City University of Hong Kong
Department of Computer Science

BSCCS/BSCS Final Year Project Report 2004-2005

04CS018

The KlogMS

Knowledge Log Management System

(Volume 1 of 1)

Student Name : Lai Ho Kwan
Student No. : 50333816
Programme Code : BSCCS

For Official Use Only

Supervisor : Dr. Chun, H W Andy
1st Reader : Liu, W Y
2nd Reader : Mong, Florence

Abstract

Many Knowledge Management System (KMS) have been developed to handle knowledge. Companies may use them to retain and manipulate knowledge from the employees. However, there are many problems that found on many KMS, which will prevent a KMS to be successful. To overcome the problem, we propose to merge up the concept of weblog and KMS. Then the major feature of the resulting system could be a new kind of KMS which is probably more usable and attractive to the users, while preserving the required functionality. Due to the size of the whole research, this project only covers the underlying framework of the system. The framework includes the functionality of a multi-users weblog system, with the ability to extend in later period of time. The framework provides well-written programming library to serve the major goal of the KlogMS. The whole project has undergone the processes of research, development and evaluation. The research process serves as two purposes. First, it will be a selection process of the current technologies that is useful in developing the KlogMS system. Another purpose is to know the current trend of similar systems. Due to the reason that the system is new to the industry, we have put much effort on the study of the existing blogging systems, services and related technologies. The implementation of the system has already made complete yet not perfect when this report was done. The system serves as a public weblogging web service on a music website which called Talkmusic.org. The programming was written in an object-oriented approach. This enables future development to be easily taken. To evaluate the system, we have discussed the system with the other similar systems, so a comparison can be made. This helps clearing up the future direction of research and development in the same field. To help the evaluation process to go even user-oriented, there is a forum put up on Talkmusic.org which let users to give feedback to the system. During the project, a conference paper is written and was accepted by the CCCT conference.

Acknowledgement

I would like to give a big Thank You to my supervisor Dr. Andy Chun. He has given me a lot of support during the project. He encouraged me to explore different aspect of Computer Science. I have to say that the flexibility that he gave me permitted me to enjoy my work, because I can do what I am really interested in.

I would also like to thanks Martin Ng who have shared a lot of Computer Science knowledge with me, during the placement period and the project period. He has the passion on Computer Science and could turn everything enjoyable for everyone.

Thanks Google, Firefox, PHP, Apache, Linux, and all the heroes in Computer Science technologies. Without them, my Computer Science life would not be so enjoyable.

Thanks all of my friends and my family, they are always with me and give power to me.

Lai Ho Kwan (Dan)
April 10, 2005

Table of Contents

** References are all in-text*

1. Introduction

Many KMS have been developed to handle knowledge. Companies may use them to retain and manipulate knowledge from the employees. However, there are many problems that found on many KMS, which will prevent a KMS to be successful.

1.1 The Existing Problems

Some of the problem will make the users not willing to enter knowledge into a KMS. The problem may come from an overly-sophisticated user interface or data input procedure. Another possible reason is due to psychological habits of human that they will more likely to pour knowledge into something that they can depends and have a passion on. Some cases maybe due to the system performance. Users will not be comfortable to work with a system that is very low in performance. This problem persists normally because not many server administrators (or their domain) have the budget to upgrade the hardware.

Some people have thought of using weblog as a simple KMS. A weblog used in this way is described as a K-Blog (Knowledge Weblog). By using this approach, the user interface problem could most probably be solved, because most of the weblogging system is designed to be user friendly. However, almost all of the existing weblog are not designed to handle knowledge. For the others, they could be overly-sophisticated again or just went in a different direction instead of becoming a KMS. One of the major features to organize knowledge is the category system. A typical weblog system aligns information only in chronological order. Some better system will provide simple category function. But the category system found in these systems is nearly always designed to work with a single user only. As a result, in a typical weblog system, when a lot of information is published by user(s), users will then hard to find the desired information at later days. When a typical weblog system is used to handle knowledge in a large company, another problem raised. Because there are too many individual blogs, it is very clumsy to manage these weblogs which is written by a lot of users, individually. Even by using feeding methods like the RSS, the result would be a bigger, but still a typical weblog which does not mean any improvement in terms of information structure.

1.2 Research Objective

In our research, we are going to try to improve the current KMS systems. First, for the user usage experience, we have to tackle the major problems listed in 1.1. So we propose to add the weblog idea to KMS. At least the user interface problem in a conventional KMS should be eased. Secondly, we will try to evaluate the user experience and performance of using a category system. Although, theoretically, the system will be benefit from the advantage of category, we still have to make sure if it works in actual usage by the users. Thirdly, in the research, we will try to discover features that are found to be useful by the users, and at the same time, we could also know which feature is not really needed in this kind of system. Finally, another major aspect of the research is the optimization of the system. This is another critical point because we always want to do more jobs with the same equipment. When we are doing all of these, we want to keep all of the technology used to be free of charge, which means we could develop the product without using extra money.

1.3 Importance of the research

If a better kind of tool is being used for knowledge management, more knowledge could be retained because the knowledge management tool will be used in a higher frequency. Another reason for the above benefit is that human tends to forget something even in a very short time, so more knowledge could be retained if they write it down quick. Furthermore, the same quality of knowledge will be even greater if they are stored in a well-organized structure.

1.4 Finding in related researches

Our understanding of human cognition in decision making and knowledge-intensive work is marginal[1]. There is not an accepted economic "theory of knowledge" that is applicable to business. We do not have a general understanding of how to undertake comprehensive and systematic KM within an organization and may need a new theory of the firm to manage knowledge effectively – and to link it with enterprise strategy, tactics, and daily operations.

[1] Wiig, K., M., "Successful Knowledge Management: Does It Exist?" - http://www.krii.com/downloads/does_successf_km.pdf

Knowledge Management offers the possibility of allowing organizations to tap into not just the documents they've created, but the expertise of their employees, past and present[2]. Most KM initiatives fail because of the lack of content[3]. It is important to bring people to verbalize their thoughts and make them available to others. It is always a problem about how can the knowledge management tool make that as close as possible to a conversation, which is far better to connect people than long structured articles. A diary benefits its writer because it helps him clarify his thoughts. It also benefits its reader because it can be a starting point for a very meaningful conversation, one that the blog writer will jump into. A weblog is really an incubator of shared ideas.

The term Weblog or Enterprise Weblog does not readily indicate to a reader what is it all about[4]. People, in general, are starting to figure out that a Weblog is a personal journal on the web. Once the technology becomes more popular, people will see the Enterprise Weblog as means for relevant business communication. Weblogging is interesting because it is a fairly non-intrusive way of allowing workers to share the process by which they do research, do analysis, and select information. Weblogs have a very important role to play in the innovation process, and thus should be considered very seriously in corporate environments[5].

A popular weblogging service Blogger[6] is a kind of typical weblogging system. However, it does not support categories, which is very important in a KMS. The web is huge. One way to help people find useful pages is a directory service[7]. The same could be applied to a KMS. The directory service is just like the category function. Such as Yahoo![8], or The Open Directory Project[9]. Typically directories are manually created, and the judgments of where a page goes are done by a human. For example, Yahoo! puts "General Motors"

[2] Wohl, A. D., "Life On The Internet: Could Blogging Assist KM?", Amy Wohl's Opinions - http://www.wohl.com/wa0156.htm
[3] Michael, A., "Supporting enterprise knowledge management with weblogs: A weblog services roadmap", A presentation in Washington: Computers in Libraries 2004 conference - http://urlgreyhot.com/personal/cil2004
[4] Traction Software, "Post Modern Knowledge Management and Social Enterprise Blogging "- http://www.masternewmedia.org/2003/02/07/post_modern_knowledge_management_and_social_enterprise_blogging.htm
[5] Mopsos, "Ecology of weblogs for business" - http://blog.mopsos.com/archives/000081.html
[6] Blogger Weblogging Service – http://www.blogger.com
[7] Glover, E., J., Tsioutsiouliklis, K., Lawrence, S., Pennock, D., M., Flake, G., W., "Using Web Structure for Classifying and Describing Web Pages", In International World Wide Web Conference - http://citeseer.ist.psu.edu/cache/papers/cs2/155/http:zSzzSzwww.neci.nec.comzSz~lawrencezSzpaperszSzclass-www2002zSzclass-www2002.pdf/glover02using.pdf
[8] Yahoo! Search Engine - http://www.yahoo.com

into several categories: “Auto Makers”, “Parts”, “Automotive”, “B2B – Auto Parts”, and “Automotive Dealers”. Yahoo! puts itself “Yahoo!” in several categories including “Web Directories”.

To put it all in all, the resulting KMS should be “Decentralizing for innovation, centralizing for reuse”[10]. Value is added to existing information through a process of aggregation, simplification and dissemination, which is exactly the concept underlying a web of blogs. In essence, the more decentralized, the richer information is; the more centralized, the more structured it is.

[9] Open Directory – http://dmoz.org
[10] Michael, A., “Supporting enterprise knowledge management with weblogs: A weblog services roadmap”, A presentation in Washington: Computers in Libraries 2004 conference - http://urlgreyhot.com/personal/cil2004

2. Background Studies

The background study is done to have a thorough understanding of the technologies that are useful in implementing the system. Moreover, the study also looks into some existing services and systems in order to grip an understanding about the related development nowadays. I want to remind one of the research objectives again – we want to use technologies and tools that are free of charge.

2.1 Supporting Technologies

This section will cover the introduction of some useful technologies and tools, which would be helpful in developing the system. This study is essential because I would like to choose the best option among the large variety that we can choose from.

2.1.1 Web Programming Combos

In developing web application, we want to study the whole combination of the tools that required running the web application, instead of concerning only the web programming language. The combination would include the web programming language, the web server, the database server, and the Operating System which works well with the first three things. Database Server becomes a point of concern because most of the web application could not work well without an individual system to manipulate the data.

For web-programming languages, one way to classify them is to divide them into two sub-sets: Client-side programming languages and Server-side programming languages. Each programming language has its own advantage and disadvantage, so the selection criteria depends on whether it is appropriate to use the language in the specific domain or environment.

2.1.1.1 LAMP

LAMP[11] is an acronym for a set of free software which is commonly used together to run web applications. The L, A, M and P in the word LAMP stands for:

- Linux, the Operating system
- Apache, the Web server
- MySQL, the DBMS (Database Management System), or Database Server

[11] LAMP on Wikipedia - http://en.wikipedia.org/wiki/LAMP

- PHP, Perl or Python, which are the web programming languages. While in this project, PHP will be used.

PHP stands for Hypertext Preprocessor. PHP is a type of HTML- embedded scripting language, most of its syntax is borrowed from C, Java and Perl. The portion of Perl in PHP has mixed with some unique PHP-specific features. The major goal of PHP is to let programmers to finish their work in a short time[12].

Beside the development speed, another advantage of PHP is that it has bundled MySQL client API Library, so PHP can be used with MySQL Database server natively. The word "natively" means they can work very well together, instead of just the performance issue[13]. Some people even refer PHP and MySQL as "Dynamic Duo" in dynamic website creation.

In order to run PHP application, we only need to download the Apache web server (which is free) and add PHP support to it. When a user requests a PHP script, the PHP module in the Apache web server interprets it and generates the result dynamically.

2.1.1.2 Java

When Java is used in developing web applications, we refer to the J2EE technology, which includes: JAX-RPC, JSP (Java Server Pages), Java Servlets, EJB (Enterprise JavaBeans components) and other useful tools.

In order to run Java web applications, we will need a Java container or an Application server. Some Java containers are free of charge[14], such as the Apache Tomcat.

Some of these Java container and application servers can be runs on free Operating System, such as the Linux. Java can be used with free Database servers, such as the MySQL DBMS (Database Management System), via external API library. So we can say web applications can be developed free of charge on the Java combo too.

[12] PHP – General Information - http://hk2.php.net/manual/en/faq.general.php
[13] Introduction to PHP - http://www.zend.com/zend/art/intro.php
[14] Top 5 free Java Servlet Engine - http://java.about.com/cs/j2ee/tp/devservletengin.htm

2.1.1.3 Other Combos

There are other kinds of web application combo available too, for example, the Microsoft ASP platform. However, this kind of platform either is not good enough, or they are not free. For example, in order to run ASP, at least we will need Microsoft Windows, which is not a free operating system. So the other combos are not in consideration list and are out of the focus of the background study.

2.1.2 Document Format

In order to store and manipulate information, we have to store the information in a computer file of a specific format. Sometimes file format also affects the efficiency of data exchange. This study is to find out which file format is right for which job.

2.1.2.1 XML

XML[15] is a very simple and flexible text format, which is derived from SGML. It was originally designed to meet the challenge of large scale electronic publishing. Nowadays, XML plays an important role in the data exchange of a wide variety of web applications.

XML Documents allows information to be stored in a structured format[16]. This kind of structured information contains both the content (words, pictures, etc.) and what role does the content plays (is it the title? or the content in a piece of weblog post?). In order to archive this kind of structure, we need to use markup language as a mechanism, while the XML defines a standard way to add markup to the documents.

2.1.2.2 HTML

HTML is a "lingua franca" for publishing hypertext on the Internet. It is non-proprietary and based on SGML. It can be created and processed by a wide range of tools, can be as simple as a text editor[17]. HTML can be also generated by web application, such as PHP and Java server pages. In this case, the generation process is taken on the web server, application server or application container.

As most of the web browsers are basically a HTML viewer, so if the web application can generate HTML document, the compatibility of the web application for the web

[15] Extensible Markup Language (XML) - http://www.w3.org/XML/
[16] XML.com – What is XML? - http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN78
[17] Hypertext Markup Language (HTML) Homepage - http://www.w3.org/MarkUp/

browsers can be ensured. However, different browsers may display the same HTML document in different ways.

2.1.2.3 RSS

RSS is a format for syndicating news and content of news-like sites, for example, Slashdot, Wired, and personal weblogs. In fact the usage of RSS is not limited to be only on news, it is applicable on everything that could be broken down into pieces of news-like information, like the change-log of WIKI sites, change-log of CVS repository, and even the revision history of some published content. Once the RSS is generated and put up on the Internet, RSS program can check the feed and react to the changes in an appropriate way, say, to notify the user about the change via a desktop notification pop-up box[18].

A useful tool to accompany with RSS is something called the news-aggregators. This kind of software can group up all the RSS feeds that is of interest, to let users to view them in a convenience way.

2.1.2.4 Atom

The use of Atom is similar to RSS. It is also used to let people keep track of a lot of site in an efficient way. It also let people to notify another people about the information that what have published on the Internet[19]. Atom feeds are just like RSS feeds that they can be read by feed readers so we could keep track of a number of sites in a single user interface.

2.1.3 Templating

Templating is a way to facilitate a manageable way to separate the application layer and the presentation layer[20]. This separation could be well explained, if we say, for example, when the application programmer and the template designer are two different people. They do their own job, so that the application keep passing information into the templates and the resulting pages are then returns to the web site visitors. If one day the designer changes the design of the template, or the application programmer changes the

[18] What is RSS? - http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html
[19] What is Atom? - http://www.atomenabled.org/
[20] What's a templating engine and why should I use one? - http://smarty.incutio.com/?page=SmartyFrequentlyAskedQuestions#basics-1

way to retrieve and manipulate the information, the changes they made should not affect each other if the presentation layer and the application layer are separated. So they could do their own job without affecting another part.

2.1.3.1 Smarty and Velocity

Smarty is a popular templating engine for PHP applications, while Velocity is a popular templating engine for Java web applications. They perform the templating action on the web server so that the process is transparent to web browsers. The browser will only get the resulting web pages, where these resulting pages have already undergone the templating process in the web server, i.e. they are the result of the combination of the application generated data and the template files. As the resulting web pages could be in an arbitrary format, including the "lingua franca" HTML, so the result could have a high compatibility with most of the web browsers.

Smarty is very popular because it is very to easy and it shows outstanding performance in the templating process, especially when it is used with the ionCube PHP Accelerator[21].

2.1.3.2 XSL

XSL consists of three parts, XSL Transformation (XSLT), the XML Path Language (XPath), and the XSL Formatting Objects (XSL-FO)[22]. An XSLT style sheet specifies how a XML document is transformed into an XML document which uses a formatting vocabulary, such as (X) HTML or XSL-FO documents.

When used in the purpose of templating, for most of the time the transformation will be from XML to (X) HTML. The application layer will first generate the information that is going to be templated, in the format of XML. Then the XML will specify the URI of the XSL, which is going to be used to transform this piece of XML document. Then the transformation will take place either in client-side or in server-side.

2.1.3.3 CSS

CSS stands for Cascading Style Sheets[23], which is a feature that is added to HTML in order to let web developers to have more control over how the pages are displayed. By using CSS, web developers could defines how different elements are displayed, such as

[21] ionCube PHP Accelerator – Performance Test - http://www.php-accelerator.co.uk/performance.php
[22] The Extendible Style sheet Language - http://www.w3.org/Style/XSL/

the <h1> section or the section in a HTML document, or even some self-defined elements such as “news_content” or “news_title”. Once the CSS has been defined, they can be reused over and over again, which then provides a way to unify the presentation of the whole site.

CSS is useful in the last step of templating, which is to tweak the presentation layer so that it meets the artistic standard and individual requirement of different people.

2.1.4 Programming Style

In this project, I need to choose between procedural (including functional) and object-oriented programming. There is always discussion about this issue because no one is better than the other. For procedural programming, the major attraction is the speed of execution, while for OOP (Object-oriented programming), the major attraction is about the readability, extendibility and maintainability of the code.

2.1.4.1 Procedural

Procedural programming is fine for small projects[24]. It is the most natural way of telling the computer what to do. The computer’s own language, the machine code, is procedural, so the translation of procedural high-level language to machine code is a straight-forward and efficient process. Most procedural programming languages have built-in way to split big lists of instruction into smaller lists, which is the use of functions. In this way, not only the program will be easier to read, but also can the segments of codes be reused again. Most of the procedural programmers want to make things runs faster, and then the readability of the code usually becomes not so important when compared to the performance[25].

Common languages that encourage procedural programming: C, Perl, Assembler

2.1.4.2 Object-Oriented Programming

OOP (Object-oriented programming) requires more planning and design when the system is initializing[26]. Sometimes we may even have to over-write programs. But once

[23] Web definition and Glossary at Rusty Brick - http://www.rustybrick.com/definitions.php
[24] Functional vs. Procedural programming - http://www.colorado.edu/its/scico/Info/details/funcproc.html
[25] The best tool for the job – OO versus procedural programming in PHP - http://www.zend.com/zend/art/oo-proc.php
[26] OOP University – Part nineteen - http://www.applelinks.com/rbu/095/

the program is done, we are almost finished. Modifying and extending functionality of object is easy. The most important thing is that the advantage of OOP comes when it is time to rebuild, rewrite, enhance, or debug the system. Most of the object-oriented programmer wants to put things to work everywhere, and do not always care about the performance issues behind the code. One of the practices found on this kind of program is about the abstraction of concept behind the code.

Common languages that encourage OOP: Java, Smalltalk, C++

2.1.4.3 3-tiers system architecture

The 3-tiers design pattern separates the system into 3 layers in the web application development. They are the presentation layer, the application layer and the database layer[27].

The presentation layer provides a way to let users interact with the application. This layer can be ranged from HTML, DHTML to complex DCOM and Java Applets.

The application layer provides the logic and procedures that is required to perform complex action through the interface, and it maybe written in PHP, Java, or other web programming languages.

The database layer retrieves and updates information at a high level. This layer maybe a database, file system, or writable media. In web application, the most practical is to use a database server.

By separating the layers, it enables parallel development of different tiers of the web application[28]. Another benefit is that it provides encapsulation for different layers, which can result in a more robust web application. Thirdly, it allows easier maintenance and support, say, when a part of the business layer is needed to change, we only need to change business tier on one server. On the monolithic application, a change in the business layer may also mean the change on the website template and the database accessing method.

[27] Web application development – A guide to success - http://www.sitepoint.com/article/development-guide-success

2.1.5 Blogging Specific Technology and terms

There are a number of common features that enable blogging in an easier and comfortable way, and some of them helped to exchange information between weblogs.

2.1.5.1 Trackback

The Trackback technology is originally developed by Movable Type[29], which is a very popular weblogging engine that is written in Perl.

Trackback is a framework for peer-to-peer communication and notification between websites[30]. The mechanism is based on Trackback ping, which is a request saying that "resource A is related to resource B", where a Trackback resource is represented by a Trackback Ping URL, which is just a standard URI.

By using Trackback, websites could notify others about the resource that they are related. For example, when Weblogger A has written something that is related to Weblogger B, A could send a Trackback ping to B, which will result in two consequences. First, B will list all the sites that have referenced a particular post on B's site, this could allow visitors to read all the information on the Internet that is related to B's site, including A's site's content. Secondly, the ping also provides a firm and explicit link between the entries on two sites.

2.1.5.2 Published Content versus Dynamic Generation

Some public blogging services publish the written weblog into static pages. This is in the contrary of the way of dynamic generation, which is more common to be found on web application.

Published content only demand the web application to be used only when the blog content has changed, say, in the process of creating, editing and commenting. When the user made these actions and chose to publish the content, the pages will be generated

[28] Visual Basic 6 Application Development 2 – Distributed Applications - http://tutorials.findtutorials.com/read/id/195/headline/Visual+Basic+6+Application+Development+Part+2+-+Distributed+Applications

[29] Movable Type - http://www.movabletype.org

[30] Trackback technical Specification - http://www.sixapart.com/pronet/docs/trackback_spec

dynamically as static pages once, and reside on the web server. When the visitors browse the weblog, they browse the static content that is already generated.

Dynamic generation of weblog is more common to be found. In this way, the web application is involved for any request that the visitors or writer have made. So when the visitors browse the blog, the content is generated on the fly for every time.

By using the published content approach, the web server which holds the static page could have a lower resource usage. However, the publishing process takes up a lot of system resources, especially if the weblog is large. Moreover, static pages can only presented in a limited way, so if we want to change the view of the page without publishing again, we have to generate a lot more files.

By using the dynamic generation approach, every visitor to the blog will request the web application to process the data. Usually these requests are small because most visitors will only browse the blog. However, when there are a lot of visitors to the blog, the server resource usage will be a problem. This is due to the sum of resource consumption is large, when there are a lot of small requests. The benefit of this approach is that it bypassed the publishing procedure, which could be a problem when the blog is very large. Moreover, because the page is generated every time when the request is sent to the server, the page can be formatted in any way and contains any amount of content, say, we may request the page to contains only the content that are in the category “Computer Science”.

2.2 Systems for reference

There are already many existing blogging service providers and blogging system packages available nowadays. In order to understand the current trend of blogging, a study is done on these systems.

2.2.1 Public blogging services

There are many blogging service providers on the internet. Most of them are provided free of charge, while upgrading to a paid service plan could provide user with more features, for example, image hosting service. Once a user has registered to the service, they can start to create their own blog. With this kind of services, user's weblog is kept in the database which resides on the blogging service provider's server, so the users have to visit the service provider's website when they want to manipulate their blogs. The major advantage is that these kinds of services are very simple to use.

When this report is released, Xanga and Blogger.com are two very popular blogging services[31], and they represent two different style of blogging.

2.2.1.1 Blogger.com

Blogger and the Blog*Spot hosting service has been purchased by Google[32]. So the Blogger Pro premium service has finally becomes a free and ad-supported services.

Posting with Blogger is very simple. Users just need to enter plain text or HTML code in the editor, then save them. Then users could browse and edit the old post any time they want with the browsing interface. The posting action has been made even easier when Google has added the BlogThis! Function to the Google Bar. They also provide a function that let users to post by e-mail. So users only need to e-mail the post to an e-mail address with an e-mail client, and then the post will appear in the user's blog. Blogger also provides a paid service which called the Audio Blogger, upon subscription, a user can phone in audio-only posts.

Blogger does not put a lot of effort in building up a community between individual bloggers, so the interaction between users is not common and neither easy. In Blogger, blogs do not have any kind of access limit, which means the blogs are always open to the

[31] Blogger vs. Xanga - http://www.calicocat.com/2004/06/blogger-vs-xanga.html

public. They also do not support image hosting at the time being. However, due to the reason that it is free, there are already 1.5 million registered users using the blogging service from Blogger.

2.2.1.2 Xanga.com

Xanga started in 1999 as a site for sharing book and music reviews[33]. The site was driven by user preference, then not long after that it has became a full-scale blogging service provider. About their blogging service, it provides useful function for easy editing of blog posts. Xanga allows users to post by e-mail, or via the WYSIWYG editor that allows user to add smiley, symbols and links easily. They have also automated the process of image uploading and formatting.

Xanga has provided some useful function in building up member communities. For example, they provide BlogRing function, so users of similar interest can join the same ring, or even subscribe to one another's blog. Moreover, users can comment on the other users' blog, or give "eProps" to the good posts.

2.2.2 Web based Blogging systems

Some blogging system can be downloaded and installed in an arbitrary machine. In this way, the blog post database can be kept in a private place. Moreover, the blogging system can be more customized to the user's liking. However, the setup process is harder than setting up an account on public blogging service, and it is not an easy job if the user do not have experience in system administration.

2.2.2.1 Movable Type (Perl)

Movable Type is a very popular Perl software that designed for people who want to host the blog in their own server. It was written by Six Apart[34] and is the hosted version of TypePad. It is very easy to install provided that the user has a little system administration expertise[35].

All of the installation processes are step-by-step, though some steps are still tricky and it is better to have some experience on Linux machines. Once the script is installed, the

[32] Blogger, Blog*Spot review by PC magazine - http://www.pcmag.com/article2/0,1759,1400332,00.asp
[33] Xanga review by PC Magazine - http://www.pcmag.com/article2/0,1759,1400391,00.asp
[34] Six Apart - http://www.sixapart.com/about/history
[35] Do it yourself – Movable Type 2.64 – Reviewed by PC Magazine - http://www.pcmag.com/article2/0,1759,1400400,00.asp

script will generate the necessary database files, then the system will be up and running. Users will then get full control over a nicely-packaged blogging system, with support for XML-based template to speed up new content. To compare with its hosted version TypePad, Movable Type only has removed a small portion of admin options.

In a Movable Type system, administrator can set up multiple weblogs. But it is not a straight forward process because the administrator has to access the file system on the server, which is about creating some new directories and changing some permission. Except for this, everything can be done with the web interface. Some of its functions are widely appreciated by users, such as the control over the notification and reader comments, altogether with the ability to block spammers' and troublemakers' IP addresses.

3 Implementation

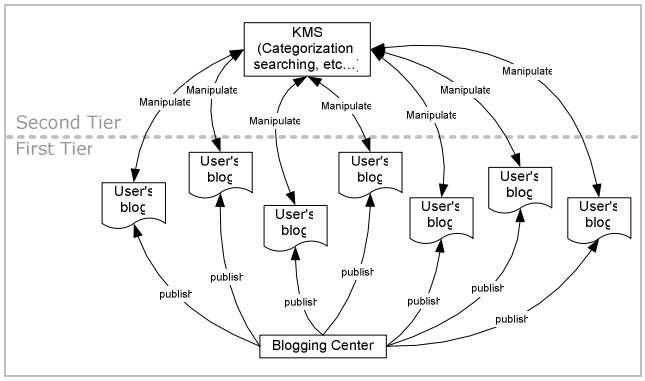
The first implementation of KlogMS is the Talkmusic.org Blogging Center, which is a publicly accessible blogging service. The user database of Talkmusic.org is already running and there are already more than 600 users who are registered in the website. This could help getting more feedback from users, so that the development of the system could be more user-oriented.

3.1 Application Domain (Talkmusic.org)

Talkmusic.org was started at in the summer of 2004 (the start of the Final Year Project). The objective of this website is to build up a music community so that music lovers could gather up together. Before the blogging system has been set up, Talkmusic.org was running forum service. The forum service was set up with PhpBB[36], which is a free open source forum script that runs on PHP platform. Talkmusic.org let bands and music entity to open forum free of charge. Up to the April of 2005, there are more than 600 users and about 9000 posts were posted by the users. With this kind of high activity, the number of user base should be useful when doing user-feedback activities.

3.2 High-level Requirement (Features)

The high-level specification will be divided into two tiers. The first tier is of a lower level, which defines the Blogging Center as a personal blogging tool. The second tier is of a higher level, which defined the system as a categorized KMS (Knowledge Management System). The specification list will then be fall into either one of the tier. However, in this project, we will only define the system framework what fall into the first tier, while the second tier will be out of the scope of the project.



System Layout

[36] PhpBB – http://www.phpbb.com

3.2.1 Personal Blogging tool

The first tier is to make the Blogging Center functions as a normal blogging service provider. The major objective of this tier is to make the Blogging Center easy to use, so users can make their own blog easily and comfortably.

3.2.2 Every user has a blog

Once the users have logged into their own account, they could start writing their own blog. Each user has one blog.

3.2.3 Template system

The template for every user's blog can be customized. In order to accommodate different kinds of users, two different template setting modes will be provided.

Standard: The user can change the style for different part of the Blog. For example, the user can change the Blog title to be red in color, while the post will be set to 12px in size. The basic layout will be following a standard provided by Talkmusic.org

Advanced: Users can create and edit the XSL file for their blog. So they can make the blog look whatever they like.

3.2.4 Post management

Users could add/edit/delete their own post, which lives in their own blog. They can also browse their posts and make changes when they want.

3.2.5 Commenting/replying system

Blog visitors could leave their own comment or feedback to the posts. The blog owner can control the visibility of these comments. The blog owner could also manage the comments/replies left by the visitors, which is similar to the post management.

3.2.6 Publish mechanism

Once the users are satisfied with their posts, they could publish their personal blog. The blog will then be published to the Talkmusic server.

3.2.7 Talkmusic.org API (for developers)

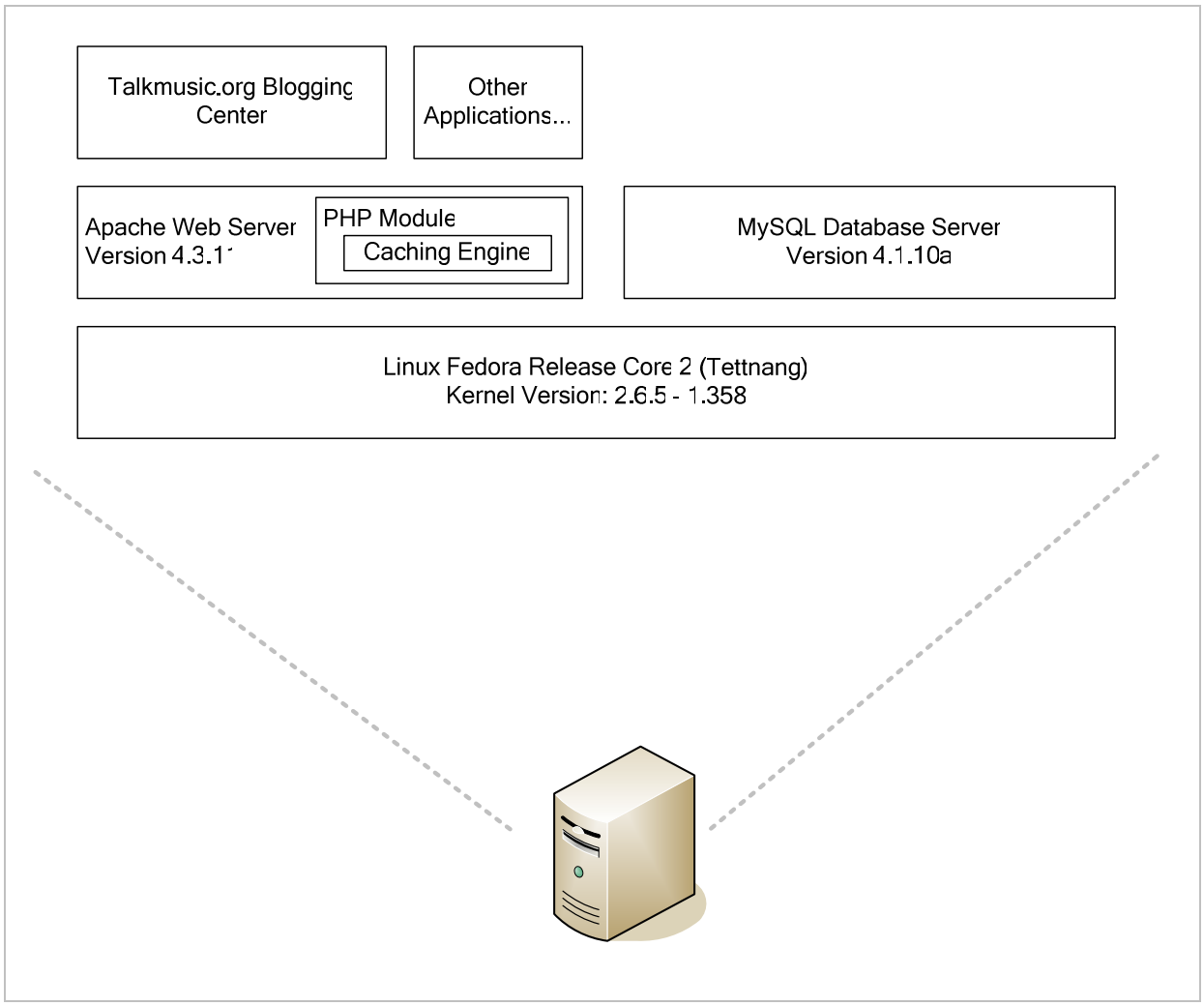
When the development of the system is finished, Talkmusic.org could provide a set of API for future development. This set of API should be easy to use and should be mission critical as the future developers could use them in some extreme ways. The API will also be used when the system needs to be extended.

3.3 System Development Process

In this section, I will explain issues related to the software development process. To let readers obtain a clearer understanding of the software, and the practice that I used during the software development period.

3.3.1 System Environment

The system is developed remotely on the host “rose” in the Talkmusic.org network. The Blogging Center co-exists with other existing web applications, which runs on the Apache web server with PHP module enabled. The apache web server is in turns run on the operating system Linux Fedora Core 3. The environment could be demonstrated in the following diagram. Notice that the version numbers and configuration shown are only correct for the moment when this document is written, because all the software are subject to upgrade or patching in the future.



The next few sections will be about some information about the specific parts of the system.

3.3.1.1 PHP 4.3.11 versus PHP5.0.3

At the time when the system was developing, I have to choose between PHP 4.3.11 and PHP 5.0.3. Initially PHP 5.0.3 wins and I have once compiled it and installed it onto the server, because I want to benefit from the new OOP environment of PHP5. The PHP5 OOP environment is of better performance and has more features because it is completely rewritten[37].

PHP5 compatible with most of the PHP4 features, except some very few incompatibilities[38]. Those incompatibilities are not very crucial as they does not affect the other existing systems in an obvious way. The existing system includes PhpBB[39]

[37] Classes and Objects (PHP5) - http://hk.php.net/manual/en/language.oop5.php
[38] PHP5 – Backward incompatibility changes - http://hk.php.net/manual/en/migration5.incompatible.php
[39] PhpBB – http://www.phpbb.com

2.0.13, Wakka[40] 0.1.2, phpMyAdmin[41] 2.5.7-pl1, Gallery[42] 1.4.4-RC3, and some other systems are not seriously affected by the change from PHP4 to PHP5. The most annoying problem was about the problem in the Administration Panel of the PhpBB when I have installed PHP5.0.3. Some other PhpBB users also found this problem[43].

The major reason for me to stay in PHP 4.3.11 is because PHP 5.0.3 does not have any caching systems developed to help speeding up the PHP scripting engine. Caching system becomes crucial when the concurrent access rate is high, because it applies caching mechanism on the page generation process, so it could relief the system resource usage. For PHP 4.3.11, I use ionCube PHP Accelerator[44] as the caching engine, so the request/reply time is significantly decreased. For more information about the influence of a caching engine, please read the next section.

3.3.1.2 Caching Engine

During the development process, I have discovered the power of a caching system, so I have took a deeper look into it. For PHP, there are a number of caching systems available. To list the more popular engine, that will include the APC, the afterburner*cache, the Zend Accelerator (in the Zend Optimizer Suite[45]), and the PHP Accelerator (by ionCube). In this list, the most effective one should be the Zend Accelerator, which was concluded by testing result[46]. However, the Zend Optimizer is closed-source and is a proprietary technology which owned by the Zend Technology Ltd[47]. So to support the open source world, I have finally chosen the ionCube PHP Accelerator, which is not very bad when compared to the Zend Accelerator.

This is a more detailed benchmark for the caching systems mentioned above[48].

[40] The website of Wakka has been down for unknown reason.
[41] phpMyAdmin - http://www.phpmyadmin.net/home_page/
[42] Gallery - http://gallery.sourceforge.net/
[43] Php5+phpbb problem - http://www.phpbb.com/phpBB/viewtopic.php?t=269488&highlight=php5+admin+panel
[44] ionCube PHP Accelerator - http://www.php-accelerator.co.uk/
[45] The Zend Optimizer - http://www.zend.com/store/products/zend-optimizer.php
[46] Cache Special II – Cache im test - http://www.dynamicwebpages.de/50.tutorials.php?dwp_tutorialID=24&partNo=2
[47] Zend Optimizer – General FAQ - http://www.zend.com/store/products/optimizer-faq.php
[48] Zend Optimizer – General FAQ - http://www.zend.com/store/products/optimizer-faq.php

Benchmarks

Die Benchmarks wurden mit Apaches `ab` durchgeführt. Als Parameter wurde ein Concurrency-Level von 50 (Anzahl gleichzeitiger Zugriffe) und eine Requestsanzahl von 200 übergeben. Hier sehen Sie die Ergebnisse der Caches im Vergleich:

index.php

	APC	afterBURNER*Cache	Zend Accelerator	PHP Accelerator
Time taken for tests	19.963	19.553	17.645	17.825
Requests per second	10.02	10.23	11.33	11.22
Transfer rate	364.19 kb/s	359.66 kb/s	398.28 kb/s	406.54 kb/s
Connect (min)	0	0	0	0
Connect (avg)	184	23	6	1045
Connect (max)	1264	100	28	3834
Processing (min)	982	311	263	738
Processing (avg)	4227	4295	3930	2965
Processing (max)	9330	8311	7290	5899
Total (min)	**982**	**311**	**263**	**738**
Total (avg)	**4411**	**4318**	**3936**	**4010**
Total (max)	**10594**	**8411**	**7318**	**9733**

tutorials.php

	APC	afterBURNER*Cache	Zend Accelerator	PHP Accelerator
Time taken for tests	39.809	39.480	33.742	34.257
Requests per second	5.02	5.07	5.93	5.84
Transfer rate	198.38 kb/s	197.68 kb/s	229.61 kb/s	224.66 kb/s
Connect (min)	0	0	0	0
Connect (avg)	33	625	294	13
Connect (max)	280	4451	1411	55
Processing (min)	578	2361	1638	442
Processing (avg)	8529	8188	7057	7489
Processing (max)	16774	13036	12447	12853
Total (min)	**578**	**2361**	**1638**	**442**
Total (avg)	**8562**	**8813**	**7351**	**7502**
Total (max)	**17054**	**17487**	**13858**	**12908**

Cache Engine Test

Idea

We could do the caching by ourselves on PHP, when the users make a request, we could try to fetch the cached data first. In this way, the caching mechanism could be more tailored-made to the application.

3.3.1.3About the system Security

During the development, I have also studied some ways to secure the server. In fact this is one of the things that I have learnt most from doing this project. Basically, the security issues include a proper firewall setting, PHP script security, and access right control for system services. However, they are beyond the scope of this report.

3.3.2 System Architecture

After making the study about the option of technologies and tools that I can choose from (please reference to the Chapter “Background Study”), I still have not made a choice about the programming style. So I have started off by using procedural PHP. However, when the system grows larger, I have discovered the problem of procedural programming. The major problem for me was as below.

Code maintenance: When the system was started to reach about 2000 lines, I had a hard time about continue to edit the code, because it is very hard for me to find a specific place to edit in these 2000+ lines of code in a text editor.

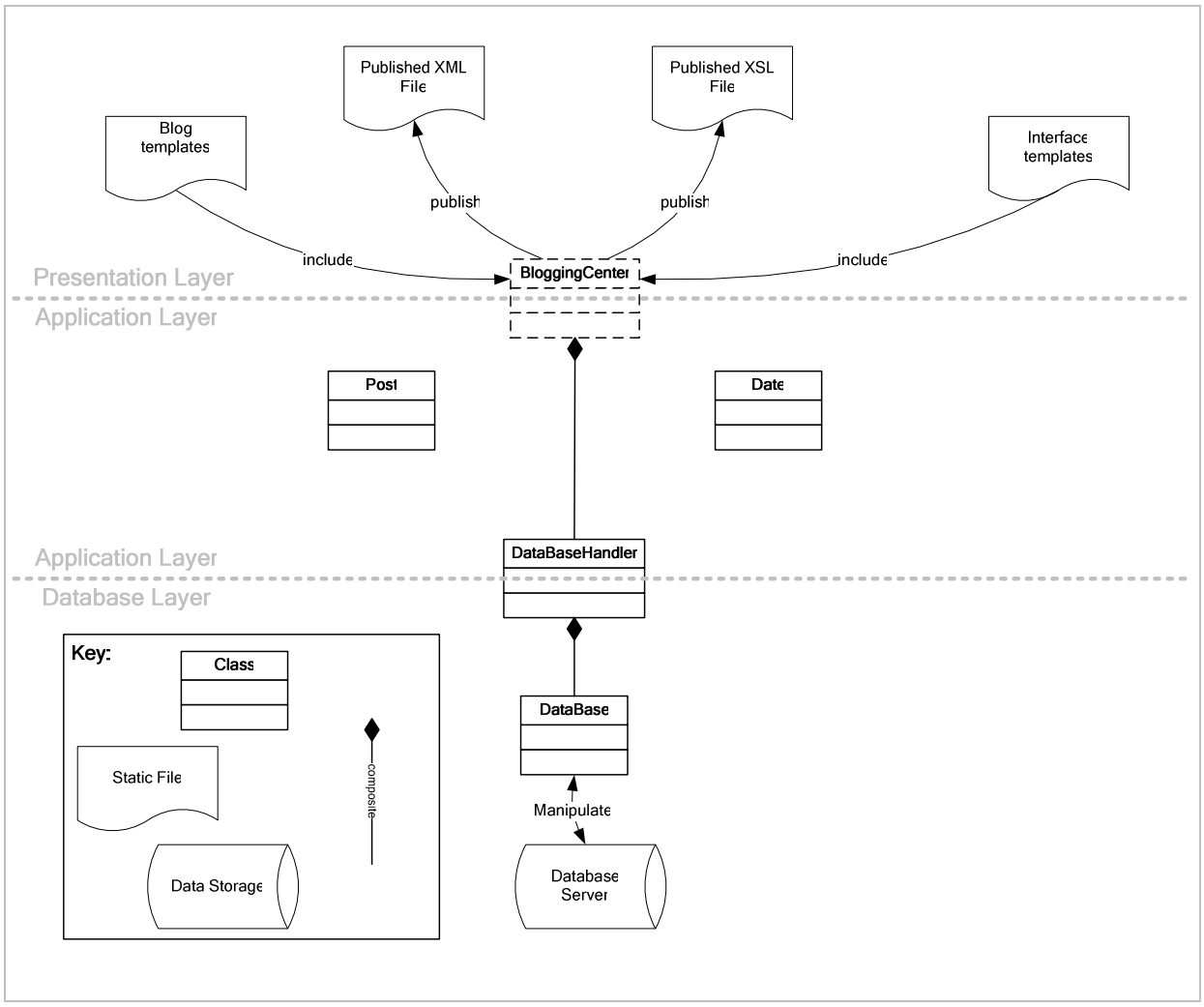
Code readability: Apart from code editing, I also had a hard time in understanding what I have wrote for every time I editing the code again. So I had to make a lot of comment in order to remind me what have I wrote last time. Moreover, I could not make a logical understanding with the procedural code, for example, I have to guess “There are so many Id variable in the problem, so is this Id variable belongs to a post or a user?” for many times.

Code extendibility: When I want to extend the system in some way, most of the time I have to copy the code from places to places, after some time, my code has a lot of repeated sections.

So finally, I have adopted the way to program in an object-oriented style. In this way, I found it is easier to extend the system because everything is abstracted and is very easy to be reused in the future.

3.3.3 Brief Description

The system is developed in a 3-tiers style. The presentation layer is facilitated by the template files which are static in nature. Then the application layer is made up of classes that represent different entities. The database layer involves the database server and a class which handles the database query and query result row fetching operations. The diagram below outlines the system overview.



System Overview

3.3.3.1 Presentation Layer Component

Blog templates are the template files which define the look of the user's blog, so every user have their own blog template setting. They are stored in database and are customizable with the blog setting interface.

Interface templates provide the static part of the Blogging Center layout. For example, a form for which the user input the blog posts and a table at which the user browse the existing posts. Due to the static nature of these files, the dynamic content have to be filled in by the class in the Application layer, by templating engine.

Published XML files are the documents that store the user's posts in XML format. Currently, there are a number of XML files generated because archived blog posts will be published to individual files depends on the date period of archiving.

Published XSL documents are the style sheet for formatting the published XML files. Without these style sheets, the XML files will only looks like raw tagged documents. These XSL style sheets are published depends on the user's blog template setting.

3.3.3.2 Application Layer Component

The class *BloggingCenter* specifies the major logic for the application layer. It behaves as a central point that coordinates the other components in the system. For example, it handles the requests that comes to the web server and do the appropriate thing.

The class *Post* represents a post object in the blogging center. This is a common data exchange format when we need to perform post operation, such as the post fetching action from database, post saving action to the database and post updating.

The class *Date* defines a date object, which is useful in storing and manipulating date information. Date is important when we want to specify the time of blog posting event, on the other hand, it also helps generating id of other objects depends on the current data and time.

The class *DataBaseHandler* provides simple way to access the database, via some pre-defined methods. For example, developers could fetch a post easily by calling the "getPostById($postid)" function. This eliminates the need to run raw SQL query on other classes.

3.3.3.3 Database Layer Component

The class *DataBase* handles SQL query directly on the database server. At the current system snapshot, all of the call to the DataBase class are comes from the DataBaseHandler. It also stores the database server connection once it is established, so the connection could be reused again.

The *database server* is a full set of database management system, which handles the database operation on its own. It is independent of the other components.

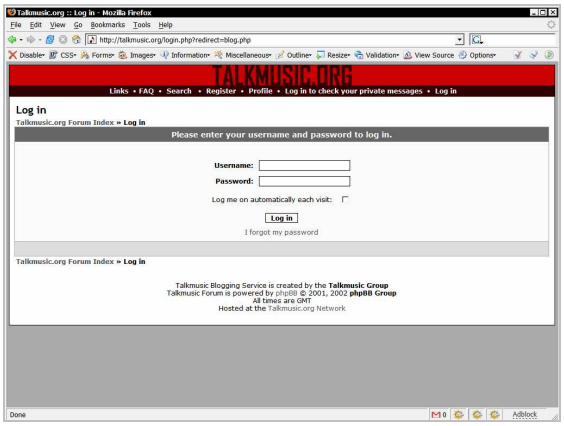
3.4 System Screenshot and Discussion

This section will demonstrate the user interface of the resulting system by trying out all the user functions. We will also discuss the similarity and difference between the

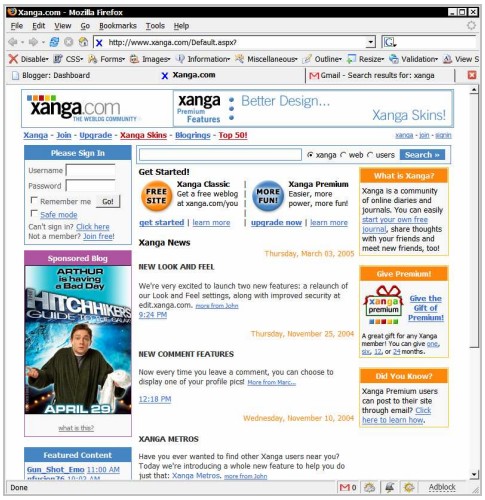
Talkmusic.org Blogging Center and the other existing systems. All of the screen shots are taken with Firefox[49] 1.0.2.

3.4.1 Login Screen

Users have to login before using the system. Notice that the user database is the same as the existing forum system which already resides on the server, so the login interface is acquired from PhpBB.



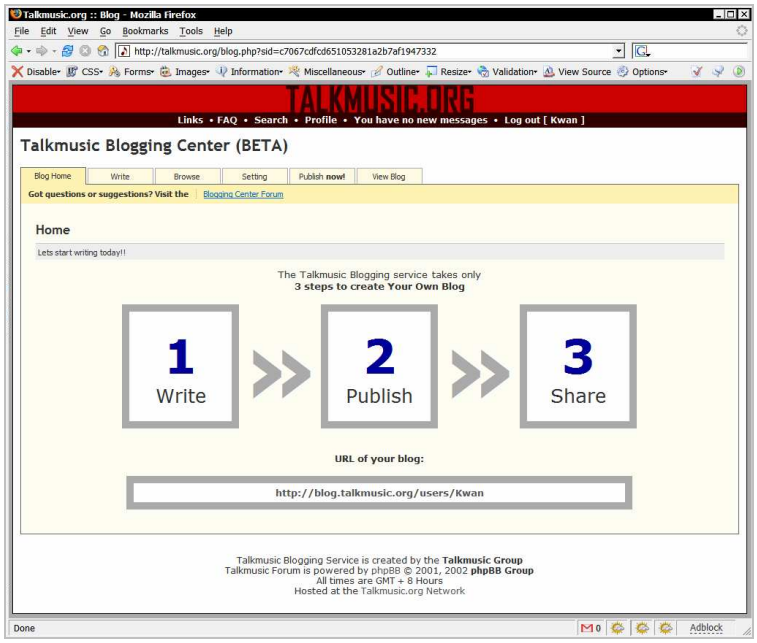
Discussion: The login screen is similar to the login interface of other systems, where users have to supply their own username and password in order to enter the system. The screenshot at the bottom left corner is taken from Blogger.com. They have integrated the login dialogue box into an introductory page, so new users could know a bit about what they can do in blogger.com. For Xanga, they use a 3-column style, the interface is quite complication but users should be easy to find out the login box.



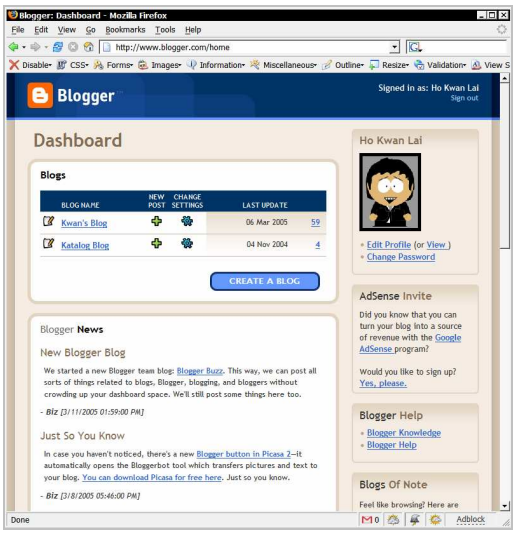
[49] Firefox Web Browser – http://www.firefox.com

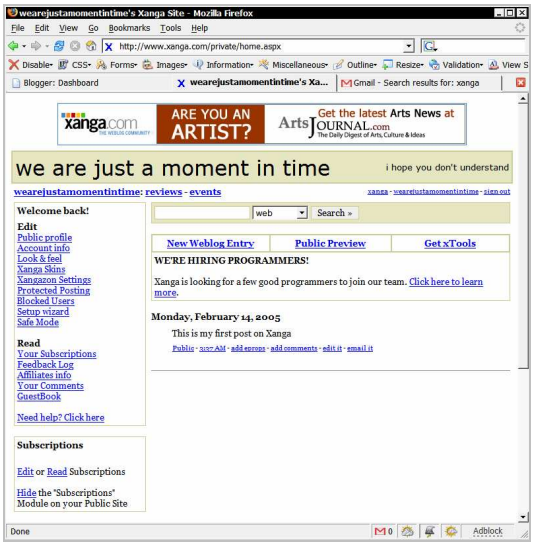
3.4.2 Home Page

When the user has logged in, the user will reach the homepage of the Blogging Center. This will give some simple direction about what the user can do here, and how simple it is to establish a weblog in the Talkmusic.org Blogging Center.



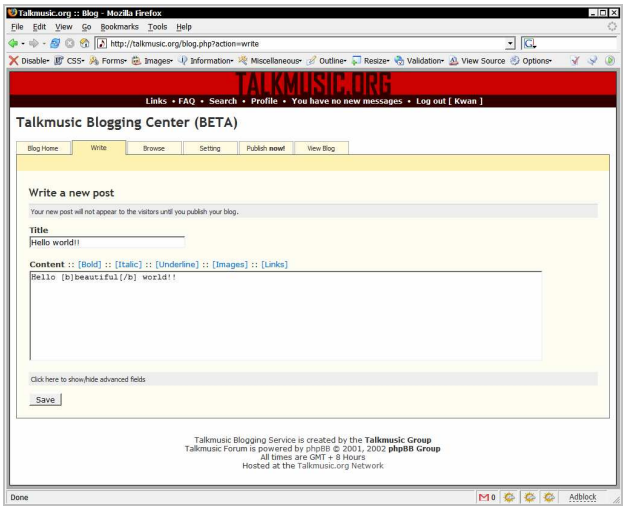
Discussion: At the current state of the system functionality, it is adequate for the home page. Users can also enjoy a clean and simple interface. For the Blogger.com interface, it has more information, while keeping the interface simple. However, the extra information could confuse some new users. For Xanga, the concept is simple. There is a menu on the left, which let users to customize their setting. However, there is a bit too many words on the screen, this may scares impatient users away.

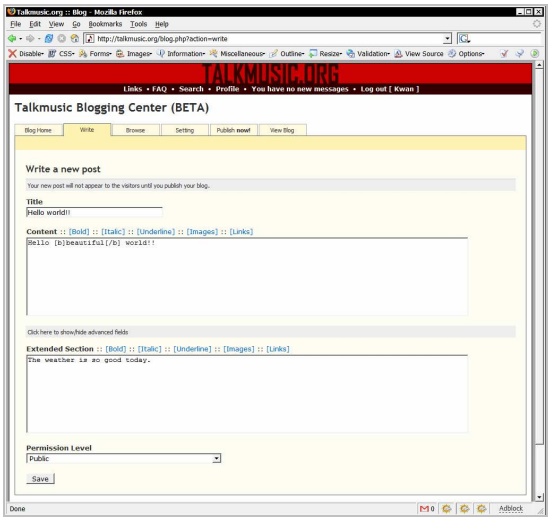




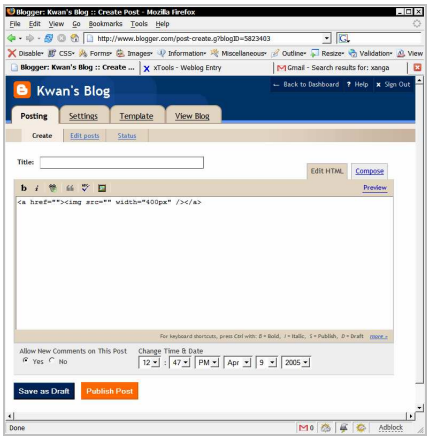
3.4.3 Writing Interface

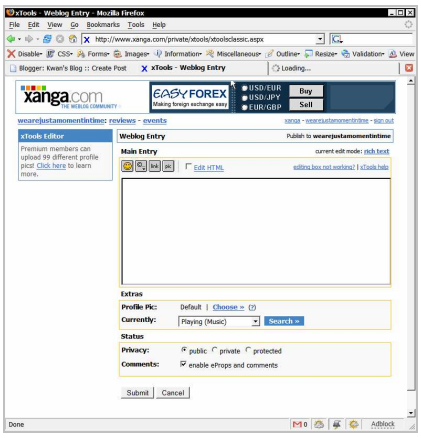
When the user clicks on the “Write” tab, or the label with a big number “1” on the homepage, the user will be brought into the blog writing interface. Then the user can start to write the title and the content of the blog post. Notice that there is an advanced fields section. Two more fields will appear on the interface when it is clicked. The advanced fields section lets user to input an extended section of the blog post, as well as setting the permission of the blog post.





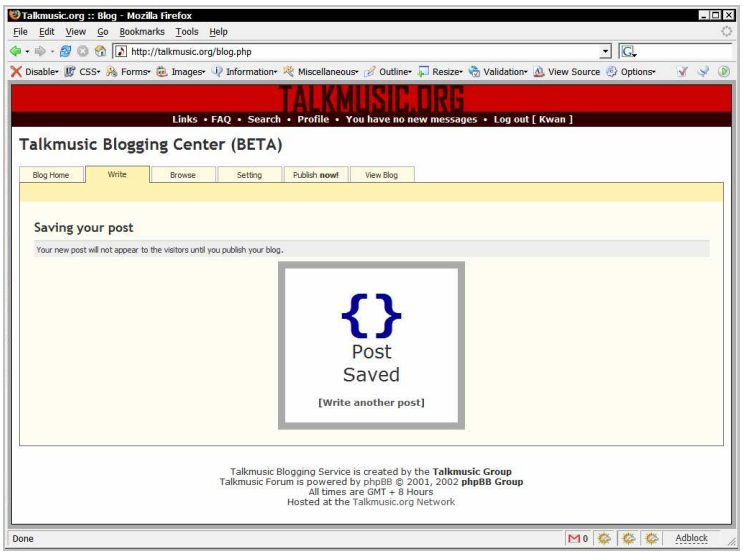
Discussion: The objective of the interface is to keep blog writing as simple as possible. For normal users, they only have to enter the title and content, which is the minimal requirement for a blog post in the Blogging Center. For advanced users, they may choose to enter some extra information. For Blogger and Xanga, they are simple to use too. But it would be even better if they can hide the unnecessary fields by client side scripting. One functionality that the Talkmusic.org Blogging Center can adopt is the preview function which can be found on both Blogger.com and Xanga, so users could ensure the result before actually saving the post.



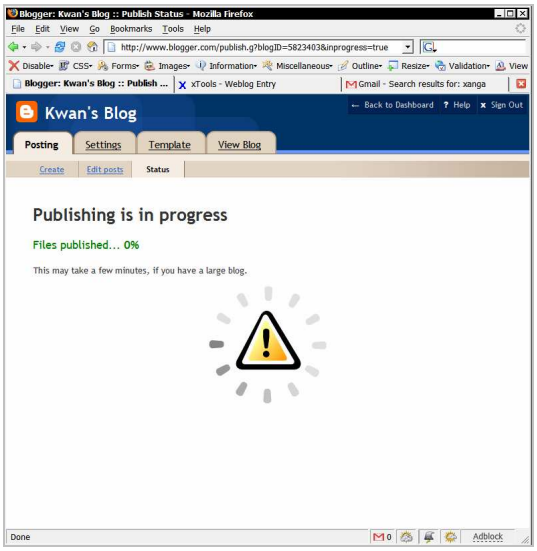


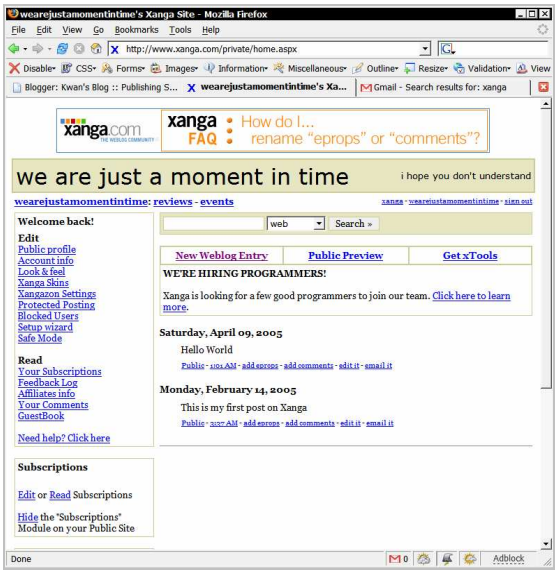
3.4.4 Post-Posting Action

When users save a post, they will be prompted with a notice message. It is also designed to be easy to read and understand. The “Write another post” link is designed to work as a short cut to write another post. All of the notice message in Talkmusic.org Blogging Center have a similar format.



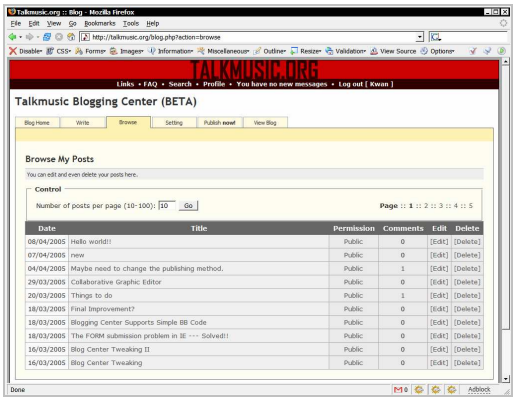
Discussion: The flow of the system would encourage users to review their posts before publishing. So there is less chance to publish multiple times for a single change. For Blogger.com, they let users to choose whether to save the post as draft or to publish the post directly. When publish action is chosen, the user will be brought into an interface which shows the current publish progress, which is very useful for large blogs. For Xanga, the post will be saved automatically, and then it will be publicly accessible if no special setting is set. The user will then be brought back to the home page after the post saving.



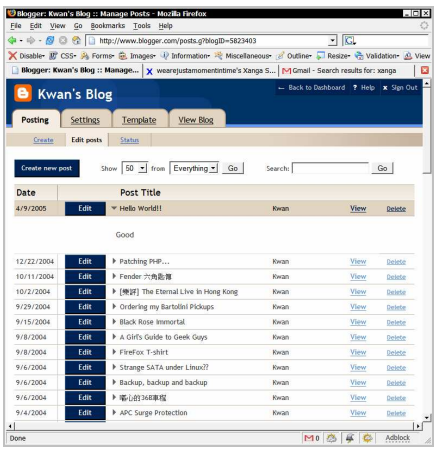


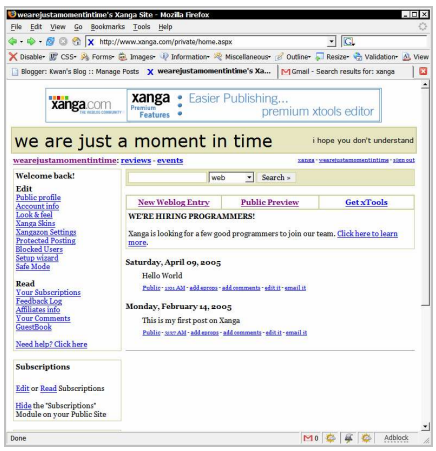
3.4.5 Post Browsing

Users can browse the existing saved posts at the browsing interface. The interface lets users to customize how many posts to be displayed for each page. For every post, there will be option to edit or delete the post. The number of comments for the posts is also shown. When the user clicks on the number of comments, they will be brought to the comment browsing interface. On the other hand, this when the user edit a post, they will be brought to an editing interface which is similar to the writing interface.



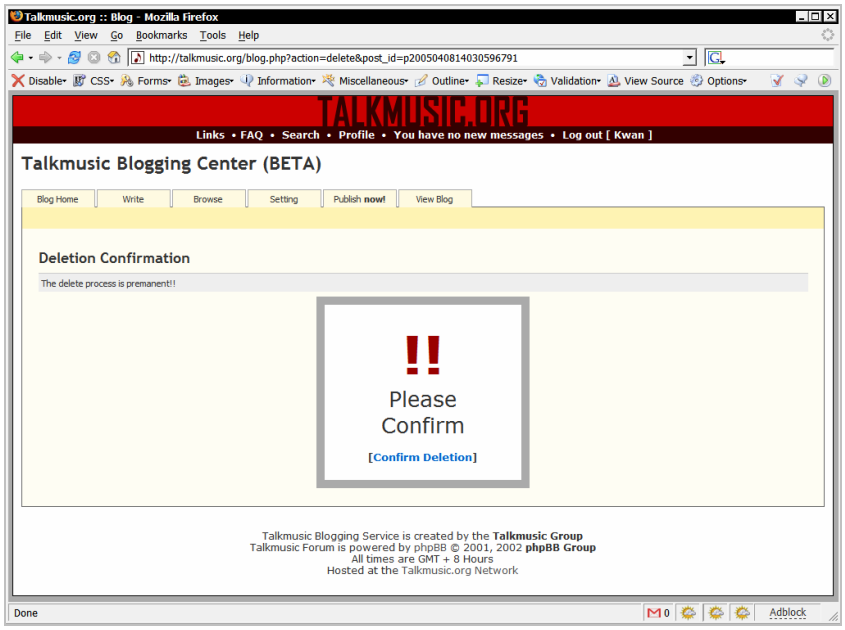
Discussion: The browsing panel used a standard web application interface, which could provide an easy navigation to the posts and comments. For Blogger.com, they provide a similar interface. The major advantage of the Blogger.com interface is they let users to browse the content of a post immediate on the browsing panel. There is a pointer at the right hand side of the edit button. When a user clicks on the pointer, the content of the post will be shown immediately. This is done by using client-side scripting to hide or show the web page content, so the speed is very fast. For Xanga, they use the home page as the browsing panel, so users can have a better picture about how it looks likes when the visitor actually browse the blog. However, the number of post on one page is very limited, so the users have to use more time in locating a post.



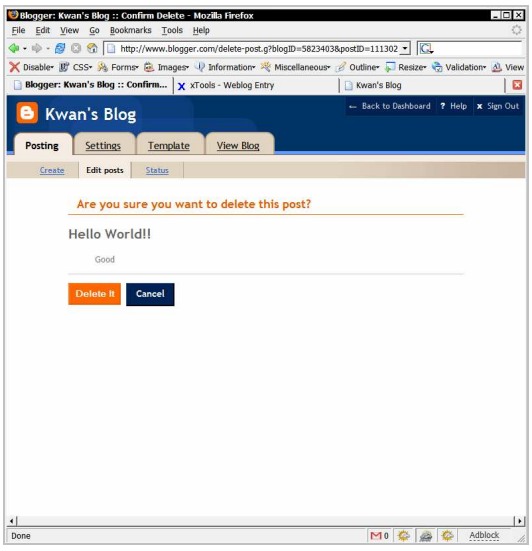


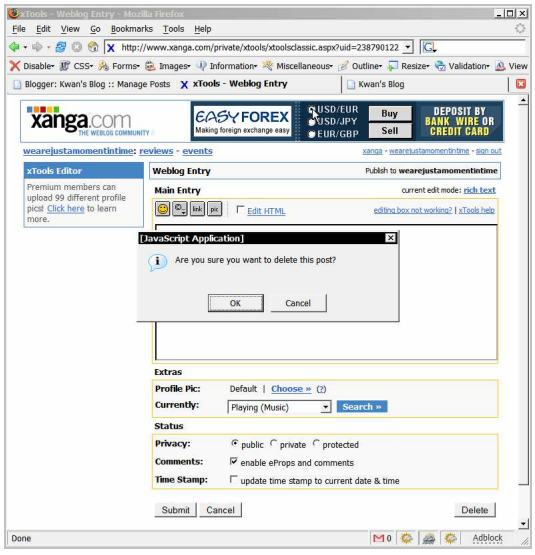
3.4.6 Post deletion

To continue on the browsing panel, when the user clicks on the delete link, the system will ask for confirmation about the post deleting action. The notice is designed so that the user will stay alerted. If the user clicks on the "Confirm Deletion" link, the post will be deleted permanently. The confirmation message before deleting a post can decrease the chance that a user carelessly deletes a wrong post.



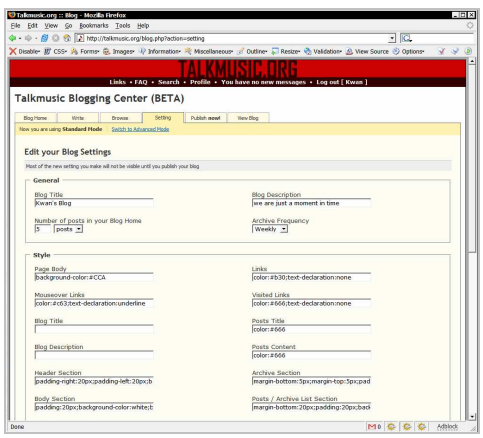
Discussion: The design objective for this interface is to try to magnify the alerting effect of the confirmation message. Blogger.com and Xanga shows up confirmation message before actually delete a post. Xanga uses client-side scripting to call up the confirmation window, the bandwidth and system resources usage thus then can be reserved for this job.

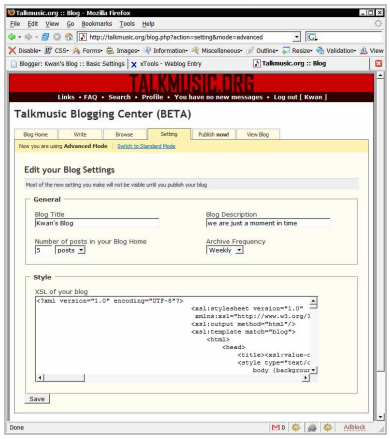




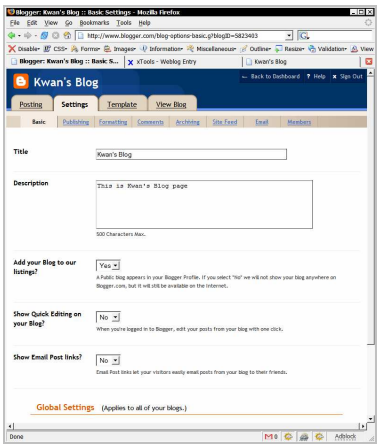
3.4.7 User Settings

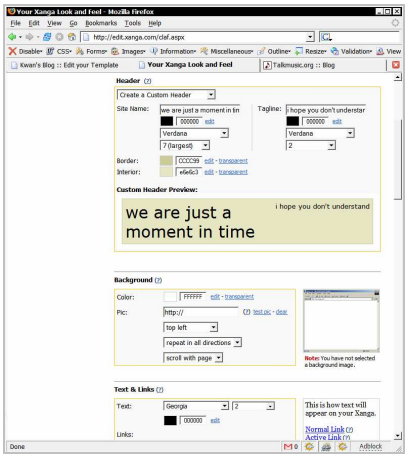
In the user setting interface, users can customize the general information about the user's blog, such as the title and description. Users' templates are also set up here. Standard setting interface provides fields for users to customize the specific part of the template by using CSS. Advanced users may use the advanced setting mode, so they can edit the whole XSL template by themselves. Assume that they have the ability to write a XSL document.





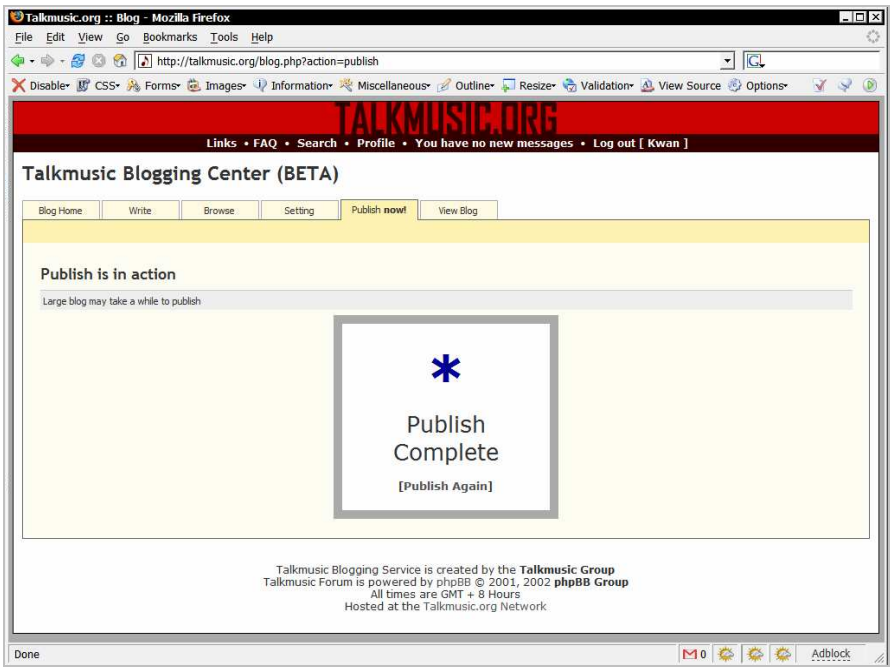
Discussion: The current setting interface of the system can satisfy the current setting options. At the same time, standard users and advanced users could have different interface so to satisfy their ability in tweaking the blog template. Blogger.com provides different section for setting different kinds of setting. This is needed because they provide a lot of setting options for users to customize. For the template setting, Blogger.com lets users to write their own HTML template, this is very convenience if and only if the user knows HTML. On the other hand, Blogger.com provides a lot of ready-to-use templates for users to choose from, this is very useful for beginners and users who do not know HTML. Xanga provides a very user-friendly interface for tweaking the blog template, which will be a valuable reference for the system future development.



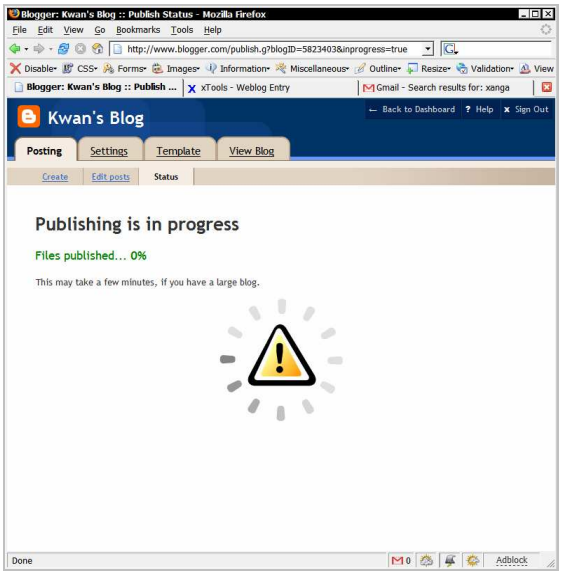


3.4.8 Publishing

When the user satisfies with the content in the browsing interface, the user could publish the blog by pressing the “Publish Now!” tab. Then the blog will be published according to the user setting.

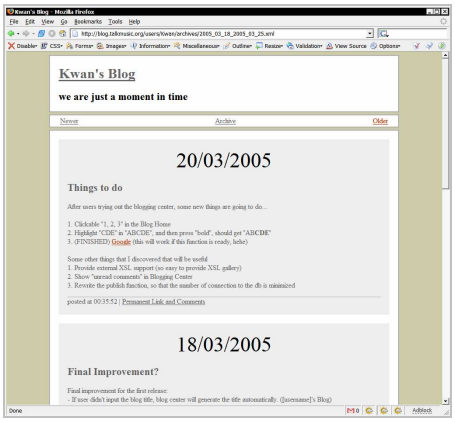


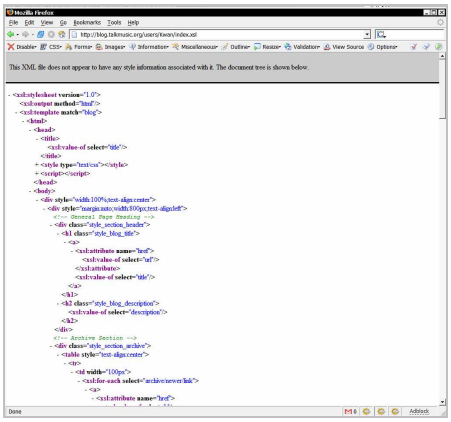
Discussion: The publish completion message is designed so that it is also easy to understand by any type of users. Blogger.com displays the publish progress when publish is in action, this is very useful when users publish large blogs. Xanga does not have this interface because all the posts are saved in the database, and the blog page is dynamically generated when requested, so there is no publishing process required.



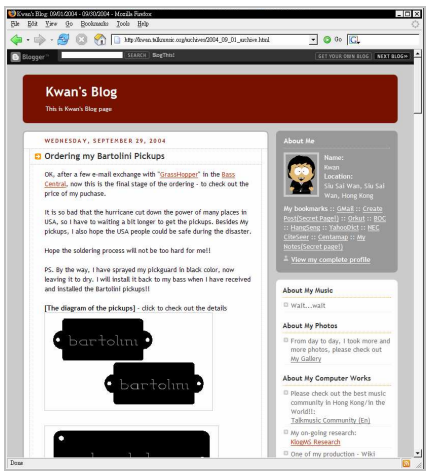
3.4.9 Blog View

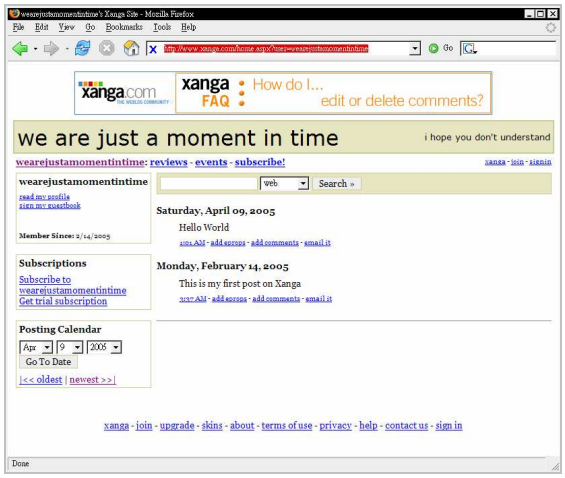
When users press the “View Blog” tab, they will go to their own blog page, which is published according the setting provided by the users. Notice that the blog page is an XML document, which is formatted by an XSL document which is published according to the user’s template setting. The XSL for the blog at the left is displayed in the screenshot at the right hand side.





Discussion: Talkmusic.org Blogging Center offers XML blog view, which uses the advantage of current browsers development, which is the support of XSLT. So the URL of the blog can be unified for HTML and XML content, where the term “HTML content” in here means the HTML code that generated at the client side by XSLT. Blogger.com blogs are in HTML format. Atom feeds can also be generated if users want. The HTML content and the Atom feed are accessed with two different URL. Archive list in Blogger.com is on the blog page, which could cause the list to be oversized when the archive list goes long. Blog view in Xanga blog pages provides blog subscription and blog ring functionality. Archive list in Xanga is not very necessary because visitors could use the “next” and “previous” function to go to the next page and previous pages. Xanga also provides a date form which let visitors to browse posts that close to that day.





4 Conclusion

In this project, different aspect that is related to KlogMS is studied and summarized. The basic implementation of the KlogMS framework has also been done. The framework will be put up on Talkmusic.org as a public blogging service. Once it is put up, we could continue to refine the system by investigating the use experience of users.

4.1 Next step

Due to the reason that the scope and time of this project is limited, I could not make the Talkmusic.org Blogging Center to be the perfect blogging service available. However, the vision is already cleared up about how to enhance the system. Most of these ideas in fact have already been investigated well, but will takes up a significant time to implement. Some of these ideas will be covered in this section.

4.1.1 More features

One of the development concerns of the Talkmusic.org Blogging Center is the extendibility of the system, so that extra modules or web services could be developed to aid the system in the future. This is especially meaningful for enhancing the knowledge management functionality set of the system.

One of the important modules that will be very useful is the categorization engine, which helps to categorize blog posts into appropriate category. The process of categorization is abstract to users, and is done immediately when the user write a post. Via categorization, knowledge in blog posts can be easier to be queried at later time.

4.1.2 Blogging Everywhere

Currently, users can only write blog posts on the web-based interface. This is not bad, but sometimes users want to write the blog in other places or with other kinds of tools.

E-mail to Blog: Users can write the blog post with e-mail editor, and then send the blog post to a specific e-mail address. The Blogging Center will check and fetch the e-mail regularly and put them into the user's blog page.

Desktop editor: Sometimes users may find it more convenience to use desktop tools to compose their blog posts. This maybe because they do not want to spend those extra time which is on loading the web based editing interface. In this case we could develop desktop editor to aid users to write posts without loading the web interface.

Mobblogging: Mobblogging means to make blogging mobile. With the phone with camera, people could upload blog post with photos from their cell phone directly to the blog. However, it is still limited by a number of factors in the current state. The most obvious problem is that uploading a photo from a cell phone could charge a huge amount of service charge[50]. However, the limitation will go away someday.

4.1.3 Published Content versus Dynamic Pages

As we have discussed in the Background Studies section, developers have to make choice between these two approaches. During the implementation period, I have used the published content approach to get things done. The system works fine now. However, I have found this is not very capable of handling category or label structure. Theoretically we will need n sets of pages if we have n categories, so that will be very messy when the user has a lot of categories. One way to solve the problem is to use client-side scripting to control the view of the page, this maybe done by changing the content of the XSL file on client side.

However, the logic and implementation method will be much simpler with dynamic pages. So we still have to consider to change back to dynamic pages even they uses more system resources generally.

[50] Weblog goes mobile - http://www.ojr.org/ojr/glaser/1043947239.php

Appendix A. Interim Report

City University of Hong Kong
Department of Computer Science

BSCS Final Year Project 2004-2005
Interim Report

Project Title

A Weblog-based Knowledge Management System

(Volume 1 of 1)

Student Name	:	**Lai Ho Kwan**
Student No.	:	**50333816**
Programme Code	:	**BSCCS**
Supervisor	:	**Dr. Andy Chun**
Date	:	**19th, November, 2004**

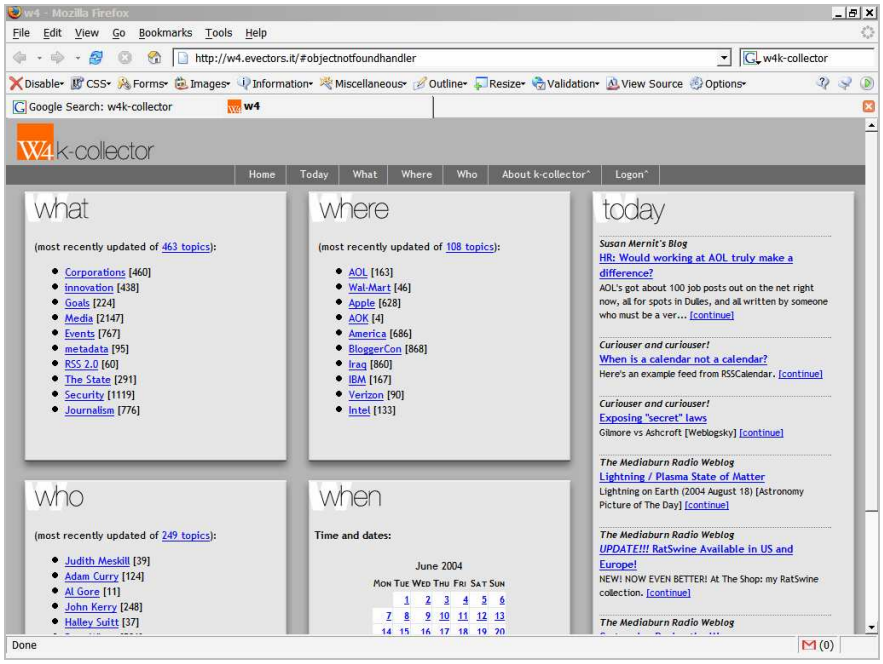
For Official

** Delete where not applicable.

"Decentralizing for innovation, centralizing for reuse." –Michael Angeles

Abstract

Many Knowledge Management System (KMS) have been developed to handle knowledge. Companies may use them to retain and manipulate knowledge from the employees. However, there are many problems that found on many KMS, which will prevent a KMS to be successful. To overcome the problem, we propose to merge up the concept of weblog and KMS. Then the major feature of the resulting system could be a new kind of KMS which is probably more usable and could really do its job. To assess the system, it will be open for public usage. Users are then asked to finish a survey-like list of questions. Then the result will be analyzed and be discussed. Furthermore, additional feature requests will be received from users. Then these features will be carefully analyzed. Finally they will be integrated to the system if we find it is worthy, by the best approach available. On the other hand, in order to optimize the system, hardware resource usage will be continuously monitored. If there is suspicious operation which takes up a lot of hardware resources, we will study the way of optimizing the specific operation. Finally, we could know if it is a good approach to merge up the two systems from the analyzed result. After the system reviewing period, we could also have an overall view on the features that is appeal to the users, in a Knowledge Management System. On the other hand, we could discover and understand the methods of optimizing a similar system in terms of hardware resources.



Screenshot – A preview of the system? [W4KC]

1. Introduction

Many KMS have been developed to handle knowledge. Companies may use them to retain and manipulate knowledge from the employees. However, there are many problems that found on many KMS, which will prevent a KMS to be successful.

1.1 The Existing Problems

Some of the problem will make the users not willing to enter knowledge into a KMS. The problem may come from an overly-sophisticated user interface or data input procedure. Another possible reason is due to psychological habits of human that they will more likely to pour knowledge into something that they can depends and have a passion on. Some cases maybe due to the system performance. Users will not be comfortable to work with a system that is very low in performance. This problem persists normally because not many server administrators (or their domain) have the budget to upgrade the hardware.

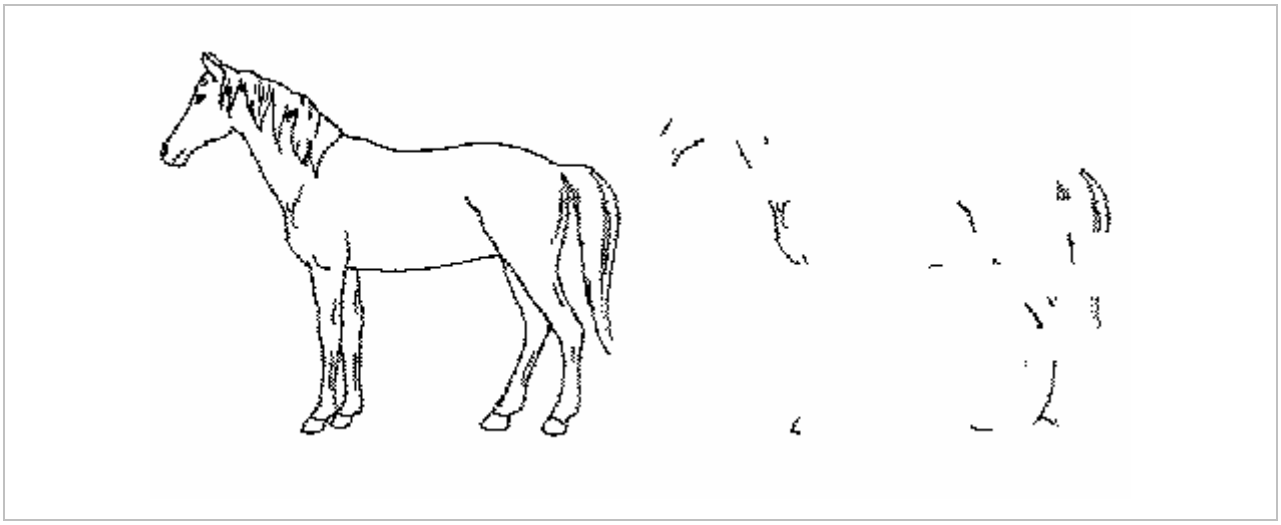
Some people have thought of using weblog as a simple KMS. A weblog used in this way is described as a K-Blog (Knowledge Weblog). By using this approach, the user interface problem could most probably be solved, because most of the weblogging system is designed to be user friendly. However, almost all of the existing weblog are not designed to handle knowledge. For the others, they could be overly-sophisticated again or just went in a different direction instead of becoming a KMS. One of the major features to organize knowledge is the category system. A typical weblog system aligns information only in chronological order. Some better system will provide simple category function. But the category system found in these systems is nearly always designed to work with a single user only. As a result, in a typical weblog system, when a lot of information is published by user(s), users will then hard to find the desired information at later days. When a typical weblog system is used to handle knowledge in a large company, another problem raised. Because there are too many individual blogs, it is very clumsy to manage these weblogs which is written by a lot of users, individually. Even by using feeding methods like the RSS, the result would be a bigger, but still a typical weblog which does not mean any improvement in terms of information structure.

1.2 Research Objective

In our research, we are going to try to improve the current KMS systems. First, for the user usage experience, we have to tackle the major problems listed in 1.1. So we propose to add the weblog idea to KMS. At least the user interface problem in a conventional KMS should be eased. Secondly, we will try to evaluate the user experience and performance of using a category system. Although, theoretically, the system will be benefit from the advantage of category, we still have to make sure if it works in actual usage by the users. Thirdly, in the research, we will try to discover features that are found to be useful by the users, and at the same time, we could also know which feature is not really needed in this kind of system. Finally, another major aspect of the research is the optimization of the system. This is another critical point because we always want to do more jobs with the same equipment.

1.3 Importance of the research

If a better kind of tool is being used for knowledge management, more knowledge could be retained because the knowledge management tool will be used in a higher frequency. Another reason for the above benefit is that human tends to forget something even in a very short time, so more knowledge could be retained if they write it down quick. Furthermore, the same quality of knowledge will be even greater if they are stored in a well-organized structure.



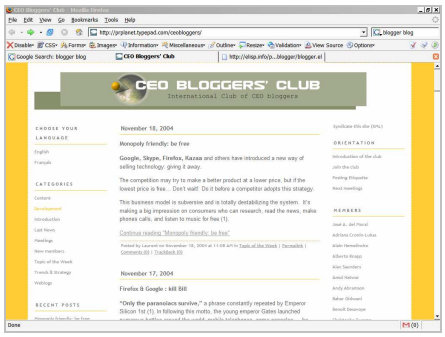
Advice - Let's write it down before you forget! [SNODGRASS1980]

2. Related studies

Our understanding of human cognition in decision making and knowledge-intensive work is marginal [Wiig1999]. There is not an accepted economic "theory of knowledge" that is applicable to business. We do not have a general understanding of how to undertake comprehensive and systematic KM within an organization and may need a new theory of the firm to manage knowledge effectively – and to link it with enterprise strategy, tactics, and daily operations.

Knowledge Management offers the possibility of allowing organizations to tap into not just the documents they've created, but the expertise of their employees, past and present [AMY2001]. Most KM initiatives fail because of the lack of content [MICHAEL2004]. It is important to bring people to verbalize their thoughts and make them available to others. It is always a problem about how can the knowledge management tool make that as close as possible to a conversation, which is far better to connect people than long structured articles. A diary benefits its writer because it helps him clarify his thoughts. It also benefits its reader because it can be a starting point for a very meaningful conversation, one I am confident that the writer will jump into. A weblog is really an incubator of shared ideas.

The term Weblog or Enterprise Weblog does not readily indicate to a reader what is the technology [TRACT2003]. People, in general, are starting to figure out that a Weblog is a personal journal on the web. Once the technology is more mainstream, people will see the Enterprise Weblog as means for relevant business communication. Weblogging is interesting because it is a fairly non-intrusive way of allowing workers to share the process by which they seek, analysis, and select information [AMY2001]. Weblogs have a very important role to play in the innovation process, and thus should be considered very seriously in corporate environments [MOPSOS2004].



Screenshot – A Blogger published blog [BLOGGER]:
Even there is categories, it is not scalable enough to use in a KMS

A popular weblogging service Blogger [BLOGGER] is a kind of typical weblogging system. However, it does not support categories, which is very important in a KMS. The web is huge. One way to help people find useful pages is a directory service [ERIC2002]. The same could be applied to a KMS. The directory service is just like the category function. Such as Yahoo! [YAHOO], or The Open Directory Project [DMOZ]. Typically directories are manually created, and the judgments of where a page goes is done by a human. For example, Yahoo! puts "General Motors" into several categories: "Auto Makers", "Parts", "Automotive", "B2B – Auto Parts", and "Automotive Dealers". Yahoo! puts itself "Yahoo!" in several categories including "Web Directories.".

Screenshot - The Yahoo Web Index, 2004 [YAHOO]

To put it all in all, the resulting KMS should be "Decentralizing for innovation, centralizing for reuse." [MICHAEL2004]. Value is added to existing information through a process of aggregation, simplification and dissemination, which is exactly the concept underlying a web of blogs. In essence, the more decentralized, the richer information is; the more centralized, the more structured it is.

3. Methodology

To assess the result of our research idea, we will develop a system called the Katalog. I have already confirmed to write the system in PHP, which then will be running on the Apache HTTP server.

3.1 Assessment Method

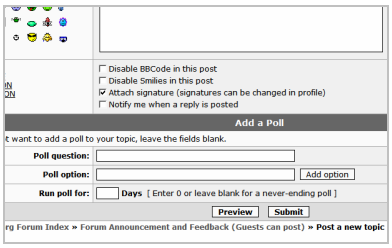
To try to assess the result of the system, I will finish the development of the basis (or the "skeleton") of the system at the end of semester A. This "skeleton" will be fully workable and should be under beta testing period at the end of Semester A.

The system will then be open to public for beta testing. I will use the Talkmusic.org server for testing purpose, because there are already a number of users who use the Talkmusic.org Forum. As a result, we should gather more comments in the run. The more result that we gather, the more accurate our study will be, because the result will reflect the user group which compose of more user samples.

3.1.1 Use experience assessment

Continuous review will be acquired from users via survey. The survey will be conducted on the internet. The format of the survey will be figured out when the "skeleton" of the system is finished. However, it is already confirmed that the survey will contains questions that ask about the users' usage experience, in order to see if the research has met.

Tools will also be used to acquire comments of the system from the users continuously. Then during the semester B, the Katalog system will be continuously improved base on these acquired comments. For the comments that are acquired from other channels other than this specific tool, I will also copy them into this tool, in order to maintain a centralized pool of comments. The "tool" used here is not confirmed, and maybe just a category in the system.

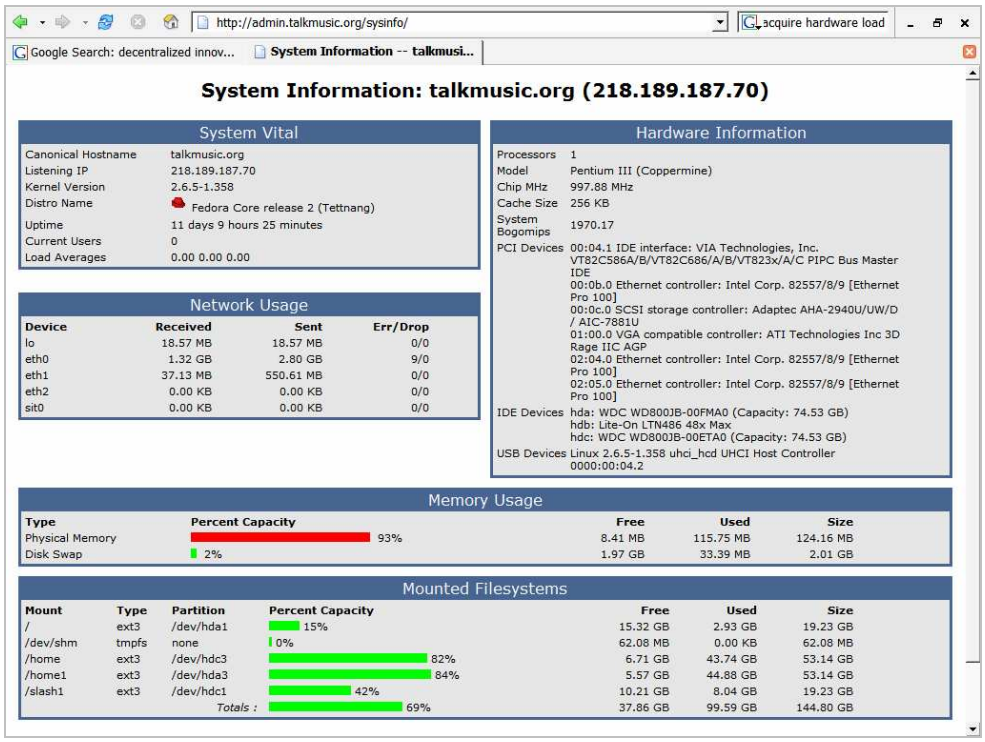


Screenshot – A possibility is to use the "Poll" function in forum systems

3.1.2 Hardware resource usage assessment

On the other hand, I will continuously monitor various aspect of the hardware load. I will monitor the hardware load in terms of the server loads index, CPU usage, Memory Usage, or network bandwidth usage. If there are hardware changes during the assessment period, I will also note them down and try to eliminate those hardware factors, as our research objective is about how to do more jobs with a single piece of equipment.

The received data will be then treated like the usage experience review and feature investigation method. But for the hardware usage statistic, I will try to note any suspicious operation which is not optimized, and then try to figure out the way to optimize those operations. For example, for a SQL statement or a line in the PHP program, it could be written in different ways so that it will takes up different hardware resources. So a specific study will be done on this specific area, in order to find out the best solution.



Screenshot – The current way to know hardware load, however, the information is not persistent.

3.1.3 Handling of the data

After the long system run in the Semester B, the statistical data will be discussed through a specially designed procedure. This procedure will be designed near the end of semester B. Although our major research objective (i.e. "It is a good approach to combine weblog system with KMS") may be refused, we could gather more information about making a good KMS or weblog system, because there is other information acquired about other objectives, for example, about the features that they are wanted. Moreover, we could figure out various ways of optimizing our Katalog system, as well as other similar systems. All of the discussion and study result will be written another separate report.

.

3.2 Design and Implementation of the system

To sum up the development objective, I will first build the system with only the features that implement our research objectives (the major one will be "Merging KMS with weblog", for all of the objectives, please refer to the section 1.Introduction). For the other features, that will not be built from the beginning. Instead, they will be built under the procedures described in the following section "Handling feature requests.

3.2.1 Handling feature requests

We will not sudden develop a feature without users' request. When a user is requesting a feature, the feature will then first be studied in various aspects. For example, when a user request a feature which is about "Tell the Katalog to generate e-mail when my blog page is being reviewed/commented", I will try to figure out the pros and cons of such a features, then figure out the best approach to do so. When everything is clear, I will implement it and wait for the users' review/comments on the new feature (if any).

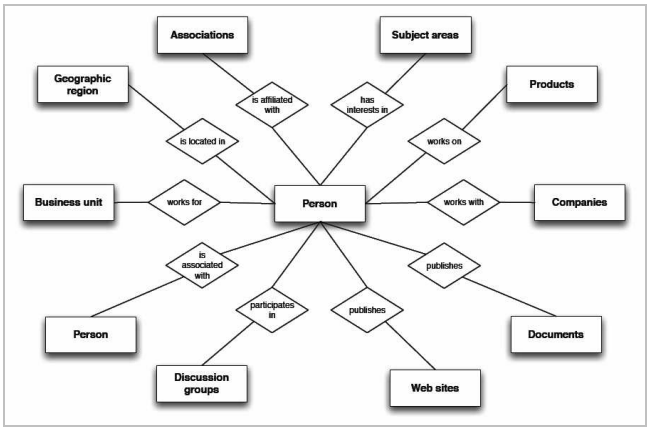


Diagram – Semantic Level features may be proposed by advanced users [MICHAEL2004]

3.2.2 Overview of the "skeleton"

At the beginning, the Katalog will only contain features that are going to be evaluated, which are the objectives on the system that are listed in section 1 (Introduction). So the basis of the Katalog "skeleton" will be containing the following properties/features.

- Every user has one or more personal blog
- If a user chooses to publish one of his/her blog post to a category, the post will show up in that category
- The category index is freely browsable by everyone (in the range that is defined by the server administrator, for example, a LAN, or the entire Internet)

There are also some features that are obvious to include in the skeleton of Katalog. These features have received positive comment from users of other system. The detailed analysis of these features will be done along with other feature requests. The list of these features will include the following.

- Blogs are possible to be published to an external FTP account
- Blogs can have a (pre)customizable template

3.2.3 About the Category System

In the skeleton version, categories are managed by an administrator of the Katalog. For example, if a user wants to add a category "Foo" to the Katalog, he must ask the administrator to it. In the future, when there is an automatic categorization tool available, the Katalog administrator could choose to let the automatic categorization tool to do the same thing. These tools most probably will determine the suitable category for a post by its content. However, this tool is not in the scope of our research, but instead, it is on going by another person who called Jack Shum.

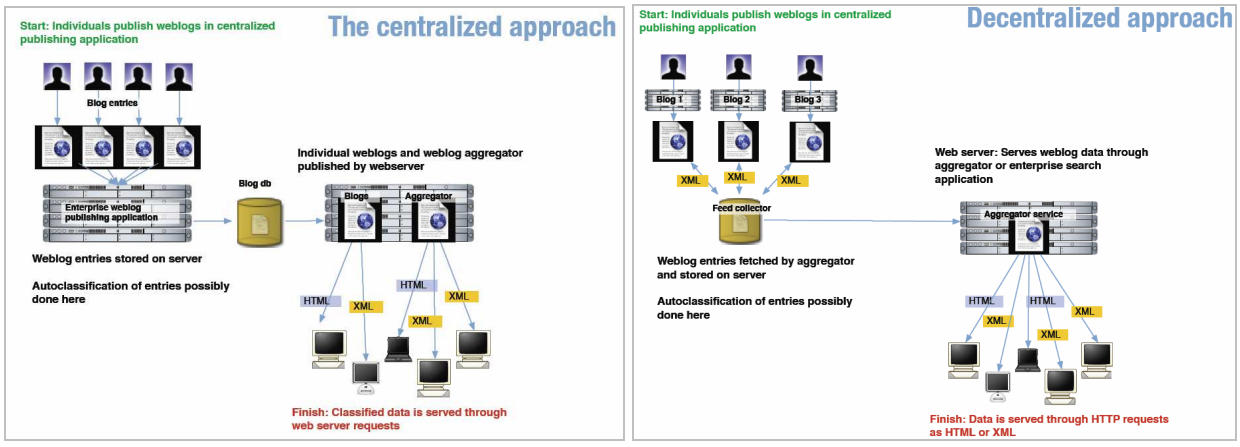


Diagram – An approach proposed by Michael Angeles [MICHAEL2004]

References

[AMY2001] Wohl, A. D., "Life On The Internet: Could Blogging Assist KM?", Amy Wohl's Opinions, http://www.wohl.com/wa0156.htm

[BLOGGER] Blogger Weblogging Service, http://www.blogger.com

[DMOZ] Open Directory, http://dmoz.org

[ERIC2002] Glover, E., J., Tsioutsiouliklis, K., Lawrence, S., Pennock, D., M., Flake, G., W., "Using Web Structure for Classifying and Describing Web Pages", In International World Wide Web Conference, http://citeseer.ist.psu.edu/cache/papers/cs2/155/http:zSzzSzwww.neci.nec.comzSz~lawrencezSzpaperszSzclass-www2002zSzclass-www2002.pdf/glover02using.pdf

[Wiig1999] Wiig, K., M., "Successful Knowledge Management: Does It Exist?", http://www.krii.com/downloads/does_successf_km.pdf

[MOPSOS2004] Mopsos, "Ecology of weblogs for business", http://blog.mopsos.com/archives/000081.html

[MICHAEL2004] Michael, A., "Supporting enterprise knowledge management with weblogs: A weblog services roadmap", A presentation in Washington: Computers in Libraries 2004 conference, http://urlgreyhot.com/personal/cil2004

[SNODGRASS1980] *Snodgrass, J., G., & Fan, J.*, "Experiments in Human Memory Programs and Pictures", http://lifesciassoc.home.pipeline.com/instruct/humemory/ehm.htm

[TRACT2003] Traction Software, "Post Modern Knowledge Management and Social Enterprise Blogging ", http://www.masternewmedia.org/2003/02/07/post_modern_knowledge_management_and_social_enterprise_blogging.htm

[W4KC] W4K-Collector, http://w4.evectors.it/#objectnotfoundhandler

[YAHOO] Yahoo! Search Engine, http://www.yahoo.com

Appendix B. The conference paper

There is a conference paper written and got accepted by the CCCT conference during the project period. The paper is available for download at Andy Chun's Website.[51]

Information about the paper:

Chun, H.W. and Lai, H.K., "KlogMS - Semantic Knowledge Chunking," In the *Proceeding of the International Conference on Computing, Communications and Control Technologies*, August 14-17, 2004, Austin, Texas, USA.

[51] Dr. Andy Chun – FYP Students Publications - http://www.cs.cityu.edu.hk/~hwchun/FYP/publications.shtml