

Community Software Applications

T. B.Dinesh¹ and S. Uskudari²

1 Janastu, Bangalore, INDIA

2 Computer Engineering, Bogazici University
Istanbul, Turkey

Abstract. This is a case study of developing and using community knowledge management software. The context of the case study is a school. During a training workshop staff at a school was introduced to a community knowledge management system. The intent was for them to use it for lesson planning and discussions. Instead, they built a series of applications that they could use to make the school administration more efficient. These applications consisted of Admission, Library, Store and Personnel Management. This case study demonstrates that the ICT need for the next billion is to provide them with systems that they can customize and manage for their needs. An anticipated domino effect would be that they help customize applications for their neighbors' needs, thereby narrowing a digital divide caused by the mystification of application development.

1 Introduction

India has demonstrated how knowledge of English and technical-capacity has yielded a globally competitive IT force that has created a presence in everything from software development to business process outsourcing to research and product development. This is a good example of how capability building builds on itself.

The proliferation of computing and information technology (especially through Internet applications) has enabled users to envision useful applications that would serve their needs. Many such applications would be rather simple to create only if the know how were present. In another words, the avenues for creating what is envisioned by those who envision are lacking.

In this paper, we demonstrate how non programmers are able to create useful applications with a model that enables them to define domain specific information. The model, called Pantoto [1-2], in consideration enables knowledge based community applications that allow user to create domain specific content and communities. With this simple model, users are able to create a multitude of useful applications without resorting to any programming.

Please use the following format when citing this chapter:

Dinesh, T. B., Uskudarih, S., 2007, in IFIP International Federation for Information Processing, Volume 241, Home Informatics and Telematics: ICT for the Next Billion, eds. Venkatesh, A., Gonsalves, T., Monk, A., Buckner, K., (Boston: Springer), pp. 103-112.

We have developed a Web based application based on a knowledge community model called Pantoto. At Janastu [3] we use this model, where we work with a variety of communities and organizations to assist with community knowledge. The aim of this model is to enable communities to create and manage knowledge that is relevant to their context. The online communities that have been developed are essentially simple Web applications. Over the past few years some NGOs, small organizations and communities have used the system for a variety of purposes. Previous cases have been sociologists, disaster response teams, social workers, non governmental organizations, etc. The community SAATHI [4] is such a community application concerning the HIV infection in India.

2 Background and Related Work

Recent development in social network applications such as groups [5-7], blogs [8-9], wikis [10], folksonomies [11-12] have rendered individual publishing trivial. The impact of these technologies is manifested in collaborative inquiries and actions. The Web has enabled a platform that is widely distributed and connected. The social software that is built on it enjoys the benefit of these properties that ease collaborative efforts to be coordinated. The success of such systems is also an indication of how the community orientedness leads to further empowerment through a space that promotes learning and sharing.

There are various community applications that can be installed on websites, home machines, and servers. Caucus [7] is an application that allows groups of people to share and create new information. Information is categorized into segments that are called conferences. Users are part of conferences, can post and read content in each conferences (called Items and responses) in an asynchronous manner. Similarly, Google Groups [5] and Yahoo Groups [6] enable users to communicate regarding topics of interest to them. Some systems have a repository where documents can be uploaded. Wikis allow distributed content management. But in this case the content is the current snapshot without the historical content or separate contributions. Instead wikis favor group over individual ownership of content. There are numerous wiki systems with different features. A recent addition wikis, called Semantic Wiki [13] enables structuring the content with relationships, resulting in semantic processing of the content.

Most of these systems provide some customization capabilities enabling enthusiasts the ability to do their own website or a community site. However, these applications are limited in enabling a community to build a "software application" of sorts or allow members to build/customize such applications to their needs

3 Proposition

Many users are easily able to envision applications that are rather simple, which they are unable to construct due to the non trivial skills required. We propose that the next phase of application building needs those who are not software developers.

Furthermore, these users must be able to maintain the use and evolution of the system. Essentially we are proposing a model for an online knowledge community that is self defined and managed. This proposition is a technology enabling model, where end users are enabled to create and maintain an online space for sharing information and communicating within a domain specific context. Finally, information in this system must be structured – an essentially significant difference from free form online communities, such that the information lends itself to machine processing in order to render information useful in the long term. With access to such applications, these users become empowered as their role extends beyond being consumers to being producers of information technology.

4 Case Study: A K-12 Knowledge Community

This case study presents an account of a workshop held at CHIREC, a K12 school in Hyderabad, India. CHIREC is a progressive school with over 1000 students. Teachers develop structures and processes towards the delivery of effective class room experience for the students. CHIREC is a reasonably affluent private school that employs modern teaching methods and technology. They also have a computer department that supports teachers and students with course material and developing some small applications. Students are well versed with the Internet and most of them have a computer at home.

A few of the students are from outside Hyderabad and some from abroad. We approached this school to see if they would find it interesting to enable a process where the school knowledge - that of the student, teachers, staff inclusive - is pooled into a structured, interactive, and networked space. We studied some of their lesson planning processes and suggested approaches to creating a school knowledge base. A workshop was organized to introduce the teachers to the system where they could collaboratively develop teaching applications. Upon the realization that they would be building the applications, teachers instead voiced their preference for building applications that have been long overdue from the busy computer department. This was quite an unexpected surprise. The teachers decided to go ahead and build applications during the training workshop. This case indicated to us the need for platforms to build applications, and also as how this direction can enable a whole new generation of software builders who are in the right place within the communities for which they build the software.

The teachers had a need for various software applications: admission process management, library catalogues and checkout information, personnel office that caters to students/teachers and parents needs, web-site management, and such. The collective of all these applications, we learnt, is a good way towards enabling the community of teachers to manage their own knowledge. We can foresee that this experience will help them to continuously develop similar applications for their other evolving needs, including handling of lesson plans and such.

4.1 Pantoto Approach

We outline the Pantoto approach. Pantoto Communities software is Web-browser based. All configurations and definitions are done using forms. While some of the forms are related to administration of the software, most common need of a user is to use the forms to define other forms related to their domain of interest. Like a file system with files and access control, a user with an appropriate role can set access control to the folders of the file system. With in these folders, called categories, reside Pagelets -- which are structured, form filled information or rich content. The forms that are created by users are called templates. When a pagelet needs to be added to a category, relevant templates are used to help the user fill in these forms -- which becomes the content. If a form is about a base line survey, the contents of that category will be a database of these surveys. Pantoto handles the meta information management, of users, groups, collation of data and dissemination by searching or reporting. It provides editorship management, which helps the community to delegate and perform admin related tasks concerned to their domain or sub community.

Most of the school applications needed the following process of tasks:

1. Define Roles and access control sets.
2. Define a category tree of various information buckets.
3. Specify a set of templates/forms that need to be active in these buckets.
4. Copy and paste some action links in the text, to be used by application users.
5. Select or configure a presentation for different target groups.

4.2 Example

This section describes some of the things that the *Admissions Process Management* (APM) defined. Admissions personnel work on recruiting applications for the school. In this process, there are several groups with different roles that are in charge of processing the applications. With a simple application that allows them to document the applications according to date, class, name, etc, they are not only able to process the applications but also maintain historical records of the applications, which can be analyzed for reviewing and improving the application management process.

Roles

Application Issuer, Admission Officer, Admissions Incharge, Administrator

Categories

School > Admissions > Applications Sold
 School > Admissions > Applications Submitted
 School > Admissions > Applications Approved

Templates defined

Template Name	Fields
Application Issue	Name (person sold to), Application Number, Date Sold

Application Form	Student Name, Parents, Class applied for, Other details
Application Status	Status (Approved/Rejected/Pending), Remarks

Sample forms

Sample forms present in the system can be harvested in order to create new forms. This group found such an approach appealing and used sample forms and customized them. Forms are customized by setting some form properties such as field names and their types.

The content to be presented in the admin context is essentially a description for the category. In this case it is the description of the admissions application.

A look and feel is selected by the APM group. Finally, authorization rights are defined by selecting which roles have access to the categories.

4.3 User Perspective

Given the previous community definition, the system can be used as follows: Application Issuer issues an application (fills form). Admissions Officer receives application (fills rest of form). Admissions Incharge approves some (fills more details). All the information is collated according to templates in the categories. And default reports generated, and search enabled. Using the search, one can find a pagelet and further edit it (as defined by the above process).

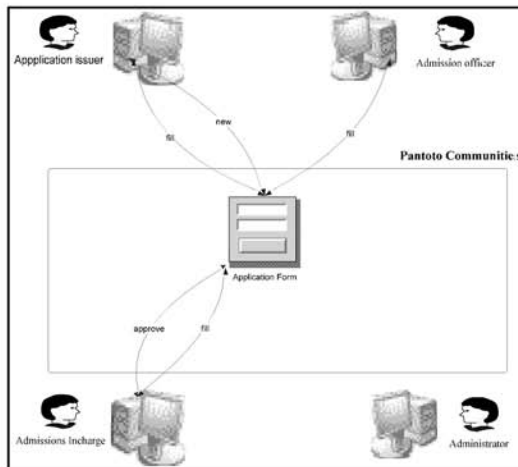


Figure 1: Administrative group's user perspective on creating and filling application forms.

Once application forms are created they lend themselves to any processing offered in the system. One of the automatically available services is context based search, where information can be searched according to criteria.

The context dependent search expression below is a request to find all applications for class 10 that are in pending status. Results of the search result in a

report consisting of the application number, the date, that name of the person it is sold to, the parent and the class of application (all fields that have a check mark). The status is not shown as it was not requested. The report may be viewed as a Web page or be exported as a spreadsheet.

No		<input checked="" type="checkbox"/>
Date		<input checked="" type="checkbox"/>
Name		<input checked="" type="checkbox"/>
Parent		<input checked="" type="checkbox"/>
Class	10	<input checked="" type="checkbox"/>
Status	Pending	
Search		

The following are some screen shots from the application. Due to the private nature of the information most of the application can not be shown, but the following should be sufficiently indicative to the nature of applications built.



Figure 2: Main Page that gathers access to various applications at the school.

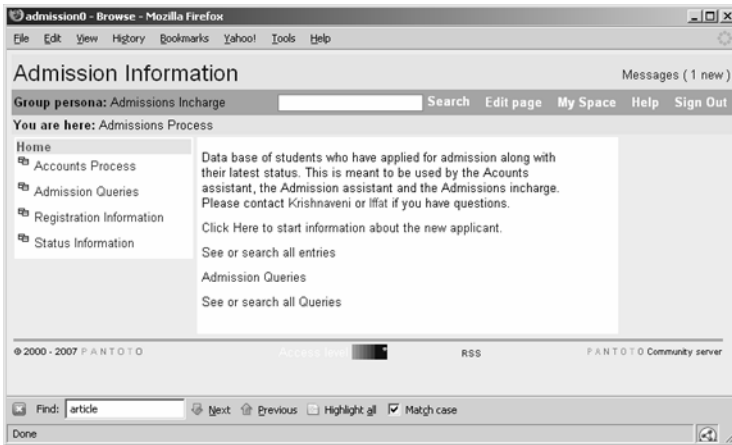


Figure 3: Main page of Admissions Information application

Title	Date of Registration	Admission Status	Letter Sent On	Fees Due on	Final Status	Class Allotted	Serial Number	Academic year	Application Number
	27-12-2006	Confirmed	04-02-2007	08-02-2007	Vacancy Filled	Nursery		2007-2008	SS 708008
	27-12-2006	Confirmed	25-01-2007	02-02-2007	Vacancy Filled	Nursery		2007-2008	SS 708016
	27-12-2006	Confirmed	25-01-2007	02-02-2007	Vacancy Filled	Nursery		2007-2008	SS 708124
	27-12-2006	Confirmed	04-02-2007	08-02-2007	Vacancy Filled	Nursery		2007-2008	SS 708125
	27-12-2006	Confirmed	25-01-2007	01-02-2007	Vacancy Filled	Nursery		2007-2008	SS 708251
	29-12-2006	Confirmed	25-12-2006	02-02-2007	Vacancy Filled	Nursery		2007-2008	SS 708081
	02-01-2007	Confirmed	25-01-2007	01-02-2007	Vacancy Filled	Nursery		2007-2008	SS 708255
	29-12-2006	Confirmed	25-01-2007	03-02-2007	Vacancy Filled	PP I		2007-2008	SS 708009

Figure 4: Report of applications resulting from a context based search. Titles and serial numbers are obscured to protect privacy.

5. Implementation

Pantoto Communities has been developed as Open Source Software (OSS) in order to facilitate free access. It is platform independent using JAVA [14-15], Apache Tomcat Web Server [16], MySQL [17] for database, and Lucene [18] as a search engine. We typically install it on machines with Linux operating system, but there are also several Microsoft Windows installations. Client access is through browsers. It has been tested on various forms of Internet Explorer, Firefox, as well as various other Web browsers on Linux and Apple platforms.

6. Installation and Requirements

Pantoto Communities is a Web application and thus must be hosted on a Web server with Tomcat support. The server may be locally owned with in an office network or provided by Internet Service Provider The space requirements depend on the community that will use it. Pantoto Communities and all the dependent software will need less than 100Mb space. Our experiences have mostly involved rather limited space usage with predominantly textual content. For a community that would want to share large video or audio content, the space requirements would be much greater. In our experience, many communities have preferred a local server, which can easily be supplied with a standard PC. For communities with more intense use, a server computer would be advisable with about 1GB of RAM. Pantoto is open source, with APL/GPL licenses.

7. Review

It was impressive to observe school teachers get excited and consider their software needs. Each group built an application in just a few hours. After a server was setup they were able to build, maintain, use and share the software with other relevant parties. In spite of the presence of an in house IT support team at the school, it was evident that this system offered empowerment of teachers and staff. People were able to quickly build applications without the need of the computer professionals. The wait time related to requests as well the appropriate customization seriously hinders such persons from taking part in the development of such applications. The IT department also gets liberated from minor customizations that they may perform incorrectly due to their lack of domain knowledge. More importantly, the bottleneck position of such IT staff is a source of frustration for all parties. By removing them from the cycle when appropriate a more efficient and productive system is achieved.

Some of the other applications that are done by non-software development people, in this manner were:

- HR Job application created by the HR department, where jobs are announced, applications received, short-listed and applicants selected.

- An NGO that works with disabled people built a patient management system for their every day needs.
- An NGO developed a baseline survey and processing application.
- A crafting community built various web-sites that are cross linked to each other.
- An organization has used it for almost all of their internal information management needs, where several small applications interface define how the information is collected the related work flow.

8. Conclusions

Our experiences demonstrate that by using such a system like Pantoto Communities, people who are not computer savvy are able to create very useful Web applications. For many cases Web applications are not very sophisticated. They involve contributing content, searching, and reporting. For community or team applications they also involve communication. By providing means for developing applications based on these patterns, we enable the building of a significant number of custom applications. In an age where fast information access is expected while a great portion of the global population is left on the dark side of the digital divide it is essential to deliver enabling technologies. We are encouraged through our experiences, where we encounter case after case that non programmers are easily able to create applications such as demonstrated in the case described in this paper (application handling team within a school).

We continue to learn from each case and strive to improve Pantoto Communities, which we develop using technologies that enable our project. We maintain a low cost development process and are engaged with the communities who are the target users of this system. Another interesting observation is that that in-spite of the rush to join large companies and to earn large salaries, there are those who are interested in developing community oriented low-cost social applications.

We foresee that our next steps would be to enable simple sms based mobile applications, integration with other services such as GIS, synchronization to offline servers and such. Most significantly, to improve the usability of Pantoto Communities software based on the experiences of the teachers and staff at CHIREC.

Acknowledgements

Pantoto Communities is the result of a collaborative effort. Its implementation and its utilization have been impacted by numerous sources. We gratefully thank all those who have generously contributed to this effort -- volunteers who wrote code, those who contributed free servers space and offered maintenance, and those who provided encouragement.

References

1. Pantoto Communities. url=<http://pantoto.com>, (extracted March 2007)
2. S. Uskudarli and T. Dinesh. Pantoto: A participatory model for community information. In *Proceedings DyD'02: Development by Design 02*, (2002).
3. Janastu – Let it Be People. url=<http://janastu.org>, (extracted March 2007)
4. SAATHII, Solidarity and action against the HIV infection in India - online resource center. <http://saathii.org> (extracted March 2007).
5. Google. Google Groups. url=<http://groups.google.com>.(extracted March 2007).
6. Yahoo! Yahoo! Groups. url= <http://groups.yahoo.com>.(extracted March 2007).
7. Dutches BOGES (Board of Cooperative Educational Services), Caucus: Online Conferencing, url= <http://www.dcboces.org/conference> (extracted March 2007).
8. Pyra Labs. Blogger. <http://www.blogger.com> (extracted March 2007).N. P. Group. Connotea - organize. share. discover. <http://www.connotea.org>, (extracted 2007).
9. B. A. Nardi, D. J. Schiano, M. Gumbrecht, and L. Swartz. Why we blog. *Commun. ACM*, 47(12) (2004) pp 41–46.
10. W. Cunningham and B. Leuf. *The Wiki Way. Quick Collaboration on the Web*. (Reading, Mass. Addison-Wesley , 2001)
11. C. Marlow, M. Naaman, D. Boyd, and M. Davis. Ht06, Tagging Taper, Taxonomy, Flickr, Academic Article, ToRead. In *HYPertext '06: Proceedings of the seventeenth conference on Hypertext and hypermedia* (New York, NY, USA, ACM Press , 2006.), pp 31-40.
12. del.icio.us. del.icio.us, social bookmarking. url=<http://del.icio.us> (extracted March.2007).
13. Max Völkel, Markus Krötzsch, Denny Vrandečić, Heiko Haller, and Rudi Studer. Semantic Wikipedia, In *Proceedings of the 15th international conference on World Wide Web, WWW 2006*. (May 2006)
14. D. Flanagan. *Java in a Nutshell, Fourth Edition*. 4th. (O'Reilly & Associates, Inc, 2002)
15. Sun Microsystems. The Source for Java Developers. Sun Developer Network. url=<http://java.sun.com> (extracted March 2007).
16. The Apache Software Foundation. Apache Tomcat. url=<http://tomcat.apache.org> (extracted March 2007)
17. MySQL. Mysql 5.0 reference manual. url=<http://dev.mysql.com/doc/refman/5.0/en/index.html>, (Extracted March 2007).
18. Erik Hatcher. Lucene Intro. Java.net – The Source for Java Technology Collaboration. url=<http://today.java.net/pub/a/today/2003/07/30/LuceneIntro.html>, (2004).