Building an Enterprise Information Portal Using Open Source Software

Mohsen Kahani

Faculty of Engineering, Ferdowsi University of Mashhad, Iran
kahani@um.ac.ir
http://www.um.ac.ir/~kahani/

Abstract.

Information portals are considered as a gateway to internal and external information within enterprises. There are many commercial software for building such gateways, which are usually expensive. Open source software is being considered seriously as an alternative to commercial software by many organizations. This is partly because of uncertainty and unreliability of commercial systems and software exploited by recent computer viruses, and significant saving on total cost of ownership. In this article, we present our experience in building an open source based information portal for Ferdowsi University of Mashhad and discuss some of the issues encountered during the development process.

1. Introduction

Internet has had dramatic effects on different sectors of technology, particularly information technology. The rapid growth of demands for access to the Internet has urged organization to build high speed Intranet to provide an enterprise level access to the information. The use of this strong backbone to connect internal information systems has changed the information model towards a Web-centric architecture.

Organizations usually have a lot of internal and external information accessible via a range of applications. Each set of information can be accessed by a specific group within the organization. For example, customers usually have access to public information, while staff and managers can access both public and private information depending on their position within the organization.

With the proliferation of Web and dominance of Web-based application, use of the organization's home page as the access point for enterprise-level application is now quite common. These gateways to the information are called portals.

Although there are many definition for Enterprise Information Portal (EIP) in the literature [1], we define an EIP as "a secure, Web-based interface that provides a single point of integration for and access to information, applications and services for all people involved in the enterprise, including employees, partners, suppliers and customers" [2].

In this paper, we discuss the implementation of a portal to access these mostly dispersed systems in an integrated environment. However, we use open source software for our development.

After brief description of basic concepts, we explain the architecture of the system. Then, the open source components that have been used for implementation are discussed. Finally, our experience and some of the issues that we encountered are mentioned.

2. Open Source Software (OSS)

Although free software was available on the Internet from the early days, it has recently attracted a lot of attention as an alternative to commercial software. However, open source software does not only mean free software, or access to the source code. An OSS must comply with the following criteria [3]:

1. No restriction on giving away the software as a component of an aggregate software distribution.
2. The distribution must include the source code.
3. The license must allow modifications and derived works and distribution of them.
4. The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code
5. No discrimination against persons or groups is allowed.
6. There should not be any restriction for using the program in a specific field of endeavor.
7. The program rights must apply to all to whom the program is redistributed, automatically.

8. The license must not depend on the program's being part of a particular software distribution.
9. The license must not place restrictions on other software that is distributed along with the licensed software.

The proponents of OSS mention many reasons for using OSS as an alternative to commercial and well-supported software. The discussions include both non-quantitative and quantitative measures. In [4] quantitative comparisons between major open source solutions and their commercial counterparts have been reported. The report discusses market share, reliability, performance, scalability, security, and total cost of ownership (TCO) issues.

The report compares Windows (all flavors), Linux and Solaris as operating systems, and Apache [9] and IIS as Web-servers. It clearly shows that OSS is far ahead of commercial counterpart in reliability, performance, scalability, security, and TCO issues. In term of market share, OSS is moving neck-to-neck, but the trend seems to benefit OSS more.

In another survey, several DBMSs have been compared against several benchmarks, including performance, reliability, speed and ease of use [5]. The survey concludes that MySQL [7], the open source DBMS, has nearly the same performance as Oracle, and is as easy as MS-SQL to install and use.

Besides these quantitative comparisons many issues have raised the position of OSS. The recent attacks of malicious computer viruses, such as Code Red and Nimda, have exposed many potential security hazards of Windows platforms. Talking with other IT director in person or in newsgroups and list-servers indicated that most of them have great concerns over the security issues, and are considering switching to more reliable platforms.

3. Development Process

Before starting the development of EIP, we had a complete requirement analysis and system analysis using structured methodologies. This study took nearly a year to complete. Upon completion of the project, we had a good understanding of the system and the type of information and reports that are required.

The problem that we encountered was that most sections either did not have a proper computerized system or their system could not be integrated into the Web-based system. So, we not only had to build the portal, but to restructure and sometimes rebuild the applications as well.

It was decided to use an incremental development process, because the amount of work was too much, and the requirements were subject to change. Figure 1 shows an incremental development process stages.

Based on this model, the architecture of the system (as described in the next section) was designed, and the initial order in which system was to be built was defined. As the university did not have a Persian home page and most system had to be built in Persian language, the first stage included design and construction of the home page. The English home page, to provide information for foreign users, was planned to be implemented later (This phase is currently under construction).

4. System Design and Implementation

Designing the architecture of the system is the most critical task. The architecture should be extensible and scalable so that it can contain all parts of the system. This is more difficult in an incremental development process, as not all parts of system have been clearly defined.

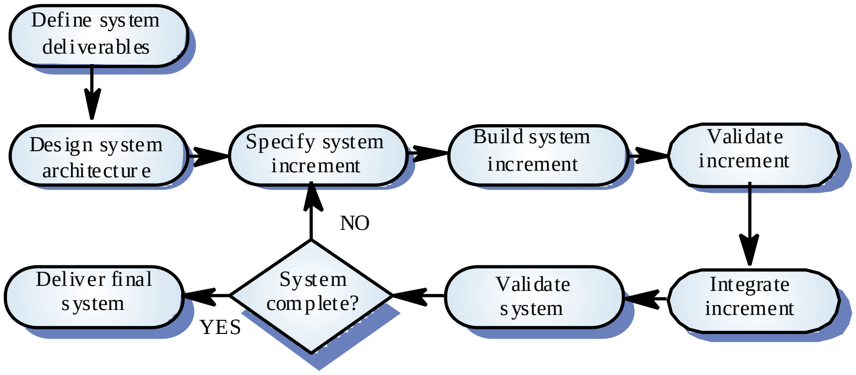


Fig. 1. Incremental development process

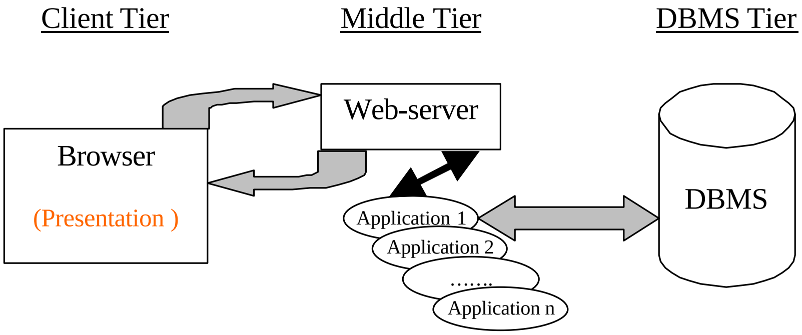


Fig. 2. The architecture of a Web-based system.

This task might seem too difficult to achieve in a traditional two-tier architecture. However, Web-based systems use an *n*-tier client/server architecture, as shown in Figure 2. This approach allows for maximum modularity as each system can be defined as one application in the middle-tier.

Based on this architecture, the design and implementation of the system started. Because of the reasons mentioned in the previous sections, it was decided to use open source software. So, comprehensive evaluations of the available solutions for each port of the system were achieved. Eventually, the following software was selected and the system was implemented. The view of the main page is shown in Figure 3.

Operating System. Operating system is the most important software in a computer. The strength and the weakness of OS are the merit and pitfall of the system. Linux for years has been proven to be a reliable and strong OS particularly for server computers. We use Linux RedHat 7.2 [6] as the operating system for our Web, SMTP, FTP and database servers. That was partly because the staff were quite familiar with it and also that RedHat Linux has a strong position in the Internet (both in terms of market share and technical support).

RedHat 7.2 has built-in support for firewalling (through *ipchain*), SMTP (Simple Main Transfer Protocol), DNS (Domain Name Server), several OSS DBMSs, HTTP servers, SAMBA (Windows file sharing), NFS (Network File Service), NIS (Network Information Service), several scripting languages and many other useful utilities. It also has two Graphical User Interfaces (GUI), Gnome and KDE, and allows graphical configuration of services.

HTTP-SERVER. Apache [9] is the most widely used web-server in the Internet [4]. The server is available for both Linux and Windows platforms. It provides support for many scripting languages through *modules*. Modules are dynamically loaded into the server as needed and provide a more robust and efficient solution than CGI approach.

Fig. 3. Ferdowsi University of Mashhad main page

DBMS. We opted for MySQL [7] as our DBMS. MySQL is the most popular Open Source SQL-based relational database. It has two types: MySQL and MySQL MAX. MySQL MAX support transaction and is used for more critical systems, but is a little slower than MySQL. Both types are available for a variety of platforms, including Linux and Windows. The latest release (V.4) provides most features that expensive commercial DBMSs, such as Oracle, provide, including nested SQL and replication. Stored procedure has been planned for V4.1 release.

Scripting Language. Initially Java Server Pages (JSP) was used as the scripting language. As there is no direct JDBC driver available for many DBMSs, we had to connect to our DBMS throu a JDBC-ODBC driver. This caused some problem for four Persian letters: *pe*, *che*, *gaf* and *zhe*. Besides, the configuration of the JSP server was rather difficult and platform-dependent. So we switched to PHP (Hypertext Pre-Processor)[8].

PHP is an HTML-embedded scripting language, which recently has attracted a lot of attention. Its syntax is similar to C and support object-orientation, as well. There are PHP interpreters available for major platforms, and Apache provides a *php-module* for better handling of PHP requests. Because of its popularity, currently, there are many useful utilities, libraries, database connection routines and plugins available for PHP, which makes the development process fast and easy.

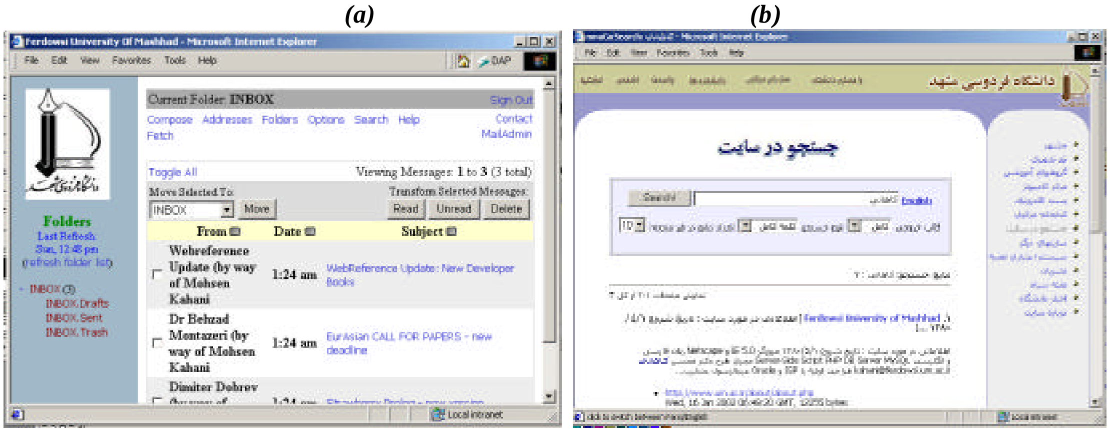
WebMail. Traditionally, people access their emails via terminal (telnet-based) or POP3 mail readers such as Eudora and MS-Outlook Express. Although POP3 email readers provide good GUIs, they have two main drawbacks. First, when emails are read via POP3, the emails are transferred to the local machine, so if the mail server is accessed via another computer, there is no access to the previous read messages. (Though it is possible to keep messages on the server, it causes some other issues, as well). Second, because of security threads, POP3 access from the outside of the intranet is usually prohibited by the corporate firewall. Therefore, Web access gateway to emails is now quite common.

Although, there are a lot of OSS Webmail programs available, some measures should be taken to select a proper one. The Webmail should access emails via IMAP protocol, as people switch between Web and POP3 accesses. Access to other mailboxes should be possible, too.

SquirrelMail [10] was chosen as the Webmail program. It is one of the most active OSS projects, has all of the above characteristics. Its architecture is based on PHP, uses MySQL to store users' profiles, and can be easily expanded via plugins. Currently over 30 plugins are available, which add significant functionalities to the system. A snapshot of the Webmail system is shown in Figure 4-a.

Search Engine. Finding information on the Web without help of search engines is nearly impossible. This is true for a corporate information system, too. There are a few OSS search engines available. We found that mnoGoSearch [10] (formerly udmSearch) can be easily fit into out architecture. Its backend is based on C languages, which reads URLs, indexes them, and stores indexes in a database. It support several DMBS, including MySQL. There are two kinds of front-ends, one C-based and one PHP-based. The output format can be easily redefined and it support non-Latin character maps. A view of the search engine is shown in Figure 4-b.

Figure 5 illustrates how different parts of the system have been assembled together. The web-server acts as a portal, connecting different parts together. Also note that, although it is depicted separately, all services can run on only a single computer.



(a)

Ferdowsi University Of Mashhad - Microsoft Internet Explorer

Current Folder: INBOX

Viewing Messages: 1 to 3 (3 total)

Folders
Last Refresh
Sat, 12:48 pm
(refresh folder list)

- INBOX (3)
  - INBOX.Drafts
  - INBOX.Sent
  - INBOX.Trash

	From	Date	Subject
□	Webreference Update (by way of Mohsen Kahani	1:24 am	WebReference Update: New Developer Books
□	Dr Behzad Montazeri (by way of Mohsen Kahani	1:24 am	EurAsian CALL FOR PAPERS - new deadline
□	Dimiter Dobrev		

File Edit View Favorites Tools Help
DAP
Sign Out
Compose Addresses Folders Options Search Help
Contact
MailAdmin
Fetch
Toggle All
Move Selected To:
INBOX
Move
Transform Selected Messages:
Read Unread Delete
Local intranet

(b)

دانشگاه فردوسی مشهد

جستجو در سایت

Wed, 16 Jan 2002 05:49:20 GMT, 12255 bytes

File Edit View Favorites Tools Help
Search!
English
Ferdowsi University of Mashhad
Local intranet

Fig. 4. Some screen shots of the system: a) WebMail b)Search engine

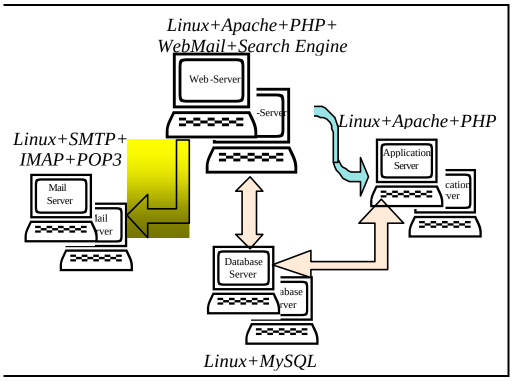


Fig. 5. The integrated system

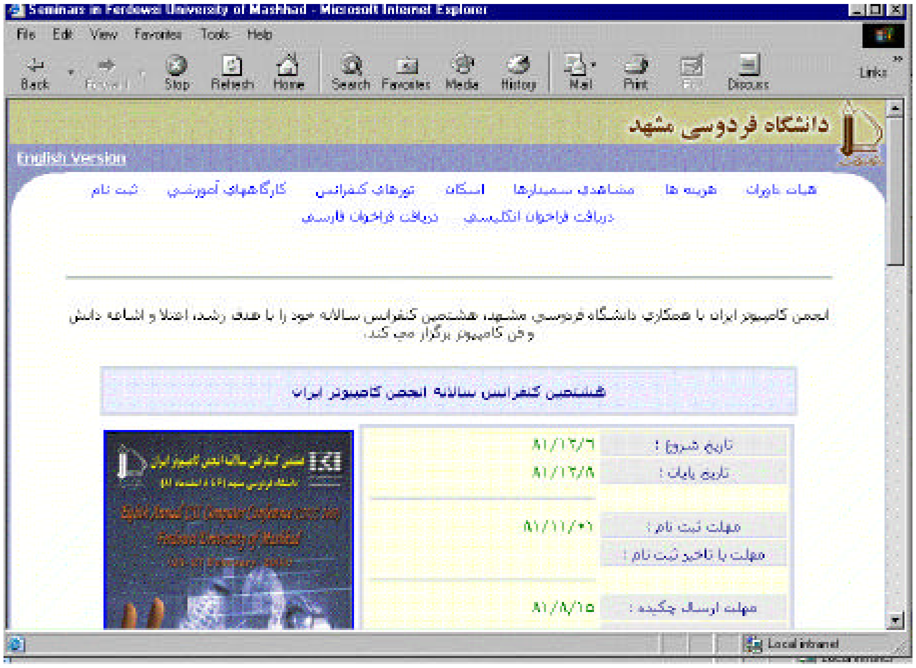
Besides these general systems, many modules have been incorporated into the system. A telephone directory system, conference management system, academic staff resume are among the main important modules. An snapshot of the conference management system which has been used for this conference is shown in Figure 6.

The implemented system not only acts as a gateway to information organized in the main site, but also provides access to several Web-based system available on the intranet. Currently, it provides links to central library systems, nutrition system, student record keeping system and web-based learning system.

5. Important Issues

During the development process, we have experienced several issues, which worth mentioning. These issues are discussed here.

- We used a structured methodology and incremental life cycle model for system development. However, a more effective methodology is to use object-oriented techniques. Use of Unified Modeling Language (UML) [11] and Rational Unified Process (RUP) [12] provide more modularity and better visually modeling. As, iterations and increments are strongly supported by RUP, making it a good candidate for Web-based information systems. The main drawback is that, it is difficult to find engineers familiar with the technology.
- Although, developing big information system is inherently difficult to manage. Using incremental methods adds to the complexity. The management becomes more difficult if some parts of the system are outsourced (which is usually the case). The manager should have complete knowledge of both technology and problem domain.



دانشگاه فردوسی مشهد

هیات داوران　هزینه ها　مشاهده سمینارها　اسکان　تورهای کنفرانس　کارگاههای آموزشی　ثبت نام
دریافت فراخوان انگلیسی　دریافت فراخوان فارسی

انجمن کامپیوتر ایران با همکاری دانشگاه فردوسی مشهد، هشتمین کنفرانس سالانه خود را با هدف رشد، اعتلا و اشاعه دانش و فن کامپیوتر برگزار می کند.

هشتمین کنفرانس سالانه انجمن کامپیوتر ایران

تاریخ شروع :	۸۱/۱۲/۶
تاریخ پایان :	۸۱/۱۲/۸
مهلت ثبت نام :	۸۱/۱۱/۰۱
مهلت با تاخیر ثبت نام :	
مهلت ارسال چکیده :	۸۱/۸/۱۵

Fig. 5. The conference management module

- There are too many alternatives available for each component of the system in the open source community. For example, the SourceForge [10] site contains about 50,000 registered projects. Therefore, choosing the right software is rather difficult. It is quite common to select a component, but later have to change it. This is because each component has been developed by different groups of developers, and a total solution is not explicitly evident.
- Choosing open source software means that it might be difficult a single company to support the system. So, the maintenance and support of the system should be carefully planned. Training the technical staff periodically is very important, otherwise because of the rapid change in the technology, after a while, the support of the system would become virtually impossible.
- Platform independency is an important factor to consider, too. Some OSS are platform-dependent (mostly Linux). Even if there is no plan to transfer to another platform, platform independency helps during the development process. Developers are usually more familiar with Windows platform. Therefore, if it is possible to develop everything on Windows and easily transfer them to the main platform would makes the development process more efficient. For example, in our case, the system was so platform-dependent that switching between platforms is as easy as copying files into another directory.

6. Conclusion and Future Works

In this paper, we discussed the possibility of using open source software for building an information portal for our university. After explaining the architecture of the system, the OSS that we used for each part of the system was discussed in some details. Our experience for implementation of such a system was briefly mentioned. The system is currently installed and utilized at Ferdowsi University of Mashhad.

As mentioned in the paper, the portal provides gateway to some enterprise-level applications. However, these systems have not been integrated very well. For instance, each user has different user-ids and passwords to access different applications. This causes some problems and inconsistencies, especially when user-ids are issued by different units. It is planned to have a single sign-on screen for authentication, and allow access based on users' privileges. Moreover, the integration of more systems, especially legacy ones are also planned.

The system uses a template-based style for storing the static content. Modern content management systems, such as Postnuke [13], use a totally database-driven approach to store contents. This allows modification of the content from any location and provide much more flexibility. Addition of these featues are planned for the next phases of the project.

Acknowledgement

I would like to thank my colleagues at University Computer, Statistics and Information Center, especially Mrs. Marjani and Ms. Ghahremani, for helping me build and maintain the discussed system.

References

1. Firestone, J., Defining the Enterprise Information Portals. available at URL http://www.dkms.com/EIPDEF.html.
2. Fitzgerald, P., Tie it together and open it up with Enterprise Information Portals. Intranet Design Magazine, http://idm.internet.com/articles/200104/pt_04_25_01a.html.
3. The Open Source Definition : version 1.9., available at URL http://www.opensource.org/docs/definition_plain.html.
4. Wheeler, D., Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!. Available on URL http://www.dwheeler.com/oss_fs_why.html.
5. Dyck, T., Server Databases Clash, eWeek Magazine, February 25, 2002, available also online at http://www.eweek.com/print_article/0,3668,a=23115,00.asp
6. Linux RedHat can be found at http://www.redhat.com/
7. MySQL can be found at http://www.mysql.com/
8. PHP can be found at http://www.php.net/
9. Apache can be found at http://www.apache.com/
10. SourceForge site (Home to many OSS), http://www.sourceforge.net/
11. OMG Unified Modeling Language Specification, Version 1.3, June 1999.
12. Rational Unified Process, http://www.rational.com/
13. PostNuke CMS, http://www.postnuke.org/

Disclaimer: All mentioned URL were available on October 2002.