Microblogging: A Semantic and Distributed Approach

Alexandre Passant[1], Tuukka Hastrup[2], Uldis Bojārs[2], John Breslin[2]

[1] LaLIC, Université Paris-Sorbonne,
28 rue Serpente, 75006 Paris, France
firstname.lastname@paris4.sorbonne.fr
[2] DERI, National University Of Ireland,
Galway, Ireland
firstname.lastname@deri.org

Abstract. While microblogging has quickly gained a lot of interest in the Web 2.0 community, it still has not been leveraged to the Semantic Web unlike blogs and wikis. This paper describes the features, methods and architecture of a distributed Semantic Web microblogging system, as well as the implementation of an initial prototype of this concept that provides ways to leverage microblogging with the Linked Data Web guidelines.

Key words: Microblogging, Semantic Web, Web 2.0, SIOC, Data Portability, Linked Data Web

1 Introduction

Microblogging is one of the recent social phenomena of Web 2.0. It fills a gap between blogging and instant messaging, allowing people to publish short messages on the web about what they are currently doing. As a simple and agile form of communication in a fluid network of subscriptions, it offers new possibilities regarding lightweight information updates and exchange. Yet, current microblogging services are centralised and confined, and efforts are still to be made to let microblogging be part of the Social Semantic Web [5]. This is in stark contrast to blogs and wikis that can already be considered as components of the Semantic Web after a lot of work leveraging their data and metadata in machine-readable formats with projects like SIOC [6] and systems like Semantic MediaWiki [16].

In this paper, we introduce the main idea and a first implementation of distributed microblogging systems, enabled by Semantic Web technologies and providing machine-processable views of microblogging content and metadata. This way microblogging can become part of the Semantic Web as Linked Data [3]. First, we introduce classical microblogging and some of the issues it raises. Second, we see how Semantic Web can help in getting rid of these issues and what it can offer that traditional services could not achieve. Especially, we see how metadata and data can be represented using Semantic Web technologies to interlink multiple services and related datasets. Third, we describe the functions of our prototype for distributed semantic microblogging and give an overview of the current source code of the system. Finally, we conclude with ideas for future work and with thoughts on connections between this paper and projects like data portability on Web 2.0 on one hand and relationships with existing microblogging services such as Twitter[1] on the other hand.

2 Overview of microblogging

2.1 Why microblogging?

After blogging that let people put their thoughts online to an open audience, podcasting where people record it and even videoblogging (also known as vlogging) where they deliver messages in video, microblogging enabled anyone to exchange short messages within their community or simply to write in brief to the general public on the Web. This new form of blogging allows individuals to publish brief text *updates* using a multitude of various communication channels such as text messages from cell phones, instant messaging, e-mail and the Web. The simplicity of publishing such short updates in various situations and in a fluid social network based on subscriptions and response posts makes microblogging a groundbreaking communication method that can be seen as a hybrid of blogging, instant messaging and status notifications, and that some already studied from a social point of view [11]. Moreover, this way of publishing can be extended with more advanced communication means like video recording, as in Seesmic[2], which is considered a video microblogging service.

This communication method is also promising for corporate environments in facilitating informal communication, learning and knowledge exchange. Its so far untapped potential can be compared to that of company-internal wikis some years ago. Microblogging can be characterised by rapid (almost real-time) knowledge exchange and fast propagation of new information. For a company, this can mean real-time Q&As and improved informal learning and communication, as well as status notifications, e.g. about upcoming meetings and deliveries. Yet, potential for microblogging in corporate environments still has to be demonstrated with real use cases, which we hope to happen in the next years, as already was the case for blogging, wikis and other Enterprise 2.0 [12] services.

Nevertheless, microblogging is currently mostly used by technically-minded Web users and bloggers. Twitter is one of the largest microblogging services and the value of microblogging is manifested by its popularity - now ranked at website number 635 in the world - and by Google's recent acquisition of Jaiku[3], another leading microblogging service. Microblog-type publishing can also be setup on personal services, since for example the WordPress blogging software

[1] http://twitter.com

[2] http://seesmic.com

[3] http://jaiku.com

offers a dedicated template interface (Prologue[4]) that lets people publish this kind of short and real-time updates. However, there is no aggregation for personal microblogs that would take into account the special characteristics of it as a new medium.

2.2 Current issues

While microblogging gained a lot of interest on the Web and quickly became one of the main knowledge management schemes in the Web 2.0 world, like blogs or wikis, it also raises various issues.

First, most microblogging services act as closed worlds like, actually, most of Web 2.0 services: only a few of them allow interlinking with other services. For example, merging your latest blog posts or your Flickr pictures with your Twitter updates cannot be done, except using simple HTML links between them, or using RSS. RSS provides syndication, i.e. real-time export of latest updates for a given user, but cannot be used to retrieve the complete update history at any later time. Moreover, those services do not expose their metadata in a way that could be easily reused. Twitter has adopted microformats for describing *follower* (subscriber) lists, but there is no simple way to retrieve metadata about the complete updates of any user (e.g. who did the update and when). One solution would be to combine the RSS feed of latest updates with the XML export of each update. Some scripts could then map them to Semantic Web vocabularies and URIs with potential use of external data, as SWAML [9] does to find people URIs[5]. Yet, the process can be quite complex, and since it is based on RSS, only the latest updates would be available.

In addition to these metadata concerns, the content of the updates does not carry any semantics, making its reuse difficult. Twitter users have adopted certain short-hand conventions in their writing called hash tags[6], but their semantics are not readily machine-processable thus raising the same ambiguity and heterogeneity problems that tagging causes. For example, the *hash tag* `#paris` could mean various things (cities, people etc.) depending on the context, and so cannot be automatically processed by computers. This lack of data formalism also makes finding relevant content difficult. While some services provide plain-text search engines, there is no way to answer queries like *"What are the latest updates talking about a programming language"* or *"What is happening now within ten kilometres from here"*.

Finally, one issue with current services is their centralised architecture. Most services do not act in a client–server way, but require users to post their updates on a given platform, which is the same for publishing and reading data. This means that most of the time, published data belongs to the publishing site, and cannot be automatically reused on multiple microblogging sites, or even re-used locally for other purposes. It can also be a problem to private communities, since

[4] `http://wordpress.com/blog/2008/01/28/introducing-prologue/`

[5] `http://www.wikier.org/blog/using-sindice-to-get-the-best-uri-for-a-person`

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

users need to rely on an external service where they cannot completely control privacy and security.

We believe that the Semantic Web is an elegant solution to opening these data from proprietary silos and to providing machine-processable data and metadata to microblogging as well as to delivering an open and distributed environment for microblogging, as we will expose in the next section.

3 Architecture of a semantic microblogging service

3.1 Metadata modelling

In order to model the metadata of a microblogging service, we rely on two widely used ontologies on the Social Semantic Web: FOAF [7] and SIOC.

As expected, the former is used to model microbloggers and their properties (name, email etc.). Using FOAF allows the reuse of an existing URI for a person that wants to start microblogging, instead of creating a new identity URI. Moreover, since some Web 2.0 services already offer FOAF export, either directly as LiveJournal[7] or thanks to external services as Flickr [13], people can reuse their existing URI from these services without having to dig in RDF modelling. Yet, in case the person needs for any reason to create a new instance of `foaf:Person`, Linked Data principles allow them to identify uniquely with an already existing profile via a `owl:sameAs` link.

While FOAF aims to model the people aspect, SIOC is used to define the related user contexts, providing a way to identify a user account on a given microblogging service. The SIOC model provides for one person subscribing to various services, i.e. a single `foaf:Person` can be connected to various `sioc:User`. This employs the distributed architecture of the Semantic Web to enable people to consolidate their identity across a network of services.

In addition to the account aspect, SIOC is used to model the microblogging service itself and the updates of any user. In order to do so, we extended the SIOC types module[8] [4] with two new types: `sioct:MicroblogPost` and `sioct:Microblog`, as respective subclasses of `sioc:Item` and `sioc:Container`, thus allowing a Microblog to contain (using `sioc:container_of`) instances of MicroblogPost. This also provides for modelling a microblog post with the same SIOC and FOAF properties as blog posts and wiki pages. Moreover, having such a class hierarchy of SIOC types allows people to access a dataset containing a set of `sioct:BlogPost`, `sioct:WikiArticle` and `sioct:MicroblogPost` with a single SPARQL query for all the data while leaving open the option to refine their search by restricting the type of the items to be retrieved.

Thus, modelling metadata in a machine-readable format is the first step in getting rid of proprietary data silos for microblogging content as it becomes possible to merge it with other Web 2.0 content that has been described in

[7] `http://community.livejournal.com/ljfoaf`

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

RDF. We can also rely on the connection between `sioc:User` and `foaf:Person` to find relevant content, e.g. answer queries like *"List all my activity on the Web during the first week of January"*, something that could not be done with non-semantic Web 2.0 data. To a large extent, this combination of FOAF and SIOC can be used as a solution to the data portability issues[9] since it relies on machine readable and interlinked data models to represent people, user accounts and data, as shown in Fig. 1

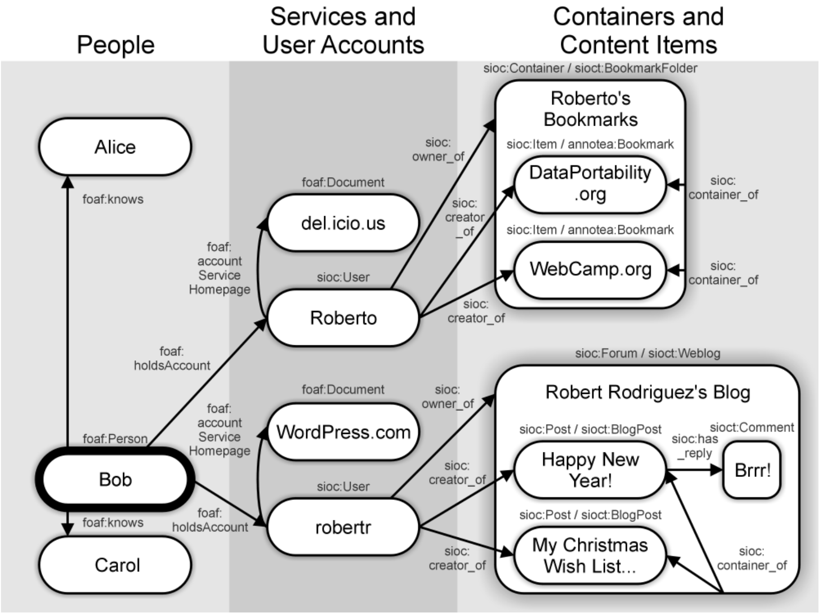


Fig. 1. FOAF and SIOC for data portability

3.2 Data modelling

While microblog posts are by nature relatively light in content, it is interesting to identify some of the data they contain, which is one of the problematic areas for current systems as mentioned earlier. While hash tags can be useful, there is a need to describe some content more formally because of the problems of plain text-only descriptions. Instead of plain text or tags, we think that using URIs and RDF to model this data can be useful for two reasons: (1) we rely on existing, unambiguous resource definitions to model the content and (2) we open microblogging entries to the Linked Data Web in the case these URIs are available on the Web and in the better case, already linked to other content, providing a path to the Giant Global Graph [2].

[9] `http://dataportability.org`

Thus, there is a need to (semi-)automatically extract those URIs or concepts from plain text or to let users annotate it similarly to what they can already do on Twitter with hash tags, but with more powerful processing that can extract and define URIs based on those tags. For example, rather than writing *"Visiting #Eiffel_Tower in #Paris"*, someone could microblog *"Visiting #dbp:Eiffel_Tower in #geo:Paris_France"* so that the processor would be able to extract the two hash tags and thanks to a predefined prefix mapping process, query DBpedia [1] and GeoNames[10] to retrieve URIs of the related concepts. Thus, the updates would be automatically linked to existing URIs rather than to simple and meaningless – from a software agent point of view – text strings.

Such a way to extract data and to interlink with existing URIs makes content more easily searchable on the Semantic Web. Indeed, thanks to lookup services such as Sindice [15] that crawl the web for RDF data and links between documents, one could be suggested to look at the update above when searching for *"Eiffel Tower"*.

3.3 Distributed content

We want the microblogging system to be open and distributed, following the spirit of the Web architecture. We envision a multitude of publishing services and aggregation servers interacting which each other. A publishing service makes the posts of one or more authors available on the Web in RDF. When a new post is available, the service pings one or more aggregation servers, defined by the user, with the URI of the post that is then retrieved by the servers. As with blogs today, we expect some people to deploy their own publishing services while others use public ones, as well as aggregation servers that can be public or dedicated to private communities of interest.

An aggregation server receives pings from publishers and retrieves posts it deems relevant for further use. The relevancy depends on the nature of the aggregation function of each server. Some servers may have a strict list of sources they aggregate while others try to provide inclusive views on the global activity on the Web. In any case, the system is open for new aggregators to provide new views. An open question is how publishers decide which aggregators to ping and whether publishers should let aggregators subscribe to them.

3.4 Distributed aggregation

Aggregators function as super-peers in the network, taking the burden of following publishers off the readers and making it simpler for publishers to announce new posts. Pinging is essential to meeting the timeliness requirement without excessive polling. In this sense it is a push technique, but since the posts are already published, pinging results in a simpler interface and makes it cheaper for aggregators to disregard posts from irrelevant sources.

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

If an aggregator disconnects and returns to the network, it may have missed pings. In this situation, it may make sense to poll known publishers for new content. This would be a typical situation for personal aggregators. Further, aggregators would be able to crawl and readers to browse more posts as long as the Linked Data principles are followed.

We can even envision intelligent readers, that will accept new posts only if they are linked to relevant URIs. For example, we could setup a "Travel microblogging" server that will accept only posts that contain links to one or more URIs from the GeoNames dataset.

3.5 Users own their data

As a consequence of the distributed nature of the system, one feature of our architecture is that people can really own their data. By self-hosting a publishing service and then publishing to a microblog aggregator server, they keep all their updates even if one service closes. Moreover, by hosting their data, people can reuse it in other applications, including future microblogging servers they want to publish to and any Semantic Web applications. They can also mash it up with other RDF data they own or that is publicly available on the Web, or in case of corporate microblogging, in their organisation.

We think that this feature is really important, especially from a user rights and data portability point of view on the Web, following some thoughts that have been expressed in "A Bill of Rights for Users of the Social Web" [14]. Combining the distributed architecture and this data ownership and reuse aspect, Fig. 2 provides an overview on the complete architecture of the process.

3.6 Security and privacy issues

The open and distributed nature of the architecture complicates the authentication requirements in some use cases. It is easy to publish posts in someone else's name or fill a public aggregator with spam. Moreover, aggregators may need to authenticate to publishers if the posts are for a restricted audience only.

One solution is to require publishers to register using OpenID on an aggregator server. The server delivers each registrant an API key (a password) for publishing their content on that server. Relying on OpenID allows servers to automatically discover the FOAF profile and the URI of a user[11] as long as the OpenID provider can offer FOAF autodiscovery[12].

Combined with the use of the `foaf:openid` property that was recently introduced in the FOAF specifications, this is a way to provide a lightweight authentication and security layer, since the server can ensure that someone publishing on it is really the person identified by the FOAF URI. Of course, one can deliver false information in a fake FOAF profile, thus additional strategies

[11] `http://apassant.net/blog/2007/09/23/retrieving-foaf-profile-from-openid/`

[12] `http://wiki.foaf-project.org/Autodiscovery`

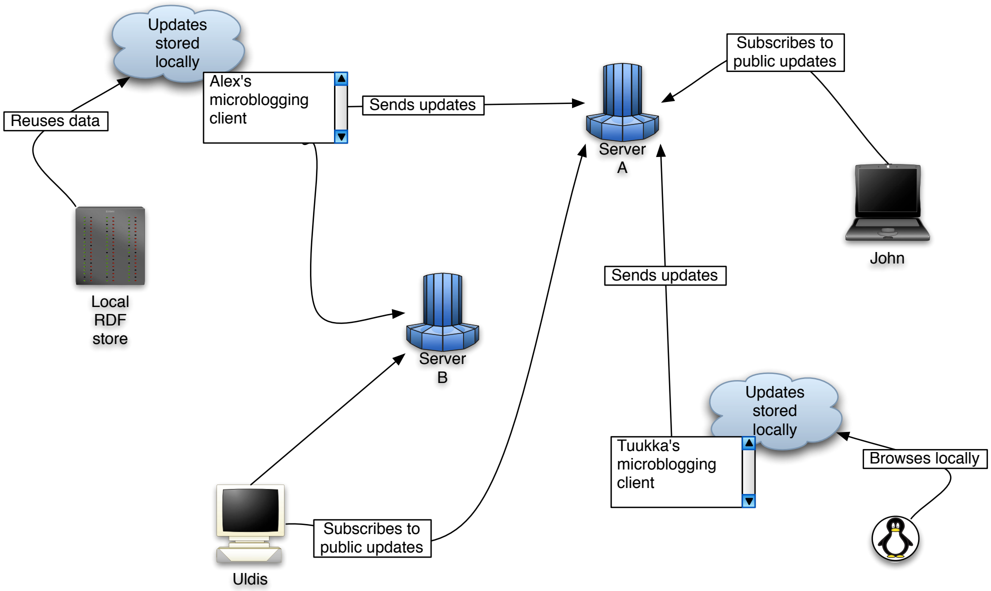


Fig. 2. Global architecture of distributed semantic microblogging

such as a network of trust between community members or graph signing[13] with public-key cryptography (PGP) should be considered.

4 A prototype: SMOB

In order to demonstrate our thoughts, we have implemented SMOB[14], a prototype of the publishing client and server web services for semantic microblogging.

4.1 Publishing Content

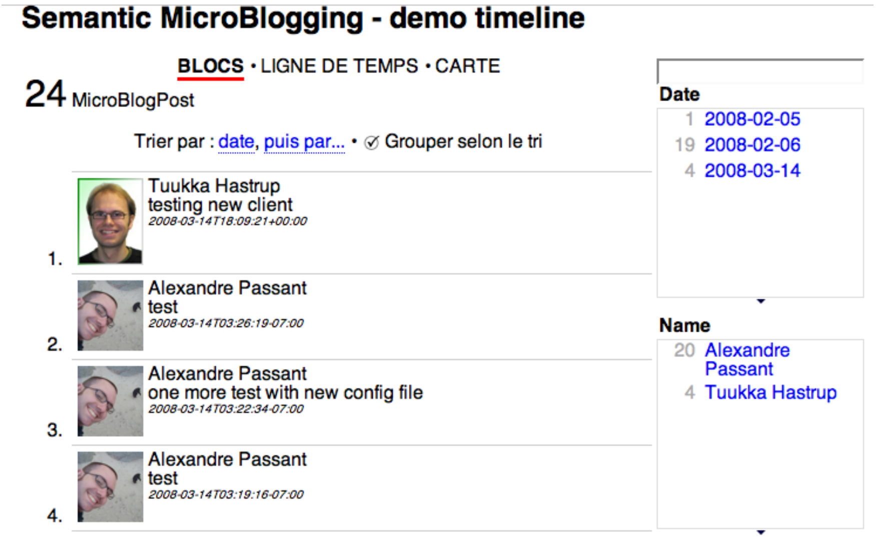
The publishing client is accessed as a web page that contains a small form field for content. Submitting the form creates the post and makes it available on the Web in RDF. Further, there are checkboxes for choosing which of the configured aggregation servers are pinged about the new content, so that within the same client, a user can decide that some update will ping a server while another update will ping another server.

The publishing client is configured for its location, the list of servers it can ping and the `foaf:Person` URIs of the author and related file. An existing FOAF URI can be reused for this service, thus providing it a new `sioc:User` account for this URI.

[13] `http://usefulinc.com/foaf/signingFoafFiles`
[14] `http://code.google.com/p/smob/`

4.2 Reading content

Based on pings received from clients, the server loads all posts into its triple store using SPARUL. SPARUL[15] (or SPARQL/Update) is an update language for RDF graphs and currently implemented in Jena, OpenLink Virtuoso as well as partially in ARC2 within its SPARQL+ support. The server uses the `LOAD` instruction to load all statements for any incoming URL of an RDF file (i.e. of a microblogging item) into the triple store. Then, people can browse and read the posts using a web interface that implements faceted browsing as shown on Fig. 3. The posts are available as a sortable and groupable list, as an ordered list, a timeline and on a map. We rely on Exhibit [10] to provide this interface, and facets are created using metadata (author and date) but also data extracted from the semantic hash tags as described before. Currently, we have implemented two of those facets: (1) locations, which are mapped with the Google Maps view of Exhibit thus providing a user-friendly geolocation interface for microblogging (Fig. 4), and (2) topics, which can be based on DBPedia URIs.



Semantic MicroBlogging - demo timeline

BLOCS • LIGNE DE TEMPS • CARTE
Trier par : date, puis par... • ✓ Grouper selon le tri

24 MicroBlogPost

1. Tuukka Hastrup
testing new client
2008-03-14T18:09:21+00:00

2. Alexandre Passant
test
2008-03-14T03:26:19-07:00

3. Alexandre Passant
one more test with new config file
2008-03-14T03:22:34-07:00

4. Alexandre Passant
test
2008-03-14T03:19:16-07:00

Date

1	2008-02-05
19	2008-02-06
4	2008-03-14

Name

20	Alexandre Passant
4	Tuukka Hastrup

Fig. 3. Latest updates rendered in Exhibit

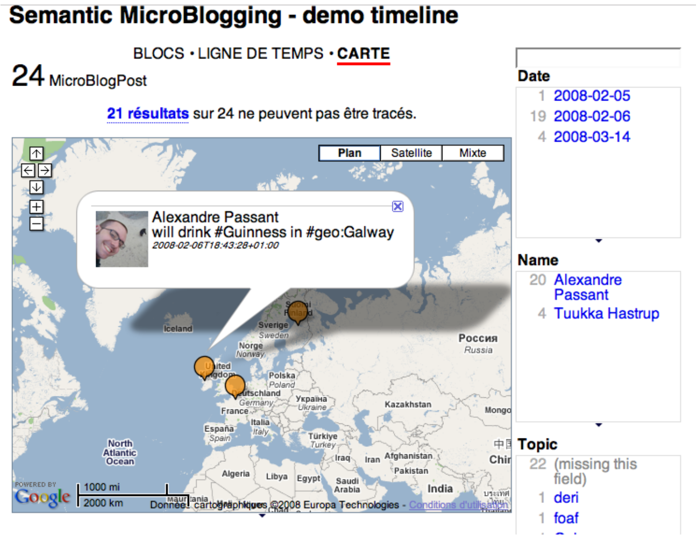
4.3 Code overview

Our client is a simple 57-line PHP page that presents a submission form and handles the received content. The content is wrapped in an RDF-XML document using SIOC PHP Export API[16] and saved as a file locally. The URI of the file

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

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

is then sent to the server(s) as a HTTP GET ping using CURL. The SIOC PHP Export API implementation is 678 lines of PHP but fairly generic, and is already used on other prototypes such as the DotClear weblogging exporter and the currently in-development SIOC plugin for VBulletin forums. Using the API offers a way to update with zero-cost to new version of the ontology in case it changes.



Semantic MicroBlogging - demo timeline

24 MicroBlogPost

21 résultats sur 24 ne peuvent pas être tracés.

Alexandre Passant
will drink #Guinness in #geo:Galway
2008-02-06T18:43:28+01:00

Fig. 4. Map view of latest updates with Exhibit

The server uses ARC2[17] to store the data, since it supports the `LOAD` instruction in SPARQL+[18], and relies on a single SPARQL query to render the JSON file needed for Exhibit (assuming properties are single-valued):

```
SELECT ?post ?date ?content ?maker ?name ?depiction
WHERE {
  ?post rdf:type sioct:MicroblogPost ;
    foaf:maker ?maker ;
    sioc:content ?content ;
    dct:created ?date .
  ?maker foaf:name ?name .
  { ?maker foaf:img ?depiction } union
  { ?maker foaf:depiction ?depiction }
} ORDER BY DESC(?date) LIMIT 20
```

The preprocessor for hash tags uses simple regular expressions and mappings between prefixes, and URIs and services are mapped internally. It is less than

[17] http://arc.semsol.org

[18] http://arc.semsol.org/docs/v2/sparql+

100 lines of code, excluded libraries, and can be easily deployed in shared hosting environments, so that people can set-up their own service for their community.

5 Conclusions and future work

In this paper, we introduced the architecture and a first implementation of a distributed semantic microblogging platform. While existing approaches to convert microblogging services to RDF already exist for Twitter[19] or Jaiku[20], our approach relies on a complete open and distributed view, using some standards of the Social Semantic Web. Moreover, some parts of our work, as the hash tag processing could be adopted to services such as Twitter to enable some semantics in existing tools.

Some issues still remain to be resolved, such as data privacy, private aggregation communities, and building personalised views and aggregation services for public updates. For this latter point, we can imagine personal aggregators based on the `foaf:knows` list of a user to automatically accept or reject new updates. More generally, and since this field is quite new to the Semantic Web, we think that microblogging can be one of the next steps of semantically-enhanced blogging systems [8].

Acknowledgements

This material is based upon work supported by Science Foundation Ireland under grant number SFI/02/CE1/I131. The authors of would like to thank Richard Cyganiak of DERI Galway for his valuable comments about this paper.

References

1. Sören Auer, C. Bizer, G. Kobilarov, Jens Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. *6th International Semantic Web Conference, Busan, Korea*, 2007.
2. Tim Berners-Lee. Giant Global Graph. `http://dig.csail.mit.edu/breadcrumbs/node/215`, November 2007.
3. 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/`, 20 July 2007.
4. Uldis Bojārs, John Breslin, Aidan Finn, and Stefan Decker. Using the Semantic Web for Linking and Reusing Data Across Web 2.0 Communities. *The Journal of Web Semantics, Special Issue on the Semantic Web and Web 2.0 (Forthcoming)*, 2008.

[19] `http://sioc-project.org/node/262`

[20] `http://sioku.sioc-project.org/`

5. John G. Breslin and Stefan Decker. Semantic Web 2.0: Creating Social Semantic Information Spaces. In *Tutorial in the 15th International World Wide Web Conference (WWW 2006)*, Edinburgh, Scotland, May 2006.
6. John G. Breslin, Andreas Harth, Uldis Bojars, and Stefan Decker. Towards Semantically-Interlinked Online Communities. In *Proceedings of the Second European Semantic Web Conference, ESWC 2005, May 29–June 1, 2005*, Heraklion, Crete, Greece, 2005.
7. Dan Brickley and Libby Miller. FOAF Vocabulary Specification. Namespace Document 2 Sept 2004, FOAF Project, 2004. `http://xmlns.com/foaf/0.1/`.
8. Steve Cayzer. What next for semantic blogging.from visions to applications. OCG Verlag, 2006.
9. Sergio Fernández, Diego Berrueta, and Jose E. Labra. Mailing lists meet the semantic web. In Dominik Flejter, editor, *Procedings of SAW2007 Workshop*, pages 45–52, 2007.
10. David Huynh, David Karger, and Rob Miller. Exhibit: Lightweight structured data publishing. In *16th International World Wide Web Conference*, Banff, Alberta, Canada, 2007. ACM.
11. Akshai Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: Understanding microblogging usage and communities. Jul 2007. Simple description of Twitter.
12. Andrew P. McAfee. Enterprise 2.0: The dawn of emergent collaboration. *MIT Sloan Management Review*, 47(3):21–28, 2006.
13. Alexandre Passant. :me owl:sameas flickr:33669349@n00 . In *Linked Data on the Web (LDOW2008)*, 2008.
14. Joseph Smarr, Marc Canter, Robert Scoble, and Michael Arrington. A bill of rights for users of the social web. http://opensocialweb.org/2007/09/05/bill-of-rights/, 4 September 2007.
15. Giovanni Tummarello, Renaud Delbru, and Eyal Oren. Sindice.com: Weaving the open linked data. In *ISWC/ASWC*, pages 552–565, 2007.
16. Max Völkel and Sebastian Schaffert, editors. *SemWiki2006, First Workshop on Semantic Wikis - From Wiki to Semantics, Proceedings, co-located with the ESWC2006, Budva, Montenegro, June 12, 2006*, volume 206 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006.