Distributed feed networks for learning

UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>

Publisher
UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by Novática <http://www.ati.es/novatica/>, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática,* <http://www.ati.es/>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by Novática

UPGRADE was created in October 2000 by CEPIS and was first published by Novática and INFORMATIK/INFORMATIQUE, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>)

UPGRADE is the anchor point for UPENET (UPGRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
- Informatica, journal from the Slovenian CEPIS society SDI
- Informatik-Spektrum, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- ITNOW, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- Mondo Digitale, digital journal from the Italian CEPIS society AICA
- Novática, journal from the Spanish CEPIS society ATI
- OCG Journal, journal from the Austrian CEPIS society OCG
- Pliroforiki, journal from the Cyprus CEPIS society CCS
- Pro Dialog, journal from the Polish CEPIS society PTI-PIPS
- Tölvumál, journal from the Icelandic CEPIS society ISIP

Cover page designed by Concha Arias-Pérez and Diego Blasco-Vázquez
"Abacus" / © CEPIS 2008
Layout Design: François Louis Nicolet
Composition: Jorge Llácer-Gil de Ramales

Editorial correspondence: Llorenç Pagés-Casas <pages@ati.es>
Advertising correspondence: <novatica@ati.es>

UPGRADE Newslist available at
<http://www.upgrade-cepis.org/pages/editinfo.html#newslist>

ISSN 1684-5285

Monograph of next issue (August 2008)

"EUCIP: A Model for Definition and Measurement of ICT Skills"

(The full schedule of **UP**GRADE is available at our website)

CEPIS UPGRADE
The European Journal for the Informatics Professional
http://www.upgrade-cepis.org

Vol. IX, issue No. 3, June 2008

Monograph: Technology-Enhanced Learning
(published jointly with Novática*)
Guest Editors: *Carlos Delgado-Kloos and Fridolin Wild*

UPENET (UPGRADE European NETwork)

CEPIS NEWS

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

Distributed Feed Networks for Learning

Fridolin Wild and Steinn E. Sigurdarson

Recent studies indicate that blogs are the breakthrough user application of this decade. Yet, the blogosphere in its current form is suffering from various problems. The fuzziness of the audience, disconnectedness, fragmentation, and lack of conversational coherence may have their roots not only in sociological factors but also in technological shortcomings of the current infrastructure. These problems hinder an effective deployment of blogs in collaborative learning activities. Within this contribution, an interface specification for user-centred distribution of feed aggregation activities is proposed which is both a prerequisite and basic infrastructure for blog-based collaboration. By presenting an overview on the current state of the art in feed and interaction standards, a clear lack of support for active network management will be elaborated. The design requirements for a solution to fill this gap will be sketched and complemented by a step by step description of the communication process of the proposed "FeedBack" specification. Preliminary results from a trial with a reference implementation for WordPress provide a proof of concept.

Keywords: Feeds, Information and Communication Technologies (ICT), Technology-Enhanced Learning (TEL).

1 Learning with Blogs

The conglomerate of all blogs available online, the so-called "blogosphere", has been certified to show a bursty evolution at least since 2001, where an eruptive rise can be identified not only regarding metrics of scale but also with respect to deepening community structures and higher degrees of connectedness [1] [2]. As of December 2007, for example, the blog index and search site Technorati is indexing over 112 million blogs [3]. Blogging is obviously an increasingly popular phenomenon, although meta-studies reveal that between one half and two thirds of all blogs are abandoned within only two months after their creation [4].

One of the reasons that blogs became so attractive is their ease of use, removing barriers of technoliteracy from Web self-publishing [5]. There is a plethora of Web-publishing tools, allowing the user to choose from a large variety of (non-)commercial hosting services (often available free of charge). Moreover, users can set up their own Web-applications choosing from a rich portfolio of open- and closed-source products. Learning Light's eLearning Centre, for example, lists already back in 2006 more than 56 different products and online services in a vendor directory for blogging tools [6].

Publishing rich content with weblogs does not require any profound technical knowledge, such as language skills in the HyperText Mark-up Language (HTML) required to create pages with a desktop HTML editor, or skills such as those necessary to set up a fully-fledged Content Management System.

Other than virtual classrooms, wikis, or forums, blogs inherently offer the option to build open networks for collaboration, without the need to establish a dedicated community engaging in communication first. A single blog may be considered to be individual publishing; the blogosphere as a whole, however, is participatory by nature [7].

It is not surprising then, that blogs became vehicles for knowledge management to already often form an integral part of teaching and learning processes. Blogs can be used to organize lectures, seminars, and discussions both between teachers and students. Herring et al. [8] found in their study on blog genres that, from a random sample of 203 blogs, 57.5% of the authors were students on a secondary and tertiary level. However, at the same time, most of the blogs (70.4 %) were **personal journals** reporting on the lives of their authors. Clearly the minority of the blogs are deployed for filtering, i.e. commenting on external content and knowledge sharing. Similar results are reported by Schmidt & Mayer [9] in their end-2005 study among German speaking

Authors

Fridolin Wild, M.A., is researching within ProLearn, the EU Network of Excellence (NoE) for Technology-Enhanced Professional Learning, and additionally within EU IST funded iCamp project, there being the technical manager and leading a work package on interoperability of social software tools for learning. Fridolin is the treasurer of the European Association of Technology-Enhanced Learning (EATEL). He works as a scientist at the Institute of Information Systems of the Vienna University of Economics and Business Administration. <fridolin.wild@wu-wien.ac.at>.

Steinn E. Sigurdarson is a Technology-Enhanced Learning researcher at the Vienna University of Economics and Business Administration, working on interoperability and social software tools for learning within the iCamp project. Steinn has a background in data integration projects in the corporate sector with a focus on open-source developments. Since 2006 he has been concentrating on social software development, among others as phase leader in the software development of the Covcell Minerva-funded project. <sesigurd@wu-wien.ac.at>.

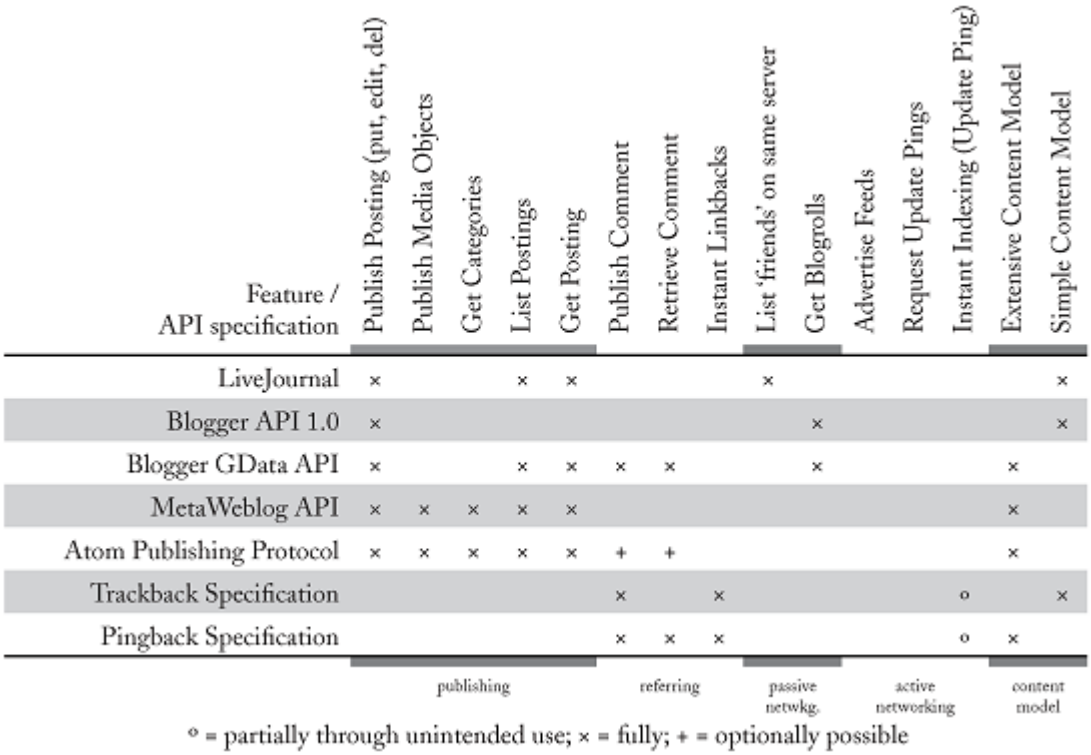

Figure 1: Functionality of APIs.

bloggers: users in education (pupils, students) are underrepresented among the blog writers with the (primal) aim for knowledge sharing (a.k.a. knowledge-bloggers = k-loggers); the major share of k-loggers stems from a work context.

The reasons why people create and maintain blogs vary to a large extent, however, always also including community building and social networking among the key motivations substantiated through empirical studies [10] [11]. Besides the obvious (group blogs), social networks were proven to exist between individual but networked blogs [12].

However, when deploying blogs in collaboration, many obstacles can be found that have not been overcome so far. To facilitate "productive blog conversations" which are necessary in knowledge management and learning, "*more carefully tailored socio-technical systems are needed*", as De Moor and Efimova claim [13]. They identify the problem of **notorious fragmentation** of conversations to be responsible for difficulties in reconstructing discussions, even for their own authors. Furthermore, they see (initial) **response times** as a problem that may slow down dialogues, especially when comparing to push technologies such as mailing lists.

Another identified obstacle to productive conversations in collaboration processes is the **low number of links** to blog posts, "*lower than often expected*". Only 51.2 % of all blogs link to other blogs, only 53.7 % link to other websites. 30.5 % of all blogs do not link to anything at all (besides badges) [8]. Entries received an average of 0.3 **comments**, with most of of entries receiving none. **Multimodality** poses yet another problem: replies and comments are often distributed across comment fields, but can also be found in the blogs of the repliers. Krause [14] identifies the **fuzziness of the audience** as a problem that may be responsible for failing discussion in his course experiment (his article is titled "When Blogging Goes Bad"), as it is unclear whether the desired audience (course participants) will be reached in time or at all.

To summarize, the fuzziness of the audience, disconnectedness, fragmentation, and lack of conversational coherence seem to be pressing problems that may have their roots not only in sociological factors but also in technological shortcomings of the current infrastructure. Moreover, blogs are rarely used *for*, although their users are most often *in*, education. They can be interpreted as clear indications of a lack of interoperability in the blogosphere.

The rest of this article is organised as follows. First, an overview on the current state of the art regarding feed and interaction standards is given, from which a clear lack of support for active network management is elaborated. In a next step, design requirements for a solution to fill this gap are elaborated and complemented by a step by step description of the communication process in the proposed specification. Subsequently, preliminary results from a trial with a reference implementation for WordPress are given. Finally, open issues and future possibilities are explored.

2 Feed and Interaction Standards

Blogs most commonly offer the possibility of data integration by sharing data in a common format: feeds are one of the important building blocks on the road to interoperability. The origins of these formats date back as early as 1995, although today only a few of them are still in use [14]. There have been various proposals for meta-data standards. Wittenbrink counts eleven different standards in altogether 30 different versions [14]. Today, however, there are three main meta-data standards for content syndication: RSS 1.0, RSS 2.0, and Atom. The basic mark-up vocabulary of these standards can be regarded as of similar expressiveness and all of them can be enhanced productively with extensions. Out of 37 blog and aggregator software products studied in [15], 65% supported RSS 1.0, 97% were RSS 2.0 compatible, and 70% had support for Atom.

When looking at the complementary building blocks, i.e. interaction standards, developers as well as users are confronted with an abundance of different standards. Current interaction standards can be grouped into functionalities for publishing, referring, passive networking, and active networking. Publishing subsumes functionalities with which postings, media objects and structuring information can be retrieved, added, updated, and removed remotely. Referring embraces commenting and linking features. Passive networking refers to identifying friends and their blogrolls, whereas active networking enables users to push feed offers, request update pings, and syndicate changes.

Looking more closely at Figure 1, today's interaction standards seem to concentrate primarily on publishing and referring functionalities, while the areas of passive and active networking are clearly less supported. Active networking features even are missing completely.

To conclude, the current lack of conversational coherence we have outlined above is a result of absent active networking support and limited passive networking facilities. An interaction standard directly addressing these functionalities is needed. Without advancements in infrastructure, tight collaboration can be achieved only at the users' expense.

3 Feed Management Specification

The process of collaborating via blogs can be divided into two independents sub-steps, i.e., the management of feeds and communication channels including authorisation and the exchange of items or item collections (the materialisation, the content transmission itself). The following section describes the missing link, a specification for managing feed subscriptions in a distributed setting which complements existing standards as analysed above. This specification is subsequently addressed as "FeedBack".

Aggregation services are already "abusing" the pingback specification in so far as they are using the pingback-derived weblog.ping XML-RPC[1] to inform about new and updated items and no longer inform about replies to existing blog postings. However, at the same time, there are no standardised options to inform a system about the existence of a feed and about updates to enable better synchronisation management.

This document proposes a set of XML-RPC to transport blog management information from one system to another. It is light-weight in so far, as implementation is made as easy as possible and dependencies on other components are reduced to a minimum. The whole communication process imitates human behaviour and shifts control to the user wherever possible.

The extensive performance analysis of push versus pull described by Deolasee et al. (2001) shows that for small temporal coherency requirements, pulling bears performance disadvantages [16]. Push-based data synchronisation provides several advantages beyond pull-based ones, the most important one being immediacy. Pulling requires two steps of communication, while pushing maintains state information and sends data only when necessary: it preserves information about clients' interests and pushes only relevant information. Pull interactions require many queries without effect, as weblogs usually do not change very rapidly (but at different times).

As a consequence, pulling causes a larger communicative network load, especially with a larger number of clients. In a push approach, action is performed only when needed. Yet in the inherent immediacy of a push-based approach may lie its greatest pitfall if deployed in a large, distributed system. Depending on the volatility of data involved in a large distributed system, as well as on the number of nodes requiring synchronisation, a push-based approach may be affected by excessive overhead of minor notifications being generated and transmitted. This problem is generally solved by using an intelligent buffer system combining several update notifications in a given time window (a "buffered push" approach).

The FeedBack specification dictates four distinct properties for a compliant system. First, for discovery purposes a FeedBack-enabled system should be able to return a Uniform Resource Identifier (URI) to its respective FeedBack-enabled XML-RPC-endpoint either using a HTTP header called "X-Feedback" or by having a `link` element in every FeedBack-enabled page, with the `rel` attribute set to "FeedBack". The endpoint provided should support the following three methods:

- `feedback.offer`: used for advertising a feed to the owner of a particular URI at the endpoint.
- `feedback.request`: used to register for update notifications.
- `feedback.notify`: used to inform about existence and content of updates.

Systems willing to communicate detect the endpoint first via the HTTP header or the `link` element. Subsequently, they enter a communication process using the three methods mentioned above. Using these four steps; discovery, offer, request, notification, the specification provides the means for simple, secure push-based feed subscriptions.

[1] XML Remote Procedure Call.

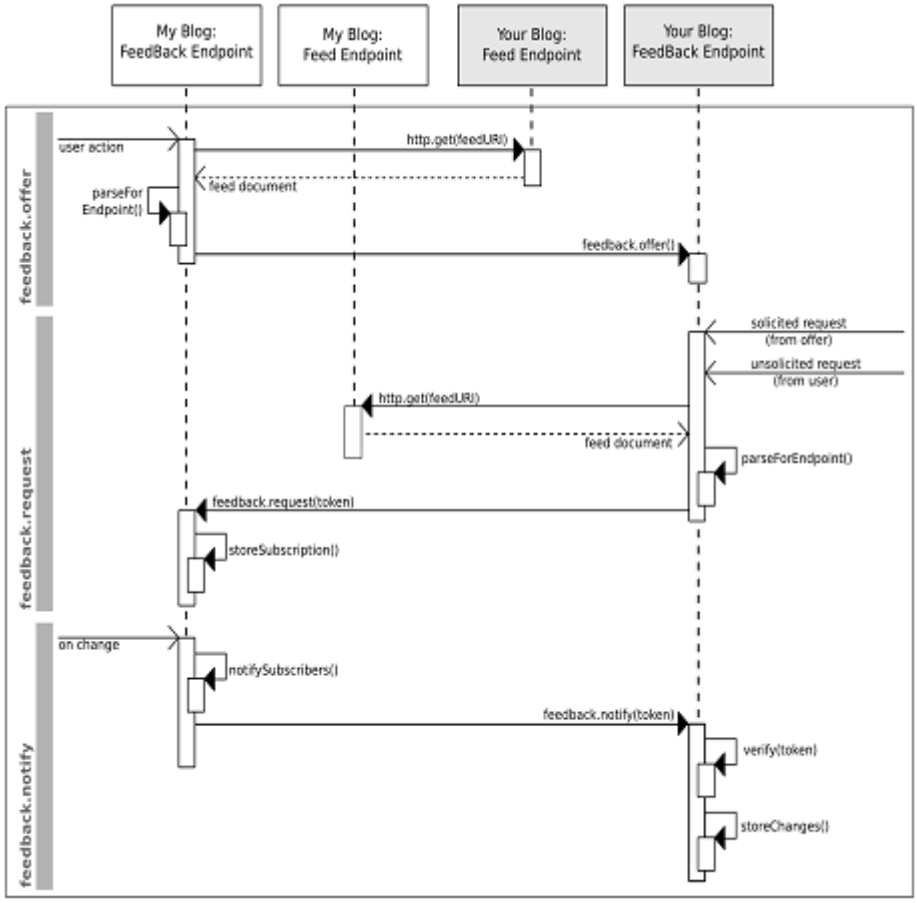

Figure 2: Communication Process with FeedBack.

4 Communication Process

In the following section, we describe data exchange, interaction steps, and state-transitions between two blog systems and their users, when communicating management information about a particular feed and specific items in this feed.

FeedBack can be used to facilitate two modes for entering into feed subscription. Subscriptions can be solicited, where there has been an official offer of a feed from the source system to the target system. However, subscriptions can also be unsolicited, where the target system directly sends a request for feed updates.

Below, the communication process is described in the form of a full solicited FeedBack subscription and update notification cycle. Note that the first step ("offer") is optional and can be left out. The description vocabulary should be familiar from Figure 2, and refers to a fictional scenario of two blogs: "My Blog" which will be the source of updates in this scenario, and "Your Blog" which will be the recipient of the update notifications.

The offer and request calls executed between the two systems are the ones necessary for such a relationship to be established securely. Each step will contain a small block of easily understood pseudo-code in order to clarify further the mechanism. Since the FeedBack specification is a server-side Application Program Interface (API), all the functionality offered in this scenario is accessed from the control panel of each respective system.

To ease understanding, the following description of the communication process is phrased as if one of the authors of this article was addressing you, the reader, directly.

4.1 Offer Subscription

If I want to use FeedBack to share my blog feed with you, I go into the control panel of My Blog, and tell it to offer Your Blog a subscription to the feed of My Blog. I do this by entering the URI of Your Blog (ideally any dynamic page belonging to Your Blog).

My Blog reads the source of the given URI, to try and find either an X-Feedback HTTP header, or a link element containing a URI with a valid XML-RPC endpoint supporting the FeedBack methods. Once it finds a valid URI, it will proceed to creating an instance of Your Blog's endpoint and make a `feedback.offer` call.


```
xrpc = new xmlrpc("http://your.blog.org/xmlrpc")
xrpc.feedback.offer(
   "http://your.blog.org/uri",[2]
   "http://my.blog.org/advertised_feed"[3]
)
```

The feed My Blog offers, must of course adhere to the same conditions as any FeedBack-enabled system, and have the X-Feedback HTTP header (or the above mentioned `link` element) to provide a valid XML-RPC endpoint, for Your Blog to interact with. Upon receiving the subscription offer, Your Blog should fetch the offered feed, store some of its metadata details to assist you in deciding whether or not to subscribe, and save the XML-RPC endpoint URI provided for later interaction purposes in case you accept the offer.

4.2 Subscription Request

Your Blog has displayed a pending subscription offer from My Blog. If you decide to accept this offer, Your Blog will make a `feedback.request` call to the XML-RPC endpoint found in the feed offered. The request will tell the XML-RPC endpoint of My Blog which feed is being requested, where (to which XML-RPC endpoint) to send update notifications, and Your Blog will generate a token to securely identify the future update notifications as they arrive.

```
xrpc = new xmlrpc("http://my.blog.org/xmlrpc")
xrpc.feedback.request(
   "http://my.blog.org/advertised_feed",[4]
   "http://your.blog.org/xmlrpc",[5]
   "098f6bcd4621d373cade4e832627b4f6"[6]
);
```

In case of the `feedback.request` being made directly, without the offer step, then the natural way for Your Blog to handle it would be to first see if the feed you are requesting supports FeedBack. If not, or if the endpoint provided does not support the `feedback.request` method, it is recommended that Your Blog's FeedBack implementation gracefully degrades to pulling the feed at regular intervals.

4.3 Update Notifications

At this stage, Your Blog has made a `feedback.request` to My Blog, and when I post a new entry which should appear in the feed Your Blog has requested, My Blog will send an update notification, via the `feedback.notify` method:

```
xrpc = xmlrpc.server("http://your.blog.org/xmlrpc")
xrpc.feedback.notify(
   "http://my.blog.org/post/data.xml",[7]
   "098f6bcd4621d373cade4e832627b4f6"[8]
);
```

Your Blog looks up the token received by the update notification to find the corresponding subscription. If Your Blog supports several ways of subscribing to updates via FeedBack, it should look up based on the token how to handle this particular notification. Possible reactions would include re-fetching the original feed, inserting, deleting, or updating relevant entries found. This approach is the simplest to implement, and so it may be the most common.

Other reactions could include fetching the data pointed to by the URL in the payload parameter, to find, for example, a transaction log of changes to the feed. In the case of blogs, this could be a feed containing more detailed update information as offered by Atompub [17].

5 Reference Implementation

To demonstrate the applicability of the missing link "FeedBack", a reference implementation has been developed for the popular blogging environment WordPress. The plug-in can be downloaded via SourceForge [18]. Currently, the use of FeedBack is investigated in a field trial within the EU funded research project ICAMP. So far, 17 blogs have been registered using the FeedBack plugin resulting in 68 offers reflecting in 45 accepted offers. A total of 469 update notifications were received by the blogs, whereas 504 notifications were sent by the blogs (also including notifications to other sites).

A validation service to test compliance with the specification has been set up at [19] supporting developers in writing new implementations.

6 Conclusion and Outlook

FeedBack complements existing standards and specifications in the blogosphere to support articulation work for managing offers, subscriptions, and update routines in collaborative learning processes. It is designed to be simple and robust and can be applied for almost any data synchronisation task where systems are subject to dynamic usage scenarios of social software, heterogeneity of systems is common, and complexity is not required at the transport level.

Future work needs to explore the applicability of FeedBack in more depth for complex, cascading processes and with different data formats.

[2] The URI I entered in my control panel, which My Blog used to find Your Blog's XML-RPC endpoint.
[3] The URI to the feed at My Blog which I wish to share with you.
[4] In this case the requested feed is naturally the same one as My Blog offered, and I wished to share.
[5] My Blog has no way of knowing where to send the update notifications without being provided a target XML-RPC endpoint.
[6] The token generated by Your Blog to identify incoming update notifications.
[7] The payload part should point to a URI containing the updates which Your Blog should be aware of.
[8] The token originally supplied to My Blog by Your Blog, identifying that this update is desired.

References

[1] R. Kumar, P. Raghavan, J. Novak, A. Tomkins. "On the Bursty Evolution of Blogspace". Proceedings of the WWW2003, Budapest, Hungary, ACM Press, pp. 568 – 576. 2003.

[2] D. Sifry. "The State of the Live Web". April 2007. <http://www.sifry.com/alerts/ archives/000493.html>, last access: Jan 9th, 2008.

[3] Technorati: About Us <http://technorati.com/about/>, last access: Jan 10, 2008.

[4] D. Gurzick, W. G. Lutters. "From the Personal to the Profound: Understanding the Blog Life Cycle". In: Proceedings of the CHI 2006, Montreal, Canada, ACM Press. 2006.

[5] M. Tepper. "The Rise of Social Software". In: netWorker 7(3), ACM Press, pp. 18 – 23. 2003.

[6] eLearning Centre: Products & Services: Blogging Tools, eLearning Centre, Learning Light. 2006. <http://www.e-learningcentre.co.uk/eclipse/vendors/weblogging.htm>, last access: July 5, 2007.

[7] T. O'Reilly. "What is Web 2.0". 2005. <http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, last access: March 17, 2008.

[8] S. Herring, L. Scheidt, S. Bonus, E. Wright. "Bridging the Gap: A Genre Analysis of Weblogs". In: Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS-37), IEEE Computer Society Press. 2004.

[9] J. Schmidt, F. Mayer. "*Wer nutzt Weblogs für collaborative Lern-und Wissensprozesse?*", In: Dittler, Kindt, Schwarz (Eds.): *Online-Communities als Soziale Systeme*, Waxmann, New York/München/Berlin. 2007.

[10] B. Nardi, D. Schiano, M. Gumbrecht, L. Swartz. "Why We Blog". In: Communications of the ACM 47(#2), ACM Press, pp. 41– 46. 2004.

[11] D. Schiano, B. Nardi, M. Gumbrecht, L. Swartz.. "Blogging by the Rest of Us". In: Proceedings of the CHI 2004, Vienna, Austria, ACM Press. 2004.

[12] A. Chin, M. Chignell. "A Social Hypertext Model for Finding Communities in Blogs". In: Proceedings of the Hypertext 2006 (HT'06), Odense, Denmark, pp. 11 – 22. 2006.

[13] A. De Moor, L. Efimova. "An Argumentation Analysis of Weblog Conversations". In: Aakhus, Lind (eds.): Proceedings of the 9th International Working Conference on the Language-Action Perspective on Communication Modelling (LAP 2004), New Brunswick, USA, pp. 197-211. 2004.

[14] S. Krause. "When Blogging Goes Bad: A Cautionary Tale about Blogs, Email Lists, Discussion, and Interaction". 2004. <http://english.ttu.edu/KAIROS/9.1/praxis/krause/index.html>, last access: January 11, 2008.

[14] H. Wittenbrink. "*Newsfeeds mit RSS und Atom*", Galileo Press, Bonn. 2005.

[15] F. Wild. "An Interoperability Infrastructure for Distributed Feed Networks", Deliverable D3.3, iCamp Consortium. 2007.

[16] P. Deolasee, A. Katkar, A. Panchbudhe, K. Ramamritham, P. Shenoy. "Adaptive Push-Pull: Disseminating Dynamic Web Data". ACM. 2001. <http://www10.org/cdrom/papers/ pdf/p269.pdf>, last access: January 14, 2008.

[17] J. Gregorio, B. de hOra. "The Atom Publishing Protocol". RFC 5023, IETF. 2007. <http://tools.ietf.org/html/rfc5023>, last access: January 14, 2008.

[18] S. Sigurdarson, F. Wild, A. Soylu. FeedBack Wordpress Plug-in. 2007. <http://sourceforge.net/project/showfiles.php?group_id=191261&package_id=224627>, last access: January 14, 2008.

[19] S. Sigurdarson. FeedBack validator. 2007. <http://validator.icamp.eu>, last access: January 14, 2008.

