SEVENTH FRAMEWORK PROGRAMME
FP7-ICT-2009-6

BlogForever
Grant agreement no.: 269963

D2.3 BlogForever Report: Weblog Ontologies

Editor:	A. Cristea, M. Joy
Revision:	0.1
Dissemination Level:	Public
Author(s):	Hendrik Kalb, Yunhyong Kim, Paraskevi Lazaridou
Due date of deliverable:	31 May 2012
Actual submission date:	31 May 2012
Start date of project:	01 March 2011
Duration:	30 months
Lead Beneficiary name:	Technische Universität Berlin (TUB)

Abstract: This report outlines an inquiry into the area of ontologies, conducted within the context of blog preservation, management and dissemination. Three different scenarios regarding the application of ontologies are presented and evaluated including interoperability with Linked Open Data, Semantic Extension of Tags and utilisation of Microdata, Microformats and RDFa. This report provides an insight into some of the most promising applications of ontologies that should be further examined and implemented within the BlogForever project.

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)

The **BlogForever** Consortium consists of:

Aristotle University of Thessaloniki (AUTH)	Greece
European Organization for Nuclear Research (CERN)	Switzerland
University of Glasgow (UG)	UK
The University of Warwick (UW)	UK
University of London (UL)	UK
Technische Universitat Berlin (TUB)	Germany
Cyberwatcher	Norway
SRDC Yazilim Arastrirma ve Gelistirme ve Danismanlik Ticaret Limited Sirketi (SRDC)	Turkey
Tero Ltd (Tero)	Greece
Mokono GMBH	Germany
Phaistos SA (Phaistos)	Greece
Altec Software Development S.A. (Altec)	Greece

History

Version	*Date*	*Modification reason*	*Modified by*
0.1		initiation	Hendrik Kalb (TUB)
0.2		extension	Yunhyong Kim (UG)
0.3	18.07.11	extension	Yunhyong Kim (UG)
0.31	19.07.11	extension	Hendrik Kalb (TUB)
0.32	21/07/11	Inclusion of feedback (from Karen Stepanyan), modification	Yunhyong Kim (UG)
0.33	27.07.11	Inclusion of FOAF and minor changes to the format	Hendrik Kalb (TUB)
0.4	28.07.11	Some modifications to section 3 ("ontologies that support..."), including inclusion of further ontology descriptions	Yunhyong Kim (UG)
0.5	01.08.11	Inclusion of OPO, Tag Ontology, SCOT, MOAT and some modifications	Hendrik Kalb (TUB)
0.6	19.03.12	Complete revision of the document according to the revised approach of the BF Weblog Ontology Task (adapted structure, elimination of obsolete chapters)	Hendrik Kalb (TUB)
0.7	09/05/12	Further revision to refine and extend the document. Mostly related to Chapter 3.	Yunhyong Kim (UG)
0.9	25/05/12	Finalisation of a first draft.	TUB & UG
	30/05/12	Review and minor corrections	UW
1.0	31/05/12	Finalising the deliverable.	TUB, UG, AUTH
1.1	31/08/12	Extension of Chapter 4.	Paraskevi Lazaridou (TUB)
1.2	05.10.12	Minor changes according to the internal review.	TUB, UW, Tero

Table of Contents

Executive Summary

Ontologies, in the context of this report, describe a "formal, explicit specification of shared conceptualisations" (e.g. see Guarino, Oberle, & Staab, 2009; Sowa, 2000; Studer, Benjamins, & Fensel, 1998) that allow an agent (whether a person, a machine, or an application) to interpret and reason with information. The specification is intended to be explicit and machine-readable. Therefore, a formal description language is used instead of natural language. The conceptualisation represented in such an ontology typically describes a shared understanding (consensual knowledge) of a group, not a personal understanding (Studer et al. 1998).

Based on the description of several objectives that can be addressed by the application of ontologies, this report examines the following three scenarios:

- Facilitation of the interoperability among BlogForever repositories as well as between a BlogForever repository and other digital libraries, through the exposure and linking of data including explicit semantics.
- Enriching user-generated tags with explicit semantics in order to struggle problems of free indexing, like the absence of hierarchical relationships among the tags or homonym and synonym ambiguities, which increase due to the aggregation of tags from various blogs.
- Improving the quality of blog crawling by the utilisation of available explicit semantic markups inside the HTML representation of blogs.

To facilitate the interoperability with other repositories, the adoption of the Linked Open Data (LOD) concept is proposed, to expose preserved data from BlogForever repositories and their relationships as openly available data with an explicit machine-readable semantic. Thereby, different BlogForever repositories would be able to interoperate easily on a data level and can also be integrated with other digital libraries, like databases for scientific publications, to enable search queries and navigation that are not limited to the authors and publications available in the blog repositories. Additionally, the data can be linked to other repositories in order to connect terms to publically available definitions and descriptions of these terms. For example, the topic of a blog or blog post can be linked to a definition in Wikipedia or other repositories. Based on the exposed data, new relationships can be easily created and expressed (e.g. with SPARQL). Furthermore, the development of third party applications is facilitated through the openly available standardized format and the unique identifier for each object.

The enriching of user-generated tags with explicit semantics leads to the integration of an ontology. The semantic enrichment of crawled tags after the crawling process is a field of on-going research from different perspectives. A methodology to identify relations between the tags of weblog posts by revealing the associations of tags in the repository data and exploiting also available lexical databases is proposed. For the explicit description of tag semantics, existing vocabularies were examined. The SCOT ontology was found to be the most promising because it is able to not only model the relationships between tags, tagged resource, and tag creator but can further handle different meanings in different tag clouds, e.g. in different blogs. Apart from SCOT, there are also other available ontologies like Tag Ontology and MOAT that can be useful in the description of semantics. Therefore, these ontologies should be chosen if tag semantics should be expressed explicitly in the BlogForever repository.

For the utilisation of explicit semantics in HTML pages for data extraction, the three initiatives of microformats, microdata, and RDFa have been examined regarding their adoption in webpages and specifically in blogs. Microformats is currently the most adopted format while microdata will probably be the most adopted in the future. Therefore, both should be prioritized against RDFa. An evaluation of microformats and microdata vocabularies regarding their semantic compliance with

the data model shows that processing these data during the crawling process will probably further improve the quality of crawled data.

In summary, the application of ontologies is promising for various purposes in the BlogForever project. This report gives an insight into some of the possibilities that should be further examined and implemented in the BlogForever project. Even if the described scenarios are limited mainly to the ability of expressing semantics explicitly using shared vocabularies, additional functionalities like automatic reasoning will also be possible.

1 Introduction -What is an ontology

The term *ontology* has its roots in philosophy as the study of the nature of existence and reality. It has often taken the form of categories that describe reality and their relations (e.g. Aristotle's Categories[1], and Ramon Llull's Tree of Science[2]). The terminology has been inherited by many areas in information sciences today, including knowledge representation, artificial intelligence, and the semantic web to describe a "formal, explicit specification of shared conceptualisations" (e.g. see Guarino, Oberle, & Staab, 2009; Sowa, 2000; Studer, Benjamins, & Fensel, 1998) that allow an agent (whether a person, a machine, or an application) to interpret and reason with information. The specification is intended, in this context, to be explicit and machine readable, achieved by using a formal description language (e.g. OWL[3]) instead of a natural language. While the conceptualisation that is represented in such an ontology typically describes a shared understanding (consensual knowledge) of a group and not personal understanding (Studer et al. 1998) recent efforts have also included approaches to extract personalised ontologies[4].

While we have presented concepts as the subject matter of ontologies, the notion of "concept" itself is ambiguous (B. Smith 2004). A concept can represent anything from a physical object (e.g. "chair") to a complex event, activity, task, and/or process involving several objects. Process models are increasingly used to represent information from interactive environments such as interactive multimedia performances, business, scholarly, and administrative processes, and social network (e.g. wikis, blogs, twitter) dynamics.

Ontologies can be distinguished on several levels: for example, ontologies can be characterised by how specific it is to a subject domain (e.g. biology), they can be characterised by the task or application that they aim to support, and they can be characterised by whether the ontology is accessible only to a selected group, system, or application (back-end), or explicit to the end-user of the system (front-end). For example, top-level or foundational ontologies (e.g. Lenat, 1995) are subject domain independent and should represent a broad view of the world that can be applied to many different domains. Reference or core ontologies[5] are specific to a domain and either derived from foundational ontologies or built from highly reusable concepts in the domain.

It has been stated that ontologies for selected applications are less re-usable because they are modelled for direct use in an application (e.g. reasoning engine) (Zemmouchi & Ghomari 2009; Guarino et al. 2009). However, if the application is built to support activities common across many organisations (e.g. multilingual access to collections, library sciences, digital preservation) and the ontology is adequately documented and exposed, it could alleviate unnecessary labour (e.g. see McHuh & Lalmas, 2010) and facilitate interaction by assisting information search (cf. Navigli & Velardi, 2003; Zhuhadar, Nasraoui, Wyatt, & Romero, 2010), and navigation[6].

There are ontologies designed for selected applications that also intersect with domain core ontologies or foundational ontologies. For example, for the Chat-80 question-answering system, David Warren and Fernando Pereira designed an ontology for a microworld of geographical

[1] http://plato.stanford.edu/entries/aristotle-categories/

[2] http://quisestlullus.narpan.net/eng/713_arbre_eng.html

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

[4] http://www.sciweavers.org/publications/extracting-personalised-ontology-data-intensive-web-application-html-forms-based-revers

[5] http://www.cs.man.ac.uk/~stevensr/menupages/background.php

[6] http://www.slideshare.net/sa.intui/ontological-navigation-pdf

concepts (Warren & Pereira 1982), and applications in word sense disambiguation (cf. R. Navigli, 2009).

The increasing number of fields and projects that investigate or apply ontologies causes different understandings to arise with respect to what constitutes an ontology. For example, it is sometimes controversial whether a simple vocabulary can be considered as ontology or not. This report adopts a broad view that accepts as ontologies simple catalogues of terms as well as complex systems (see Figure 1) with logical rules and constraints (B. Smith & Christopher Welty 2001).

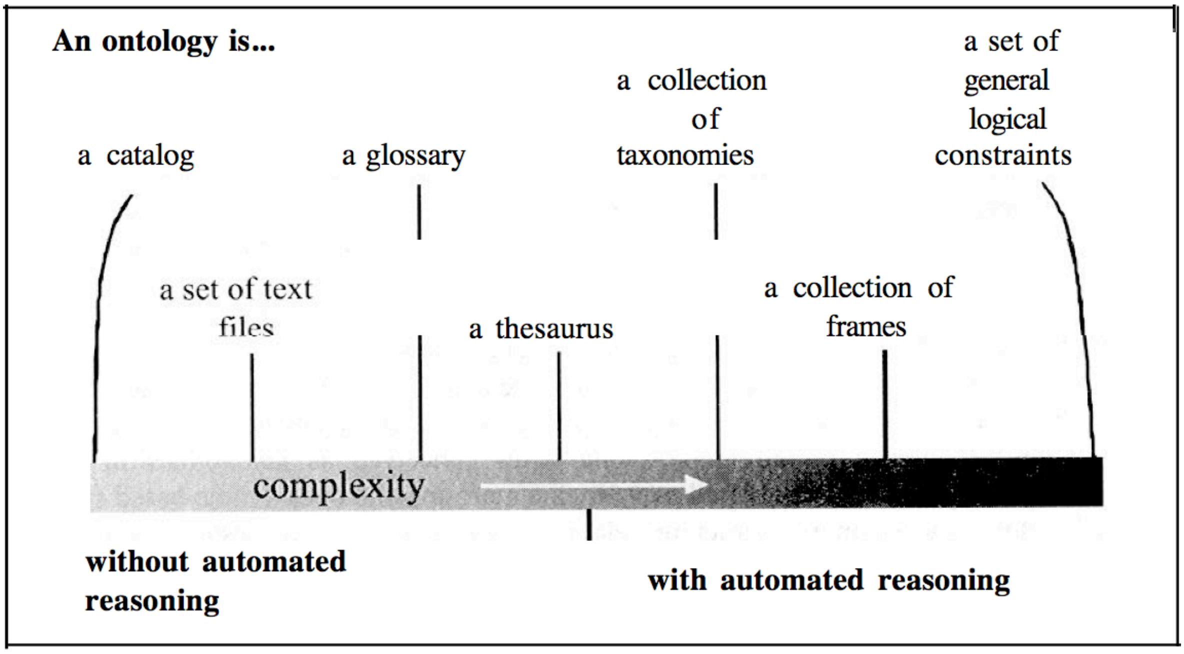


Figure 1: Different kinds of ontologies (B. Smith & Christopher Welty 2001)

The main aim of this report is to describe scenarios for the application of ontologies that might be relevant to a repository of blogs, to identify relevant ontologies within this context, and to provide a development process for selected scenarios that would lead to valuable ontology extensions for the BlogForever repository prototype. In order to do this, we first need to identify core objectives and possible use cases for the ontologies with respect to BlogForever to be used in assessing the relevance of a given ontology. Preferably, this should be done in a way that answers the following questions:

1. What are the high level objectives with respect to ontologies that we are aiming to achieve within BlogForever?[7]
2. What are the specificity and scope of the ontologies required to meet these objectives?
3. To meet the objectives, what are the applications (both back-end and front-end) that we could support with the ontologies internally (e.g. in terms of administration and service – say, for preservation activities and/or semantic search) and externally (e.g. for interoperability with other archives or social network initiatives)?
4. What characteristics of the blogs should the ontology capture in order to achieve 1, 2, and 3?

[7] The Description of Work indicates the main objectives. They are considered and refined in chapter 2.

Last but not least, observe that ontologies can be created and populated in a variety of ways: some are created automatically from natural language texts, semi-structured texts, and/or previously assigned metadata. Others are created by expert knowledge or external knowledge sources such as Wikipedia[8]. They are also expressed using different syntaxes and grammar. In selecting an ontology, we need to consider how we might create and populate the adopted ontology, and whether it can be expressed in a language suitable for our purposes, i.e. we need to consider aspects of implementation feasibility, before proceeding.

In the next chapter (Chapter 2), we discuss the scope of objectives that we will be examining with respect to ontologies within BlogForever. While these objectives present the first step towards narrowing down the avenues of investigation we propose to undertake, the manifold possible applications of ontologies remain (theoretically speaking) limitless even within the objectives specified. Given the finite resources available within the current project, in Chapters 3 to 5, we have opted to further zoom into selected scenarios that will be examined as potentially the most immediately valuable and viable application context for ontologies with respect to the repository prototype. The remaining chapters will be dedicated to employing concrete ontologies in the weblog context and demonstrating their potential.

It should be mentioned that the results of this report are intended as a *proof of concept* to explore the potential of ontologies and develop a possible application framework with respect to weblog repositories. The study is not intended as a conclusive evaluation of ontologies in general. By conducting this initial study and pilot test, it is intended that the potential for further research, implementation, and refinement in this area be opened up.

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

2 Objectives – why use an ontology?

In the following, we identify and describe objectives for the use of ontologies in BlogForever. The identification of objectives is necessary for the evaluation of possibly relevant ontologies later in the document. The consideration of objectives for ontology use has been performed in parallel to the user requirement analysis in the BlogForever project[9]. However, there is an overlap between the ontology objectives and user requirements that can facilitate the design and implementation process of the weblog spider and repository. Therefore, several references to specific requirements in the remainder of this chapter show some of the connections but further considerations should be undertaken in the design tasks.

The aim of BlogForever is the provision of complete and robust digital preservation, management and dissemination facilities for weblogs[10]. Therefore, each use of an ontology in the project should contribute to this aim. The objectives for ontology use can further be described on different levels and with regard to different dimensions.

On the very top level, the BlogForever weblog ontology is intended to provide a thorough description of an ontology-based representation of the domain[11]. This representation together with the weblog data model is intended to support[12]:

- effective data mining
- efficient preservation
- robust repository features

These high level objectives are not disjunctive or competitive. Therefore, a function or objective on a lower level can support more than one high level objective but should address at least one.

Each of the high level objectives can be understood from an internal or external perspective. Once the archive has been filled with some data, an ontology can support efficient data mining with respect to the archived data for internal semantic analysis and end-user services. However, semantic description of blogs can also be used to mine and select external data to assess their suitability for inclusion into the repository before they are archived (e.g. to identify potential blogs for a specific domain and extract target components that conform to the data model). Internally, an ontology can support efficient preservation planning such as risk assessment with respect to archival holdings, and the automation of preservation processes that involve several knowledge bases, while, externally, an ontology could help in assessing priority with respect to preserving selected blogs or blog components. External repository features could include semantic search linked to other repositories (in contrast to an internal semantic search that would be limited to internal resources) or a machine readable semantic access by applications outside of the archive. Internal features can be the provision of a semantic access to the data that supports data management and enhanced services that provide analysis of complex characteristics and dynamics related to blogs held within the archive.

The use of ontologies can be categorised by the types of objects that are described semantically. For example, we can distinguish between the blog itself (posts, comments, author etc.), the network of

[9] See BlogForever Deliverable 4.1: User Requirements and Platform Specifications Report.

[10] Section “B 1.1.2 Objectives”, Part B, Description of Work, p. 3.

[11] Paragraph 1 of Task 2.2 description of WP2, Subsection “Description of work and role of partners”, Section “WT3: Work package description”, Part A, Description of Work, page 7 of 25.

[12] Paragraph 2 of Task 2.2 description of WP2, Subsection “Description of work and role of partners”, Section “WT3: Work package description”, Part A, Description of Work, page 7 of 25.

blogs and actors (links, answers, ties, etc.), and the content of the blog (e.g. for a specific domain)[13]. All these perspectives can be conceptualised in a different way using ontologies.

A simple process that is composed of aggregating blogs (input), management, and external access/distribution (output) can describe the preservation of blogs in the archive. The use of ontologies can be supportive in each step of the process.

On an operational level, we can identify several functions of the archive that can be facilitated by the use of ontologies. These functions relate to interoperability (between archives as well as with external agents), archival functions, policy-based management, semantic search and navigation, merging of social web aspects with ontologies, and additional ontology-based services. The integration of these functions as a unified service can at times be referred to as semantic portals.

The solutions developed as part of the BlogForever project may, in fact, resemble the semantic portals for publishing cultural heritage (Hyvönen 2009). Hyvönen (ibid.) highlights the value of ontologies, and particularly domain ontologies, for annotating the content. However, such portals may also use other ontologies (instance-rich ontologies) that enable description of persons, organisations or geographical locations. While the expected solutions provided by the BlogForever project may resemble a semantic portal, the evaluation of relevant ontologies summarised in this report is not limited to only the types or instances of ontologies commonly used in semantic portals.

The BlogForever ontology will reflect the understanding of weblogs as[14]: physical phenomena, logical encodings, conceptual objects that have meaning to humans, structural objects of networked discourse and collaboration for knowledge creation in large groups of humans, sets of essential elements that must be preserved in order to offer future users the essence of the object, and ontologies created in a bottom-up manner by communities rather than specialists. The final BlogForever ontology aims to merge aspects of semantic web with the social web, draw information from weblog data, metadata and user generated folksonomy[15].

In this document we would like to re-cast the operational level objectives of the BlogForever archive with respect to ontologies to be as follows:

The ontologies should be selected to:

1. **Promote interoperability between BlogForever and other archives and social network initiatives (to support sustainability and robustness of the repository)**:
    - Enabling the possibility of enriched exposure of our metadata and data model to an outside agent[16].
    - Allowing for efficient dissemination of material, succession plan, and shared applications between archives[17].

[13] For the elements of blogs see also BlogForever Deliverable 2.2: Weblog Data Model, Chapter 9: Blog Data Model, pages 44-56.

[14] Section “B 1.1.2.1 Study weblog structure and semantics” , Part B, Description of Work, page 38.

[15] Paragraph 3 of Subsection “c) Steps towards extraction of the Weblog abstract data model and the Weblog ontology- based representation” , Task 2.2 description for WP2, Section “B 1.2.1.1 Detailed work description”, Part B, Description of Work, page 60.

[16] See also the requirement IR3: Export data using OAI-PMH protocol and Dublin Core schema in the BlogForever Deliverable 4.1: User Requirements and Platform Specifications Report.

[17] See also the requirement IR6: Facilities to enable interoperability in the BlogForever Deliverable 4.1: User Requirements and <<<<<< Report.

2. **Facilitate management functions (e.g. preservation activities) within the archive**:
   - Providing a more transparent, explicit and formal statement of policies, objectives, and infrastructure.
   - Supporting archival functions (ingest, access control, storage management, technical and organisational infrastructure, policy implementation and management, risk assessment and management, preservation planning and management)[18].
3. **Expose complex weblog semantics and characteristics for data mining**:
   - Forming a collection level organisational structure based on concepts and relations as a basis for encouraging semantic search with respect to weblogs[19].
   - Capturing weblog characteristics beyond concepts, e.g. with respect to dynamics, interactions, and networks[20].

To support the first objective it is recommended that the ontology is selected to support: general applicability, extensibility, and wide scale adoption within the weblog domain. The selection of a general foundational formal ontology that is applicable to a wide range of data types, accompanied by a weblog specific solution might be appropriate for this purpose. The foundational ontology selected ideally should be compatible with other ontologies that might be selected or constructed to support the second and third objective.

To support the second objective, the ontology must: go beyond concepts representing data content, i.e. include relationships between agents, objects, rights, policies, mandates, risks, events and changes that take place as part of the daily management of the archive. The ontology should ideally facilitate querying for provisions related to archive policies and support automation of management processes where possible.

To support the third objective, the selected ontology should ideally be able to: predict and model the dynamics of the blog environment: for example, how one event might trigger another event, changes with respect to objects and networks (within the spatio-temporal domain). It should strive to account for networks that form spontaneously and passively, and make sense of scattered instances of micro-processes. Note, that in the weblog environment, there is often no formal establishment of networks or activities. High level descriptions comprising activity, event, or business process ontologies may not be adequate to capture the complexity involved in passive formation of networks.

The different levels of objectives and dimensions for the use of ontologies are integrated in Table 1. It should be possible to associate at least one cell of each row to every proposed use of an ontology. Otherwise, a reconsideration might be needed on whether the objectives are not fully or adequately described or whether the proposed use is out of the scope of the project.

[18] See also the requirement OP2: OAIS in the BlogForever Deliverable 4.1: User Requirements and Platform Specifications Report.

[19] See also the requirement FR26 – Context-sensitive search by keyword in the BlogForever Deliverable 4.1: User Requirements and Platform Specifications Report.

[20] See also the requirement UI27: Dynamic network view on topics, blogs, posts, etc. in the BlogForever Deliverable 4.1: User Requirements and Platform Specifications Report.

Table 1, BlogForever objectives and dimensions for the use of ontologies

BlogForever aims	Preservation		Management		Dissemination	
High level aims of ontology use	Effective data mining		Efficient preservation		Robust repository functions	
Internal/External	Internal perspective			External perspective		
Described object	Blog structure		Network of blogs		Blog content	
Process perspective	Blog aggregation		Management		Access & Distribution	
Functions of the archive	Interoperability	Archival functions	Policy-based management	Semantic search & navigation	Merging of social web aspects with ontologies	Ontology-based services

Even if we can associate the use of an ontology to the objectives, it is not sufficient to assess its relevance for the project. The following additional criteria should be taken into consideration and be estimated during evaluation:

- State of the ontology (e.g. initial draft, W3C recommendation, etc.),
- Dispersion of the ontology,
- Availability of applications,
- Vitality of the community around the ontology.

Finally, as we mentioned at the end of Section 1, ontologies must be selected while considering implementation feasibility. That is to say, given the resources of BlogForever, the implementation of the ontology must be open to a suitable description syntax, grammar, representation and population method.

Three scenarios have been identified and chosen to reflect the above considerations, strengthening the probability of its high impact on the success of the BlogForever archiving and preservation system. Each scenario description consists of the particular scenario purpose, a brief overview of how the scenario meets the objectives discussed in this chapter, how it should operate, and an explanation of the relationships to other tasks and deliverables in the project.

3 Interoperability with Linked Open Data

The following chapter describes the scenario how the BlogForever preservation system should apply the concept of Linked Open Data (LOD) to facilitate interoperability between different instances of the BlogForever repository as well as with other digital libraries. Thereby, the interoperability is based on the exposure and linking of preserved data with explicit semantics. The chapter starts with a description of the scenario and its objectives followed by an introduction of the concept of LOD with respect to the scenario objectives. Thereafter, the vocabularies Dublin Core, Friend of a Friend, Semantically-Interlinked Online Communities, and Preservation Metadata Implementation Strategies are introduced because of their importance for the exposure of blogs. These vocabularies are widely accepted and cover the classes and properties that should be exposed to facilitate interoperability. Chapter 3.4 describes the selected classes and properties as well as their relations to the BlogForever data model[21]. Based on the selected vocabulary, examples are given in chapter 3.5 to illustrate how data can be (a) exposed, (b) linked to other repositories, and (c) queried. In consideration of the fact that the data in the BlogForever repository are stored in a SQL database, chapter 3.6 examines tools for an automatic generation of RDF triples from SQL databases. Conclusions are given in the chapter 3.7.

3.1 Scenario description

The successful completion of the BlogForever project may lead to the adoption of the archiving prototype by a diverse range of institutions that will deploy it to preserve a selective collection of blogs. The selection depends on the specific preservation aims of each institution. Thus, interoperability between the collections will become an immediate concern for those who want to avoid isles of isolated data that cannot be easily shared, re-organised, and/or re-used by end-users. The application of ontologies can enhance the interoperability by the provision of open standards for describing, accessing, and connecting data. According to the descriptions in chapter two, Table 2 gives an overview of the aims addressed by this scenario.

Table 2, Objectives Addressed in the Interoperability with Linked Open Data Scenario

BlogForever aims	Preservation		**Management**		**Dissemination**	
High level aims of ontology use	**Effective data mining**		Efficient preservation		Robust repository functions	
Internal/External	Internal perspective			**External perspective**		
Described object	**Blog structure**		Network of blogs		**Blog content**	
Process perspective	Blog aggregation		Management		**Access & Distribution**	
Functions of the archive	**Interoper ability**	Archival functions	Policy-based management	Semantic search & navigation	Merging of social web aspects with ontologies	Ontology -based services

[21] BlogForever Deliverable 2.2: Weblog Data Model, Chapter 9: Blog Data Model, pages 44-56.

The interoperability of the BlogForever environment has to be considered on two levels. First, there should be interoperability among different BlogForever archives. For example, a retrieval process for weblog data could operate on several archives and the results of complex search queries can be merged automatically. Assume that a selected repository contains the blogs of the academic staff of a specified set of universities, and another repository B preserves the blogs of the members of a selected scientific association. It is probable that the data of the two repositories overlap partially. Now, the use of shared vocabularies and a common ontology would allow an application to automatically merge the data from both repositories, providing a user of the repository with the means of searching and exploring the data as if they are from one repository.

Furthermore, interoperability with respect to other external repositories could be supported, for example, with other digital libraries. Digital libraries contain endless amounts of data that can be related to the data preserved in a BlogForever archive. Unlike interoperability between two BlogForever archives, the connection with another digital library will extend the amount of concepts in the resulting ontology. In other words, two BlogForever archives share a common set of concepts (e.g. blog, post, blog author) and a merging means to merge instances of these concepts. However, another digital library has its own concepts like author, book, newspaper, etc. The relations between the concepts of both repositories have to be expressed (e.g. a blog author is a kind of author). Once the relations between the concepts are expressed formally, a merging of instances of both repositories will be possible.

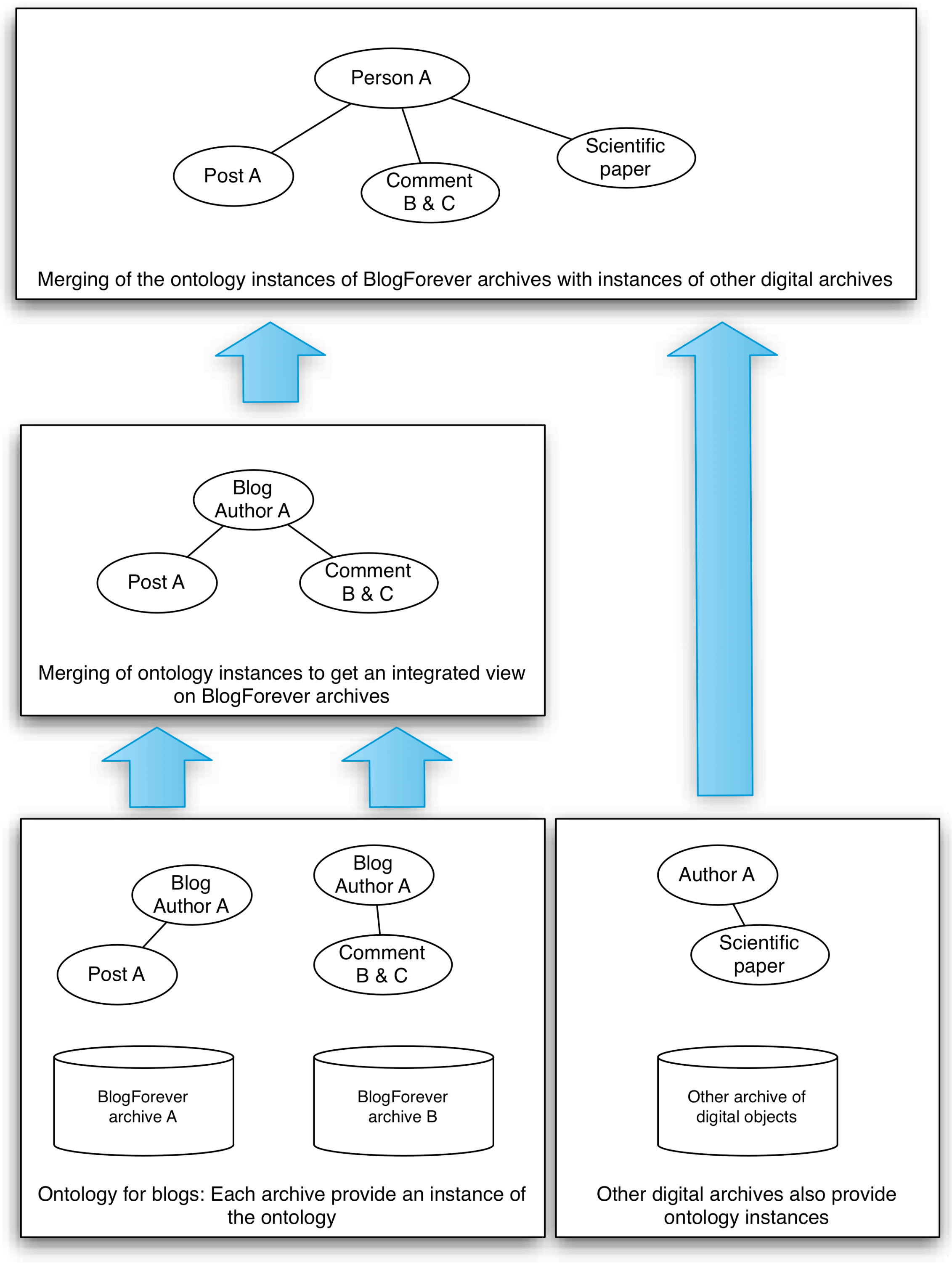

Person A
Post A
Comment B & C
Scientific paper
Merging of the ontology instances of BlogForever archives with instances of other digital archives
Blog Author A
Post A
Comment B & C
Merging of ontology instances to get an integrated view on BlogForever archives
Blog Author A
Post A
Blog Author A
Comment B & C
BlogForever archive A
BlogForever archive B
Ontology for blogs: Each archive provide an instance of the ontology
Author A
Scientific paper
Other archive of digital objects
Other digital archives also provide ontology instances

Figure 2: Interoperability example

Figure 2 gives a visual description of how interoperability could be established in this scenario.

In this scenario, there are some challenges that lead to the following tasks:

1. Relevant concepts of blogs have to be identified. To merge instances of these concepts, it is necessary that an instance can be identified uniquely.
2. Relevant concepts have to be described formally. Thereby, existing vocabularies should be preferred over a new developed vocabulary.
3. Relations between BlogForever concepts have to be identified and described formally.
4. Relations to other concepts (described in other ontology specifications) have to be identified and described formally.

The general scenario depicted in Figure 2 captures an aspect of interoperability within a general search scenario integrating information based on authorship attribution. However, supporting interoperability can go further to include the exposure of information that would facilitate decision-making processes, regarding whether or not to integrate blogs into an organisational repository or personal collection. To this end, there are four immediate concerns that might arise:

1. Legal requirements, mandates, policies with which the organisation needs to comply (e.g. restrictions on use, copy, modification).
2. Trustworthiness, authenticity, reliability, validity of the target information.
3. Determining whether the information is a near-duplicate of material already within the holdings of the organisation or person
4. Maintaining consistent information mining potential (e.g. the format of the material should be amenable to management by resources and processes that are available to the organisation and/or person using the collection).

Therefore, the selection of blog concepts should not only be guided by a content description of the blog intended for discovery. That is, it is recommended that basic pragmatic metadata of the resource (e.g. information about rights, any hash values that have been calculated, format information, and information how it was collected and from what source) be also exposed through the ontology. This touches on possibilities of applications that strengthen support for blog aggregation processes.

It may also be worth mentioning that the way we expose the classes and properties could result in setting the trend for future organisations with respect to blog rights management, evidencing trust, authenticity, reliability and validity, and what information can be mined. For example: if we select a blog property “author” intended to conform to that of traditional publication conventions bound by copyright law, then it may be assumed in the future that blogs should also be bound by the same rules. On the other hand, we can select properties that, while they intersect in meaning with the traditional concepts of authorship, they would be extended to be congruent with a notion of social collaborative authorship that is growing within social network media environments. In this way, we may be able to foster a new approach to rights that attributes creative license by social consensus much the same as the approach to socially controlled validation of information in operation within platforms such as Wikipedia[22].

3.2 Linked Open Data

The term of Linked Open Data (LOD) refers to the approach of exposing the structures of data openly in the Web, and of linking them to other structured data. It is strongly related to the idea of a

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

Semantic Web and the related standards XML, RDF, and OWL. Four design principles describe how the linked data should be provided (Berners-Lee 2009):

- URIs are used as names for *things*,
- HTTP URIs are used and can be accessed,
- A URI provides useful information in a standardized way (e.g. RDF), and
- URIs are linked to other URIs.

As the design principles indicate, the uniform resource identifier (URI) takes a central role in LOD. The URI identifies the things that should be further described and linked. The nature of the things that are identified by URIs is not limited and can be anything from an electronic document (e.g. a web page) to real life objects (e.g. a human being) or abstract concepts (e.g. a topic).

Even if, in principle, a URI could be any identifier, only HTTP URIs should be used for LOD. That is because of the following two reasons: 1) names that are globally unique can be easily created even if only the owner of a domain (or his delegate) creates URIs under their local domain, and, 2) the HTTP URI can be dereferenced to access the describing information of the identified thing (Health & Bizer 2011).

The two ways of *303 URIs* and *Hash URIs* have been established for dereferencing. Both ensure that the result of a request for a real-world object cannot be interpreted as the object itself and has to be interpreted as a description of the object. The first variant utilises the HTTP response code *303 See other* to redirect the HTTP request for a real-world object to the resource that describes this object. Thus, it is announced that the object itself cannot be returned. The second variant utilises a fragment identifier in a URI indicated by the hash symbol (#). The fragment part is stripped off for the HTTP request and, therefore, the object described by the URI cannot be retrieved directly. Instead a document is returned that contains the description for the object identified by the URI as well as other object descriptions. The client has to process the document to get the necessary objects (Health & Bizer 2011).

303 URIs have the advantage that only the information for the requested object will be returned but the disadvantage that two HTTP requests are needed to get the information. Hash URIs need only one HTTP request but the whole document will be returned. Hence, 303 URIs are often used for large data sets while Hash URIs are often used in RDF vocabularies (Health & Bizer 2011). This leads to the following recommendations:

1. A BlogForever archive should use 303 URIs for identifying objects stored or described in the archive.
2. The BlogForever project should use Hash URIs for identifying own concepts in a BlogForever specific vocabulary.

The format for describing the objects in LOD is the resource description framework (RDF). RDF allows making statements about things with the triple of subject, predicate, and objecting. The subject identifies the thing that should be described. The predicate defines a property of the thing and the object represents the value of this property. Each part of the triple can be a URI. The RDF triple links the described thing (the subject) to another thing if the object in the triple is a URI. However, the object can also be a literal, e.g. a text string or a date (Manola & Miller 2004).

The RDF statements can be serialised in different formats. RDF/XML and RDFa are standardised by the W3C but other formats (Turtle, N-Triples, RDF/JSON) are also used for specific needs (Health & Bizer 2011). However, the BlogForever project should concentrate on the W3C formats. The RDFa format is described in another chapter below. It is not suitable for the here mentioned case because it is intended to include RDF statements into HTML webpages but the data should be exposed as "pure" RDF for processing by other applications. Therefore, the RDF/XML format is recommended for the BlogForever project to expose archived data as serialised RDF.

While RDF provides the structure (subject, predicate, object) to describe things, it does not provide any domain specific terms and concepts. Therefore, vocabularies are needed to define the classes and related properties in a domain. Thereby, each thing (or object) can be an instance of one or more classes. Several properties can be defined for a class and, thus, the meaning of the properties is also defined for an instance object of this class. It is recommended to use already existing and widely deployed vocabularies because this facilitates linking with other data sets. New concepts should only be defined if they are not contained in the existing vocabularies (Health & Bizer 2011).

Vocabularies are defined with RDFS (Brickley & Guha 2004) or OWL (W3C OWL Working Group 2009). The concepts in a vocabulary are identified with URIs. The URIs are used in RDF statements which utilize this vocabulary to express that an object belongs to a class or has specific properties. The URIs should be HTTP URIs and the domain of the URIs should be under control of the community that is maintaining the vocabulary. Therefore, it is recommended for the BlogForever project:

1. To use existing vocabularies if possible,
2. To express own concepts in RDFS or OWL if an own vocabulary is necessary, and
3. To use the domain blogforever.eu for creating URIs of the own vocabulary.

RDFS and OWL provide already some important properties that should be used in BlogForever and, therefore, are explained here. The **rdf:type** property indicates that a thing is instance of a specific class (Brickley & Guha 2004). For example, in BlogForever the type property can be used to indicate that the archived thing is a blog, a blog post, a comment, etc. The **rdfs:seeAlso** property "is used to indicate a resource that might provide additional information" (Brickley & Guha 2004). This property can be used very flexible to add any information on the described thing even if the relation between the thing and the information is not really clear or a specific property for describing the relation does not exist yet. The **owl:sameAs** property allows to declare that two instances are identical (M. K. Smith et al. 2004). For example, it can be indicated that a blog or a blog post archived in two different BlogForever repositories is the same. Thus, overlapping data sets from different repositories can be linked and a view on these data sets can be merged.

The mentioned properties indicate already some kinds of links between the data set. More speicifically, three types of RDF links can be utilized for LOD (Health & Bizer 2011):

- Identity links: they are created by the above mentioned owl:sameAs property and indicate that two objects linked using this property are identical.
- Vocabulary links: they link the exposed data to existing vocabularies and, thereby, indicate that the described thing is instance of a defined class or has a defined property.
- Relationship Links: they connect the exposed data with other data sources by pointing from the described thing to other things. For example, places or locations can be described in an external data source. Now, the value of a property that specifies the location of a described thing could be a URI from the external data source. Thus, the internal data about the thing are linked to the data in the external data source.

A lot of datasets have been already exposed within LOD[23]. Relationship links can be used to link BlogForever archives to some of these datasets. The following datasets show much promise to be exposed as data linked to blogs:

To be linked to music discussed in blogs or people who are musicians who write blogs:

- BBC Music http://www.bbc.co.uk/music, http://www.bbc.co.uk/music/faqs
- MusicBrainz http://musicbrainz.org/

[23] See http://richard.cyganiak.de/2007/10/lod/ for a recent overview of available datasets.

- Last FM Wrapper http://dbtune.org/last-fm/

To be linked to researchers who write blogs:

- DBLP http://www4.wiwiss.fu-berlin.de/dblp/, http://dblp.l3s.de/d2r/

To be linked to development projects discussed in blogs or mime types that are used within the blogs:

- DOAP space 43,000 DOAP profiles of Freshmeat projects, 15,000 SourceForge projects, 1,720 Python Package http://doapspace.org/
- Related to doapspace.org we also have http://www.ohloh.net/ which tracks activity within each development project. There also is a RDF-iser of ohloh.net at http://rdfohloh.wikier.org

To be linked to images within blogs and their location:

- flickr wrapper http://www4.wiwiss.fu-berlin.de/flickrwrappr/. Example: http://www4.wiwiss.fu-berlin.de/flickrwrappr/photos/Paris

To be linked to any place names associated to be mentioned in blogs:

- Geonames http://www.geonames.org/ontology/documentation.html. Example: http://www.geonames.org/2950159/berlin.html, http://sws.geonames.org/2950159/about.rdf

To be linked to specific topic mentioned within blog:

- Project Gutenberg Catalogue http://www4.wiwiss.fu-berlin.de/gutendata/
- Protein Data Bank http://semanticscience.org/projects/pdb2rdf/

In particular, linking blogs to DBLP and other records of formal publication could be a means of associating levels of authority to blogs that moves beyond general popularity counts (available through services such as Technorati) that could enhance blog queries that support academic research.

In the next section, we introduce core vocabularies, which are most common among the datasets mentioned above and are already exposed as Linked Open Data. We suggest their use as a foundation for the BlogForever ontology framework and increasing the potential of utilising available datasets.

3.3 Important Vocabularies

Several existing vocabularies have been considered for exposing data from a BlogForever repository as LOD. Dublin core (DC), friend of a friend (FOAF), and semantically-interlinked online communities (SIOC) have emerged as the most important and, therefore, are described below. Additionally, PREMIS is introduced because the vocabulary defines specific concepts for preserved objects that are used in chapter 3.4.

The Dublin core metadata element set[24] (DC) is maintained by the Dublin Core Metadata Initiative (http://dublincore.org/). The name comes from two aspects of the context of its creation: the term "Dublin" is indicative of the location of the initial proposal (Dublin, Ohio) which took place at an invitational workshop in 1995, while the term "core" resulted from the original workshop purpose of identifying descriptive terms to serve as a core set of properties, broad and generic and usable for a wide range of resources. Since then the set of terms have been extended and have evolved to

[24] http://dublincore.org/documents/dces/

include more detail[25] but still retains the flavour of its original purpose. The metadata schema is adopted either in part or in its entirety by a large number of European and international projects (e.g. several national libraries including the National Library of Netherlands, resources such as MusicBrainz[26], and record creating tools such as OCLC's Connexion[27]). The schema is also widely employed as properties to expose resources using the principles of Linked Open Data (e.g. Digital Bibliography and Library Project[28], MusicBrainz, Project Gutenberg Catalogue[29] and Geonames[30], Linked Movie Database[31] and Data-gov Wiki[32]). The core schema consists of fifteen properties:

- Title – the name of the resource
- Creator – author
- Subject – subject as keywords, classification codes
- Description – can of worms
- Publisher – entity responsible for making available
- Contributor – responsible entity
- Date – usually creation or publication date
- Type – nature or genre of the content
- Format – physical or digital manifestation
- Identifier – unambiguous URI
- Source – Reference to contributing material
- Language
- Relation – a reference to a related resource
- Coverage – jurisdiction, temporal validity
- Rights – Information abut rights over the resource

Extensions to the above core set tend to comprise refinements and/or elaborations of existing concepts (e.g. expanding "rights" to include different types of rights, say, for instance, "access rights") rather than additions of new concepts.

The **FOAF** project aims on machine-readable descriptions of people, the links between them and things they create and do[33]. Therefore, several classes and properties are defined. The FOAF vocabulary can be divided in a Core category and a Social Web category. The former contains

[25] http://dublincore.org/documents/dcmi-terms/

[26] http://musicbrainz.org/

[27] http://www.oclc.org/connexion/

[28] http://www4.wiwiss.fu-berlin.de/dblp/

[29] http://www4.wiwiss.fu-berlin.de/gutendata/

[30] http://www.geonames.org/ontology/documentation.html

[31] http://www.jobs.ac.uk/job/AEK779/lectureship-in-computer-science/

[32] http://data-gov.tw.rpi.edu/wiki

[33] http://www.foaf-project.org/about

concepts that can be used for universal descriptions of people, groups of people and information. The latter contains concepts that are specific for the Web[34].

The FOAF Specification consists of classes and properties with the statuses unstable, testing, and stable. Stable terms will not be changed. Stable classes of the Core category are Agent, Person, Organization, and Group. Additionally, the classes Project, Document, and Image are included with the status testing. Agent is the central class and superclass of Organization, Group, and Person. The Agent class has properties like weblog, account or openid that can belong to every subclass. The Person class represents people regardless of whether they are alive, dead, real or imaginary. Therefore, the class has specific properties like firstname, lastname, family_name, etc. Persons can link to other Persons by using the property "knows". The Group class represents a collection of individual agents[35]. The most recent version of the FOAF vocabulary specification is 0.98. Figure 3 illustrates the core classes and properties of FOAF vocabulary.

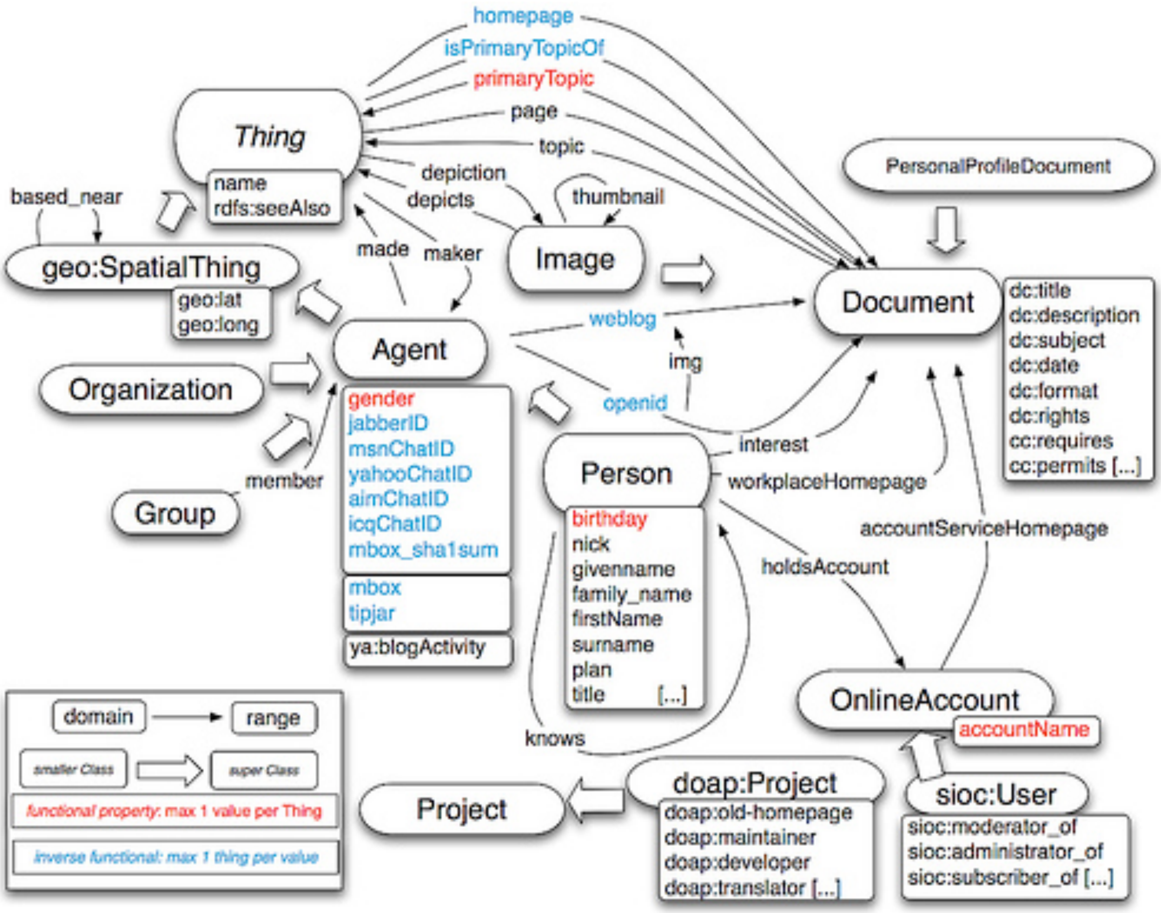


Figure 3: Part of FOAF core ontology[36]

The **SIOC** project is an attempt to create an ontology that fully describes the structure and content of online community sites. Thereby, it should be possible to create and browse new connections

[34] FOAF Vocabulary Specification 0.98: http://xmlns.com/foaf/spec/

[35] FOAF Vocabulary Specification 0.98: http://xmlns.com/foaf/spec/

[36] http://danbri.org/words/2007/11/04/222

between discussion channels and posts. Examples for online community sites in the sense of the SIOC project are blogs, bulletin boards, mailing lists and newsgroups[37].

The SIOC ontology is composed of the SIOC Core ontology and the three modules Access, Services and Types. The SIOC Core ontology defines the main classes, which are represented in Figure 4. The modules extend the Core ontology. The Access module provides the classes Permission and Status to describe access rights (e.g. users' permissions). The class Service in the Services module enables the indication that a web service is associated to the community site. The Types module provides several subclasses of the Core ontology, e.g. Weblog as a subclass of Forum and BlogPost as a subclass of Post (Bojars & J. G. Breslin 2010).

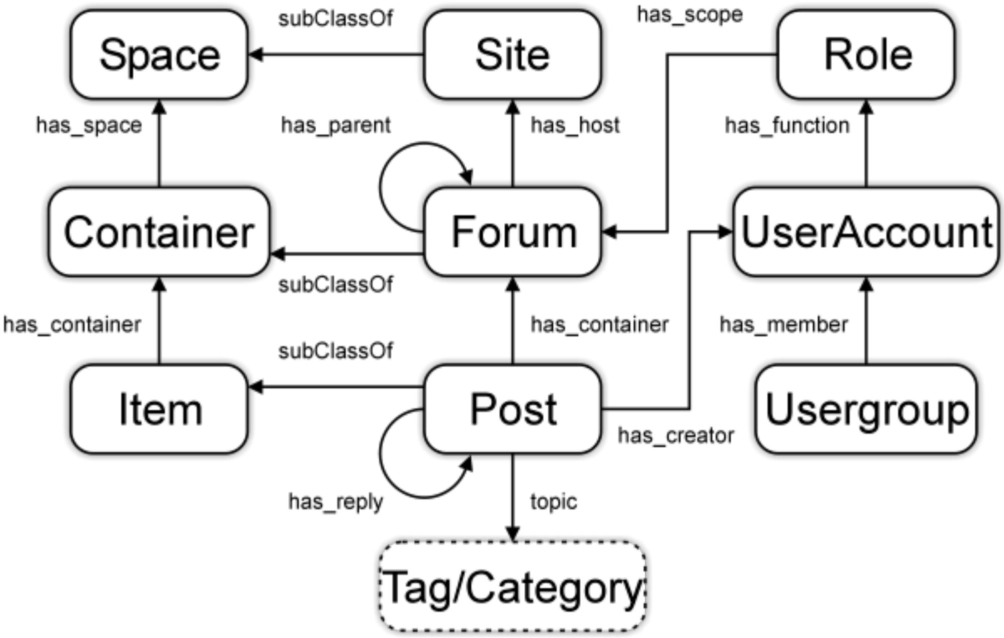


Figure 4: SIOC Core Ontology (Bojars & J. G. Breslin 2010)

The Revision 1.26 of the SIOC Core ontology is published by the W3C as a W3C member submission[38]. The latest version of the SIOC Core ontology is Revision 1.35[39].

The **PREMIS** (Preservation Metadata Implementation Strategies) ontology originates from the PREMIS data dictionary for preservation metadata (version 2.1 can be found at http://www.loc.gov/standards/premis/). This metadata standard was initially developed by the PREMIS working group organised by the Research Libraries Group (RLG) and the Online Computer Library Center (OCLC)[40]. The objective of the working group was to identify metadata that specifically "supports digital preservation processes" and that "helps to ensure that digital materials remain usable over the long term"[41]. It is now maintained by the Library of Congress[42]. Its manifestation as an ontology has been proposed by many including the PREMIS Ontology Working

[37] http://sioc-project.org/faq

[38] http://www.w3.org/Submission/sioc-spec/

[39] http://sioc-project.org/ontology

[40] RLG has now merged with OCLC. For more information on OCLC, see http://www.oclc.org

[41] See description at http://www.oclc.org/research/activities/past/rlg/premis.htm

[42] http://www.loc.gov/

Group at the University of Gent in Belgium[43]. The ontology, in accordance with the PREMIS data dictionary, revolves around classes that reflect the dimensions of an intellectual entity described in terms of Object, Agent, Rights, and Event (see Figure 5) and associated properties[44].

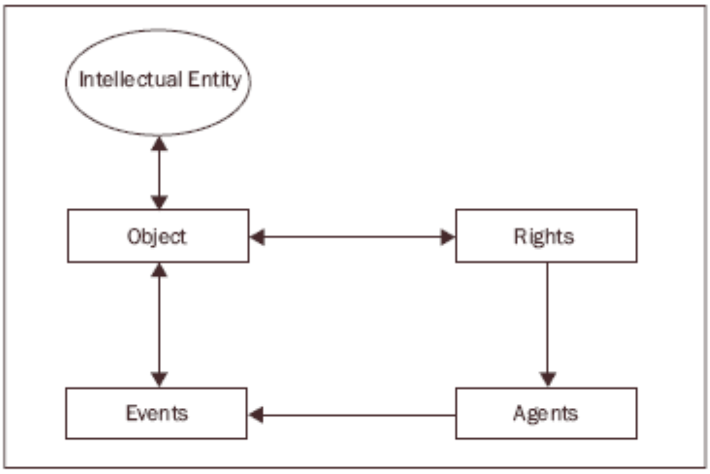


Figure 5: PREMIS classes and direction of relationships[45]

3.4 Classes and properties to expose BlogForever data

The selection of classes and properties proposed here has been guided by the observations offered in the interoperability scenario description. The outcome has been summarised here by the following objectives:

1. To allow an application to automatically merge the overlapping data from several repositories (blog archive and other libraries), say, produced by the same group of persons and/or organisations.
2. To foster a new notions of "authorship" and "rights" that are congruent with the social network media environment. For example to set a trend for:
    - Socially acknowledged collective rights over an artefact without explicit contracts and/or agreement between parties involved. Providing properties that include not just authorship but any type of contributor can facilitate this.
    - Social attribution of credit that does not rely on official records held by formal organisations. By providing ways to extract networks of blogs and associated actors concurrently, a better understanding of socially supported credit can be established.
3. To enable information search that supports decision making processes by exposing evidence of authority, authenticity and reliability.
    - Information sourced from formal publications (e.g. Digital Bibliography and Library Project records[46]) can be related to socially generated blogs as an indicator

[43] See the announcement at the Library of congress http://www.loc.gov/standards/premis/owlOntology-announcement.html along with the description at http://multimedialab.elis.ugent.be/users/samcoppe/ontologies/Premis/index.html

[44] http://multimedialab.elis.ugent.be/ontologies/PREMIS2.0/v1.0/

[45] The PREMIS data model: http://www.paradigm.ac.uk/workbook/metadata/preservation-model.html

[46] http://www4.wiwiss.fu-berlin.de/dblp/

of authority, authenticity, reliability and trustworthiness regarding the content. The issue of trust with respect to social network media is currently an active research area (Agichtein et al. 2008).

- Source information (i.e. the URI where the blog was originally published) and full crawler information (i.e. the algorithm and specification of how the blog content was collected) also provides support for inferring the quality of information.

4. To leave an audit trail of "information signature" exposed for use in inferring completeness, integrity and verisimilitude. For example,

   - Checksum values and crawler information as indications of how the target information relates to previous versions and other material already held within the repository or personal collection.

5. To open up avenues for deeper levels of information mining and synthesis.

   - Author's publications across genres (formal publishing versus blogs)- being able to examine relationship between formal and informal publication and how they influence each other can be invaluable in verifying research contributions.

   - Agents that occur frequently together as a group of contributors across different information sources can be recognised as a social network. Relationships within these networks can also be queried. For example, are persons more likely to be a blog post contributor or a comment contributor? Are two people in a reciprocal relationship (e.g. both have equally played the role of blog post contributor and commenter in relation to each other) or an asymmetrical relationship?

   - A window into examining information climate change over time – for example, event driven publication landscape can be tracked including investigations of possible correlations. Do discoveries in one form of publication influence the other?

Item 1 above suggests that the classes selected for the ontology be as widely applicable across as many blogs as possible, and that the associated properties have sufficient overlap with the properties of material held within other repositories that do not necessarily specialise in weblogs. The former criterion imposes, initially, that we focus on a core data model shared across weblogs as a basis for our ontology classes (Table 3). The latter criterion imposes a weight on properties that are shared by several different types of resources (such as those proposed by Dublin Core – see section 3.3). The items 2, 3, 4, 5 suggest an emphasis on being able to identify relations between authors and resources, not only in terms of authorship and topicality but in terms of rights and events that bind them together (in agreement with the PREMIS data model – see section 3.3). The classes and properties, therefore, are chosen not to merely replicate the entire data model that has been reported in the deliverable D2.2, Weblog Data Model[47] but to reflect digital rights management policies and support preservation objectives that are in development in WP3 BlogForever Policies.

In light of the above, Table 3 shows the classes and Table 4 the properties that have been selected as promising candidates for building the ontology to expose archived data from a BlogForever repository as LOD. Each thing that will be described in the exposed data has to belong to at least one of the following classes.

[47] BlogForever Deliverable 2.2: Weblog Data Model, Chapter 9: Blog Data Model, pages 44-56.

Table 3: Classes for the exposure of preserved BlogForever data

Class	Description	Vocabulary & URI of the Ontology class	Related entities in the data model (see D2.2)
Weblog	The entire weblog. It must not be confused with individual posts or other parts of the blog.	SIOC http://rdfs.org/sioc/types#Weblog	Blog
BlogPost	A blog post. It must not be confused with the blog or comments in the blog.	SIOC http://rdfs.org/sioc/types#BlogPost	Entry, Post
Comment	A comment on a blog post.	SIOC http://rdfs.org/sioc/types#Comment	Comment
UserAccount	A user account used in the blog. A user account can be used e.g. to create a post or a comment in the blog. A user account is often owned by a Person but other owners (e.g. an organisation) are possible as well.	SIOC http://rdfs.org/sioc/ns#UserAccount	Author, User_Profile
Agent	An agent is a general class for things that do something (e.g. create a blog post). Well known subclasses are Person and Organisation.	FOAF http://xmlns.com/foaf/0.1/Agent	
Person	A person. Person is a subclass of Agent.	FOAF http://xmlns.com/foaf/0.1/Person	Author, User_Profile
Organisation	An organisation, e.g. a company, university, library, etc. Organisation is a subclass of Agent.	FOAF http://xmlns.com/foaf/0.1/Organization	
Tag	A tag that is added to a blog post or comment.	SIOC http://rdfs.org/sioc/types#Tag	Tag

Table 4 shows the properties that should be used to describe the instances and to link the exposed data to data in other repositories.

Table 4: Properties for the exposure of preserved BlogForever data

Property	Domain of the property	Description	Vocabulary & URI of the Ontology class	Related attributes in the data model (see D2.2)
		Descriptive properties		
title	Weblog, BlogPost	The title and subtitles of a blog or a blog post.	Dublin Core http://purl.org/dc/terms/title	title in Blog, title in Entry
abstract	Weblog, BlogPost, Comment	A summary of the resource.	Dublin Core http://purl.org/dc/terms/abstract	
description	Weblog, BlogPost, Comment	Description may include but is not limited to: an abstract, a table of contents, a graphical representation, or a free-text account of the resource.	Dublin Core http://purl.org/dc/terms/description	
source	Weblog, BlogPost, Comment	The original URI where the resource has been crawled.	Dublin Core http://purl.org/dc/terms/source	URI in Blog, URI in Entry, URI in Comment
creation date	Weblog, BlogPost, Comment	The date when the resource has been created.	Dublin Core http://purl.org/dc/terms/created	date_created in Entry, date_added in Comment
modification date	Weblog, BlogPost, Comment	The date when the resource has been modified.	Dublin Core http://purl.org/dc/terms/modified	date_modified in Entry, date_modified in Comment
language	Weblog, BlogPost, Comment	The language of the resource.	Dublin Core http://purl.org/dc/terms/language	language in Blog
coverage	Weblog, BlogPost,	The spatial or temporal topic or a jurisdiction of	Dublin Core http://purl.org/dc/terms	

	Comment, Agent	the resource. The refined properties "spatial" and "temporal" should be used if possible.	/coverage	
spatial	Weblog, BlogPost, Comment, Agent	A spatial characteristic of the resource, e.g. a place or region.	Dublin Core http://purl.org/dc/terms/spatial	location_city and location_country in Blog, geo_longitude and geo_latitude in Entry and Comment
temporal	Weblog, BlogPost, Comment, Agent	A temporal characteristic of the resource, e.g. a period or an era.	Dublin Core http://purl.org/dc/terms/temporal	
creator	BlogPost, Comment	The author of the resource.	Dublin Core http://purl.org/dc/terms/creator	Author <> Entry, Author <> Comment
creator_of	UserAccount	A resource that the UserAccoung is creator/author of.	SIOC http://rdfs.org/sioc/spec/#creator_of	Author <> Entry, Author <> Comment
has_creator	Weblog, BlogPost, Comment	The UserAccount that created the resource.	SIOC http://rdfs.org/sioc/spec/#has_creator	Author <> Entry, Author <> Comment
contributor	Weblog, BlogPost, Comment	An entity responsible for making contributions to the resource.	Dublin Core http://purl.org/dc/terms/contributor	
type / genre	Weblog	Nature or genre of the blog.	Dublin Core http://purl.org/dc/terms/type	
container_of	Weblog	Blog posts or comments that the weblog contains. The inverse of has_container.	SIOC http://rdfs.org/sioc/ns#container_of	Blog <> Post, Blog <> Comment
has_contain	BlogPost,	The blog to which the blog post or comment	SIOC	Blog <> Post, Blog <>

er	Comment	belong.	http://rdfs.org/sioc/ns#has_container	Comment
reply_of	Comment	The blog or comment to which the comment answers.	SIOC http://rdfs.org/sioc/spec/#reply_of	Post <> Comment
has_reply	BlogPost, Comment	The comment that is a reply to the resource.	SIOC http://rdfs.org/sioc/spec/#has_reply	Post <> Comment, has_reply in Entry and Comment
links_to	BlogPost, Comment	Resources that are linked in the blog post or comment.	SIOC http://rdfs.org/sioc/spec/#links_to	URI in Link
name	Agent, Person, Organisation	The name of the entity. The properties givenName and familyName can also be used if suitable.	FOAF http://xmlns.com/foaf/0.1/name	name_displayed in Author, name in User_Profile
givenName	Person	The given name of a person.	FOAF http://xmlns.com/foaf/0.1/givenName	
familyName	Person	The family name of a person.	FOAF http://xmlns.com/foaf/0.1/familyName	
subject	Weblog, BlogPost, Comment	The topic of the resource. For (user generated) tags and categories, the refined property topic should be used.	Dublin Core http://purl.org/dc/terms/subject	Category <> Blog
topic / tag	Weblog, BlogPost, Comment	A tag or a category that describes the resource.	SIOC http://rdfs.org/sioc/spec/#topic	Category <> Blog, Content <> Tag
Administrative properties				
accessRight	Weblog, BlogPost, Comment	Information about who can access the resource or an indication of its	Dublin Core http://purl.org/dc/terms/accessRights	access_rights in Blog and Content

		security status.	
license	Weblog, BlogPost, Comment	A legal document giving official permission to do something with the resource.	Dublin Core http://purl.org/dc/terms/license
dateCopyrighted	Weblog, BlogPost, Comment	The date of copyright.	Dublin Core http://purl.org/dc/terms/dateCopyrighted
identifier / DOI	Weblog, BlogPost, Comment	The digital object identifier for the resource.	Dublin Core http://purl.org/dc/terms/identifier
format	Weblog, BlogPost, Comment	The file format, physical medium, or dimensions of the resource.	Dublin Core http://purl.org/dc/terms/format
checksum_type	BlogPost, Comment	The algorithm used for check sum.	PREMIS http://multimedialab.elis.ugent.be/users/samcoppe/ontologies/Premis/premis.owl#messageDigestAlgorithm
checksum_value	BlogPost, Comment	Hash sum value calculated using checksum algorithm.	PREMIS http://multimedialab.elis.ugent.be/users/samcoppe/ontologies/Premis/premis.owl#messageDigest
accrual_method	Weblog, BlogPost, Comment	The method by which items are added to a collection, e.g the web spider name description.	Dublin Core http://purl.org/dc/terms/accrualMethod

3.5 Examples for the application of LOD

The following examples illustrate how archived BlogForever data are exposed as RDF triples.

The example in Figure 6 shows the example of a blog post in the BlogForever blog[48]. The ovals contain URIs and represent things that can be described and linked. The arrows and their label represent properties that are used to describe a thing. These properties are the predicates of the RDF

[48] http://blogforever.eu/blog/category/blog/

triples. The sender of an arrow is the subject of the RDF triple. The receiver of the arrow is the object. The rectangles represent literals and can only be receiver of an arrow or rather objects in a RDF triple.

The domain of the exemplary BlogForever archive is www.foo.foo and the location of the ontology that describes the archived data is http://www.foo.foo/ontologies/2012/3/OntologyTest.owl. Four entities are described in this example:

- The blog as whole: http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogForever-Blog,
- A single blog post: http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogForever_and_migration,
- A user account of the person who wrote the blog post: http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#edpinsent_user_account_on_blogforever,
- A person that the user accounts belongs to: http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Ed_Pinsent

Each entity is linked via the property rdf:type to a specific class from SIOC or FOAF. The dcterms:source property is used to indicate the origin URL where the blog or blog post was crawled. Other properties describe titles, names, languages, etc. with literals. A specific language of the literal can be indicated (e.g. @en for English) and, thus, the same property can be described in many languages.

The properties that represent the relationships between blog and blog post, blog post and user account, and user account and a specific person are reciprocal because each of these properties has an inverse property. For example, if the blog is the container of a blog post then this blog post has the blog as a container.

Figure 7 shows an example where the blog post has been commented and the comment is also archived in the repository. The comment is modelled as a reply to the post. Additional comments can also reply to either the blog post or the comment. Furthermore, every comment is also part of the blog. Therefore, the blog is the container of blog posts and comments. Furthermore, it is assumed in the example that the author of the comment could also be identified with a user account in the blog environment as well as the person behind this account. Hence, the authors of both blog post and comment are described explicitly.

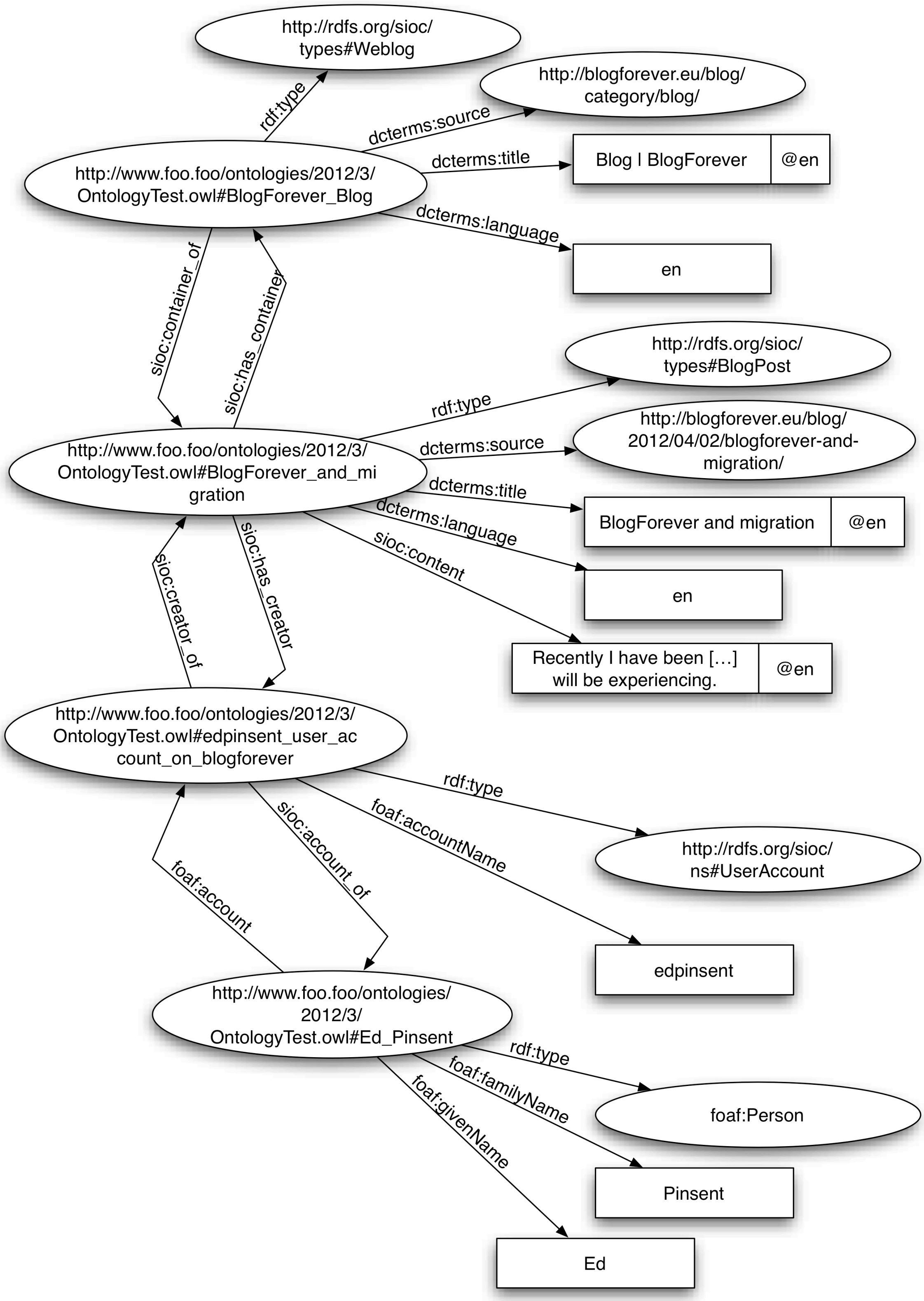

http://rdfs.org/sioc/types#Weblog
rdf:type
http://blogforever.eu/blog/category/blog/
dcterms:source
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogForever_Blog
dcterms:title
Blog | BlogForever
@en
dcterms:language
en
sioc:container_of
sioc:has_container
http://rdfs.org/sioc/types#BlogPost
rdf:type
http://blogforever.eu/blog/2012/04/02/blogforever-and-migration/
dcterms:source
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogForever_and_migration
dcterms:title
BlogForever and migration
@en
dcterms:language
en
sioc:content
Recently I have been […] will be experiencing.
@en
sioc:creator_of
sioc:has_creator
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#edpinsent_user_account_on_blogforever
rdf:type
foaf:accountName
http://rdfs.org/sioc/ns#UserAccount
edpinsent
sioc:account_of
foaf:account
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Ed_Pinsent
rdf:type
foaf:familyName
foaf:Person
foaf:givenName
Pinsent
Ed

Figure 6: A blog post in the BlogForever blog as RDF triples

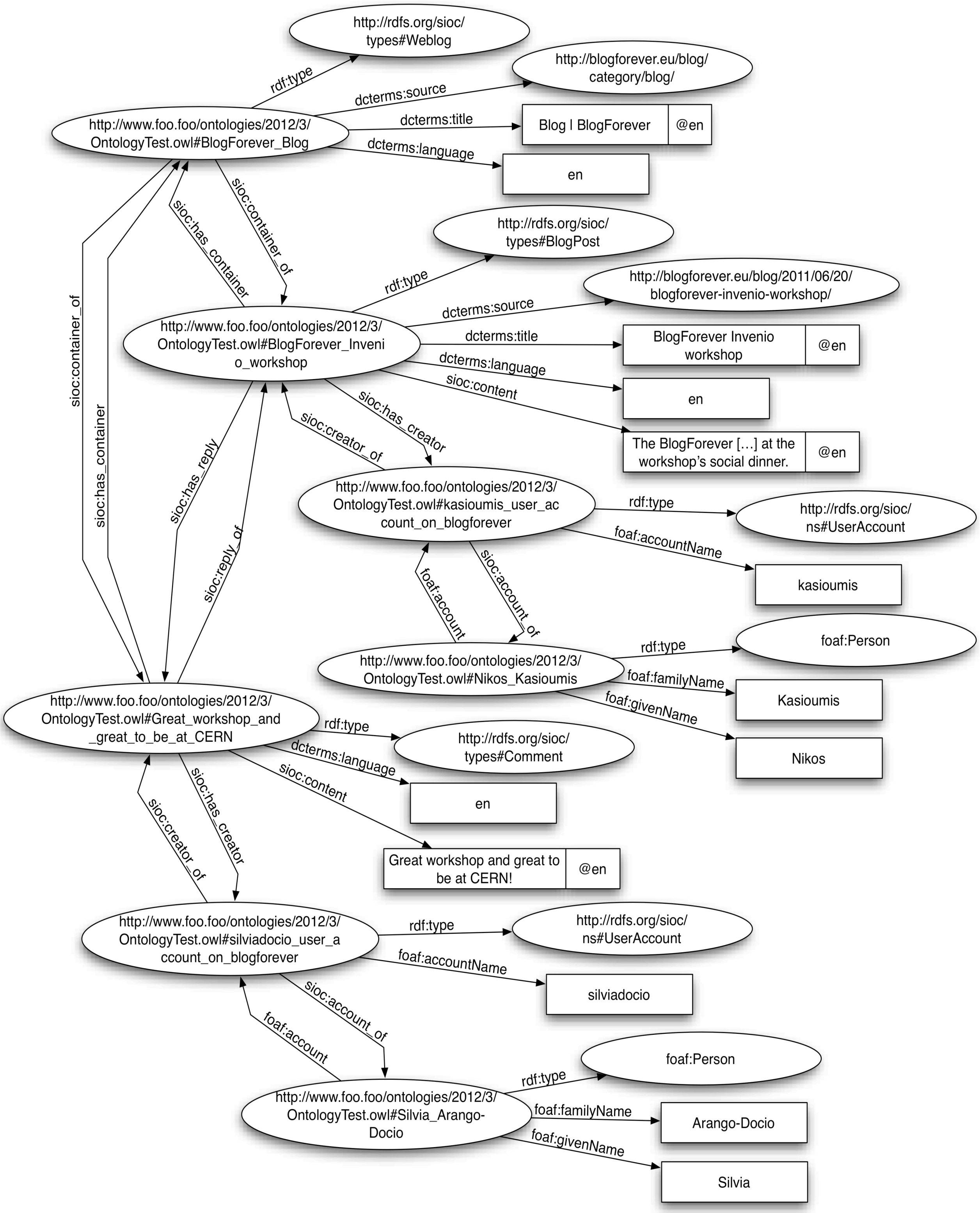

http://rdfs.org/sioc/ types#Weblog
http://blogforever.eu/blog/ category/blog/
rdf:type
dcterms:source
dcterms:title
Blog | BlogForever
@en
http://www.foo.foo/ontologies/2012/3/ OntologyTest.owl#BlogForever_Blog
dcterms:language
en
sioc:has_container
sioc:container_of
http://rdfs.org/sioc/ types#BlogPost
http://blogforever.eu/blog/2011/06/20/ blogforever-invenio-workshop/
rdf:type
dcterms:source
http://www.foo.foo/ontologies/2012/3/ OntologyTest.owl#BlogForever_Invenio_workshop
dcterms:title
BlogForever Invenio workshop
@en
dcterms:language
sioc:content
en
The BlogForever […] at the workshop's social dinner.
@en
sioc:container_of
sioc:has_container
sioc:has_reply
sioc:reply_of
sioc:creator_of
sioc:has_creator
http://www.foo.foo/ontologies/2012/3/ OntologyTest.owl#kasioumis_user_account_on_blogforever
rdf:type
http://rdfs.org/sioc/ ns#UserAccount
foaf:accountName
kasioumis
foaf:account
sioc:account_of
foaf:Person
rdf:type
http://www.foo.foo/ontologies/2012/3/ OntologyTest.owl#Nikos_Kasioumis
foaf:familyName
Kasioumis
foaf:givenName
Nikos
http://www.foo.foo/ontologies/2012/3/ OntologyTest.owl#Great_workshop_and_great_to_be_at_CERN
rdf:type
http://rdfs.org/sioc/ types#Comment
dcterms:language
en
sioc:content
Great workshop and great to be at CERN!
@en
sioc:creator_of
sioc:has_creator
http://www.foo.foo/ontologies/2012/3/ OntologyTest.owl#silviadocio_user_account_on_blogforever
rdf:type
http://rdfs.org/sioc/ ns#UserAccount
foaf:accountName
silviadocio
sioc:account_of
foaf:account
foaf:Person
rdf:type
http://www.foo.foo/ontologies/2012/3/ OntologyTest.owl#Silvia_Arango-Docio
foaf:familyName
Arango-Docio
foaf:givenName
Silvia

Figure 7: A blog post with a comment as RDF triples

The representations in Figure 6 and Figure 7 should just facilitate the understanding for the reader. They show the graph structure of the RDF triples. However, these descriptions have to be provided in a machine-readable format to enable automatic processing and reasoning. As mentioned before, these RDF triples could be serialized in several formats. The preferred format for BlogForever is RDF/XML. Therefore, the following code example demonstrates how the RDF triples from the figures would be represented in RDF/XML.

```
<?xml version="1.0"?>

<rdf:RDF xmlns="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#"
     xml:base="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl"
     xmlns:sioc-types="http://rdfs.org/sioc/types#"
     xmlns:dc="http://purl.org/dc/elements/1.1/"
     xmlns:foaf="http://xmlns.com/foaf/0.1/"
     xmlns:terms="http://purl.org/dc/terms/"
     xmlns:dcam="http://purl.org/dc/dcam/"
     xmlns:authors="http://dblp.l3s.de/d2r/resource/authors/"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:sioc-ns="http://rdfs.org/sioc/ns#"
     xmlns:aboutdcmi="http://purl.org/dc/aboutdcmi#"
     xmlns:skos="http://www.w3.org/2004/02/skos/core#">

<!-- Description of the blog
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogForeve
r_Blog -->
<sioc-types:Weblog
rdf:about="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogFore
ver_Blog">
        <dc:source>http://blogforever.eu/blog/category/blog/</dc:source>
        <dc:title xml:lang="en">Blog | BlogForever</dc:title>
        <dc:language>en</dc:language>
        <sioc-ns:container_of
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogF
orever_Invenio_workshop"/>
        <sioc-ns:container_of
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogF
orever_and_migration"/>
        <sioc-ns:container_of
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Great
_workshop_and_great_to_be_at_CERN"/>
        <sioc-ns:container_of
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#What_
relationships_among_blogs_do_you_know"/>
</sioc-types:Weblog>

<!-- Description of the blog post
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogForeve
r_Invenio_workshop -->
<sioc-types:BlogPost
rdf:about="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogFore
ver_Invenio_workshop">
```

```
        <dc:source>http://blogforever.eu/blog/2011/06/20/blogforever-
invenio-workshop/</dc:source>
        <dc:title xml:lang="en">BlogForever Invenio workshop</dc:title>
        <sioc-ns:content xml:lang="en">The BlogForever Invenio workshop
... at the workshop’s social dinner.</sioc-ns:content>
        <sioc-ns:has_container
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogF
orever_Blog"/>
        <sioc-ns:has_reply
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Great
_workshop_and_great_to_be_at_CERN"/>
        <sioc-ns:has_creator
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#kasio
umis_user_account_on_blogforever"/>
    </sioc-types:BlogPost>

<!-- Description of the blog post
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogForeve
r_and_migration -->
<sioc-types:BlogPost
rdf:about="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogFore
ver_and_migration">
        <dc:title xml:lang="en">BlogForever and migration</dc:title>
        <dc:source>http://blogforever.eu/blog/2012/04/02/blogforever-and-
migration/</dc:source>
        <sioc-ns:content xml:lang="en">Recently I have been ... will be
experiencing.</sioc-ns:content>
        <sioc-ns:has_container
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogF
orever_Blog"/>
        <sioc-ns:has_creator
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#edpin
sent_user_account_on_blogforever"/>
    </sioc-types:BlogPost>

<!-- Description of the person
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Ed_Pinsent
-->
<foaf:Person
rdf:about="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Ed_Pinse
nt">
        <foaf:givenName>Ed</foaf:givenName>
        <foaf:familyName>Pinsent</foaf:familyName>
        <foaf:holdsAccount
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#edpin
sent_user_account_on_blogforever"/>
</foaf:Person>

<!-- Description of the comment
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Great_work
shop_and_great_to_be_at_CERN -->
<sioc-ns:Comment
rdf:about="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Great_wo
rkshop_and_great_to_be_at_CERN">
        <sioc-ns:content xml:lang="en">Great workshop and great to be at
```

```
CERN!</sioc-ns:content>
        <sioc-ns:has_container
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogF
orever_Blog"/>
        <sioc-ns:reply_of
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogF
orever_Invenio_workshop"/>
        <sioc-ns:has_creator
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#silvi
adocio_user_account_on_blogforever"/>
</sioc-ns:Comment>

<!-- Description of the person
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Nikos_Kasi
oumis -->
<foaf:Person
rdf:about="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Nikos_Ka
sioumis">
        <foaf:familyName>Kasioumis</foaf:familyName>
        <foaf:givenName>Nikos</foaf:givenName>
        <foaf:account
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#kasio
umis_user_account_on_blogforever"/>
</foaf:Person>

<!-- Description of the person
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Silvia_Ara
ngo-Docio -->
<foaf:Person
rdf:about="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Silvia_A
rango-Docio">
        <foaf:familyName>Arango-Docio</foaf:familyName>
        <foaf:givenName>Silvia</foaf:givenName>
        <foaf:account
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#silvi
adocio_user_account_on_blogforever"/>
</foaf:Person>

<!-- Description of the user acount
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#edpinsent_
user_account_on_blogforever -->
<sioc-ns:UserAccount
rdf:about="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#edpinsen
t_user_account_on_blogforever">
        <foaf:accountName>edpinsent</foaf:accountName>
        <sioc-ns:creator_of
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogF
orever_and_migration"/>
        <sioc-ns:account_of
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Ed_Pi
nsent"/>
</sioc-ns:UserAccount>

<!-- Description of the user account
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#kasioumis_
```

```
user_account_on_blogforever -->
<sioc-ns:UserAccount
rdf:about="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#kasioumi
s_user_account_on_blogforever">
        <foaf:accountName>kasioumis</foaf:accountName>
        <sioc-ns:creator_of
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#BlogF
orever_Invenio_workshop"/>
        <sioc-ns:account_of
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Nikos
_Kasioumis"/>
</sioc-ns:UserAccount>

<!-- Description of the user account
http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#silviadoci
o_user_account_on_blogforever -->
<sioc-ns:UserAccount
rdf:about="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#silviado
cio_user_account_on_blogforever">
        <foaf:accountName>silviadocio</foaf:accountName>
        <sioc-ns:creator_of
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Great
_workshop_and_great_to_be_at_CERN"/>
        <sioc-ns:account_of
rdf:resource="http://www.foo.foo/ontologies/2012/3/OntologyTest.owl#Silvi
a_Arango-Docio"/>
</sioc-ns:UserAccount>

</rdf:RDF>
```

The root of the XML tree is the RDF element (rdf:RDF). It contains a node for each of the archived and described entities (sioc-types:Weblog, sioc-types:BlogPost, etc.). The name of each of these elements indicates the type. Thus, an additional sub-element rdf:type is not necessary. The rdf:about attribute contains the URI of the particular entity. Each sub-element represents a property-value-pair and, therefore, represents a specific RDF triple where the URI of the described entity is the subject and the name of the sub-element is the predicate. The object could be either the value of the sub-element (in case of literals) or the value of the attribute rdf:resource (in case of linking to another URI).

Figure 8 shows an example how the description of the preserved data in a BlogForever repository can be linked to the description of scientific publications in DBLP. Therefore, the following existing resources are modelled:

- A blog post taken from the personal blog of John Baez:
  http://johncarlosbaez.wordpress.com/2010/11/11/our-future/
- A post of John Baez taken from a corporate blog:
  http://golem.ph.utexas.edu/category/2012/05/quivering_with_excitement.html#more
- The representation of the person John Baez in DBLP:
  http://dblp.l3s.de/d2r/page/authors/John_C._Baez
- The representations of two publications of John Baez in DBLP:
  http://dblp.l3s.de/d2r/page/publications/conf/ctcs/Baez97 and
  http://dblp.l3s.de/d2r/page/publications/journals/corr/abs-1010-2067

The example illustrates how the data from both repositories can be linked by the indication that the Person John Baez is the same. Therefore, the property owl:sameAs is used. It can be easily added to the own RDF descriptions of the BlogForever repository. For the inverse property in DBLP, it has to be announced to DBLP and they can consider adding it as well.

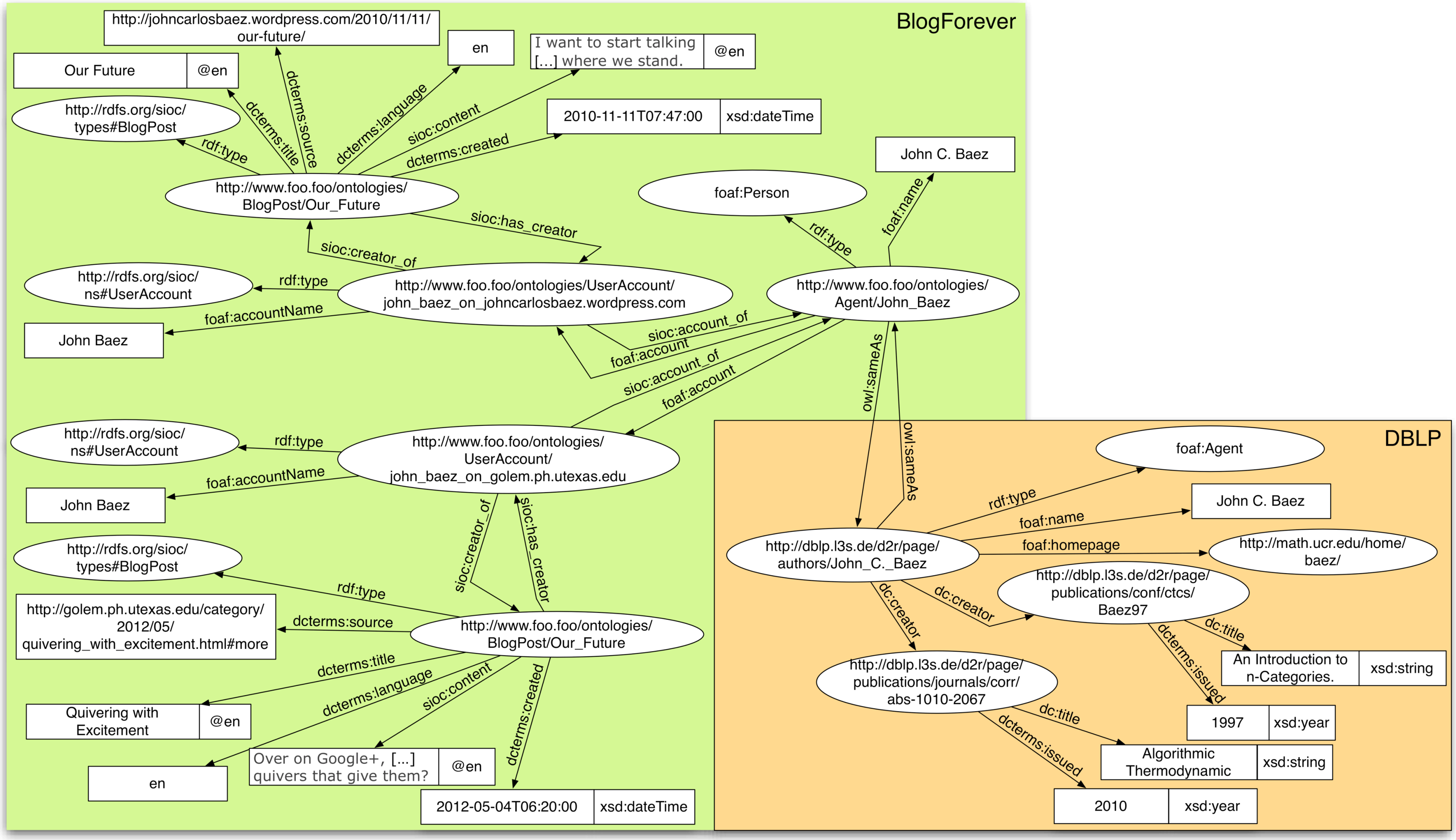

BlogForever
http://johncarlosbaez.wordpress.com/2010/11/11/our-future/
en
I want to start talking […] where we stand.
@en
Our Future
@en
http://rdfs.org/sioc/types#BlogPost
rdf:type
dcterms:title
dcterms:source
dcterms:language
sioc:content
dcterms:created
2010-11-11T07:47:00
xsd:dateTime
John C. Baez
foaf:Person
http://www.foo.foo/ontologies/BlogPost/Our_Future
sioc:has_creator
sioc:creator_of
rdf:type
foaf:name
http://rdfs.org/sioc/ns#UserAccount
rdf:type
http://www.foo.foo/ontologies/UserAccount/john_baez_on_johncarlosbaez.wordpress.com
http://www.foo.foo/ontologies/Agent/John_Baez
foaf:accountName
John Baez
sioc:account_of
foaf:account
sioc:account_of
foaf:account
owl:sameAs
owl:sameAs
http://rdfs.org/sioc/ns#UserAccount
rdf:type
http://www.foo.foo/ontologies/UserAccount/john_baez_on_golem.ph.utexas.edu
foaf:accountName
John Baez
sioc:creator_of
sioc:has_creator
http://rdfs.org/sioc/types#BlogPost
rdf:type
http://golem.ph.utexas.edu/category/2012/05/quivering_with_excitement.html#more
dcterms:source
http://www.foo.foo/ontologies/BlogPost/Our_Future
dcterms:title
dcterms:language
sioc:content
dcterms:created
Quivering with Excitement
@en
en
Over on Google+, […] quivers that give them?
@en
2012-05-04T06:20:00
xsd:dateTime
DBLP
foaf:Agent
rdf:type
John C. Baez
foaf:name
foaf:homepage
http://math.ucr.edu/home/baez/
http://dblp.l3s.de/d2r/page/authors/John_C._Baez
dc:creator
http://dblp.l3s.de/d2r/page/publications/conf/ctcs/Baez97
dc:creator
dc:title
dcterms:issued
An Introduction to n-Categories.
xsd:string
http://dblp.l3s.de/d2r/page/publications/journals/corr/abs-1010-2067
1997
xsd:year
dc:title
dcterms:issued
Algorithmic Thermodynamic
xsd:string
2010
xsd:year

Figure 8: Linking between BlogForever and DBLP

The exposed data can be queried with SPARQL if a SPARQL endpoint is available. SPARQL queries on the DBLP data can be tested at http://dblp.l3s.de/d2r/snorql/. The following simple SPARQL query applied on the data from Figure 8 would return the URI <http://dblp.l3s.de/d2r/resource/publications/conf/ctcs/Baez97> for the publication that John Baez wrote in 1997:

```
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT DISTINCT * WHERE {
  ?publication dc:creator
<http://dblp.l3s.de/d2r/resource/authors/John_C._Baez> .
  ?publication dcterms:issued "1997"^^xsd:gYear
}
```

A more complex query (see below) would select all the things created by John Baez (e.g. publications, blog posts, comments) from both repositories for the time between 2005 and 2011. Thereby, the query uses the owl:sameAs property to indicate the requested person in the BlogForever repository. The query contains two sub-queries that are merged with the UNION operation. The filtering of the requested years is slightly different for the sub-queries because DBLP uses only the year while BlogForever should use an extended timestamp with date and time. Applying this query on the data from Figure 8 would return only the URIs for the publication and the blog post in 2010.

```
SELECT DISTINCT ?publication WHERE {
  {
     ?publication dc:creator
<http://dblp.l3s.de/d2r/resource/authors/John_C._Baez> .
     ?publication dcterms:issued ?year.
     FILTER ( ?year < "2011"^^xsd:gYear &&  ?year > "2005"^^xsd:gYear) .
  }
  UNION
  {
  ?author owl:sameAs
<http://dblp.l3s.de/d2r/resource/authors/John_C._Baez> .
  ?author foaf:account ?authorAccount .
  ?authorAccount sioc:creator_of ?publication .
  ?publication dcterms:created ?year .
  FILTER (?year < "2011-01-01T00:00:00"^^xsd:dateTime &&  ?year > "2005-
01-01T00:00:00"^^xsd:dateTime) .
  }
}
```

Furthermore, SPARQL allows the creation of new RDF triples based on the existing dataset. Thus, implicit relationships in a dataset can be made explicit. For example, it is implicitly contained in example in Figure 8 that John Baez is the creator of the two blog posts because of the transitive relationship between blog post, user account, and a person. These implicit relationships can be expressed explicitly, e.g. using the creator property from Dublin Core for a direct connection between the person and the blog post. Applying the following SPARQL query on the example creates the requested relationships.

```
CONSTRUCT { ?publicaton dc:creator ?person }
WHERE {
  ?publication sioc:has_creator ?userAccount .
  ?userAccount sioc:account_of ?person
}
```

The examples shown above are kept simple and easy to understand because they should illustrate how the exposed data would look like and how they can be queried. However, the technology allows much more complex and, therefore, probably more beneficial queries. The following list shows some promising complex queries.

1. Comparison of preserved objects: Given that the same source (e.g. blog post) is preserved in two or more BlogForever repositories, it can be queried how similar or different they were crawled and stored (e.g. different formats or copyrights).
2. Co-Occurence: Given a name of a person, organisation, URI, and/or service, it can be queried for other names that co-occur with the given name.
3. Affiliations: Given a resource, it can be queried for the names of persons, organisations, and source URIs associated with the resource as evidence of trust-worthiness, authenticity, reliability, and validity.
4. Network structures: Given a group of people and/or organisations in a social network, it can be queried to find information shared or authored by the same network of people across several sources.
5. Events: Event driven query, i.e. given an event, looking for blogs and other information published during that period.
6. Topics: Given an important discovery in research, it can be queried for blogs on the same topic, by the same author, or occurring at the same time, to detect correlated publications and/or additional source of information.
7. Reciprocity: Querying for pairs of names (A, B) where A is associated to post and B is associated to a comments as a means of extracting relationships between agents. For example if (A, B) appears as often as (B, A) in a blog archive then there is a reciprocal relationship between A and B, whereas if the names are correlated to a specific order then it may indicate a different type of relationship.

3.6 LOD extraction from an SQL data base

The importance of revealing relational data and making it available as RDF and, more recently, as Linked Data can hardly be contended because large parts of Web content remain confined in relational databases that support database-driven Web applications. Rewriting a relational database, only with the public data in a RDF knowledge base would require significant efforts. Therefore an important number of tools have been developed within this scope. Some well-known applications are R2O (Barrasa et al. 2004), RDBToOnto (Cerbah 2008), Triplify (Auer et al. 2009) and D2R Server (Bizer 2006). Among these, we would describe D2R Server and Triplify as the most suitable candidates for the BlogForever project scenario. Since the BlogForever repository component is based on Invenio, which is using a MySQL database to store data, we suggest using these tools to generate LOD data on the fly from the existing database by mapping SPARQL and HTTP-URI requests to SQL.

D2R Server was introduced by (Bizer 2006), as a part of D2RQ project[49] and it is a system for extracting the content of relational databases on the Semantic Web. D2R Server uses a declarative and customizable mapping in order to map database content into its format, and enables RDF data browsing. Based on this mapping, the navigation to the content of non-RDF databases by RDF and HTML browsers becomes possible and applications can query a database using the SPARQL query

[49] http://d2rq.org/

language over the SPARQL protocol. The server takes requests from the Web and rewrites them to SQL queries via the mapping. This on-the-fly translation presents the advantage of accessing the content of large databases with acceptable response times. The generated representations are richly interlinked on RDF and XHTML level in order to enable browsers and crawlers to navigate the database content (Bizer 2006). In order to perform the mapping between database schemas and RDFS schemas or OWL ontologies, D2R Server uses the D2RQ mapping language (Bizer & Seaborne 2004). This mapping defines how resources are identified and how the property values are generated from the database content. In D2RQ, the *ClassMap* is the central object and represents a mapping from a set of database entities to a class or a group of similar classes of resources. A *ClassMap* owns some *PropertyBridges* that specify how instance properties are created and how given URIs or literals are reversed into database values. There are two types of property bridges: *DatatypePropertyBridges* for literals and *ObjectPropertyBridges* for URIs and for references to instances that are created by other class maps (Bizer & Seaborne 2004). As far as property values, they can be produced directly from database values or by employing patterns or translation tables (Bizer 2006).

A tool included in D2R Server generates the D2RQ mapping from the table structure of a database automatically. Thus, a new RDF vocabulary is generated for each database where the class names derive from the table names and the property names from the names of the columns. It is important to mention that this mapping can be customized later by substituting the automatically generated terms with terms from well-known and publicly accessible RDF vocabularies (Bizer 2006). In fact, such a customization would enable Semantic Web client applications to understand more of the data.

The mapping defines a virtual RDF graph that contains information from the database content. This is similar to the SQL concept with the difference that the virtual data structure is an RDF graph instead of a relational table. Moreover, the access to the virtual RDF graph can be achieved with various ways depending on the implementation. The D2RQ Platform provides SPARQL access, a Linked Data server, an RDF dump generator, a simple HTML interface, and Jena API access to D2RQ-mapped databases. The structure of a D2RQ mapping example is shown in Figure 9. The database scheme is mapped to RDF terms using the aforementioned *d2rq:ClassMaps* and *d2rq:PropertyBridges*. A class map specifies how URIs are generated for the instances of the class and it has a set of property bridges, which specify how the properties of an instance are created.

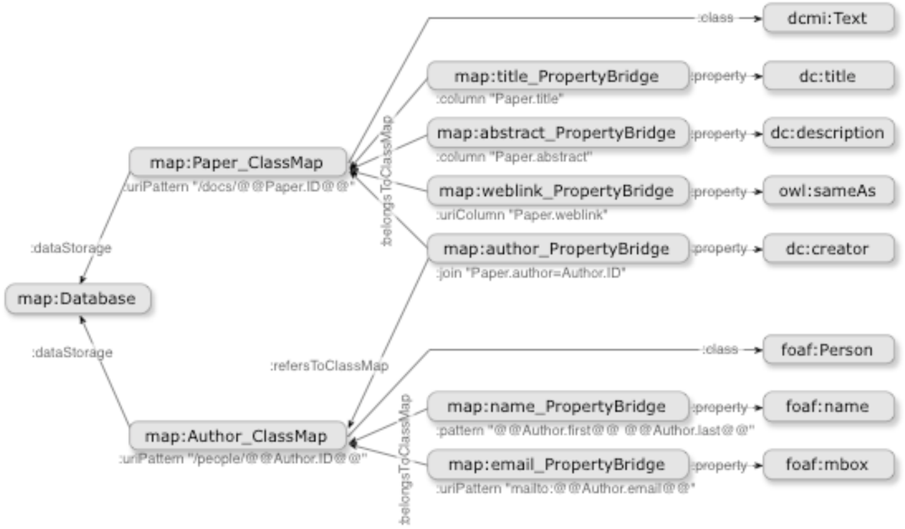


Figure 9: D2RQ mapping example[50]

As mentioned above, D2R server supports hyperlink navigation by providing links on RDF and XHTML level. Any RDF triple whose object is a dereferenceable URI can be seen as a hyperlink (Berners-Lee 2009). This is how resources published by D2R Server are interlinked with other databases and external RDF documents.

To achieve the revelation of related resources, D2R Server includes an *rdfs:seeAlso* triple with every resource description pointing to an RDF document with links to other resources created by the same ClassMap. If resources are identified with external URIs, then an additional *rdfs:seeAlso* link points to a local RDF/XML document that contains all the information that the database own about the resource. It should be noted, that by dereferencing the external URI and following the *rdf:seeAlso* link, RDF browsers can have access to both authoritative and non-authoritative information about the resource (Bizer 2006).

RDF-level hyperlinks work as “breadcrumbs” for RDF crawlers and RDF browsers through which a user is allowed to explore the Web of interlinked RDF documents. All RDF links are also available in XHTML representations while additional XHTML hyperlinks lead to navigation pages that contain lists of other resources produced by the same ClassMap, and also to an overview page that lists all of these navigation pages. This overview page provides an entry point for crawlers of external Web search engines to index the content of the database (Bizer 2006).

The applications can query non-RDF databases using the SPARQL query language over the SPARQL protocol. SPARQL provides a standard interface to data and defines a formalism by which data can be seen but, unlike SQL, which provides possibilities for changing or deleting data, SPARQL provides support only for querying the triples. The RDF repositories enable these features also, but not through the standard SPARQL query language (Pérez et al. 2009).

In brief, D2R Server presents some of the following features[51]:

- A simple web interface that allows navigation through the database's contents and gives users of the RDF data a “human-readable” preview.
- D2R Server assigns a URI to each entity described in the database according to the Linked

[50] http://d2rq.org

[51] http://d2rq.org

Data Principles (Berners-Lee 2009) and makes those URIs resolvable.

- The SPARQL interface enables applications to query the database using the SPARQL query language.
- When new classes and properties are introduced for a D2R deployment, the server can make their URIs resolvable in the spirit of Linked Data, and enables the configuration of their labels, comments, and additional properties.
- Metadata can be attached to every RDF document and web page published by D2R Server.

However, tools like D2R Server that aim to provide partially automatic generation of suitable mappings from relations to RDF vocabularies present some obstacles mainly because of the complexity in generating mappings. Such obstacles include issues about, for example, the identification and discrimination of confidential and public data contained in web applications.

From a different perspective, **Triplify** was developed by (Auer et al. 2009) as a simply applied but effective approach to publish Linked Data from relational databases aiming at lowering the entrance barrier for Web application developers. Its simplicity lies on the fact that Triplify neither defines a new mapping language nor requires the use of a new one. It exploits specific SQL notions with suitable conventions for transforming database query results into RDF and Linked Data. It is based on mapping HTTP-URI request on relational database queries and then transforming the resulting relations into RDF statements that can be later published on the Web in various RDF serializations, in particular as Linked Data.

In order to be able to reveal the structured information stored in relational databases behind the current Web, the developers (Auer et al. 2009) implemented Triplify as a light-weight solution of a software component, that would be easily integrated into already existing Web applications. Their main intentions were:

- to provide to Web developers with the ability to publish easily RDF triples, Linked Data, JSON, or CSV from existing Web applications
- to offer preconfigured mappings of some popular Web applications such as WordPress and Drupal
- to allow updates retrieval from published content without re-crawling the unchanged content

The basic concept of Triplify is the definition of relational database views for a specific Web application in order to gain the useful information that is contained in a database, and the conversion of the results of these queries into RDF, JSON and Linked Data. According to (Auer et al. 2009), for most Web applications a small number of queries is adequate to extract the important public information. When these views are generated, the Triplify tool is used to convert them into an RDF, JSON or Linked Data representation, which, thereby, can be shared and accessed on the Web. An overview scheme of Triplify is presented in Figure 10.

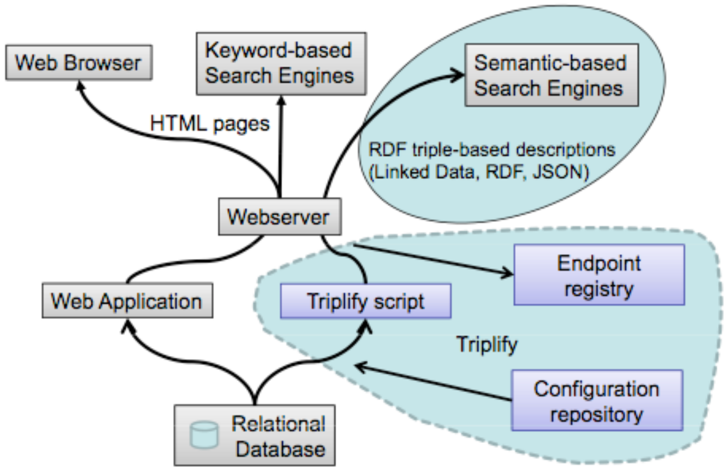


Figure 10: Triplify overview scheme (Auer et al. 2009)

Since the purpose of this scenario is the generation of Linked Data we put our focus on this representation type. The Linked Data paradigm is based on the idea of making URIs used in RDF documents accessible via the HTTP protocol. Such an HTTP request gives a description of the resource identified by the URI, i.e. a collection of all the available information related to this resource. The Linked Data paradigm of publishing RDF enables the Web crawlers to obtain information in small chunks and follow RDF links to gather additional, linked information (Auer et al. 2009). Furthermore it solves several other issues like the validity and authority verification.

Linked Data are generated by Triplify with the possibility to publish data on different levels of a URL hierarchy. To explain this Linked Data generation the developers use the following example (Auer et al. 2009):

- On the top of the hierarchy, Triplify publishes only links to classes, which corresponds to an endpoint request. An URI of an endpoint request will usually look as follows: http://myblog.de/triplify/.
- Moving down to the hierarchy, there are the URIs of a class request which would thereby look like this: http://myblog.de/triplify/posts.
- Finally, the individual instances from the classes could be accessed using the id of the instance, like: http://myblog.de/triplify/posts/13.

Triplify also simplifies the process by allowing to use the class names as URL patterns in the Triplify configuration. From the SQL queries associated with those class names (base SQL queries) in the configuration, it derives queries for retrieving lists of instances and individual instance descriptions. The base SQL view just selects all relevant information about all instances (Auer et al. 2009).

Another significant issue to examine is the update data retrieval because it is important to keep track of data updates so that crawlers know what has changed after the last crawl and should be retrieved again from that endpoint. The approach that Triplify follows is based on Linked Data Update Logs. Each Linked Data endpoint provides information about updates performed in a certain timespan as a special Linked Data source. Updates occurring within a certain timespan are grouped into nested update collections. Moving back to the previous example, retrieving http://myBlog.de/lod/update would return, for example, the following RDF (Auer et al. 2009):

```
http://myBlog.de/lod/update/2007 rdf:type    update:UpdateCollection
http://myBlog.de/lod/update/2008 rdf:type    update:UpdateCollection
```

In a similar way, the nesting continues until we reach a URL, which exposes all blog updates performed in a specific time point. Consequently, for the same example, the resource http://myBlog.de/lod/update/2008/ Jan/01/17/58/06 would contain RDF links and metadata to the Linked Data documents that were updated on Jan 1st, 2008 at 17:58:06 that would look like this (Auer et al. 2009):

```
http://myBlog.de/lod/update/2008/Jan/01/17/58/06/user123
update:updatedResource http://myBlog.de/lod/user/John;
update:updatedAt       "20080101T17:58:06"^<xsd:dateTime>;
update:updatedBy       http://myBlog.de/lod/user/John
```

Triplify is implemented in PHP and the core of the implementation is not longer than 500 lines of code, which simplifies integration into Web applications to a great extent. The implementation needs direct access to the relational database by means of either a PDO object or the MySQL driver. However, the Web applications into which Triplify can be integrated may use any other database abstraction framework (Auer et al. 2009). The deployment of Triplify in an existing Web application can be achieved within these two steps:

1. Including the Triplify script into the Web application's directory.
2. Creating a configuration for the specific Web application's database schema by defining a number of SQL queries which select the information to be made publicly available or alternatively, downloading a suitable one from the Triplify configuration repository. This repository so far contains mappings for many popular Web applications, including WordPress, Drupal and Gallery.

It should be mentioned that Triplify was initially developed for small or medium Web applications. However, since Triplify's application logic is simple and almost all work is pushed down to the database, Triplify can also be useful to large databases. The fact that Triplify uses SQL as a mapping language, instead of a newly developed one makes it advantageous in the following ways (Auer et al. 2009):

- SQL supports many features, which are currently unavailable in other mapping approaches like, for instance, aggregation and grouping functions or complex joins.
- The fact that it is based on SQL views allows to push almost all expensive operations down to the database which positively affects the overall scalability of Triplify
- Triplify does not requires any additional knowledge since software developers and even server administrators are usually skilled in SQL. Therefore, users can employ semantic technologies while working in a familiar environment.

In summary, Triplify is a small, lightweight plugin for Web applications, which reveals the semantic structures encoded in relational databases by making database content available as RDF, JSON or Linked Data. Probably the largest benefit when using Triplify is that a Web application becomes easily mashable with other Web data sources.

3.7 Scenario conclusion

The scenario described in this chapter demonstrates how the exposure of preserved data in a BlogForever repository as LOD can facilitate interoperability with other BlogForever repositories as well as with other digital libraries. The vocabularies Dublin Core, Friend of a Friend, and Semantically-Interlinked Online Communities, which are widely adopted in the Semantic Web community, as well as the PREMIS ontology, which covers specific concepts for preservation, has been used to describe an important subset of the BlogForever data model. The limitation to a purposive subset has been made because the exposed concepts should be applicable across as many blogs as possible, and should sufficiently overlap with the properties of material held within other repositories that do not necessarily specialise in weblogs. Based on the exposed linked data, queries can be performed for various reasons (see the given examples) using the query language SPARQL.

Furthermore, the tools D2R server and Triplify, which are able to generate LOD from SQL databases, have been examined because the BlogForever repository is using a MySQL database. Thereby, Triplify is the more simple while D2R server provides more functionality. For further developments in the BlogForever project, the D2R server is recommended because of a missing SQARQL feature in Triplify (it is planned but not yet implemented).

In the BlogForever project, the findings from this scenario are influencing for the Task 3.2: Assessment of Interoperability Prospects[52] and during the design and implementation in Work Package 4: Software infrastructure[53].

[52] Description of Work, Part B, pages 32-33

[53] Description of Work, Part B, pages 36-42

4 Semantic extension of tags

User generated Tags and resulting folksonomies are widespread in social media applications like social bookmarking, social networking, wikis, and blogs. However, the collaborative processes that generate these folksonomies in these platforms can differ significantly. In platforms like Delicious[54] many users can tag the same resources, while in the case of blogs the resources are tagged by a single user. In fact, this distinction is commonly known as the distinction between *broad* and *narrow* folksonomies. Thus, broad folksonomies are generated as a result of aggregating data for many people tagging the same resource while narrow folksonomies are the result of data aggregation from single users tagging their own resources (Helic et al. 2012). Consequently, it should be noted that a folksonomy generated by weblogs is considered a narrow one.

Folksonomies offer a flexible bottom-up approach to organise resources. However, while tags can organise blog posts inside a single blog according to the understanding of the blog author(s), it becomes more complicated if posts are aggregated from various blogs with possibly different contexts and topics. Therefore, it is necessary to identify and expose the meaning of the tags to overcome problems that result from the free choice of tags by different users, like homonyms and synonyms, and impair content retrieval.

This scenario addresses these particular challenges. It is the description of a methodology that aims at enriching the folksonomy not only with explicit relations between tags, but also with additional related terms that can be either extracted by the weblog post texts or found in lexical databases. This ontology will be modelled with existing available tag ontologies and will facilitate information organisation and retrieval.

The scenario is organised as follows. Chapter 4.1 presents the scenario description. Chapter 4.2 presents briefly an approach to expose semantics from the weblogs folksonomy and build an ontology by them. Chapter 4.3 examines several vocabularies intended to represent the semantics. Chapter 4.4 presents a summary of the approach and, finally, Chapter 4.5 contain the scenario conclusion.

4.1 Scenario description

Beside the interoperability with other repositories, the BlogForever project puts an emphasis on the possibilities of data retrieval from a single BlogForever archive. Requirements taken from project descriptions and interviews with several interested groups indicate the need for an extensive capability of exploring activities. One powerful possibility is the utilisation of user generated tags and resulting folksonomies. The exploitation of user-generated tags can be highly beneficial presenting advantages: the effort for classifying content is outsourced to the user crowd, the classification terms reflect the language of the users, and the tag concept is flexible enough for the demands of a fast changing and growing field of subjects. However, folksonomies lack explicit semantic. Tags are just words or phrases without an explicit description of their meaning, or their relations to other tags. Therefore, the use of tags is accompanied by problems, e.g. a lack of hierarchies between the tags, and the synonym or homonym use of tags by different authors (Trant 2009; Passant et al. 2009). The problems that arise in blogs are described extensively later in 4.2.2. The objectives that are addressed in this scenario are summarised in Table 5.

BlogForever aims	Preservation	Management	**Dissemination**
High level aims of	**Effective data mining**	Efficient preservation	Robust repository

[54] http://delicious.com/

ontology use					functions	
Internal/External	Internal perspective			**External perspective**		
Described object	Blog structure		**Network of blogs**		Blog content	
Process perspective	Blog aggregation		Management		**Access & Distribution**	
Functions of the archive	Interoperability	Archival functions	Policy-based management	**Semantic search & navigation**	**Merging of social web aspects with ontologies**	Ontology -based services

Table 5: Objectives addressed in the Semantic extension of tags scenario

There are possible ways to overcome the above-mentioned problems with folksonomies by the use of ontologies. Firstly, users could be asked to extend their tag expressions with additional semantic (e.g. affiliation to an unique identifier related to an explicit meaning (Passant et al. 2009)). Depending on the additional effort for the user, in terms of intellectual effort as well as physical effort like additional clicks, this approach may reduce the user's willingness to contribute further tags. However, another possibility, that does not require user effort, is to exploit the potential semantic relations between tags that can be revealed through data mining techniques, social network analysis and online lexical resources. Thus, ontological structures like hierarchies between tags can be proposed (Mika 2007). Research in this area has already been conducted, but identified approaches can still be considered as experimental.

In the BlogForever project, three kinds of tags should be considered. First, tags will be crawled from the blogs that should be preserved and archived with them. These tags should be called *author tags* because it is very likely in most cases that the author of the blog respectively the blog post has linked the tag to the post. Users who explore and use the BlogForever platform will probably create a second kind of tags in the repository. These tags should be called *reader* or *user tags* because people who have read the blog in the archive create them. The creation process of user-generated tags can be influenced only for reader tags. Therefore, the extension of user generated tags with an explicit semantic description by the user crowd can be facilitated (or forced) only for reader tags. A third kind of tags is generated *automatically*. For example, they can be extracted from the text corpus of the resource[55] or they can be generated from related resources (Kurz et al. 2012). The latter is interesting particularly for these resources that contain little or no text (e.g. pictures, movies) and, therefore, cannot be searched with a full text search. Furthermore, the generated tags can also be enriched with explicit semantics[56]. However, the automatic extraction of tags is limited to predictable tags while user generated tags can create new and unexpected annotations depending on the tagging context of the user. Therefore, it will be promising to pursuit different approaches in the BlogForever project.

[55] See BlogForever Deliverable D2.6: Data Extraction Methodology, Chapter 8: Post-Processing and Data Extraction Associated Technologies.

[56] See for example DBpedia Spotlight: http://dbpedia.org/spotlight

The aim of this scenario is to enrich the semantics in the blogs folksonomy. The basic idea is to build an ontology derived from all the tags that describe the posts within a BlogForever archive. In such ontology, tags will be associated with each other with relationships that will indicate whether there is equivalence, synonymy, hierarchical relationship or just some association. In addition, this ontology will not derive only from the existing authors' tags, but it will be also enriched by additional terms that we will either extract from the text of blogs or discover with the help of online sources like WordNet[57].

In the following section we present possible approaches and methods of building this ontology. More specifically, these techniques aim at extracting relationships between tags, tags pre-processing, discovering possible synonym terms and mapping the tags in available ontologies. However, these methods are not independent but support each other and so they are combined and summarized in one approach. After the ontology is built, the semantics must be described and modelled and therefore in section 4.3 we discuss the existing ontologies and vocabularies that could be used for the structure and modelling of this ontology. These vocabularies provide classes and properties to represent relational information of tags in our ontology, like equivalent terms, associated terms, and broader or narrower concepts.

4.2 Building the ontology

A collection of weblogs is certainly a rather large source of information that can be exploited and therefore it presents a great potential for exposing semantics. Building an ontology out of this information can improve the organisation of this often-large collection and can facilitate readers' queries and information retrieval.

To achieve this, we need to consider carefully which information is valuable for our purpose,

- Which methods we could adopt to extract it,
- What existing resources and tools could contribute, and
- What possible constraints we may need to handle.

In this section we examine methods to extract and enrich semantics through the folksonomy of a collection of weblogs, like a BlogForever archive, with respect to the particular type of folksonomy resulting from blogging platforms. This means that the fact that tags are freely chosen and assigned only by authors is strongly taken under consideration and the methods follow this direction.

In the beginning of the section, there is a description of possible techniques with which it is possible to extract additional terms from the text of the posts. Afterwards, we briefly explain what problems arise by the freely chosen tags in weblogs and propose some lexical filtering methods to address some of them. The rest of them are mostly addressed in last section, where we describe how we can deduce relationships between tags using the folksonomy, generate groups of associated tags, identify the type of these relations using mainly lexical resources and, lastly, map the tags to existing available concepts.

4.2.1 Term extraction

Undoubtedly, the tagging activity differs significantly among bloggers. Some bloggers may use several tags to describe their posts while others can pick just a few. Besides, not all blog posts contain tags neither all bloggers tend to assign tags to their posts. Furthermore, even when bloggers provide sufficient number of tags, the tags they choose to describe same concepts can considerably vary as well (see next section). However, while tags for the same concepts can be so different, the

[57] http://wordnet.princeton.edu/

texts can proved to be a more reliable constant to identify the topic of a post since it is rather likely that the bloggers will commonly use the same vocabulary and terms when referring to same topics. In any case, it is very likely that the text of a post will contain several useful words that are not defined as tags by the author and could contribute to our aim in two ways: to enrich the tag set or to extract words that even if they are not suitable as keywords (because, for example, they are too general), they can contribute in discovering associations between related tags in the entire tagspace. Thereby, it becomes clear that the exploitation of the text of a post can be beneficial to our aim.

Therefore, the adoption of a term extraction method that could provide an additional set of tags for each post based on the text would help us mine more information. Term extraction or, more widely known as, keyword extraction is the task of identifying a set of words or phrases that describe the meaning of a text. Table 6 illustrates an example of a weblog post[58], the actual tags by the author and some additional derived from the tag cloud of the text. As we can see, the extracted tag set contains some relevant words that could be also tags of the post. For example, the terms "camera" and "pictures" are certainly words that are very associated with the topic photography.

Post title	Author Tags	Tag cloud[59]
How to Get Sharp Photographs	photographic techniques, photography tips	tripod, camera, lens, shutter, pictures

Table 6: Example additional tags from tag cloud of a post

For the development of a term extraction technique there is already a rich bibliography and several proposed methods that vary from simple to more sophisticated techniques (Kaur & Gupta 2010; Hulth 2003; Feifan Liu et al. 2009) According to (Kaur & Gupta 2010) the existing methods can divided in four categories: Statistical, Linguistic, Machine Learning and Combined approaches. Fore example, a simple method could be easily implemented by extracting the most frequent words in text that are not stop-words, verbs and adverbs. Stop-words are these words that appear very often in a text but do not provide any useful information (like "the", "after", "without" etc.) and can be found in free available lists. Obviously, stop word lists are different for each language and that must be considered in the implementation. Finally, a definition of the most frequent word must become explicit. For example, the extracted keywords could be the n most frequent words that appear more than k times in the text where n and k would be variants that depend on the size of text and the number of unique non-stop words. Alternatively, another quite simple method could be the use of *tf-idf*[60] weights, that are able to indicate the terms that are more important in one document in comparison with the whole dataset and consequently they are more representative of a single blogpost.

However, other more complex but definitely more effective and sophisticated approaches could be adopted alternatively. In fact, it is recommended to apply a rather sophisticated method in order to avoid adding noise to the tagspace by the extraction of words that are frequent but do not contain information.

Another promising idea is to use an existed available tool for tag suggestion through the text. For example, Zemanta[61] API analyses unstructured text and return five types of content objects like

[58] http://blog.proudphotography.com/2011/02/25/how-to-get-sharp-photographs/

[59] Tagcloud by http://tocloud.com/

[60] http://en.wikipedia.org/wiki/Tf*idf

[61] http://zemanta.com

machine-readable static tags, relevant pictures from Flickr or related articles. For the example of Table 6, Zemanta suggested the tagset {Camera, Shutter speed, Digital single-lens reflex camera, Gitzo, Tripod, Photograph, Image stabilization, Photography}, in which most of the tags are associated with the specific post. Similarly, Calais[62] provides a set of suggested tags for a given text. Thirdly, DBpedia Spotlight[63] could also be used for this purpose. Although DBpedia Spotlight is not really a tag recommendation system, however it can distinguish key words inside a text. However, it should be considered that there are limitations in daily requests to APIs and thus, a developed approach would be probably more beneficial.

Eventually, adopting a term extraction approach similar to the aforementioned would provide additional tags for each post that contains text. However, it should be mentioned that the extracted set of tags aim only at assisting in the ontology building without influencing the original post and tags. Within the framework of preservation, the original tag set of each post will be preserved unaffected. The extracted tags from such a technique will constitute an additional automatically generated tagset that will be used to support the latter phases and enrich the semantics.

4.2.2 Tag Filtering

When freely chosen tags, without any pre-defined vocabulary or recommendations, create a folksonomy there are a number of issues that must be taken under consideration. Tagging seems to be an easy and natural way for people to classify objects and discover new material, simply and without requiring a lot of thinking. However, people think and tag differently and even a single user's tagging practice may vary over time (Begelman et al. 2006).This creates a noisy and rather boundless tagspace, which makes it difficult for someone to discover material that is tagged by other people.

Fortunately, in the case of weblogs authors are more mindful in their choice of tags, than in cases of tagging systems like Delicious, because they actually intend their posts to be easily retrieved. Thus, it is rather rare to meet in a blogpost tags that were created in a blogger's personal context and are not understandable to others. Nevertheless, a number of important issues must be considered in order to facilitate the queries for blog content based on tags and the problems that appear must be examined. Among them, the most important weaknesses that affect content retrieval in blogs' folksonomy are:

- Spelling variants of the same term, which can be plural or singular or other forms of the same word, for example "blog", "blogs", "blogging" and also "weblog" and "weblogs" are considered as having the same meaning. Not to mention that multiple variations in the use of capital letters make the different conventions even more.
- Synonyms, i.e. different words having identical or similar meaning, like "lorry" and "truck" or "buy" and "purchase".
- Ambiguity, when some users may use very specialized terms, whereas other tend to use more general tags to annotate similar topics.
- Polysemy, when the same tag can have different meanings in different contexts. For example, the tag "apple" can refer either to the fruit or to the company depending on the context in which it appears.
- Key Phrases instead of keywords, which in fact are parsed separately, e.g. "web tool" would be handled as two terms

[62] http://www.opencalais.com

[63] http://dbpedia-spotlight.github.com/demo/

- Concatenating words, for example “SemanticWeb”, “Semantic_Web” and “Semantic-Web” can be the results of the lack of consistency among the bloggers in choosing tags for a compound term.
- Encoded words, for example “nyc” instead of “new york city”.
- Typo errors that may happen and result in a noisier tagspace, for example “sofware” instead of “software”.

This freely generated tagspace makes it difficult for a user to retrieve all the desired resources without knowing all the possible variants of the tags that may have been used. Thus, a filtering process is necessary to contain a lexical processing of the tags in order to deal with as many issues as possible, like misspelled words and the multiple variants of the same term.

Therefore, this filtering step should be able to recognize the correct form of the misspelled or encoded words and match them to their equivalent, to identify the similar variants of words and associated them with each other and to separate encoded key phrases. To accomplish this phase we can adopt string similarity methods like the Levenshtein distance[64], which is a string metric that counts the differences between string sequences and can be used to identify slight changes between two strings like plural forms, title-case or typos (for example “blogs” with “blog”, “Web” with “web” and “library” to misspelled “libary”). For misspelled words, it is also possible and simple to exploit the lists of some common typos that are available by Wikipedia[65] and lexical resources like WordNet, Leo Dictionary and Wikipedia. In case a tag is not retrieved by any of these resources then we can use the frequency of the term as an indication of a new word (high frequency) or a misspelled word (low frequency) (Damme et al. 2007). Last, since similar words like “blog” and “blogging” won’t be classified as similar according to the Levenshtein distance, a stemming algorithm could also be applied or stemming software like ClairLib[66] could be used, to identify the words that derive from the same stem.

It is important to note that the term extraction technique can also contribute to the identification of spelling variants and to polysemy problem as well. For example, let’s assume that a weblog post about benefits of apple in health have “apple” as an author tag and this process extracts also the plural form “apples” from the article. Such an extraction would not only help to identify the two tags as variants but would make the context more clear as well, and therefore it would be more easy to distinguish whether the term is referred to the fruit or to the corporation.

Since the original tagset must be preserved and any possible noise in the initial tags cannot be edited, the above-mentioned filtering step takes place at the ontology level. When it is completed, a great part of incorrect or informal words has been mapped to the correct terms and words that are spelling variants are identified. However, this method isn’t enough to handle all the aforementioned constraints of a resulted folksonomy. Ambiguity, polysemy and joined terms would be handled by the exploitation of context and tags correlations, which are addressed in following methods.

4.2.3 Enriching folksonomy with terms and relations

It is worth mentioning that the tagging process generates more information than merely tags since not only tags but also objects and actors are involved in this process. This information lies under cover in a network of objects, people and tags that are associated with each other and provide a great potential to enrich semantically the folksonomy by discovering and utilizing these relations.

[64] http://en.wikipedia.org/wiki/Levenshtein_distance

[65] http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings

[66] http://www.clairlib.org/

Related works in this area follow basically two lines: they either utilize these associations or use online sources like WordNet to identify the concepts of the tags. The approaches also depend on the type of data and tagging systems. Blogs, as previously mentioned, result in a narrow folksonomy since only authors are allowed to tag their own posts. Consequently, in the BlogForever platform, the main entities involved, that is the blogposts, the tags and the authors, are connected with implicit associations between them with the difference that authors can use the same tags but not annotate the same objects. An illustration of these connections is showed in Figure 11. Therefore, in our case both of the research lines can be followed, meaning that we can utilize both online sources and the implicit relationships between the actors (authors), the items (posts) and the tags.

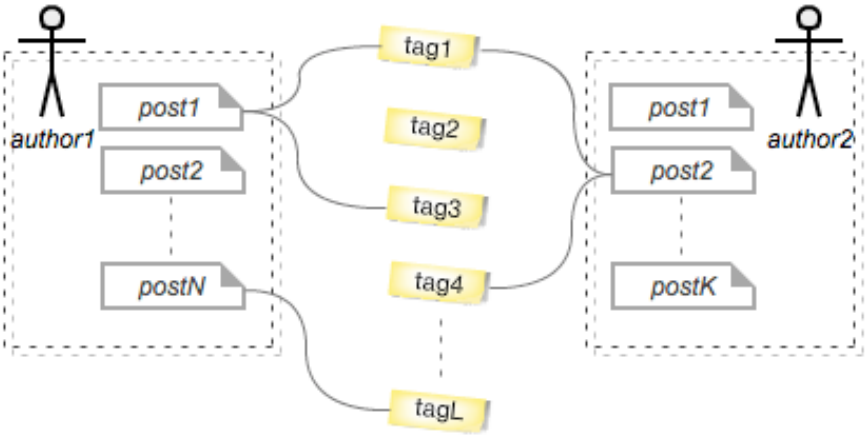


Figure 11: Entities association in Blogs

We decide to exploit this opportunity and use a combination of the two directions believing that none of them alone would be sufficient. On the one hand, exploiting relations using co-occurrences and network provide tags associations without defining the exact relationships between tags (Angeletou et al. 2008). On the other hand, using only lexical resources and treating each post or tag individually would present several other weaknesses like:

- Ignoring tags associations that exists in whole blog collection
- Difficulty to identify concatenated or encoded terms
- Some terms may not exist in any lexical database
- Time consuming and non-scalable process

Thus a combination of the two lines seems more promising and more efficient as the one approach will support the other. Consequently, the core idea for the semantic enrichment of the folksonomy in a BlogForever archive would be summarised following:

- Utilizing the relationships in the network of folksonomy, meaning to discover related tags based on tags co-occurrences by applying statistical analysis and clustering techniques. The result of this analysis could be either groups of similar tags or graphs that indicate the more related to each tag terms in the whole tagspace. The exploitation of blogrolls, when available, could be also useful to this point adding perhaps some additional weights to the tags relations, since an author is very likely to recommend blogs of similar topics.
- Discovering synsets (set of synonyms) from online resources (WordNet, Semantic Atlas) and adding them as related tags. Furthermore, retrieving translations of terms from DBpedia or other resources, when available, could also be useful.

- Identifying the type of relations between tags using the available lexical resources and make semantic relations explicit.
- Mapping the concepts to correspondent URIs when it is possible.

By the end of this phase, the initial author tags will be associated, and several of them explicitly, with each other or with additional terms derived from online resources. These steps are described in detail below.

4.2.3.1 Deducing tag relations from the network

In this section we examine how the network of bloggers, posts and tags can be utilized for the identification of tag associations. Figure 12 shows the relations between the concepts of tag, post, and author. In the physical representation a tag is directly related (linked) to the post, as well as the author, for being stated together with the post. Semantically, it means that the author has written the post and the tag describes the post with a single word or phrase. Additionally, it can be assumed for the most weblog posts that the author has linked the tag to the post.

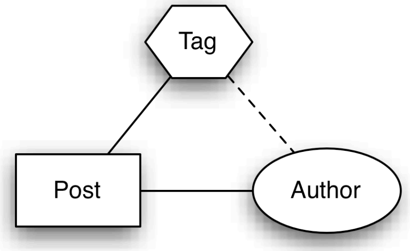


Figure 12: Relations between tag, author, and blog post

The relations between tag, post, and author can be used to deduce different two-mode-networks. For example, a network of posts and tags can be created (see Figure 13). Thereby, a post can be described by several tags and, therefore, is linked to these tags. Additionally, a tag can be used as description for several posts and, therefore, links to each of these posts. The strength of the links is equal because in blogging platforms a tag can be linked to the same post only once.

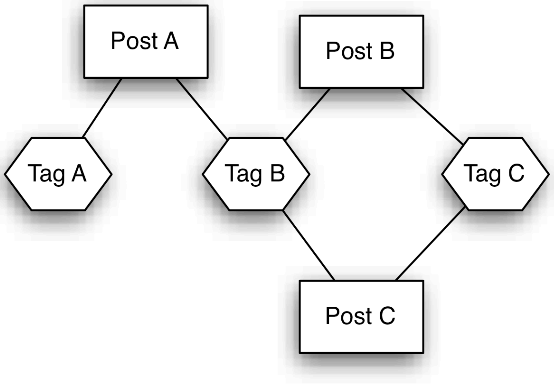


Figure 13: Two-mode network of posts and tags

A second two-mode-network can be created based on the indirect relation of authors and tags (see Figure 14). An author would be linked to all the tags that he has assigned to his posts. Thus, the tags would be linked to several authors if they describe posts by using them. The strength of the links in such a network would represent how often the author has used the tag in his posts.

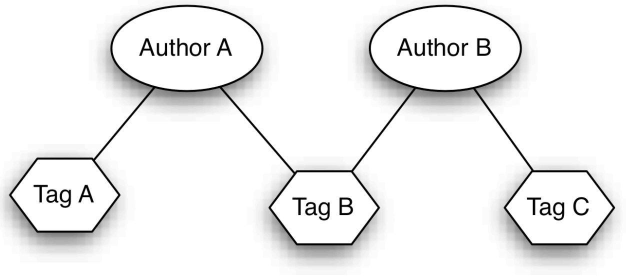


Figure 14: Two-mode network of authors and tags

The described networks could be similar, e.g. if authors write very few posts. However, they would differ at least in the strength of the links if the amount of authors writing activities varies. Both networks can be used separately or in conjunction to deduce a one-mode-network of tags (see Figure 15). Such a network can be represented by a graph in which nodes stand for tags and the edges between tags are weighted according to how strongly the tags are related to each other.

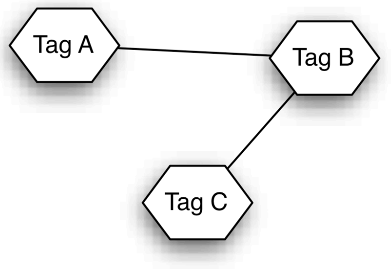


Figure 15: One-mode network of tags

The structure of this network can be beneficial to deduce conclusions about the tags and the relationships among them. For example, network measurements for centrality (e.g. betweenness) could indicate if a tag is more general (high betweenness) or specific (low betweenness) and thematic clusters of related tags can be identified as well. This becomes more clear in the evaluation results in (Mika 2007).

In several works so far the authors were based on tag co-occurrences to extract relations between two tags. In (Begelman et al. 2006) an algorithm was proposed to identify strongly related tags based on co-occurrences of tags in pages, producing a graph where related tags are connected with weights that arise from their co-occurrences and then applying a graph-clustering algorithm. In (Mika 2007), authors model the network of a folksonomy as a tripartite graph that could be simplified in three bipartite graphs that model the associations between actors, concepts and items. In (Specia & Motta 2007) authors performed statistical techniques in the tagspace to identify clusters of possibly related tags based on the similarity among tags given by their co-occurrences. Thus, they produced tag vectors that were indicating the co-occurrences of each tag with all the others and afterwards, using a similarity measure, were able to find for each tag the most related tags in the corpus.

Similar approaches can also find place in the weblogs folksonomy. However, it should be noted once again that the blog data network differs from a tagging system in the sense that each item is annotated by a single actor, that is the author. Thus, our approach would be oriented in this direction. Nonetheless, since the option for BlogForever users to add their own tags is under consideration, that would change the structure of network to one more similar to a tagging system and various other approaches could be also used, exploiting also the co-occurrences of tags in same resources and the relations among users who annotate same resources.

Hence, the idea, which is strongly motivated by the above and other similar works, is to find relations between tags applying statistical analysis and unsupervised learning techniques to the available data. (Mika 2007) described the network of a folksonomy as a tripartite graph that could be simplified in three bipartite graphs that model the associations between actors, concepts and items. However, in a blogging platform there is no network between authors and posts since each resource, i.e. each post, is tagged only by one actor, the author. On the other hand, we believe that the actor-concept network might lead us to very generalized relations between the terms. The concept-item graph, however, where the links between terms are weighted by the number of instances (posts) that are tagged with both terms (co-occurrences), seems more promising to be examined for our purpose. A combination of the two networks, both actor-concept and concept-item, is also a thought that remains under consideration, but currently we focus on the network between posts and items.

Thus, the blogposts of a BlogForever archive will constitute the input instances in this analysis and the corresponding tags will be the attributes that describe them. That means that each post would be represented as a vector of *n* features, where features will derive from the terms or can also be the terms themselves.

Therefore, the first task is to define the feature space of our problem and ensure that it would be as less noisy as possible and that it will help to make the division of the problem to as distinct regions as possible, where each of the regions will contain similar instances. However, the initial tags, as noted previously, might be phrases separated with spaces or other punctuation marks and might also contain stop-words (see **Error! Reference source not found.**). So, a slight pre-processing of terms is need and thus the tags will be parsed as single words, meaning that phrases or concatenated terms will be divided, and stop-words will not be included in the feature space. We can also decide if we prefer this feature space to include also the extracted tags by the text of the posts or not. Eventually, a collection of *k* unique words will be produced and will comprise the final feature space of our problem, since every item of the data, that is every post, will be represented as a linear combination of these features that would indicate the presence of tags or not in the post.

In particular, each one of the posts will be represented by a *k*-length binary vector *u*, where the value u_i will be 1 if the post is described by the *i* tag and 0 otherwise. Gathering all these vectors in a matrix of instances *B*, where each line stands for a post and each column for a tag, and extending the (Mika 2007) graph matrices for the case of concept-object network, we are then able to produce the following significant matrices:

- $S = B \cdot B^T$, which contains the numbers of tags that are shared between each pair of posts
- $O = B^T \cdot B$, which contains the co-occurrences of each pair of tags in the whole collection of blogs.

It is rather obvious that these matrices can help us to extract valuable information. For example, since matrix S, contains information about the number of tags that two posts share, it can be used for post recommendations. More specifically, let's think of a reader who is interested in the post *j*. Then, we can have an indication of similar posts by simply taking line *j* of matrix *S*, and see which of the posts present the higher numbers of shared tags or, in a different version, adopting some rules and thresholds like, for instance, recommending the posts that are described by all of its tags or the majority of them.

However, in our case we are more interested for the information contained in the co-occurrences matrix *O*. This matrix contains a very helpful representation for each tag and that because for the description of each tag all the other tags are taken into account as context, meaning that in order to identify two tags *i* and *j* as similar it is not enough to ensure that tag *j* often co-occur with *i* but also that it co-occurs with the other tags co-occurring with tag (Specia & Motta 2007). This representation provides some interesting options of exploitation.

Firstly, such a correlation matrix could give a relation indicator for each pair of tags and could help to construct a graph of tags where associated tags would be connected through edges weighted with these indicators like in (Begelman et al. 2006). It should be noted, though, that the number of tag co-occurrences could lead to unbalanced weights in cases where some topics are addressed to several posts while others are only described in a few. Thus, a kind of normalisation could be applied to these weights like and possibly to the vectors as well, for example, a division by the total number of commonly shared posts or something similar.

The co-occurrence vectors, i.e. the lines of the co-occurrence matrix, are representations of each tag in the tagspace and indicate how regularly the tag appears in common with all the other tags. Thus, an additional use of these vectors would be to use a similarity measure to count how similar are the vectors that belong to a pair of tags, meaning how similar pattern of co-occurrence they have. Then, we can obtain for each tag a list of its similarities to all the other tags (Specia & Motta 2007) and therefore, find for each tag the most similar terms in the corpus. There are several similarity or distance metrics for vectors, like Euclidean and Manhattan distance, each of them might perform better to different kinds of data or problems. We choose to use angular separation like in (Specia & Motta 2007) or, more widely known as cosine similarity, which is simple and sensitive to slight changes. However, we can also experiment other metrics to discover which could be more suitable for the specific data.

The cosine similarity between two tag vectors would provide a value between 0 and 1, where 1 indicates equal vectors. To identify the most similar tags for a single tag a threshold value of similarity matrix must be defined. This threshold could be either defined from the beginning or, better, be decided after some experiments on the data. An alternative version could be to choose and keep the *n* most similar tags for each term.

There is also a further and more suitable option to be considered; the most widely applied method in similar problems is clustering. Clustering could be applied in both kinds of vectors:

- to the co-occurrences vectors in order to produce clusters of similar tags
- to the initial vectors of posts, i.e. lines of matrix B, producing clusters of thematically similar posts from which we can also extract similar tags using the centroids of these clusters.

Either way, clustering can provide groups of highly related tags. This way, these tags can initially identified as related and can provide small easy-handled groups where we can then apply some of the rest techniques to identify equivalent words and spelling variants (as mentioned in 4.2.2) or make these relations more explicit using lexical resources like WordNet (see following section).

It should be mentioned that a tag might often belong to more than one cluster. For example, the tag “apple” can appear in a group with tags that regard health and fruits, and appear also in other groups concerning MacBook and iPad. Therefore, by clustering we also deal with the constraint of polysemy and exploit the context by each group to find the correct meaning for a term with multiple different meanings.

4.2.3.2 Grounding relatedness and extracting additional synonym terms using an online lexical database

Available lexical databases in web have proven to be useful in discovering synonyms. Some of them can also contribute in identifying relations between terms. WordNet is probably the most widely appointed to similar folksonomy-related projects. In (Angeletou et al. 2008), WordNet was used to assign senses to tags based on context and to extract relevant synonyms and hypernyms to achieve a richer representation of tag. In (Specia & Motta 2007) authors used WordNet to decide the representative term of a group of similar tags when in the stage of pre-processing the tags.

WordNet can be exploited in two ways. On the one hand, WordNet can provide a set of synonym terms, i.e. a synset, which may have not appeared in the tagspace, neither from the authors nor from the term extraction process. In this way, the ontology is enriched with additional terms, which are synonyms or derivationally related forms and for which we already know how they are related to the initial tags. This can be also be achieved by the Semantic Atlas[67], which provides an English and a French synonym dictionary. Note that, unless these terms were also found in posts, the extraction of synonyms via these sources does not affect the tagspace. The retrieved words will be mapped as synonyms to the original and extracted tags.

On the other hand, given a group of related tags produced by the aforementioned statistical method, WordNet can be used to ground the association between these tags or make these associations more explicit. Such a group would contain tags that are probably related to each other, since they co-occur frequently, but we do not know what are the exact relationships. Besides, there are also cases where terms appear in common repeatedly but are not really related as meanings.

Consequently, these lexical resources will contribute in the identification of three kinds of relations between terms:

- Synonyms
- Derivationally forms
- Hierarchical relations (see following paragraph)

Therefore, taking each pair of tags in a group of related, and searching in the lexical database if there is a formally defined connection between the terms, will lead to make some of the relations more explicit. Furthermore, we can identify if some of the tags in the group are not found related to any of the rest terms and eliminate them from the group. Finding similarities can be accomplished using some similarity rules, which can be based, for example, on the distance in the WordNet hierarchy (Cattuto et al. 2008) or on the common ancestors (Angeletou et al. 2008).

Other lexical resources that have been also used for the same purpose are Google and Wikipedia (Damme et al. 2007). Another promising database is UWN[68], which is an automatically constructed multilingual lexical knowledge base based on WordNet, which provides meanings, parent-terms and the translation of terms.

4.2.3.3 Identifying Hierarchies

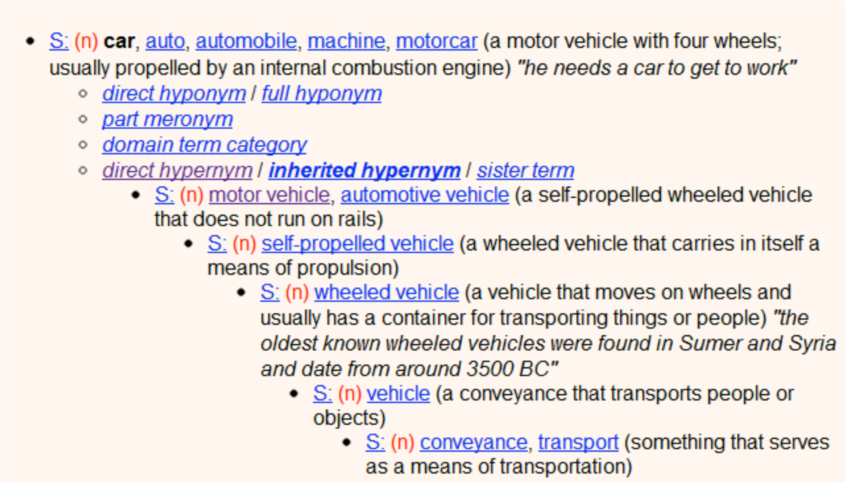
So far with the previous phases, we accomplish to semantically enrich the folksonomy with synonym term and relations that indicate related or equivalent terms. However, to reduce ambiguity we also need to identify the hierarchical relationships that may exist between the tags. Additionally,

[67] http://dico.isc.cnrs.fr/en/index.html

[68] http://www.mpi-inf.mpg.de/yago-naga/uwn/

we can add some parent-terms for tags, enriching this away further the ontology. This last step can be accomplished by the combination of the following ways:

- Using existing hierarchies from WordNet, which is a generic hierarchy of concepts, when that is possible. Such an hierarchy is demonstrated in Figure 16.



- S: (n) **car**, auto, automobile, machine, motorcar (a motor vehicle with four wheels; usually propelled by an internal combustion engine) *"he needs a car to get to work"*
    - *direct hyponym* / *full hyponym*
    - *part meronym*
    - *domain term category*
    - *direct hypernym* / ***inherited hypernym*** / *sister term*
        - S: (n) motor vehicle, automotive vehicle (a self-propelled wheeled vehicle that does not run on rails)
            - S: (n) self-propelled vehicle (a wheeled vehicle that carries in itself a means of propulsion)
                - S: (n) wheeled vehicle (a vehicle that moves on wheels and usually has a container for transporting things or people) *"the oldest known wheeled vehicles were found in Sumer and Syria and date from around 3500 BC"*
                    - S: (n) vehicle (a conveyance that transports people or objects)
                        - S: (n) conveyance, transport (something that serves as a means of transportation)

Figure 16: WordNet hypernym example of term "car"

- To identify if there are hierarchical relations in a group of related tags, produced as explained before, we find the hierarchy of each tag and then search whether the rest of the tags are identified in it. To include concepts of the hierarchy in the ontology as new additional terms, a threshold upper-level can be decided, and the hypernym concepts that are no higher in the hierarchy than the threshold can also be included to the ontology as parent-concepts of the particular word. For example, in Figure 16, the term car is a child-concept of terms vehicle and transport. To adopt such techniques, a set of rules must be adopted. For example, if some of the words don't exist in WordNet, then equivalent words, if available, can be checked alternatively.

- Deducing hierarchical relationships from the folksonomy based on the assumption of superconcept proposed in (Mika 2007) for the tags that are not available in WordNet. According to this assumption, in an ideal situation we can say that concept A is a superconcept of a concept B if the set of entities, i.e. posts, classified under B is a subset of the entities classified under A ($B \subseteq A \Leftrightarrow A \cap B = B$). As suggested in Mika 2007), we can also add as a criterion that the set of A must be significantly larger than the set of B using some threshold value. To identify superconcepts in the corpus of post, we can use matrix O (see 4.2.3.2). For instance, if concepts A and B appear in common in x posts ($O_{AB}=x$) and these are also the total posts that concept B describes in the whole collection ($O_{BB}=x$), while concept A appears in more posts ($O_{AB}>x$) or, better, concept A appears in significantly more posts ($O_{AB}>x+q$, where q a constant defined by criterion), then A is a superconcept of B. However, we claim that this criterion is not enough to define reasonable hierarchical associations. In case that concept B appears in only one or few posts, it may not indicate hierarchical relations but only an association. Nevertheless, it is a promising core idea that could certainly be extended and enriched with more sophisticated criteria.

- Lastly, a common attitude in tagging is the use of multiword tags. This regularly provides separate tags but it can be useful sometimes by indicating an hierarchy. For example, if a post is tagged with "social software" and "software" as well, that could probably indicate that social software is a special kind of software[69]. However, we should take into account that although this is a common attitude in English text, it could differ from language to language.

4.2.3.4 Identifying the meaning

So far, original and extracted by the text tags, and also those that were retrieved from online resources are associated with explicit relations. Each tag is associated with a few other tags and thereby it is easier to identify the indented meaning of it and map it to a URI from an available public ontology. For example, we can identify when “apple” does not refer to the fruit if it is associated with tags like “MAC” or “technology” and assign it to a correspondent semantic web entity. This can be accomplished by the use of semantic search engines like Swoogle[70] or the Watson[71] semantic engine. In (Angeletou et al. 2008) authors performed queries to the Watson semantic web gateway to connect tags with relevant Semantic Web Entities. They searched for all possible ontological entities that contain in their local name or in their label one of the identified spelling variants or synonyms for each tag. Since such queries could often result in several entities, some of which similar, they had to apply some similarity measures to reduce redundant query results. Therefore, it is obvious that this step also needs the adoption of some rules concerning how the meaning will be identifying through the related tags. The adoption of (Angeletou et al. 2008) approach to this phase is under consideration.

4.3 Modelling Ontology with Tag Ontologies

Several ontologies expressed formally with OWL and RDFS, already exist, describing different aspects of tags and their relationships with other concepts. These ontologies can help to represent all the semantic relationships that are identified and therefore to model the ontology.

The *Tag Ontology* models the tagging activity. Thereby, it can be described that an agent assigns a tag to a resource at a specific date. Furthermore, tags can be indicated as equivalent or related. Thus, a explicit description of simple tagging activities is possible (Newman et al. 2005). Some of the ontology properties that can be particularly useful in modelling relations between tags are:

- equivalentTag, for equivalent tags
- relatedTag, for tags asserted as related
- taggedBy, for tagger
- taggedOn, for time and date of tagging activity.

[69] http://calvinconaway.com/2005/01/23/folksonomies-how-we-can-improve-the-tags/

[70] http://swoogle.umbc.edu/

[71] http://watson.kmi.open.ac.uk/WatsonWUI/

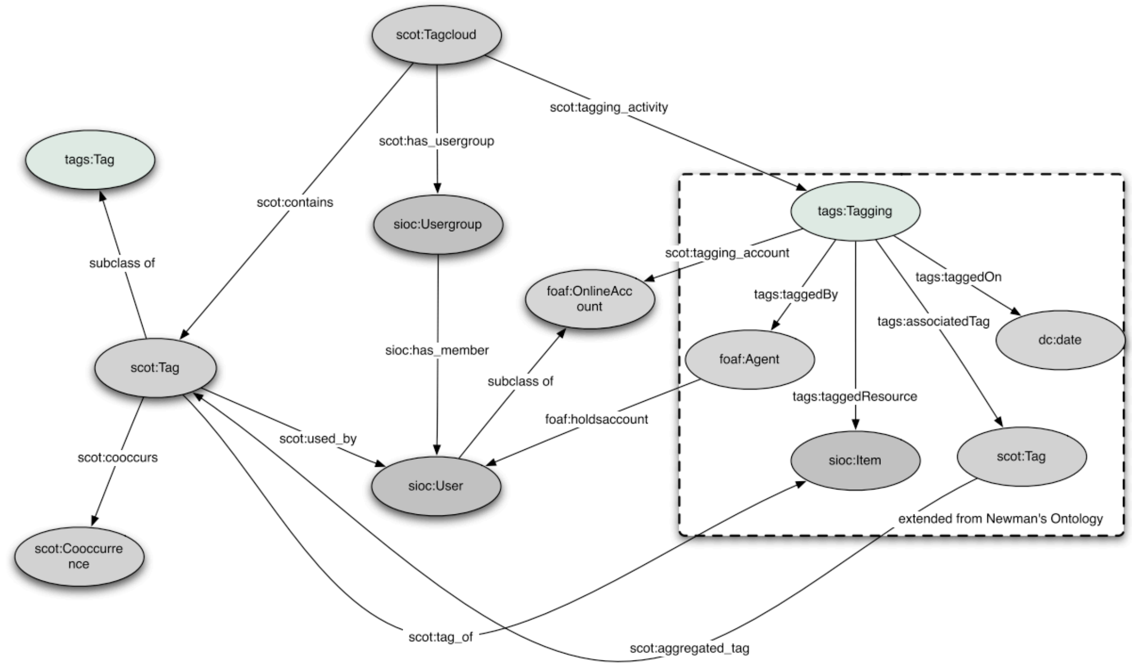


Figure 17: Relationships between SCOT and Tag Ontology (H. L. Kim et al. 2008)

The *Social Semantic Cloud of Tags (SCOT)* ontology allows the description of tag clouds. These clouds consist of tagging activities, tags, and user groups. The importance and difference of this ontology is that it provides statistical and linguistically properties for a tag. Thus, this ontology can help us to model statistical information about tags, like frequencies and co-occurrences of tags using the statistical properties, and different conventions of a concept with the use of the linguistic ones.

In particular, the statistical properties that SCOT provides:

- Frequency, representing a single tag's frequency
- cooccursWith, representing co-occurrence between two tags
- cooccurTag, represents the tag with which it co-occur
- cooccurFrequency, for the frequency of this co-occurrence.

Statistical properties they can help us modelling the statistical associations that result from the statistical analysis described in paragraph 4.2.3.1. Figure 18 illustrates an example of the co-occurrence of tags "blog" and "web2.0" modelled with the above properties.

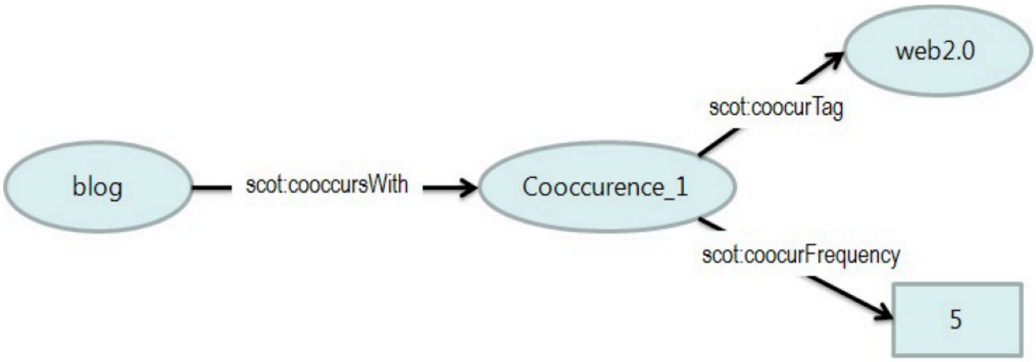


Figure 18: Co-occurrence and its frequency between blog and web2.0 (H.-L. Kim et al. 2008)

The linguistic properties allow the expression of various conventions of the term, e.g. plural, singular or acronym, and can contribute to reduce the ambiguity between terms and make semantic relations more explicit. Among these, we distinguish for our purposes the properties *plural*, *singular*, *acronym*. Furthermore, the property *synonym* will be adopted as well, to represent synonymy between tags, regardless if it is for spelling variants or completely different terms that have synonymous meaning, since no other property to represent synonymy was found in existed ontologies.

The SCOT ontology reuses and extends the Tag Ontology as can be seen in Figure 17. Additionally, the SCOT ontology is linked to SIOC, FOAF (both are described in chapter 3.3), and SKOS[72] (H.-L. Kim et al. 2008).

The last aforementioned ontology, *SKOS*, provides properties with which it is possible to describe hierarchical relations that are identified between tags. In particular, the *broader* property is used to assert that a concept is more general than another, while *narrower* is the inverse property, asserting that a concept is more specific than one other.

The *Meaning of a tag (MOAT)* ontology differentiates between a local and global meaning of a tag. A local meaning is added to each triple of a tag, tagged resource and tag creator. The global meaning consists of all the different meanings and each of these meanings are related to a set of users. MOAT reuses the Tag Ontology for the description of the local meaning. Thereby, a restriction is added that each tag must have a unique label. Furthermore, the MOAT ontology is accompanied by a client-server-architecture that is shown in Figure 19 (Passant & Laublet 2008).

[72] Simple Knowledge Organisation System: http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/

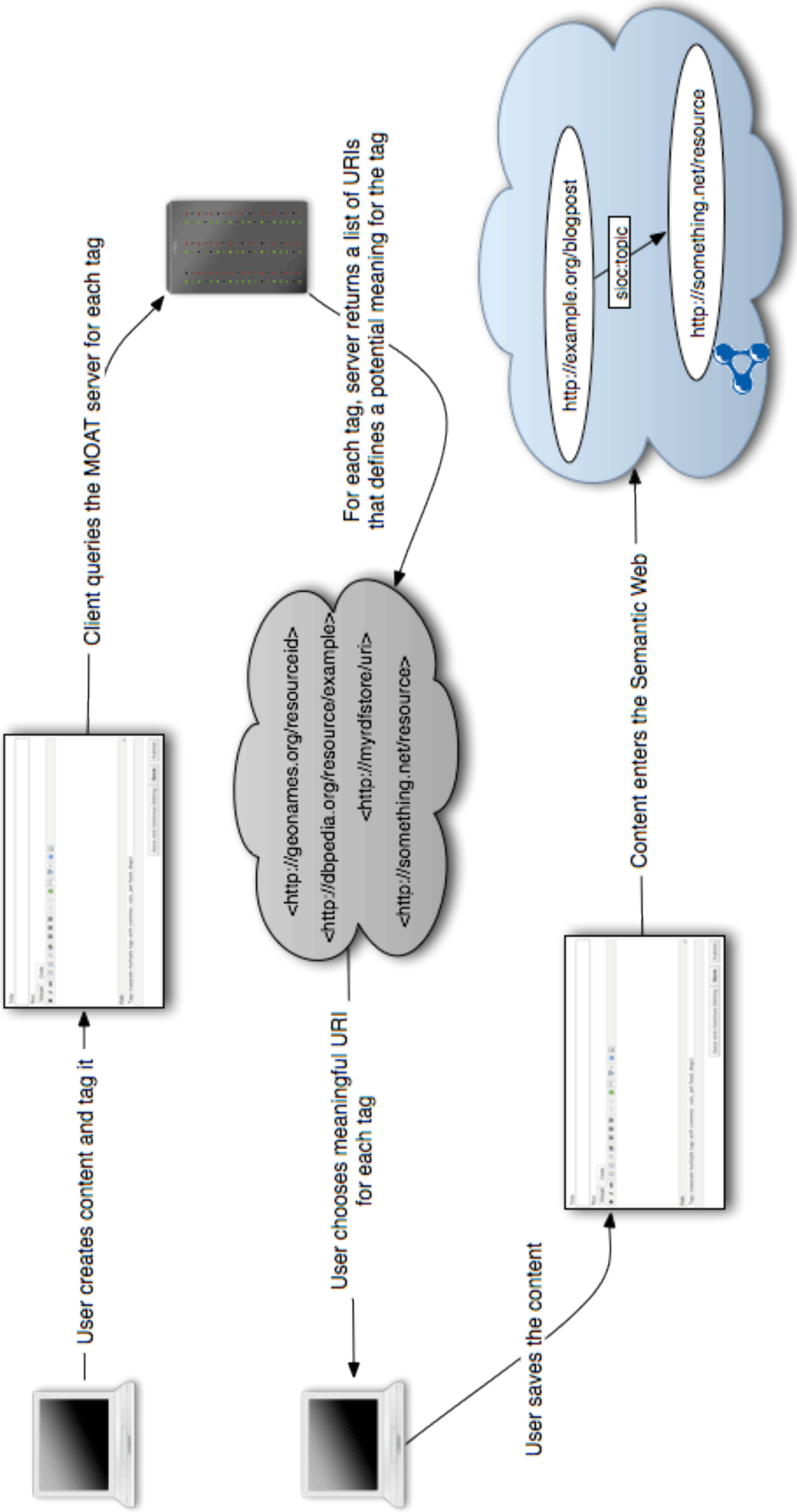

User creates content and tag it
Client queries the MOAT server for each tag
For each tag, server returns a list of URIs that defines a potential meaning for the tag
<http://geonames.org/resourceid>
<http://dbpedia.org/resource/example>
<http://myrdfstore/uri>
<http://something.net/resource>
User chooses meaningful URI for each tag
User saves the content
Content enters the Semantic Web
http://example.org/blogpost
sioc:topic
http://something.net/resource

Figure 19: Globat architecture of MOAT (Passant & Laublet 2008)

While other tag ontologies aim at the specific meaning of the tag, the *NiceTag* ontology enables a richer description of the relationship between the tag and the resource. Therefore, classes are defined to further describe the action of tagging, e.g. differentiating between manual and machine tagging, or individual and collective tagging. Additionally, a property hierarchy with "hasSign" as the main property is defined. The hasSign property assigns a tag to a resource and can be further refined as one of the following properties (Limpens et al. 2009):

- hasFactualSign
    - isAbout - the tag is the topic of the resoure
    - hasKind - the tag defines the kind of the resource (e.g. "video", "blog")
- hasSubjectiveSign
    - hasQuality - the tag expresses a quality (e.g. "good", "bad")
    - emotionalReaction - the tag expresses an emotion (e.g. "wow", a smiley)
- hasPersonalSign - the tag is intended to just make sense for the tag creator (e.g. to organise things)
- hasCommunitySign - the tag has an intended audience (e.g. Twitter hash tags are often used during specific events, "#ecis2012")
- hasNetworkingSign - the tag indicates a specific networking task
    - suggestedBy
    - suggestedTo

The *CommonTag*[73] vocabulary is intended to be used in RDFa statements enriching HTML documents with explicit semantics. However, while the vocabulary is defined as an RDF vocabulary it can be also used in other RDF serializations like RDF/XML. The CommonTag vocabulary defines the superclass Tag and its subclasses[74]:

- AuthorTag (assigned by the author of the resource),
- ReaderTag (assigned by the reader or user of the resource), and
- AutoTag (assigned by an automated system).

Additionally, four properties are defined:

- means - links a tag to a concept that the identifies the meaning of the tag,
- tagged - links the tag to the resource that is being tagged,
- taggingDate - creation date of the tag, and
- label - human readable label for the tag.

CommonTag is a very simple vocabulary. However, the differentiation between author and reader tags could be utilised in the BlogForever repository.

[73] http://commontag.org/ns

[74] http://commontag.org/Specification

4.4 Approach summary

In chapter 4.2 we described in detail all the techniques we propose to adopt in order to build the ontology and what are the purposes of adopting them. However, the techniques were described solely and since there are connections between them, it is necessary to summarize them in a way that is obvious how do they connect and support each other and in what order they take place. Therefore, the entire approach, which is a combination of them, is summarized in the following steps in Table 7.

Step	Name	Input	Output	Description
1	**Term Extraction**	Post text	A set of tags for each post	Applying a term extraction technique to get an additional, auxiliary set of tags extracted by the text of a post.
2	**Tag Pre-processing**	Dataset of posts and tags	Pre-processed data (Tagspace, Post instances)	Identify the feature space (tagspace) of problem and represent data according to it. a. Identifying all unique single-word tags, omitting stop-words. Tagspace is defined. b. Represent data in the produced feature space of tags.
3	**Grouping tags**	Dataset of posts and tags *Posts are the instances while tags are the features that describe them.*	Groups of related tags	Performing statistical analysis of archived blog posts (correlation matrices and clustering) to produce: a. A list of most similar tags of each term or a tag graph b. Clusters of similar tags
4	**Relations detection**	Group of related tags	Explicit associations between pair of tags in such groups and additional synonym terms.	Given a group of similar words from previous step, we can identify more explicit relations between tags like equivalence, synonymy and hierarchies. This is accomplished using: a. Stemming algorithms and Levenshtein distance to reveal different forms of the same word and denote them as equivalent terms (e.g. “semantic” and “semantics”). That also includes the matching of misspelled words to the correct ones. b. Exploiting WordNet, Semantic Atlas and other possible lexical

				resources to identify further relations of synonymy or hierarchy. Finally, tags that are not identified with none of the above methods are classified as related terms.
5	**Term enrichment**	Group of related tags	Additional terms as synonyms or derivative forms of existing tags.	Exploiting WordNet, Semantic Atlas and other possible lexical resources to enrich ontology with additional terms that are not contained in the current tagspace. These can be words that are found in lexical databases as either synonym to a word or very close ancestors of it in WordNet hierarchy.
6	**Meaning Identification**	Group of related tags *Note:* *The original tagset of the correspondent tag may also be necessary*	Linking of tags to URIs	After great part of tags is associated with other terms, it is easier to identify the meaning of them and map them to URIs of available public ontologies. Certainly, it is likely that many of them might not be mapped, in cases that there aren't enough available ontologies to cover each possible meaning. But, as semantic web is growing and evolved and blogs will be archived over a long time, it is highly probable that future ontologies will cover that need.

Table 7: Summary of the building ontology approach

As previously mentioned, the ontology is modelled with available public ontologies described in 4.3. Therefore, all the identified associations are modelled using existing vocabularies and that process is parallel to all aforementioned steps.

This is an iterative procedure. As the collection of blogs grows up from the updates more information is available. Consequently, this procedure should iterate in a periodic base. Relations that were not identified in previous iterations due to lack of sufficient information, may become more explicit in a future one. Furthermore, since the preservation will be a long-term activity, new terms will probably appear in Web and new relations will come up. Thus, the ontology will be updated and enriched periodically.

The frequency of iteration is under consideration. However, it seems more beneficial and meaningful to perform iteration after a certain amount of data updates in the BlogForever archive. Additionally, that must be also an option for the administrator so he can apply iteration when he estimates that it is necessary, for example when new blogs are inserted to be preserved.

Furthermore, to make things more clear, we summarise in Table 8 how the issues that arise in a folksonomy and impair content retrieval are handled in our proposed methodology. This table can also provide an overview of the importance of each technique in the entire strategy. It is already clear that the generation of groups of related tags is necessary and contributes in addressing all of the folksonomy weaknesses as an assisting step. The reason is that we can apply the other techniques to small, easy-handle groups of tags that we know that are somehow related instead of applying them to the whole corpus of tags. Furthermore, as someone can see in the table, an efficient extraction of additional terms from post texts can also be beneficial in dealing with most of these aspects.

	Levenshtein distance	Stemming Algorithm	Term extraction	Grouping related tags	WordNet & Lexical DB
Spelling variants *“blog”, “blogs”, “blogging”, “weblog”*	✗	✗	✗	✗	✗
Synonyms *“buy” & “purchase”*				✗	✗
Ambiguity *“programming” & “java”*			✗	✗	✗
Polysemy *“apple” as fruit or company*			✗	✗	✗
Key Phrases *“web tool”*				✗	
Concatenating words *“SemanticWeb”, “Semantic_Web” “Semantic-Web”*			✗	✗	
Encoded words *“nyc” & “new york city”*			✗	✗	
Typo errors *“sofware” & “software”*	✗		✗	✗	

Table 8: Blog Folksonomy weaknesses addressed in approach

In order to give a small-scale overview of the approach we performed most of the statistical and pre-processing techniques on a small dataset of 35 random blogposts about the four topics of semantic web, photography, apple benefits in health and Apple products. The last two topics were chosen in order to observe the ability in handling polysemy. The tags were parsed separately as single words, transformed in lower case and cleaned from stop-words. Then statistical analysis was performed, extracting similar tags, superconcepts and co-occurrences vector that were later used as input data for clustering in Weka[75]. This example is illustrated in Figure 20. Obviously, the example

[75] http://www.cs.waikato.ac.nz/ml/weka/

is a simplified and the data sample is insignificant and not enough to make strong connections, but even in this amount of data the proposed approach is able to distinguish groups of related data and find similar tags and broader concepts.

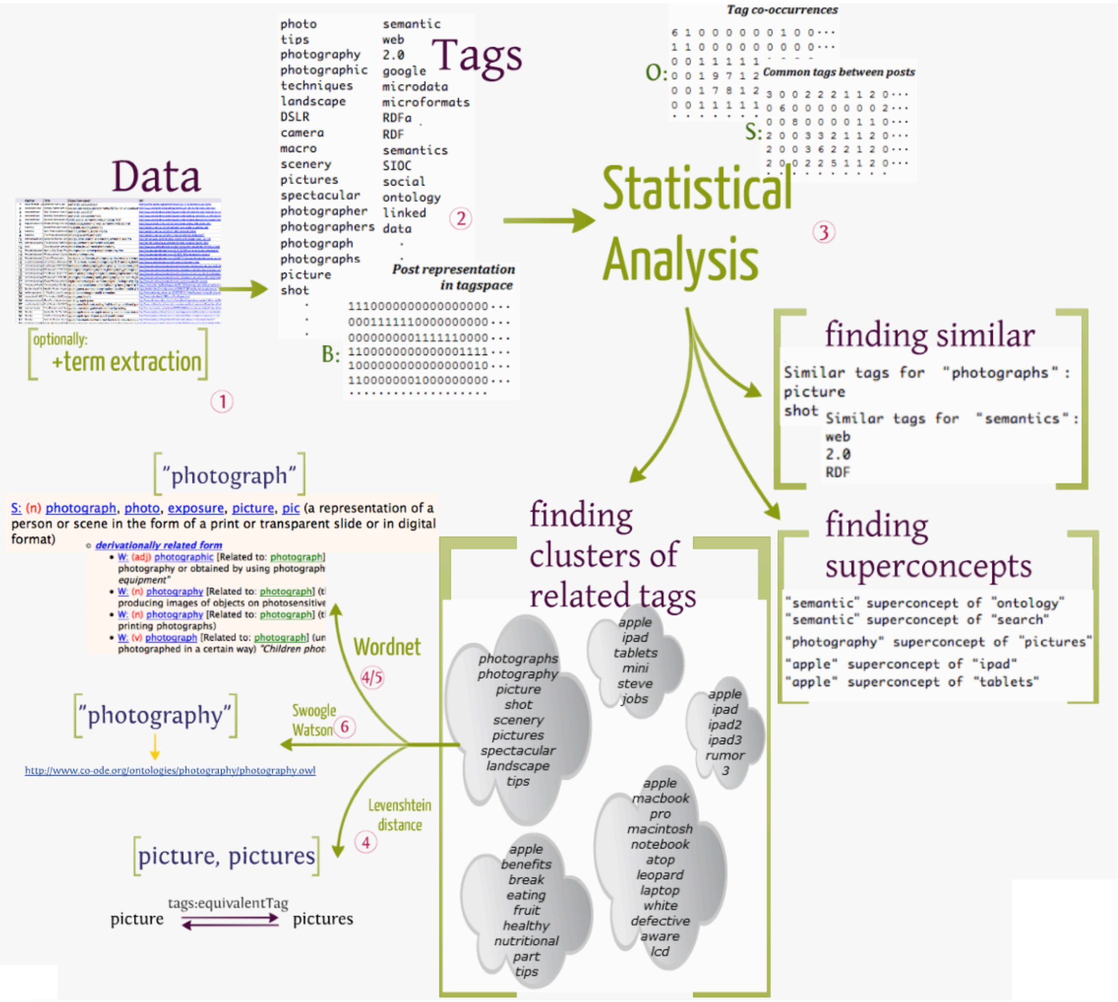

Data
optionally:
+term extraction
1
Tags
photo
tips
photography
photographic
techniques
landscape
DSLR
camera
macro
scenery
pictures
spectacular
photographer
photographers
photograph
photographs
picture
shot
semantic
web
2.0
google
microdata
microformats
RDFa
RDF
semantics
SIOC
social
ontology
linked
data
2
Post representation in tagspace
B:
1110000000000000000···
0001111110000000000···
0000000001111110000···
1100000000000001111···
1000000000000000010···
1100000001000000000···
Tag co-occurrences
O:
6 1 0 0 0 0 0 0 1 0 0···
1 1 0 0 0 0 0 0 0 0 0···
0 0 1 1 1 1 1
0 0 1 9 7 1 2
0 0 1 7 8 1 2
0 0 1 1 1 1 1
Common tags between posts
S:
3 0 0 2 2 2 1 1 2 0···
0 6 0 0 0 0 0 0 0 2···
0 0 8 0 0 0 0 1 1 0···
2 0 0 3 3 2 1 1 2 0···
2 0 0 3 6 2 2 1 2 0···
2 0 0 2 2 5 1 1 2 0···
Statistical Analysis
3
finding similar
Similar tags for "photographs":
picture
shot
Similar tags for "semantics":
web
2.0
RDF
finding superconcepts
"semantic" superconcept of "ontology"
"semantic" superconcept of "search"
"photography" superconcept of "pictures"
"apple" superconcept of "ipad"
"apple" superconcept of "tablets"
finding clusters of related tags
photographs photography picture shot scenery pictures spectacular landscape tips
apple ipad tablets mini steve jobs
apple ipad ipad2 ipad3 rumor 3
apple macbook pro macintosh notebook atop leopard laptop white defective aware lcd
apple benefits break eating fruit healthy nutritional part tips
"photograph"
S: (n) photograph, photo, exposure, picture, pic (a representation of a person or scene in the form of a print or transparent slide or in digital format)
derivationally related form
W: (adj) photographic [Related to: photograph] photography or obtained by using photograph equipment"
W: (n) photography [Related to: photograph] (t producing images of objects on photosensitive
W: (n) photography [Related to: photograph] (t printing photographs)
W: (v) photograph [Related to: photograph] (un photographed in a certain way) "Children phot
Wordnet
4/5
"photography"
http://www.co-ode.org/ontologies/photography/photography.owl
Swoogle
Watson
6
Levenshtein distance
4
picture, pictures
picture
tags:equivalentTag
pictures

Figure 20: Approach performance on 35-blogposts data sample

As we can see in the graph example, the tags are firstly pre-processed. Afterwards the statistical analysis takes place, producing the matrices of tag-occurrences and common tags in posts. The first matrix is then used to find similar tags, superconcepts and, more important, clusters of thematically related tags. As we can notice there are different clusters where “apple” is used in different meaning which means that clustering can also distinguish the context in cases of polysemy. Using these clusters later, we can identify the meaning of words, find synonyms and hierarchies, equivalences between different spelling variants of words and map the concept in available ontologies.

Consequently, it becomes clearer through the example, that the proposed technique is sufficient to provide an ontology where semantics are as explicit as possible and which can handle most of the issues that arise because of the freely chosen tags and bloggers inconsistency.

Last, another significant and potential aspect to consider is that, in the case that BlogForever users will be able to assign their own tags, a two-side beneficial interaction can take place. On the one hand, ontology can be used to provide recommended tags to the BlogForever user. On the other hand, the user can either confirm the automatically deduced relations by assigning recommended tags or enrich the ontology by proposing new terms.

4.5 Scenario Conclusion

This scenario has examined whether the exploitation of the blogs folksonomy can be beneficial in exposing semantics in order to handle problems that arise from the freely-chosen tags of blogs and, most important, to provide organizing and retrieval facilities. Thus, this scenario is a description of an approach proposed to expose semantics and the problems that must be handled in order to accomplish this.

For that purpose we propose a combination of techniques in order to build an ontology of the folksonomy of a BlogForever archive. To find semantics in this folksonomy and build the ontology these techniques utilize in brief:

- The two-mode network of tags and posts (and possibly in combination with the author-tag network) which can provide clusters of related tags,
- Available lexical databases, which can help to identify the kind of relations between tags and enrich the ontology with additional terms, and
- Term extraction techniques, which can produce additional tags from post text.

Consequently, this approach seems promising and advantageous for our purpose. Furthermore, the performance of the methods in a small sample of posts, that was formed for the needs of an example, could considered as a small-scale evaluation that strengthen our thoughts about the potential of the proposed scenario. However, further evaluations will be performed in large scale datasets to observe the strengths and any possible weaknesses of the approach in a weblog archive.

Furthermore, another significant point to be considered is that the BlogForever data model provides more concepts than authors, tags, and posts. It also provides a collection of other valuable information like metadata, time and geographical information, comments etc. Therefore, several of these entities can be used to increase the precision of identified tag relationships or to evaluate them. Especially the influence of time should be considered because the preservation activity will be probably performed over very long periods and, therefore, algorithms and measures for relationship identification should shape the changes in tag relationships appropriately.

5 Utilisation of Microformats, Microdata, and RDFa for data extraction purposes

While other scenarios in this report address how to expose archived data with explicit semantics (output side of the preservation system), the following scenario focuses on the utilisation of already available explicit semantics in the webpages that should be preserved (input side of the preservation system). For this purpose, three kinds of formats that enable a machine-readable semantic markup inside of HTML webpages are examined regarding their potential benefit to facilitate the correct extraction of blog data. Thereby, statistics about the current deployment are considered as well as assumptions about future development. The two most promising formats are further examined how they match with the BlogForever data model[76]. The chapter is organised as follows: Chapter 5.1 introduces the scenario objectives before the chapters 5.2, 5.3, and 5.4 provide a short description of the different formats. Chapter 5.5 examines available statistics about the deployment of the formats and concludes with the proposal to prioritize microformats and microdata for further considerations. Hence, chapter 5.6 evaluates how specific microformats and microdata vocabulary match to the BlogForever data model. The scenario conclusion in chapter 5.7 summarises the findings and shows how they will be further used in the BlogForever project.

5.1 Scenario description

Microformats[77], microdata[78], and RDFa[79] can be used to include metadata in XHTML pages. Thus, a webpage created for consumption by humans can be enriched with explicit semantics that can be processed by machines (Tomberg & Laanpere 2009). In the BlogForever project, the spider (or the repository) can take advantage of these explicitly described data to check or further improve the validity of the crawled data. While a data extraction on the layout specified by XHTML requires some heuristics to identify the meaning of data (e.g. to identify the author of a document), it can be obtained directly if microformats are available. Therefore, microformats, microdata, and RDFa will be examined regarding their possible utilization for data aggregation in BlogForever.

BlogForever Deliverable 2.4[80] provides already a high-level overview of several microformats, microdata, and RDFa specifications and proposes the use of an external library to extract these data from the actual blog. In this scenario, the available format specifications will be further examined regarding their potential to contribute to the data model described in Deliverable 2.2[81]. Thereby, the total amount of possible data formats should be ranked regarding their impact on general blog preservation initiatives, and, thus, it will be possible to propose a reduced set of formats that should be considered for the design of the weblog spider.

[76] See BlogForever Deliverable 2.2: Weblog Data Model, Chapter 9: Blog Data Model, pages 44-56.

[77] http://microformats.org/

[78] http://www.w3.org/TR/microdata/

[79] http://www.w3.org/TR/xhtml-rdfa-primer/

[80] See BlogForever Deliverable 2.4: Weblog spider prototype and associated methodology, Chapter: Advanced information retrieval from blog content, pages 16-22.

[81] See BlogForever Deliverable 2.2: Weblog Data Model, Chapter 9: Blog Data Model, pages 44-56.

BlogForever aims	**Preservation**		Management		Dissemination	
High level aims of ontology use	**Effective data mining**		Efficient preservation		Robust repository functions	
Internal/External	Internal perspective			**External perspective**		
Described object	Blog structure		Network of blogs		**Blog content**	
Process perspective	**Blog aggregation**		Management		Access & Distribution	
Functions of the archive	Interoperability	**Archival functions**	Policy-based management	Semantic search & navigation	Merging of social web aspects with ontologies	Ontology-based services

Table 9: Objectives addressed in the Microformats, Microdata and RDFa for data extraction scenario

The inquiry concentrates on the semantic purpose of the different formats. This means that the focus is on the meaning of the data. It will be examined how these meanings match with the concepts already specified in the BlogForever data model. An explanation how the data could be extracted technically is out of the scope in this document but it is given in BlogForever Deliverable 2.6[82].

Microformats, microdata, and RDFa aim on the extension of XHTML or HTML pages with explicit semantics. Thereby, machines can also (at least partially) understand webpages that are intended for being consumed by people. Even if the three approaches of microformats, microdata, and RDFa are similar, they have some main distinctions.

5.2 Microformats

Microformats use mainly the attributes "class" and "rel" to add explicit metadata to a (X)HTML page. Microformats can be distinguished in elementary and compound. Elementary microformats describe small or simple things and concepts, e.g. geographic coordinates can be stated with the geo microformat. Compound microformats use a nested structure to describe more complex concepts, e.g. the hCard microformat can be used to describe a person by declaring the forename, surname, nickname, address, telephone, and other attributes (Tomberg & Laanpere 2009). The specification of microformats utilizes existing standards and reuses their names, properties, types, etc. instead of reinventing completely new formats (Khare 2006).

Even if microformats are an easy solution to add explicit metadata to a webpage, they cause some problems. First, the specification process for microformat vocabularies is centralized. Thus, the quality of the vocabularies can be ensured avoiding interferences between different vocabularies. However, the adoption of new vocabularies is slowed down. A second problem can occur because

[82] See BlogForever Deliverable 2.6: Blog Data Extraction

the microformats specification uses existing HTML attributes but with a different meaning than it was intended in the original design. Thus, incompatibilities with other technologies can occur (Adida et al. 2011).

Currently, the microformats community specifies 9 stable vocabularies and 17 drafts (Anon 2012).

5.3 Microdata

The microdata specification was developed because RDFa (see section 5.4) turned out to be too complicated for many web developers. Therefore, one aim of microdata is keeping it simple for webpage authors to add explicit semantics (Ronallo 2012). The microdata specification is published as a W3C working draft. The central concept of the microdata specification is the item. Each item can have several properties that are represented by name-value pairs (Hickson 2011). Thereby, the names or types of items and their properties are not restricted. Thus, items can be defined arbitrary and the amount of possible items is unlimited. Even if this is a very flexible approach, it causes problems for automatic processing by machines (e.g. the BF weblog spider) because they could only process these meanings that they are able to understand. Therefore, the three search engines Bing[83], Google[84], and Yahoo[85] defined a vocabulary (available at Schema.org[86]) that they support and understand (Ronallo 2012). This vocabulary is likely to be adopted by webpage developers because of the support of the "big players". Therefore, the concepts described in this vocabulary are examined in the following regarding their relationships to the BlogForever data model.

5.4 RDFa

RDFa also uses attributes to enrich XHTML documents with additional metadata. However, contrary to microformats which use the existing "class" attribute, RDFa has an own set of attributes. Furthermore, RDFa uses URIs to avoid conflicts between different vocabularies and concepts. Thus, vocabularies can be defined decentralised while microformats only allow the centrally specified concepts (Adida et al. 2011). Therefore, RDFa is more flexible than microformats but more complex to handle as well.

RDFa is serialisation format of RDF even if it does not provide the full RDF capabilities (Adida et al. 2011). It is standardized from the W3C (Adida et al. 2008).

5.5 Statistics about the deployment of Microformats, Microdata, and RDFa

In general, there are only few statistics available that indicate the deployment of microformats, microdata, and RDFa.

A statistic from 2009 indicates that microformats and RDFa are not widely adopted. RDFa metadata were only available in less than 1 per cent of the examined webpages. "tag" and "hcard" were the most often detectable microformats but also just in less than 3 per cent of the pages available (Mika et al. 2009). However, a repeated analysis in 2010 shows that the portion of RDFa had been grown to 3.6% while the microformats did stay on almost the same level (Mika 2011). The study further examined the distribution of microformats and RDFa in the results which embedded at least one

[83] http://www.bing.com/

[84] http://www.google.com/

[85] http://www.yahoo.com/

[86] http://schema.org/

occurrence of embedded metadata. The resulting statistics show that microformats were available much more often (up to 38% for the tag format) than RDFa (3%) (Mika et al. 2009).

The inquiries in the BlogForever project found a usage of the XFN microformat in more than one third of the examined blogs (35,6%). In contrast to the study above, the hcard microformat was found just in 0.3% of the blogs. Microdata could be detected in only 27 instances, which means a marginal impact (Arango-docio et al. 2011; Banos et al. 2012).

Another study inquired the formats and semantics provided with OpenID identifiers. The identifiers were crawled from comments linked to OpenID identifier. Thereby, the focus was primary on blogs but other pages were also examined. Dereferencing the identifiers lead to HTML pages which were examined (beside others) regarding the existence of microformats and RDFa. Microformats appeared in 63.66% of the cases. Thereby, the tag format (42.56%) was the most popular followed by hcard (32.14%). The relevance of RDFa was ambiguous in the study. RDFa appeared in 88.01% of the cases but after they eliminated XHTML vocabulary, the occurrences were reduced to 3.22% of the cases (Tapiador & Mendo 2011).

The Web Data Commons project[87] provides statistics for huge data sets crawled in 2009/2010 and in 2012. The data sets were analysed regarding the availability of RDFa, microdata, and several microformats. How the distribution of the formats changed in both data sets indicates trends in the deployment of the different formats. Thereby, the amount of domains with triples and URLs with triples is most interesting in the context of this report. The relative portion of RDFa had the biggest increase (23.14% for domains, 26.28% for URLs) followed by microdata (6.02%, 14.22%). The portions of the microformats were stable except hcard (-18.91%, -24.03%) and XFN (-9.95%, -11.61%) which decreased. Nevertheless, the hcard microformat was also in the 2012 data set the most popular format for the domains with triple statistic even if it had the biggest decrease in relative portion (Bizer et al. 2012). However, even if the statistics by the Web Data Commons project provide a good overview about the availability of the different formats in web pages, they are less useful for the considerations in this report because they are for webpages in general and do not enable a statement about the deployment in blogs.

In summary it can be said that microformats seems to be the dominating format in blogs at the moment but the relative impact decreases. Microdata are still new but it is probable that the impact of microdata, especially with the vocabulary defined at Schema.org, increases heavily because of the support of the big search engines Bing, Google, and Yahoo. Nevertheless, the available statistics do not provide a final picture for the deployment of the formats in blogs. Therefore, the following assumptions and recommendations are made:

- Microformat could be seen as the legacy format. It has an impact, especially in the blogs that already exist. Therefore, it should be supported and further examined.
- Microdata is the most promising format (from the three formats) for the near future. Therefore, it should be supported and further examined.
- RDFa could hardly compete with microformats in the past because of the complexity. A success of RDFa in the future will be even less probable because of the emergence of microdata. Therefore, the support and examination of RDFa do not have a priority in the BlogForever project.
- The availability of microformats, microdata, and RDFa in blogs should be further examined. Next to a lack in general statistics, there is a lack in understanding regarding who uses the formats, when, and for what reasons.

[87] http://webdatacommons.org/

5.6 Microformat and Microdata Evaluation

The aforementioned reasons led us to examine and evaluate all the possible microformat specifications and microdata concepts regarding their potential relationship with the BlogForever data model. The result of this evaluation procedure is presented in Table 10.

Table 10: Microformat specification related to BlogForever data model

Name of Microformat	Purpose description	Related attributes in the data model (see D2.2)	URL of Microformat specification
	Stable Microformats		
hCalendar	Calendaring and events format	name, location, date in Event (vevent)	http://microformats.org/wiki/hcalendar
hCard	Representing people, companies, organizations, and places	Author, User Profile, Affiliation	http://microformats.org/wiki/hcard
rel-license	Indicating content licenses	copyright, ownership-rights, distribution_rights, and access_rights *in* Blog, Multimedia, Content and Text	http://microformats.org/wiki/rel-license
rel-nofollow	By adding rel="nofollow" to a hyperlink, a page indicates that the destination of that hyperlink should not be afforded any additional weight or ranking by user agents which perform link analysis upon web pages (e.g. search engines).	No direct match	http://microformats.org/wiki/rel-nofollow
rel-tag	By adding rel="tag" to a hyperlink, a page indicates that the destination of that hyperlink is an author-designated "tag" (or keyword/subject) for the current page. Note that a tag may just refer to a major portion of the current page (i.e. a blog post). The linked page should exist, and it is the linked page, rather than the link text, that defines the tag.	Tag	http://microformats.org/wiki/rel-tag

XFN (XHTML Friends Network)	A simple way to represent human relationships using hyperlinks.	Author	http://gmpg.org/xfn/
XMDP (XHTML MetaData Profiles)	A simple XHTML-based format for defining HTML meta data profiles easy to read and write by both humans and machines. The markup is a profile of XHTML. XMDP is more a meta format because it describes the way to define profiles by the use of other microformats, e.g. rel-license.	No direct match	http://gmpg.org/xmdp/
	Drafts of Microformats		
adr	Marking up address information. It is also a property of hCard.	location_city, location_country *in* Blog	http://microformats.org/wiki/adr
geo	Marking up WGS84 geographic coordinates (latitude; longitude)	geo_longitude, geo_latitude *in* Entry, Comment	http://microformats.org/wiki/geo
hAtom	hAtom is a microformat for content that can be syndicated, primarily but not exclusively weblog postings. hAtom is based on a subset of the Atom syndication format.	title, URI *in* Entry, Post. Also: Author, Content, Tag	http://microformats.org/wiki/hatom
hAudio	Embedding information about audio recordings	Multimedia, Audio	http://microformats.org/wiki/haudio
hMedia	Publishing Images Video and Audio.	title, creator *in* Multimedia	http://microformats.org/wiki/hmedia
hReview	Embedding reviews (of products, services, businesses, events, etc.)	URI, copyright, ownership_rights, distribution_rights, access_rights *in* Entry Also: Author, Tag	http://microformats.org/wiki/hreview
robots exclusion	The Robots META tag is used to provide page-specific direction	No direct match. However, the	http://microformats.org/wiki/robots-

	for web crawlers. While being useful in many cases, its page-specific nature means it cannot be used to restrict crawlers from indexing only certain sections of a document. Several attempts have been made to create more granular solutions through various methods but have perceived shortcomings that limit their use; the Robot Exclusion Profile defines a microformat that can be applied to any element or set of elements in a page.	microformat can be utilised by the weblog spider to identify parts that should not be crawled.	exclusion

The examination of the microdata concepts with respect to the data model revealed a subset of the ones that were found to be the most suitable. The correspondence between microdata properties that are related to data model attributes are presented in Table 11. It is noteworthy to mention that there is a similarity between some concepts in Schema.org vocabulary and the BlogForever data model. However, there are no absolute matches between a class and a data model entity, not even between those who refer to the same objects (e.g. Blog), since they present considerable differences.

Table 11: Microdata specification related to BlogForever data model

Microdata Class	Schema.org url	Microdata Property	Description	Related attributes in the data model (see D2.2)
Thing	http://schema.org/Thing	name	Blog, Entry, Post, Page, Link	title
CreativeWork	http://schema.org/CreativeWork	url	Blog, Entry, Post, Page, Multimedia	URI
		keywords	Expression_Meta	keyword_set
		keywords	Tag	tag
Article	http://schema.org/Article	dateCreated	Entry	date_created
		dateModified	Entry	date_modified
		articleBody	Content	full_content
Blog	http://schema.org/Blog	inLanguage	Blog	language
		copyrightHolder	Blog	copyright
		provider	Blog	distribution_rights

		dateModified	Blog	last_activity_date
BlogPosting	http://schema.org/BlogPosting	dateCreated	Entry	date_created
		dateModified	Entry	date_modified
MediaObject	http://schema.org/MediaObject	encoding	Content	encoding
		description	Multimedia	description
		title	Multimedia	title
		duration	Audio,Video	duration
		inLanguage	Text	language
		creator	Multimedia	creator
		provider	Multimedia	distribution_rights
		copyrightHolder	Multimedia	copyright
		encodingFormat	Audio, Video, Image, Text, Document	format
AudioObject	http://schema.org/AudioObject	bitrate	Audio	bit_rate
ImageObject	http://schema.org/ImageObject	width	Image	width
		height	Image	height
		thumbnail	Image	thumbnail_uri
VideoObject	http://schema.org/VideoObject	videoFrameSize	Video	resolution
		thumbnail	Video	thumbnail_uri
Event	http://schema.org/Event	name	Event	name
		location	Event	location
		startDate	Event	date
UserComments	http://schema.org/UserComments	replyToUrl	Comment	is_child_of_post, is_child_of_comment

		name	Comment	subject
PostalAddress	http://schema.org/PostalAddress	addressCountry	Blog	location_country
		addressLocality	Blog	location_city
GeoCoordinates	http://schema.org/GeoCoordinates	latitude	Entry, Comment, Multimedia	geo_latitude
		longitude	Entry, Comment, Multimedia	geo_longitude
Person	http://schema.org/Person	name, familyName, givenName	Author	name_displayed/ name
		email	Author	email_displayed
		url	Author, User_Profile	profile_uri

5.7 Scenario conclusion

The scenario has examined how the utilisation of semantic markup in HTML pages can facilitate the precision of crawled data in the BlogForever project. It can be stated as a result that

- Microformats are already deployed in a significant amount of webpages while microdata will probably be better adopted in the near future, and
- The vocabularies of both formats can be matched partially to the BlogForever data model.

Therefore, the processing of microformats and microdata is promising to facilitate the data extraction in the crawling process. However, it has been only evaluated if the processing would be beneficial from a semantic perspective. Considerations regarding existing technologies or tools for processing microformats and microdata have not been made but can be found in the BlogForever Deliverable D2.6[88]. Therefore, the findings from both reports will be combined and used to further improve the BlogForever weblog spider design and implementation.

[88] BlogForever Deliverable D2.6: Blog Data Extraction, Chapter 6.6: Extraction of Structured Data.

6 Conclusion

In this report, examinations about how to benefit from the application of ontologies in the BlogForever project have been performed. It has started with broadening the perspective by creating a framework to describe possible objectives of ontology application (see chapter 2). This framework has been used to classify the three specific application scenarios described in this report but it can also be used to identify further scenarios.

The first scenario in chapter 3 has considered the provision of data stored in BlogForever repositories and their relationships as openly available data with an explicit semantic that is machine readable. Therefore, it has been proposed to adopt the concept of Linked Open Data (LOD). Thereby, several purposes will be served. First, different BlogForever repositories would be able to interoperate easily on a data level. Second, they can be integrated with other digital libraries like databases for scientific publications to enable search queries and navigation that are not limited to the authors and publications in the blog repositories. Third, the data can be linked to other repositories in order to connect terms to public available definitions and descriptions of these terms. For example, the topic of a blog or blog post can be linked to the definition in Wikipedia or other repositories. Fourth, new relationships can be easily created and expressed (e.g. with SPARQL) based on the existing data and relationships. Fifth, the provision of the data in an openly available standardized format and with a unique identifier for each object facilitates the development of third party applications.

The exposure of data from BlogForever repositories has been further described and examined. Four vocabularies (Dublin Core, FOAF, SIOC, and PREMIS) has been identified that should be used to expose an important subset of the BlogForever weblog data model. The data model has been limited to a purposive subset because the exposed concepts should be widely applicable across as many blogs as possible, and the associated properties should sufficiently overlap with the properties of material held within other repositories that do not necessarily specialise in weblogs. However, the vocabulary used to expose the weblog data can be easily extended in further developments. The examination in this report has shown that all the considered weblog data could be exposed with existing vocabularies. Thereby, the vocabularies SIOC, FOAF, and DC cover the context of blogs and should be accompanied by the PREMIS vocabulary that provides specific concepts in the preservation context. Thus, the creation of a new vocabulary cannot be recommended at this point. However, the consideration of existing vocabularies has led to suggestions for adaptation of the data model.

Given that the BlogForever repository will be based on an SQL database, an examination of two tools for the automatic extraction and provision of LOD from SQL databases has supplemented the considerations. Triplify is the more simple solution while D2R server provides more functionality. Especially, the missing feature of SPARQL in Triplify (it is planned but not yet implemented) leads to a recommendation for D2R server.

The second scenario in chapter 4 has considered the integration of folksonomy and ontology. Therefore, the concept of tagging has been described and it must be differentiated between a semantic enrichment of crawled tags after the crawling process and the binding of tags created by repository users to an explicit semantic at the time of the tagging activity. While the latter can be addressed with existing solutions like the MOAT architecture, the first is a field of on-going research from different directions. One promising attempt is the use of social network analysis to deduce explicit relations in the folksonomy, generating eventually an ontology. To accomplish that, a combined exploitation of weblog post text, statistical analysis in the two-mode network of posts and tags and available lexical databases is proposed. The general approach has been outlined and should be further considered in the BlogForever project because it promises a significant improvement in repository content organisation and retrieval as well as a contribution to the scientific knowledge base.

For the explicit description of tag semantic, existing vocabularies were examined. Thereby, the SCOT ontology is the most promising because it is able to not only model the relationships between tags, tagged resource, and tag creator but can further handle different meanings in different tag clouds, e.g. in different blogs. Furthermore, SCOT provides statistical properties that describe tags co-occurrences and frequencies and which can be useful in the identification of tag's context. Therefore, the SCOT ontology should be chosen if tag semantics should be expressed explicitly in the BlogForever repository but properties from other vocabularies like Tag Ontology can also be included.

The third scenario in chapter 5 has considered the utilisation of explicit semantics in HTML pages for the data extraction by the weblog spider. The three initiatives Microformats, Microdata, and RDFa have been examined regarding their adoption in webpages and specifically in blogs. This has led to the conclusion that Microformats is the most adopted format at the time while Microdata will probably be the most adopted in the future. Therefore, both should be prioritized against RDFa. Existing Microformats and Microdata vocabularies has been further examined and described regarding their contribution to the weblog data model. Thereby, relevant formats can be chosen more easily in the weblog spider design and implementation.

Overall, the application of ontologies is promising for various purposes in the BlogForever project. This report has given a deeper insight into some of the possibilities. However, the application of ontologies is a complex process that involves a technical and a community aspect. While technical challenges can be solved with existing technologies, the community has to perform a process that enables an agreement about semantics and their use. Therefore, the application of ontologies should be kept focussed in the project and the activities, to the clarify semantics in the blog context should be emphasised.

7 References

Adida, B. et al., 2008. RDFa in XHTML: Syntax and Processing. *W3C Recommendation 14 October 2008.*

Adida, B., Birbeck, M. & Herman, I., 2011. Semantic Annotation and Retrieval: Web of Hypertext – RDFa and Microformats. In J. Domingue, Dieter Fensel, & J. A. Hendler, eds. *Handbook of Semantic Web Technologies*. Berlin, Heidelberg: Springer Verlag, pp. 157–190.

Agichtein, E. et al., 2008. Finding high-quality content in social media. In *Proceedings of the international conference on Web search and web data mining - WSDM '08*. New York, New York, USA: ACM Press, pp. 183–194.

Angeletou, S., Sabou, M. & Motta, Enrico, 2008. Semantically Enriching Folksonomies with FLOR. In *1st International Workshop on Collective Semantics Collective Intelligence the Semantic Web CISWeb 2008 at The 5th Annual European Semantic Web Conference ESWC 2008*. 5th Annual European Semantic Web Conference (ESWC 2008). Available at: http://oro.open.ac.uk/23497/.

Anon, 2012. Microformats.

Arango-docio, S. et al., 2011. *BlogForever: D2.1 Survey Implementation Report*, London.

Auer, S. et al., 2009. Triplify: light-weight linked data publication from relational databases. *Proceedings of the 18th international conference on World Wide Web*, pp.621–630. Available at: http://dl.acm.org/citation.cfm?id=1526793 [Accessed May 25, 2012].

Banos, V. et al., 2012. Technological foundations of the current Blogosphere. In *International Conference on Web Intelligence, Mining and Semantics (WIMS) 2012*. Craiova, Romania.

Barrasa, J., Corcho, Ó. & Gómez-pérez, A., 2004. R 2 O , an Extensible and Semantically Based Database-to-ontology Mapping Language. *In Proceedings of the 2nd Workshop on Semantic Web and Databases (SWDB2004)*, (August), pp.1069–1070.

Begelman, G., Keller, P. & Smadja, F., 2006. Automated Tag Clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop at WWW2006 Edinburgh Scotland*. Citeseer. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.5736&amp;rep=rep1&amp;type=pdf.

Berners-Lee, T., 2009. Linked Data - Design Issues.

Bizer, C., 2006. D2R Server - Publishing Relational Databases on the Semantic Web. *5th International Semantic Web.*

Bizer, C. et al., 2012. Web Data Commons.

Bizer, C. & Seaborne, A., 2004. D2RQ-treating non-RDF databases as virtual RDF graphs. *Proceedings of the 3rd International Semantic*. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:D2RQ+–+Treating+Non-RDF+Databases+as+Virtual+RDF+Graphs#0 [Accessed May 25, 2012].

Bojars, U. & Breslin, J.G., 2010. SIOC Core Ontology Specification. *25 March 2010.*

Brickley, D. & Guha, R.V., 2004. RDF Vocabulary Description Language 1.0: RDF Schema. *W3C Recommendation 10 February 2004*.

Cattuto, C. et al., 2008. Semantic Grounding of Tag Relatedness in Social Bookmarking Systems A. Sheth et al., eds. *The Semantic WebISWC 2008*, 5318, pp.615–631. Available at: http://www.springerlink.com/index/9044260283881V78.pdf.

Cerbah, F., 2008. Learning Highly Structured Semantic Repositories from Relational Databases: The RDBToOnto Tool. In S. Bechhofer et al., eds. *THE SEMANTIC WEB: RESEARCH AND APPLICATIONS*. Springer Berlin / Heidelberg, pp. 777–781. Available at: http://dx.doi.org/10.1007/978-3-540-68234-9_57.

Damme, C.V., Hepp, M. & Siorpaes, K., 2007. FolksOntology: An Integrated Approach for Turning Folksonomies into Ontologies S. Decker et al., eds. *Social Networks*, 2, pp.57–70. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.67.6592&amp;rep=rep1&amp;type=pdf.

Guarino, N., Oberle, D. & Staab, S., 2009. What is an Ontology? In S. Staab & R. Studer, eds. *Handbook on Ontologies*. Dordrecht: Springer, pp. 1–17.

Health, T. & Bizer, C., 2011. *Linked Data: Evolving the Web into a Global Data Space* 1st ed., Morgan & Claypool.

Helic, D. et al., 2012. Navigational efficiency of broad vs. narrow folksonomies. In *Proceedings of the 23rd ACM conference on Hypertext and social media HT 12*. ACM Press, p. 63. Available at: http://dl.acm.org/citation.cfm?doid=2309996.2310008.

Hickson, I., 2011. HTML Microdata. *HTML Microdata W3C Working Draft 25 May 2011*.

Hulth, A., 2003. Improved automatic keyword extraction given more linguistic knowledge M. Collins & M. Steedman, eds. *Proceedings of the 2003 conference on Empirical methods in natural language processing*, 10(2000), pp.216–223. Available at: http://portal.acm.org/citation.cfm?doid=1119355.1119383.

Hyvönen, E., 2009. Semantic portals for cultural heritage. In S. Staab & R. Studer, eds. *Handbook on Ontologies*. Dordrecht: Springer, pp. 757–778.

Kaur, J. & Gupta, V., 2010. Effective Approaches For Extraction Of Keywords. *Journal of Computer Science*, 7(6), pp.144–148. Available at: http://ijcsi.org/papers/7-6-144-148.pdf.

Khare, R., 2006. The Next (Small) Thing on the Semantic Web? *IEEE Internet Computing*, 10(1), pp.68–75.

Kim, H.-L. et al., 2008. Social Semantic Cloud of Tag: Semantic Model for Social Tagging. In N. Nguyen et al., eds. *KES-AMSTA 2008, LNAI 4953*. Springer Berlin / Heidelberg, pp. 83–92.

Kim, H.L. et al., 2008. Review and Alignment of Tag Ontologies for Semantically-Linked Data in Collaborative Tagging Spaces. In *2008 IEEE International Conference on Semantic Computing*. IEEE, pp. 315–322.

Kurz, T. et al., 2012. Adding wings to red bull media: search and display semantically enhanced video fragments. In *Proceedings of the 21st international conference companion on World Wide Web*. New York, NY, USA: ACM, pp. 373–376.

Lenat, D.B., 1995. A large-scale invesment in knowledge infrastructure. *Communications of the ACM*, 33(11), pp.33–38.

Limpens, F. et al., 2009. NiceTag Ontology : tags as named graphs. In *International Workshop in Social Networks Interoperability (2009).*

Liu, Feifan et al., 2009. Unsupervised approaches for automatic keyword extraction using meeting transcripts. *Proceedings of Human Language Technologies The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on NAACL 09*, (June), pp.620–628. Available at: http://portal.acm.org/citation.cfm?doid=1620754.1620845.

Manola, F. & Miller, E., 2004. RDF Primer. *W3C Recommendation 10 February 2004.*

McHuh, A.J. & Lalmas, M., 2010. Rethinking Preservation Validation with the Preserved Object and Repository Risks Ontology (PORRO). In *ACM/IEEE Joint Conference on Digital Libraries (JCDL) 2010*. Surfers Paradise, Australia.

Mika, P., 2011. Microformats and RDFa deployment across the Web. *Tripletalk.*

Mika, P., 2007. Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1), pp.5–15.

Mika, P., Meij, E. & Zaragoza, H., 2009. Investigating the Semantic Gap through Query Log Analysis. In Abraham Bernstein et al., eds. *8th International Semantic Web Conference, ISWC 2009*. Chantilly, VA, USA: Springer, pp. 441–455.

Navigli, R., 2009. Word Sense Disambiguation: A Survey. *ACM Computing Surveys*, 41(2).

Navigli, Roberto & Velardi, P., 2003. An analysis of ontology-based query expansion strategies. In *Proceedings of the International Workshop & Tutorial on Adaptive Text Extraction and Mining held in conjunction with the 14th European Conference on Machine Learning and the 7th European Conference on Principles and Practice of Knowledge Discovery in Data*. Cavtat–Dubrovnik, Croatia, pp. 42–49.

Newman, R., Ayers, D. & Russell, S., 2005. Tag ontology.

Passant, A. et al., 2009. A URI is Worth a thousand tags: from tagging to Linked Data with MOAT. *International Journal on Semantic Web and Information Systems*, 5(3), pp.71–94.

Passant, A. & Laublet, P., 2008. 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 (LDOW)*. Beijing, China.

Pérez, J., Arenas, M. & Gutierrez, C., 2009. Semantics and complexity of SPARQL. *ACM Transactions on Database Systems*, 34(3), pp.1–45. Available at: http://portal.acm.org/citation.cfm?doid=1567274.1567278 [Accessed March 4, 2012].

Ronallo, J., 2012. HTML5 Microdata and Schema.org. *code{4}lib*, (16).

Smith, B., 2004. Beyond Concepts: Ontology as Reality Representation. In A. Varzi & L. Vieu, eds. *International Conference on Formal Ontology and Information Systems (FOIS) 2004*. Turin.

Smith, B. & Welty, Christopher, 2001. Ontology: Towards a New Synthesis. In *FOIS '01 Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*. New York, NY, USA: ACM, p. iii–ix.

Smith, M.K., Welty, Chris & McGuinness, D.L., 2004. OWL Web Ontology Language Guide. *W3C Recommendation 10 February 2004*.

Sowa, J.F., 2000. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Pacific Grove, CA.: Brooks Cole Publishing.

Specia, L. & Motta, E, 2007. Integrating folksonomies with the semantic web. In E. Franconi, M. Kifer, & W. May, eds. *The Semantic Web Research and Applications*. Springer, pp. 624–639. Available at: http://oro.open.ac.uk/15676/.

Studer, R., Benjamins, R. & Fensel, D., 1998. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1-2), pp.161–198.

Tapiador, A. & Mendo, A., 2011. A survey on OpenID identifiers. In *7th International Conference on Next Generation Web Services Practices (NWeSP) 2011*. Salamanca, Spain, pp. 357–362.

Tomberg, V. & Laanpere, M., 2009. RDFa versus Microformats: Exploring the Potential for Semantic Interoperability of Mash-up Personal Learning Environments. In F. Wild et al., eds. *Mash-Up Personal Learning Environments (MUPPLE'09)*. Nice, France, pp. 102–109.

Trant, J., 2009. Studying Social Tagging and Folksonomy: A Review and Framework. *Journal Of Digital Information*, 10(1), pp.1–42.

W3C OWL Working Group, 2009. OWL 2 Web Ontology Language Document Overview. *W3C Recommendation 27 October 2009*.

Warren, D.H.D. & Pereira, F.C.N., 1982. An efficient easily adaptable system for interpreting natural language queries. *Computational Linguistics*, 8(3-4), pp.110–122.

Zemmouchi, L. & Ghomari, A.R., 2009. Reference Ontology. In *Fifth International Conference on Signal Image Technology and Internet Based Systems*. Marrakesh, pp. 485–491.

Zhuhadar, L. et al., 2010. Multi-Language Ontology-based Search Engine. In *Third International Conferences on Advances in Computer-Human Interactions (ACHI '10)*. Washington, DC, USA: IEEE Computer Society, pp. 13–18.