing tagged content from Delicious. Furthermore, we identified that experts often did not use any broader terms when tagging content. Only 1% of the 194 items tagged with *"TF"* in our system were tagged together with *"solaire"*, while less than 0.5% of the 704 items tagged with *"solaire"* were tagged with *"TF"*.

Hence, clustering algorithms cannot be efficiently used to find related tags since they are too weakly related as discussed in [19], and lots of valuable content (*i.e.* created and tagged by experts) cannot be retrieved by non-experts who use generic keywords in their queries. This is a real problem in terms of knowledge sharing inside organizations and limits the possibilities offered by these collaborative platforms.

[10]An acronym for *Thin Film*, a particular kind of solar cell.

[11]French for *solar*.

III. SEMSLATES: SOCIAL SEMANTICS FOR ENTERPRISE 2.0

A. The Semantic Web and Linked Data

The Semantic Web is defined as *"an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation"* [7]. One of its goals is to enable large-scale data interoperability between applications, whatever or wherever the data structures are originally, leading to a Web of Data, *i.e.* a Web in which data is meaningful and machine-readable, in addition to the current Web of Documents. Various components are needed to achieve this goal (as depicted in Fig. 2), and we will now describe the ones that are used in our approach.

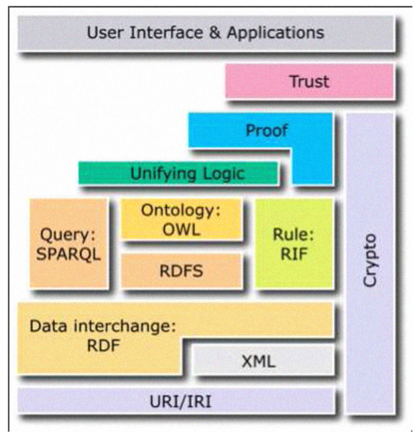

Fig. 2. The Semantic Web layer cake (source: http://www.w3.org/2001/sw/)

A first requirement is the use of URIs [6] to identify things on the Web, *i.e.* not only pages (as on the Web we currently known), but people, places, projects, etc. Thanks to URIs, each thing or information object that can be represented on the Web has its own identifier that can be shared between applications. Then, there is a need to represent statements about these things, and this is the goal of RDF (Resource Description Framework) [23]. RDF provides a way to create triples in the form of `<subject> <predicate> <object>`, with `subject` and `predicate` being URIs, and `object` being a URI or a literal. For example, `<http://apassant.net/alex>` `<http://xmlns.com/foaf/0.1/knows>` `<http://example.org/john>` would mean that Alex knows John. RDF is an abstract graph model and various serializations exist in order to provide machine readable RDF data, including RDF/XML, N3, Turtle, and more recently RDFa. The statements that can be defined using RDF are then used to provide semantic annotations on top of existing Web content, *i.e.* formal descriptions of data described on the Web. Moreover, there is a need for shared vocabularies to represent the meaning of these URIs and to define classes and properties for things through the use of ontologies [20].

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8369
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8369

In an ontology, one would define (in a formal way), that `<http://xmlns.com/foaf/0.1/knows>` represent an acquaintance relationship. It is important to mention that the goal of the Semantic Web is not to build a single broad Web-scale ontology, but to let everyone define their own, and to agree on a set of core ontologies (some of which are described in [9]) that could be extended if needed, in a distributed way. The two main languages used to represent ontologies on the Web are RDFS [12] and OWL [4], the latter being more expressive than the first one (*e.g.* using cardinality constraints on properties). Finally, SPARQL provides a graph-based query language [34] as well as a protocol [16] to retrieve information from RDF graphs. Other technologies are required to achieve the complete Semantic Web vision, and the combination of all these forms the *"Semantic Web layer cake"* (Fig. 2).

Recently, a pragmatic vision of the Semantic Web has emerged via the Linking Open Data project[12] (LOD), focusing on translating various datasets available on the Web into RDF and interlinking them, following the Linked Data principles described by [5]. Lots of different datasets have been provided via this LOD initiative, such as DBpedia [3] — the RDF export of Wikipedia — or GeoNames[13], a large geolocation database. All together, they form a complete Web-scaled graph of interlinked knowledge, commonly known as the Linked Open Data Cloud, and depicted in Fig. 3. We will describe later how these public datasets can be used to improve collaborative applications in an Enterprise 2.0 context.

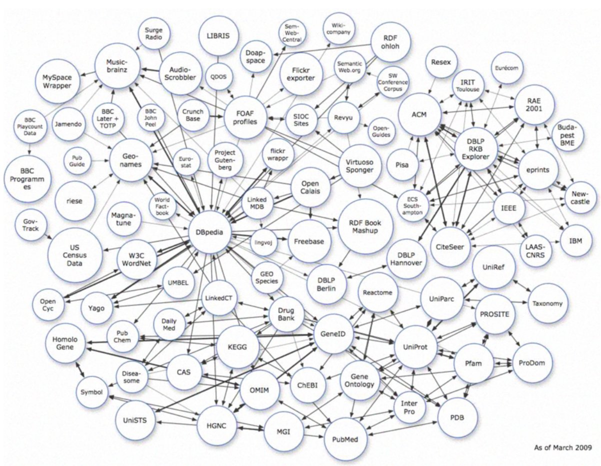

Fig. 3. The Linked Open Data Cloud from March 2009

B. The SemSLATES proposal

While we agree that most Enterprise 2.0 tools ease the *Authoring* process (from the SLATES acronym), we have shown in the previous section that they are somehow limited regarding some other features, especially *Search*. To solve these problems and to offer new value-added services to end users, our proposal consists of using Semantic Web technologies and principles, as described earlier, to provide interoperability between applications as well as modeling data from various business domains in a machine-readable way. In particular, our approach focuses on:

- using lightweight semantics and lightweight add-ons to existing tools, rather than building a new monolithic application that would require one to rethink existing information systems;
- re-using existing models and data already available on the Web, hence bridging a gap between the Web and Enterprise 2.0 information systems and taking advantage of structured data available on the Web to enrich Enterprise 2.0 information systems;
- considering the users as the core of the system, by being producers and consumers of semantic annotations and hence strongly emphasizing the collaborative side of Semantic Web knowledge management.

This additional stack of semantics on the top of existing Enterprise 2.0 information systems has led us to define the SemSLATES paradigm, *i.e.* Semantic Slates, that emphasizes how Semantic Web technologies can enhance the SLATES approach (Table 1). Thanks to this paradigm, a query such as *"List all the blog posts written last week about a project involving a company based in France"* can be answered, while it cannot be carried out using current Enterprise 2.0 systems. We will describe in detail the various improvements mentioned in this table in the next section, focusing on the implementation of our approach.

C. A lightweight middleware architecture for Enterprise 2.0

In order to realize the SemSLATES vision, there is a need to provide a layer of semantic annotations on the top of existing Enterprise 2.0 systems, thus integrating data from different heterogeneous components in a meaningful way. Here, it is important to keep in mind that our goal is not to define a new knowledge management suite for Enterprise 2.0, but rather to provide a means to integrate various existing components together in a transparent way for end users. In this sense, we defined a middleware process comparable to the *RDF bus* architecture proposed by Berners-Lee[14]: semantic annotations are produced from existing tools, and these are then interlinked and queried using Semantic Web standards (RDF(S)/OWL and SPARQL). In particular, various kind of semantic annotations are required to enable this SemSLATES layer, as depicted in Fig. 4, and we have defined these as follows:

- **socio-structural metadata** is required in order to model metadata about social interactions between users and the resulting documents in a uniform way, solving the issue of heterogenous data formats and APIs between different Enterprise 2.0 applications. We use ontologies such as FOAF (Friend Of A Friend[15]), a lightweight model used to represent people and their social networks [13], and

[12]http://linkeddata.org
[13]http://geonames.org
[14]http://www.w3.org/2005/Talks/1110-iswc-tbl/(26)
[15]http://foaf-project.org

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8369
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8369

	SLATES	SemSLATES
Search	Plain-text search	Semantic search based on RDF annotations
Link	Hyperlinks between documents	Relationships between resources
Authoring	Documents	Data and metadata
Tags	Tagging	Semantic indexing based on ontologies
Extension	Hyperlinks navigation	RDF graph-based navigation
Signals	RSS feeds	Semantically-indexed RSS feeds

TABLE I
SEMSLATES: EXTENDING SLATES USING SEMANTIC WEB TECHNOLOGIES

SIOC (Semantically-Interlinked Online Communities[16]), an ontology for representing online communities, their activities and contributions [11];

- **domain-specific ontologies and semantic annotations** are required in order to provide representations of the business information stored inside these Enterprise 2.0 systems (*e.g.* informations about companies, projects, etc.). Domain-specific ontologies can be used along with popular existing vocabularies [9]. In order to create the related annotations, semantic wikis are of particular interest, since they combine wiki principles (open editing, versioning, multi-authorship, etc.) and Semantic Web knowledge representation principles for a user-driven, open and evolving population of related knowledge bases, as described in more detail by [14] and [35];
- **semantic indexing** is required so as to allow content to be annotated with the URIs of resources instead of simple and unstructured keywords (hence solving the issues of free-text tagging) and creating links between domain ontologies and socio-structural metadata for a complete interlinked graph of data on top of existing Enterprise 2.0 systems. This can be done thanks to models such as MOAT — Meaning Of A Tag[17] [32] — an ontology and process to bridge the gap between free-keyword tagging and semantic indexing.

In addition, to enable these annotations, enhancements of the various tools must be as lightweight as possible in order to avoid disturbing users in their existing publishing habits. In the next section, we will describe some of the tools that have been built in our use case for the aforementioned purposes, as well as some new services that we were able to provide to users thanks to this architecture, improving the whole user experience of querying and browsing information in Enterprise 2.0 environments.

Regarding the latter aspect, in order to consume the RDF(S)/OWL data produced within this architecture, SemSLATES promotes the use of user interfaces that do not confront the end user with any Semantic Web modeling principles (URIs, triples, etc.). These interfaces include geolocation mashups and faceted browsing, some of which will be detailed in the next section. Hence, SemSLATES bridges a gap between Enterprise 2.0 and the Semantic Web in both directions (Fig. 5) by (1) providing a Semantic Web layer for unified and interlinked data from Enterprise 2.0 applications, and by (2) using user-friendly Web 2.0 interfaces for browsing and querying complex RDF(S)/OWL graphs.

[16]http://sioc-project.org
[17]http://moat-project.org

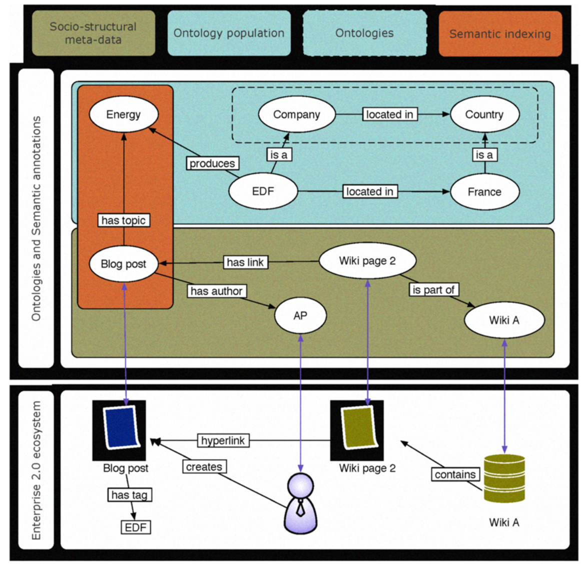

Fig. 4. Semantic annotations on top of an Enterprise 2.0 ecosystem

IV. SEMSLATES IN USE

We will now provide a complete report on our experiences regarding the implementation of the SemSLATES methodology at Électricité de France R&D, enhancing an existing Enterprise 2.0 ecosystem in the company that consisted of blogs, wikis and aggregated RSS feeds. In particular, we will focus on how this data integration process helped to solve the identified issues as well as to provide advanced services that could not have been possible without this additional layer of semantics, thus enriching the user experience.

A. SIOC and data integration

In order to automatically produce SIOC data from existing applications, and thus benefiting from a unified model for data management between our different tools, we developed different SIOC data exporters for the various blog and wiki systems, as well as a service translating incoming RSS feeds into

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8369
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8369

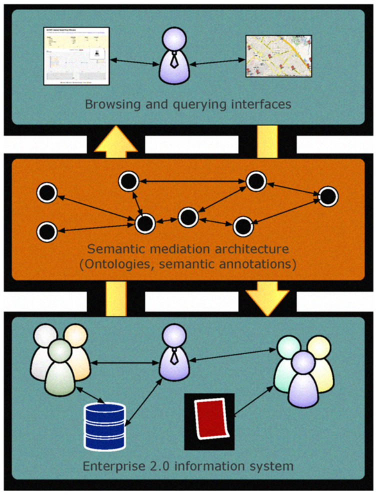

Fig. 5. Combining Enterprise 2.0 and Semantic Web technologies

SIOC data[18]. For each exporter, we relied on specific classes from the SIOC Types module[19], *e.g.* `sioct:BlogPost` to model blog posts and `sioct:WikiArticle` for wiki pages. For example, the following snippet of code shows how we represent that a particular blog post has been created by Alex on his own blog. In addition, we also made use of FOAF for modeling people's personal profile information (name, etc.).

```
<http://example.org/blog/post/212>
  rdf:type sioct:BlogPost ;
  sioc:has_creator
    <http://example.org/user/alex> ;
  sioc:has_container
    <http://example.org/blog/alex> .
```

From the time the exporters are set up, any new content is automatically provided in RDF using SIOC and FOAF as soon as it is created and this data is produced in a completely transparent way (from the users' perspective). They can therefore keep their existing publishing habits on their current tools of choice, and they do not need to change their behaviors to enable this first layer of socio-structural metadata. In this way, we instantly benefited from this common level of semantics for representing socio-structural metadata, generating SIOC and FOAF data from various existing applications that had heterogeneous underlying data structures. A total of about 20,000 RDF graphs were provided in this way during a period of approximately three years duration.

B. Ontology population with semantic wikis

The first part of this second step consisted of defining a set of ontologies for modeling data such as people, companies, projects, locations, etc. As we mentioned in the previous section, we relied on popular vocabularies such as FOAF,

[18]In addition, we would like to mention that various SIOC exporters and APIs are available as open-source applications, see http://wiki.sioc-project.org/index.php/Category:Applications

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

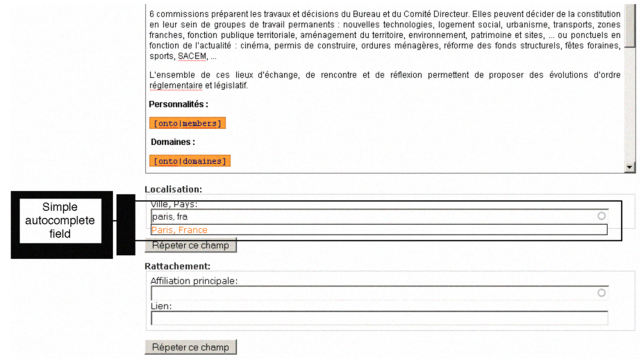

Fig. 6. Wiki editing interface

and we also created some custom extensions to meet our needs (e.g. defining classes such as `Company`). We also used GeoNames to define geographic locations as well as SKOS (Simple Knowledge and Organization Scheme), a model for defining controlled vocabularies [29], to represent a taxonomy of industrial topics. In order to create related instances, we extended our wiki engine with semantic modeling capabilities. This wiki add-on, called UfoWiki — Unifying Forms and Ontologies in a Wiki [33] — provided the ability to define form-based templates and to map them to classes and properties in the ontologies. Using this, users created and maintained instances by simply editing wiki pages and filling in forms that appeared in conjunction with the main text field for the page. For example, instead of writing *"EDF is an organization located in France"*, a user filled in a *Company* page template (mapped to our `foafplus:Company` subclass of the `foaf:Group` class) and a *Location* field (mapped to the `geonames:locatedIn` property) so that the following RDF triples would be immediately created when saving the page. An example of such an interface is displayed in Fig. 6. In addition, as we can see from this figure, our system features a live autocompletion system (based on SPARQL queries) which permits the reuse of URIs between various pages and ensures a correct interlinking of resources.

```
ex:EDF rdf:type foafplus:Company ;
  geonames:locatedIn
    <http://sws.geonames.org/3017382/> .
```

Moreover, our wiki engine features a triggering system that queries the GeoNames web service for each *Location* field in order to retrieve its URI instead of creating a new identifier for each location. This can be seen in the previous code snippet, where the URI `<http://sws.geonames.org/3017382/>` identifies the city of Paris, capital of France, on GeoNames. While users are required to type in an exact location, *e.g. "Paris, France"*, we allow the reuse of external data in our system to provide advanced browsing features, as we will describe later.

In terms of usage statistics, we studied the evolution of one of these wikis, with 18 users participating in collaborative ontology population using wiki principles. These users

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8369
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8369

contributed to more than 300 instances during a six-month period. In addition, six users were interviewed and agreed that the additional complexity of filling in forms (rather than using a plain-text wiki interface) was relatively minor compared to the various advantages and features it provides. This will be described in the next section.

C. From tagging to ontologies

In order to provide semantic tagging capabilities, we relied on the MOAT framework, a process that lets users give meaning to their tags when annotating content. This meaning is represented by the URIs of resources, thus providing a way to enable semantic indexing of tagged content [32].

While most of the generic MOAT clients[20] require users to enter URIs to define the meaning of their tags, we enhanced this approach to make it as user friendly as possible. Firstly, users are never shown any URIs as the meanings of tags are suggested via their human-readable labels. Moreover, when a user decides to link a particular resource to a tag, this choice becomes his or her default choice for that particular tag / resource relationship, making the process simpler. Finally, if no relevant meaning has been suggested, our client includes a browser that lets users navigate through a taxonomy of classes and instances in our internal knowledge base to choose another meaning for their tag, once again without seeing any URI (Fig. 7). In case no corresponding resource exists, users can create a new instance (this step is generally dedicated to wikis as we have shown before). This is also an innovative aspect of our approach: users create new ontology instances using semantic wikis, and then use these instances to define the meanings of their tags. Moreover, since these two steps can be achieved by different users, we enable a complete social process for instance management, tag meaning identification — and consequently semantic indexing.

In terms of statistics, 1,176 tags and about 17,000 instances of `sioc:Post` (and related subclasses) have been linked to 715 different URIs of resources, these links being represented via MOAT. Hence, while the process is more complex than normal tagging, users were willing to complete this additional step, the main reason being the incentives they were provided with to do so, *i.e.* the advanced search capabilities that we will describe in the next section. Finally, while we focused on a user-contributed approach for defining these links, frameworks such as FLOR (FoLksonomy Ontology enRichment [2]) may be used in combination with MOAT to automatically assist with the mappings between tags and URIs and make the process easier for users.

V. Solving the initial issues and enabling new features

A. Architecture considerations

Combined together, the different tools produce a complete interlinked RDF graph on top of which additional services can be plugged in (Fig. 5). However, as it is created using

[20]http://moat-project.org/clients

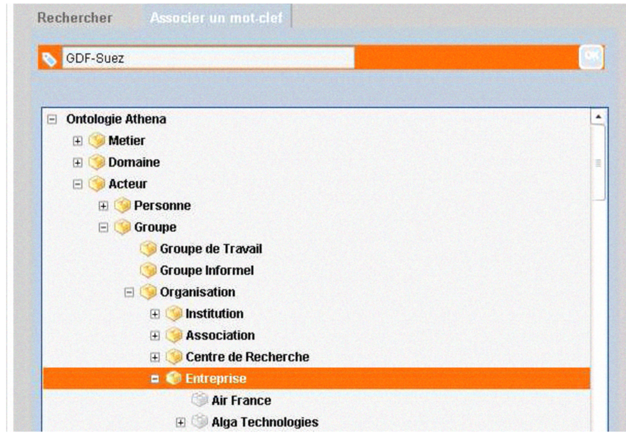

Fig. 7. Local ontology browser to assign tag meanings with MOAT

various distributed services, RDF data is then spread across the company's network. In order to efficiently take advantage of it, our architecture relies on a central triple store (RDF storage system), since it provides better performance than distributed querying (due to our industrial setting, we could not consider approaches in which several seconds were needed to answer a query). Based on our requirements (including support for the SPARQL syntax and protocol as well as management of named graphs to keep provenance of each RDF statement [15]), we chose the 3store [21] engine. In order to synchronize the store with the original RDF data, we developed an architecture based on an event-driven ping service, similar to what we proposed in [10]. The add-ons described in the previous section send a ping to the middleware access point each time new RDF data is created, updated or deleted, with this data being instantaneously stored in (or removed from) the central triple store thanks to the SPARUL `LOAD` clause that we implemented in our approach [36]. That way, each time a user participates by editing a wiki page, or creating a new blog post, the central repository is updated, taking advantage once again of the Enterprise 2.0 philosophy of voluntarily and collaboratively publishing content in an enterprise.

Then, additional services (new applications and enrichments to the original ones) have been developed and these communicate with the server using the SPARQL protocol, via HTTP. In terms of protocols, it is worth noting that this combination of SPARQL and SPARUL offers a complete abstraction layer for data integration with regards to the triple store itself, allowing us to switch to a new storage engine if needed *e.g.* for performance or scalability issues (Fig. 8).

B. Extending the wiki's capabilities

In order to solve the issues mentioned in Section II, one of the features we enabled was the use of semantic macros in the wiki system, inspired by Semantic MediaWiki [25] inline queries. They allowed us to embed the results of SPARQL queries in wiki pages — without having to face their

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8369
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8369

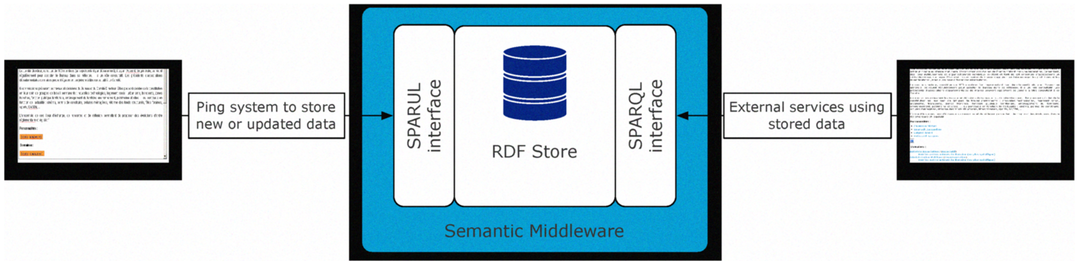

Fig. 8. Using SPARQL and SPARUL as abstraction layers for data integration

complexity. These macros are defined by wiki administrators and are mapped to SPARQL query patterns so that they can be used by end users in wiki pages via a simple grammar syntax. For example, `[onto|members]` will be translated into a SPARQL query that retrieves all the members of the currently-displayed organization. Such macros offer a way to integrate information from different wiki pages and use it to augment knowledge discovery, and several interviewed users agreed that it helped them to discover new facts about these objects (for instance, companies working in a similar domain).

Moreover, these macros do not only benefit from domain ontologies and related instances but also take advantage of the socio-structural metadata layer and of the semantic indexing capabilities of our proposal. As each wiki page is related to a particular instance, some macros are used to include a list of the latest 10 posts (as defined by `sioct:BlogPost`) that are linked to this instance (via MOAT), as well as the latest RSS items since we also provided a service that automatically indexes incoming RSS feeds using these instances.

C. Semantic search for Enterprise 2.0

In addition to this macro systems, we also developed a dedicated semantic search engine. When the user searches for a particular keyword, a SPARQL query over data in the triple store identifies related instances, using regular expressions based on the tags and their meanings (via MOAT) as well as using the `rdfs:label` (and subproperties) values of the instances. If various instances are identified, the system asks the user to select the relevant one: for example, if a user searches for *"France"*, the system asks the user to choose between *"Association des Maires de France"*, *"France"* or *"Électricité de France"*.

Once the resource has been identified, the engine lists all information about it, *i.e.* (1) the corresponding tags, (2) the main wiki page, (3) the related wiki pages (*i.e.* pages about an instance linked to the current one) and (4) every content item linked to the current resource, thanks to MOAT and the RSS indexing process. An example of such a result page is given by Fig. 9. In this way, our system solves the information fragmentation issue that we initially mentioned by providing a single entry point to access any content regarding a particular instance, identified from several distributed information sources within an enterprise. An analysis of the logs showed that more than 30 users used it over a one-month period, and user interviews confirmed that users were satisfied by the system, which was rated 3.5/5 in general.

The use of MOAT provides a clear advantage regarding the search process: we identified that 205 resources were linked to more than one tag; in fact, 39 were linked to more than four different tags. Consequently, it implies that four different tag-based queries would have been necessary to identify all the related content, while a single one is sufficient using this search engine.

D. LOD-based semantic mashups for enterprise environments

Furthermore, in addition to the previously-described services, our system provides advanced browsing interfaces and semantic mashups. In particular, we provided a faceted browsing interface using Exhibit[21] to navigate through instances from the various domain ontologies created using the wikis. The novelty of this approach is in the reuse of RDF data from GeoNames to provide a semantic mashup combining internal and external data sources, thanks to the trigger feature in UfoWiki that we explained previously.

We believe that such semantic mashups can be a significant part of the future of Enterprise 2.0 applications. Similar to how RSS allows companies to benefit from public information (news feeds, blogs, etc.), reusing RDF data brings knowledge about different topics into the enterprise for zero cost — not only for public services as described by [24], but also within intranets as in our case. Another interesting aspect of this integration is that some of these data sources have also been created collaboratively (*e.g.* DBpedia), hence allowing enterprises to benefit from the collaboration of individuals taking place on the Web.

VI. Related work

Related work can been considered regarding two main areas, the first one being middleware architectures based on Semantic Web technologies, the second one being more specifically related to Semantic Web Enterprise 2.0 information systems.

There is a vast amount of research and many commercial products in the area of Semantic Web middleware. To name but a few, OntoBroker [17], one of the first Semantic Web applications, is now available as a commercial product and mainly focuses on ways to expose various databases via

[21] http://simile.mit.edu/wiki/Exhibit/

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8369
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8369

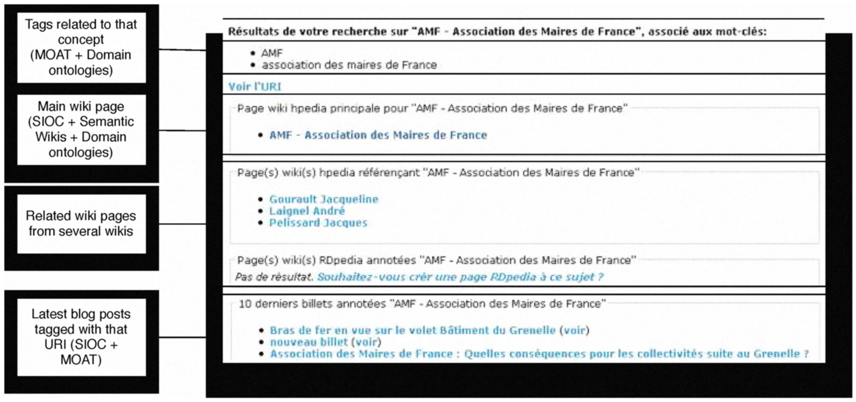

Fig. 9. Query result using our semantic search engine

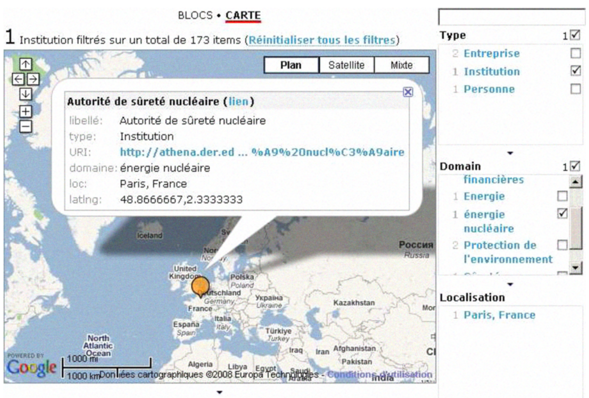

Fig. 10. A semantic mashup combining internal and external RDF data

RDF, using advanced querying capabilities. Other relevant tools include Information Integrator [1], SCORE [37] and the work discussed in [26] regarding an ontology-based knowledge management system (OKMS), all providing transparent integration and access to distributed data information sources.

However, they do not take into account the role of users as strongly as we do. Hence, they consider neither (1) modeling community members, their activities and their social networks, nor (2) the role of the user in the creation of RDF data, both topics we take into consideration within our SemSLATES approach. Our proposal thus emphasizes an original combination of social and semantic aspects for enterprise information systems — in which the collaborative aspect plays a central role, as we have detailed throughout this paper.

Some other projects also focus on enabling Semantic Web technologies for Enterprise 2.0 systems. Organik [8] shares similar goals but, regarding ontology population, it focuses on knowledge extraction rather than user-driven semantic annotations. While not restricted to enterprise environments, Talis Engage[22] and OpenLink Data Spaces (ODS)[23] also provide complete platforms for enabling social interactions within communities thanks to Semantic Web technologies. Both use SIOC and some related extensions to model the activities of their communities, and ODS also provides MOAT capabilities for semantic tagging. However, so far, they do not provide methods for creating instances of domain ontologies and are thus more limited in terms of knowledge representation.

Hence, our proposal fills a gap between generic middleware architectures for enterprise and social systems, by modeling both the social aspect of the communities and their knowledge representation aspects.

[22]http://www.talis.com/engage/
[23]http://virtuoso.openlinksw.com/wiki/main/Main/Ods

VII. Conclusion

As we have shown in this paper through the SemSLATES approach, Enterprise 2.0 ecosystems raise various issues but can successfully benefit from Semantic Web technologies. By integrating various components together and lifting existing information up to a layer that combines socio-structural metadata, domain ontologies and related instances, we demonstrated that existing issues can be solved and new browsing and querying capabilities can be imagined. In addition to the proposed principles and tools, particular outcomes from our work are (1) that the re-use of lightweight and Web-oriented ontologies allows us to efficiently deploy semantic mashups combining enterprise and Web data, while (2) using lightweight add-ons permits us to not disturb users from their existing habits and to consider them as active components of the system for producing and using semantic annotations.

While we have mainly considered Web-based applications in our case study, data from desktop systems, mobile phones or even sensor networks can be integrated into such infrastructures, providing new data sources to help build *"Social Semantic Information Spaces"* in enterprise environments. Finally, one must also keep in mind that even with the most desirable software architecture possible, Enterprise 2.0 can only be a success if people are willing to voluntarily share information together in such environments. Hence, the realization of such semantically-enhanced Enterprise 2.0 information

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8369
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8369

systems might reside more in the social than in the semantic aspect.

Acknowledgements

The work presented in this paper has been funded in part by Science Foundation Ireland under grant number SFI/08/CE/I1380 (Líon 2).

References

[1] Jrgen Angele and Michael Gesmann. Data Integration Using Semantic Technology: A Use Case. In *Proceedings of the Second International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML-06)*. IEEE Computer Society.

[2] Sofia Angeletou. Semantic Enrichment of Folksonomy Tagspaces. In *International Semantic Web Conference*, volume 5318 of *Lecture Notes in Computer Science*, pages 889–894. Springer, 2008.

[3] Sõren Auer, Chris Bizer, Jens Lehmann, Georgi Kobilarov, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007), Busan, South Korea*, volume 4825 of *Lecture Notes in Computer Science*, pages 715–728. Springer, November 2007.

[4] 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 Recommentation, W3C, February 2004.

[5] Tim Berners-Lee. Linked Data, July 2006. http://www.w3.org/DesignIssues/LinkedData.html.

[6] Tim Berners-Lee, Roy Fielding, and Larry Masinter. Uniform Resource Identifiers (URI): Generic Syntax. rfc 2396, IETF, August 1998. http://www.ietf.org/rfc/rfc2396.txt.

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

[8] Dimitris Bibikas, Dimitrios Kourtesis, Iraklis Paraskakis, Ansgar Bernardi, Leo Sauermann, Dimitris Apostolou, Gregoris Mentzas, and Ana Cristina Vasconcelos. Organisational Knowledge Management Systems in the Era of Enterprise 2.0: The case of OrganiK. In *BIS 2008 Workshops Proceedings*, volume 333 of *CEUR Workshop Proceedings*, pages 45–53. CEUR-WS.org, 2008.

[9] Chris Bizer, Richard Cyganiak, and Tom Heath. How to publish linked data on the web. http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LinkedDataTutorial/, 2007.

[10] Uldis Bojars, Alexandre Passant, Frederick Giasson, and John Breslin. An architecture to discover and query decentralized RDF data. volume 248 of *CEUR Workshop Proceedings*. CEUR-WS.org, June 2007.

[11] J.G. Breslin, A. Harth, U. Bojars, and S. Decker. Towards Semantically-Interlinked Online Communities. In *Proceedings of the 2nd European Semantic Web Conference (ESWC '05)*, Lecture Notes in Computer Science, Heraklion, Crete, Greece, May 2005. Springer.

[12] Dan Brickley and Ramanatgan V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.

[13] Dan Brickley and Libby Miller. FOAF Vocabulary Specification. Namespace Document 2 Sept 2004, FOAF Project, 2004. http://xmlns.com/foaf/0.1/.

[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] Jeremy Carroll, Christian Bizer, Patrick Hayes, and Patrick Stickler. Named graphs, provenance and trust. In *The Fourteenth International World Wide Web Conference (WWW2005), Chiba, Japan*, May 2005.

[16] Kendall Grant Clark, Lee Feigenbaum, and Elias Torres. SPARQL Protocol for RDF. W3C Recommendation, W3C, January 2008. http://www.w3.org/TR/rdf-sparql-protocol/.

[17] Stefan Decker, Michael Erdmann, Dieter Fensel, and Rudi Studer. Ontobroker: Ontology Based Access to Distributed and Semi–Structured Information. In Meersman, editor, *Semantic Issues in Multimedia Systems. Proceedings of DS–8*, pages 351–369, Kluwer Academic Publisher, 1999.

[18] Scott Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[19] Frank Smadja Grigory Begelman, Philipp Keller. Automated tag clustering: Improving search and exploration in the tag space. In *Proceedings of the Collaborative Web Tagging Workshop at WWW*, 2006.

[20] Tom Gruber. Tagontology - a way to agree on the semantics of tagging data. http://tomgruber.org/writing/tagontology-tagcamp-talk.pdf.

[21] Steve Harris and Nicholas Gibbins. 3store: Efficient Bulk RDF Storage. In *Proceedings of the First International Workshop on Practical and Scalable Semantic Systems*, volume 89 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[22] Conor Hayes, Paolo Avesani, and Sriharsha Veeramachaneni. An analysis of the use of tags in a blog recommender system. In *Twentieth International Joint Conferences on Artificial Intelligence*, pages 2772–2777, 2007.

[23] Graham Klyne and Jeremy J. Carroll. Resource Description Framework (RDF): Concepts and abstract syntax. W3C Recommendation, W3C, February 2004.

[24] Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, and Robert Lee. Media Meets Semantic Web — How the BBC Uses DBpedia and Linked Data to Make Connections. In *6th European Semantic Web Conference (ESWC2009)*, volume 5554 of *Lecture Notes in Computer Science*. Springer, 2009.

[25] Markus Krötzsch, Denny Vrandecic, and Max Võlkel. Semantic mediawiki. In *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, volume 4273 of *Lecture Notes in Computer Science*, pages 935–942. Springer, 2006.

[26] Alexander Maedche, Boris Motik, Ljiljana Stojanovic, Rudi Studer, and Raphael Volz. Ontologies for Enterprise Knowledge Management. *IEEE Intelligent Systems*, 18(2):26–33, 2003.

[27] Adam Mathes. Folksonomies - cooperative classification and communication through shared metadata. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html, December 2004.

[28] Andrew P. McAfee. Enterprise 2.0: The dawn of emergent collaboration. *Management of Technology and Innovation*, 47(3), 2006.

[29] Alistair Miles and Sean Bechhofer. SKOS simple knowledge organization system reference. Working draft, W3C, 2008.

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

[31] Alexandre Passant. *Technologies du Web Sémantique pour l'Entreprise 2.0 (Semantic Web Technologies for Enterprise 2.0)*. PhD thesis, 2009.

[32] Alexandre Passant and Philippe Laublet. Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data. In *Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008), Beijing, China*, volume 369 of *CEUR Workshop Proceedings*. CEUR-WS.org, April 2008.

[33] 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.

[34] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. W3C Recommentation, W3C, January 2008. http://www.w3.org/TR/rdf-sparql-query/.

[35] Sebastian Schaffert. IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. In *WETICE '06. 15th IEEE International Workshops onEnabling Technologies: Infrastructure for Collaborative Enterprises*, pages 388–396, 2006.

[36] Andy Seaborne and Geetha Manjunath. Sparql/update: A language for updating rdf graphs. http://jena.hpl.hp.com/ afs/SPARQL-Update.html.

[37] Amit P. Sheth, Clemens Bertram, David Avant, Brian Hammond, Krys Kochut, and Yashodhan S. Warke. Managing Semantic Content for the Web. *IEEE Internet Computing*, 6(4):80–87, 2002.

[38] James W. Tanaka and Marjorie Taylor. Object categories and expertise: Is the basic level in the eye of the beholder? *Cognitve Psychology*, 23(3):457–482, 1991.

[39] Thomas Vander Wal. Folksonomy Coinage and Definition, 02 2007. http://www.vanderwal.net/folksonomy.html.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8369
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8369