Mapping the Blogosphere with RSS-Feeds

Justus Bross, Matthias Quasthoff, Philipp Berger, Patrick Hennig, Christoph Meinel
Hasso-Plattner Institute, University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
{justus.bross, matthias.quasthoff, office-meinel}@hpi.uni-potsdam.de
{philipp.berger, patrick.hennig}@student.hpi.uni-potsdam.de

Abstract— **The massive adoption of social media has provided new ways for individuals to express their opinions online. The blogosphere, an inherent part of this trend, contains a vast array of information about a variety of topics. It is thus a huge think tank that creates an enormous and ever-changing archive of open source intelligence. Modeling and mining this vast pool of data to extract, exploit and describe meaningful knowledge in order to leverage (content-related) structures and dynamics of emerging networks within the blogosphere is the higher-level aim of the research presented here. This paper focuses on this project's initial phase, in which the above-mentioned data of interest needs to be collected and made available offline for further analyses. Our proprietary development of a tailor-made feed-crawler meets exactly this need. The main concept, the techniques and the implementation details of the crawler thus form the main interest of this paper and furthermore provide the basis for future project phases.**

Keywords – weblogs, rss-feeds, network analysis, content analysis

I. INTRODUCTION

Since the end of the 90s, weblogs have evolved to an inherent part of the worldwide cyber culture [8]. In the year 2008, the worldwide number of weblogs has increased to a total in excess of 133 million [16] [18]. Compared to around 60 million blogs in the year 2006, this constitutes the increasing importance of weblogs in today's internet society on a global scale.

Technically, weblogs are an easy-to-use, web-enabled Content Management System (CMS), in which dated articles ("postings"), as well as comments on these postings, are presented in reverse chronological order [3]. Their potential fields of application are numerous, beginning with personal diaries, reaching over to knowledge and activity management platforms, and finally to enabling content-related and journalistic web offerings [9] [13]. This makes their point of origin indefinable.

One single weblog is embedded into a much bigger picture: a segmented and independent public that dynamically evolves and functions according to its own rules and with ever-changing protagonists, a network also known as the "blogosphere" [19]. A single weblog is embedded into this network through its trackbacks, the usage of hyperlinks as well as its so-called "blogroll" – a blogosphere-internal referencing system.

This huge think tank creates an enormous and ever-changing archive of open source intelligence [14]. However, the biggest congeniality of the blogosphere – the absence or independence of any centralized control mechanism –is meanwhile its biggest shortcoming: Modeling and mining the vast pool of data generated by the blogosphere to extract, exploit and represent meaningful knowledge in order to leverage (content-related) structures and dynamics of emerging social networks residing in the blogosphere seemed so far virtually impossible.

II. PROJECT SCOPE

Facing this unique challenge we initiated a project with the objective to map, and ultimately reveal, content-, topic- or network-related structures of the blogosphere by employing an intelligent RSS-feed-crawler. A crawler, also known as an ant, automatic indexer, worm, spider or robot, is a program that browses the World Wide Web (WWW) in an automated, methodical manner [10].

A feed is a standardized format, provided as RSS or ATOM by almost all content providers in the internet, to easily distribute content information or news about their website [17]. In the blogosphere, RSS-feeds are usually provided whenever a new post or comment is published in weblogs. Due to the standardized format of RSS-feeds, machines or program routines can automatically analyze them, and are consequently able to provide subscribers with updated and current content of these feeds. You could thus say that the sum of all feeds represents the network's entire structure.

To allow the processing of the enormous amount of content in the blogosphere, it is necessary to make that content available offline for further analysis. Our feed-crawler completes this assignment as described in the following sections.

While section three focuses on the crawler's functionality and its corresponding workflows, section four is digging deeper into the technical realization of the crawler. The time since notification of acceptance till final submission of this research paper was meanwhile used for further enhancements of the crawler - hereafter presented in section five. Section six is dedicated to related academic work that describes distinct approaches of how and for what purpose the blogosphere's content can be mapped. Based on these insights, we provide an outlook as well as recommendations

for further research with the overall objective to further enhance our crawler and to ultimately use the data collected to show interesting patterns and insights of segmented blogosphere networks. A conclusion is given in section nine, followed by the list of references and an appendix with attendant figures of the research discussed in this paper.

III. Original Crawler Functionality and Workflow

A. Action-Sequence of the Crawler

The crawler starts his assignment with a predefined and arbitrary list of blog-URLs. It downloads all available post- and comment-feeds of that blog and stores them in a database. It than scans the feed's content for links to other resources in the web, which are then also crawled and equally downloaded in case these links point to another blog. Again, the crawler starts scanning the content of the additional blog feed for links to additional weblogs. The crawler repeats this iterative process till it comes up against a link that was either already scanned, or till he comes across so-called "isles" – a smaller network of blogs that only link to each other and have no connection to the rest of the blogosphere. To avoid that the crawler gets stuck on one of these isles, we include blog URLs from different geographical regions, as well as blogs that cover diverse and unrelated topics as regards content in the arbitrary starting list. This furthermore increases the odds that the crawler covers the whole of the blogosphere within a minimum of time. With maximal content-related diversity in the arbitrary starting list, data can also be meaningfully analyzed in an early stage. By following-up links, it can be guaranteed that any URL of the worldwide top weblogs will sooner or later be crawled. The representation of the most influential opinion leaders is therefore feasible.

B. Recognizing Weblogs

Whenever a link is analyzed, we first of all need to assess whether it is a link that points to a weblog, and also with which software the blog is created. Usually this information can be obtained via attributes in the metadata of a weblogs header. It can however not be guaranteed that every blog provides this vital information for us as described before. There is a multitude of archetypes across the whole HTML page of a blog that can ultimately be used to identify a certain class of weblog software. By classifying different blog-archetypes beforehand on the basis of predefined patterns, the crawler is than able to identify at which locations of a webpage the required identification patterns can be obtained and how this information needs to be processed in the following.

Originally the crawler knew how to process the identification patterns of the three most prevalent weblog systems [1] around: MovableType, Blogger.com and Wordpress.com [11]. In the course of the project, identifications patterns of other blog systems will follow. In a nutshell, the crawler is able to identify any blog software, whose identification patterns were provided beforehand.

C. Recognizing Feeds

The recognition of feeds can similarly to any other recognition-mechanism be configured individually for any blog-software there is. Usually, a web service provider that likes to offer his content information in form of feeds, provides an alternative view in the header of its HTML pages, defined with a link tag. This link tag carries an attribute (rel) specifying the role of the link (usually "alternate", i.e. an alternate view of the page). Additionally, the link tag contains attributes specifying the location of the alternate view and its content type. The feed crawler checks the whole HTML page for exactly that type of information. In doing so, the diversity of feed-formats employed in the web is a particular challenge for our crawler, since on top of the current RSS 2.0 version, RSS 0.9, RSS 1.0 and ATOM among others are also still used by some web service providers. Some weblogs above all code lots of additional information into the standard feed. Momentarily, our crawler only supports standard and well-formed RSS 2.0 formats, of which all the information of our currently employed object-model is readout. It is the aim of our project team to include as many RSS-formats as possible in the future.

D. Storing Crawled Data

Whenever the crawler identifies an adequate (valid) RSS-feed, it downloads the entire corresponding data set. The content of a feed incorporates all the information necessary, to give a meaningful summary of a post or comment – thus a whole weblog and ultimately the entire blogosphere. General information like title, description, categories as well as the timestamp indicating when the crawler accessed a certain resource, is downloaded first. Single items inside the feed represent diverse posts of a weblog. These items are also downloaded and stored in our database using object-relational mapping[2] (refer to figure two in the appendix). The corresponding attributes are unambiguously defined by the standardized feed formats and by the patterns that define a certain blog-software.

On top of the general information of a post, a link to the corresponding HTML representation is downloaded and stored as well. In case this information is not provided in the feed information of a blog provider, we are thus still able to use this link information at a later point for extended analyses that would otherwise not be possible.

Comments are the most important form of content in blogs next to posts, and they are usually provided in form of feeds as well. However, we do need to take into account that a comment's feed-information is not always provided in the same form by all blog software systems. This again explains why we pre-defined distinct blog-software classes in order to provide the crawler with the necessary identification patterns of a blog system. Comments can either be found in the HTML header representation or in an additional XML attribute within a post feed. Comment feeds are also not

[1] http://wordpress.org, http://blogger.com, http://www.movabletype.org

[2] https://www.hibernate.org/

provided by every blogging system. With the predefined identification patterns, our crawler is however able to download the essential information of the comment and store it in our database.

Another important issue is the handling of links that are usually provided within posts and comments of weblogs. In order to identify network characteristics and interrelations of blogs within the whole of the blogosphere, it is not only essential to store this information in the database, but to save the information in which post or comment this link was embedded.

E. Refreshing Period of Crawled Data

How often a single blog is scanned by our crawler depends on its cross-linking and networking with other blogs. Blogs that are referenced by other blogs via trackbacks, links, pingbacks or referrers are thus visited with a higher priority than others by the crawler. Well-known blogs that are referenced often within the blogosphere are also revisited and consequently updated more often with our original algorithm. It can be considered possible that with this algorithm blogs of minor importance are visited rarely – a side-effect that we do not consider to be limiting at this time.

It would be fairly easy to put a different algorithm into action that might for instance make use of a ranking-score. With this "importance" score, it could be guaranteed that blog feeds that are considered as not that important in the community would not that often, but at least regularly be updated. Implementing a different algorithm can at all times be realized by substituting the so-called "scheduler" of our crawler (refer to figure one in appendix). It might prove to be necessary at a later time to apply certain heuristics on the blogs, to decide upon the sequence of their eradication.

F. Practical Experience

It is only a matter of minutes till the crawler finds a considerable amount of links that need to be analyzed. With a predefined starting list of only four weblog-URLs that hold a fairly good ranking within one of the major weblog ranking sites like Technorati[3] for instance, the crawler originally found up to several hundred links within 3-4 minutes. It should also be noted that with such a starting list of well-known blogs, the chance that the crawler ends up on one of the before-mentioned isles is minimal. As a matter of fact we did not experience such a dead end within this setting during our testing phase. If you choose four blogs of minor importance, the chance of ending up in a dead end is considerably higher, since unimportant blogs usually have a rather low degree of referencing and interlinking within the blogosphere.

We did also experience time critical limitation during our testing phase. Due to the enormous amount of links that are waiting for further analysis after only a couple of minutes the crawler is active, it takes time until a blog with all its postings and comments and the corresponding links is completely covered and downloaded. We acknowledge this time issue, since it just clearly demonstrates how widely ramified the blogosphere actually is. We therefore consider it more crucial in the first project phase that the crawler collects links of as much distinct blogs as possible, rather than covering blog's content in its entirety. At the time the crawler has identified a reasonable share of a (national) blogosphere, it might in a second phase than be more important to cover the entire content of blogs (refer to section V regarding "ongoing optimization efforts"). It might then make sense to adapt the crawling algorithm after a certain period of time accordingly. It is in any case essential to allow the crawler a minimum operational time to actually deliver meaningful data for further analyses.

IV. Implementation Details

The feed crawler software is implemented in Groovy[4], a dynamic programming language for the Java Virtual Machine (JVM) [7]. Built on top of the Java programming language, Groovy provides excellent support for accessing resources over HTTP, parsing XML documents, and storing information to relational databases. Features like inheritance of the object-oriented programming language are used to model the specifics of different weblog systems. Both the specific implementation of the feed crawler on top of the JVM, as well its general architecture separating the crawling process into retrieval scheduling, retrieval, and analysis as described in section three, allow for a distributed operation of the crawler in the future. Such distribution will become inevitable once the crawler is operated in long-term production mode. Both the structured nature of RSS and ATOM feeds and the best practices developed for weblog systems make the crawling of the blogosphere a task different from regular web crawling. Instead of indexing "documents" only, the feed crawler is aware of different types of documents in a weblog system, i.e., postings and comments, and can handle semantic relations between these different types of documents, such as a comment being a reply to a posting or another comment. Passing this information from retrieval to analysis and to the next round of retrieval scheduling, the information collected by the crawler will be much more valuable for the analyses we have in mind than regular web-crawler data.

V. Ongoing Optimization Efforts

We did observe several critical issues during the crawler's testing phase that apparently put a severe strain on its performance of.

It proved to be a major problem to download non-HTML content such as pictures or videos. Since the crawler follows any link on a webpage, it often downloads multimedia data exceeding 5 MB that does not contain any relevant information for our project. Downloading such huge data elements results in higher network traffic and consequently in lower crawler performance. Our solution was the inclusion of a so-called "black list" of file extensions like *avi*, *mov*, or *png*. Whenever the crawler encounters a link that points

[3] http://technorati.com/

[4] http://groovy.codehaus.org/

towards such a file extension, the downloading process is interrupted and an absolute term assigned as the content of that link in the database.

Another major issue was that the crawler can only handle valid RSS 2.0 feeds, because of which a considerable amount of blogs could not be downloaded and consequently analyzed at all. We therefore tested several frameworks (e.g. Project ROME[5]) to circumvent this problem, but ultimately realized that all these frameworks make too high demands on the validity of XML pages. Since the share of blogs not offering RSS 2.0 or ATOM feeds is minimal, it was decided to solely include the compatibility of ATOM feeds in our crawler implementation.

Contrarily to this cognizance, the share of those weblogs providing invalid feeds because of non-valid or inaccurate tags or other illicit character strings was exceptionally high.

The crawler makes use of the *XMLSlurper* embedded within Groovy, which is build upon the SAX implementation of Java. Since we are dependent upon this internal implementation, we can at this moment only react on bugs during the parsing process. A solution that allows us to analyze virtually all XML pages is therefore crucial (refer to section VIII). We also intend to optimize the crawler framework by possibly abandoning the currently employed hibernate library. Even though this library simplifies complexity by mapping java objects on the database, the framework seems not always be able to cope with the excessive data-collection of our crawler. Another enhancement was realized by extending the known blog-classes in our framework (refer to section III.B) to the ones of Serendipity and TypePad, as well as all those blogs that support the *XHtml Friends Network*[6].

We also intended to improve crawler-performance via the amplification of hardware resources employed. One approach we took was to rely on synergies of distributed computing. In doing so, our central server hosted the database while the crawler-software was running on up to three different clients downloading and parsing data to the central database. As can be inferred from figure three (see appendix), an increase in performance could indeed be observed. However, this increase was minimal, since the clients were wasting a considerable amount of time and internal resources in waiting for the central database to process their information.

This finding called for investments in central computing power. The new project hardware features a rack-server with 24 GB Ram and eight cores with each 2,4 GHZ. The first five days into operations this hardware boosted the number of processed jobs with factor 10. The indexing of those database-fields used within the WHERE-clause doubled this performance to meanwhile 500 blogs that the crawler finds – as opposed to 24 in the beginning. In summary, we could increase the performance of the crawler by factor 20 since the start of the project (refer to figure three in the appendix).

[5] https://rome.dev.java.net
[6] http://gmpg.org/xfn/

VI. Related Work

Certainly, the idea of crawling the blogosphere is not a novelty. But the ultimate objectives and methods behind the different research projects regarding automated and methodical data collection and mining differ greatly as the following examples suggest:

While Glance et. al. employ a similar data collection method in the blogosphere as we do, their subset of data is limited to 100.000 weblogs and their aim is to develop an automated trend discovery method for the blogosphere in order to tap into the collective consciousness of the blogosphere [6]. Song et al. in turn try to identify opinion leaders in the blogosphere by employing a special algorithm that ranks blogs according to not only how important they are to other blogs, but also how novel the information is they contribute [15]. Bansal and Koudas are employing a similar but more general approach than Song et al. by extracting useful and actionable insights with their *BlogScope*-Crawler about the 'public opinion' of all blogs programmed with the blogging software blogspot.com [1]. Extracting geographic location information from weblogs and indexing them to city units is an approach chosen by Lin and Halavais [12]. Bruns tries to map interconnections of individual blogs with his *IssueCrawler* research tool [4]. His approach comes closest to our own project's objective of leveraging (content-related) structures and dynamics of emerging networks within the blogosphere. His data set and project scope are however not as extended as ours, since he focuses on the Australian blogosphere that is concerned with debating news and politics.

VII. Outlook and Further Research

The feed crawling framework presented in this paper will allow for a variety of promising future applications. The most basic analysis will focus on the social structure exposed by the graph of interlinked weblogs. Per blog author or per region, interesting variables can be measured, e.g. the dominance of uni-directional links (e.g., followers linking to the opinion leader's blog) vs. balanced hyperlinks (e.g., authors mutually quoting each other), the frequency of new appearing postings, clustering coefficients etc. For this analysis, weblogs and postings can also be grouped by categories and user annotation [5] assigned by the authors, and the same variables can be measured per category or tag.

More advanced studies of the material gathered will involve temporal analyses. Special focus will be put on event detection: It is a frequently recurring phenomenon in the blogosphere that either an author starts a controversy in one posting, or several authors pick up topics from the general media, e.g. on politics, marketing, or other society-related topics. Afterwards, myriads of other webloggers start quoting these initial postings, comment on them, even do serious research work underpinning either side of the controversial debate, which frequently leads to the traditional media picking up the topic again. It has not yet been completely understood why and how the interest in certain topics grows, while other, probably equally important topics do not reach beyond occasional discussions in single

weblogs. To investigate such questions, the data gathered by our crawler can be used to track topics (i.e. keywords such as categories and tags) across hyperlinks, and also across inverse hyperlinks which cannot be accessed directly otherwise, as they are insufficiently mimicked by track-backs, link-backs etc. A large part of these studies will deal with an appropriate and meaningful visualization of these data. The visualization efforts are in a first step based upon the open source visualization tool flare [7] that will be configured to fit the needs of content representation in single weblogs and of larger subsets like national blogospheres. We will make use of an interactive visualization technique based on the 'Eigenfactor™ Metrics'[8] and called 'Well-Formed-Eigenfaktor', which are well-suited in providing graphical overviews about citation networks.

The third major direction of research deals with making the huge dataset created during the course of this project available to researchers and the public. This will be accomplished following the Linked Data principles [2]. The Semantically Interlinked-Online Communities (SIOC) project[9] provides an RDF vocabulary to expose the structure of online communities such as forums or blogs as RDF graphs. Publishing these valuable data in a standardized format will help a wide range of researchers from social scientists over computer network specialists to semantic web researchers to investigate various aspects of web communities such as the blogosphere, or the numerous national blogospheres.

VIII. LIMITATIONS

RSS-feed information of the blogosphere allows us to gain insights into the blogosphere that are not available elsewhere. Since the analysis of the data collected by our crawler is part of our project's second stage, it is out of scope for this paper. Our focus here mainly lies on describing the functionality and working-method of our crawler in detail. In doing so, our present crawler implementation constitutes a decent solution, sufficient for initial research in our research topic and for beginning data analyses, but is due to the following reasons not fully optimized yet:

Firstly, currently the crawler can only handle well-formed RSS 2.0 and ATOM feeds. RSS 0.9 or RSS 1.0 feeds, as well as feeds that have additional information coded into their standard information and are thus not well-formed are currently not supported by our crawler. Embedding a robust feed-parser framework therefore constitutes an essential element of future optimization efforts as described in sections III.C. and V.

An eventual configuration of our crawling algorithm might pose an additional challenge the project team. While the implementation of a new crawling algorithm can be quickly realized by substituting the "scheduler" of the crawler (refer to section III.C), it seems rather challenging to choose the algorithm in the first place. The current setting of the crawler dictates to gather as many links of diverse blogs as possible, rather than downloading the entire content of one weblogs before moving along to the next. It might therefore prove to be essential in a next step to make the crawling algorithm more intelligent and thus more sensible to our specific requirements regarding data collection to analyze network characteristics in the blogosphere. It is indeed important for us to get as many links from as many different weblogs as possible, but when a critical mass is reached, the crawler should understand that it is more important at this point to reproduce a blog's entire content. In case the crawler would have indefinite time to collect feeds of weblogs, it will eventually reflect the whole blogosphere. Regrettably, this issue is highly time-critical and the crawler might under circumstances not always have that time.

This is due to the fact that the number of visible feeds is predefined and limited within every blog-software. Even though this value can be changed after the software installation, no weblog usually displays more than 10-20 feeds in the corresponding list. This means that when a blog with a predefined feed-number of 10 for instance publishes its 11th post or comment, the feed of the corresponding first post published would fall out of that list and would as a consequence not be attainable for our crawler. Fortunately, the widely-deployed WordPress weblog software allows to query feeds arbitrarily back in time, and the WordPress module of our crawler knows how to issue these queries. It proves to be a major challenge to find a fair balance between minimum time a blog will not be visited and the objective to collect as many distinct blog-feeds as possible here.

IX. CONCLUSION

Generally, we try to investigate in what patterns, and to which extent blogs are interconnected. In doing so we want to face the challenge of mapping the blogosphere on a global scale, maybe limited to national boundaries. The visualization of link patterns, a thorough social network analysis, and a quantitative as well as qualitative analysis of reciprocally-linked blogs will to a large extend form the second part of our overall project, which will build upon the data collection method and technique described in this paper. We do expect the so-called "A-list blogs"- the most influential blogs- being overrepresented and central in the network, although other groupings of blogs can also be densely interconnected. We do however also expect that a majority of blogs link sparsely or not at all to other blogs - a notion we referred to as "isles" before, suggesting - contrarily to common belief - that the blogosphere is only partially interconnected. Currently, the main objective in our project work is to implement a meaningful revisiting algorithm for our crawler that allows us to analyze daily updated content of (partial) blogospheres.

[7] http://flare.prefuse.org

[8] http://eigenfactor.org/methods.htm

[9] http://sioc-project.org

REFERENCES

[1] N. Bansal, N. Koudas, “Searching the blogosphere”, Proc. 10th International Workshop on Web and Databases (WebDB 2007), June 15, Bejing, China, 2007, doi:10.1234/12345678.

[2] T. Berners-Lee, “Linked data”, 2006, Available: http://www.w3.org/DesignIssues/LinkedData.html

[3] J. Bross, A. Acar, P. Schilf, C. Meinel: “Spurring Design Thinking through educational weblogging”, Pro. 2009 IEEE International Conference on Social Computing, IEEE Press, Volume 4, 29-31 Aug. 2009, pp. 903 – 908, doi: 10.1109/CSE.2009.207

[4] A. Bruns, “Methodologies for Mapping the Political Blogosphere: An Exploration Using the IssueCrawler Research tool”, First Monday, vol. 12 no.5, 2007, available at: http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1834/1718

[5] B. Gaiser, T. Hampel, S. Panke, Good Tags - Bad Tags: Social Tagging in der Wissensorganisation, 1st ed. Waxmann, 2008

[6] N. S. Glance, M. Hurst, T. Tomokiyo, “BlogPulse: Automated Trend Discovery for Weblogs”, WWW 2004 Workshop on the Weblogging Ecosystem, ACM, New York, 2004, http://www.blogpulse.com/papers/www2004glance.pdf

[7] J. Gosling, B. Joy, G. Steele, G. Bracha, The Java Language Specification, 3rd ed. Amsterdam: Addison Wesley; 2005

[8] S. C. Herring, L. A. Scheidt, S. Bonus and E. Wright, "Bridging the Gap: A Genre Analysis of Weblogs," hicss, pp.40101b, Proceedings of the 37th Annual Hawaii International Conference on System Sciences - Track 4, 2004, http://www.ics.uci.edu/~jpd/classes/ics234cw04/herring.pdf

[9] H. Kircher, Web 2.0 - Plattform für Innovation. it - Information Technology (49) 1, Oldenbourg Wissenschaftsverlag, pp. 63-6, 2007.

[10] M. Kobayashi, K. Takeda, "Information retrieval on the web". ACM Computing Surveys (ACM Press) 32 (2): 144–173, 2000

[11] C. Leisegang, S. Mintert„Sieben frei verfügbare Weblog-Systeme – Liebes Tagebuch...“ iX, 7/2008, p. 42: Blogging-Software

[12] J. Lin, A. Halavais, “Mapping the blogosphere in America”, Presented at the Workshop on the Weblogging Ecosystem at the 13th International World Wide Web Conference (2004) http://www.blogpulse.com/papers/www2004linhalavais.pdf

[13] M. Ojala, Blogging for knowledge sharing, management and dissemination. Business Information Review, Vol. 22, No. 4. S. 269-276, 2005.

[14] J. Schmidt, Weblogs – Eine kommunikations-soziologische Studie., 2006, UVK Verlagsgesellschaft mbH.

[15] X. Song, Y. Chi, K. Hino, B. L. Tseng, “Identifying Opinion Leaders in the Blogosphere”, Proceedings of the sixteenth ACM conference on information and knowledge management (CIKM '07), pp. 971-974, ACM, New York, USA, 2007

[16] Technorati, State of the blogosphere, 2008, http://technorati.com/blogging/state-of-the-blogosphere.

[17] S. Thies, Content-Interaktionsbeziehungen im Internet. Ausgestaltung und Erfolg, 1st ed., Gabler, 2005

[18] Universal McCann, International Scoial Media Research Wave 3, 2008, http://www.universalmccann.com/Assets/2413%20-%20Wave%203%20complete%20document%20AW%203_20080418124523.pdf

[19] D. Whelan, In a fog about blogs. American Demographics, Vol. 25, No. 6, July/August, S. 22-23, 2003.

APPENDIX

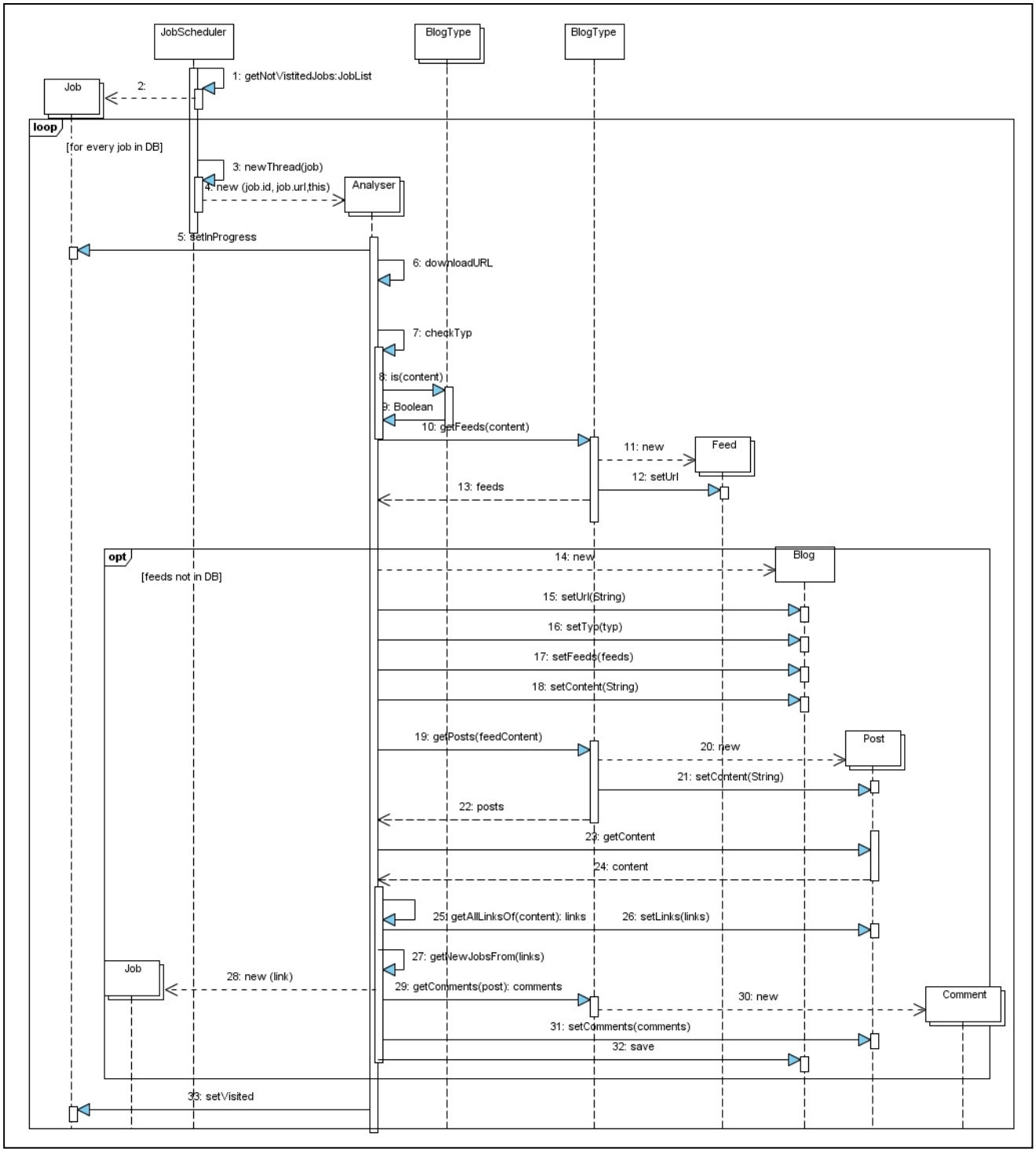

Figure 1. Action Sequence of RSS-Feed Crawler

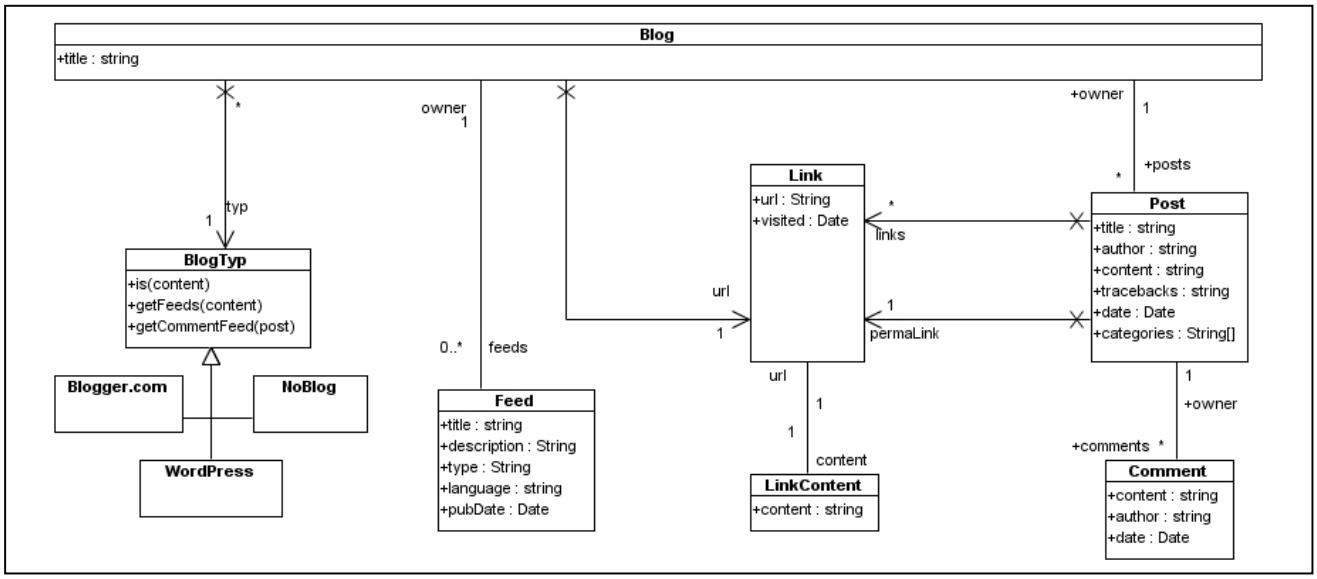

Figure 2. Data Structure

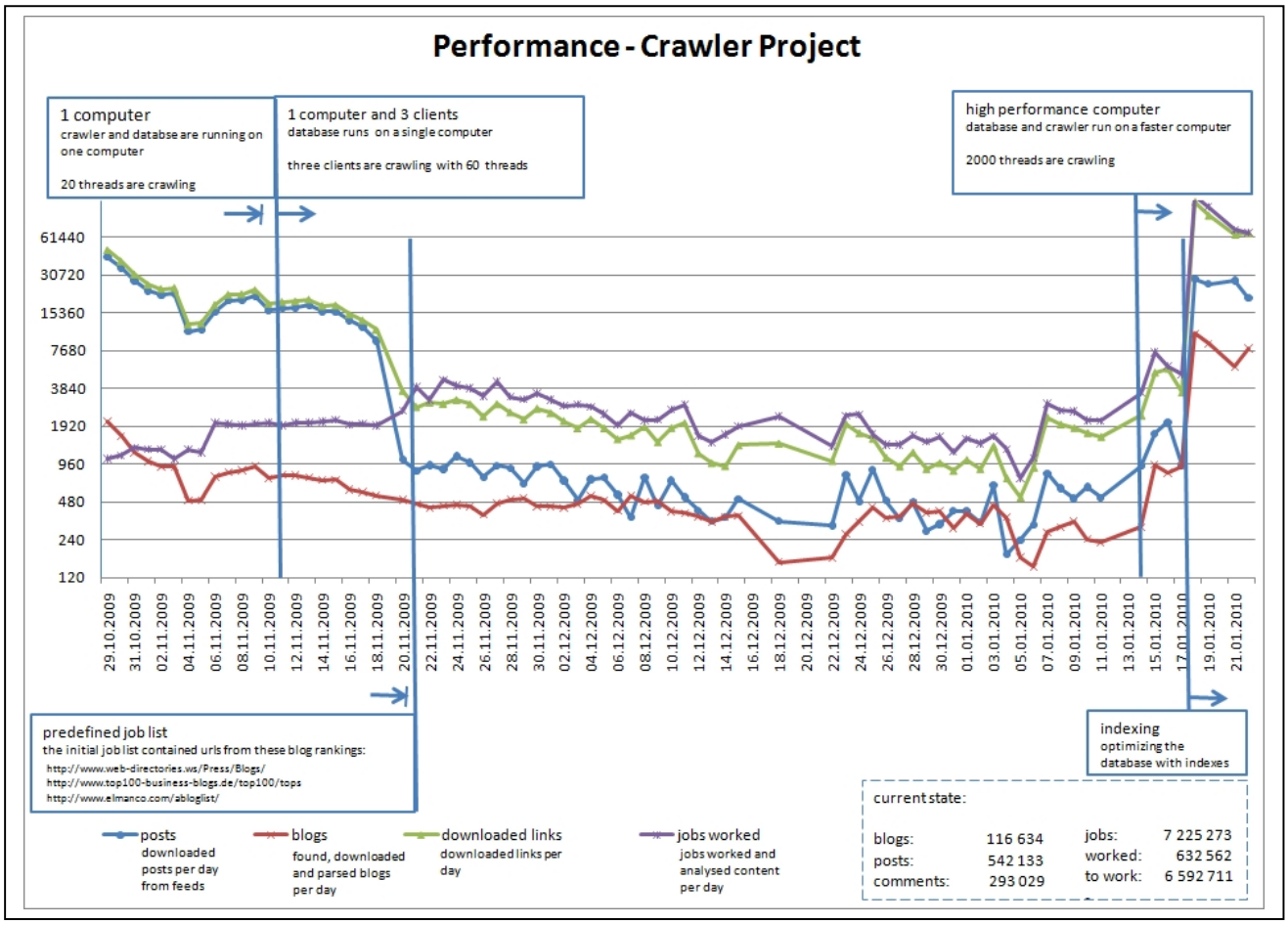

Figure 3. Performance Development of RSS-Feed Crawler (Note: Logarithmic Scale on y-Achsis)