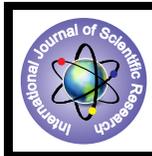


The role of Content Management For the betterment of Online Communities



Management

KEYWORDS : Online Community, Content Management Engine, Message Center, Host

Prof. Bhagyashri R. More

Assistant Professor, PDVVPF's IBMRD, Ahmednagar, Maharashtra, India.

ABSTRACT

The intention of this paper is to help the Websites by unfolding the significance of Content Management System for online communities. This paper shows the how a good CMS will work better for Online Communities users to include different features in the website using plugging and the widget. Content Management System is used tool to manage website content and depository for information. Today content management systems (CMS) help them to deliver targeted information for visitors interested in their products and services, there are many software packages available; however selecting a CMS is sometimes difficult because trend and evaluation data does not always coexist in the same reports. In this paper, I have tried to give an insight to the reader the relevance of content management systems and their abstraction using XML.

Introduction:

Communities are groups of people tied together by some common purpose or kinship. It is no different online. While lots of people talk about creating online communities, few do so from this sort of understanding. Generally, their concept is to be "the place" to go for some sort of information. They add a chat or a threaded discussion, collect user data, and call it a community. However, without the core of common purpose or some sort of kinship (in the widest sense of sharing some important aspect of life) these sites will never fulfill their goals. To succeed, an online community needs to fulfill its members' needs for affiliation and knowledge. Affiliation is the members' desire to belong to something. Knowledge is the members' desire to know something. The web system behind the online community needs to support affiliation and knowledge.

What are the components of an online community site?

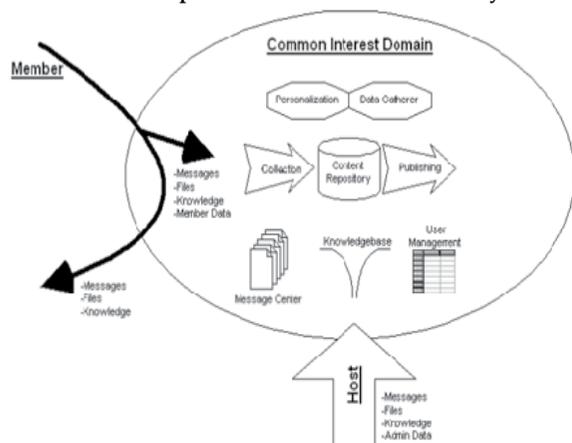


Fig 1.1 Major components of online community system

The Common Interest Domain

The common interest domain is the boundary around the community. It is the realm of content and interaction. It is the basis of a community. For all of the members, there is a reason why they would come together. Specifically, the domain is a statement of purpose for the community. The statement can be as general as "We love Barbie dolls" or as specific as "We are all female C++ developers working at atomic accelerators on software designed to track the trails of sub-nuclear particles." Whether specific or general, the statement must clearly define the entrance requirement of the community. It is absolutely the first thing that must be determined about the community and the rest of the structuring of the system should spring naturally from it.

The common interest domain defines what members will become affiliated to and on what subject they want knowledge.

The Personalization/Data Gathering System

Personalization, in general, (see Personalization and Content Management) is the process of collecting user data and using it to sub-select content to present to that user. This is true for the community site too. However, in addition to using member data to direct content to the member, on the community site member the data can be used to target other members to the member. This member match up provides for a much greater sense of affiliation. Members, in the end, want to be affiliated to other members -- not to a Web site. To perform this match, the system must collect member data that falls within the constraints of the common interest domain but narrows the focus so that members who share specific interest can be found. Of course, there is more to making friends than answering questions the same way. So the successful member matching system will need to be open and configurable enough to let in some subjectivity.

On the more mundane side of matching members to content, as in any personalization system, the site must have mechanisms for:

- Gathering member data
- Tagging content
- Mapping the type of data gathered to the appropriate tags in the content
- Dynamically rendering the selected content within a standardized page

The Content Management Engine and Knowledgebase

The site must have a viable engine for building the knowledge that members want to get to. As with any content management system, the site must be able to collect, reposit, and publish content. Specifically, in the context of a community site, the content management engine must:

Allow members to actively contribute to the knowledgebase of the site. Not only does this provide a wide base of knowledge flowing into the site, it also brings affiliation to its maximum depth. Members are most affiliated when they contribute as much as they receive from the community. Lots of member contributions are good only as long as they are pertinent well-structured content. Thus, it is particularly important in a community site to build a strong, simple metadata framework that naturally guides members to contribute relevant well-tagged information.

Have a repository with a fine level of granularity to support maximum personalization.

Be able to take feeds from the semi-structured sources that will come out of the message center.

Allow the repository to grow in a constrained way with content expiring when needed, missing or scanty information clearly identifiable, and new content areas able to be presented and pushed out to the members and host as needed.

The Message Center

The message center is the communication hub for the site. It includes any or all of the following technologies:

- Basic email
- Chat and hosted forums
- Threaded message boards
- Net meetings
- Net presentations
- Member location services
- Member classifieds and goods or services exchanges

The exact number and types of technologies used depends on the common interest domain, the computer savvy of the members, and their degree of affiliation. Generally, the more affiliation your community can muster, the more members will put the time and energy into these communication channels. The best sign of a low affiliation community is one where all of the bulletin boards are empty.

The system behind the community must obviously support these communication vehicles. In addition, it must harvest from them and successfully transition their semi-structured, real-time output into more enduring knowledge that can be delivered along with the rest of the content in the site's knowledgebase.

User Management System

Member data is essential to the system behind the community. In addition to being the basis for personalization, the system needs this data for a variety of other purposes such as:

- Member bulletins and global emails
- Member rights to particular content in the knowledgebase
- Member rights to the communication services in the message center
- Member rights to submit and modify content
- Administration of member fees or other initiation rites

For all of these purposes, the site needs a strong and extendable user data management system.

The Members

Members join the community for affiliation and knowledge. Again, when they are fully immersed in the community, they contribute as much of these goods as they receive. The purpose of the site and its underlying system is to facilitate the exchange of affiliation and knowledge among the members. Much more tangibly, members come to the site to:

- Find new content of interest
- Participate in a communication forum
- Find members to interact with
- Contribute content
- Get updates on content that they have previously stated is of interest

- See what is going on

All of these activities consist of uploading and downloading messages, files, content, and data. The goal of the site's system is to use these mundane upload and download actions in such a way that they create a sense of place and belonging in the members.

The Host

The community's host is the organization that is in charge of the site's infrastructure and maintenance. There are two typical hosts:

A commercial host has the members as a target market for goods or services. This host is willing to trade the cost of maintaining the site in return for exposure to the members. In a typical scenario, the host has the original idea for the community, creates an initial implementation of the site's system, fills the system with enough content to be viable, and then launches the site and opens it to members. The host continues to feed content in, administer user data, and create communication events. The major issue to resolve in this circumstance is what rights the host will need in order to use member data outside of the community. There is a delicate balance here between the members leaving because they feel exploited and the host leaving because they see a lot of cost and little return from the community.

A member host is one or more potential members who decide to create a web presence. Typically, there is some existing trade or interest organization with a current membership that organizes and funds the initial system. As with the commercial host, the member host creates an initial implementation of the site's system, fills the system with enough content to be viable, and then launches the site and opens it to members. The key issues here are continued funding of the site from often cash-strapped organizations and sufficient attention paid to the site maintenance by what is often volunteer run organizations.

For both the commercial host and the member host the primary issue is to make the site truly belong to the members. As time goes on, members should be the major contributors to the site, with the host having to supply less and less content. In a high affiliation community, members even plan and execute the communication events (chats, net meetings, etc.) If the community is successful, it is because the host has created the system that promotes affiliation and targeted knowledge gathering among a group of people who naturally gravitate to a clearly stated and well founded common interest.

Conclusion

Web content management has evolved far beyond the management of static html pages. Content management is more than presenting internal or external communications, more than publishing newsletters or event listings. Content management today is a complex set of processes, oftentimes involving a geographically distributed production team from diverse functional areas, multiple process steps, and exceptional amounts of information regarding publishing requirements and the targeting of content. In this paper, I have tried to give an insight to the reader the relevance of content management systems and their abstraction using XML.

REFERENCES:-

1. Maryam Alavi and Dorothy E. Leidner. Knowledge management systems: issues, challenges, and benefits. *Commun. AIS*, 1999.
2. OSGi Alliance. OSGi Service Platform - Core Service Specification Ver-

- sion 4.3, 2011. <http://www.osgi.org/Release4/HomePage> (July 27, 2011).
3. Tim Arnold-Moore, Michael Fuller, Alan Kent, Ron Sacks-Davis, and Neil Sharman. Architecture of a Content Management Server for XML Document Applications. In 1st International Conference on Web Information Systems Engineering (WISE 2000), 2000.
 4. Tim Berners-Lee. Semantic Web - XML2000, 2000. <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html> (July 27, 2011).
 5. JeenBroekstra, ArjohnKampman, and Frank van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In Proceedings of the first Int'l Semantic Web Conference (ISWC 2002), Lecture Notes in Computer Science, pages 54–68. Springer, 2002.
 6. Fabian Christ, Gregor Engels, Benjamin Nagel, Stefan Sauer, Sebastian Germesin, Enrico Daga, and OzgurKilic. IKS Alpha Development. Deliverable, 2010. <http://www.iks-project.eu/resources/iks-alpha-development> (July 27, 2011).
 7. Fabian Christ, Gregor Engels, Stefan Sauer, Gokce B. Laleci, ErdemAlipay, TuncayNamlı, Ali Anil Sinaci, and FulyaTuncer. Requirements Specification for the Horizontal Industrial Case. Deliverable, 2009.
 8. Harry Chen, Tim Finin, and Anupam Joshi. Semantic Web in a Pervasive ContextAware Architecture. IN ARTIFICIAL INTELLIGENCE IN MOBILE SYSTEM 2003 (AIMS 2003), IN CONJUNCTION WITH UBICOMP, pages 33–40, 2003.
 9. Michael Thomas Egner, Markus Lorch, and Edd Biddle. UIMA GRID: Distributed Large-scale Text Analysis. In Seventh IEEE International Symposium on Cluster Computing and the Grid, CCGRID 2007., pages 317–326, 2007.
 10. Roy Thomas Fielding. Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of California, 2000. <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm> (July 27, 2011).
 11. David Angelo Ferrucci and Adam Lally. Building an example application with the unstructured information management architecture. IBM Systems Journal, pages 455– 475, 2004.
 12. EJB 3.0 Expert Group. JSR 220: Enterprise JavaBeans,Version 3.0 Java Persistence API, 2006. <http://jcp.org/aboutjava/communityprocess/final/jsr220/index.html> (July 27, 2011).
 13. Ian Horrocks, BijanParsia, Peter Patel-Schneider, and James Hendler. Semantic Web Architecture: Stack or Two Towers? In Principles and Practice of Semantic Web Reasoning, pages 37–41. Springer, 2005.
 14. Berners Lee, J Hendler, and O Lassila. The Semantic Web. Scientific American, 2001.
 15. OMG. Unified Modeling Language (OMG UML), Superstructure, V2.3, 2010. <http://www.omg.org/spec/UML/2.3/Superstructure/PDF> (July 27, 2011).
 16. Hagen Overdick. The Resource-Oriented Architecture. IEEE Congress on Services, pages 340–347, 2007.
 17. SunitaSarawagi. Information Extraction. Foundations and Trends in Databases, pages 261–377, 2008.
 18. W3C. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax, 2009. <http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/> (July 27, 2011).