Knowledge Management and Web Content Management Systems

Thomas Ferris Nicolaisen

University of Oslo, Department of Informatics

29. November 2005

Content

Introduction

Open standards versus software development is a discussion that can be pulled into several fields of academia, from programmatical software engineering and methodologies to organizational structure and anthropology. Even though I am a programmer at heart, I choose an approach via the scene of knowledge management for two personal reasons, which I am sure others can relate to.

Foremost, the principles behind knowledge management take on a more human approach than traditional software engineering. Even though a knowledge management process will at some point include hard programming and software development, the process as a whole has a more noble goal: While the goal of software engineering is to make software smarter, the knowledge management goal is to make **people** smarter[1].

The second reason is that my favorite category of information system at the moment is the one of content management systems. Knowledge management system implementors in most larger organization probably recognize the content management system (CMS) as the heaviest and most complex piece of knowledge management software[2].

A CMS is a piece of software responsible for taking care of all the digital documents and files in an organization. The functionality of such a system includes document repository control (the company's digital library), which by itself is quite an advanced implementation depending on whether it includes access-rights management, product management, versioning, data import/export, workflow and search functionality[3]. These systems were invented because the amount of digital content exploded through the last two decades.

[1] http://www.phredsolutions.com/glossary.html

[2] http://en.wikipedia.org/wiki/Content_Management_System

[3] http://www.cmswatch.com/Feature/53

After the successful upbringing of the world wide web, the same problem occurred with company websites. As sites grew in numbers of pages, their content became intangible and unstructured, upgrades and redesign became a hassle[4]. The demand of layout and graphics increased, but the skill of doing this was stuck with the web designers, not the page authors.

Thus web content management systems was introduced in all their shapes and sizes.

A WCMS is responsible for making sure the organization's website displays the proper information in a proper way. A very basic WCMS was, and still is, often built in-house by the company's web designers as dynamic web pages with template engines, running on top of a relational database with the actual content.

Fortunately, websites have evolved quite a bit since the dawn of the world wide web. They are no longer flat and simple web-pages, but now feature searching, media content, surveys, news-feeds, mail subscription, web shops and more. They can act as large web-applications which in theory can match the functionality of normal desktop applications. Modern web applications induce the ability to put more CMS-functionality into the WCMS.

They already have a lot in common. They display some of the same content (like company and product information) and they both have functionality like searching. Perhaps the company has put its knowledge base online for assisting customers[5].

If the website has a username/password sign-on for employees, you suddenly have an "intranet" on your WCMS. This magic word releases a multitude of possibilities, because as soon as you can verify the identities of your employees or members online, they are potentially able to do everything they are normally able to do on your CMS. The key advantage of doing your content online is portability. Your users can access and modify content from anywhere in the world, as long as they have an internet connection.

The company's WCMS, which controls the digital material published online, can either be implemented as a front-end to the company's CMS, or as a stand-alone application. Since many companies have no suitable CMS in place, or their CMS lack a proper web front-end, the latter solution is often the case.

Profiling the WCMS as an isolated product has resulted in quite a number of WCMS-products available[6], some of which are based on an open source business model. My theory are that these open source WCMS are ideal (read: largest ROI, smallest TCO) because they are open and because they embrace open standards.

Before I go on elaborating on open source and open standards, I wish to devote one chapter to the requirements of a WCMS, and why they are so difficult to define.

Requirements of web content management systems[7]

The most important functionality of the WCMS is the composition and **publishing** of articles, simply put, the art of writing something and putting it online. Advanced composition features a WYSIWIG-editor, spell checking, insertion of pictures, tables, etcetera. It should also be possible to later edit published pages, as well as taking them offline, hiding them from the public view without deleting them. The last point is actually part of the next requirement.

Workflow is a feature of WCMS featuring several authors and perhaps an editorial staff. A web page or document has status which perhaps only certain individuals are authorized to

[4] http://www.diss.fu-berlin.de/2004/319

[5] http://en.wikipedia.org/wiki/Knowledge_base

[6] http://cmsreview.com

[7] Bob Boiko, Content Management Bible 2nd ed, 2004

change, for example the editor accepting an article for publishing. Time-limits are also part of the workflow. One page can be scheduled to go on- or offline at a given point in time.

Administration becomes necessary when website structure grows complex. Pages must be moved around, sorted in folders and categories, referenced and tagged with meta-data.

International companies need websites with **internationalization** (content in several languages) and **localization** (country specific date and economic format).

To increase the availability of content, larger websites feature **syndication**, or off-site publishing. This can be approached by subscribing to receive new pages through e-mail (newsletters), or as the increasingly popular news-feed (RSS).

Templating is not as much a feature as it is a necessity. A company is often judged by the outlook and consistency of its website. While the web designers no longer need to author the content of web sites, they still need full control of the site. Templates allow to do the design once, and then apply the same design to whole parts of the site at once.

The importance of the last requirement is proportional with the size of the website – the **search**. Although a very basic search engine is sufficient for most sites, it is also possible to implement smarter searches that accord for miss-spelling, try different word ending(s), and use context specific dictionaries. A good search engine also searches your online binary files (PDF and Microsoft Office documents for instance).

If a WCMS satisfies all the above requirements to certain degree, it has come a long way already. However, as companies realize the potential of their websites, they come up with new needs an requirements. I will name some recent cases of personal experience:

--

An organization selling membership-articles used their website for displaying product information on their merchandise. At one point, they decide it would be better if merchandise can be bought directly off the site and require a webshop functionality built into their WCMS.

--

The Norwegian Association of Allergic people daily update their website with the most recent pollen-forecast. They realize their members would like historic information and require the page to connect to a repository of pollen-numbers, where queries can produce graphs of pollen levels.

--

A trading company see the need to feed various stock quotes on to their front page automatically. A service reading information from another site needs to be set up and connected to the WCMS.

--

A news-company comes up with a new category of news which should be displayed for paying-customers only. Customers gain access to this privileged part of the WCMS by entering a personal user name and password on the site.

--

Another economic consultancy company decides that their web page should also display information from a data sheet (which is being constantly updated) directly onto their site. The WCMS needs a data sheet reading service which flushes its content onto a web page at regular intervals.

All these custom applications can be implemented outside the WCMS, but sooner or later they need to interconnect in some way or the other. This is where commercial off-the-shelf WCMS fails. They are not customizable enough to allow such modifications done by the web designer. Including new custom functionality may prove to be very expensive for the WCMS user.

Open standards

A standard is an agreement of two or more parties regarding a product, specification or other. Standards used by web applications are mostly guarded by the Internet Engineering Taskforce (IETF) and the World Wide Web Consortium (W3C). Examples are hypertext markup language (HTML), hypertext transfer protocol (HTTP) and resource description framework (RDF).

System developers can choose either to use existing standards or invent their own. Sometimes not having to follow a standard is easier and quicker than having to fulfill the specification's every need for quirky details, but along the circumference of the system where other systems interact, open standards must be implemented. This applies to the technology used for **transport** (all applications connected to the world wide web must be accessible through HTTP) and **storage** (a web page must output format in HTML, pure text or a standardized binary format like Bitmap pictures or even Macromedia's Flash), some of which are subjectively recognized de facto standards.

Proprietary standards can be open (Adobe's PDF format and Macromedia's Flash file format) or closed (Microsoft Office Word documents and Powerpoint presentations). A proprietary standard can only be changed by its owner. You can make software that reads both open and closed standards, but figuring out how the closed standard is built up internally can be quite difficult, and often not cost-effective. This process is often coined reverse engineering.

A WCMS will naturally output its content through HTML on a web site. Internally, however, the implementation may store the content in a home-grown format, for example a relational database with a streamlined scheme following no standard whatsoever. As long as the owning company uses the WCMS normally, the inside workings of the content repository is not important. The problem arises when the company either wishes to change the output or use of the content, or to replace the WCMS all together.

How will the content be exported from the old WCMS and imported into the new one? Manually copying the HTML code from each web page will no doubt be a very tiresome effort. Another alternative is to read content directly from the relational database with an exporter-application. If the WCMS has not supplied one, developing this application could be a large task. And then an application would have to be developed for importing the content into the new WCMS.

The best solution would be if the storage of both WCMS-es utilized a standard content repository, so the content of the old system could simply be dragged-and-dropped into the new one. Unfortunately, today there exists almost as many different content repositories as there are content management system vendors.

Open source

Simply put, open source software refers to programs whose source code is made available for use or modification[8]. This means that open source software is in fact free.

A lot of people find this hard to believe, and many presume that such software is produced on a volunteer basis, and therefore lacks quality, security and consistency. This is true for a lot of smaller open source projects, but larger companies do in fact develop open source software on an economical business model[9].

[8] http://www.free-soft.org

[9] http://www.manageability.org/blog/archive/20030611%23101_ways_to_make_money1/view

The revenue can be generated by offering support, customization and plug-ins. Large software companies have for the last years been funding, as well as founding, open source projects to ensure that their ideas and standards are established throughout the open software community[10].

I will not delve further into the principles and ideas of the open source movement. The interests of WCMS users lie in the TCO of the software and its content. It is important to remember that most open source material comes without guarantees and warranty unless you buy support from the vendor/developer, and this is where the cost of “free” software lies.

Proprietary software

Why does proprietary software use less open standards than open source software?

Because **they do not have to**. Having a closed set of developers invalidates the need to use common standards. While some of the developers most likely know several applicable open standards that can be used in the project. Including these might increase the learning curve for the other developers. An open source project does perhaps not have the collaborative luxuries of an office with a crew of geographically concentrated developers. The learning curve of joining in on the project must be overcome with explicit documentation, and specifications of the standards involved is a great place to start.

Proprietary software **is not necessarily shared[11]**.

The software is made, and put to use. Libraries (software components) of an open source project, on the other hand, might have to be used by other open source projects, and thus they need standards to enforce possible interaction between projects. Proprietary software seldom has to cross borders between companies, with the exception of larger software like operating systems and tool suites. Microsoft uses a multitude of proprietary standards to enable other vendors to produce software for the Windows platform, at a cost.

It takes away the software's edge. Using open standards gives the world a window into the code and its workings. This makes it possible for others to use or exploit functionality or the storage of the software directly instead of using the intended client software. An example is if Microsoft used an open document standard for storing Word-files, it would be easy for competing software vendors to develop Word-like tools, increasing the pressure on Microsoft's developers. Exemplifying a transport standard, WebDAV[12], enabling this protocol on a proprietary content management system would give other software access to the content, and as proprietary CMS-es go, this is not always the desired result. An open source project would only welcome such competition, as it would verify the worth of the project's effort, as well as increase the user base of the standard and further establish a community around it.

Obscurity means security. Or does it? As well as the previous paragraph reasons to guarantee that the software vendor keeps as much of the customer's money as possible, it also gives hackers a harder time getting into your system. The security holes that can not be seen can not be exploited. However, this is a false sense of security, but it has acquired mythical status having end-users believing that open source means insecure while it actually means well-tested and security hole-less software[13].

ActiveX is a Microsoft specification allowing powerful functionality in Internet Explorer that can be activated across the net, but this protocol has been heavily exploited by malware, viruses and worms to such a degree that many technical administrators have disabled this feature on

[10] http://www-128.ibm.com/developerworks/opensource/newto/#9

[11] http://heim.ifi.uio.no/~oleha/Publications/ib_ISR_3rd_resubm2.html

[12] http://www.webdav.org

[13] http://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/open-source-security.html

company computers[14].

Note that even though I use Microsoft and their Office products as examples of a proprietary software, they are not the big bad wolf regarding use of open standards. Such advanced software can not always suffice for the bureaucratic democracy and slow development of open standards. Microsoft is more and more embracing the use of open standards like WebDAV and SOAP[15] in their newest software, in fact the next version of the Office suite will use zipped XML-files for storage, like OpenOffice has been doing for many years[16].

Why does a WCMS need to use open standards?

Exposure. Web content must be available to as many visitors as possible, regardless of browser and operating system. There is much controversy regarding Google's newest service of video search, which is only possible on the Windows operating system.

Utilization. The content must be available for third party software and plug-ins. As a WCMS has an infinite set of requirements which no single software company can hope to satisfy by itself.

Portability. This goes for all kinds of server-side software. The point is that different customers rely on different operating systems for their servers, and to maximize the segment of the customer base, your software should be built on open standards to ensure platform independence.

Lock-in. Or rather the improbability of it. A customer's alarm should go off if the WCMS does not use standards for storage and transport. A WCMS quickly builds up a huge amount of content, and being locked to a single vendor could prove to be a gold mine, for the vendor.

Customization. If the customer wishes to alter the system and change functionality or looks of the website, can these changes be done by any qualified developer, or does the vendor have to be payed in full? With off-the-shelf software, perhaps no customization is offered or allowed at all.

Reusability. Both content and functionality should be reusable in new systems. The customer might have bought expensive plug-ins and built an excessive amount of well-structured content. Does the new WCMS store meta-data in the same way as the old one? Can the plug-ins function in the new environment, alternatively be ported over? There are open standards for both content storage and content/web services.

Case: e-Learning and their standards

One year ago, I was part of a team developing a combined student feedback and report management system for the University of Oslo. The purpose of the system was to enable students to provide feedback per the courses and programs they participated in, as well as giving the tutors and professors a platform for producing course- and program reports. We started off with a month describing the domain model for our database, how the university was built up with students, groups, courses, programs, institutes and faculties in a relational database. This information design was already existent in the University's IT-systems, but we figured that we would do it on our own, then later connect our system to the installed base.

This did not go very well. First the University's technical staff could not give us access to the existing relational database because of security and confidential reasons. There were no open interfaces for communicating with the existing system that we knew of, and the database was inaccessible. We were stuck and the project was canceled.

[14] http://vx.netlux.org/lib/aas05.html

[15] http://www.w3.org/TR/soap

[16] http://www.microsoft.com/whdc/device/print/metro.mspx

Some months later, I discovered that there are international standards for describing the relations in a educational system. These open standards are heavily in use in e-Learning tools and related software, and I learned that such a standard was even in use at the University of Oslo[17].

Why are open standards so emphasized in educational institutions? The same reasons I advocated for WCMS-es apply. Remember that universities and high schools all over the world are remarkably similar when it comes to organizational structure. Almost all of them are hierarchies with faculties, institutes, courses, teachers, students and grades.

These institutions are also connected to each other. Several countries have a central application office for handling the flow of students into all the institutions in the nation. Many students exchange between courses and programs, even across borders. There needs to be standardized information on how grades and courses from external institutions are handled in IT-systems, thus open standards for storage and transportation are implemented where possible[18].

Case: JSR-170, the standard for Java Content Repository

Early this spring, I was part of a new team developing a new version of a small Norwegian commercial WCMS. We decided to do a “big-bang” development, meaning that we would build it from scratch. After pondering on the design of storage for the system, I stumbled upon a specification done by the Java Community Process (JCP), a community tribunal deciding on Java Specification Requests (JSR) that are used throughout and about the Java programming platform, which we use for development in our company. The specification in question was JSR-170, Java Content Repository[19]. It had a lot of elegant solutions and ideas which fitted our project excellently.

At first we were tempted to implement a content repository ourselves, based on the JSR-170, but this soon estimated to become a huge amount of development, as the specification is quite large. Instead, we grabbed the one and so far only open source implementation, Apache Jackrabbit. We were of course curious of which other systems using JSR-170, and after some searching, we found an open source WCMS called Magnolia. We tried it out, and even though it did not fulfill all of our requirements, we were pretty impressed with it, and decided that instead of building our own WCMS, we would put our effort into learning Magnolia and participate in the open source project to improve the functionality where our requirements were unfulfilled.

Using a standard content repository makes it possible for our customers to change vendor and replace us as they see fit. This might be unfortunate for our part, but we feel that independence of vendor is a service we offer the customer to increase the probability of our solution being chosen.

There is a second advantage that benefits developers directly. As Jackrabbit is merely one implementation of the specification, we are free to change to another should it prove to be beneficial to our cause. As of today, there is a second open source implementation in the eXo portal platform[20], as well as a commercial enterprise implementation by Day[21]. Since the JSR-170 standard specifies that content-export/import must be implemented in a certain way, it is fully possible to transfer content from one implementation to the other.

[17] http://www.imsglobal.org

[18] http://www.cu.edu/policies/General/TCO.html

[19] http://jcp.org/en/jsr/detail?id=170

[20] http://today.java.net/pub/n/eXo1.1RC1

[21] http://jcr.day.com

Knowledge Management and open standards

Earlier on, I stated that I would approach the discussion with an element of knowledge management. A WCMS is a small part of the knowledge management process in any company, but it can be quickly become a larger part. Carefully still sticking to the subject, knowledge management IT-tools, I suggest that more collaborative tools be included in the WCMS. Some fitting examples are web-mail, forum, project management, wiki, knowledge base and customer relation management (CRM). Some WCMS products already include some of this functionality.

These collaborative tools might also, as often is the case, be separated into a **portal**. The definition has about a handful of meanings when it comes to websites, ranging from being a starting point on the web to something equivalent of a WCMS. Some experts of the industry view the (enterprise) web-portal as a fresh kick from the knowledge management camp in establishing a online knowledge platform. I prefer to think of a portal as the collaborative part of a website, the term groupware coming to mind.

One can also go the other way and induce web content management via a portal product. The WCMS and portal are sisters differed perhaps by the former tendency to offer workflow, while the latter is stronger on the collaborative part.

A collaborative nature only braces the reasons for choosing a product based on open standards. The Java Community Process has defined three Java Specification Requests (162, 167, 168) concerning portals and portal components (portlets), and these are implemented by a mass of both commercial and open source vendors.

The aforementioned Magnolia WCMS has in fact been integrated in existing portal solutions[22]. This shows that WCMS and portals can exist side by side.

Summary

More and more of the companies' information is being brought online as content to their websites. To ensure the value of this content, it must be managed with a worthy management system.

New technology makes us able to create powerful web applications that can bring the functionality of CMS and groupware online.

A WCMS is never finished. The requirements are indefinite, and more functionality will be demanded as time goes by. It needs to have customizable or extensible functionality. Open standards makes it easier to integrate plug-ins. An open source WCMS is freely modifiable.

Open source and open standards go hand in hand. Proprietary WCMS must follow recognized standards to ensure minimal TCO. Larger use of the standard further decreases TCO as the population standard-proficient developers increases.

Interaction with other systems is mandatory. Open standards assure possible communication, which is crucial to a WCMS.

The WCMS can become part of the knowledge management process in an organization by introducing more collaborative applications into the system.

[22] http://www.owasp.org/index.jsp