SEVENTH FRAMEWORK PROGRAMME
FP7-ICT-2009-6

BlogForever
Grant agreement no.: 269963

D2.2 Report: Weblog Data Model

Editor:	K. Stepanyan
Revision:	First Version
Dissemination Level:	Public
Author(s):	K. Stepanyan, M. Joy, A. Cristea, Y. Kim, E. Pinsent, S. Kopidaki
Due date of deliverable:	31/10/2011
Actual submission date:	31/10/2011
Start date of project:	01 March 2011
Duration:	30 months
Lead Beneficiary name:	University of Warwick (UW)

Abstract: This report outlines the development of a data model to support the preservation, management and dissemination of blogs. It outlines the literature and relevant approaches to data modelling and proceeds to describe the inquiries that informed the development of the proposed data model.

The report identifies the data structures considered necessary for preserving blogs by revisiting the earlier inquiry summarised in the BlogForever Report, Deliverable 2.1 [1]. The report includes an inquiry into [a] the existing conceptual models of blogs, [b] the data models of Open Source blogging systems, and [c] data types identified from an empirical study of web feeds.

The results, the report progresses to propose a data model intended to enable preservation of blogs and their individual components. Following internal consultation exercises from blog service providers and preservation experts, the proposed data model may require further refinement in accordance with the anticipated development of preservation policies (WP3), data extraction methodologies (WP2) and user requirements for platform BlogForever specification (WP4).

Finally, the report positions and discusses the proposed data model alongside the Invenio software suite by highlighting the anticipated data flow and suggesting directions for their integration.

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 Gelistrirme 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

Revision History

Version	*Date*	*Modification reason*	*Modified by*
0.1	22.06.2011	Report Outline	K. Stepanyan
0.2	05.07.2011	Draft	K. Stepanyan
0.3	15.07.2011	Extension: evaluation of RSS feeds	K. Stepanyan
0.4	15.08.2011	Extension: blog data models and general progress	K. Stepanyan
0.5	02.09.2011	General progress: insertion of the model	K. Stepanyan
0.6	28.09.2011	Draft for an internal review	K. Stepanyan
0.7	05.10.2011	Review, corrections or feedback	V. Banos, Y. Kim, M. Joy, K.Stepanyan, H. Kalb
0.8	10.10.2011	Consultation exercise on blog preservation	E. Pinsent
0.9	15.10.2011	Second Draft of the Report	K. Stepanyan
1.0	21.10.2011	Pre-final draft: addressing comments and feedback	K. Stepanyan, V. Banos, Y. Kim, N. Kasioumis, J. Llopis, H. Kalb
1.1	21.10.2011	Extension: internal consultation from Phaistos	S. Kopidaki
1.2	22.10.2011	General review and corrections	K. Stepanyan
1.3	25.10.2011	Styling and clarifications	N. Kasioumis, M. Joy, K.Stepanyan
1.4	25.10.2011	Extension and Clarification of 2.1.2	Y. Kim
1.5	28.10.2011	Review and clarification	M. Joy

Table of Contents

List of Figures

List of Tables

List of Terms and Abbreviations

Blog (Weblog): a website with commentary typically presented in a reverse chronological order.

Blogosphere: a collective noun to denote the dynamic network of blogs that exist on the Web.

Data Model: a way of perceiving, organising and describing data
- *Entity:* An abstract representation of an object of interest to preservation.
- *Attribute:* Inherent or intrinsic characteristic of an entity.
- *Relationship:* Association between entities.

Entity: an abstraction from a specified domain about which data can be stored.

Web Feed: XML-based data format for distributing and syndicating web content.

Application Program Interface (API): a set of rules, protocols and methods for interacting with software and reducing software development workload.

Executive Summary

This report summarises a data modelling exercise conducted to support the development of a blog repository solutions within the boundaries of the BlogForever project. Prior to proposing the data model, the report outlines the requirements of the project and the chosen data modelling approach.

Referring to the relevant literature and the requirements of the project, this report elicits a working definition of a blog. Using the working definition to set a terrain for data modelling, the report proceeds to outline a set of exercises informing the modelling process. The data modelling is supported by the following.

1. A review of existing conceptual models for blogs (Chapter 3).
2. An insight into the database structure of open source blog systems (Chapter 5).
3. A retrospective view on an earlier user survey conducted online to identify important aspects and types of blog data to be preserved (Section 6.1).
4. A retrospective view on the technologies and standards used within the Blogosphere (Section 6.2).
5. Suggestions derived from an earlier conducted inquiry into the recent developments and prospects for analysing networks and dynamics of blogs (Section 6.3).
6. An inquiry into blog structure based on evaluation of 2,695 blog feeds (Chapter 7).
7. An insight into blog APIs (Chapter 8).

Combining and corroborating the results from the aforementioned exercises, the report proceeds to define and propose a blog data model for preserving blogs. The proposed data model is presented in two formats: conceptual and logical. Each of the formats serves a distinct purpose. The conceptual model is presented for communicating the ideas at a higher level. The conceptual model highlights the proposed generic, core data model for blogs that is further extended by introducing components that enrich the generic blog data. The conceptual model omits some of the details for the purposes of readability. The logical model, on the other hand, provides greater details by listing identified entities, relationships and attributes.

The report extends to include internal consultation exercises from:

a) project partners specialising in digital preservation, and
b) blog service provider Phaistos Networks.

As a result, some clarifications and modifications have been introduced into the proposed model. However, further refinements may be necessary to comply with the anticipated development of preservation policies (WP3), data extraction methodologies (WP2) and user requirements for the BlogForever platform specification (WP4).

Finally, the report positions the proposed data model in relation to the processes and data flow anticipated from the solutions being developed as part of the BlogForever project. It discusses possible ways forward for integrating the data model into the existing structures underpinning the Invenio[1] software suite – a digital library system to support blog preservation.

[1] http://invenio-software.org/

1 Introduction

The Web plays an increasingly important role in our society. It constitutes a social activity space that is integral for information exchange and knowledge creation. Web content and user generated content, in particular, embody a valuable source of information. The ephemeral nature of these resources justifies the strategies and actions taken for ensuring their long-term accessibility and preservation. Preserving these resources can enable people, and generations to come, to gain insight into the social, economic, political and cultural lives of people reflected within this interrelated communication space. The abundance of exchanged information from a variety of sources can help them tap into the wisdom of crowds and harness it in a range of different ways.

The BlogForever project aims to develop solutions for preserving, managing and disseminating blogs. This document refers to the goals and objectives of the project, as well as the challenges associated with this preservation task. It introduces a data model, designed to address the goals and objectives of the project and ensure sufficient operation of the anticipated blog preservation solutions for the Blogosphere. Prior to advancing any further, this section outlines the objectives of this report and the chosen approaches. It starts by providing background information into the rationale and main aims of the BlogForever project.

1.1 Background Information

Blogs are dynamic and versatile web spaces. As one of the Web 2.0 technologies, blogs are user-centred web tools that promote social connectedness, sharing, content creation and collaboration. These primary characteristics of blogs constitute a blogging platform that sets them apart from the more traditional Web of inert web pages. Furthermore, an inherent characteristic of blogs is their immense diversity. Blogs vary in their choice of subject, writing style, purpose, media use, platform and presentation. They can represent individuals, organisations or companies and connect to different audiences. Blogs can assemble into specialised communities and interweave into sparse networks. The inter-related network of blogs is commonly referred to as the Blogosphere.

The Blogosphere represents a relatively new and dynamic medium that has experienced rapid growth in the recent years. The continued growth and evolution is one of the primary challenges when embarking on the task to preserve blogs. The scale of the Blogosphere exceeds a hundred million blogs. Within the last 10 years over 133 million blogs have been indexed by Technorati[2]. Thousands of new blogs are still being created every day with millions of blog entries posted daily. The medium has acquired a large readership with an estimated 77% of active Web users [2]. The continued growth of the Blogosphere and its diversity attract the attention of various stakeholders. Capturing and preserving blogs that exist within the volatile environment of the Web will provide some valuable information to the interested stakeholders and to generations to come. The dynamic nature of blogs, their diversity and rapid growth bring challenges for developing suitable solutions for their preservation. This report highlights the requirements and discusses challenges associated with blog preservation prior to proposing the data model.

1.2 BlogForever Background

The main aim of the BlogForever project is to develop robust digital preservation, management and dissemination facilities for blogs. Addressing project aims requires development of new, considerably improved solutions that capture the dynamic and continuously evolving nature of blogs, their networks and social structure. The developed solutions should enable tracing exchange

[2] http://technorati.com

of concepts within blogs and ideas that they foster. The solutions should also ensure authenticity, integrity, completeness, usability and long term accessibility of blogs as a valuable cultural, social and intellectual resource.

The outcomes of the project are expected to benefit a number of stakeholders, in particular, libraries, information centres, museums, universities, research institutes, businesses, as well as blog authors and readers in general. The solutions, however, should not be restricted to institutional use only. Achieving the aims requires an investigation into the structure of blogs and their semantics. It requires understanding of blog networks and their dynamics. The investigation should inform the design and development of the BlogForever solutions and the development of the data model in particular.

1.3 Main Goals of Work Package Two and their Inter-Relation

Work Package Two (WP2) is one of the six work packages of the BlogForever project that has three primary and sequential tasks[3]:

Task 2.1: Weblogs Survey. Conduct user survey into: blogging practices, preservation of blogs, technologies and structural patterns (Deliverable D2.1).

Task 2.2: Weblog Semantics. Conduct semantic analysis of blogs to inform the development of the data model and the ontological representation of the domain (Deliverables D2.2 and D2.3).

Task 2.3: Weblog Data Extraction. Review existing web data extraction methodologies and tools, assess their potential use for blog data extraction, review spam filtering methods and develop a prototype application (Deliverables D2.4, D2.5 and D2.6).

The completion of the first task (2.1) plays an important role for informing the development of the data model. This task has already been completed and submitted as Deliverable D2.1[4]. The outcomes reported in the submitted deliverable are used here to inform the development of the data model. Subsequently, the data model and ontological representation of blogs are anticipated to serve as a framework for further progress towards the data extraction task.

1.4 Objectives and Rationale of Data Modelling

The main objective behind this report (Deliverable D2.2) is to study the structure of blogs and develop a generic data model for their preservation and archiving. This data model is also expected to support effective data mining, efficient preservation and robust repository features. It is necessary, however, to explain what the data model is and what the rationale is behind developing it.

Data models are considered essential for designing and developing data systems. Systems, like those anticipated from BlogForever, rely heavily on storing, retrieving and maintaining large amounts of data. The development of data systems, however, requires a thorough understanding of the information needed by the stakeholders and users of the system. Data models provide a method for *exploring* and *describing* existing information requirements, and *communicating* with various stakeholders [3].

To explore and describe information requirements, data models must record the content, shape, type and rules of data elements used for the operational processes of the system *(ibid.)*. They should serve as a conceptual replica of the data structures required in the database system. Data models

[3] *For details see: Grant Agreement Annex I - Description of Work (DoW), page 7.*
[4] *Submitted on 30 August, 2011, and available for download at http://blogforever.eu*

describe the ways the data should be organised, without necessarily reflecting the operations expected to be performed on the data.

Communicating requirements to a number of involved stakeholders is another integral role of a data model. Being an abstract representation of requirements for the system data, a data model omits technical or implementation details – making it easier to communicate with administrators, customers and system users. On the other hand, data models can be used as blueprints by developers and other technical staff. Therefore, data models bridge real-life information systems and database systems, as well as administrative and technical staff.

The formal definition of the term 'data model' varies from one source to another. Most of the definitions, however, emphasise the importance of the structure and semantics of data to be captured by the data model. Klein and Hirschheim [4, p. 8], for instance, define a data model as: "a way of perceiving, organising and describing data". Similarly, West [5, p. 5] characterises data models as a means for "defin[ing] the structure and intended meaning of data". Subsequently, data modelling is considered to be "the technique for exploring the data structures" [3] or "the activity by which the data model is applied to derive a logical organisation of what is documented in a (conceptual) schema" [4, p. 8]. For the purpose of this report, the approach to data modelling defined by Klein and Hirschheim *(op. cit.)* is considered sufficient, subject to identifying and selecting a standardised approach for representing the developed data model as explored in the following two sections. The chosen approach to data modelling is guided by the principles described by West *(op. cit.).*

1.5 Approaches to Data Modelling

Data Modelling is considered to be an integral phase for designing and developing data systems. Although essential to a design process, the methods for developing data models vary widely.

The differences across data modelling practices are reflected in the principles/paradigms of modelling, approaches and methods used, and representational notations and standards. The diversity of approaches and methods is driven by variations in requirements, size or complexity of the designed system. Hence, there is no single solution for conducting data modelling, but a set of approaches available for choosing the most appropriate ones.

It is argued that data modelling approaches are confined to philosophical assumptions about the nature of reality [4]. The two major perspectives on reality – objectivist and subjectivist – are believed *(op. cit.)* to be reflected in data modelling approaches. Objectivist approaches mirror the reality, while subjectivist ones attach importance to expressing socially constructed meaning. A subjectivist position claims that data can only make sense when passed to someone and cannot have an objective meaning.

Subjectivist positions may not be reflected in entity-based approaches to data modelling. The entity-based approaches rely on the notion of an entity that represents an object in the real world. Information about the object is usually recorded as consisting of descriptive properties and relationships with other entities. Although entity-based approaches require a unique ontological view of the reality, these approaches are widely adopted and most frequently used. The tools and technologies that support entity-based modelling are also well established and accepted. The use of entity-based models for the BlogForever project is sufficient and (due to their wide adoption) cost-effective.

In addition to philosophical paradigms, data models can vary according to their information level. Depending on the audience of the data model, some complexity of data structures may be either hidden or made explicit as necessary. More abstract, high level data models can be useful for

communicating general ideas. Developers and programmers, however, may require additional details for using the model as a blueprint for building the database system. Figure 1 demonstrates the levels of information and the transition of data models from one level to another. This report, first, discusses the model at conceptual level and then discusses the details relevant to the logical level.

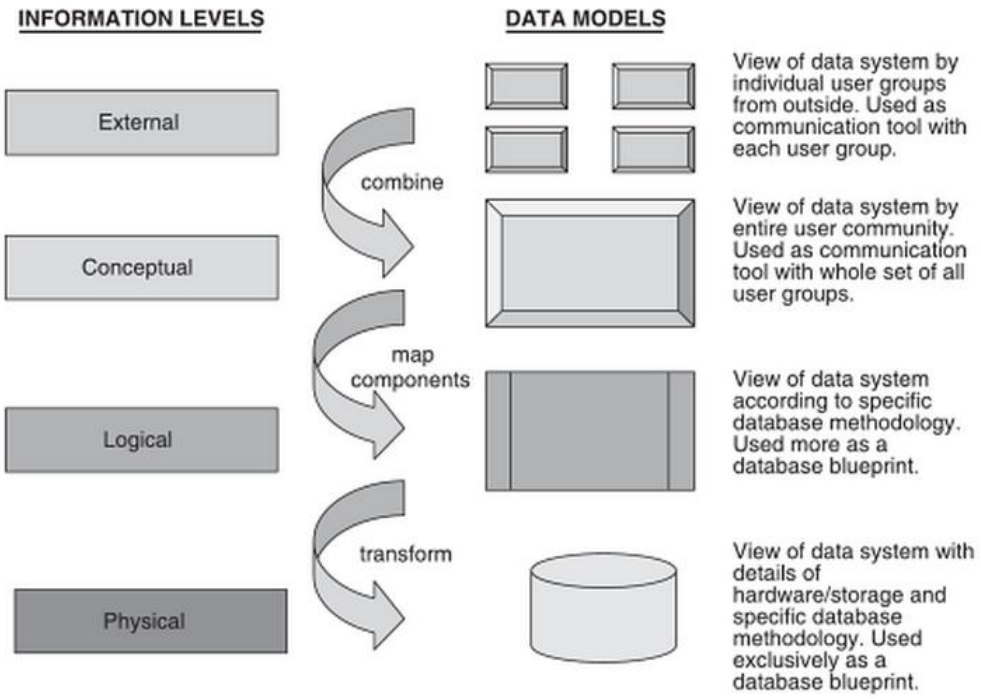

Figure 1 - Data models at different information levels [from 3, p. 14].

1.6 Data Modelling for BlogForever

For the purposes of the BlogForever project, conceptual and logical information levels have been chosen for representing the proposed data model. The decision was based on the necessity to provide both a high level view of a conceptual model as well as the more detailed logical one. However, steps have been taken to enable efficient transition of the proposed logical model into a physical one. This section continues reporting on the process of developing the model, beginning with the methods and representational standards chosen.

There are two primary approaches to data modelling – the more traditional approach of modelling by *normalisation* and an *ontological* one [5]. Normalisation is based on looking for repeating patterns and anomalies in the data, and adding new entities to eliminate them. The process of normalisation is a systematic step-by-step approach that starts from listing data attributes and proceeds by applying normalisation rules for identifying irregularities. Hence, normalisation is usually referred to as a bottom-up approach. In contrast, the ontological approach to data modelling can be seen as top-down. Database design that is compliant with a certain normal form remains outside of the scope of this report.

There is a conceptual difference between the ontological approach and normalisation. The ontological approach looks at ‘things’ (or concepts) that data are about to represent and uses these ‘things’ as a basis for structuring the data. It requires understanding what concepts are and what relationships they have among them. Although normalisation is widely used and well documented

in the field of data modelling, it is believed that application of ontological principles to data modelling is more (or at least similarly) robust compared to normalisation [5].

The process of data modelling undertaken as part of the BlogForever project (and reported in this document) conforms to the general principle of the ontological approach. The representation of the derived model is based on standard notation guidelines. An Entity-Relationship model has been chosen for representing the model graphically. The logical data model presented in this report (Section 9.2) complies with the Information Engineering (IE) notation. This notation is widely adopted and is considered readable by both technologist and wider audiences. The primary and foreign keys have been collapsed to optimise the use of the limited graphical area. The conceptual model (Section 9.2), however, does not strictly follow the standardised notation. This fully conscious decision was made to improve general readability of the model. Graphical representation of the conceptual model was created using Microsoft Visio 2010[5]. The logical model was developed using IBM InfoSphere Data Architect[6] software. Hence, the developed model can be made accessible to other Data Architect users, or exported into one of the large number of formats supported by the system.

[5] http://office.microsoft.com/en-gb/visio/
[6] http://www-01.ibm.com/software/data/optim/data-architect/

2 Related Work: Modelling and Mining the Blogosphere

The Blogosphere constitutes an immense and increasingly rich source of information. The explosive adoption of blogs by individuals and institutions has resulted in massive amounts of data being published online. Today blogs exhibit commentary, images, audio and video content. Bloggers use the blogging tools in innovative ways to communicate, test ideas, gain visibility and so on. Reflecting the explosive growth of other social media such as, YouTube[7], Flickr[8], Facebook[9], Twitter [10]or Delicious[11], the Blogosphere is becoming *increasingly rich* and *complex*.

The complexity and richness of today's Blogosphere pose considerable challenges for the preservation and information retrieval communities. The BlogForever project is aiming to develop solutions for blog preservation to overcome these challenges. Developing a data model that is fit for purpose is the immediate challenge that is discussed here. Given the restrictive nature of data models, extra care should be taken in developing a model that enables *effective data mining*, *efficient preservation* and *robust repository features*[12].

To ensure that the developed model is capable of supporting the aforementioned objectives the following actions have been identified as necessary preconditions:

1. Derivation of a working definition of a blog
2. Definition of a methodology for developing the model
3. Alignment of the developed model with the requirements of the project.

A working definition that describes the object of preservation is one of the most important prerequisites. This is due to emerging and continuously evolving nature of blogs. A working definition will allow establishing solid grounds before proceeding with the data modelling and drawing necessary boundaries for keeping the project manageable. The next section of this report looks into the common definitions of blogs and sets out the grounds for further progress in developing the model. The methodology adopted in the data modelling and the alignment of the proposed data model with the requirements is described in Chapters 4 and 10 respectively.

2.1 Working Definition of a Blog

A brief review of the academic literature shows that the definitions of blogs vary. The Oxford English Dictionary (OED) defines a blog (weblog) as: "a personal website or web page, on which an individual records opinions, links to other sites, etc. on a regular basis". The verb 'to blog' is defined as the process of adding new material or updating a blog regularly. These definitions of the terms 'blog' and 'blogging' highlight the temporal nature and periodic activity on blogs.

A similar view is adopted by Nardi and his colleagues [6, p. 43] who define a blog as follows: *"Weblog is a kind of web page [with] frequent, usually brief posts, with the immediacy of reverse chronological order"*. Unlike the OED definition, Nardi highlights the characteristic of having timed and sorted blog entries.

Other definitions, for instance [7, p. 200], deviate from a standpoint that looks into the technical aspects of blogs into the socio-cultural role of blogs. As is put more cogently *"…blog represent a new medium for computer-mediated communication and offers insights in the way bloggers present*

[7] *http://www.youtube.com*
[8] *http://www.flickr.com*
[9] *http://www.facebook.com*
[10] *http://www.twitter.com*
[11] *http://www.delicious.com*
[12] *For details see: Grant Agreement Annex I - Description of Work (DoW), Part A, Task 2.2, page 15.*

themselves online, especially in the form of self-expression and group relationships, both of which impact the construction of identity.".

These conceptually distinct perspectives taken for describing the same object – a blog – demonstrate the importance of the identified and earlier discussed (see Chapter 2) need for defining a blog.

2.1.1 Blog as a Platform

In an attempt to define clear boundaries for the consistent and structural progression with the development of a data model and to keep this research project (i.e. BlogForever) within manageable limits, the following working definition of a blog has been adopted.

> *A blog is a web service that encompasses an accessible and widely accepted mechanism for creating, maintaining and automatically distributing chronologically published material on the Web, along with the feedback and user domain associated with it.*
>
> *The view of a blog that includes any other features/services (i.e. customising and enriching blogs) is defined in this report as an extended blog.*

Notes of clarification to the working definition:

- ✓ *Accessible and widely accepted mechanism* implies that no HTML or programming skills are required for using the service that is accepted widely enough to be considered a norm.
- ✓ *Mechanism for creating and maintaining* implies that users have control over adding, editing or deleting their content.
- ✓ *Mechanism for automatically distributing* requires the existence of RSS or similar techniques for automatically distributing the content.
- ✓ *Published material* can include a range of media, including embedded content.
- ✓ *Feedback and user domain* implies that platform can be open to interact with the published material.

This working definition excludes websites that encompass static HTML content – restricting the concept of a blog to a range of commonly represented blogging platforms and tools. The included tools should ensure accessibility of the mechanism to those with little technical knowledge and few skills necessary for publishing material on the web. Web feeds, such as RSS and Atom that are primarily used for distributing content, are made integral to the concept of a blog. Users (including authors and readers) are also essential to the definition of blogs. Extended blogs, on the other hand, constitute a richer environment that enables stylistic customisation and functional extension by means of internal modules or external applications or widgets.

2.1.2 Blog as an Object for Preservation

The definition of a blog presented in section 2.1.1 describes the technical infrastructure required before an object can be considered to be a blog. The goal of this section is to describe what aspect of the object describes the blog viewed as "an information object" [e.g. 8, p. 6]. Typically, an information object is described as a combination of its attributes as a conceptual object (e.g. as it is recognized and perceived by a person), as a logical object (e.g. as it is understood and processed by software), and, as a physical object (e.g. as a bitstream encoded on some physical medium).

On a conceptual level, the dominating characteristic of a blog can be described as the periodic transmission of material that appears on the blog, resulting in a collection of publications displayed in reverse chronological order, where updates of the continuous stream of user activities, associated with these communicative actions, are distributed to subscribers of notification services such as

Really Simple Syndication (RSS)[13], Atom Syndication Format[14] and Atom Publishing Protocol[15]. As a logical and physical object, a blog represents a compound entity, broken down into a set of components: posts, pages and comments, which, in turn, consists of a range of different logical objects (e.g. links, images, audio). The representation of a logical object as an integral part of the blog can be seen as physical objects: for example, textual content, as a set of characters, along with its stylistic representation (e.g. font, colour, size), encoding, language, and bitstream expressed on the selected medium would be a typical physical object. The physical level description primarily deals with files, their storage and retrieval.

On the highest conceptual level, a blog can be defined as a stream of information in the following way:

> *Blog (as an object for preservation) is a chronological stream of user-generated content and associated activities to be represented with a level of granularity enabling its use by humans and computer systems.*

This interpretation of a blog (conceptual object) is demonstrated in a graphical format in Figure 2. The stream is not limited to the content delivered via web feeds and includes pages and other user activity. The granularity of the object (i.e. existence of logical objects, such as blog, post, comment, author, etc.) should be ensured during the data modelling stage. Low level, detailed representation of the digital object is discussed further down in the report (Chapter 9).

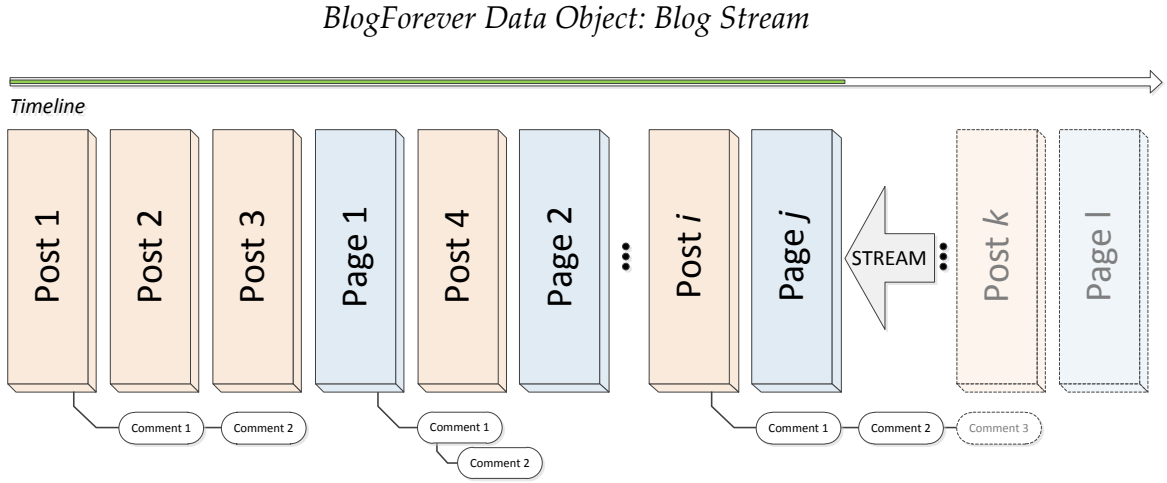

Figure 2 - Blog stream: a high level representation of a digital object.

At the heart of the BlogForver project lies the aim to establish best practices for maintaining the "authenticity, integrity, completeness, usability, and long term accessibility"[16] of weblogs.

To maintain authenticity, digital preservation must go beyond preserving the bitstream *content* and *conceptual structure* that constitutes the fundamental building blocks of the digital information. The preservation approach must support the extraction and preservation of the object *context*, such as circumstances that led to its creation (e.g. formulae that led to numbers and figures), how it was used, how the object was acquired, what changes took place during the object's life time, and its custodians before its arrival into the repository.

Some aspects of *functionalities and behaviour* (e.g. expectations following the clicking of links, whether it is editable, searchability) of the object might also be crucial in establishing integrity and

[13] *http://www.rssboard.org/rss-specification*
[14] *http://tools.ietf.org/html/rfc4287*
[15] *http://bitworking.org/projects/atom/rfc5023.html*
[16] *page 3, Part A, BlogForever project Description of Work*

completeness of the object, to the community of users who want to re-create the *experience* of the object as a *performance* and/or to establish its reliability as *evidence* [cf. 9]. Just as no two performances of a musical piece can be identical, no two instantiations of the object can be identical. In essence, there is no physical realisation of the *original object* that can be preserved. It is only possible, at most, to maintain an *acceptable range of variance* amongst recurring instantiations of an object [cf. 10, 11]. It has been noted, however, that formally expressing functionality and behaviour of objects is not a straightforward task [e.g. see 12], and, the complexities of understanding object context has been observed [e.g. see 13]. For example, the same element could signify different contextual entities: just as the same text "September 11" could denote a date or an event[17], the text of a blog post can constitute blog content or the context of creation for a comment on the post.

Likewise, while the organisation of object characteristics into conceptual, logical, and physical aspects is useful, the boundaries can become somewhat blurred when supporting a robust preservation framework for the retention of semantics. For example, some of the logical and physical components of the blog object, given as examples above, can also be translated into conceptual components as recognised by human beings: in the end, conceptual understanding of the blog as a piece of information is only possible with the understanding of the language, embedded links, what they do, and stylistic layout. There are specific conventions (i.e. pragmatics) in using the English language within a community that allows the members of the community to correctly interpret what it means when, say for instance, text is surrounded by quotation marks, italicised, or displayed in larger fonts. It is not only the representation of the page that is affected by the failure to retain such semiotics, but also the semantics as it is realised by a user, directly impacting on the usability and accessibility of the object over time.

This brings the discussion to the central role of the semiotics triad, *syntax*, *semantics*, and *pragmatics*, in describing an object, respectively, in terms of shared expression, understanding, and conventions of use, for all agents represented by both humans and machines alike. Even the afore mentioned *context* has manifested itself through these three channels of semiotic communication in several areas of knowledge management and information extraction [e.g. see 14, 15]. These concepts are also mentioned in the preservation community: their interest in semantic accessibility is especially noticeable (on all the levels of conceptual, logical and physical objects) at initiatives that bring together, for example, the semantic web and digital archiving practices[18].

Chapter 12 describes possible scenarios to illustrate the potential of the data model in addressing some of the issues described above: the concepts of content, context, functionality and behaviour, and experience will be revisited to present examples of how elements of the data model (Chapter 9) might be applied to capture selected aspects of these concepts. The discussion will be in light of the technical requirements identified during the technological survey (Section 6.1 of Chapter 6), proposed methods for data extraction (Section 6.3 of Chapter 6, to be discussed further in deliverable D2.4 and D2.6 of the BlogForever project), and the user community's perception of blogs as evidenced by the user survey (Section 6.1 of Chapter 6).

The precise formulations, however, of how the data model will be absorbed into the preservation strategy, including proposed solutions for the difficulties observed above, will be investigated further within Work Package 3 (WP3) of the project.

[17] *page 106, Lee, Christopher A. "A Framework for Contextual Information in Digital Collections." Journal of Documentation 67, no.1 (2011): 95-143*
[18] *http://sda2011.dke-research.de/index.php/topics*

3 Blogs and Conceptual Blog Models

Prior to proceeding with the task of data modelling it is necessary to review the literature that conceptualises blogs and their structures. The review of conceptual models (as described and used in the literature) can improve the general understanding of a blog and provide foundations for identifying their most prominent elements that are common in the models. Furthermore, this section outlines some of the available data models suggested for developing blogging tools or already implemented and used within blogging platforms. The existing data models can reflect the current structure of the data used for operating a blogging website.

A notable conceptualisation and representation of a blog is presented by Nakajima *et al.* [16]. They view blogs as journals that are available on the web. However, their representation and conceptualisation of the blogs extends beyond this simple definition. The authors view blogs at different levels: [a] blog site, [b] blog entry, and [c] blog thread (defined by reply or trackback links). Each of those perspectives defines a set of concepts and associations between them. The general concept of a blog as they describe it is presented in Figure 3.

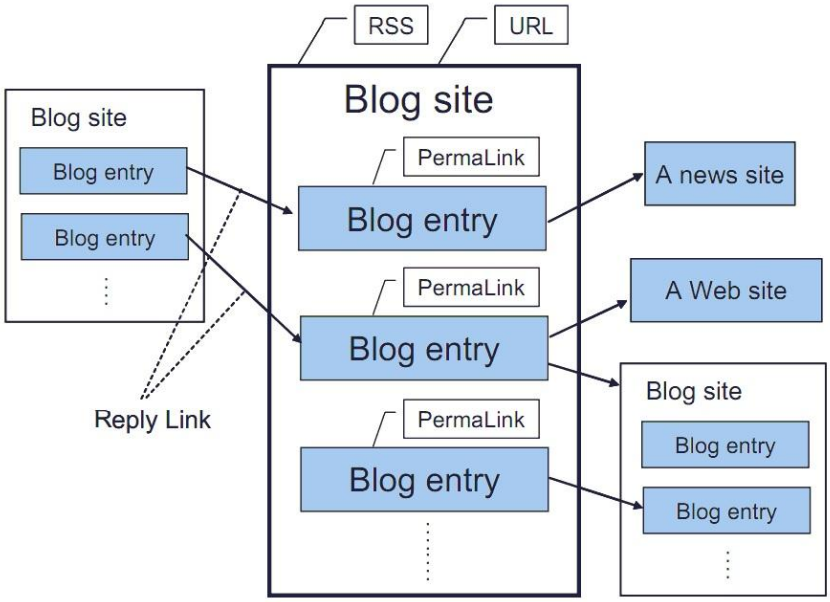

Figure 3 - Conceptual model of a blog at a website level [from 16]

The model (Figure 3) does not exclude links to other blogs as websites. They use arrows to denote the directions of the links from/to external sources. RSS, URL and blog entries are among the highlighted concepts. *Site name* and *blogger/author* are added to the more detailed representation of the blog at the level of a website (Figure 4). Blog entries are, subsequently, presented in relation to the blog website or other entries and include: *title*, *time*, *description* and *comment* (Figure 5). A blog thread (Figure 6) represents a set of blogs, connected with each other via reply or trackback links.

The paper by Nakajima *et al. (op. cit.)* defines the model for a particular purpose, that is, discovering and categorising bloggers to help identify trending topics. However, it implicitly defines the importance of a perspective on blogs that does not exclude related blogs and resources. It also highlights some of the prominent attributes of blogs and blog entries as shown in Figure 4 and Figure 5.

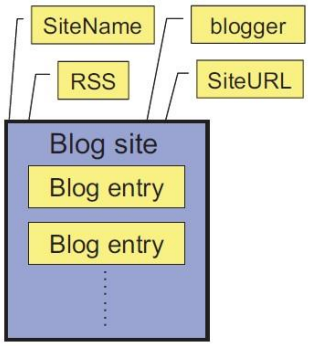

Figure 4 – Representation of a blog website

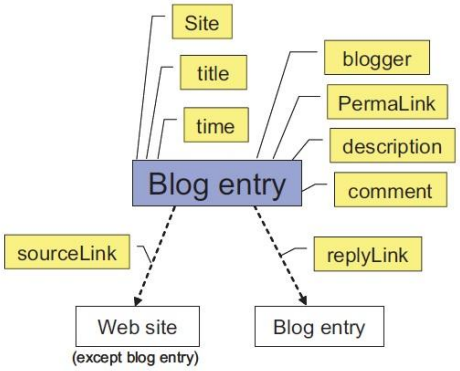

Figure 5 – Representation of a blog entry

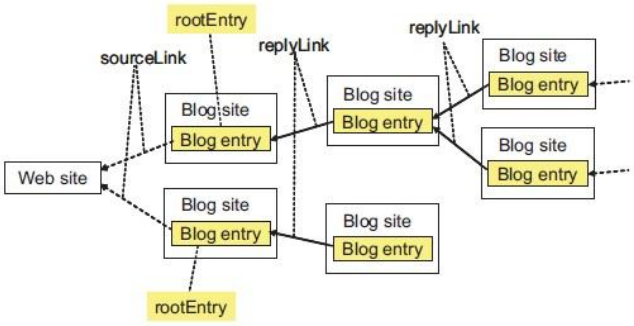

Figure 6 – Representation of a blog thread

Another, high level conceptual blog model is defined by Ta *et al.* [17]. The proposed model is presented in RDF and is inspired by the Annotea social bookmarking project. The authors employ the model to propose an architecture – the Web of People – for discovering and sharing information. The main contribution of this schema is to inform the development of a semantic portal. Interestingly, the authors define two subclasses of an entry: Topic and Post (Figure 7). Posts are described as user observations about one or more web resources, while topics denote categories for indexing posts.

The model is using an already existing namespace as well as introducing a native one relevant to the 'Web of People'. They are used to define associated metadata. For instance, atom:author or atom:title are using the Atom schema to define the necessary metadata associated with an Entry. Other properties are introduced as necessary to specify possible interrelations between the identified classes and instances.

The blog model proposed by Ta *(ibid.)* is abstract and at a high level, but the representation of the described blog does not fit the definitions derived in Chapter 2 of this report. Ta's model is intended for the specific web applications discussed as part of the proposed architecture. Nevertheless, this model remains informative within the context of BlogForever, as it demonstrates an alternative method (i.e. RDF in this case) for representing a blog model and the possibility of having recursive references that may exist between the constituent parts of a blog.

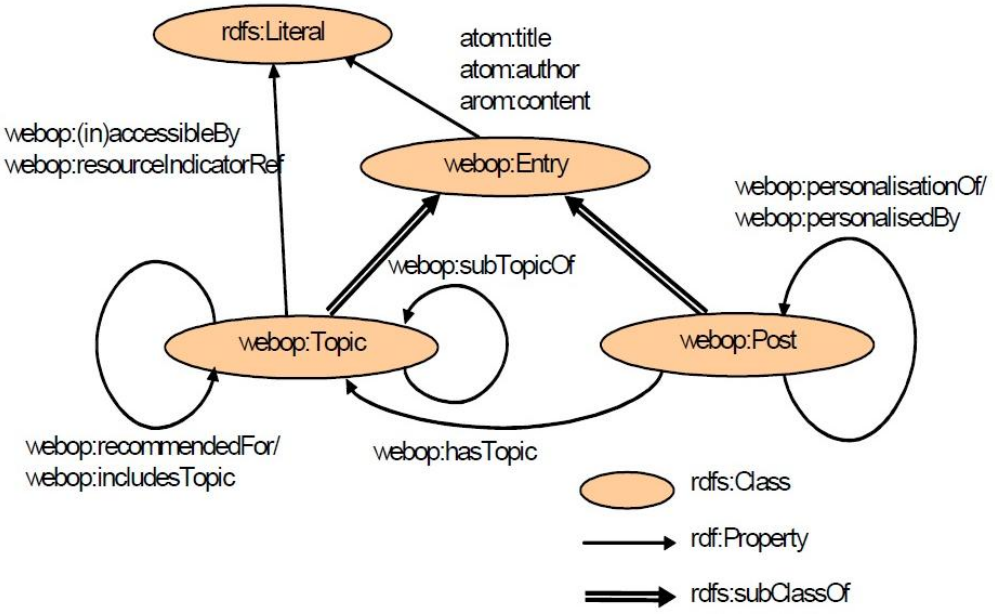

Figure 7 - Blog model [from 17]

4 Data Modelling: Planning and Progress

The development of a data model is generally advised to be guided by a requirements definition phase. The rationale behind drawing a set of requirements is to ensure that the data model addresses functional requirements and physical requirements for the solution being developed. The requirement definition stage, as suggested by Ponniah [3], may include interviews, groups sessions, documentation, change management and so on. Some similar exercises are discussed in Chapters 6, 7 and 8. Others are still being conducted, including interviews for deriving user system requirements (WP4), establishment of preservation policies (WP3) and identification of data extraction methodologies (WP2). However, the primary requirements of the project have already been defined and agreed as part of the project agreement[19], and summarised in sections 1.2 and 1.3 of this document.

While this document proposes a data model for preserving blogs, to meet the requirements of the BlogForever project it would necessary to position the model along with expected mechanisms of the developed preservation solutions. Therefore, in addition to addressing the issue of blog preservation, this report extends to discuss the issues related to alignment of the data model with the adopted Invenio software suite (i.e. repository management) and the interoperability of blog data between repositories. This report first describes the proposed blog data model (Chapter 9) and follows to discuss the issues of repository management and interoperability (Chapter 10).

The proposed blog data model, as summarised in this report, was developed in a number of consecutive phases. Each of the phases contributed to the process of defining the requirements expected from the data model and, therefore, informing the development of the proposed model.

To ensure that the developed data model is sufficiently rigorous a set of iterative modelling cycles have been established. This structured approach required each of the development cycles to inform the process of data modelling leading to the review and refinement the data model. The cycles included the following consecutive steps.

1. An insight into the database structure of open source blog systems (Chpater 5).
2. A retrospective view on an earlier conducted online user survey to identify important aspects and types of blog data to be preserved (Section 6.1)[20].
3. A retrospective view on the technologies and standards used within the Blogosphere (Section 6.2)[14].
4. Suggestions derived from an earlier inquiry into the recent developments and prospects for analysing networks and dynamics of blogs (Section 6.3) [14].
5. An inquiry into blog structure based on evaluation of 2,695 blog feeds (Chapter 7).
6. An insight into blog APIs (Chapter 8).
7. Proposal of the data model (Chapter 9).

The results from each of the above allowed development of a coherent view that guided the development of the required blog data model.The outcomes and implementation details for each step are discussed further down this report.

Last but not least, the report extends to accommodate internal consultation from blog service provider Phaistos Networks and other partners specialising in digital preservation. The consultation exercises introduced clarifications and modifications, as well as indicated about possible refinements emerging from the work towards the anticipated preservation policies (WP3), data extraction methodologies (WP2) and user requirements for the platform BlogForever platform specification (WP4).

[19] *Grant Agreement Annex I - Description of Work (DoW).*
[20] *For details, see submitted deliverable - BlogForever: D2.1 Survey Implementation Report. 30 August, 2011*

5 Blog Data Models

An insight into database structures designed for operating a blog can provide clues for developing the required data model for archiving solutions (as anticipated from the BlogForever project). This chapter aims to refer to existing data models that describe blogs to derive core structures that must be represented in the data model developed for the BlogForever platform. Hence, this chapter draws from the already operational data structures that define and constrain the data stored behind the blog (e.g. date/time attribute associated with each post) to ensure that important data attributes are captured as part of the data model development within the context of BlogForever.

Despite the rapid growth of the Blogosphere and the related web services, the number of papers that describe explicit data structures behind the blogging tools is limited. Yet, there are several open source tools and generic models that may offer the needed view. The following sections present data models, which were reverse-engineered from WordPress[21] and Movable Type[22] open source blogging platforms, as well as from common recommendations for developing blog data models given in given textbook references.

5.1 Generic Data Model

The database design for blogging software may vary from one application to another. The complexity of the given web application that powers the blog will affect the number entities/tables and relationships between those.

A simple blog application that allows its users to make posts and accept comments will still require a form of a database. Examples of database designs for simple blogs appear in software development textbooks and references. They use development of a blog as a simple example to illustrate programming constructs and demonstrate techniques for handling the data-related aspects of the application. For instance, Davis and Phillips [18] suggest using the following entities:

```
CREATE TABLE 'posts' (
  'post_id' int(11) NOT NULL auto_increment,
  'category_id' int(11) NOT NULL,
  'user_id' int(11) NOT NULL,
  'title' varchar(150) NOT NULL,
  'body' text NOT NULL,
  'posted' timestamp,
  PRIMARY KEY  ('post_id')
);
CREATE TABLE 'categories' (
  'category_id' int(11) NOT NULL auto_increment,
  'category' varchar(150) NOT NULL,
  PRIMARY KEY  ('category_id')
);

CREATE TABLE 'comments' (
  'comment_id' int(11) NOT NULL auto_increment,
  'user_id' int(11) NOT NULL,
  'post_id' int(11) NOT NULL,
  'title' varchar(150) NOT NULL,
  'body' text NOT NULL,
  'posted' timestamp,
  PRIMARY KEY  ('comment_id')
);
CREATE TABLE 'users' (
  'user_id' int(11) NOT NULL auto_increment,
  'first_name' varchar(100) NOT NULL,
  'last_name' varchar(100) NOT NULL,
```

[21] http://wordpress.org
[22] http://www.movabletype.org

```
'username' varchar(45) NOT NULL,
'password' varchar(32) NOT NULL,
PRIMARY KEY  ('user_id'));
```

Another example of a textbook, PHP5 and MySQL Bible [19], demonstrates another, similar, example by storing user data and the published posts. A generic and simple model, represented as an entity relationship diagram is contributed by Williams [20].

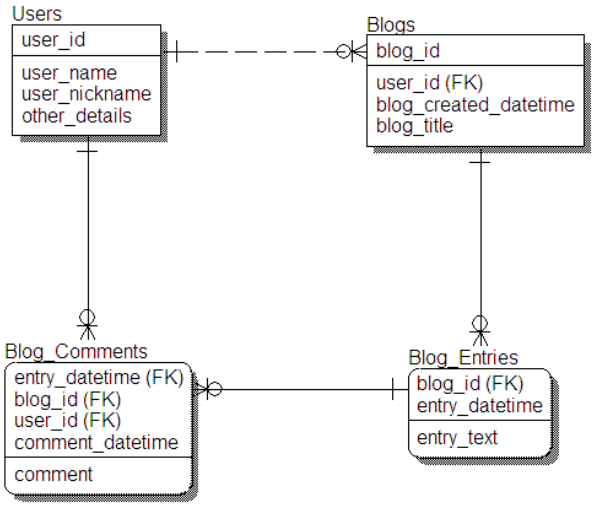

Figure 8 - Generic and simple data model for blogs [from 20]

5.2 WordPress Data Model

Generic blog models can provide some basic information on what programmers and software developers may think blogs are. Beyond these simple examples, however, lie already established blogging platforms and services that have been tested over time. WordPress is one of these widely used and accepted blogging tools.

WordPress is an open source blogging tool powered by PHP and MySQL. As an open source software with a modular design, WordPress encourages third party development and offers a range of plugins, extensions and themes. To acquire an insight into the data structures behind WordPress (v.3.2.1) a reverse engineering exercise was conducted on a newly installed blogging system. The IBM Rational Data Architect (RDA)[23] data modelling tool was used to visualise the logical data model from the operational database. The resulting diagram is presented in Figure 9.

In addition to the entities used in simpler examples, WordPress contains data structures to store metadata associated to posts, comments, users and tags. A separate entity for links stores the name, owner, URL, associated RSS and so on.

[23] *IBM RDA allows development, exploration and modification of database models and their constituting databases.* http://www-01.ibm.com/software/data/optim/data-architect/

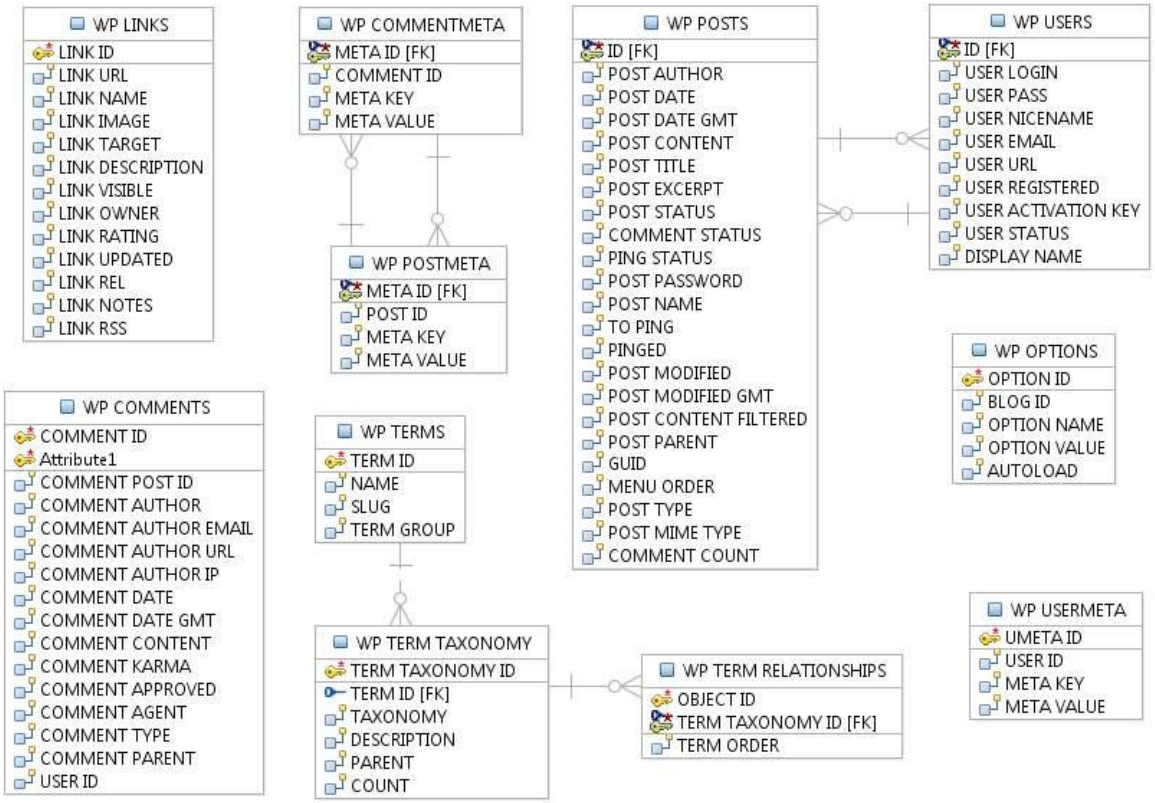

Figure 9 - Entity relationship model for a default installation of WordPress software.

5.3 Movable Type Data Model

Movable Type is another open source blogging platform. It is one of the first blogging tools that appeared on the market. Its history as an open source system, however, is relatively new – having the sources released in the end of 2007. Movable Type powers significantly fewer but still a considerable number of blogs. For instance, TypePad[24] is one of the free blogging services powered by Movable Type.

Similar to alternative blogging platforms, Movable Type offers a number of features for managing standalone content pages, files, user roles, themes, categories, and trackback links. Unlike WordPress, Movable Type requires a Perl interpreter and support for executing CGI-scripts. It is possible to deploy multiple blog sites – a feature that comes as an inbuilt option of the general distribution package.

The database of the Movable Type system contains 43 tables – offering a wider range of structures for maintaining blacklisted IPs, logs, roles, file/asset management and so on. The list of tables is shown in Table 1. It is apparent that database structures of Movable Type enable email notifications, archiving options and information filtering, while WordPress may require installation of extensions to enable the support of similar features.

[24] http://www.typepad.com/

Table 1 - Listing of database tables form the default installation of Movable Type.

ASSET	**ENTRY_REV**	**TAG**
ASSET_META	**ENTRY_SUMMARY**	**TBPING**
ASSOCIATION	**FILEINFO**	**TBPING_META**
AUTHOR	**FILTER**	**TEMPLATE**
AUTHOR_META	**IPBANLIST**	**TEMPLATEMAP**
AUTHOR_SUMMARY	**LOG**	**TEMPLATE_META**
BLOG	**NOTIFICATION**	**TEMPLATE_REV**
BLOG_META	**OBJECTASSET**	**TOUCH**
CATEGORY	**OBJECTSCORE**	**TRACKBACK**
CATEGORY_META	**OBJECTTAG**	**TS_ERROR**
COMMENT	**PERMISSION**	**TS_EXITSTATUS**
COMMENT_META	**PLACEMENT**	**TS_FUNCMAP**
CONFIG	**PLUGINDATA**	**TS_JOB**
ENTRY	**ROLE**	
ENTRY_META	**SESSION**	

The comparison of database structures of WordPress and Movable Type demonstrates an overlap between some of the entities and their attributes. They also relate to the generic data models shown above. These commonalities can be used to corroborate or identify commonly perceived important elements of blogs. For instance, if entities or attributes appear in more than one system, it serves as an indication for considering their use in the model that is being developed. An insight into the database structures of both WordPress and Movable Type is particularly useful in regards to understanding basic blog structures, differentiation and management of MIME types, representation of various types of links and mechanisms for representing metadata.

6 Inquiry into Blogging, Technology and Analysis of Blogs

As an integral part of the BlogForever project and as an important prerequisite to its progress, a study has been conducted to inform the development of preservation and dissemination solutions for blogs. This broad study covered [a] a user survey exploring the aspects of blog preservation and blogging practices in general, [b] an investigation into the use of tools and technologies within the Blogosphere, and finally [c] an inquiry into the recent theoretical and technological advances for analysing blogs and their networks.

As a prerequisite to the data modelling task, the study informed on aspects and types of blog data that should be preserved. The detailed analysis of the survey is available in Section 4.3 of the BlogForever: D2.1 Survey Implementation Report [1]. A brief summary of the results that are relevant to data modelling are presented in this section.

6.1 Online User Survey

The online user survey targeted both blog authors and readers. Nine hundred blog authors and readers completed the questionnaire. The demographics of survey participants were fairly diverse with a fair distribution of age groups, nationalities, counties and spoken languages.

The results demonstrate that most of the blog users published a variety of multimedia content with an overwhelming predominance of text (98%), photographs (83%) and video/animation (43%). When asked to specify the elements of the blogs they would like to preserve, the responses varied, with large number of respondents expressing willingness to preserve entire blogs. The overview of the results is presented in Table 2 and Figure 10.

Table 2- Perceived importance of preserving blog elements

Preserved Element	Very Important (%)	Important (%)	Neutral (%)	Very Unimportant (%)	Unimportant (%)
Whole blog	46.3	29.7	15.4	2.5	2.0
Posts	45.7	25.8	15.8	2.1	2.3
Comments	25.4	35.9	21.1	2.5	6.8
Specific sections	20.7	27.5	31.8	2.3	5.5
Date tags	20.7	26.2	28.7	5.9	7.6
Categories	18.4	28.7	27.9	6.1	7.8
Contributing Authors	18.4	22.7	31.4	10.4	6.6
Visual layout	17.8	31.3	22.5	8.6	9.4
Topic tags	17.6	27.3	28.7	6.6	8.8
Design	17.2	29.9	23.0	8.6	10.9
Attachments	16.8	23.6	32.8	7.4	6.4
Internal links	15.6	23.2	33.8	6.3	10.0
Author tags	15.6	23.0	31.8	7.8	9.0
Metadata	15.4	20.3	35.7	7.0	9.0
Registered users	15.4	20.1	33.0	9.6	10.5
External links	15.0	27.3	31.6	6.4	8.4
Communities of users	12.9	17.8	38.3	10.5	8.6
Commenting systems	12.5	20.1	37.5	6.3	11.7

Search box	12.5	18.8	34.0	13.3	10.4
Embedded Widgets	11.1	18.2	36.1	12.5	10.4
Feeds	10.4	14.1	41.2	8.8	13.5
Calendar	9.8	15.4	36.7	13.5	13.3
Blogroll	8.6	14.8	37.9	10.0	16.2
Slide show	7.8	11.7	40.0	13.5	14.6
Sponsors of the blog	7.6	9.2	32.6	20.3	17.0

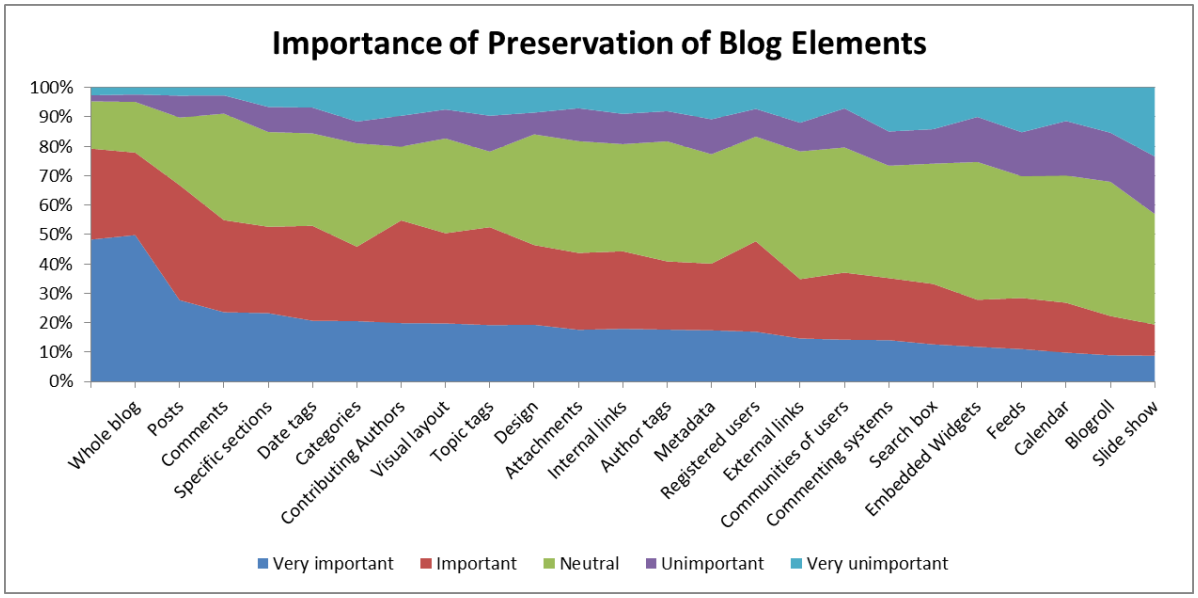

Figure 10 - Perceived importance of preserving blog elements.

The results of the survey suggest that archiving the entire blog would be the most preferable option for blog authors. However, some of the elements were perceived as less important, including: slide shows, sponsors, calendars, feeds, blogroll, embedded widgets and search boxes.

While the majority of blog authors (85.7%) never used an archiving service/software for preserving their blog, around 77% of the respondents were positive about using a trusted repository for blogs. A large number of respondents expressed interest in repository features that could contribute to increasing their readership.

Inferring from the results of the survey, the developed data model should enable storing a variety of data types. Costs and benefits of preserving elements of blogs perceived as less important should be considered.

6.2 Inquiry into the Use of Technologies, Tools, Formats and Standards in the Blogosphere

The section casts a retrospective view over the inquiry into the technological foundations of the current Blogosphere. The details of the inquiry are available in Chapter 5 of the BlogForever: D2.1 Survey Implementation Report [1]. The aim of this chapter is to reflect upon the primary findings and align the proposed data model to reflect the most prominent technological aspects of the blogosphere or trends.

The study was based on a large-scale evaluation of active blogs. The sample has been acquired primarily from the Weblogs.com[25] ping server and extended with a list of top ranked blogs from Technorati[26] and Blogpulse[27].

The study looked into the:

- ✓ Platforms and Software Used
- ✓ Document Character Sets
- ✓ Use of CSS, Images, HTML5 and Flash
- ✓ Semantic Markup: Microformats, Microdata and Metadata
- ✓ RSS and Atom Feeds
- ✓ APIs and Libraries
- ✓ Social Media
- ✓ Media Types and Common File Formats
- ✓ Single Posts versus Websites
- ✓ Differences between the Blogosphere and Web

The results lead to the following suggestions that in relation to the development of the data model.

6.2.1 Recommendations for the Data Model

The study revealed that WordPress and Blogger are the two dominating (55%) blogging platforms available on the market. However, a considerable number of resources (40%) had no indication of a platform. It is, therefore, necessary to state that

1. The data model should allow storing the data from WordPress and Blogger, yet, should not be limited to the dominating platforms only.

Due to a spread of software versions across blogging platforms it is necessary to ensure that:

2. Preservation of blogs should not be limited to specific versions of blog platforms.

Evaluation of the `content` attributes across the studied resources showed that 61% of the studied resources explicitly specify the content type via HTML `<meta>` tag to be of type `text/html`. The use of other content was also recorded including: `application/xhtml; /xml; /xhtml+xml; /vnd.wap.xhtml+xml`, as well as `text/xml; / javascript; / phpl; / shtml; and / html+javascript`. A wide variety of charset specifications were also identified. While the dominating charset is utf-8, other standards were also fairly widely used. It is therefore inferred here that

3. The data model should store the identifiers of the collected content type and charset specifications.

The evaluation of the use of CSS and Images has been shown to be widely accepted. 81% of resources used CSS. The average number of graphics such as JPG, GIF and PNG varies between 4 and 8. Given the importance of elements of presentation of blogs it is suggested that

4. The data model should enable the storage of CSS files and associated images.

A considerable use of HTML5 was recorded (25%). The number of Flash elements used (15%) within the Blogosphere was identified as lower than within the general Web (44%). Given the lower

[25] *http://weblogs.com*
[26] *http://www.technorati.com*
[27] *http://www.blogpulse.com*

usage and the fact that Flash content can include static graphics as well as animation/video, it is suggested that

5. Flash content should be distinguished by its type and stored along with animation/video content.

Open Graph (OG) and Dublin Core (DC) were among most frequently used metadata within the studied resources. The use of Microdata and Microformats has also been demonstrated to be sufficiently common. However, the standards adopted in the Blogosphere are evolving and are likely to remain in flux due to changes within the general Web. Given the importance of preserving the information encoded into blogs using various formats and metadata standards, it is suggested that

6. The data model should reflect the semantic structure of blogs and enable preservation of metadata, and
7. The data model should exemplify a common mechanism for interoperability of data across known or emerging standards.

The use of web feeds has traditionally been associated with blogs. Web feeds provide a simple mechanism for distributing blog content and can be used for aggregating, pulling or archiving blog resources. However, feeds are normally generated dynamically to reflect blog content. It is suggested here that

8. The data model should preserve information about the feeds generated on the collected blogs without archiving the feeds themselves.

The use of libraries and APIs has been demonstrated to be widely popular. JavaScript code and libraries were used in a prevailing majority of the cases (82%). The number of instances of JavaScript code as identified within a single resource was also considerably high (more than 8 in 46% of the time). It is therefore proposed that

9. The data model should preserve the source code as found appropriate.

The use of social media like Twitter, Facebook and Google+ in some cases interweave with the use of blogs. The use of Google+ was shown to frequent among the studied resources. The evidence of existing integration of services from third party social network services encourages identification of external profiles and the communication that flows through these services to be potentially informative. It is therefore suggested that

10. The data model should enable the storage of information about the third party accounts and profiles as well as the interaction activities they are associated with.

The use of the YouTube service was investigated as part of the study. The results suggest a wide popularity of the service. 10% of all the studied resources embedded YouTube content with an average of 3.51 YouTube instances per page. It is therefore suggested that

11. A mechanism should be put in place to enable the storage of information about the embedded video content.

The analysis of various media types demonstrates a wide variety of files used. However, the frequency of appearance of file formats (as shared within the studied resources) varies widely. Among the most popular formats are: PDF, DOC, MP3, MP4, and AVI. While it may not be possible to ensure efficient preservation for all of the formats, a mechanism for preserving the frequently occurring formats should be put in place. It is recommended here that

12. The data model should provide a mechanism for preserving different types of media within broad categories such as Document, Audio, Video and Image. Information about the formats should also be preserved along with the files.

Some of these recommendations corroborate the data collected from the survey as well as consultation inquiries conducted as part of this report. Further recommendations on development of the data model are derived from an inquiry into blog dynamics as discussed in the following section (6.3).

6.3 Inquiry into Blog Networks and Dynamics

In addition to the user survey, an inquiry into the theoretical and technological advances of network analysis and blog dynamics was conducted. Requirements for retrieving and storing data were deduced. The details of the inquiry are available in the Section 6.3 of the BlogForever: D2.1 Survey Implementation Report [1].

6.3.1 Requirements for a Weblog Spider and Data Retrieval

Blog spider software is an anticipated component of the solutions developed as part of the BlogForever project. The spider should be able to access blogs and extract the necessary data. Table 3 describes the elements that are important to perform network analysis for blogs and the Blogosphere, and annotations that should be taken into consideration for the development of the spider.

Firstly, the spider should be able to distinguish blogs and their individual components such as blog posts, comments, blogrolls and so on. Each of the posts should be retrieved with its permalink URI (often generated to reflect the date or title of the post). All the blog/post URIs have to be retrieved to enable the possibility of generating various networks.

The spider should be able to differentiate between different types of links citation, linkbacks and blogrolls. Furthermore, it is necessary to distinguish between external links (pointing to another blog) and internal ones (pointing to different elements of the same blog).

A different type of network can be generated by capturing comments and their inter-relations. Comments are primarily associated with a blog post or pages. When viewed as part of a discussion, comments can be posted as a response to a post or to other comments. It is important to enable capturing the interplay of the discussion that includes information about the poster and addressee.

To analyse changes in the structure of the Blogosphere, it is essential to obtain the date and time of each posting. The minimum requirement should include capturing the dates and times of creation of blog posts, comments, and linkbacks. Changes in the blogs should also be captured and recorded with relevant date-time measures.

Table 3 - Important elements and annotations for the spider development

Element	Annotation
Blog	Blog has a unique resource identifier (URI). Blog has at least one author.
Blog post	Blog post has a unique resource identifier (URI) that is normally composed of the blog URL with a specific extension. Blog posts have normally just one author.

Links	Differentiation between link, linkback and blogroll. Differentiation the link destination (e.g. blog, blog post, other part of a blog or other resource) Date of the post inferred from the link (if possible). For external links, distinctions made for website (homepage), single page inside the website or specific part on a page.
Embedded objects	URI of embedded/syndicated objects (e.g. embedded YouTube videos) should be captured.
Comments	Reference from comments to blog posts or to (where possible) previous comments.
Author	URI of authors where available. Additional information about authors and their relationships to others (e.g. FOAF data). Differentiation (where possible) between authors as real people and software robots.
Time/Date	Date/time of adding blog posts, comments and linkbacks. Other dated/timed events and changes. Date/time of crawling and capturing the data.
Tags and Categories	Tags and Categories where available. Hierarchy of categories captured if available.
Meta data	Relational meta data (e.g. XFN) Other semantic meta data that provides additional information. Meta data that helps identifying the author or other Internet representations of the author.
Context/Affiliation	Affiliation of a blog (e.g. blog portal, organisation).

Normally, the author of a blog post can be identified or at least marked with an ID so that blog posts of the same author can be found. For social network analysis, it is more important to know which activities belong to the same author than the identification of the real name or real life identity of an author. Therefore, aliases or pseudonyms can be used for identification as well, e.g. to distinguish the blog posts of different authors in a corporate blog. The identification of authors of comments is often more complicated because some blogs allow anonymous comments or guest comments. In these cases the author of a comment cannot be tracked. In some cases, a blog author has a name or label but it cannot be used as a unique id, and in other cases, the name or the picture of the user is connected to a login of a blog host or to a URL. Therefore, the spider should be able to capture the identities of authors of blog posts and comments as accurately as possible.

Blogs (like any other web pages) may contain various metadata. These should be captured because they can contain relational information (e.g. "known" element in FOAF) or information about the author that facilitates the identification of other relevant digital representations (e.g. Twitter or Facebook representation of the author).

6.3.2 Requirements for the Data Model

The data model should accommodate the network data collected by the spider.

The inner structure of blogs, including: blogs, blog posts and comments, should be preserved. Blogs and blog posts are usually assigned a URI. Comments, on the other side, do not have a URI. They can, however, relate to other comments posted within the same area. The inter-relations of

comments, blog posts and blogs in general can provide a basis for conducting social network analysis. Therefore, the following requirements to the data model are being formulated.

1. Elements such as blogs, blog posts, and comments should be differentiated from one another.
2. The URIs of blogs, blog posts and comments (where available) should be preserved.
3. Relationships between comments should be preserved.

There may be a number of various relationships from a blog to [a] other blogs or [b] other resources (e.g. web pages). To differentiate the relationships associated with a blog is highly important for conducting a meaningful network analysis. Hence, come the following requirements.

4. Links should be made explicit in the data model.
5. Links should be differentiated by several well defined types. At least, a distinction between blogroll, links to [a] another blog, [b] another blog post, [c] another web page, and [d] another resource.
6. Links should have a sender and a recipient.
7. External links to a web page or any other resource should be categorised as: [a] whole website, [b] single page on the website, or [c] a specific part of a single page. A flexible approach is necessary since other classification categories may be possible.

The names of the authors often appear next to a blog post or comment. In some cases, the author has a URI (e.g. the account in the blogging software). Nevertheless, a lot of people use aliases or guest accounts. However, it is necessary to preserve information on

8. Authors of blogs, blog posts and comments.

Authors, blogs, blog posts, comments and relations can have various attributes or properties. These attributes allow a categorisation or aggregation of the elements in an analysis. For example, an analysis can focus on blog posts written in English. In this case, the necessary property would be the language of the blog post. There are many possible properties. Some can be extracted directly (e.g. category of a blog post), others have to be derived from the content after additional processing (e.g. derivation of the subject of a blog post through text analysis). Hence, the following can be formulated.

9. A wide range of properties should be allowed by the data model.

The affiliation of a blog or an author can facilitate the identification and understanding of groups and subgroups in a social network. For example, blogs associated with a specific blog portal or the authors that belong to the same organisation can be in the interests of network analysis. Therefore, the following requirement can be formulated.

10. There should be an opportunity to add affiliations to authors and blogs.

Often, blogs provide additional functions for their readers to give feedback or to connect blog posts with other platforms such as Facebook, Delicious and Twitter. These external connections are useful for conducting an analysis that looks into the interrelation of the blogs with the resources beyond the Blogosphere. Hence, the following is suggested.

11. There should be affordances to add relationships to a blog or blog post which describe the external relations from other platforms to the blog or blog post. The character of such relationships can vary from the indication how many people have created a connection (e.g. how many “likes” the page) to meaningful statements about the blog post (e.g. a comment about the blog post via Twitter).

Blogs and the Blogosphere are changing permanently even if the intervals of changing vary greatly among the blogs. Date and time information is necessary to analyse dynamics in the Blogosphere and, therefore, should be included in the data model. Three types of events – publication (or creation), edition, and deletion should be captured with a timestamp. Therefore, the following requirements are formulated.

12. Elements of the data model like blogs, blog posts, comments, authors and links should each have a property for their initial creation and deletion.
13. Elements that can change over time like blogs or blog posts should each have an additional property indicating the date and time of the change. This property is envisaged as part of the versioning mechanism. For some elements (like links) date and time elements are not necessary.

7 Review of Blog Structures Based on Web Feeds

This chapter aims to investigate structures and data types distributed via blog web feeds. Investigation into the use of modules within web feeds can give an insight into developing a blog data model that enables blog preservation.

Web feeds, like RSS and Atom, have been widely used across blogging platforms and services. Represented in a machine readable format, web feeds enable data sharing among applications. The most common use of web feeds is to provide content syndication and notification of updates from multiple websites into a single application [21]. Aggregators or news readers are commonly used for syndicating web content. They allow users to subscribe to various web feeds and gain access to the content within the aggregator software. The simple mechanisms for accessing and distributing web content justify the wide adoption of web feeds by blogging platforms.

Employing feeds, blogging platforms are becoming capable of distributing the specified elements of published information. The distributed content is usually represented in an XML format and is structured to represent the data that are intended for syndication. The data channelled via web feeds are usually deprived of the original presentation layer or web pages published alongside blogs. However, while distribution channels often lack contextual information, it is considered acceptable to use the feeds for harvesting and preserving blog content [22]. The analysis of blog feeds can, therefore, shed light on understanding the structure of blogs and properties of their distributed elements. The analysis of web feeds, therefore, can inform the development of the blog data model by providing information about the metadata, attributes and data types used.

This chapter first analyses a small and carefully selected list of blogs, then progresses to scale the investigation to include a larger number of blogs from an external data source.

7.1 Small-Scale Study into the Use of Web feeds

7.1.1 Data Source

As part of this report, a set of web feeds have been collected and analysed in order to inform data modelling. A sample of blogs, identified as individually– and centrally – hosted, have been considered. The selection of blogs was based on ensuring a reasonable variety of hosting platforms and services, as well as diversity in the size of blogs, popularity and update frequency. For the initial analysis, a list of 23 manually selected blogs has been included in the initial screening (see Table 4 and Table 5). The snapshot of the feeds used for the study has been acquired and saved as XML files[28] to enable reproduction of the analysis if necessary.

Table 4 - Centrally-hosted blogs selected for analysis.

No	Service/Blog	Service Size	Software	URL	Blog	RSS/Atom File
1.	WordPress	Large	WordPress MU (OS)	www.wordpress.com	cfordphotography.wordpress.com	wordpress_the_ward_prism.xml
2.	TypePad	Medium	Movable Type (OS)	www.typepad.com	www.beerleaguer.com	typepad_beerleaguer.xml
3.	Squarespace	Medium	Proprietary	www.squarespace.com	www.uppercasegallery.ca	squarespace_uppercase.xml
4.	Blogger	Large	Proprietary	www.blogger.com	alpinebirds.blogspot.com	blogger_alpine_birds.xml
5.	LiveJournal	Medium	LiveJournal (OS)	www.livejournal.com	ontd-football.livejournal.com	livejournal_ontdfootball.xml

[28] *Collected web feeds are available at: http://blogforever.eu*

6.	Warwick Blogs	Small	Proprietary Blogbuilder 3.25	blogs.warwick.ac.uk	blogs.warwick.ac.uk/researchexchange	warwickblogs_phdlife.xml
7.	Wired	Small	WordPress	www.wired.com/blogs	www.wired.com/wiredscience/frontal-cortex	wired_frontalcortex.xml
8.	ScienceBlogs	Small	Proprietary	scienceblogs.com	scienceblogs.com/sciencepunk	scienceblogs_sciencepunk.xml
9.	Blog.de	Medium	Proprietary	www.blog.de	demokratievonunten.blog.de	blog_de_democracy.xml
10.	Mark Kermode's film blog	Journalism	Movable Type	www.bbc.co.uk/blogs	www.bbc.co.uk/blogs/markkermode	bbc_mark_kermode
11.	Guardian Money	Journalism	Proprietary	www.guardian.co.uk/money/blog	www.guardian.co.uk/money/blog	guardian_money
12.	OYNAGKI	Entertainment	Proprietary	pblogs.gr	oynagki.pblogs.gr	oynagki-phaistos.xml
13.	Survival Guide	Retail Information	Proprietary	pblogs.gr	genia700euro.pblogs.gr/	genia700euro-phaistos.xml

Table 5 - Individually-hosted blogs selected for analysis.

No	Service/Blog	Area	Software	URL	RSS/Atom File
1.	Empirical Zeal	Life Science	WordPress (OS)	www.empiricalzeal.com	empirical_zeal.xml
2.	Mind Hacks	Life Science	WordPress (OS)	mindhacks.com	mindhacks.xml
3.	Gizmodo	Technology	Proprietary	uk.gizmodo.com	gizmodo.xml
4.	Stephen Downes	Education/Technology	Proprietary	www.downes.ca	oldailycombined.xml
5.	Not Even Wrong	Physical Science	WordPress (OS)	math.columbia.edu/~woit/wordpress	not_even_wrong.xml
6.	tinsology.net	Computer Science	WordPress (OS)	tinsology.net	tinsology.xml
7.	e4innovation.com	Education/Technology	WordPress (OS)	www.e4innovation.com	conole.xml
8.	Art Blog UK	Art	WordPress (OS)	www.arts.co.uk/blog	art_uk.xml
9.	Boris Johnson	Politics	Proprietary	www.boris-johnson.com	boris_johnson.xml
10.	Steven Fry	General	WordPress (OS)	www.stephenfry.com/blog	stephen_fry.xml

7.1.2 Elements of Blogs as Represented via Web Feeds

The analysis of the blog feeds has been based on:

a) Parsing[29] the XML-based structure of RSS/Atom feeds
b) Extracting and listing XML nodes alongside the source
c) Identifying individual XML nodes and analysing their frequency
d) Looking for common components to inform the development of the data model.

Parsing the collected RSS/Atom dataset allowed identifying the list of nodes. The following nodes (see Table 6) have been extracted (excluding <rss> and <channel>).

Table 6 - List of all the nodes extracted from the collected RSS/Atom feeds.

List of Extracted Nodes	
<atom/10: id /link /updated>	<link>
<author>	<lj: journal/music/poster>
<category>	<managingEditor>

[29] JAR file and Java sources available at: http://blogforever.eu

<cloud>	<media: content /credit /description /keywords /thumbnail /title>
<comments>	<name>
<content/:encoded>	<openSearch: itemsPerPage /startIndex /totalResults>
<contributor>	<pubDate>
<copyright>	<published>
<dc: creator /date /identifier /subject /type>	<rights>
<description>	<slash: comments>
<docs>	<source>
<entry>	<subtitle>
<feed>	<summary>
<feedburner: browserFriendly /emailServiceId /feedburnerHostname /info /origLink>	<sy: updateFrequency /updatePeriod>
<generator>	<thr: total>
<guid>	<title>
<id>	<ttl>
<image>	<updated>
<item>	<url>
<language>	<webMaster>
<lastBuildDate>	<wfw: commentRss>

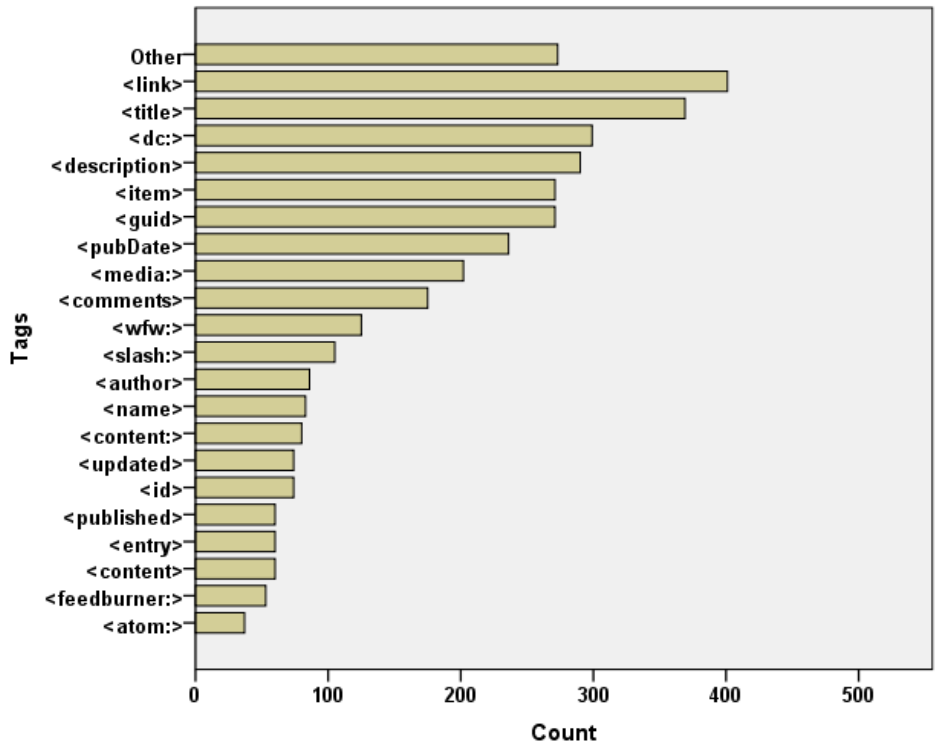

Figure 11 - Distribution of most frequently occurring nodes

The frequency across the identified nodes varied (Figure 11). Among the most frequent elements were `category`, `link`, `title`, `description` and `item`. The node `category` was 3.6 times more frequently occurring compared to the node `link`. The node `item` usually specifies blog entries. The nodes `link` and `description` can be used at various levels, for instance to provide a link to the parent blog or an external resource made available within a published blog entry. The frequently occurring elements overlap with the reviewed blog models, particularly in the parallels among tags such as `item`, `comments` and `author`. Hence, all of the frequently used nodes are to be integrated into the data model.

Modules within web feeds provide additional sets of elements that give feeds a greater level of expression. Most frequently observed feeds included `dc`, `media` and `wfw`. Each of those modules can provide metadata that are useful for blog preservation. Figure 12 demonstrates the distribution of the elements within the `dc` module.

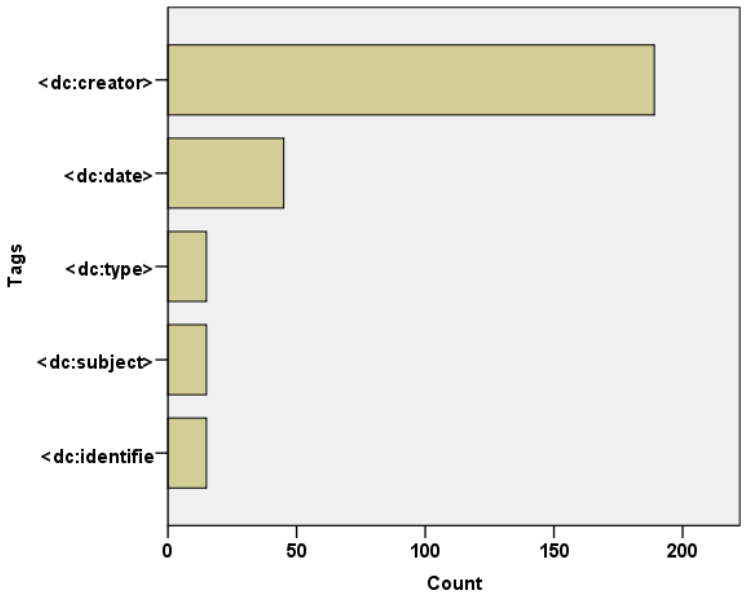

Figure 12 - Distribution of nodes for the DC module

While the initial study demonstrates the potential of acquiring insight into the data distributed via web feeds, it is necessary to conduct a wider study to draw conclusions relevant to the data modelling exercise. The following section (7.2) discusses extends the initial study.

7.2 Large-Scale Study into the Use of Web feeds

Similarly to the small-scale study described in the previous section (7.1), this study employed the same method. This study, however, considered a considerably larger dataset of web feeds.

7.2.1 Data Source

As the primary data source for this study, the 2011 ICWSM Spinn3r30 dataset was used. Spinn3r[31] is an American company that specialises in web crawling and indexing. The ICWSM 2011 dataset was published by the company for research purposes as part of the ICWSM data challenge. Access

[30] http://icwsm.org/data/index.php
[31] http://www.spinn3r.com/

to the dataset was requested on behalf of the University of Warwick to conduct the study described here and for future use.

The Spinn3r dataset contains 3Tb of data collected between 13 January and 14 February 2011. It covers a range of blogs, forums and social media. The data is categorised by its source, language and date. This study used a single slice of a data corpus that covers blog posts that were written in English and updated on the 13 of January 2011.

The following procedure was conducted for analysing a large number of web feeds.

1. The Spinner data was accessed and RSS/Atom feed located
2. RSS/Atom feed was retrieved and stored locally
3. Each feed was parsed to extract the XML nodes
4. The nodes, along with the feed file names, were stored in a comma delimited format.
5. Frequency analysis was performed to identify more widely used items and modules as observed in web feeds.

A total of 2,695 RSS/Atom feeds were analysed. The total number of nodes considered for the analysis was 713,434. The general pattern of the node in use indicates that only a small number of XML nodes are used widely. Figure 13 and Figure 14 outline the most frequently used ones. However, there is still a considerable amount of information delivered by the feeds that describes blog elements and resources.

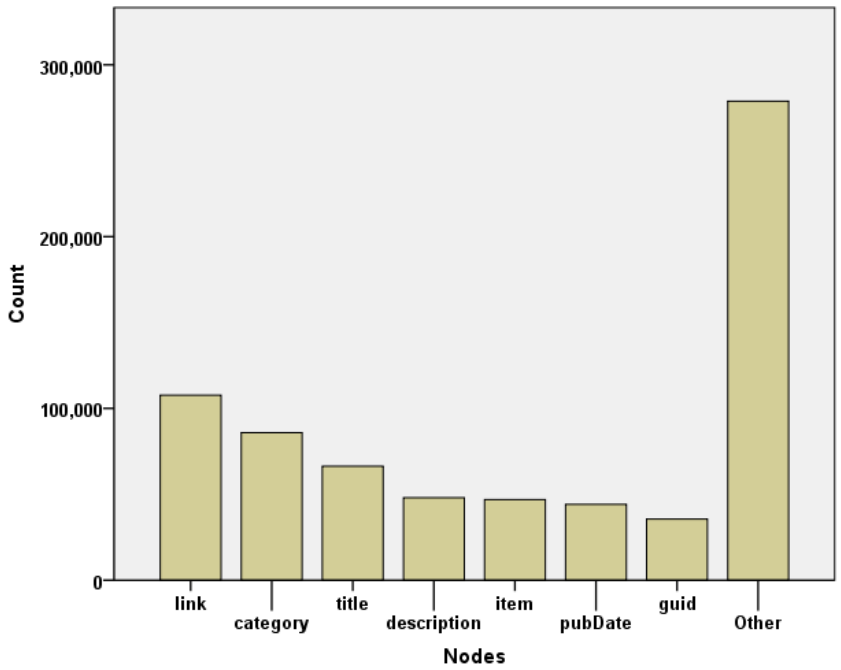

Figure 13 - Most frequently used nodes in web feeds (3% of cases collapsed).

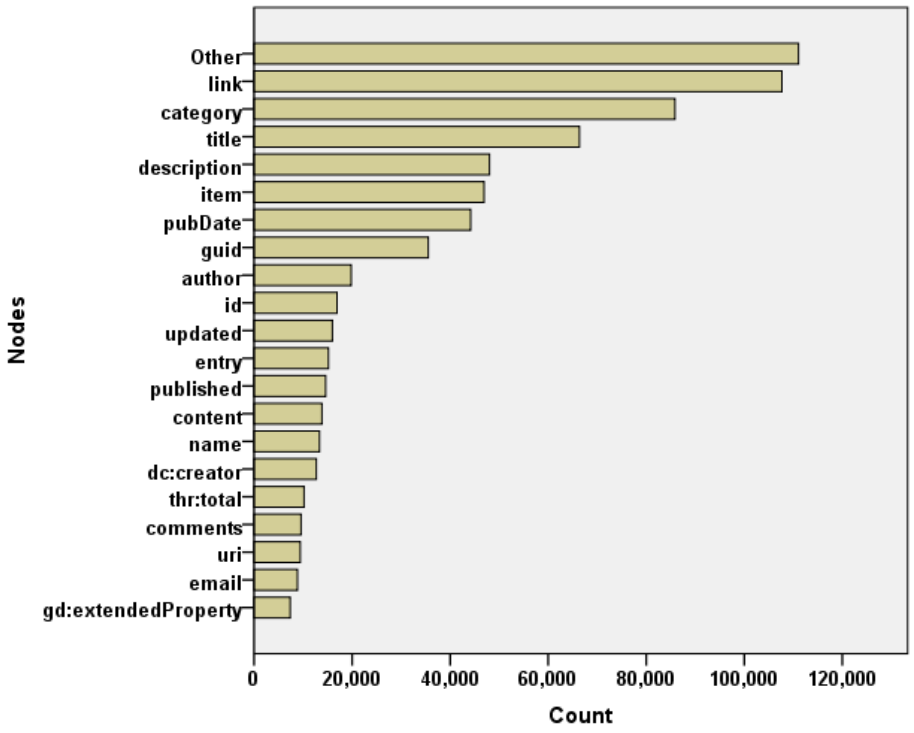

Figure 14 - Most frequently used nodes in web feeds (1% of cases collapsed).

This inquiry into the elements of web feeds identifies the data types that are commonly distributed by blogs. It would be reasonable to infer that the most frequent nodes should be centrally positioned as part of the data model. For instance, the nodes described in Figure 13 would require greater attention and a more central role in the data modelling exercise.

A considerably larger number of modules were identified within the web feeds studied. The total number of unique modules used was 80. The complete list of the modules is presented in Appendix A and Appendix B. Appendix A contains:

- ✓ List of Modules Used
- ✓ List of Modules and their Attributes
- ✓ List of all the nodes and the frequencies of their use

Appendix B contains:

- ✓ Frequencies of modules and their attributes (all nodes) sorted in alphabetical order
- ✓ Percentage of occurrences within all nodes
- ✓ Cumulative percentage of occurrences within all nodes

Among most popular modules (in addition to `dc` discussed in 7.1.2) identified within the RSS are `gd`, `media`, `wfw`, `content`, `slash` and `feedburner`.

- ✓ `gd`[32] is used as part of the Google Data API. The GD module can be used for working with YouTube videos, calendar events or many other data types. The 'extendedProperty' in particular allows storing a limited amount of custom data as an auxiliary property.
- ✓ `media`[33] is used to extend enclosures to handle media types, such as short films or TV. It is also used to provide additional metadata along with the media. The attributes identified to be used with the media module are: `adult`, `category`, `community`, `content`, `copyright`, `credit`, `description`, `group`, `keywords`, `player`, `rating`, `statistics`, `text`, `thumbnail` and `title`.

[32] http://code.google.com/apis/gdata/
[33] http://www.rssboard.org/media-rss

- ✓ `wfw`[34] is commonly referred as Well-Formed Web. It is used for enabling item-level commenting that can be delivered along with the posts.
- ✓ `content` is used to present a complete RDF description of the content as well as allow much richer content [21].
- ✓ `slash` was developed by the news site Slashdot. It provides information about the articles, number of comments acquired and so on.
- ✓ `feedburner`[35] is used for syndication purposes. Google's FeedBurner API is used but its further development has been stopped as of May 2011 [23].

The inquiry into the modules used by the blogs and their wide diversity suggests that greater attention should be devoted to understanding the types of data distributed via feed modules. Attributes specified within the frequently modules that describe the content should be taken into consideration when developing the data model. It is also necessary to investigate whether data distributed via web feeds can be acquired by crawling blog content.

In summary, the insight into the use of web feeds by blogs across the Web demonstrates that many blogs are currently being customised to distribute additional content and metadata via their feeds. Within the studied 2,695 feeds, 80 unique modules have been used. Many of the modules appear in web feeds rather frequently. More specifically, the descriptive nature of the attributes used by the feeds and the customising modules provides a better understanding of the constituent elements of blogs. Hence, this inquiry informs on the types and the popularity of specific media formats, the actions and changes highlighted within the feeds, and definitive objects like events and applications. It reveals the attributes that may not be frequently used but may become more widely adopted in the future. For instance, the `geo` module with its attributes denoting standard position is one of those identified. Furthermore, this inquiry highlights the different types of data and metadata that can be acquired from the feeds. To which extent the same data can be acquired via crawling requires further exploration.

While this study may appear to include a large number of feeds, the dataset is still relatively small, in absolute terms, to provide a coherent overview on the use of feeds across the Web. Furthermore, a more thorough analysis that includes the use of various attributes and modules is necessary. Yet, as part of the data modelling exercise, this enables developing a greater understanding of the data used as part of the blogs.

[34] *https://developer.mozilla.org/en/RSS/Module/Well-Formed_Web*
[35] *http://code.google.com/apis/feedburner/feedburner_namespace_reference.html*

8 Inquiry into Blog APIs

The aim of this chapter is to look into the APIs offered by the major blog providers and explore their technological basis for informing the development of the data model. The use of APIs is anticipated for capturing blog data[36]. The data model should, therefore, provide the necessary structures to store the acquired data. A comprehensive review of the available APIs remains outside of the scope of this document. However, a more detailed review of the available APIs and their potential use for data extraction will be discussed as part of the subsequent D2.6 Report (WP2).

8.1 WordPress Database APIs

WordPress API provides developers with a mechanism to interact, modify and customise the functionality of the system. WordPress provides a separate Database API for actions related to retrieval and storage of data. It enables developers to extend the system through plugins and reducing the amount of development necessary for accessing the database.

The Database API is subsequently categorised into:

- ✓ Options API[37]
- ✓ Transients API[38]
- ✓ Metadata API[39]

For the purposes of data modelling, an inquiry into the Options and Metadata API may be potentially useful, given that the database structures of WordPress.com may differ from that of the default WordPress.org. While WprdPress.org provides Open Source software that can be used for self-hosting a blog, WordPress.com provides a service registering a blog on a WordPress.com platform the database structure for which is not publicly available.

A brief review of the Metadata API shows that metadata elements are stored as simple key-value pairs. It also shows that metadata can be associated with users, posts and comments. The review of the API calls reveals that there is no specific type of data that can be associated with users, posts or comments. The system provides a general mechanism for recording any kinds of metadata as long as they are associated with any of the supported three concepts (i.e. users, posts and comments). Although, the inquiry into Metadata API does not inform on the possible data structures that can be expected recorded metadata, it corroborates the distinctive components that should be essential to the data modelling of blogs.

Similarly Options API shows that WordPress can provide a standardised mechanism for storing range of data within the available wp-options table. Unlike Metadata API, which requires the association of the metadata to certain elements of blogs, Options API remains open to a wide range of data (i.e. limited to 'long text') and use. Hence, rather than providing answers to the possible data structures that can be associated with blog elements, the inquiry into the WordPress APIs highlighted the challenge of dealing with open possibilities. It also suggests that providing general solutions for enabling storage of a range of data types might be useful.

Unlike, Metadata or Options API, the role of Transient API is to provide a mechanism for a temporal storage of data in the database if needed. It is assumed here that the use of Temporal API for informing the data modelling exercise is limited and, therefore, is not discussed.

An examination of Filesystem API[40] was found potentially useful for future use for informing on how to access and use the file system behind WordPress. Although this API will not directly inform

[36] *For details see: Grant Agreement Annex I - Description of Work (DoW), Task 2.3, p. 25.*
[37] *http://codex.wordpress.org/Options_API*
[38] *http://codex.wordpress.org/Transients_API*
[39] *http://codex.wordpress.org/Metadata_API*
[40] *http://codex.wordpress.org/Filesystem_API*

the development of the data model, it can shed light on possible ways of capturing files associated with the content of the specific post. Given the wide spread use of WordPress platform, an inquiry into the API can support development of data extraction methodologies. The Dashboard Widgets API[41] shows that WordPress developers are free to create dashboards of their choice. This demonstrates that the number of elements and components exhibited as part of a single blog can (hypothetically) be unlimited.

8.2 Blogger APIs

Similarly to WordPress API, the Blogger Data API[42] provides a mechanism for applications to view and update Blogger content. This means that third-party client applications can use the Blogger Data API to create new blog posts, and edit or delete existing blog posts. It is also possible to use the API and query for blog posts that match particular criteria.

For instance, a possible application of the API could enable making blog entries via email or making postings to multiple blogging systems. Currently, Blogger API also includes a backend for MS Word and other Office products, web-based blogging clients implemented in various programming languages, and platform-specific client applications [24].

The functionality of Blogger Data API includes the possibility of authenticating to the Blogger service, retrieving a list of blogs, creating posts, publishing blog posts, creating a draft blog post, retrieving posts, retrieving all blog posts, retrieving posts using query parameters, updating posts, deleting posts and implementing similar functions for comments. To inform data modelling an insight into the attributes used in retrieving posts and comments could be most appropriate.

The API specifications include the following data types that can be retrieved for each of the post: Entry ID, Entry Title, Entry Link, Entry Summary, Entry Content, Entry Author, Entry Category, Entry Category Scheme, Entry Publication Date and Entry Update Date. It is therefore necessary to align the data model to the API data structures.

Using an API for retrieving comments integrates the use of Atom Threading Extension (`thr` module). It can return information such as the resource associated with the particular reply (i.e. in_reply), total number of comments, publication date, update date or summary. API calls for authorisation are necessary to perform some operations. However, there was no specific Blogger API call identified for retrieving information about author. Retrieval of author details appears to be limited to the data acquired from retrieving posts (i.e. Author Name, Author Email and Author URI).

[41] http://codex.wordpress.org/Dashboard_Widgets_API
[42] http://code.google.com/apis/blogger/

9 Blog Data Model

The main approach for developing the data model was based on understanding the concepts that were identified as integral to blogs. It is evident that blogs are multi-faceted objects that may require a range of different data structures to be put in place. However, it is also apparent that most of the blogs share common features and a general outline. It is therefore possible to develop a generic and simple data model that could suffice the preservation of the basic components of the blogs. This basic model – referred here as the core model – can then be extended to ensure the integrity and authenticity of preserved blogs, satisfactory to the requirements of successful preservation and archiving.

9.1 Generic Blog Data Model

The main concepts of the blogs are captured in the following generic blog model presented in Figure 15. This model integrates the core components that derive from the available conceptual models, user perceptions of blogs and inquiries discussed in the proceeding sections. This data model enables storing information about the `Blogs`, `Blog Entries`, `Blog Authors`, `Pages`, `Posts` and `Comments`, as well as the `Content` they carry. The interrelation between the identified entities is shown and described by the connected lines. The small triangles indicate the directions of the relationships.

While the proposed generic model is sufficient to capture the essence of the blog, to address the requirement of preservation and the BlogForever project in particular, it is necessary to extend this model further.

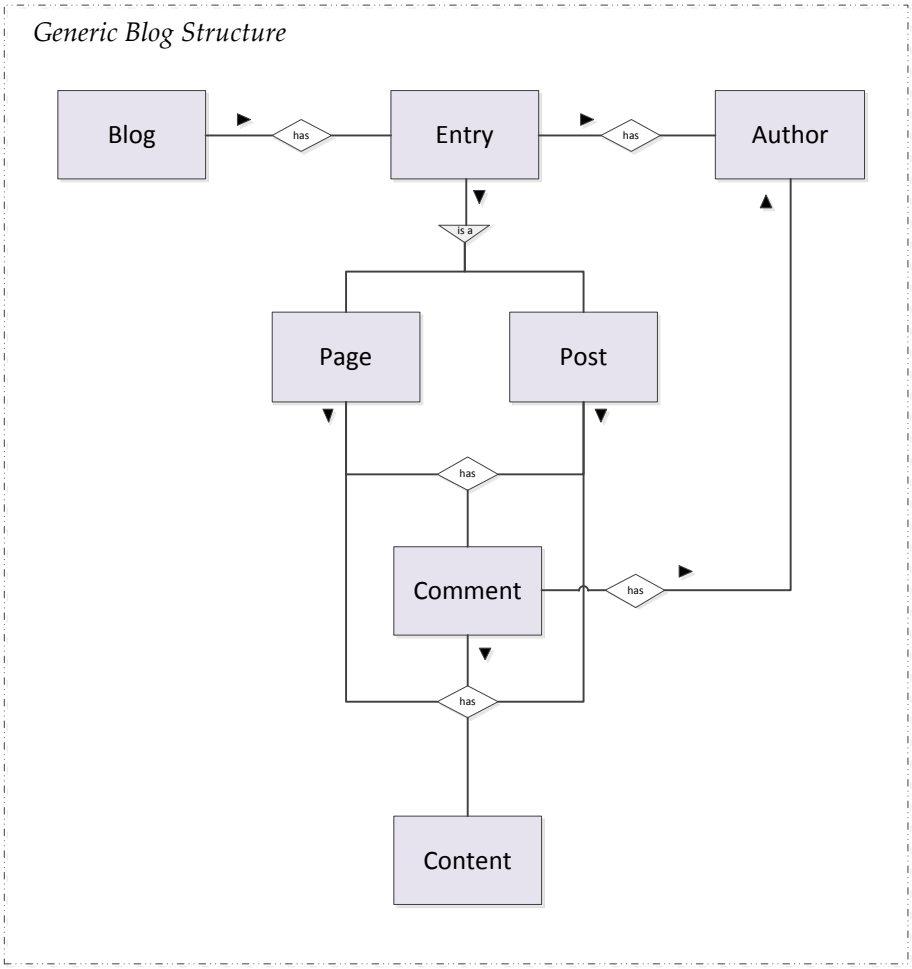

Figure 15 - Generic blog data model

The generic data model is extended by adding a set of entities that have been identified by the broad inquiry including (as described above) user survey, use of technologies, web feed data analysis and perspectives of network analysis. However, prior to extending the generic model, it has been decided to provide the extension by using a set of clearly marked out components that could be easily integrated into the proposed generic data model.

As a result, the extended conceptual model consists of the following components: `Web Feeds`, `Blog Context`, `Network and Linked Data`, `Community`, `Ranking`, `Category and Similarity`, `External Widgets`, `Crawling Info`, `Spam Detection`, `Semantics`, `Standards and Ontology Mapping`, and, finally, `Categorised Content`. A set of entities were then identified for each of the component and integrated into a larger model. The details are available in Figure 16.

The data model contains data structures that are either inherent to blogs or derived (calculated) based on the collected data. The conceptual model is colour-coded to distinguish between these structures. Dark green colour and dashed outlines are used to denote the entities that are intended to store data that were processed/generated and were not directly acquired from the crawler. Pink entities denote the data extracted by the crawler.

The remaining part of the chapter describes the components, entities and their attributes in greater detail. The detailed description of the proposed data model starts from listing the entities and attributes identified for the generic blog model (Table 7).

Table 7 - Data Specification for the Generic Blog Model

Entity	*Attributes*	*Description*
Blog	title	*Title of the blog*
	subtitle	*Subtitles of the blog*
	URI	*URI of the blog*
	status_code	*Status defines whether the blog ceased to exist*
	language	*Retrieved language field, as defined by the blog*
	charset	*Retrieved charset field, as defined by the blog*
	sitemap_uri	*URI of the blog sitemap if exists*
	platform	*Platform of the blog powering service, retrieved where available*
	platform_version	*Versioning information about the platform*
	webmaster	*Information about the webmaster where available*
	hosting_ip	*IP address of the blog*
	location_city	*Location city based on the hosting details*
	location_country	*Location country based on the hosting details*
	last_activity_date	*Date as retrieved from the blog*
	post_frequency	*As retrieved from the blog*
	update_frequency	*As retrieved from the blog*
	copyright	*Notes of copyright as retrieved from the blog*
	ownership_rights	*Notes of ownership rights as retrieved from the blog*
	distribution_rights	*Notes of distribution rights as retrieved from the blog*
	access_rights	*Notes of access rights as retrieved from the blog*
Entry	title	*Title of the entry*
	subtitle	*Subtitle of the entry if available*
	URI	*Entry URI*
	date_created	*Retrieved from the blog or obtained from the date/time crawling*
	date_modified	*Retrieved from the blog or obtained from the date/time crawling*
	version	*Auto-increment: derived version number (versioning support)*
	status_code	*Information about the state of the post: active, deleted, updated (versioning support)*
	geo_longitude	*Geographic positioning information*
	geo_latitude	*Geographic positioning information*
	visibility	*Information about accessibility of the post*
	has_reply	*Derived property (also SIOC)*
	last_reply_date	*Derived property (also SIOC)*
	num_of_replies	*Derived property (also SIOC)*
	child_of	*ID of entry parent if available*
Page	template	*Information about the design template if available and if different from the general blog*
Post	type	*Type of the post if specified (e.g. WordPress): attachment, page/post or other custom type*
	posted_via	*Information about the service used for posting if specified*
	previous_URI	*URI to the previous post is available*
	next_URI	*URI to the next post if available*
Comment	subject	*Subject of the comment as retrieved*
	URI	*URI of the comment if available*
	status	*Information about the state of the comment: active, deleted, updated (versioning support)*
	date_added	*Date comment was added or retrieved*
	date_modified	*Date comment was modified or retrieved as modified*
	addressed_to_URI	*Implicit reference to a resource*
	geo_longitude	*Geographic positioning information*
	geo_latitude	*Geographic positioning information*
	has_reply	*Derived property (also SIOC)*
	num_replies	*Derived property (also SIOC)*

	is_child_of_post	*Indicates information about the parent post*
	is_child_of_comment	*Indicates information about the parent comment*
Author	name_displayed	*Name of the poster as displayed*
	email_displayed	*Email address of the poster as displayed*
	is_anonymous	*Boolean property to indicate anonymity*
Content	full_content	*Content as extracted*
	full_content_format	*Content format (i.e. HTML, XML)*
	note	*Additional notes if available*
	encoding	*Information on encoding of the content*
	copyright	*Notes of copyright as retrieved from the blog*
	ownership_rights	*Notes of ownership rights as retrieved from the blog*
	distribution_rights	*Notes of distribution rights as retrieved from the blog*
	access_rights	*Notes of access rights as retrieved from the blog*

9.2 BlogForever: Blog Data Model

This section describes the details elaborated for describing the components of the blog data model proposed for the BlogForever platform. The higher level conceptual diagram is presented in Figure 16. The components of the data model that extend the generic one are denoted in the diagram by dashed lines and are labelled accordingly. Each of the components of the conceptual model is outlined in the following tables.

However, the extended model may require alteration according to the BlogForever project tasks that are expected to be completed at a later stage. More specifically, some changes may be expected as a result of progressing with WP3 that are expected to define preservation policies. Further changes may be necessary to ensure successful implementation and integration of the data model with Invenio software suite as part of WP4. Finally, some alterations may be necessary as a result of the WP2 task to define data extraction methodologies. Nevertheless, the use of the core model and a set of components that extend it provide the necessary foundation for possible alterations in the future.

9.2.1 Weblog Context

The entities described as part of the `Blog Context` component provide descriptive information about the blog and its elements in particular. It includes information about the selected presentation layer of the blog, description and keywords provided by the blogger, or the specific mark-up of individual elements of the blog. This type of information is usually provided by bloggers and may remain unnoticed by readers of the blog. However, the presentation layer of the blog, as well as the keywords and descriptions used by the bloggers to describe their work, may provide useful contextual information about the published content and about the blogger. The entities (Table 8) identified as part of this component enable storing information about the template/themes used by the blogs, CSS code with associated images that define the presentation layer, the visual snapshot of the blog, `<meta>` keywords and descriptions provided by the blogger, and structured metadata identified within the blog.

Table 8 – Data specification of the Blog Context component

Entity	***Attributes***	***Description***
Layout	theme_title	*Title of the layout theme*
	theme_designer	*Author credentials where available*
	theme_generator	*Generator of the theme where available*
	date_added	*Date added to the blog or date of crawl*

	date_updated	*Date of update or date of crawl*
	status_code	*Status of the theme: active, no longer in use (versioning)*
Layout_Stylesheet	URI	*URI of the CSS*
	file_path	*Relative path to the CSS file*
Layout_Image	URI	*URI of an image that is part of blog layout*
	alt	*<alt> text of an image*
	longdesc	*< longdesc > URI to the long description of the image*
	file_path	*Relative path to the file*
Expression_Meta	description	*Title of the entry*
	keyword_set	*Subtitle of the entry if available*
Structured_Meta	content	*Annotated tag*
	name	*Name of the annotation*
	property	*Value of the annotation*
	standard_description	*Description of the annotation*
Snapshot_View	format	*Format of the captured view of the blog (e.g. PDF, JPG)*
	file_path	*Path to the file*
	date_added	*Date of taking the snapshot*
	software_used	*Software used for taking the snapshot*

9.2.2 Web Feed

The `Web Feed` component (Table 9) consists of entities that are necessary to preserve information about web feeds of the blog. It includes a table to describe various types and versions of web feeds available (e.g. RSS1.0, RSS2.0, Atom, etc.), as well as a table to contain information about the feeds exhibited on the blog.

Table 9 - Data specification of the Web Feed component

Entity	***Attributes***	***Description***
Feed	URI	*URI of the feed*
	title	*Associated title (e.g. all posts, news)*
	format	*Format of the feed (e.g. RSS1.0, Atom, etc.)*
	generator	*Information about the software generating the feed (e.g. WordPress)*
	updated	*Date/time of the feed is last updated as retrieved*
	last_build_date	*Date/time of last built as retrieved*
Feed_Type	content_type	*Type of content distributed (e.g. posts, comments, etc.)*
	content_format	*Supportive information associated with content type*

9.2.3 Network and Linked Data

This component (Table 10) contains two tables that provide a mechanism for recording the necessary associations that may exist across the blogs. The combination of the following two entities provides a structure for recording *triples*. They can store information about the network of blogs, networks of authors, or any other associated relation that can be described in the entity called `Association_Type`. The data denoting the relationship are described as part of the `Association_Triple` entity.

Table 10 - Data specification for the Network and Linked Data component

Entity	***Attributes***	***Description***
Association_Triple	subject_id	*ID reference to a row described by Association_Type*
	association_type_id	*Reference to Association_Type Entity*
	object_id	*Format of the feed (e.g. RSS1.0, Atom, etc.)*

Association_Type	predicate_name	*Name of the association predicate*
	subject_entity_name	*Name of the associated subject entity name*
	object_entity_name	*Name of the associated object entity name*

9.2.4 Community

The `Community` component (Table 11) enables storing additional information about the active users – authors of posts and comments. It provides a mechanism for extending the information captured as part of the `Authors` entity (described within the generic blog model, see Section 9.1).

Table 11 - Data specification for the Community component

Entity	***Attributes***	***Description***
User_Profile	username	*Username of the profile*
	name	*Author name credentials where available*
	profile_uri	*URI to the profile*
	avatar_uri	*URI to the avatar file*
External_Profile	profile_uri	*URI to an external profile*
	profile_name	*Name registered on the external profile*
External_Profile_Type	profile_type	*Type of the profile*
	service_uri	*URI of the service where the profile is registered*
Affiliation	affiliation_name	*Organisation/Institution the author is associated with*
Affiliation_Type	affiliation_type_name	*Name of the affiliation of this type.*

9.2.5 Categorised Content

Categorised Content contains a number of entities that store the content collected from the blogs, which is decomposed into a number of smaller, but 'meaningful' pieces. The rationale for decomposing the acquired content is to provide the users of BlogForever platform with an ability to access selected types, of information (e.g. images, videos, PDF documents). Furthermore, it is necessary to break the content down and store it along with the multitude of relationships that may be identified across its various elements. For instance, unless the links are extracted from the acquired content and their targets are analysed, offering the required user service that demonstrates the network of links would require considerably more time and possible system overhead.

There is a generalisation within the module that combines the attributes of various multimedia resources when they are shared. The entity called Multimedia is suggested to contain attributes shared by the entities `Image`, `Video`, `Document` and `Audio`. The entities and attributes are summarised in Table 12.

Table 12 - Data specification for the Categorised Content component

Entity	***Attributes***	***Description***
Multimedia	URI	*URI of the multimedia resource*
	title	*Title of the resource*
	is_embedded	*Boolean value to indicate whether the resource is embedded*
	description	*Description of the resource acquired from the crawled data*
	geo_latitude	*Associated GEO positioning information where available*
	geo_longitude	*Associated GEO positioning information where*

		available
	creator	*Information about the creator where available*
	file_path	*File path to the media as stored on the disk*
	restriction	*Requires extension to specify age, country or technical restrictions*
	copyright	*Notes of copyright as retrieved from the blog*
	ownership_rights	*Notes of ownership rights as retrieved from the blog*
	distribution_rights	*Notes of distribution rights as retrieved from the blog*
	access_rights	*Notes of access rights as retrieved from the blog*
Image	format	*Image format*
	thumbnail_uri	*URI of the thumbnail associated with the acquired image*
	thumbnail_path	*File path to the thumbnail of an image as stored on the disk*
	height	*Dimensions of the image*
	width	*Dimensions of the image*
	additional_meta_i	*Additional columns to capture the necessary metadata for images as found necessary*
Video	codec	*Information about the codec of the video*
	format	*Format of the video file*
	duration	*Duration of the video*
	thumbnail_uri	*URI of the thumbnail image for the video*
	thumbnail_path	*File path to the thumbnail of an image as stored on the disk*
	resolution	*Information about the resolution of the video*
	additional_meta_i	*Additional columns to capture the necessary metadata for images as found necessary*
Document	format	*Format of the document file*
	language	*Language in which the document is written (candidate entity)*
	abstract	*Abstract of the document or excerpt*
	text	*The content of the document*
Audio	format	*File format of the audio*
	bit_rate	*Bit rate of the audio*
	duration	*Duration of the audio track*
	additional_meta_i	*Additional columns to capture the necessary metadata for images as found necessary*
Tag	tag	*Tag that was added by a user*
	language	*Language of the tag*
Link	title	*Title of the link if available*
	type	*Recognized link types as identified from the data*
	URI	*The value of the link*
	rel	*Recognised link relationship between resources*
	rev	*Reverse link relationship between resources*
Text	format	*Information on text formatting as extracted from documents*
	language	*Language in which the text is written*
	abstract	*Abstract or excerpt from the text if available*
	text	*Textual content*
	copyright	*Notes of copyright as retrieved from the blog*
	ownership_rights	*Notes of ownership rights as retrieved from the blog*
	distribution_rights	*Notes of distribution rights as retrieved from the blog*
	access_rights	*Notes of access rights as retrieved from the blog*
Event	name	*Name of the event as identified form the crawled data*
	location	*Location of the event (or compound address)*
	event_uri	*Main URI describing the event*
	date	*Date and time of the event*

	affiliation	*Organisation, companies, groups the event is affiliated*
	type	*Event type categorising the events (candidate entity)*

9.2.6 Standards and Ontology Mapping

The following three entities (Table 13) exemplify a mechanism for enabling the representation of stored blog data in specific standards, or for mapping it to certain ontologies. This mechanism proposes capturing and maintaining an up-to-date version of the standard or ontology.

Table 13 - Data specification for the Standards and Ontologies component

Entity	*Attributes*	*Description*
Ontology_Mapping	blog_entity_id	*Association with entity type as maintained in that is going to be mapped as maintained in BlogEntity table*
	ontology_class_id	*Association with the class of the ontology maintained in Ontology_Class table*
	ontology_property_id	*Association with the property of the ontology maintained in Ontology_Property table*
	mapped_entity_id	*ID of the relevant entity to be mapped (i.e. actual id to be mapped)*
	date_assigned	*Date mapped*
	status	*Status of the mapping*
Ontology_Class	value	*The name of the class from a specific ontology*
	child_of	*ID of the parent class if available (to enable nested structures)*
Ontology_Property	value	*The name of the property of a specific ontology*
	child_of	*ID of the parent property if available (to enable nested structures)*
Blog_Entity	value	*Name of the entity to be mapped (i.e. Blog_Entry)*

9.2.7 Semantics

The entities of this component contain only derived data. They provide necessary structures to store the results of some analysis into the semantics of the content. For instance, the results of the sentiment analysis (i.e. sentiment scores) conducted on a specific piece of content can be stored along with additional data describing the algorithm, its version and the status of the results association with the content. This means that the sentiment analysis results will remain in place even if new analysis with a new algorithm is being applied to the same piece of content.

The structures presented here intent to record data on `Sentiment Analysis`, `Content Similarity`, identified topics and keywords (Table 14).

Table 14 - Data specification for the Semantics component

Entity	*Attributes*	*Description*
Sentiment	positive_score	*Positive Score - one of the usual three scores of sentiment analysis*
	negative_score	*Negative Score - one of the usual three scores of sentiment analysis*
	neutral_score	*Neutral Score - one of the usual three scores of sentiment analysis*
	total_score	*Overall score (if available) as a result of the sentiment analysis*

	date	*Date of analysis*
	status	*Status of the analysis (to record the history of score changes)*
	algorithm	*Reference to the algorithm used*
	version	*Version of the algorithm used*
Blog_Similarity	algorithm	*Reference to the algorithm used*
	version	*Version of the algorithm used*
	score	*Derived score*
	date	*Date of the analysis*
	status	*Status of the analysis (to record the history of score changes)*
Topic	derived_topic	*Association with a specific topic*
	date	*Date of the analysis*
	status	*Status of the analysis (to record the history of score changes)*
	algorithm	*Reference to the algorithm used*
	version	*Version of the algorithm used*
Keyword	keyword	*Identification of the keywords from the content*
	date	*Date of the analysis*
	status	*Status of the analysis (to record the history of score changes)*
	algorithm	*Reference to the algorithm used*
	version	*Version of the algorithm used*

9.2.8 Spam Detection

The `Spam Detection` component (Table 15) provides a mechanism for storing information about the algorithms and tools used for detecting spam and flagging the content included in the repository. It includes three entities to flag and categorise spam.

Table 15 - Data specification for the Spam Detection component

Entity	***Attributes***	***Description***
Spam	flag	*Flag (score) associated after the spam detection has been applied*
	date	*Date of the conducted spam analysis*
	status	*Status of the analysis (to record the history of score changes)*
Spam_Algorithm	name	*Reference to the algorithm used*
	version	*Version of the algorithm used*
Spam_Category	category	*Category associated with the identified spam*

9.2.9 Crawling Info

This component is intended to store information about the process of crawling. This will allow storage of information about the way crawling was conducted for a specific blog or sets of blogs. Storing information about crawling will make it possible to explain any differences between data along with the development of the crawler (Table 16).

Table 16 - Data specification for the Crawling Info component

Entity	***Attributes***	***Description***
Crawl	authentication	*Authentication mechanisms used if any*

	date	*Date of the crawl*
Crawl_Type	crawler	*Reference to the crawler in use*
	crawler_version	*Version of the crawler in use*

9.2.10 External Widgets

External widgets make a fairly common appearance on blogs. They enable feeding external content or services through the blog. Despite their relatively wide use in the Blogosphere, only some of the data describing the widget are planned to be stored as part of the preserved blog data. These data will include the title of the widget, its size, service URI and other elements as described in Table 17.

Table 17 - Data specification for the External Widgets component.

Entity	***Attributes***	***Description***
Widget	title	*Title of the widget*
	URI	*Personalised URI of the service*
	height	*Height of the widget as it appears on the blog*
	width	*Width of the widget as it appears on the blog*
Widget_Type	name	*Specify the name of the provider associated with the widget*
	type	*The type (category) of the widget*
	provider_uri	*The URI of the service provider (e.g. Twitter, Flickr)*

9.2.11 Ranking, Category and Similarity

All of the entities described as part of this component (Table 18) are derived as a result of analysing captured blogs. These structures enable storing information about the ranking of blogs, or assigning them to certain categories.

Table 18 - Data specification for the Ranking, Category and Similarity component

Entity	***Attributes***	***Description***
Ranking	score	*Ranking score of the blog*
	date	*Date the ranking has been assigned*
	status	*Status of the assigned ranking*
	algorithm	*Algorithm used for ranking (candidate entity)*
	algorithm_version	*Algorithm version*
Category	value	*Derived category associated with the blog*
	date	*Date of associating a category*
	status	*Status of the category*
	algorithm	*Algorithm used for deriving the category (candidate entity)*
	algorithm_version	*Version of the algorithm*
Blog_Similarity	score	*Derived score of blog similarity*
	similar_to_blog_id	*Associated blog to which the blog is similar to*
	date	*Date of the assigned score*
	status	*Status of the assigned score*
	algorithm	*Algorithm used for deriving the score (candidate entity)*
	algorithm_version	*Version of the algorithm used*

The components, entities and attributes described above can be integrated into a single model. The following two diagrams offer a visual representation of the conceptual model (Figure 16) and the mode detailed logical model (Figure 17). The conceptual model is limited in the number of

attributes or some of the supportive entities presented to provide better readability. The logical model, on the other hand, describes the entities, attributes and cardinality as identified at this stage, and omits the listing of primary and foreign keys for the purposes of readability.

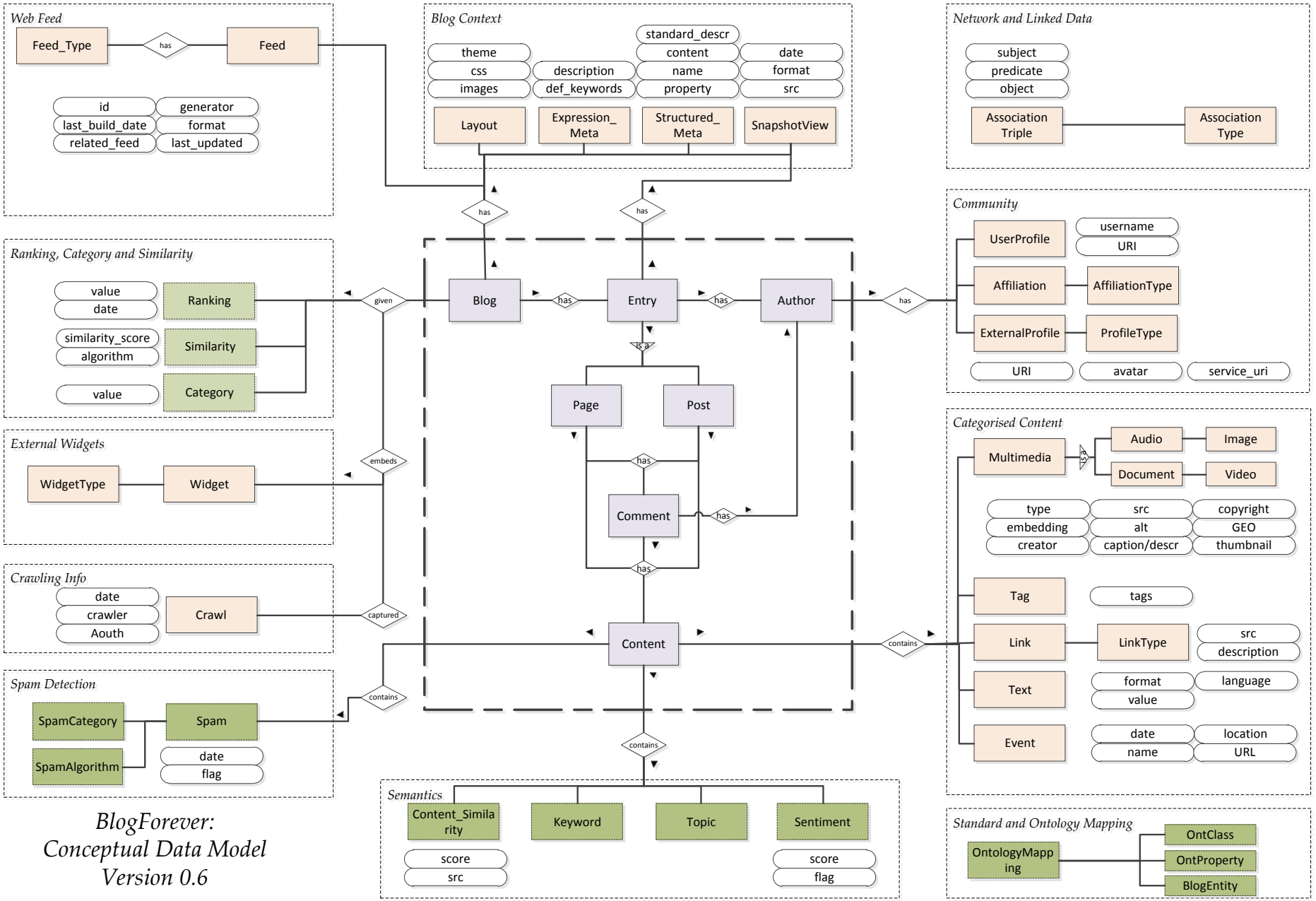

Figure 16 – Conceptual data model for blogs.

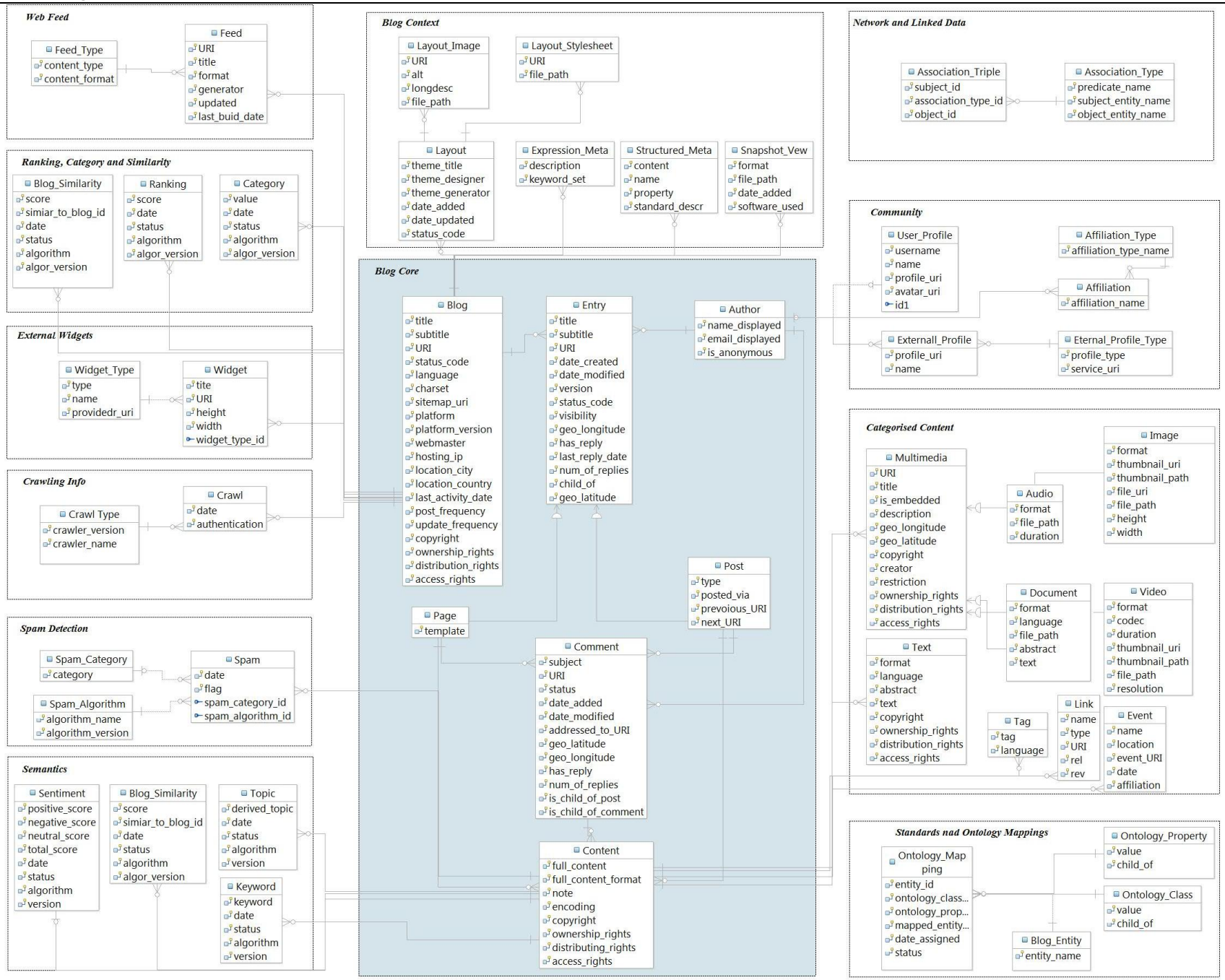

Figure 17 - Logical data model for blog preservation

10 Instantiation of Blog Data Model

This section discusses the use of the proposed Blog Data Model (Chapter 9) as part of a larger repository system. Integration of the blog data model with the repository software will enable providing the necessary services anticipated from the BlogForever platform.

10.1 Process and Data Flow

To discuss how the blog data model relates to the encompassing solutions developed by the BlogForever project, it is necessary to outline the anticipated system. The high level description of the system is described in the grant agreement[43]. The high-level architecture and workflow of the system are available in Appendix C. Building on the initial requirements design, this report outlines the anticipated processes and data flow as part on the operation BlogForever repository.

Figure 21 in Appendix C describes the primary processes that take place in an operational blog repository. It demonstrates the components of the system and outlines the flows of information between them. It highlights the user base of the system and the ways for accessing the repository. Within the boundaries of this report, however, it is necessary to define the flows of data between the various components of the system. Data flow diagrams can add the necessary details to the system description.

Data Flow Diagrams are among the most traditional modelling methods. They offer a simple graphical representation of a flow (and data flow in particular) across system components and interfaces. The elements of a data flow diagram comprise the following simple notations:

- ✓ Arrows – data flow
- ✓ Circles – data transformation
- ✓ Horizontal lines – data sources
- ✓ Rectangles – external entities

The data flow represents information or material exchanged between two transformations. Each circle defines the basic functionality provided by a system component. The representation of the information flow can be captured on different levels. Context diagrams are the top-level models that show external systems interacting with the proposed system [25].

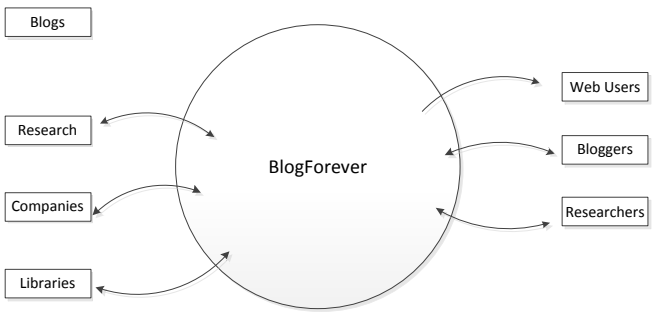

Figure 18 - BlogForever Data Flow: Context Diagram

The context diagram is usually decomposed further to represent data flow at a lower level as shown in Figure 20 and Figure 21. The dashed lines represent the boundaries of the BlogForever platform.

[43] *Grant Agreement Annex I - Description of Work (DoW), see Part B.*

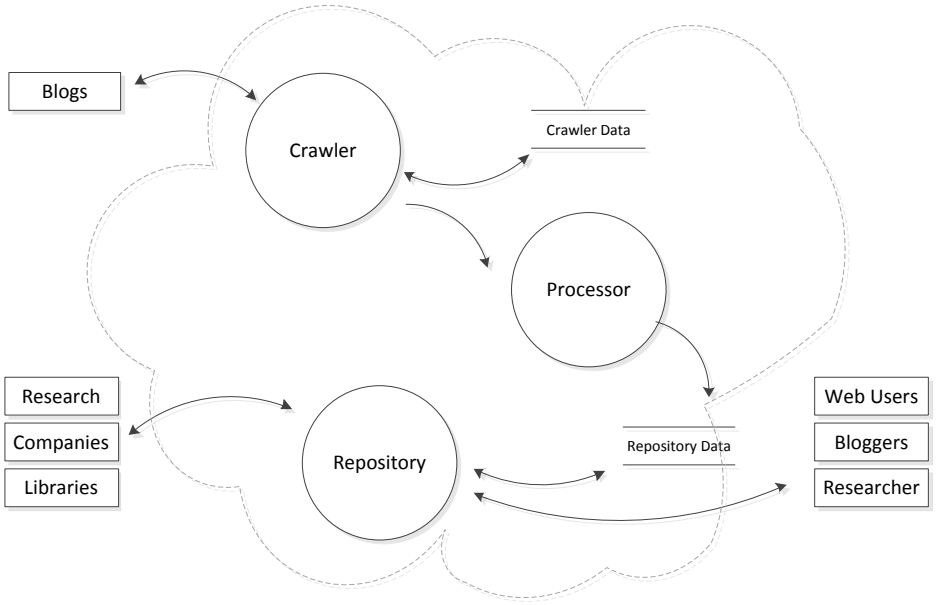

Figure 19 - Data flow diagram at level 1

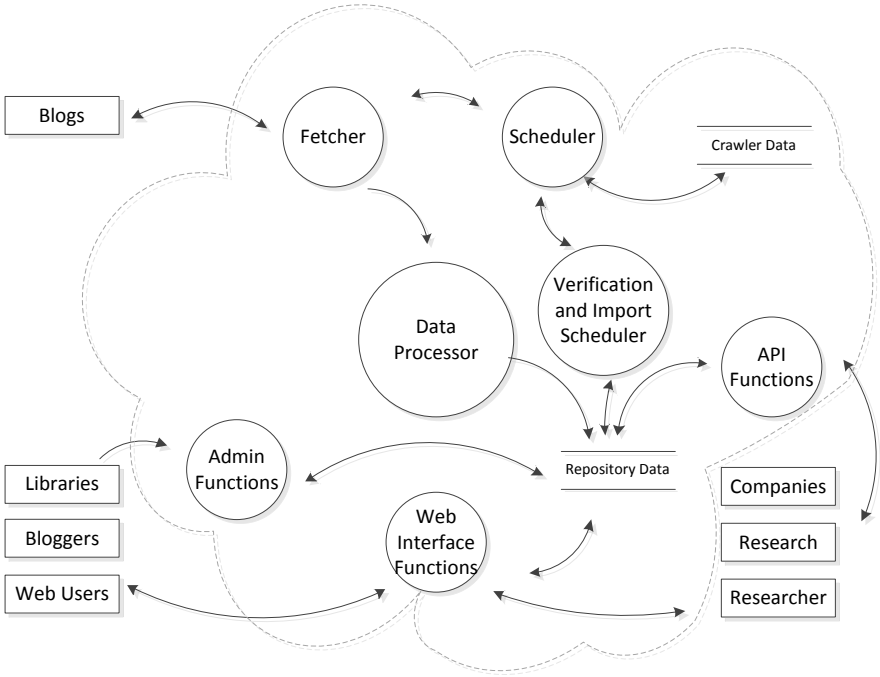

Figure 20 - Data flow diagram at level 2.

The data flow diagram at level 2 (Figure 20) can be represented at a lower level to include the details processing the data acquired from the crawler. Although this report focuses on the aspects of the necessary data structures, further decomposition to capture the transformations within the Data Processor remains outside of the scope of this report and will be addressed as part of WP4. This report, however, does attempt to make suggestions for implementation in terms of data structures.

10.2 Invenio and Blog Data Model

Achieving the aims of this project that revolve around solutions for preserving, managing and disseminating blogs requires a carefully designed software application. This software should

provide a stable, effective, efficient and user friendly experience for working with blog data. Rather than developing a repository system from scratch, BlogForever is intended to use the widely-adopted Open Source Invenio[44] software suite as a foundation.

Invenio is a digital library system developed at CERN to maintain its document server. The project originated over 15 years ago and has sufficiently matured through a number of development cycles. Invenio is available under GPL2 Open Source licence and is based on openly available Apache/WSGI, Python and MySQL. Invenio has a modular design that makes it easier to adopt the system to various needs and requirements. It can be used as a digital object repository or a fully functional digital library [26].

It makes sense to adopt the system within the context of BlogForever. The rationale behind using Invenio is twofold.

1. Repository Management: the system is designed and is shown to be capable of providing the functionality that is necessary for digital repository management and use.
2. Interoperability: the system is based on open standards of MARCXML and OAI-PMH 2.0 to support interoperability with other digital libraries.

The functional features of Invenio extend to include collaborative tools, fast search and customisable workflow engines[45]. As an integral part of a repository management system, the BlogForever platform can benefit from the tools that allow users to discuss, arrange and manage blog resources collected and stored in the system. Search functions combine metadata and full text search in a query language that is similar to using Google. On the other hand, Invenio already implements standard protocols to support the inclusion and dissemination of content. Invenio supports any metadata format in OAI-PMH due to the underpinning layers that support custom conversion templates [26]. Therefore, it is assumed that interoperability and repository management will be possible by reusing the readily available functionality.

Despite the fact that Invenio offers a powerful solution for repository management and use, it is not designed to work with dynamic and continuously evolving digital object such as a blog.

The database system of Invenio can enable storage, versioning, searching and personalised access to various types of digital objects. However, to address the requirements of the project and preserve the elements of the blogs as identified in the proposed data model, changes within the current Invenio software suite might be necessary. The data model proposed in Chapter 9 can be used to integrate necessary database components into the current structure of Invenio. It is proposed here to provide a mechanism that makes it easy to maintain database structures according to the changes within the dynamic space of the Blogosphere and document these accordingly.

[44] *http://invenio-software.org/*
[45] *See papers available at http://cdsweb.cern.ch*

11 Consultation on Technical Implementation

This chapter outlines the review of the proposed data model and consultation for its improvement from one of the BlogForever project partners Phaistos Networks. Phaistos Networks offers a blogging service via the PathFinder Blogs[46] blogging platform.

The consultation exercise has been conducted by comparing the elements of the proposed data model with the existing data scheme of the PathFinder Blogs platform. While the consultation did not identify irregularities, suggestions for additional entities and properties were made. The suggestions were primarily concerned with addition of attributes associated with entities describing blogs and blog entries. It was suggested that the storage of statistics data for blogs and blog entries, as accessed within the BlogForever platform, should be enabled. Some of the data types suggested here appeared in the compared database at a later stage after receiving feedback and requests from users of the platform.

Suggested extension to the data model is described in Table 19. The table includes an additional entity – Preservation_Options – that describes some of the elements describing presentation aspects of blogs. Since some of the proposed attributes rely on the possibility of acquiring the data by crawlers, the adjustments to the data model will be made after finalising the data extraction methodologies addressed as part of the D2.6 deliverable (WP2).

Table 19 - Suggested extension to the proposed blog data model

Entity	***Attributes***	***Description***
Blog_Settings	date_header_format	*Format of the dates used in the blog*
	time_format	*Format of the time*
	items_in_main_page	*Number of items displaying in the main page*
	items_in_days	*Display items of the last N days*
	use_popup_for_comments	*Enable use of popup in comments*
	enable_date_archive	*Set if a date archive will be displayed*
	enable_trackbacks	*Set if the blog can receive trackbacks*
	enable_comments	*Set if the users will be able to leave comments*
	enable_site_feed	*Set if an RSS feed exists*
	moderate_comments	*Set if the author have to approve the users comments*
	moderate_links	*Set if the blog can receive pingbacks*
	rss_max_items	*Max number of posts that an RSS feed contains*
	email_alert_on_comment	*Send email alert to the author when users leave a comment*
Presentation_Options	display_tags_sort_alpha	*Set if tags will be displayed and in lexicographical order*
	display_tags_cloud	*Set if a tags' cloud will be displayed*
	display_avatar	*Set if the author's avatar will be displayed*
	display_profile_link	*Set if a link to the authors' profile will be displayed*
	display_social_buttons	*Set if social buttons will be displayed*
	display_comments_emoticons	*Set if emotions will be displayed in*

[46] http://pblogs.gr

		comments
Entry	scheduled_publish	*Declare if this is a scheduled entry*
	allow_comments	*Declare if comments are allowed in this entry*
	allow_pingbacks	*Declare if this entry can receive pingbacks*
	trackback_url	*Trackback URI to send*
	flags (sticky post)	*Set if it is a sticky post*

Furthermore, the consultation exercise proposed data structures necessary for preserving data about user access and use of materials within the BlogForever platform (Table 20).

Table 20 - Data structures for storing user interaction within the BlogForever platform

Entity	***Attributes***	***Description***
Stats_Countries	country	Country name
	hits	Number of hits for this blog
Stats_Hits	hits	Number of hits for this entry
	dt	Date
Stats_Referrers	referrer	Referrer URI
	hits	Number of hits
	dt	Date
	hit_type	Type of hit (e.g. direct, web search, image search, video search, blog search)

It is necessary to clarify that the statistics highlighted in the proposed structures does not intend to capture the statistics (e.g. countries, hits, or references) as registered on the original blogs. The limitations for acquiring the statistics information from the Web resources are well known. Instead, it is suggested to ensure that the Invenio software suite supports logging and preservation of the associated usage statistics.

12 Consultation on Preservation of Blogs

This chapter summarises an internal consultation exercise that was led by ULCC[47] to provide commentary on the proposed blog data model from the perspective of digital preservation. It is written as an elaboration of the original section 2.1.2, and intends to define more closely those aspects (or dimensions) of blogs which are intended to be preserved.

There is also a possibility that this chapter will, after further discussion, be taken forward to WP3 to assist with the definition of preservation, and the description of the repository's preservation actions. For the latter, it is expected that the data model will be shown to relate closely to the OAIS model[48], either through extension of the existing data model, or through showing it as a detailed SIP in relation to the SIP-AIP-DIP [49] workflow, or through other methods.

12.1 Four Aspects that Need Preservation

According to the stated aims of the project[50], BlogForever will be able to capture:

- ✓ the dynamic and continuously evolving nature of blogs,
- ✓ their network and social structure, and
- ✓ the exchange of concepts and ideas that they foster.

A blog, as a conceptual object, can be represented as a continuous stream of published information or a stream of user activity. BlogForever aims to preserve four principal aspects of the blogosphere: (1) Content, (2) Functionality, and (3) Context, all of which in combination will aim to deliver the (4) Experience of the Blogosphere.

There is overlap between these aspects. One example is `Categorised Content` in the data model, where particular digital objects such as audio and visual files contain *content*, but also rely on specific software-dependent *functionality* in order to deliver that content. Another example is `Semantics`, where useful elements such as keywords and topics provide *context* to the blog, but are dependent on the use of blog *functions* to generate them (e.g. the functions described by WordPress[51]).

WordPress blogs also use functionality allowing users to create tags and categories which are stored in a WordPress database, and are explicitly exposed on the blog as contextual information. This contrasts with the earlier method of HTML markup where `<meta>` tags were manually inserted in the header section of pages.

In short, these aspects need to be considered, not in isolation, but in various combinations - in terms of how they help the project meets the requirements of digital preservation. Do we need to preserve the look and feel of the blog? And if so, should this be done in combination with the content, or separately, or both?

12.1.1 Content

Content is a term attributed to items such as text, images, and media. This content appears as pages and posts, and attachments to same, and will include links – links which serve to navigate to other parts of the blog, and links to other (external) targets. This is why preservation of blogs can appear difficult, as websites, they are likely to contain multiple file formats.

[47] http://www.ulcc.ac.uk/
[48] *For an introduction to the OAIS model, see for example http://www.oclc.org/research/publications/archive/2000/lavoie/*
[49] *Respectively, the Submission, Archival and Dissemination Information Packages in OAIS.*
[50] *BlogForever Description of Work document, project summary abstract.*
[51] *See for example http://codex.wordpress.org/Function_Reference*

> *Every Webpage has a digital manifestation. It consists of at least one file or bytestream which can be interpreted and rendered to show the actual content of a Webpage. A digital manifestation may comprise several files. HTML based Webpages comprise of an html page, all referenced image files and Cascading Style Sheets containing important rendering information. An information system, such as a web browser, needs all those components to render a Webpage properly [27].*

In terms of the data model elements (see Figure 16), content comprises *Blog*, *Entry*, *Page*, *Post*, and *Comment* and *Content*. It is also *Multimedia* (audio, document, image and video), *Tags*, *Links*, *Texts*, and *Events*. *Web Feed* also contributes to content; *Spam Detection* also creates content.

For digital preservation purposes, a lot of this content depends on formats and encoding. There is a tendency among archivists to focus on individual file formats.

> *File formats encode information into forms that can only be processed and rendered comprehensible by very specific combinations of hardware and software. The accessibility of that information is therefore highly vulnerable in today's rapidly evolving technological environment. This issue is not solely the concern of digital archivists, but of all those responsible for managing and sustaining access to electronic records over even relatively short timescales [28].*

Archivists try to identify formats, and ascertain whether they are supported; they consider their significant properties, which are:

> *The characteristics of digital objects that must be preserved over time in order to ensure the continued accessibility, usability, and meaning of the objects, and their capacity to be accepted as evidence of what they purport to record [29].*

It is also important to think about how to render these individual digital objects in the future. This is what is called the "“performance model”, where it is possible to render content and significant properties of a resource, regardless of what software was used to create it.

> *The performance model breaks down the concept of a digital record into components that help explain their fundamental nature [9].*

Hence the need for some sort of preservation strategy, migration is a common one. This involves migrating the content of a resource on to a new file format, where the target format is one that can be trusted and known to be supported and rendered. In this way, the content keeps “performing”.

> *The migration of digital information from one hardware/software configuration to another or from one generation of computer technology to a later one, offers one method of dealing with technological obsolescence [30].*

Copies of the original blog files from a harvest will be kept in the repository, but active preservation means that at some stage BlogForever expects to make copies of those files and undertake migration (or some other chosen method). So the preserved blog will ultimately comprise copies of files, and may be rendered in software different to the original software. When expressed like that, “authenticity” might be a misleading term, but as a repository we must ensure that (a) we have preserved all the files and a means of rendering the files, i.e. done all that is necessary to maintain the “performance” of the blog, (b) our preservation actions have not compromised that process in any way, and (c) that the repository content is secure against accidental corruption.

Metadata modelling will play a part in preserving this content. The project will develop requirements for administrative metadata, which includes the technical metadata of certain

identified file formats[52], and preservation metadata, which can be used to manage and record repository actions within an OAIS-compliant repository environment. "*Preservation metadata is information that supports and documents the process of digital preservation.*"*[31]*.

12.1.2 Functionality

Scripts, behaviours, the structural integrity of the blog, the internal links of the blog, the ability to navigate around the blog – all of these are functional elements which concern the integrity of the blog as a structured block of information.

In terms of the data model, **functionality** includes the Blog Context, particularly aspects such as Layout and Structured Metadata (see Figure 16). Within the Categorised Content block, Links, Tags and Events can be understood in terms of their functional behaviours, how they affect the blog's navigation and the user's ability to retrieve information from the blog. Widgets and Web Feeds are part of the blog's functionality. Other important elements of the Categorised Content, particularly multimedia files, will have functions and behaviours we wish to preserve and reproduce.

The project is willing to preserve and replicate as much of the original blog behaviour and functionality as possible. It remains to discover the extent of possibilities for replicating things like Flash animation or JavaScript elements in blogs, and for continuing to replicate their performance in a preserved version. Repository decisions, including choice of storage format, choice of delivery method, file format preservation actions, file migrations and so forth can all have an affect on the overall functionality.

To put this in perspective:

> *Some informed decisions will therefore be needed regarding the best way to provide future access, not just access to objects on an individual basis but access to the web archive as a functional whole. Knowing what worked for access in the past is likely to reduce some of the guesswork in formulating appropriate strategies for ensuring ongoing access to material collected in the past…Checking whether content collected in the past still functions adequately when new browsers are taken up or support changes for particular formats, may not be seen as a high priority, particularly as long as only a relatively small proportion of content is noticeably affected or accessed [32].*

12.1.3 Context

BlogForever should ideally add value to the preserved objects by also preserving the original intellectual **context** in which the resources were created. Context is often described recursively as components of information that were created to offer contextual information about an object. Context may easily become the target object [33]. Nevertheless, preservation of context should be made an important part of preserving blogs. This can be achieved by preserving the harvested descriptive metadata, and through curation and cataloguing, adding to the BlogForever collection, with descriptions, ontologies, dates, histories and interpretation which help explain the content to the user community. The users won't simply be querying a gigantic database of information which serves them blog content without some form of contextual explanation.

In terms of the data model, **context** includes any descriptive metadata added by the authors and users, among the principal elements are Meta (description and keywords), and Keywords, and Topics. Context also includes aspects of the blog that show how the original blog related to the

[52] See for example http://www.loc.gov/standards/mets/METSOverview.html#admMD

community of bloggers. Ranking, Category and Similarity provide that context, so do User Profile, Affiliation and ExternalProfile in the Community block. Contextual information about the Author of the blog also adds descriptive value.

The user community may also wish to research the scholarly content of the blogosphere in context. They will want to follow links and citations between blogs [e.g. 34], verifying references, tracing the lines of thinking and influence. It is a form of network analysis that the BlogForever repository must ultimately make possible. Therefore BlogForever is concerned with aspects of persistency of links [e.g. 35] and continuity[53]. **Context** thus also includes Semantics, and Ontology Mapping, and the Network and Linked Data (see Figure 16). All of these provide a history of contextual information very close to descriptive metadata that helps to interpret the resource to the user community. This would go some way towards the project goal of preservation of "exchange of concepts and ideas" and the "network and social structure".

The Crawling Info is also contextual, especially from the point of view of repository management. In terms of web crawling, each interaction a harvester makes with a server is potentially unique. The dialogue of request and response can be part of the contextual story, yet (in some other web archiving initiatives) this information is not always preserved[54]. BlogForever should aim to document the context of each interaction and request type.

12.1.4 Experience

All of the above in combination will offer a preserved version of the Blogosphere. This can be perceived as the entirety of the Blogosphere **experience**; it is for the project to consider how much of that experience should reasonably be offered to the user community through the BlogForever repository.

To put this in perspective:

> *Decisions about which strategies are appropriate for individual archives will be influenced by the mandate of the particular organisation and its anticipated use cases. In some cases access to individual objects may be sufficient for a particular use. In other cases being able to render the object, page or site within the temporal and structural context of the page, site, or wider archive may be important: or even being able to relate it back to the live web. Maintaining navigability at least within the archive before and after transformation (without losing the structural and temporal context) may be a desirable outcome. For some users, identifying and analysing information about aspects of web design and use of particular formats in the development of individual sites or across the whole archive contents over time may be of interest.* ***Other users may wish to use the archive as they could have experienced those sites when they were part of the live web*** *(as far as possible given the limitations imposed through the harvesting process and selection decisions).*(emphasis added) [32].

Other web-archiving initiatives have often focused on selected websites as discrete "titles" or items of accessioning, and tend to preserve and curate each archived "title" in isolation working to a very selective policy, often following existing standards for library accessioning. For example, the PANDORA web archive of Australia states:

[53] For a description of The National Archives' approach to Continuity for the gov.uk domain, see http://www.nationalarchives.gov.uk/information-management/policies/web-continuity.htm

[54] This idea is part of the *IIPC Web Archiving Metadata Set* as proposed by Julien Masanes in 2005. http://iwaw.europarchive.org/05/masanes2.pdf

> *Each item in the Archive can be fully catalogued and therefore can become part of the national bibliography…In the Library's own catalogue web resources are integrated with all other resources.*[55]

PANDORA works to selective guidelines, which state:

> *Each of the PANDORA participating agencies selects titles for the Archive according to its own selection guidelines…The National Library aims to archive titles of national significance, while the State libraries aim to archive those of State and regional significance.*[56]

BlogForever however is not working to a selective model and its scope is more broad-based:

> *A multitude of parties will benefit from the project, including libraries and information centres, museums, universities, research institutes, businesses, and bloggers. The BLOGFOREVER partners will combine and utilise multidisciplinary skills, expertise, and ongoing work in the fields of weblogs analytics, web semantics, social networks, and online preservation. Academic partners will study weblog semantics and the social importance of weblogs; business entities will guarantee the successful take-up and exploitation of the project's outputs. Representatives from bloggers communities will ensure that the results cover their needs.* [57]

12.2 Commentary on the Definition of a Blog

Referring to the argument above, it is possible to infer that for the purposes of BlogForever, this report defines a blog by referring to its perceived meaning as something that can be described as a stream of data, and through careful identification of the separate (1) **Content**, (2) **Function** and (3) **Context** elements, it will be possible to deliver an (4) **Experience** to the end-user and meet the requirements of authenticity in preservation. The adoption of the working definition of a blog as an object for preservation (introduced in Section 2.1.2) is suitable to address the aims and objectives of the BlogForever project.

[55] http://pandora.nla.gov.au/policy_practice.html
[56] http://pandora.nla.gov.au/guidelines.html
[57] BlogForever Description of Work document, project summary abstract.

13 Summary, Conclusion and Future Work

This report elaborates the process of developing a data model that aims to support the development of the BlogForever platform. Prior to embarking on the data modelling, an appropriate method has been chosen and followed. It required eliciting a working definition of a blog – setting the necessary grounds for progress.

The report starts by reinstating the requirement of the project and the work package in particular. It then progresses to discuss the steps taken to inform the development of the model. The report refers to the earlier completed BlogForever task (Task 2.1[58]) that played an important role for informing the development of the data model. The outcomes reported in the submitted work have been revisited and discussed retrospectively to highlight the components of blogs that are particularly important for achieving the aims of the BlogForever project and blog preservation in general.

Building on the information obtained the data modelling process was directed to take into account the existing conceptual and data models for blogs. Furthermore, database systems of the existing blogging platforms and their APIs were studied for extracting clues about the structure and components of blogs. The evaluation of these resources enabled eliciting a core, generic data model for blogs that has been extended by exploring the semantic structure of blogs.

Extending further, the report discussed an empirical study that evaluated 2,695 blog web feeds. The aim of the study was to acquire insight into the data structures and external modules used for distributing blog content and associated metadata. The results revealed the existence and the level of adoption of certain blog components and data types, which eventually informed the development of the model that enables the storage of the data distributed via web feeds.

The proposed model has been presented graphically as a conceptual and a logical model. The details of the data modelling approach and the specifications of the proposed data structures were discussed in the report. The report positioned the proposed data model in relation to the processes and data flow anticipated from the solutions being developed as part of the BlogForever project. It discussed possible ways forward for integrating the data model into the existing structures underpinning the Invenio software suite.

Last, but not least, the report integrated two internal consultation exercises from the perspectives of digital preservation, as well as technical relevance. These consultation exercises enabled some refinement and clarification of the proposed model, but most importantly, emphasised the possibility of further modifications according to the future project tasks.

[58] *Submitted on 30 August, 2011, and available for download at http://blogforever.eu*

14 References

[1] S. Arango-Docio, P. Sleeman, and H. Kalb, "BlogForever: D2.1 Survey Implementation Report", 2011.

[2] P. Winn (2008, 15 September 2011). *State of the Blogosphere: Introduction*. Available: http://technorati.com/blogging/article/state-of-the-blogosphere-introduction/

[3] P. Ponniah, *Data modeling fundamentals: a practical guide for IT professionals*. Hoboken, New Jersey, USA: Wiley-Blackwell, 2007.

[4] H. K. Klein and R. Hirschheim, "A comparative framework of data modelling paradigms and approaches," *The Computer Journal*, vol. 30, p. 8, 1987.

[5] M. West, *Developing High Quality Data Models*. Burlington, MA, USA: Morgan Kaufmann, Elsevier, 2011.

[6] B. A. Nardi, D. J. Schiano, M. Gumbrecht, and L. Swartz, "Why we blog," *Communications of the ACM*, vol. 47, pp. 41-46, 2004.

[7] P. Pluempavarn and N. Panteli, "Building social identity through blogging", in *Exploring virtuality within and beyond organizations: social, global, and local dimensions*, N. Panteli and M. Chiasson, Eds., ed New York, USA: Palgrave Macmillan, 2008, pp. 195-212.

[8] K. Thibodeau, "Overview of technological approaches to digital preservation and challenges in coming years", in *The State of Digital Preservation: an International Perspective*, Library of Congress, Washington D.C., USA, 2002, pp. 24–25.

[9] H. Heslop, S. Davis, and A. Wilson, "An approach to the preservation of digital records," *National Archives of Australia*, 2002.

[10] C. Webb, N. L. o. Australia, U. Information, and I. Division. (2003). *Guidelines for the preservation of digital heritage*. Available: http://unesdoc.unesco.org/images/0013/001300/130071e.pdf

[11] L. Clausen, "Opening schrödingers library: Semi-automatic QA reduces uncertainty in object transformation", in *Research and Advanced Technology for Digital Libraries, 11th European Conference, ECDL 2007*, Budapest, Hungary, 2007, pp. 186-197.

[12] S. Ross and M. Hedstrom. (2005, Preservation research and sustainable digital libraries. *International Journal on Digital Libraries 5(4)*, 317-324. Available: http://eprints.erpanet.org/archive/00000095/, http://www.springerlink.com/content/d0u4ch6ng0y7ct2j/

[13] A. Christopher, "A framework for contextual information in digital collections," *Journal of Documentation*, vol. 67, pp. 95-143, 2011.

[14] J. Sowa, "Syntax, semantics, and pragmatics of contexts", in *AAAI Technical Report FS-95-02, Association for the Advancement of Artificial Intelligence (AAAI) Symposia*, ed, 1995, pp. 1-15.

[15] M. F. Moens, *Information extraction: algorithms and prospects in a retrieval context* vol. 21: Springer, 2006.

[16] S. Nakajima, J. Tatemura, Y. Hino, Y. Hara, and K. Tanaka, "Discovering important bloggers based on analyzing blog threads", in *WWW 2005*, Chiba, Japan, 2005.

[17] T. A. Ta, J. M. Saglio, and M. Plu, "An architecture based on semantic weblogs for exploring the Web of People", in *ECAI 2004 Workshop on Application of Semantic Web Technologies to Web Communities*, Valencia, Spain, 2004.

[18] M. E. Davis and J. A. Phillips, *Learning PHP and MySQL*: O'Reilly Media, Inc., 2007.

[19] T. Converse, J. Park, and C. Morgan, *PHP5 and MySQL bible* vol. 147: Wiley, 2004.

[20] B. Williams. (2005, 15.07.2011). *Database Answers: Blog Data Model*. Available: http://www.databaseanswers.org/data_models/blogs/index.htm

[21] B. Hammersley, *Developing feeds with RSS and Atom*: O'Reilly, 2005.

[22] M. Pennock and R. Davis, "ArchivePress: A really simple solution to archiving blog content", in *iPress 2009*, San Francisco, California, USA, 2009.

[23] Google. (2011, 01.09.2011). *FeedBurner API (Deprecated)*. Available: http://code.google.com/apis/feedburner/

[24] C. Lindahl and E. Blount, "Weblogs: simplifying web publishing," *Computer*, vol. 36, pp. 114-116, 2003.
[25] E. Hull, K. Jackson, and J. Dick, *Requirements engineering*: Springer-Verlag New York Inc, 2010.
[26] J. Caffaro and S. Kaplun, "Invenio: A Modern Digital Library for Grey Literature", in *Twelfth International Conference on Grey Literature*, Prague, Czech Republic, 2010.
[27] M. Enders, "A METS based information package for long term accessibility of web archives", in *7th International Conference on Preservation of Digital Objects (iPRES2010)*, Vienna, Austria, 2010.
[28] A. Brown, "Selecting file formats for long-term preservation", in *Digital Preservation Guidance Note* vol. 1, ed: The National Archives, 2008.
[29] S. Grace, "Investigating the Significant Properties of Electronic Content over Time", King's College London 2009.
[30] NAA, "Digital Preservation Approaches", in *National Archives of Australia*, ed.
[31] P. Caplan. (2010, 01/09/2011). *Preservation Metadata*. Available: http://www.dcc.ac.uk/resources/curation-reference-manual/completed-chapters/preservation-metadata
[32] M. Davis, "Preserving access–Making more informed guesses about what works", in *Report to the IIPC Preservation Working Group*, ed: Digital Preservation, National Library of Australia, 2009.
[33] C. A. Lee, "A framework for contextual information in digital collections," *Journal of Documentation*, vol. 67, pp. 95-143, 2011.
[34] B. Hughes, "Link Rot. URI Citation Durability in 10 Years of AusWeb Proceedings.," 2006.
[35] S. Thompson. (2007, 01/09/2011). *URIs and Persistence: How long is forever?* Available: http://www.ltg.ed.ac.uk/~ht/UKOLN_talk_20070405.html#

A. Appendix A – List of Modules Identified in Web Feeds

A.1 List of Modules and Their Properties

Modules Used	Modules Attributes	All Nodes (Sorted by Stats.)	Frequencies
activity	activity:actor	link	107676
admin	activity:object	category	85879
alacra	activity:object-type	title	66378
annotate	activity:target	description	48010
apcm	activity:verb	item	46929
apnm	admin:errorReportsTo	pubDate	44162
app	admin:generatorAgent	guid	35528
article	alacra:ip	author	19827
atom	annotate:reference	id	16927
bing	apcm:ByLine	updated	15967
blogChannel	apcm:Characteristics	entry	15177
c	apcm:ContentMetadata	published	14588
c9	apcm:DateLine	content	13850
cc	apcm:HeadLine	name	13353
cf	apcm:SlugLine	dc:creator	12712
cinch	apcm:Source	thr:total	10190
clearspace	apnm:ManagementId	comments	9575
content	apnm:ManagementSequenceNumber	uri	9406
creativeCommons	apnm:ManagementType	email	8840
ctek	apnm:NewsManagement	gd:extendedProperty	7382
dc	apnm:PublishingStatus	media:thumbnail	7017
dcterms	app:edited	wfw:commentRss	6598
dig	article:availableInPlayer	source	6237
digg	article:date	content:encoded	6058
disqus	article:desktopAlertFlag	slash:comments	5871
dm	article:headerImage	media:content	5870
dwsyn	article:teaserImage	feedburner:origLink	5264
dz	article:uid	media:title	4400
ev	atom:author	summary	4261
feedburner	atom:id	enclosure	2817
g	atom:link	dc:date	2510
gCal	atom:name	channel	1929
gd	atom:summary	rss	1911
geo	atom:updated	generator	1625
georss	atom:uri	language	1500
ghs	atom10:link	p	1233
gml	bing:mediaSource	lastBuildDate	1077
ibsys	blogChannel:blink	image	1070
itunes	blogChannel:blogRoll	atom:link	1044
jf	c:body	a	1027
ka	c:image_big	url	891
lj	c:image_small	media:description	821
lw	c:slide	feed	792
media	c:slides	app:edited	775
mf	c:title	activity:object-type	764
mn	c9:pageCount	media:credit	762
moodys	c9:pageSize	subtitle	749
ni	c9:totalResults	atom10:link	682
nj	cc:license	img	669
odat	cf:treatAs	scout:premiumflag	660
on	cinch:facebookurl	scout:sourcefriendlysubdomain	660
opensearch	cinch:twitterurl	scout:sourcesiteid	660
os	clearspace:dateToText	scout:sourcesubdomain	660
pheedo	clearspace:objectType	georss:point	574
pingback	clearspace:replyCount	sy:updatePeriod	569
pink	content:encoded	sy:updateFrequency	567
poco	content:encoding	dc:subject	493

posterous	content:format	div	483
pubitems	content:item	atom:updated	475
rdf	content:items	rdf:li	475
rss	creativeCommons:license	copyright	473
s	ctek:copyright	ttl	453
sb	ctek:lastEditDate	feedburner:feedFlare	442
scout	dc:creator	br	431
series	dc:date	soup:attributes	400
slash	dc:date.Taken	size	379
soup	dc:format	openSearch:itemsPerPage	364
sx	dc:identifier	openSearch:startIndex	364
sy	dc:language	openSearch:totalResults	364
syn	dc:publisher	info_hash	359
thespringbox	dc:rights	leechers	359
thr	dc:source	seeders	359
topix	dc:subject	media:player	347
trackback	dc:title	activity:verb	338
WebWizForums	dc:type	feedburner:info	330
wfw	dcterms:created	mf:hasComments	330
wire	dcterms:modified	media:category	321
wn	dig:alt	span	306
wordzilla	dig:D3text	veranstaltungsort	300
xhtml	dig:height	topix:comments	280
	dig:image	date	269
	dig:keywords	media:keywords	255
	dig:url	itunes:summary	229
	dig:width	lj:music	226
	digg:category	moodys:index_entry_child_id	217
	digg:commentCount	thumbnail	215
	digg:diggCount	pheedo:origLink	214
	digg:submitter	activity:actor	213
	digg:userimage	activity:object	213
	digg:username	activity:target	213
	disqus:identifier	poco:displayName	213
	disqus:shortname	poco:familyName	213
	disqus:thread	poco:givenName	213
	dm:author	poco:name	213
	dm:authorAvatar	poco:preferredUsername	213
	dm:channels	sx:history	213
	dm:comments	sx:sync	213
	dm:favorites	dc:source	205
	dm:id	em	201
	dm:link	height	201
	dm:loggerURL	imageThumb	200
	dm:relativeDate	lj:poster	200
	dm:videorating	smallImage	200
	dm:videovotes	squareImage	200
	dm:views	width	199
	dwsyn:contentID	dc:language	192
	dz:commentCount	itunes:explicit	183
	dz:readCount	docs	176
	dz:submitDate	embed	176
	dz:submitter	dc:format	173
	dz:userimage	cloud	172
	dz:username	geo:lat	161
	ev:enddate	geo:long	161
	ev:location	ctek:copyright	160
	ev:startdate	ctek:lastEditDate	160
	feedburner:browserFriendly	itunes:keywords	148
	feedburner:emailServiceId	managingEditor	147
	feedburner:feedburnerHostname	media:rating	147
	feedburner:feedFlare	webMaster	129
	feedburner:info	creativeCommons:license	127
	feedburner:origEnclosureLink	categories	124

	feedburner:origLink	dwsyn:contentID	124
	g:color	timestamp	124
	g:condition	feedburner:emailServiceId	121
	g:expiration_date	feedburner:feedburnerHostname	121
	g:image_link	media:group	120
	g:location	media:text	119
	g:make	itunes:duration	118
	g:mileage	moodys:issuer	110
	g:model	moodys:org_id	110
	g:price	tags	110
	g:price_type	itunes:author	109
	g:quantity	wn:adClassification	108
	g:vehicle_type	dig:image	107
	g:vin	dig:alt	102
	g:year	dig:height	102
	gCal:timesCleaned	dig:url	102
	gCal:timezone	dig:width	102
	gd:extendedProperty	alacra:ip	101
	geo:lat	annotate:reference	100
	geo:long	ka:category	100
	geo:Point	ka:city	100
	georss:box	ka:country	100
	georss:featurename	ka:creatorId	100
	georss:point	ka:duration	100
	georss:where	ka:gadChannel	100
	ghs:dateline	ka:gadhost	100
	ghs:editor_notes	ka:gadPublisher	100
	ghs:image	ka:gadtype	100
	ghs:keywords	ka:id	100
	ghs:source	ka:keywords	100
	ghs:thumb	ka:level	100
	ghs:thumb-large	ka:mediaType	100
	gml:Point	ka:numOfComments	100
	gml:pos	ka:points	100
	ibsys:annotation	ka:rating	100
	itunes:author	ka:state	100
	itunes:block	ka:uploadedByThumbnail	100
	itunes:category	ka:uploadedByUrl	100
	itunes:duration	ka:userDisabled	100
	itunes:email	ka:views	100
	itunes:explicit	ka:votes	100
	itunes:image	ka:zip	100
	itunes:keywords	lj:journal	100
	itunes:name	media:adult	100
	itunes:new-feed-url	misc	100
	itunes:owner	moodys:docid	100
	itunes:subtitle	moodys:format	100
	itunes:summary	moodys:index_mode	100
	jf:author	moodys:issuer_list	100
	jf:creationDate	moodys:moodys_org_id	100
	jf:date	moodys:price	100
	jf:messageCount	moodys:printdate	100
	jf:modificationDate	moodys:report_class	100
	jf:replyCount	moodys:report_type	100
	ka:category	moodys:sector	100
	ka:city	dig:D3text	96
	ka:country	rights	95
	ka:creatorId	geo:Point	94
	ka:duration	pubDateNumber	94
	ka:favorites	comments-url	80
	ka:feedId	digg:category	80
	ka:gadChannel	digg:commentCount	80
	ka:gadhost	digg:diggCount	80

	ka:gadPublisher	digg:submitter	80
	ka:gadtype	digg:userimage	80
	ka:id	digg:username	80
	ka:keywords	ev:enddate	80
	ka:level	ev:location	80
	ka:mediaType	ev:startdate	80
	ka:moreResults	postid	80
	ka:numOfComments	rss-url	80
	ka:points	pink:symbol	79
	ka:rating	pink:type	79
	ka:state	strong	76
	ka:totalItems	itunes:subtitle	75
	ka:uploadedByThumbnail	dc:publisher	63
	ka:uploadedByUrl	ghs:keywords	62
	ka:userDisabled	h3	61
	ka:views	draft	60
	ka:votes	fullpubdate	60
	ka:zip	pubdate	60
	lj:journal	moodys:index_entry_industry	59
	lj:music	ghs:dateline	58
	lj:poster	ghs:editor_notes	58
	lw:itemCount	ghs:source	58
	lw:userID	ghs:image	56
	media:adult	ghs:thumb	56
	media:category	ghs:thumb-large	56
	media:community	thumb	56
	media:content	itunes:category	55
	media:copyright	atom:summary	54
	media:credit	media:copyright	53
	media:description	dig:keywords	51
	media:group	dm:link	51
	media:keywords	breaking	50
	media:player	cc:license	50
	media:rating	os:celebrant1	50
	media:statistics	os:celebrant2	50
	media:text	os:location1	50
	media:thumbnail	os:location2	50
	media:title	os:messages	50
	mf:hasComments	os:photo	50
	mn:channels	os:photos	50
	moodys:docid	os:simchaDate	50
	moodys:format	os:simchaType	50
	moodys:index_entry_child_id	issued	48
	moodys:index_entry_country	clearspace:dateToText	45
	moodys:index_entry_industry	dz:commentCount	45
	moodys:index_entry_ultparent_id	dz:readCount	45
	moodys:index_mode	dz:submitDate	45
	moodys:issuer	dz:submitter	45
	moodys:issuer_list	dz:userimage	45
	moodys:moodys_org_id	dz:username	45
	moodys:org_id	moodys:index_entry_ultparent_id	45
	moodys:price	alt_time	44
	moodys:printdate	byline	44
	moodys:report_class	credit	44
	moodys:report_type	print_pubDate	44
	moodys:sector	publish_date	44
	ni:feed	apcm:ByLine	42
	nj:rubric	g:color	42
	odat:availability	g:condition	42
	odat:detailimage	g:expiration_date	42
	odat:image	g:image_link	42
	odat:listDescription	g:location	42
	odat:price	g:make	42

	odat:priceCurrent	g:mileage	42
	odat:priceRegular	g:model	42
	odat:thumbnail	g:price	42
	odat:tinyImage	g:price_type	42
	on:short_desc	g:quantity	42
	opensearch:itemsPerPage	g:vehicle_type	42
	openSearch:itemsPerPage	g:vin	42
	opensearch:startIndex	g:year	42
	openSearch:startIndex	jf:author	40
	opensearch:totalResults	jf:creationDate	40
	openSearch:totalResults	jf:modificationDate	40
	os:celebrant1	jf:replyCount	40
	os:celebrant2	moodys:index_entry_country	40
	os:location1	apcm:Characteristics	39
	os:location2	apcm:ContentMetadata	39
	os:messages	apcm:DateLine	39
	os:photo	apcm:HeadLine	39
	os:photos	apcm:SlugLine	39
	os:simchaDate	apcm:Source	39
	os:simchaType	apnm:ManagementId	39
	pheedo:origEnclosureLink	apnm:ManagementSequenceNumber	39
	pheedo:origLink	apnm:ManagementType	39
	pingback:server	apnm:NewsManagement	39
	pingback:target	apnm:PublishingStatus	39
	pink:filingType	a10:updated	38
	pink:periodDate	dc:rights	37
	pink:symbol	feedburner:origEnclosureLink	37
	pink:type	li	37
	poco:displayName	flv	36
	poco:familyName	iframe	35
	poco:givenName	cinch:facebookurl	30
	poco:name	cinch:twitterurl	30
	poco:preferredUsername	dc:type	30
	posterous:author	jf:date	30
	posterous:displayName	feedburner:browserFriendly	28
	posterous:firstName	georss:box	27
	posterous:lastName	georss:featurename	27
	posterous:nickName	itunes:image	27
	posterous:profileUrl	keywords	25
	posterous:userImage	modified	25
	pubitems:byline	trackback:ping	25
	pubitems:category	ka:favorites	24
	pubitems:image	param	24
	pubitems:imageh	contributor	22
	pubitems:imagew	atom:author	20
	pubitems:metadata	atom:name	20
	pubitems:priority	clearspace:objectType	20
	pubitems:subcategory	disqus:identifier	20
	rdf:Bag	disqus:shortname	20
	rdf:Description	disqus:thread	20
	rdf:li	dm:author	20
	rdf:RDF	dm:authorAvatar	20
	rdf:Seq	dm:channels	20
	rdf:value	dm:comments	20
	rss:channel	dm:favorites	20
	rss:description	dm:id	20
	rss:image	dm:loggerURL	20
	rss:item	dm:relativeDate	20
	rss:link	dm:videorating	20
	rss:title	dm:videovotes	20
	rss:url	dm:views	20
	s:counts	featured	20
	sb:created	featured_headline	20

	sb:credit	featured_thumbnail	20
	scout:premiumflag	hash	20
	scout:sourcefriendlysubdomain	image_url	20
	scout:sourcesiteid	imageurl	20
	scout:sourcesubdomain	nj:rubric	20
	series:name	peers	20
	slash:comments	posterous:author	20
	soup:attributes	posterous:displayName	20
	sx:history	posterous:firstName	20
	sx:sync	posterous:lastName	20
	sy:updateBase	posterous:nickName	20
	sy:updateFequency	posterous:profileUrl	20
	sy:updateFrequency	posterous:userImage	20
	sy:updatePeriod	sb:created	20
	syn:updateBase	seeds	20
	syn:updateFrequency	short_description	20
	syn:updatePeriod	shorthead	20
	thespringbox:skin	tracker	20
	thr:total	votes	20
	topix:comments	wire:thumb	20
	topix:rsslink	rdf:RDF	19
	trackback:ping	atom:id	18
	WebWizForums:feedURL	rdf:Seq	18
	wfw:comment	atom:uri	17
	wfw:commentRss	items	17
	wire:thumb	logo	17
	wn:adClassification	rank	17
	wn:wnreadableurl	xhtml:meta	17
	wordzilla:id	itunes:email	16
	xhtml:img	itunes:owner	16
	xhtml:meta	c:body	15
		c:image_big	15
		c:image_small	15
		c:slide	15
		c:title	15
		cid	15
		created	15
		dc:identifier	15
		itunes:name	15
		lastModified	15
		news_date	15
		state	15
		wordzilla:id	15
		clearspace:replyCount	14
		field_teaser	14
		odat:availability	14
		odat:detailimage	14
		odat:image	14
		odat:listDescription	14
		odat:price	14
		odat:priceCurrent	14
		odat:priceRegular	14
		odat:thumbnail	14
		odat:tinyImage	14
		sb:credit	14
		topix:rsslink	14
		icon	13
		on:short_desc	13
		itunes:block	12
		admin:generatorAgent	11
		description2	11
		georss:where	11
		gml:Point	11
		gml:pos	11

		sy:updateBase	11
		article:availableInPlayer	10
		article:date	10
		article:desktopAlertFlag	10
		article:headerImage	10
		article:teaserImage	10
		article:uid	10
		hasDetail	10
		item_Id	10
		jf:messageCount	10
		lw:userID	10
		media:community	10
		media:statistics	10
		newsid	10
		photo	10
		pic	10
		pingback:server	10
		pingback:target	10
		pink:filingType	10
		pink:periodDate	10
		section	10
		wfw:comment	10
		textInput	9
		video	9
		cf:treatAs	8
		ibsys:annotation	8
		rating	8
		dc:date.Taken	7
		group	7
		xhtml:img	7
		ul	6
		admin:errorReportsTo	5
		blockquote	5
		c:slides	5
		dcterms:created	5
		dcterms:modified	5
		pubitems:byline	5
		pubitems:category	5
		pubitems:image	5
		pubitems:imageh	5
		pubitems:imagew	5
		pubitems:metadata	5
		pubitems:priority	5
		pubitems:subcategory	5
		series:name	5
		small	5
		thespringbox:skin	5
		h6	4
		has-gallery	4
		meta	4
		next	4
		object	4
		pages	4
		prev	4
		rss:title	4
		script	4
		rss:description	3
		rss:link	3
		content:encoding	2
		content:format	2
		content:item	2
		content:items	2
		generation_time	2
		incremental	2

		itunes:new-feed-url	2
		media	2
		pheedo:origEnclosureLink	2
		rdf:Bag	2
		rdf:value	2
		rss:image	2
		rss:item	2
		stock	2
		sy:updateFequency	2
		tagline	2
		textinput	2
		WebWizForums:feedURL	2
		a10:link	1
		b	1
		bing:mediaSource	1
		blogChannel:blink	1
		blogChannel:blogRoll	1
		c9:pageCount	1
		c9:pageSize	1
		c9:totalResults	1
		dc:title	1
		feedLink	1
		gCal:timesCleaned	1
		gCal:timezone	1
		headerImage	1
		i	1
		ka:feedId	1
		ka:moreResults	1
		ka:totalItems	1
		lastUpdatedDate	1
		logolink	1
		lw:itemCount	1
		mn:channels	1
		ni:feed	1
		noscript	1
		opensearch:itemsPerPage	1
		opensearch:startIndex	1
		opensearch:totalResults	1
		rdf:Description	1
		rss:channel	1
		rss:url	1
		s:counts	1
		syn:updateBase	1
		syn:updateFrequency	1
		syn:updatePeriod	1
		updatePeri	1
		webmaster	1
		wn:wnreadableurl	1

B. Appendix B – Complete Frequency Table of Identified Nodes

Nodes

		Statistics			
	Nodes	Frequency	Percent	Valid Percent	Cumulative Percent
Valid	a	1027	.1	.1	.1
	a10:link	1	.0	.0	.1
	a10:updated	38	.0	.0	.1
	activity:actor	213	.0	.0	.2
	activity:object	213	.0	.0	.2
	activity:object-type	764	.1	.1	.3
	activity:target	213	.0	.0	.3
	activity:verb	338	.0	.0	.4
	admin:errorReportsTo	5	.0	.0	.4
	admin:generatorAgent	11	.0	.0	.4
	alacra:ip	101	.0	.0	.4
	alt_time	44	.0	.0	.4
	annotate:reference	100	.0	.0	.4
	apcm:ByLine	42	.0	.0	.4
	apcm:Characteristics	39	.0	.0	.4
	apcm:ContentMetadata	39	.0	.0	.4
	apcm:DateLine	39	.0	.0	.5
	apcm:HeadLine	39	.0	.0	.5
	apcm:SlugLine	39	.0	.0	.5
	apcm:Source	39	.0	.0	.5
	apnm:ManagementId	39	.0	.0	.5
	apnm:ManagementSequenceNumber	39	.0	.0	.5
	apnm:ManagementType	39	.0	.0	.5
	apnm:NewsManagement	39	.0	.0	.5
	apnm:PublishingStatus	39	.0	.0	.5
	app:edited	775	.1	.1	.6
	article:availableInPlayer	10	.0	.0	.6
	article:date	10	.0	.0	.6
	article:desktopAlertFlag	10	.0	.0	.6
	article:headerImage	10	.0	.0	.6
	article:teaserImage	10	.0	.0	.6

article:uid	10	.0	.0	.6
atom:author	20	.0	.0	.6
atom:id	18	.0	.0	.6
atom:link	1044	.1	.1	.8
atom:name	20	.0	.0	.8
atom:summary	54	.0	.0	.8
atom:updated	475	.1	.1	.8
atom:uri	17	.0	.0	.8
atom10:link	682	.1	.1	.9
author	19827	2.8	2.8	3.7
b	1	.0	.0	3.7
bing:mediaSource	1	.0	.0	3.7
blockquote	5	.0	.0	3.7
blogChannel:blink	1	.0	.0	3.7
blogChannel:blogRoll	1	.0	.0	3.7
br	431	.1	.1	3.8
breaking	50	.0	.0	3.8
byline	44	.0	.0	3.8
c:body	15	.0	.0	3.8
c:image_big	15	.0	.0	3.8
c:image_small	15	.0	.0	3.8
c:slide	15	.0	.0	3.8
c:slides	5	.0	.0	3.8
c:title	15	.0	.0	3.8
c9:pageCount	1	.0	.0	3.8
c9:pageSize	1	.0	.0	3.8
c9:totalResults	1	.0	.0	3.8
categories	124	.0	.0	3.8
category	85879	12.0	12.0	15.9
cc:license	50	.0	.0	15.9
cf:treatAs	8	.0	.0	15.9
channel	1929	.3	.3	16.1
cid	15	.0	.0	16.1
cinch:facebookurl	30	.0	.0	16.1
cinch:twitterurl	30	.0	.0	16.1
clearspace:dateToText	45	.0	.0	16.2
clearspace:objectType	20	.0	.0	16.2
clearspace:replyCount	14	.0	.0	16.2

cloud	172	.0	.0	16.2
comments	9575	1.3	1.3	17.5
comments-url	80	.0	.0	17.5
content	13850	1.9	1.9	19.5
content:encoded	6058	.8	.8	20.3
content:encoding	2	.0	.0	20.3
content:format	2	.0	.0	20.3
content:item	2	.0	.0	20.3
content:items	2	.0	.0	20.3
contributor	22	.0	.0	20.3
copyright	473	.1	.1	20.4
created	15	.0	.0	20.4
creativeCommons:license	127	.0	.0	20.4
credit	44	.0	.0	20.4
ctek:copyright	160	.0	.0	20.4
ctek:lastEditDate	160	.0	.0	20.5
date	269	.0	.0	20.5
dc:creator	12712	1.8	1.8	22.3
dc:date	2510	.4	.4	22.6
dc:date.Taken	7	.0	.0	22.6
dc:format	173	.0	.0	22.7
dc:identifier	15	.0	.0	22.7
dc:language	192	.0	.0	22.7
dc:publisher	63	.0	.0	22.7
dc:rights	37	.0	.0	22.7
dc:source	205	.0	.0	22.7
dc:subject	493	.1	.1	22.8
dc:title	1	.0	.0	22.8
dc:type	30	.0	.0	22.8
dcterms:created	5	.0	.0	22.8
dcterms:modified	5	.0	.0	22.8
description	48010	6.7	6.7	29.5
description2	11	.0	.0	29.5
dig:alt	102	.0	.0	29.6
dig:D3text	96	.0	.0	29.6
dig:height	102	.0	.0	29.6
dig:image	107	.0	.0	29.6
dig:keywords	51	.0	.0	29.6

dig:url	102	.0	.0	29.6
dig:width	102	.0	.0	29.6
digg:category	80	.0	.0	29.6
digg:commentCount	80	.0	.0	29.7
digg:diggCount	80	.0	.0	29.7
digg:submitter	80	.0	.0	29.7
digg:userimage	80	.0	.0	29.7
digg:username	80	.0	.0	29.7
disqus:identifier	20	.0	.0	29.7
disqus:shortname	20	.0	.0	29.7
disqus:thread	20	.0	.0	29.7
div	483	.1	.1	29.8
dm:author	20	.0	.0	29.8
dm:authorAvatar	20	.0	.0	29.8
dm:channels	20	.0	.0	29.8
dm:comments	20	.0	.0	29.8
dm:favorites	20	.0	.0	29.8
dm:id	20	.0	.0	29.8
dm:link	51	.0	.0	29.8
dm:loggerURL	20	.0	.0	29.8
dm:relativeDate	20	.0	.0	29.8
dm:videorating	20	.0	.0	29.8
dm:videovotes	20	.0	.0	29.8
dm:views	20	.0	.0	29.8
docs	176	.0	.0	29.8
draft	60	.0	.0	29.9
dwsyn:contentID	124	.0	.0	29.9
dz:commentCount	45	.0	.0	29.9
dz:readCount	45	.0	.0	29.9
dz:submitDate	45	.0	.0	29.9
dz:submitter	45	.0	.0	29.9
dz:userimage	45	.0	.0	29.9
dz:username	45	.0	.0	29.9
em	201	.0	.0	29.9
email	8840	1.2	1.2	31.2
embed	176	.0	.0	31.2
enclosure	2817	.4	.4	31.6
entry	15177	2.1	2.1	33.7

ev:enddate	80	.0	.0	33.7
ev:location	80	.0	.0	33.7
ev:startdate	80	.0	.0	33.8
featured	20	.0	.0	33.8
featured_headline	20	.0	.0	33.8
featured_thumbnail	20	.0	.0	33.8
feed	792	.1	.1	33.9
feedburner:browserFriendly	28	.0	.0	33.9
feedburner:emailServiceId	121	.0	.0	33.9
feedburner:feedburnerHostname	121	.0	.0	33.9
feedburner:feedFlare	442	.1	.1	34.0
feedburner:info	330	.0	.0	34.0
feedburner:origEnclosureLink	37	.0	.0	34.0
feedburner:origLink	5264	.7	.7	34.8
feedLink	1	.0	.0	34.8
field_teaser	14	.0	.0	34.8
flv	36	.0	.0	34.8
fullpubdate	60	.0	.0	34.8
g:color	42	.0	.0	34.8
g:condition	42	.0	.0	34.8
g:expiration_date	42	.0	.0	34.8
g:image_link	42	.0	.0	34.8
g:location	42	.0	.0	34.8
g:make	42	.0	.0	34.8
g:mileage	42	.0	.0	34.8
g:model	42	.0	.0	34.8
g:price	42	.0	.0	34.8
g:price_type	42	.0	.0	34.8
g:quantity	42	.0	.0	34.8
g:vehicle_type	42	.0	.0	34.8
g:vin	42	.0	.0	34.9
g:year	42	.0	.0	34.9
gCal:timesCleaned	1	.0	.0	34.9
gCal:timezone	1	.0	.0	34.9
gd:extendedProperty	7382	1.0	1.0	35.9
generation_time	2	.0	.0	35.9

generator	1625	.2	.2	36.1
geo:lat	161	.0	.0	36.1
geo:long	161	.0	.0	36.2
geo:Point	94	.0	.0	36.2
georss:box	27	.0	.0	36.2
georss:featurename	27	.0	.0	36.2
georss:point	574	.1	.1	36.3
georss:where	11	.0	.0	36.3
ghs:dateline	58	.0	.0	36.3
ghs:editor_notes	58	.0	.0	36.3
ghs:image	56	.0	.0	36.3
ghs:keywords	62	.0	.0	36.3
ghs:source	58	.0	.0	36.3
ghs:thumb	56	.0	.0	36.3
ghs:thumb-large	56	.0	.0	36.3
gml:Point	11	.0	.0	36.3
gml:pos	11	.0	.0	36.3
group	7	.0	.0	36.3
guid	35528	5.0	5.0	41.3
h3	61	.0	.0	41.3
h6	4	.0	.0	41.3
has-gallery	4	.0	.0	41.3
hasDetail	10	.0	.0	41.3
hash	20	.0	.0	41.3
headerImage	1	.0	.0	41.3
height	201	.0	.0	41.4
i	1	.0	.0	41.4
ibsys:annotation	8	.0	.0	41.4
icon	13	.0	.0	41.4
id	16927	2.4	2.4	43.7
iframe	35	.0	.0	43.7
image	1070	.1	.1	43.9
image_url	20	.0	.0	43.9
imageThumb	200	.0	.0	43.9
imageurl	20	.0	.0	43.9
img	669	.1	.1	44.0
incremental	2	.0	.0	44.0
info_hash	359	.1	.1	44.1

	issued	48	.0	.0	44.1
	item	46929	6.6	6.6	50.6
	item_Id	10	.0	.0	50.6
	items	17	.0	.0	50.7
	itunes:author	109	.0	.0	50.7
	itunes:block	12	.0	.0	50.7
	itunes:category	55	.0	.0	50.7
	itunes:duration	118	.0	.0	50.7
	itunes:email	16	.0	.0	50.7
	itunes:explicit	183	.0	.0	50.7
	itunes:image	27	.0	.0	50.7
	itunes:keywords	148	.0	.0	50.7
	itunes:name	15	.0	.0	50.7
	itunes:new-feed-url	2	.0	.0	50.7
	itunes:owner	16	.0	.0	50.7
	itunes:subtitle	75	.0	.0	50.8
	itunes:summary	229	.0	.0	50.8
	jf:author	40	.0	.0	50.8
	jf:creationDate	40	.0	.0	50.8
	jf:date	30	.0	.0	50.8
	jf:messageCount	10	.0	.0	50.8
	jf:modificationDate	40	.0	.0	50.8
	jf:replyCount	40	.0	.0	50.8
	ka:category	100	.0	.0	50.8
	ka:city	100	.0	.0	50.8
	ka:country	100	.0	.0	50.9
	ka:creatorId	100	.0	.0	50.9
	ka:duration	100	.0	.0	50.9
	ka:favorites	24	.0	.0	50.9
	ka:feedId	1	.0	.0	50.9
	ka:gadChannel	100	.0	.0	50.9
	ka:gadhost	100	.0	.0	50.9
	ka:gadPublisher	100	.0	.0	50.9
	ka:gadtype	100	.0	.0	50.9
	ka:id	100	.0	.0	51.0
	ka:keywords	100	.0	.0	51.0
	ka:level	100	.0	.0	51.0
	ka:mediaType	100	.0	.0	51.0

ka:moreResults	1	.0	.0	51.0
ka:numOfComments	100	.0	.0	51.0
ka:points	100	.0	.0	51.0
ka:rating	100	.0	.0	51.0
ka:state	100	.0	.0	51.1
ka:totalItems	1	.0	.0	51.1
ka:uploadedByThumbnail	100	.0	.0	51.1
ka:uploadedByUrl	100	.0	.0	51.1
ka:userDisabled	100	.0	.0	51.1
ka:views	100	.0	.0	51.1
ka:votes	100	.0	.0	51.1
ka:zip	100	.0	.0	51.1
keywords	25	.0	.0	51.1
language	1500	.2	.2	51.4
lastBuildDate	1077	.2	.2	51.5
lastModified	15	.0	.0	51.5
lastUpdatedDate	1	.0	.0	51.5
leechers	359	.1	.1	51.6
li	37	.0	.0	51.6
link	107676	15.1	15.1	66.7
lj:journal	100	.0	.0	66.7
lj:music	226	.0	.0	66.7
lj:poster	200	.0	.0	66.7
logo	17	.0	.0	66.7
logolink	1	.0	.0	66.7
lw:itemCount	1	.0	.0	66.7
lw:userID	10	.0	.0	66.7
managingEditor	147	.0	.0	66.8
media	2	.0	.0	66.8
media:adult	100	.0	.0	66.8
media:category	321	.0	.0	66.8
media:community	10	.0	.0	66.8
media:content	5870	.8	.8	67.6
media:copyright	53	.0	.0	67.7
media:credit	762	.1	.1	67.8
media:description	821	.1	.1	67.9
media:group	120	.0	.0	67.9
media:keywords	255	.0	.0	67.9

media:player	347	.0	.0	68.0
media:rating	147	.0	.0	68.0
media:statistics	10	.0	.0	68.0
media:text	119	.0	.0	68.0
media:thumbnail	7017	1.0	1.0	69.0
media:title	4400	.6	.6	69.6
meta	4	.0	.0	69.6
mf:hasComments	330	.0	.0	69.7
misc	100	.0	.0	69.7
mn:channels	1	.0	.0	69.7
modified	25	.0	.0	69.7
moodys:docid	100	.0	.0	69.7
moodys:format	100	.0	.0	69.7
moodys:index_entry_child_id	217	.0	.0	69.7
moodys:index_entry_country	40	.0	.0	69.7
moodys:index_entry_industry	59	.0	.0	69.7
moodys:index_entry_ultparent_id	45	.0	.0	69.8
moodys:index_mode	100	.0	.0	69.8
moodys:issuer	110	.0	.0	69.8
moodys:issuer_list	100	.0	.0	69.8
moodys:moodys_org_id	100	.0	.0	69.8
moodys:org_id	110	.0	.0	69.8
moodys:price	100	.0	.0	69.8
moodys:printdate	100	.0	.0	69.9
moodys:report_class	100	.0	.0	69.9
moodys:report_type	100	.0	.0	69.9
moodys:sector	100	.0	.0	69.9
name	13353	1.9	1.9	71.8
news_date	15	.0	.0	71.8
newsid	10	.0	.0	71.8
next	4	.0	.0	71.8
ni:feed	1	.0	.0	71.8
nj:rubric	20	.0	.0	71.8
noscript	1	.0	.0	71.8
object	4	.0	.0	71.8

odat:availability	14	.0	.0	71.8
odat:detailimage	14	.0	.0	71.8
odat:image	14	.0	.0	71.8
odat:listDescription	14	.0	.0	71.8
odat:price	14	.0	.0	71.8
odat:priceCurrent	14	.0	.0	71.8
odat:priceRegular	14	.0	.0	71.8
odat:thumbnail	14	.0	.0	71.8
odat:tinyImage	14	.0	.0	71.8
on:short_desc	13	.0	.0	71.8
opensearch:itemsPerPage	1	.0	.0	71.8
openSearch:itemsPerPage	364	.1	.1	71.8
opensearch:startIndex	1	.0	.0	71.8
openSearch:startIndex	364	.1	.1	71.9
opensearch:totalResults	1	.0	.0	71.9
openSearch:totalResults	364	.1	.1	72.0
os:celebrant1	50	.0	.0	72.0
os:celebrant2	50	.0	.0	72.0
os:location1	50	.0	.0	72.0
os:location2	50	.0	.0	72.0
os:messages	50	.0	.0	72.0
os:photo	50	.0	.0	72.0
os:photos	50	.0	.0	72.0
os:simchaDate	50	.0	.0	72.0
os:simchaType	50	.0	.0	72.0
p	1233	.2	.2	72.2
pages	4	.0	.0	72.2
param	24	.0	.0	72.2
peers	20	.0	.0	72.2
pheedo:origEnclosureLink	2	.0	.0	72.2
pheedo:origLink	214	.0	.0	72.2
photo	10	.0	.0	72.2
pic	10	.0	.0	72.2
pingback:server	10	.0	.0	72.2
pingback:target	10	.0	.0	72.2
pink:filingType	10	.0	.0	72.2
pink:periodDate	10	.0	.0	72.2
pink:symbol	79	.0	.0	72.2

pink:type	79	.0	.0	72.3
poco:displayName	213	.0	.0	72.3
poco:familyName	213	.0	.0	72.3
poco:givenName	213	.0	.0	72.3
poco:name	213	.0	.0	72.4
poco:preferredUsername	213	.0	.0	72.4
posterous:author	20	.0	.0	72.4
posterous:displayName	20	.0	.0	72.4
posterous:firstName	20	.0	.0	72.4
posterous:lastName	20	.0	.0	72.4
posterous:nickName	20	.0	.0	72.4
posterous:profileUrl	20	.0	.0	72.4
posterous:userImage	20	.0	.0	72.4
postid	80	.0	.0	72.4
prev	4	.0	.0	72.4
print_pubDate	44	.0	.0	72.4
pubdate	60	.0	.0	72.4
pubDate	44162	6.2	6.2	78.6
pubDateNumber	94	.0	.0	78.7
pubitems:byline	5	.0	.0	78.7
pubitems:category	5	.0	.0	78.7
pubitems:image	5	.0	.0	78.7
pubitems:imageh	5	.0	.0	78.7
pubitems:imagew	5	.0	.0	78.7
pubitems:metadata	5	.0	.0	78.7
pubitems:priority	5	.0	.0	78.7
pubitems:subcategory	5	.0	.0	78.7
publish_date	44	.0	.0	78.7
published	14588	2.0	2.0	80.7
rank	17	.0	.0	80.7
rating	8	.0	.0	80.7
rdf:Bag	2	.0	.0	80.7
rdf:Description	1	.0	.0	80.7
rdf:li	475	.1	.1	80.8
rdf:RDF	19	.0	.0	80.8
rdf:Seq	18	.0	.0	80.8
rdf:value	2	.0	.0	80.8
rights	95	.0	.0	80.8

rss	1911	.3	.3	81.1
rss-url	80	.0	.0	81.1
rss:channel	1	.0	.0	81.1
rss:description	3	.0	.0	81.1
rss:image	2	.0	.0	81.1
rss:item	2	.0	.0	81.1
rss:link	3	.0	.0	81.1
rss:title	4	.0	.0	81.1
rss:url	1	.0	.0	81.1
s:counts	1	.0	.0	81.1
sb:created	20	.0	.0	81.1
sb:credit	14	.0	.0	81.1
scout:premiumflag	660	.1	.1	81.2
scout:sourcefriendlysubdomain	660	.1	.1	81.3
scout:sourcesiteid	660	.1	.1	81.4
scout:sourcesubdomain	660	.1	.1	81.5
script	4	.0	.0	81.5
section	10	.0	.0	81.5
seeders	359	.1	.1	81.5
seeds	20	.0	.0	81.5
series:name	5	.0	.0	81.5
short_description	20	.0	.0	81.5
shorthead	20	.0	.0	81.5
size	379	.1	.1	81.6
slash:comments	5871	.8	.8	82.4
small	5	.0	.0	82.4
smallImage	200	.0	.0	82.4
soup:attributes	400	.1	.1	82.5
source	6237	.9	.9	83.4
span	306	.0	.0	83.4
squareImage	200	.0	.0	83.4
state	15	.0	.0	83.4
stock	2	.0	.0	83.4
strong	76	.0	.0	83.4
subtitle	749	.1	.1	83.5
summary	4261	.6	.6	84.1
sx:history	213	.0	.0	84.2

sx:sync	213	.0	.0	84.2
sy:updateBase	11	.0	.0	84.2
sy:updateFequency	2	.0	.0	84.2
sy:updateFrequency	567	.1	.1	84.3
sy:updatePeriod	569	.1	.1	84.4
syn:updateBase	1	.0	.0	84.4
syn:updateFrequency	1	.0	.0	84.4
syn:updatePeriod	1	.0	.0	84.4
tagline	2	.0	.0	84.4
tags	110	.0	.0	84.4
textinput	2	.0	.0	84.4
textInput	9	.0	.0	84.4
thespringbox:skin	5	.0	.0	84.4
thr:total	10190	1.4	1.4	85.8
thumb	56	.0	.0	85.8
thumbnail	215	.0	.0	85.8
timestamp	124	.0	.0	85.9
title	66378	9.3	9.3	95.2
topix:comments	280	.0	.0	95.2
topix:rsslink	14	.0	.0	95.2
trackback:ping	25	.0	.0	95.2
tracker	20	.0	.0	95.2
ttl	453	.1	.1	95.3
ul	6	.0	.0	95.3
updated	15967	2.2	2.2	97.5
updatePeri	1	.0	.0	97.5
uri	9406	1.3	1.3	98.8
url	891	.1	.1	99.0
veranstaltungsort	300	.0	.0	99.0
video	9	.0	.0	99.0
votes	20	.0	.0	99.0
webmaster	1	.0	.0	99.0
webMaster	129	.0	.0	99.0
WebWizForums:feedURL	2	.0	.0	99.0
wfw:comment	10	.0	.0	99.0
wfw:commentRss	6598	.9	.9	99.9
width	199	.0	.0	100.0
wire:thumb	20	.0	.0	100.0

wn:adClassification	108	.0	.0	100.0
wn:wnreadableurl	1	.0	.0	100.0
wordzilla:id	15	.0	.0	100.0
xhtml:img	7	.0	.0	100.0
xhtml:meta	17	.0	.0	100.0
Total	**713434**	**100.0**	**100.0**	

C. Appendix C – BlogForever System Overview

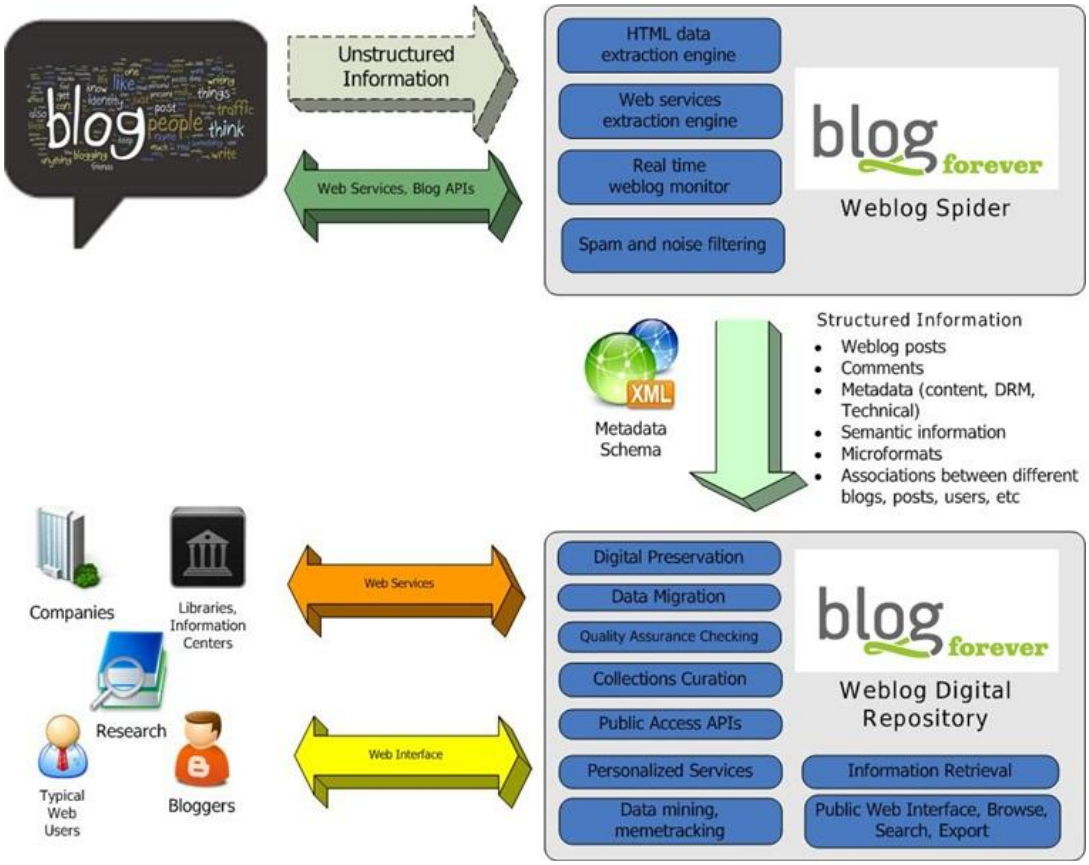

Figure 21 - BlogForever System Overview

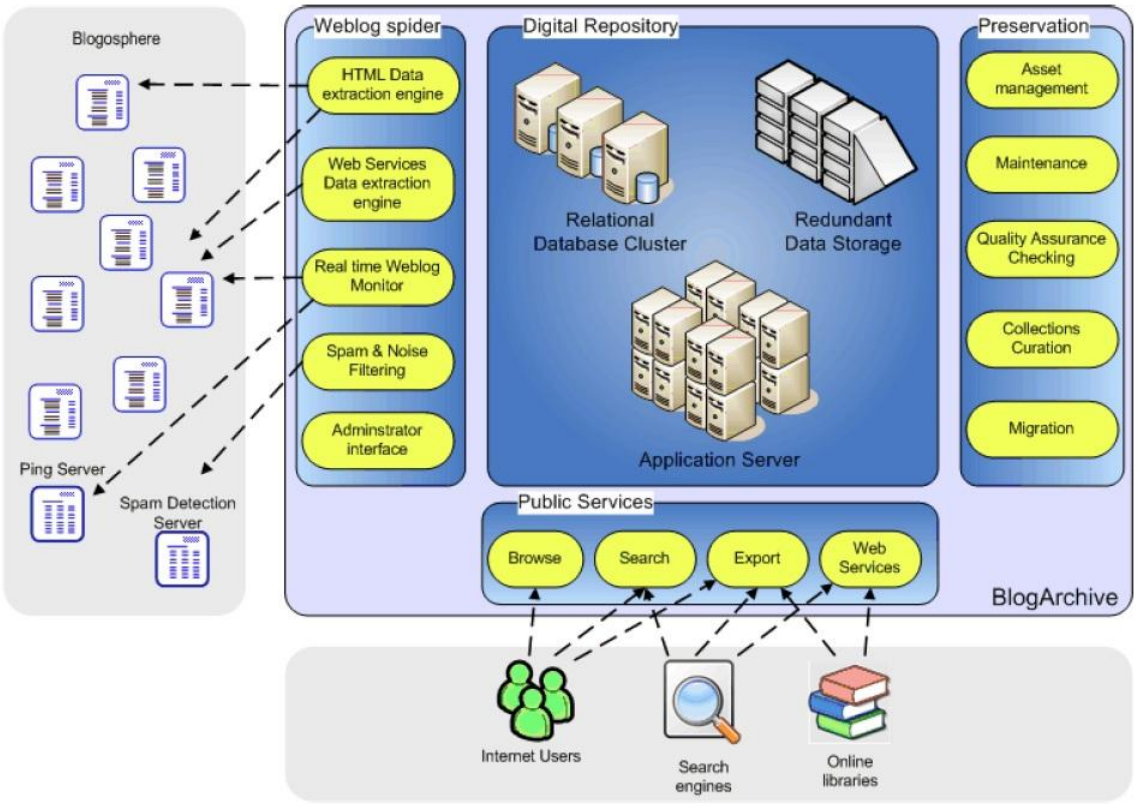

Figure 22 – BlogForever Proposed Architecture