

KlogMS – Semantic Knowledge Chunking

Andy CHUN, Hon Wai & LAI, Ho Kwan
Department of Computer Science
City University of Hong Kong
Tat Chee Avenue, Kowloon
Hong Kong SAR

andy.chun@cityu.edu.hk & kwan@finger-style.com

ABSTRACT

This paper introduces our current research progress in creating a new architecture and framework for Knowledge Management System (KMS) that borrows and builds upon ideas from the highly successful blog and wiki technologies. We call our framework KlogMS (Klog Management System). Klog (knowledge weblog) is the term commonly used for weblogs that contains content for knowledge sharing, usually professional opinions on a particular topic. Just like personal blogs, these klogs can be highly cross-linked with other related klogs, forming a virtual community of "expert" opinions, enabling the readers to be exposed to all points of view. Our research focuses on creating a new type of KMS framework that supports klogging within an organization for the purpose of internal knowledge sharing, borrowing ideas and features found in blog and wiki frameworks and from AI and psychology. The core technology behind KlogMS is an engine that performs what is equivalent to the concept of "knowledge chunking" on this repository. The paper explains the design objectives behind KlogMS, requirements for klogging as a KM tool, our architecture design, the knowledge chunking process, and the state of our prototype.

Keywords: Knowledge Management System, Weblogs, Blogs, Klogs, Wikis, Collaboration.

1. INTRODUCTION

It is common knowledge that the top most critical problem in implementing a Knowledge Management System (KMS) within any organization is employees' reluctance in sharing their knowledge. Some argue that this is due to employees' fear that they will lose control and power if they share their unique knowledge and skills with others. Seeing the vast amount of knowledge that is shared daily through blogs and wikis, this does not seem to be true. We believe that, in general, people are in fact very willing to share what they know with others and to help others to learn - given the right set of tools and motivation.

The term "knowledge management system" (KMS) may often conjure up different images for different people [WILS2002]. To some, KMS simply means a modernized version of a document management system (DMS), with newer technologies and more sophisticated features. For example, Microsoft's SharePoint Portal Server is considered by some as the "knowledge management for the masses" [ASHE2001]. SharePoint has all the basic DMS features, such as version control, check-in/check-out, etc., plus advanced search tools and within a dashboard/portal model [Share2004].

Others may consider KMS as an advanced version of an Intranet Portal with collaboration capabilities. For example, Process Based Mission Assurance KMS (PBMA-KMS), (<http://pbma.hq.nasa.gov>) deployed in 2001, is NASA's first KMS. PBMA-KMS supports mission safety and success by providing a platform to share knowledge. This Intranet portal contains over 200 documents, over 110 interviews or "video nuggets," and over 1,000 links. Advanced collaboration features include document sharing, online communities, group calendars, threaded discussions, polling, instant messaging, etc. [LIEB2002, NEWM2020].

Is advanced document management or Intranet portal combined with collaboration features enough to meet the objectives of knowledge management within a corporation? Many researchers believe KMS should be more and may require different set of tools and technologies. Static documents alone does not necessarily equate to knowledge [GREY2004]. Grey argues that for KMS to work, you need to have a "community that can appreciate the context, understand the issues, talk the language, adopt the assumptions, share the tricks, interpret and adapt the explicit stuff to changing external circumstances." He also points out some of the issues with KMS [GREY2003] - how to get people to trust and use others' knowledge, how to provide an environment where knowledge can grow organically, how to use context to determine applicability of knowledge, how to keep knowledge up-to-date with feedback from usage, etc.

Our KlogMS research also addresses some of these same issues and builds upon the successes of blogs and wikis as popular platforms for knowledge sharing. KlogMS aims at addressing some of the key problem areas within current corporate Knowledge Management System (KMS) implementations - employee reluctance in entering and maintaining knowledge [WIKIKM] and ability to quickly find relevant knowledge. For our KlogMS research, we are more interested in tackling the problem of collecting, sharing and maintaining dynamic changing knowledge within a corporate environment; requirements such as document management will therefore not be addressed by KlogMS. In addition, we are more interested in creating KM technologies that are practical and deployable in the near future. Therefore research into AI knowledge representation, automated knowledge extraction and internalization from natural language text will not be covered.

2. RELATED RESEARCH

Our work on KlogMS is related to current research on many fronts - knowledge management, blogging, and wiki. In particular, we focus on two main research areas - how to represent free text in a form that can be useful as "knowledge" within a corporation and how to form larger more meaningful collections of knowledge from these pieces of text, i.e. "chunking."

Our KlogMS work share a lot of similar ideas with concepts presented in [BAUS2002], which talks about "Using Blogs in Business" in Chapter 8. That chapter explains how blogs or klogs can be used as a tool for KM within a corporate environment and suggests that klogs can be more useful as a KM tool if they are better integrated with other office applications and tools such as email and instant messaging (IM).

In KlogMS, each piece of "knowledge" is entered as a separate klog entry or "post." We view each post as a "knowledge chunk." The knowledge chunks are stored and managed in our enterprise knowledge repository. The core technology behind KlogMS is an engine that performs what is equivalent to the concept of "knowledge chunking" on this repository. The key to the success of a KMS to be able to create a Complex Adaptive System (CAS) that can take on a life of its own. We believe "knowledge chunking" to be the key catalyst for this to happen. Chunking is, of course, the process of grouping or reorganizing smaller units into larger, more meaningful ones. In terms of klogs, this means "weaving" or linking related klogs together to form larger bodies of knowledge. This "weaving" process with new interconnections may bring about unpredicted, emerging behaviors that is typical of CAS.

Knowledge Chunks

In KlogMS, knowledge entries are referred to as "knowledge chunks," which are ubiquitous recursive structures within KlogMS. The structure of knowledge

chunks can be viewed as a form of rich blog structure that includes semantic metadata as well as traditional features such as user ranking, trackback, comments, etc. We will explain more on the KlogMS design in the next section. These "knowledge chunks" can be thought of as extended klog (knowledge weblogs) structures.

Knowledge Chunking

Besides representing knowledge, another key KlogMS research is in the "knowledge chunking" process. KlogMS can be thought of as a type of Microcontent Management System (mCMS) with a "knowledge chunking" engine, which we call the KCEngine. Our KlogMS "knowledge chunking" process aims at "weaving" a knowledge chunk entry into our knowledge base. The basic mechanism for this is the ability to find and create links to and from the current entry to existing entries and categories. In fact, many believe that links are the main power behind the Web and the source of knowledge [WALK2002]. Our "chunking" process is done through a combination of manual/automated categorization and dynamic topic group formation (a la TopicExchange (topicexchange.com)).

The result from this "chunking" process is the ability to view a collection of highly related klog entries. Each entry may belong more than one dynamically formed "knowledge contexts." Each knowledge context represents a particular perspective into our knowledge base.

Our "knowledge context" is in some ways similar to the concept of reBlogging [SEB2004], except that it is automated. This concept comes from Eyebeam (www.eyebeam.org/reblog/), which has a software to allow a person to manually reorganize and republish "interesting" blogs based on a particular theme.

Another technology that deals with grouping blogs is TopicExchange (topicexchange.com). The concept behind TopicExchange, a TrackBack-powered conversation server, originated with Sébastien Paquet [SEB-BLOG] and his "Ridiculously Easy Group Forming" concept [SEBRIDI]. TopicExchange hosts "topic channels" that serves as public archives of blogs on defined topics. Blogs are posted to a channel via TrackBack.

TrackBack is, of course, the technology invented by Movabletype (www.movabletype.com) that allows people to "ping" a server and tell it that they have written something that is related to a particular post. The server can then take a summary of the blog comment and posts it along with the comments on the original post. This technology can be used to support "remote comment" and "content aggregation."

A related technology is Pingback [PING2002], in which web publishing software will automatically inform the other relevant parties on behalf of the user that the user is linking to their blogs. Both TrackBack and Pingback allows for bi-directional links between blogs.

We call his natural formation of linked structures the "blog cohesion" factor. One way to measure this is in terms of levels of links from the original blog. Basically, level one is the number of related blogs that are linked to a particular blog. Level two is the total number of related blogs that are linked to all those blogs in level one, and so on. The higher the cohesion factor, the more important or interesting is the blog.

We believe the ability to naturally evolve clusters of ideas is key to chunking and for knowledge sharing and knowledge management to work. It also forms a key mechanism to turn klogs into complex adaptive systems (CASs) - emerging self-organizing systems. [CROSS2004] points out some interesting characteristics of complex adaptive systems:

1. Many parts, connected in multiple ways
2. Diverse components interacting both serially and in parallel
3. Spontaneously self-organizing
4. Can't be reverse-engineered
5. Interaction of parts changes the whole
6. Open, adaptive, evolving
7. Emergence occurs far from equilibrium, on the edge of chaos

For knowledge management, interactions within CAS can evolve new ideas from a collection of smaller ideas. This process of forming natural clusters or "self-assembling" [MOUL2002] is related to AI clustering techniques used in data mining, except the clusters are formed "organically" through linking. A data mining "cluster" is a collection of data objects that are "similar" to one another and thus can be treated collectively as one group. Usually, this is done in an automated unsupervised manner [BERK2002].

Part of our KlogMS research will be to investigate the use of these automated clustering approaches to klogs. Another related research is in statistical or Bayesian filtering. A common use of this technique can be found in spam filters [GRAH2003, GRAH2002] that use statistical approaches to find spam that are similar to learned examples of previous spams. Graham reports a very good filtering rate of 99.75% with his approach. In KlogMS, statistical approaches may also be useful in finding related klogs, similar to Google's "Similar Pages" feature [GOOG2004].

Most blogs support the notion of "categories" [MACI2002], as another way to organize and view blogs. Categories allow other people to subscribe to only a portion of your weblogs that are related to a common topic. Categories are usually manually specified when the blog was created.

Moreover (<http://w.moreover.com/>) on the other hand, uses a proprietary rule-based categorization process that automatically filters web content into categories. The web content can be any content, not just blogs. However, a manually created site-specific "template profiles" must be created. According to Moreover, it has over 300 pre-built industry categories as well as over 10,000 customizable

company specific categories. KlogMS will also maintain an internal list of categories that can be extended.

Our KlogMS research work is also related to the "semantic blogging" research performed by HP's Steve Cayzer [HPa] and his team [HPb]. "Semantic blogging" is built upon semantic web technologies and attaches semantics as blog metadata. Semantic blogging can improve the blogging experience in many ways - "semantic view" allows different ways to view a set of blogs, "semantic navigation" allows different ways to navigate around the blogosphere, and "semantic query" allows relevant information to be retrieved based on semantics.

Our KlogMS research is in ways also related to UK's AKT - the Advanced Knowledge Technologies project (<http://www.aktors.org/akt/>) AKT is being funded by the Engineering and Physical Sciences Research Council (EPSRC) (www.epsrc.ac.uk) with a six-year budget of £7.5 million pounds. It is an interdisciplinary research collaboration between internationally recognized research groups at the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University. AKT's main project aim is to develop and extend a range of technologies to provide integrated methods and services for the capture, modeling, publishing, reuse and management of knowledge. AKT identified six challenges that any complete approach to knowledge management must meet, namely: knowledge acquisition, knowledge modeling, knowledge reuse, knowledge retrieval and extraction, knowledge publishing and knowledge maintenance. AKT mainly deals with more lower-level issues such as knowledge representation, while our KlogMS research focuses on higher-level architectures and frameworks.

3. DESIGN OBJECTIVES

The objectives of our KlogMS research are quite straightforward: (1) make knowledge sharing easier to do and (2) make knowledge easier to find. These objectives are echoed by others as well.

For example, Evan Williams, creator of Blogger, while commenting on the potential of blogging as a KM tool also indicates that ease of use and ease of finding information is critical [CNET2003].

Large corporations are also beginning to appreciate the benefits of blogs, wikis and RSS. For example, Ross Mayfield reports [YAHOKLOG] that Disney uses blogs and wikis for a multitude of different business needs such as shift logs, RSS news distribution, discrepancy reports, ratings information, etc. [ROSS2004].

In [ITAN2002], reporting on a Gartner research: "One of the biggest challenges facing an organization today is filtering the good from the bad information. It's the classic signal/noise equation.... Gartner found, quite surprisingly, that the most useful information employees receive comes from personal networks, contact with friends and

colleagues, and emails - rather than the finely tuned information source that is supposed to be the Intranet..."

Like us, John Robb [ROBB2002] believes the solution is klogs: "The solution isn't a sophisticated KM solution, it is K-Logs. A well authored K-Log provides a filtered knowledge stream based on the Intranet. It is simple, elegant, and leverages the Intranet - the perfect way to improve the signal to noise ratio."

4. THE KlogMS ARCHITECTURE

The KlogMS architecture consists of the following key elements – the KlogMS client, KCEngine, KCRepository and the KCDirectory.

The KlogMS Client is a Web-based client that serves as the knowledge management system of a corporation. On the surface, it will look very much like other corporate intranet portals or collaboration portals for projects. The key difference is that the content or rather individual microcontent will be managed by our KCEngine (Knowledge Chunking Engine).

The KCEngine firstly acts as a microcontent management system (mCMS). Each editable content is represented as a "knowledge chunk." Unlike traditional CMS that requires separate publishing steps, our KCEngine provides wiki-like edit-on-the-fly with a wysiwyg interface. Obviously, the KCEngine will need to perform user management and control role-based access to the microcontent. This edit-on-the-fly capability means the KCEngine will also need to support versioning and content recovery in case of errors.

Traditional CMSs usually involve complicated publication and approval workflows before content will be made available on a website. For KlogMS, since the main purpose is to provide a platform where knowledge can be shared freely and instantly, the time lag in traditional CMS will hamper the flow. The gratification of seeing your content instantly displayed and shared is important in improving the dynamics of knowledge sharing. Besides, each "knowledge chunk" is editable later at any time with full audit trail if there is any issue with the content.

In KlogMS, the primitive data structure is the "knowledge chunk," which is a recursive structure. By using a common data structure, we keep the design very simple. The recursive nature of a "chunk" means we can arbitrarily and dynamically build larger structures from primitive ones. In KlogMS, everything is a "chunk." These include klog posts, users, workgroups, projects, departments, etc. The recursive property allows us to represent any arbitrary collection of related klogs as a larger "chunk," which itself may be contained in another yet larger chunk. This concept is quite powerful and allows us to write a very compact engine for KlogMS.

The growth of a KlogMS should be in an organic manner. To allow ideas and knowledge to freely flourish and easily be shared, the KlogMS platform allows a new chunk of

knowledge to be created pretty much on any page; sort of like Post-it® Notes but organized and structured. The "knowledge context," such as workgroup/project, that the chunk was created will provide valuable semantic meta-information related to that chunk. This metadata will be stored with the chunk in ways similar to "semantic blogging" [HPa, HPb]. The semantic metadata forms part of the chunking process. The organic nature of KlogMS will also allow knowledge to be self-organizing with the structure evolving naturally, similar to wiki systems [DEUR2002].

By having this recursive "knowledge chunk" design, it also allows us to depart from the traditional view of a blog or wiki. Instead of seeing only one blog or wiki on a single page, we can now display several related ones side-by-side. For example, in a project workspace, we may have the recent posts from all the project members summarized in separate portlets on the same screen. Clicking on a portlet will expand that view.

To support mobile devices, we designed each "chunk" to be emailable. Sending an email to a project "chunk" will display the email as a post in the project webpage. Alternatively, sending an email to another post will become a comment within that post. Again, the recursive structure of KlogMS allows us to very efficiently create a highly powerful environment with minimal coding.

These "knowledge chunks" are stored in a versioned repository. Conceptually, we separate the repository into two parts – the KCRepository and the KCDirectory. The KCRepository contains microcontents, which are similar to traditional blog/klog posts. This repository will be highly dynamic. The KCDirectory on the other hand is a more static repository containing predefined hierarchies (also stored as chunks) to represent the organization - the departments, projects and people. There will also be hierarchies of topics and categories that will be predefined or created on the fly. We plan to implement standard templates for these hierarchies based on open ontology such as [BIZON].

5. THE KlogMS PROTOTYPE

We implemented an initial prototype of our KlogMS platform using the Nucleus (www.nucleuscms.org) platform. Nucleus is an object-oriented blog platform written in PHP under GNU license. The main objective of the prototype is to demonstrate the core concepts behind KlogMS and to provide a test bed to perform our research experiments.

Fig.1 shows a snapshot of our current KlogMS prototype to illustrate the recursive structure of our "knowledge chunks." Although the screen, at a glance, may seem similar to other blogging systems, there are many subtle differences that highlight very powerful concepts and features provided by KlogMS.

The screen shows a klog titled "Low Cost Klog Network," which itself is linked from another blog site. The "from

here” allows one to navigate back to the original source. Entries can be condensed if needed as in this case. Users click on “Read more of this post” to get the full listing. Within this klog are two other klogs/comments embedded recursively. This illustrates a comment about a comment. (Each level has an additional “Re:” in the title.) Although common for online forums, this feature is unique for blog-based software like KlogMS and not found in other blogging systems.

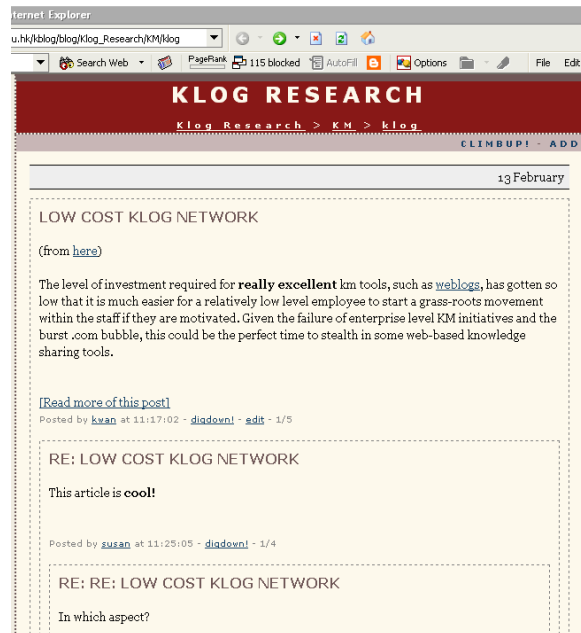


Fig. 1. Snapshot of the KlogMS prototype.

The figure also highlights a few special KlogMS commands/features – “add,” “digdown!,” “climbup!,” and internal links.

The “add” function resembles “add item” function found in most weblog systems, but the new entry becomes a knowledge chunk instead. Our “add” function adds the new piece of knowledge chunk to the current chunk. We could think of this as adding a new “post” to the current chunk. On the other hand, if the current chunk is a topic/category, we could also think of it as adding a new “sub-category” in the current category. In other situations, it might be added a new member to a project team. As KlogMS is recursive, the “add” function plays different roles depending on the context where it was performed. This is very powerful as it allows us to reuse the mechanism to perform different application level functions.

The “digdown!” command, just as the name suggests, means to dig down into a piece of knowledge chunk. When we dig down, the current chunk we are viewing will be changed to the chunk which we are digging into. This basically allows you to navigate through knowledge! Let’s take our example, when we dig down into the chunk “Re: LOW COST KLOG NETWORK”, the “path” shown on

top will become “Klog Research > KM > klog > Re: LOW COST KLOG NETWORK” instead. The fractional number x/y at the right hand side indicates there are x children for this chunk, and these are totally y children chunks under this chunk.

The “climbup!” is just the opposite of the “digdown!” command. When this action is performed, the current view will be replaced by the parent of the chunk calling this function. This function is sort of like the shell command “cd ..” except we are moving through a “knowledge space” and not a file system.

“Internal links” are similar to automatic linking provided by wikis. This mechanism is the power behind wikis. An example of internal link is the link “weblog” in the screenshot. In KlogMS, each “knowledge chunk” has a unique URI that can be referred to and linked. In addition, each “knowledge chunk” as a set of “triggers” that are used to assist with internal linking. This database of URI and triggers are made available to the KlogMS client as content is created. Whenever triggers are found, suggestions will be given to the user to form an internal link. This is sort of like the IntelliSense feature of MS Visual Studio.

Currently, we have finished the prototype and are releasing it for use internally with our Department. We plan to collect feedback from all the beta users to determine the next set of features to add as well the effectiveness of our new KM platform.

In our current implementation, the “knowledge chunking” process is supported by several means - semi-automatic “internal linking,” manual “category” selection, and semantic metadata created from the context where the knowledge chunk was created. In future implementations, we plan to explore additional knowledge chunking approaches, such as using Bayesian filters or other forms of clustering algorithms.

6. CONCLUSIONS

We believe weblog and wiki technologies have a major role to play within corporate KMSs. The current trend in klogging is growing at a tremendous rate. However, we believe that in order to fully benefit from these new technologies, we need a fundamental new design in terms of blog/wiki platform architectures. In this paper, we present our current research progress in designing such an architecture based on a simple yet powerful concept of recursive “knowledge chunks” together with the process of “knowledge chunking” to provide a platform that is conducive to organic knowledge growth as well as promoting complex adaptive “knowledge sharing” behaviors.

7. ACKNOWLEDGEMENT

The work described in this paper was substantially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China

(Project No. 9040517, CityU 1109/00E). The work described in this paper was also partially supported by a grant from the City University of Hong Kong (Project No. 7001286 and 6980002).

8. REFERENCES

- [ASHE2001] Ashenden, A. and Woods, E. "SharePoint Portal Server: Knowledge management for the masses," ComputerWeekly.com, 16 July 2001, <http://www.computerweekly.com/Article104073.htm>
- [BAUS2002] Bausch, P., Haughey, M., and Hourihan, M., *Web Blog: Publishing Online with Weblogs*, John Wiley & Sons, 2002.
- [BERK2002] Berkhin, P., "Survey of Clustering Data Mining Techniques," Accrue Software, San Jose, CA, 2002, http://www.acrue.com/products/rp_cluster_review.pdf
- [BIZON] Business Ontologies for Dynamic Web Environments, <http://www.ist-bizon.org/>
- [CAYZ2004] Cayzer, S., "Semantic Blogging and Bibliography Management," BlogTalk 2.0, Vienna, Austria, 5-6 July 2004, http://www-uk.hpl.hp.com/people/steve_cayzer/papers/scps-submitted-030228.htm
- [CNET2003] "Blog On," cnet News.com, 21 October 2003, http://news.com.com/2008-1025-5094753.html?tag=nefd_acpro
- [CROSS2004] Cross, Jay, "A Third Moment of Complexity," 23 January 2004, <http://www.internettime.com/blog/archives/001191.html>
- [CSW-RIDI] "Ridiculously Easy Group Forming," Community Server Wiki, <http://www.myelin.co.nz/wcswiki/w/RidiculouslyEasyGroupForming>
- [DEUR2002] Deursen, A. van, Visser, E., "The Reengineering Wiki," In Proceedings 6th European Conference on Software Maintenance and Reengineering (CSMR 2002), pp.217-220, IEEE Computer Society, 2002.
- [GOOG2004] "Similar Pages," Google, <http://www.google.com/help/features.html#related>
- [GRAH2003] Graham, P., "Better Bayesian Filtering," January 2003, <http://www.paulgraham.com/better.html>
- [GRAH2002] Graham, P., "A Plan for Spam," August 2002, <http://www.paulgraham.com/spam.html>
- [GREY2004] Grey, D., "Documentation and Knowledge," Knowledge-at-Work, 10 February 2004, http://denham.typepad.com/km/2004/02/documentation_k.html
- [GREY2003] Grey, D., "Corporate Memory - the Hard Way," Knowledge-at-Work, 27 September 2003, http://denham.typepad.com/km/2003/09/corporate_memor.html
- [HPa] "Semantic Blogging for Bibliographies," Steve Cayzer's homepage at HP, http://www-uk.hpl.hp.com/people/steve_cayzer/semblog.htm
- [HPb] "Semantic Blogging for Bibliography Management," project homepage at HP, <http://www.hpl.hp.com/semweb/biblio#Semantic%20blogging>
- [ITAN2002] "Data dyspepsia blights the workforce," IT Analysis, 15 May 2002, <http://www.theregister.co.uk/content/23/25283.html>
- [ITTO2002] "Which of the following components do you consider to be essential for a Knowledge Management system to be effective?," ITtoolbox, <http://knowledgemanagement.ittoolbox.com/poll/results.asp?Task=Results&PollID=158>
- [KMWO2003] "KMWorld's 100 Companies that Matter in Knowledge Management 2003," <http://www.kmworld.com/100.cfm>
- [LIEB2002] Liebowitz, J., "Knowledge Management," Systems Engineering Seminar, Goddard Space Flight Center, 5 February 2002, http://seacd.gsfc.nasa.gov/SE_Seminar/
- [MACI2002] Macintyre, A., "Understand Radio Categories," Radio, Userland, 2002, <http://radio.weblogs.com/0107846/stories/2002/08/18/understandRadioCategories.html>
- [MILE2001] Milewski, B., "Why Do You Need a Version Control System?," 25 May 2001, informIT, http://www.informit.com/isapi/product_id-%7B8580AF88-0492-4686-A8F4-912B7683AF0C%7D/content/index.asp
- [MOUL2002] Moulthrop, S., Bernstein, M., and Carton, S., "Self-Assembling Hypertexts, Weblogs, and Wikis," In the Proceedings of the thirteenth ACM conference on Hypertext and hypermedia, College Park, Maryland, 2002, pp. 149.
- [NEWM2002] Newman, J.S., "KM Implementation," In the Thirteenth Knowledge Management Roundtable, 16 May 2002, Washington, D.C., <http://www.icasit.org/km/kmrt/may02/>
- [PING2002] "Pingback 1.0 Specification," <http://www.hixie.ch/specs/pingback/pingback>
- [PLUM2004] Plumtree, http://www.plumtree.com/default_flash.asp
- [PLUM2003] Plumtree, New Nucleus Research Shows Plumtree Leads IBM and SAP in Portal ROI, http://www.plumtree.com/news_events/pressreleases/2003/press081203c.htm
- [RITT2003] Rittenbruch, M., Mansfield, T., and Cole, L., "Making sense of "Syndicated Collaboration"," 2003, <http://www.dstc.edu.au/Research/Projects/Infoeco/publications/group03.pdf>
- [ROBB2003] Robb, J., "ROI Calculations: K-Logs vs. Traditional Intranet Portals," John Robb's Weblog, July 2003, <http://jrobb.mindplex.org/stories/2003/07/29/roiCalculationsKLogsVsTraditionalIntranetPortals.html>
- [ROBB2002] Robb, J., John Robb's Weblog, 15 May 2003, <http://jrobb.mindplex.org/2002/05/15.html>
- [ROSS2004] "Disney Enterprise Weblogs and Wikis," Ross Mayfield's Weblog, http://ross.typepad.com/blog/2004/02/disney_enterpri.html
- [SEB-BLOG] Seb's Open Research, <http://radio.weblogs.com/0110772/>
- [SEB2004] "reBlogging: push-button republishing," Seb's Open Research, 10 February 2004, <http://radio.weblogs.com/0110772/2004/02/10.html#a1455>
- [SHARE2004] Microsoft Windows Server 2003, Windows SharePoint Services, <http://www.microsoft.com/windowsserver2003/technologies/sharepoint/default.mspx>
- [TROT2004] Trott, M. and Trott, B., "A Beginner's Guide to TrackBack," MovableType, <http://www.movabletype.org/trackback/beginners/>
- [TURN2002] Turnbull, G., "What is a k-log?," WriteTheWeb, 5 March 2002, <http://writetheweb.com/Members/gilest/old/123>
- [WALK2002] Walker, J., "Links and power: the political economy of linking on the Web," In the Proceedings of the thirteenth ACM conference on Hypertext and hypermedia, College Park, Maryland, 2002, pp. 72-73.
- [WIKIKM] "Knowledge Management," Wikipedia, http://en.wikipedia.org/wiki/Knowledge_management
- [WILS2002] Wilson, T.D., "The nonsense of 'knowledge management'," In Information Research, Vol. 8 No. 1, October 2002, <http://informationr.net/ir/8-1/paper144.html>
- [WOHL2001] Wohl, A.D., "Life On the Internet: Could Blogging Assist KM?" <http://www.wohl.com/wa0156.htm>
- [YAHOKLOG] Yahoo!Groups, K-Logs (Knowledge Management Weblogs), <http://groups.yahoo.com/group/klogs/>