Chapter XVII
Open Source Web Portals

Vanessa P. Braganholo
DCC/UFRJ, Brazil

Bernardo Miranda
COPPE/UFRJ, Brazil

Marta Mattoso
COPPE/UFRJ, Brazil

ABSTRACT

Open source software is required to be widely available to the user community. To help developers fulfill this requirement, Web portals provide a way to make open source projects public so that the user community has access to their source code, can contribute to their development, and can interact with the developer team. However, choosing a Web portal is not an easy task. There are several options available, each of them offering a set of tools and features to its users. The goal of this chapter is to analyze a set of existing Web portals (SourceForge.net, Apache, Tigris, ObjectWeb, and Savannah) in the hopes that this will help users to choose a hosting site for their projects.

INTRODUCTION

One of the main sustaining pillars of the open source (Perens, 1997) philosophy is that software must be widely available to the user community. In order to mature, open source projects need collaboration from the user community, and this is hard to achieve just by publishing a project on a developer's personal home page. An efficient way of reaching these requirements of availability and collaboration is by hosting the software on an open source Web portal. There are several portals that address these requirements, offering free hosting to open source projects.

Besides giving access to a project's source code, these portals also offer tools to help the development of the projects they host. Among such tools, we can cite task management tools, issue trackers, forums, mailing lists, tools to support feature requests, and version control servers.

The different portals offer different advantages to the projects they host. It is difficult for a developer who is not used to contributing to open source projects to choose the one that best

fits his or her needs. This is because there are many portal features that are only visible to those who actively contribute to an open source project. Additionally, a portal may have particular requirements that the developer must be aware of. For example, some portals require that the project be under the protection of a specific open source license. The goal of this chapter is to help such users in choosing a portal to host their projects. We analyze five Web portals and compare them in terms of the services they offer. The analyzed portals are as follows:

- SourceForge.Net (Open Source Technology Group, 2005),
- Apache (Apache Software Foundation, 1999)
- Tigris (Tigris, 2005)
- ObjectWeb (Object Web Consortium, 2005)
- Savannah (Free Software Foundation, 2000b)

They were chosen for several reasons. First, they host projects for free. Second, they are general in the sense that they host general free or open source software (Savannah hosts even nonsoftware projects). Third, they have been online for enough time for one to assume that they probably will not disappear and leave users helpless.

It is important to emphasize that this kind of analysis is new in literature. To the best of our knowledge, there is no work in the literature that provides similar analysis (DiBona, Stone, & Cooper, 2005).

It is also important to state that some of the portals may be focused on free software (Free Software Foundation, 1996) while others focus on open source software (Perens, 1997). Although their way of looking at the world is different (Stallman, 2002), the philosophies are similar. In this chapter, we do not intend to make any distinction between them. Thus, we use the term FOSS (free and open source software) as synonymous of free software and open source software.

The subsequent section describes briefly the way most Web portals work. Then we discuss the methodology of our study and the features of each portal. Next we discuss future trends and conclude with a tabular comparison of the Web portals.

BACKGROUND: HOSTING SOFTWARE ON WEB PORTALS

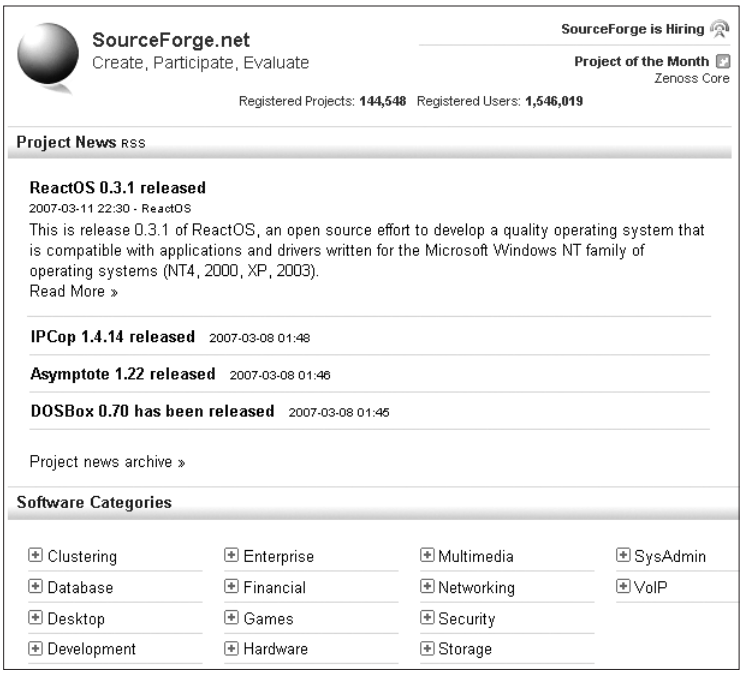
In this section, we describe how portals work in essence, hoping this will give readers a better understanding of our proposal in this chapter.

Web portals dedicated to hosting software projects are basically Web pages that offer a set of functionalities to its users. Usually, the entrance page explains the purpose of the portal and provides links to documentation, instructions to users who want to host a project, a news section, and links to the hosted projects. Such links are usually presented within categories. Figure 1 shows a cut of the main page of the SourceForge.Net portal. Notice the news section and the links to software categories (at the bottom of the figure). Such categories link to hosted projects classified under them.

Each hosted project has its own page within the portal with a URL (uniform resource locator) similar to http://www.portal.org/project, where *portal* is the portal name, and *project* is the project name. It is through this page that the portal provides tools and services to developers. Also, such pages play the role of advertising the project. Users will find projects they may be interested in through such pages.

The main page of a project within any portal has basic information about the project, news, and a list of links to source-code downloads and mailing lists, among other features. As we will discuss later on, it is a choice of the project's administrator what will appear on the main page.

Figure 1. A cut of the main page of SourceForge.Net



Another important aspect of hosted project pages is that they have public and private areas. Public areas can be seen by any person. The public area is basically the project's main page. This way, any user may read about a specific project and download its source code. The private area of a project is exclusively for the project's contributors. The list of people that may contribute to it is maintained by the project administrators. In order to contribute to a project or even to create one, you must have a user account with the portal. All of the portals we analyzed allow you to create a user account for free. Once logged in, you are in your private area. In this area, you can create a new project, or see all the projects you contribute to.

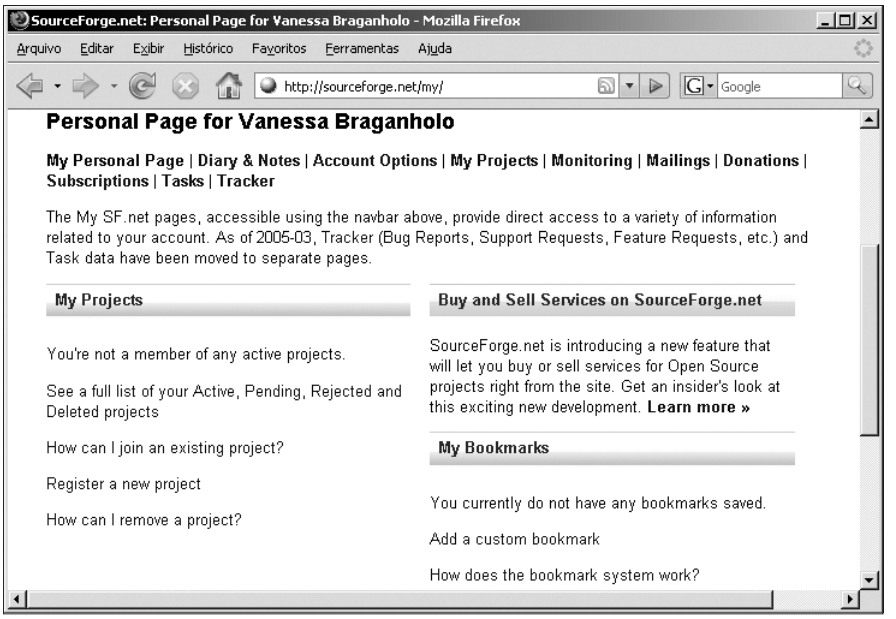
If you are not part of a project, then your private area may include several links, including one to create a new project. Figure 2 shows a private area at SourceForge.Net. Notice the "Register a new project" link on the left-hand side.

If you have access to the private area of a given project, you can solve bugs, write documentation, check in modifications to the source code, and so forth. In other words, the private area gives you ways to contribute to software projects.

Project administrators may include you as a developer, as a documentation writer, as a project administrator, and so forth. Each role you assume grants you access to certain parts of the project's private area. The project administrator has full power in choosing what you may or may not edit or develop in his or her project.

An important point that needs to be made here is on how a user can contact the project administrator and ask to be included in the project. This is usually made through the help-wanted section. Administrators can explicitly ask for the help of other developers through a specific section on the project's Web page. Interested developers (users) respond to such requests by clicking on the offer he or she is interested in. This opens a Web form

Figure 2. Private area of a user at SourceForge.Net



that sends an e-mail do the project's administrator. We will return to this point later on.

METHODOLOGY

We have studied the documentation of each of the five Web portals trying to answer 13 questions.

1. What are the requirements for the registration of a new project on the portal?
2. Does the portal offer version control systems?
3. Does it offer forums?
4. Does it offer mailing lists?
5. Does the portal supply a Web page for the project?
6. Does it offer issue tracking?
7. Does it have tools to support the documentation of the project?
8. Does the portal preserve the intellectual property of the project's owner?
9. Does it require the developers to provide support even after the project is finished?
10. Does it have tools to support task management?
11. Does it provide automatic backups of the repositories in the version control system?
12. Does it allow the developer to customize the public area (remove unwanted items from the public view)?
13. Does it have any license restrictions?

To answer these questions, we used two approaches. The first approach was the analysis of the requirements of a new project. Were there any categories the project should fit in? Were there any license restrictions? We analyzed the submission process of a new project using fictitious data in order to know what the requirements were for registering a new project. We first created a user in each of the portals and followed the submission process until the last step. We did not actually submit the project since we did not have a real project

to submit at that time. We could not have used a fake project because, in all of the portals, submitted projects first go through an evaluation step where it is approved or rejected by the Web portal managers. Only approved projects are hosted. Our fake project would probably be rejected.

The second approach was the analysis of the features of the Web portal. Since most of the features are on private areas, we mainly used the Web portal documentation to find out their features. We collected all the answers we could find about all the portals we were evaluating, and contacted the Web portal administrators, asking for confirmation. This way, we would be sure we were not making any misjudgment. Most portals (SourceForge.Net, Savannah, and ObjectWeb) replied to us promptly with feedback. In the next section, we present our evaluation in detail.

MAIN FOCUS OF THE CHAPTER: WEB PORTALS

In this section, we describe each of the five portals in detail. First, however, we focus on the features we found in all of the portals we analyzed:

- **Registered users:** Only registered users may submit projects to be hosted on the Web portals.
- **Projects must be approved:** One of the common features among the portals is the requirement of submitted projects to be approved by the portal. No project is hosted without being approved. This is done to avoid fake projects (spam), but mainly to avoid projects that are not FOSS.
- **Formal submission procedure:** Due to the necessity of approval, the portals require a series of information about the project that is being submitted. The amount and type of required information may vary from a simple description to a detailed document analyzing similar projects, planned features, components that will be used, and so forth.
- **Distribution license:** During project submission, the portals require the definition of the license under which the project results are to be distributed. In all cases, there is a list of licenses you may choose. It is usually possible to choose one of the standard licenses approved by the Open Source Initiative (OSI, 2006) such as GPL (general public license), LGPL (lesser general public license), BSD (Berkeley source distribution), MIT, and MPL (Mozilla public license), among others. It is also possible to specify a new license, but this usually increases the evaluation time of the submitted project. A license needs to be studied by Web portal administrators to check if it violates the FOSS definition. For example, the Savannah administrators check for compatibility with GPL Version 2 (Savannah-help-private@gnu.org, 2006). ObjectWeb is an exception. It requires you to choose a given license (LGPL, in this case); another license may be chosen if you explain the reasoning (Object Web Consortium, 1999).
- **Source code publication:** No portal requires that there be source code available at project submission time. The goal of these portals is to support the development of FOSS, so it is understandable that projects start with no source code at all.
- **Software development support:** All of the portals offer tools to support the development of software projects. All of the portals we analyzed offer version control systems, mailing lists, Web pages for the project, bug tracking, and task management. Some of the portals offer additional tools. We will refer to them in the sections that follow.
- **Help from external users:** All of the portals allow you to request help from the user community. This is usually done by opening

a help-wanted section in your project (only project administrators have permission to do so). After that, you may inform about the kind of help you want (interface designer, developer, tester, and support manager, among others) and wait for people holding that skill to contact you. We must warn, however, that user help is usually restricted to those projects that already provide features that may attract new developers' attention. Please do not count on external help to develop your project at the beginning.

Now we are ready to look at each portal in detail.

SourceForge.Net

SourceForge.Net (http://www.sourceforge.net) is the world biggest FOSS development site. It hosts thousands of projects developed by people from several different countries. The main goal of this portal (and also of the other portals) is to offer a centralized place where developers can control and manage the development of FOSS (Open Source Technology Group, 2005).

The philosophy of SourceForge.Net is centered on the FOSS ideas:

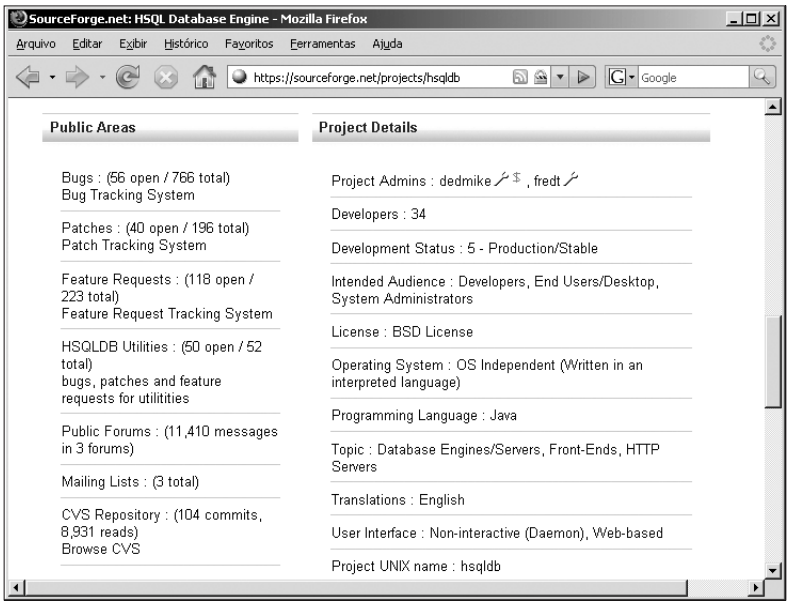
- **Facilitate the maintenance of projects:** The user community has the right to use and give support to a FOSS project, even after its activities have ceased.
- **Help to achieve the license requirements:** Some FOSS licenses require the source code to be available for a certain amount of time (usually longer than the development time period). SourceForge.Net keeps the files of finished projects to help developers to accomplish this requirement.
- **Promote reuse:** The rights of use, modification, and redistribution are guaranteed by all FOSS licenses. These rights help to promote the reuse of source code. An old project that is available at SourceForge.Net may help other developers to avoid reimplementing and testing pieces of software that other people have already implemented.
- **Allow the continuation of orphan projects:** When a project is finished, there are usually users who are interested in continuing its development. SourceForge.Net allows this to happen. Notice, however, that the project owner has to agree with this.
- **Allow project alternatives:** A project fork with alternative features may be created from a preexisting project. Both can be maintained in parallel.

To register a new project at the portal (Open Source Technology Group, 2002), it is necessary to determine the type of the project (i.e., software, documentation, Web site, peer-to-peer software, game, content management system, operational system distribution, precompiled package of existing software, software internationalization). After this, it is necessary to go through a term agreement step, and then provide a description of the project and choose the project name (which cannot be changed later). The registration process is quite simple and fast. Once registered, SourceForge.Net will take about 2 days to approve or reject the request.

After approval, the project can start taking advantage of the benefits offered by SourceForge.Net: forums, CVS (Concurrent Version System), mailing lists (public or private), the project Web page, documentation (DocManager), task management, automatic backup of the version control repository, a donation system, news and trackers for bugs, support requests, features requests, and patches (Open Source Technology Group, 2001b).

All of these tools are straightforward except for the donation system, which deserves a more detailed explanation. The donation system allows site users and projects to receive donations from other projects or site users (Open Source

Figure 3. Project at SourceForge.Net with public tools



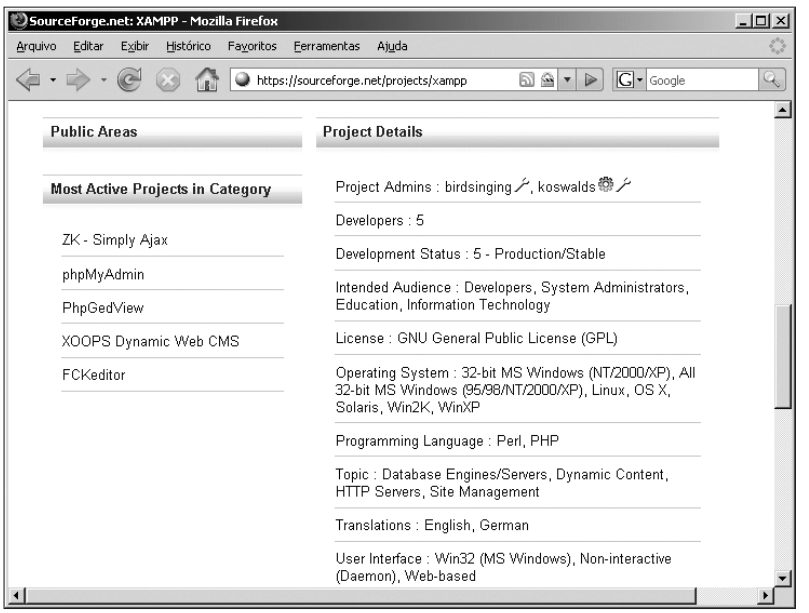
Technology Group, 2001a). The purpose of these donations is to help projects to survive. Specific projects and users may justify why they need donations. Donations are processed by the PayPal (1999) system. Both PayPal and SourceForge.Net charge fees for the donations that go through their system. The PayPal fee may vary from country to country, while SourceForge.Net charges 5% for each donation, with a minimum fee of $1. However, one cannot donate arbitrary quantities. The allowed donation values are $5, $10, $20, $50, $100, and $250 (Open Source Technology Group, 2001a).

Initially, all of these tools are visible to external users (the ones that are not registered as developers in the project). In fact, the default configuration allows even anonymous CVS checkout. However, all of this can be configured by the project administrator. This means that if necessary, some tools can be completely removed from public view. Some developers, for example, prefer to grant anonymous access to the CVS repository together with the first release, but not before that. Figure 3 shows a project for which tools are visible to external users, and Figure 4 shows a project for which every tool has been hidden from external view (only developers from that project can access the tools).

Apache

Apache Software Foundation (1999) also keeps a portal to host FOSS projects. However, Apache's stance on intellectual property is unique. Projects hosted at Apache must be donated to the Apache Software Foundation. The foundation is then responsible for deciding the project road map (Apache Software Foundation, 2005). We think this is not a disadvantage. It is just a different way of looking at things. By assuming the intellectual property, Apache takes the responsibility for the project. It can legally answer for the project and fight for the project's and the FOSS community's interests. Additionally, the project certainly gains visibility. There are cases where projects became

Figure 4. Project at SourceForge.Net with no public tools



de facto industrial standards, like Apache Web server, Tomcat, and Ant. It is also worth mentioning that the original project owner can still be involved in the development of the project.

Every project hosted by Apache must be submitted through the Apache Incubator Project (Apache Software Foundation, 2002a). The Incubator Project is responsible for informing how the Apache Foundation works, and what paths the project will go through until it is transformed into an official Apache Foundation project (or die before that). Projects currently incubated (together with unsuccessful projects) are listed in Apache Software Foundation (2006).

The registration process of a new project is quite complex. To incubate, the new project must meet the following criteria (Apache Software Foundation, 2002b):

- Be indicated by a member of the Apache Foundation
- Be approved by a sponsor

The sponsor can be one of the following:

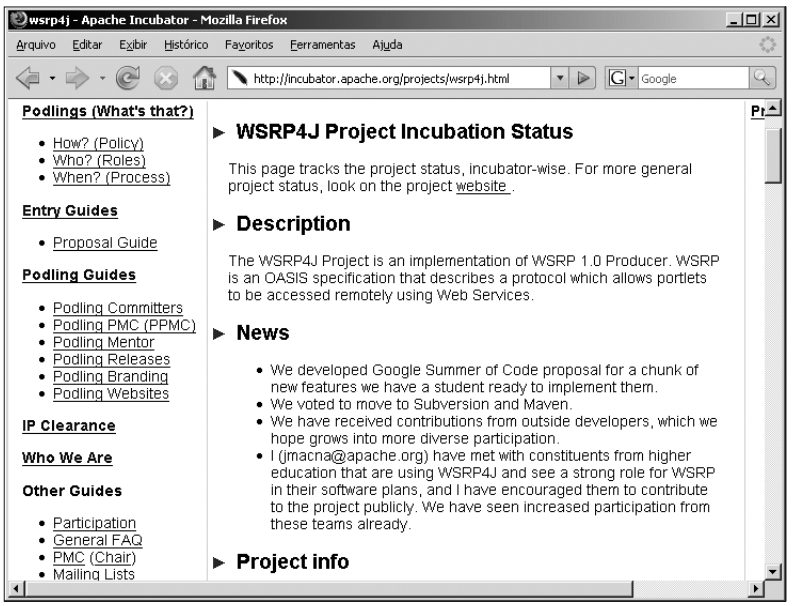
- The board of the Apache Software Foundation
- A top-level project (TLP) within the Apache Foundation, where the TLP considers the candidate project to be a suitable subproject
- The Incubator Project management committee

To initiate the hosting request process, it is necessary to submit a proposal that describes the project to the sponsor. There are no fixed items that need to be provided since the Apache Incubator documentation does not specify the level of the project detailing in the proposal or what it must contain.

After being accepted, the Incubator Project management committee is responsible for all decisions regarding the new project. Only after this point does the project receive a CVS account and a Web page under the Incubator Project.

The Apache portal offers a version control system (CVS or Subversion), mailing lists (which

Figure 5. Project incubated at Apache



can be exclusively for the project or in conjunction with the Incubator Project), a Web page, documentation (Apache Forrest), bug tracking, and task management.

Figure 5 shows a project incubated at Apache.

Tigris

Tigris (2005) is a FOSS community focused on building tools for collaborative software development that only hosts projects related to that mission. Tigris is hosted at Collabnet (Collabnet Inc., 2006), which is a provider of solutions in software development. Collabnet is currently responsible for hosting OpenOffice and Subversion, two very popular FOSS projects. It is important to notice that hosting a project at Tigris is free while it is not when hosted directly under Collabnet. As Collabnet charges a fee for this service, we do not analyze it here.

Projects hosted at Tigris must fit in one of the following categories:

- **Construction:** Tools for coding, testing, and debugging
- **Deployment:** Tools for software deployment and update
- **Design**
- **Issue tracking**
- **Libraries:** Reusable components
- **Personal use:** Personal projects of Tigris collaborators
- **Processes:** Projects related to software development processes
- **Professional use:** Professional software engineering (courses, certificates, professional practices)
- **Requirements:** Software requirement management tools
- **Software configuration management**
- **Student use:** Student class projects
- **Technical communication**
- **Testing**

The only requirements for the registration of a new project are that it falls into one of the listed

Figure 6. Project hosted at Tigris

categories and that it is a collaborative software development tool. To register, users must log in and then access the link "start new project."

Tigris offers the following features: mailing lists, task management, bug tracking, a Web page for the project, news, CVS or Subversion, and forums. Figure 6 shows a project hosted at Tigris.

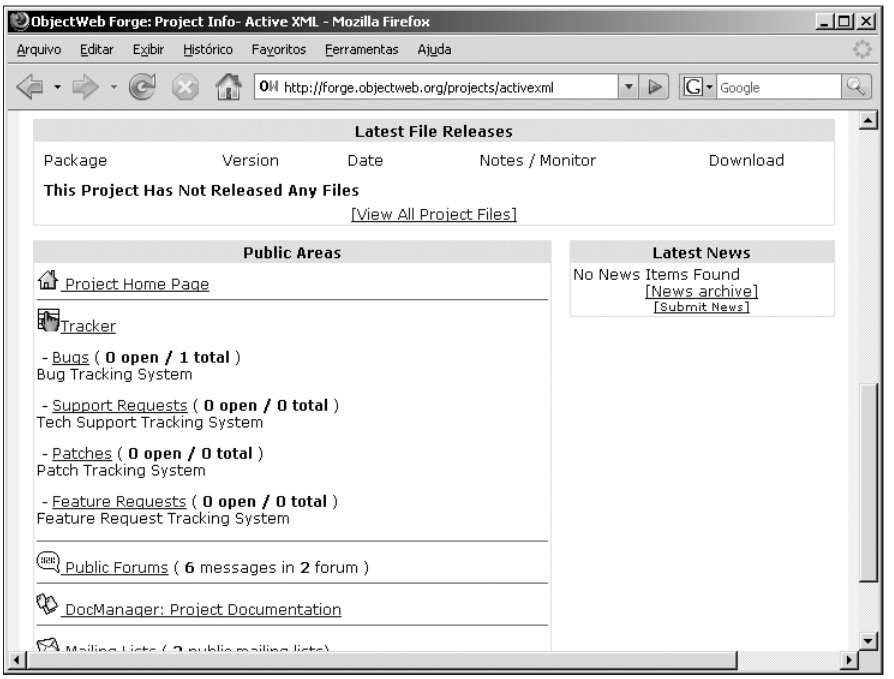
ObjectWeb

ObjectWeb (Object Web Consortium, 2005) is a consortium created in 1999 to promote the development of FOSS. It is maintained by the French National Institute for Research in Computer Science and Control (INRIA) and hosts projects such as Active XML (extensible markup language), C-JDBC, and JoNaS (Java Open Application Server), among others. The consortium is composed of a hierarchy (Cecchet & Hall, 2004).

- The board is comprised of representatives, both individuals and from companies, who are members of the consortium. The board is responsible for the policies, strategies, and direction of the consortium. The executive committee is in charge of the daily operations.
- The College of Architects is comprised of individuals chosen for their expertise and abilities. It is responsible for technically orienting the consortium, leading the development of the ObjectWeb code base, overseeing the evolution and architectural integrity of the code base, and approving new projects.

Projects on ObjectWeb, in the same way as Tigris, must be categorized. The available categories are communications, databases, desktop environments, education, games and entertainment,

Figure 7. A project hosted at ObjectWeb



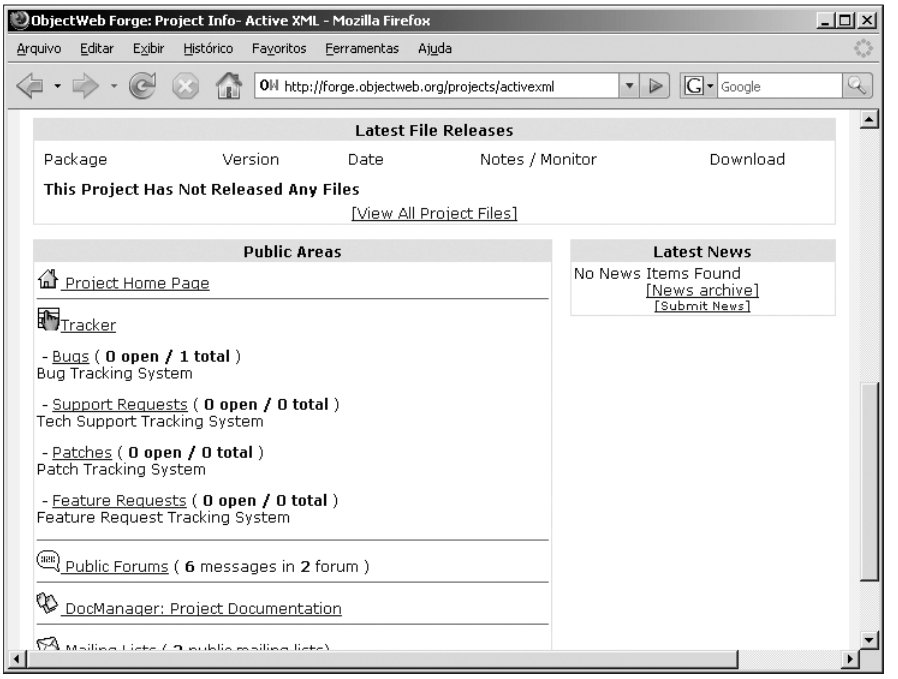
Internet, multimedia, office and business use, other unlisted topics, printing, religion, science and engineering, security, software development, systems, terminals, and text editors.

In order to be hosted at ObjectWeb, the result of the project must be a middleware component that can be reused by a great variety of software platforms and application domains. Besides this, the project members must participate in the discussions of the evolution of the code base of ObjectWeb, participate in the definition of this evolution, and apply the architectural principles and frameworks provided by ObjectWeb to maximize the reuse of the project's source code. The discussions are made through the Web portal mailing list (Object Web Consortium, 2006).

The registration process of new projects in ObjectWeb involves several project descriptions. Detailed information about the project is required, including synergies with the projects already hosted by ObjectWeb, internationalization issues, a description of similar projects, the project team and support, the user community, and the technologies and standards implemented, among others. The list of requirements is much like a formal project submission. Additionally, the LGPL is the recommended license, but a different license may be accepted if you can justify the use of another.

ObjectWeb offers several advantages to the projects it hosts. Among them, we can cite CVS, a Web page, a forum, a mailing list, task management, backup and trackers for bugs, support requests, patches, and feature requests. Figure 7 shows a project hosted at ObjectWeb. In addition, they promote annual events to gather its College of Architects and a demonstration conference that aims at approximating potential users or developers to the projects hosted at ObjectWeb.

Figure 8. Project hosted at Savannah



As with SourceForge.Net, ObjectWeb also allows projects to request help from external developers.

Savannah

Different from the other portals we analyzed here, the Savannah portal (Free Software Foundation, 2000b) is focused on free software projects. It hosts projects that fall into one of the following four categories.

- **Software project:** A software project that runs over a free operational system without depending on any non-FOSS; versions to non-free operational systems can be provided as long as there is also a (possibly more complete) version for free systems.
- **Documentation project:** A FOSS documentation project distributed under a free documentation license (Free Software Foundation, 2000a)
- **Free educational book:** A project to create free educational textbooks distributed under a free documentation license
- **Free Software Foundation/GNU project:** A project approved by the GNU project coordinator, Richard Stallman

Non-GNU projects are hosted at http://savannah.nongnu.org, but the functionalities of both portals are the same.

The registration process of a new project requires a detailed description of the project. If you already have existing source code, you must include a URL to it and a list of libraries used in the source code. This is done to make sure no non-free library is used. However, the existence of source code is not an obligation.

Table 1. Comparisons of the portals

	SourceForge.Net	Apache	Tigris	ObjectWeb	Savannah
Project registration	Depends on approval	Approved by Apache Incubator	Depends on approval	Depends on approval	Depends on approval
Version control	V	V	V	V	V
Customization of tools to avoid external access	V	V	V	V	V
Forum	V	--	V	V	--
Mailing list	V	V	V	V	V
Project Web page	V	V	V	V	V
Issue tracking	V	V	V	V	V
Documentation	V (DocManager)	V (Forrest)	--	V	--
Intellectual property	owner	Apache Foundation	owner	owner	owner
No need to support project after termination	V	--	V	V	?
Task management	V	V	V	V	V
Backup	V	?	?	V	--
Restrictions regarding the project	Categories	Find a sponsor	Collaborative software development tool	Formal submission process	Categories
Restrictions regarding license	OSI-approved license	Apache Software License (ASL)	OSI-approved license	LGPL	GNU GPL compatible

Savannah offers a smaller list of advantages to its users when compared with other portals: CVS, a Web page, a mailing list, bug tracking, support request management, and task management. As with the other portals, it is also possible to hide some of these functionalities from external users. Figure 8 shows the public area of a project hosted at Savannah.

Help from external developers can be achieved by a process similar to the ones at SourceForge.Net and ObjectWeb.

FUTURE TRENDS

Web portals play a major role in the success of free and open source software. Considering the service business around FOSS, we believe that portals will tend to follow ObjectWeb's line of FOSS promotion. We believe portals will increasingly offer more services to users in addition to hosting projects. Such services will probably include dissemination of the FOSS they host and promotion of the approximation of potential users or developers. ObjectWeb nowadays promotes this by organizing architectural meetings with its associates, where people are encouraged to approximate and collaborate. These meetings usually include presentations of newcomer projects so that the community knows what is happening and what the new projects are.

CONCLUSION

In this section, we present a comparison of the analyzed portals. The criteria for this comparison were specified previously. Table 1 summarizes

the comparison. A question mark (?) indicates that there was not enough information to evaluate the item.

Regarding support, FOSS development is voluntary. This means that you are not (and should not be) obligated to maintain your code. Some of the portals we have analyzed make this point clear by explicitly saying that you need not offer support for your project, and will not be penalized if or when you discontinue your project. Among these portals are SourceForge.Net, Savannah, and ObjectWeb. The remaining portals do not clearly state this, but they probably follow this criterion since they allow projects to be removed from the portal. The removal is not complete though. All the public information of the project prior to removal remains at the portal (existing file releases, CVS history, forums, etc.).

Intellectual property is another important issue. All of the portals (except for Apache) preserve the intellectual property of the project owner.

Portals that offer the major number of advantages are SourceForge.Net and ObjectWeb. If you pretend to host your project on ObjectWeb, you would have to consider using LGPL. Another issue to be considered in ObjectWeb is the complex registration process. Nevertheless, ObjectWeb has good reputation in academia because of the strong collaboration of INRIA. ObjectWeb requires that a new project have financial supporters in order to guarantee the continuation of the project development. As result, we found at this portal a group of well-known projects, for example, JOnAS (1999), C-JDBC (2002), and eXo Platform (2005).

Regarding automatic backup, Savannah does not have a formal backup policy. However, it does back up the data (including CVS repositories) on a nonregular basis (Savannah-help-private@gnu.org, 2006).

We hope this analysis will be useful for developers who need to choose a Web portal for their projects. We have done this study to find a Web portal to host ParGRES (http://pargres.nacad.ufrj.br/), a free software project supported by FINEP and Itautec (Brazil). After conducting a careful analysis of the hosting options (having also analyzed a Brazilian Web portal), we came to the decision of hosting ParGRES at ObjectWeb. Despite all of the advantages it offers to users, there are mainly two reasons for our decision. First, ObjectWeb opens the possibility of collaboration with a similar project (C-JDBC) that is already hosted there; second, since ParGRES is an academic project, we took the good reputation of ObjectWeb in academia as an important plus to our case. ParGRES has been available at ObjectWeb since November 2005 (http://forge.objectweb.org/projects/pargres/).

ACKNOWLEDGMENT

This research was partially supported by FINEP, Itautec, and CNPq.

REFERENCES

Apache Software Foundation. (1999). *The Apache Software Foundation.* Retrieved January 23, 2006, from http://www.apache.org/

Apache Software Foundation. (2002a). *Apache incubator project.* Retrieved January 28, 2005, from http://incubator.apache.org/

Apache Software Foundation. (2002b). *Incubation policy.* Retrieved January 23, 2006, from http://incubator.apache.org/incubation/Incubation_Policy.html

Apache Software Foundation. (2005). *How the Apache Software Foundation works.* Retrieved January 28, 2005, from http://apache.org/foundation/how-it-works.html#management

Apache Software Foundation. (2006). *Incubated projects.* Retrieved January 23, 2006, from http://incubator.apache.org/projects/

Cecchet, E., & Hall, R. S. (2004). *Objectweb projects life cycle: A practical guide for Objectweb projects.* Retrieved February 2, 2005, from https://forge.objectweb.org/register/ObjectWeb-project-lifecycle-v0.3.pdf

C-JDBC. (2002). *Clustered JDBC.* Retrieved June 26, 2006, from http://c-jdbc.objectweb.org/

Collabnet Inc. (2006). *Collabnet: Distributed development on demand.* Retrieved January 23, 2006, from http://www.collab.net/

DiBona, C., Stone, M., & Cooper, D. (2005). *Open sources 2.0: The continuing evolution.* Sebastopol, CA: O'Reilly.

eXo Platform. (2005). *eXo platform.* Retrieved June 26, 2006, from http://c-jdbc.objectweb.org/

Free Software Foundation. (1996). *GNU project.* Retrieved January 5, 2005, from http://www.gnu.org/

Free Software Foundation. (2000a). *Free software and free manuals.* Retrieved February 14, 2006, from http://www.gnu.org/philosophy/free-doc.html

Free Software Foundation. (2000b). *Savannah.* Retrieved February 1, 2005, from http://savannah.gnu.org/

Java Open Application Server (JOnAS). (1999). *Java open application server.* Retrieved June 26, 2006, from http://jonas.objectweb.org/

Object Web Consortium. (1999). *Objectweb forge: Project information.* Retrieved January 17, 2006, from http://forge.objectweb.org/register/projectinfo.php

Object Web Consortium. (2005). *Objectweb open source middleware.* Retrieved January 31, 2005, from http://www.objectweb.org/

Object Web Consortium. (2006). *Projects life cycle.* Retrieved June 29, 2006, from https://wiki.objectweb.org/Wiki.jsp?page=ProjectsLifeCycle

Open Source Initiative. (2006). *The approved licenses.* Retrieved January 17, 2006, from http://opensource.org/licenses/

Open Source Technology Group. (2001a). *Sourceforge.Net: Donation system* (Document d02). Retrieved January 23, 2006, from http://sourceforge.net/docs/D02/en/

Open Source Technology Group. (2001b). *Sourceforge.Net: Service listing* (Document b02). Retrieved January 23, 2006, from http://sourceforge.net/docs/B02/en/

Open Source Technology Group. (2002). *Sourceforge.Net project hosting requirements and the project registration process.* Retrieved January 21, 2006, from http://sourceforge.net/docman/display_doc.php?docid=14027&group_id=1

Open Source Technology Group. (2005). *Sourceforge.Net.* Retrieved January 5, 2005, from http://www.sourceforge.net

PayPal. (1999). *Paypal.* Retrieved January 23, 2006, from http://www.paypal.com/us/

Perens, B. (1997). *The open source definition.* Retrieved January 17, 2006, from http://www.opensource.org/docs/definition.php

Savannah-help-private@gnu.org. (2006). *Comparison of open source Web portals.*

Stallman, R. M. (2002). Why free software is better than open source. In J. Gay (Ed.), *Free software, free society: Selected essays of Richard M. Stallman* (chap. 6, pp. 55-60). Boston: GNU Press.

Tigris. (2005). *Tigris open source software engineering.* Retrieved January 31, 2005, from http://www.tigris.org/

Wikipedia. (2006). Retrieved from http://www.wikipedia.org

KEY TERMS

Forum: A discussion board on the Internet (Wikipedia, 2006).

Intellectual Property: Umbrella term used to refer to the object of a variety of laws, including patent law, copyright law, trademark law, trade-secret law, industrial design law, and potentially others (Wikipedia, 2006).

Issue Tracking: Also known as bug tracking, it is a system designed to manage change requests of a software. It can also be used to manage bugs.

Mailing List: A collection of names and addresses used by an individual or an organization to send material to multiple recipients. The term is often extended to include the people subscribed to such a list, so the group of subscribers is referred to as the mailing list (Wikipedia, 2006).

Task Management: Software capable of managing lists of pending tasks.

Version Control System: A system that tracks and stores changes on files (source code, binary files, and text files, among others). Such systems are able to retrieve old versions of a given artifact as long as such version has been stored some time before in the system.

Web Portal: A site on the World Wide Web that typically provides personalized capabilities to visitors (Wikipedia, 2006).