Improving community awareness in software forges by semantical aggregation of tools feeds

Quang Vu DANG
Christian BAC
Olivier BERGER
Institut TELECOM, SudParis
9, rue Charles Fourier, 91011 Evry Cedex, France
{quang_vu.dang; christian.bac; olivier.berger}
@it-sudparis.eu

Xuan Sang DAO
Institut de la Francophonie pour l'Informatique
42, Ta Quang Buu, Hai Ba Trung, Hanoi, Vietnam
dxsang@ifi.edu.vn

ABSTRACT

It is rather difficult to monitor or visualize what can be the contribution of a member in a project, especially when the project uses multiple tools to produce its results. This is the case for collaborative development of FLOSS software, that use Wiki, bug tracker, mailing lists and source code management tools. This paper presents an approach to data collection by using aggregation of feeds published by the different tools of a software forge. To allow this aggregation, collected data is semantically reformatted into Semantic Web standards: RDF, DC, DOAP, and FOAF. Resulting data can then be processed, re-published or displayed to project members. We implemented this approach in a supervision module that has been integrated into the PicoForge platform. This module is able do draw a live graph of the social community out of the different sources of data, and in turn export semantic feeds for other uses.

Keywords

free and open source software development, public data, semantic Web, social network analysis, community of practice, social filtering, RDF, FOAF, DOAF.

1. INTRODUCTION

Free libre and open source software (FLOSS) projects often use development platforms called “software forges” (such as SourceForge, Savannah, Gforge, Trac, PicoForge...). A *forge* helps them organize their community, provides collaborative tools to the members (such as Source versioning, Mailing list, Wiki, Bug tracker, forum ...).

In order to help analysis by researchers on FLOSS there are already many tools that retrieve information about FLOSS development. These tools analyze the data stored by the collaborative tools such as CVS/SVN logs, database of bugs, mail archives... To facilitate the mining, data is collected from forges, anonymized and then processed. This allows only differed studies on the projects and don't provide any real-time vision to the project members. Moreover there are FLOSS projects which are developed on multiple forges, but tools work often from independent data sources only, so one needs to integrate project data from multiple sources [3].

In this paper we propose an approach to data collection from FLOSS development projects, using aggregation of *feeds* provided by the tools in the forges, to better monitor activities. Our approach also seeks interoperability of tools, to help collect data of multiple projects across multiple forges and across multiple communities. The freshness of informations in these feeds will help members to have an accurate vision of their project's current state.

This early work is conducted in the frame of a PhD thesis on quality in FLOSS projects. In this respect, we plan to be also able to apply metrics on the resulting data to help understand the “quality” of the community. In our longer term plan we also want to find relationship, if any, between, the quality of the produced software and the liveliness of the community.

In Section 2 we recap some research initiatives and their tools focusing on public data about FLOSS, as well as the use of Semantic Web standards for representation of metadata. Section 3 describes our approach and methodology. Section 4 presents a case study using our approach to implement a supervision tool in the PicoForge forge.

2. BACKGROUND

2.1 Existing research initiatives and tools

In order to provide data to researchers interested in FLOSS projects, there have been many attempts to retrieve and analyze information about FLOSS development.

The FLOSSmole[1] project provides public data about FLOSS development for academic research. It includes data and analysis from SourceForge, Freshmeat, RubyForge, ObjectWeb... [3]. The FLOSSMetrics[2] project aims at constructing, publishing and analyzing a large scale database with information and metrics about libre software development coming from several thousands of software projects, using existing methodologies, and tools already developed. The SQO-OSS[3] project aims at providing a platform with a pluggable architecture for software development organizations to observe the OSS quality by using novel techniques and algorithms in data mining and metric analysis of source code [2].

There are many tools which were used in the above projects to retrieve the information from libre software projects hosted in forges: CVSAnaly, MailingListStars, pyTernity... Each tool has a proper data schema, and it seems difficult to integrate information across tools or to share information between communities.

2.2 Useful Semantic Web standards

Semantic Web standards can help to annotate, organize or integrate the information on projects, actors and their production. This can help finding and sharing public data on FLOSS project as was proposed in [4].

RSS generally refers to a "simple" XML dialect used in the syndication of Web content with poor semantic content[4]. This standard is usually used to publish updates information whose nature changes frequently, typically in forges, this can be lists of news, new or changed items in wikis, notification of e-mails received in public forum, bugs filed, or "commits" made to source code.

RSS 1.0[5] (aka *RDF Site Summary*, or RSS/RDF) is formulated using the RDF (Resource Description Framework) *standard*, and may be consumed either as a XML format or interpreted as a labelled graph model. A RSS/RDF *channel* has a basic set of properties (link, title, description) and is associated with an RDF Sequence of items. Each item itself has a link, title, description and optional attributes such as Dublin Core[6] elements (**dc:creator, dc:contributor**...). The great advantage of RDF here is the ability to multiplex different semantic fields inside the same document, thus helping achieve interoperability between multiple consumers of the same feeds.

FOAF[7] (Friend Of A Friend) is an XML/RDF schema for describing people and the relationships between them and the things they create and do. FOAF can be used to draw the social network of communities of practice by analyzing **foaf:knows** attributes' graph.

Each software development project may be described by using the DOAP[8] (Description Of A Project) schema that is an XML/RDF vocabulary to describe open source projects. It provides a description of a software project and its associated resources, including participants and Web resources.

Through the use of RDF, a single RSS/RDF feed can contain semantic information combining different vocabularies (for instance, FOAF + DOAP). For example, SourceKibitzer[9] generates DOAP/FOAF metadata from hosted projects and members profiles.

In [5], Simmons and Dillon propose an ontology based approach to address knowledge management in open source software development. This ontology covers the following concepts: Participant, Role, Activity, Procedure, Artefact, Tool. Other ontologies describe data in community. The SIOC[10] project defines an ontology that contains concepts necessary to express information contained in online community sites (Ex: boards, blogs, etc). Baetle[11] is an ontology to describe software bugs and trouble tickets that aims at becoming the standard used by Bugzilla and other repositories to enable people to query for bugs across repositories. These ontologies could be integrated in the RSS/RDF schema to describe the informations published, for instance inside the RSS item occurrences(For example bug number, file modified).

3. APPROACH AND METHODOLOGY

3.1 Improving project analysis frameworks

Our general questioning is whether we can integrate in the forges various interoperable tools which can both :

- improve community awareness for project members,
- and help provide high-level indicators for analysis of community/product quality.

[1] http://ossmole.sourceforge.net/

[2] http://www.flossmetrics.org/

[3] http://www.sqo-oss.eu/

[4] Here, we refer to non-RDF base variants, such as *Rich Site Summary* (RSS 0.91) and *Really Simple Syndication* (RSS 2.0)

[5] http://Web.resource.org/rss/1.0/

[6] http://dublincore.org/documents/1999/07/02/dces/

[7] http://xmlns.com/foaf/spec/

[8] http://usefulinc.com/doap/

[9] *Recognizing Contributions to Open Source Software* (http://www.sourcekibitzer.org/)

[10] Semantically-Interlinked Online Communities (http://sioc-project.org/)

[11] Bug And Enhancement Tracking LanguagE (http://code.google.com/p/baetle/)

We seek to extract semantical data (expressed in standard formats) from the forge's tools, which can be transported and processed by high-level analysis tools that will help measure quality criteria on projects and communities.

Using standard formats such as those described in Section 2 will help "plug" different interoperable analysis or visualization tools into compliant forge platforms. This should help compare various methods or tools developed independently, by allowing them to monitor the same projects.

The present initial work will address only basic measurement and visualization, that can be extracted from an initial set of semantic informations extracted from the forges, to help validate the approach.

Later research will take advantage of such tools to help address higher-end goals such as analyzing and visualizing :

- What is happening in a project?
- How does the community of a project work?
- Is a community able to integrate newcomers and share knowledge?
- What are trends and predictable events, etc.

The answers should be provided by trying to measure several related factors:

- who is working ?
- how are members working (which tools) ?
- what are members doing ?
- where/when do members work ?
- etc.

3.2 Feeds in standard formats to monitor projects activity

In general, in a Forge, an item in a RSS feed is the result of a `member` action, so aggregating RSS feeds allows to measure one person's activity. The aggregated RSS helps `member`s to easily review all recent actions in their project.

Moreover it can help measure activity of the whole community through some parameters such as: the number of actions, the number of active members, the number of used tools with number of actions in each tool.

Analyzing the aggregation of different feeds also helps constructing a social network for that community of practice by combining relations in different activities conducted in heterogeneous tools. Two members are related when they discuss the same subject in mailing lists, edit the same topic in a Wiki or commit the same module in Subversion. These combined relations give a more comprehensive vision about the collaboration in the community than the results analyzed in single sources, such as [6], for instance.

Historical data can be used to analyze activity trends of members of a project: monthly activity, daily activity, hourly activity, etc.

3.3 Semantic aggregator and processor in a forge

The forge's tools often publish RSS feeds either non-semantic, or with heterogeneous dialects. A first step will be to identify (for each tool) the syntaxic elements that can be converted to semantic information.

Then, our approach consists in plugging to the very forge used by the projects, a semantic aggregator based on RSS/RDF. The resulting RSS/RDF feeds (or channels) will mix other Semantic Web standards such as DC, DOAP and FOAF , allowing their manipulation by different tools which understand these standard formats.

To enrich information in the RSS/RDF items, we will integrate the FOAF schema which describes the developers, as authors of the notified actions.

In addition, each project will be described with DOAP in the forge's portal. It then publishes a public feed that provides the information integrated from feeds of tools used by this project.

To include a pointer in the DOAP document to the public feeds, we use the **rdfs:seeAlso** attribute and add an object **rss:channel**. In a RSS item the **doap:project/doap:name** attribute is used to point to its project. There will also be FOAF attributes such as **foaf:Person/foaf:nick, foaf:Person/foaf:mbox** which describe the contributor of an item. Figure 1 shows links in these integrated RDF schema.

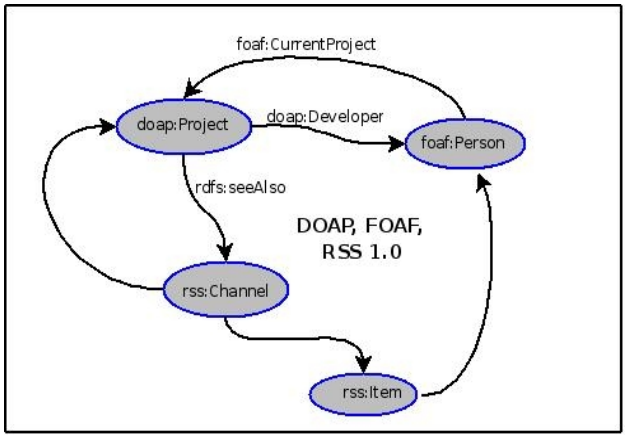

Figure 1: Integrated RDF Schema

An example of such an RSS/RDF channel description with a non-semantic reference(`rss:link`) to web page of project and semantic reference to a DOAP document of its project is :

```
<rss:channel>
<rss:title>projectA</rss:title>
<rss:description>WebSVN RSS feed - projectA
</rss:description>
<rss:link>http://forgedemo.org/projectA</rss:link>
<doap:Project>
        <doap:name>projectA</doap:name>
        <rdfs:seeAlso>
           http://forgedemo.org/projectA.rdf
        </rdfs:seeAlso>
</doap:Project>
</rss:channel>
```

As example of an RSS/RDF item's description with a semantic reference to the FOAF document of its contributor is :

```
<rss:item>
<rss:title>minor fixes in version 2 branch
</rss:title>
<rss:link>http://forgedemo.org/projectA</rss:link>
<rss:description>
Rev 2463 - toto (3 file(s) modified)
minor fixes in version 2 branch
</rss:description>
<foaf:person>
        <foaf:nick>toto</foaf:nick>
        <foaf:mbox>toto@projecta.org</foaf:mbox>
        <rdfs:seeAlso>
                http://forgedemo.org/toto.rdf
        </rdfs:seeAlso>
</foaf:person>
<dc:date>Mon, 02 Jun 2008 22:18:02 +0100</dc:date>
</item>
```

FOAF document in detail of **toto** user is:

```
<foaf:person>
        <foaf:nick>toto</foaf:nick>
        <foaf:mbox>toto@projecta.org</foaf:mbox>
        <foaf:currentProject>
          <doap:Project>
            <doap:name>projectA</doap:name>
            <rdfs:seeAlso>
               http://forgedemo.org/projectA.rdf
            </rdfs:seeAlso>
          </doap:Project>
        </foaf:currentProject>
</foaf:person>
```

4. CASE STUDY ON PICOFORGE

4.1 Adding a supervision tool in the forge

Started in order to use it as a pedagogical platform, PicoForge[12] is a libre-software system released under the GNU GPL license, which eventually evolved as a general-purpose forge. It provides a Web-based collaborative work platform built on top of several existing mature libre software tools. PicoForge provides project hosting facilities for small teams of software developers. It is mainly used for teaching and research activities nowadays.

The forge integrates several libre software Web applications: TWiki, Sympa, CVS, Subversion, WebSVN, Mantis, much of which include *RSS feeds* to track activity.

Our "Supervision" tool is a module developed in order to be integrated to a future release of PicoForge. It will fetch, mix and process, for each project, the initially non-semantic RSS feeds already published in the collaborative tools. It is also able to add other RSS feeds from outside PicoForge which are related to the projects. Figure 2 shows the architecture of the "Supervision" tool.

It allows "querying" public projects of the platform to export a list of projects in RSS format or in RDF as FOAF + DOAP. Here, DOAP describes project, FOAF describes their members , and public RSS/RDF feeds will publish semantic notifications of project activity.

In order to have a vision of community of practice of projects, the "Supervision" tool also provides graphical visualization, to members of the projects, of statistical information and the constructed social network of project members, based on their actions.

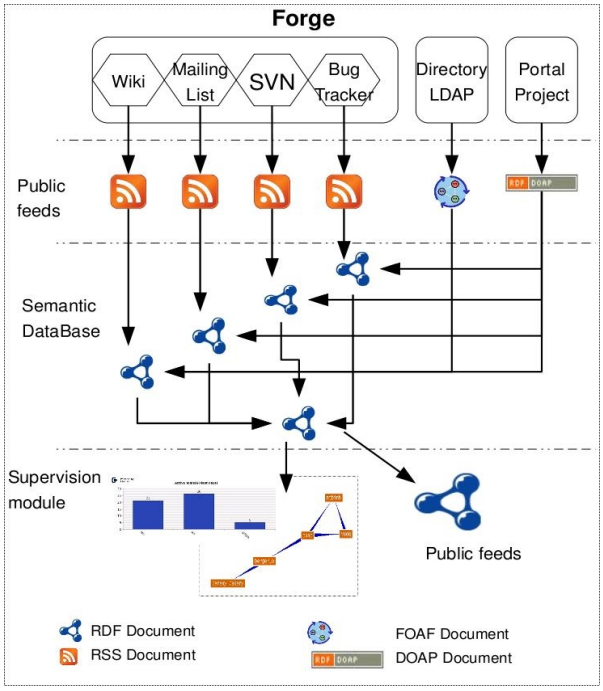

Figure 2: Supervision on PicoForge

Several libraries were used to implement the "Supervision" tool in PicoForge:

– *RDF API for PHP*[13] (aka RAP) : a Semantic Web toolkit written in PHP. It allow to parse, store, query, manipulate, serialize and serve RDF. It support for the RDQL query language.

[12] Http://www.picoforge.org/

[13] http://www4.wiwiss.fu-berlin.de/bizer/rdfapi/

– *RSS_PHP*[14]: a RSS parser and XML parser for PHP using DOMDocument. This library is used to parse, reformat RSS documents before storage in a RDF database managed through the RAP library.

– *Libchart*[15] : a PHP library to create charts such as Bar charts (horizontal or vertical), Line charts, Pie charts. It is used to generate statistical charts in the Supervision tool.

– *TouchGraph*[16] : allows the visualization of graphs such as social networks. It is a Java application. In the Supervision tools we use the TGLinkBrowser library to display the members network.

4.2 First results on a collaboration project

In our case study we used this "Supervision" tool to observe a public project for software development hosted in the PicoForge installation at Institut TELECOM, SudParis[17]. In the following, we provide screenshots of the graphs produced by the developed tool, available to project members, which try and answer the questions proposed in Section 3.

Figure 3 shows the total activity in the project with different used tools (wiki, SVN, Sympa) in the 60 last days. It helps project manager know what is the kind of common activities in recent time. The Figure 4 shows the active users with the number of actions.

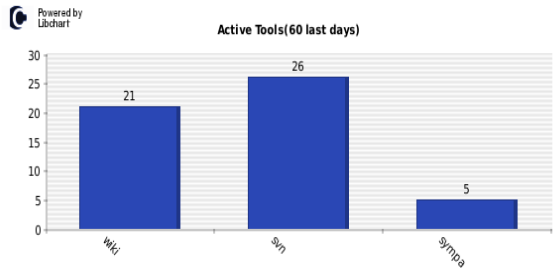

Figure 3: Total activities in project in 60 last days

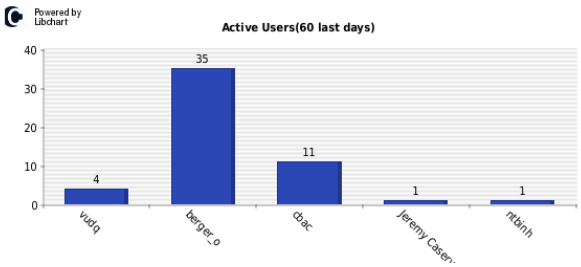

Figure 4: Active users in 60 last days

[14] http://rssphp.net/

[15] http://naku.dohcrew.com/libchart/

[16] http://www.touchgraph.com/

[17] http://picoforge.int-evry.fr/

Moreover based on interaction of members reconstructed in the "Supervision" tool, we can represent a live network of the active members and relations between them in a time frame (for example, in Figure 5, a network on the current month). In this case we have combined three kind of relations: co-commit on Subversion, co-drafting on Twiki or discussion on Sympa mailing list. So it gives a more comprehensive vision about the collaboration in the community than with results analyzed on a single source. The visualization is fresh, dynamic and updated in real-time, available to the members of the project.

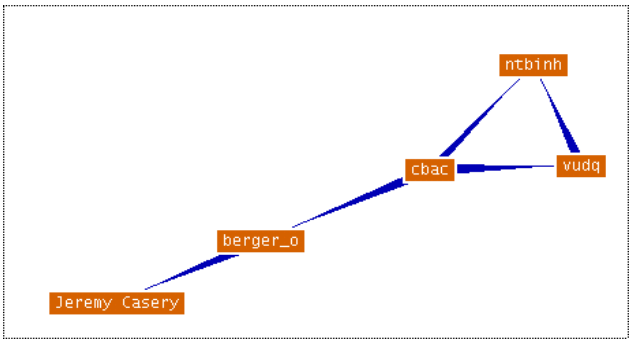

Figure 5: Members network

5. CONCLUSION

This paper presents an approach to help collect, transport, and correlate live data about multiple projects across multiple software forges to enlighten group awareness.

We extended the RSS 1.0 RDF schema by integrating FOAF, DOAP and Dublin Core which help to describe semantical information about projects activity in RSS feeds produced by the forges.

By aggregating data from different sources, using standard semantic Web formats, we seek better interoperability, which may allow the use and comparison of generic high-level analysis and visualization tools which may be plugged in various forges systems.

The produced graphs help members of teams to get a better real-time visualization of the inter-person cooperation in their projects.

In the future, we hope that the this approach will foster availability of interoperable tools, and calibration of analysis methods in order to improve these tools.

We hope it will facilitate advanced research on detection of activity patterns of members or to trace co-evolution of software and community. For instance, based on social network detected, we hope that it will be possible to apply social filtering technique to improve peer review process. The content of data collected and the interaction of members could be used to analyze the communication quality, the cognition process of teams in projects hosted on software development forges.

Availability of such advanced tools, interoperable with the forge platforms, for projects' direct benefit would certainly help improve FLOSS development environments, provided that good use is made of the informations. There may actually be privacy concerns, requiring that access rights to such tools or correlated data be considered. Still these were not considered in this initial work, and remain to be discussed with communities using the forges.

6. REFERENCES

[1] Howison, J., Conklin, M., Crowston, K. (2006). *FLOSSmole: A collaborative repository for FLOSS research data and analyses*, International Journal of Information Technology and Web Engineering. 1(3). July-September, 2006. pp 17-26.

[2] de Groot, A., Kugler, S., Adams, P.J. and Gousios, G., *Call for Quality: Open Source Quality Observation*, in IFIP International Federation for Information Processing, Volume 203, Open Source Systems, eds. Damiani, E., Fitzgerald, B., Scacchi, W., Scotto, G., (Boston: Springer), pp. 57-62.

[3] Conklin, M. (2007). *Project entity matching across FLOSS repositories*. In Proceedings of the 3rd International Conference on Open Source Systems. Limerick, Ireland. June 11-14, 2007. pp. 45-57.

[4] Anupriya Ankolekar, James D. Herbsleb, Katia Sycara, *Addressing Challenges to Open Source Collaboration With the Semantic Web* , in Taking Stock of the Bazaar: The 3rd Workshop on Open Source Software Engineering, the 25th International Conference on Software Engineering (ICSE). 2003. Portland OR, USA.

[5] Gregory L. Simmons, Tharam S. Dillon, *Towards an Ontology for Open Source Software Development* , In IFIP International Federation for Information Processing, Volume 203, Open Source Systems, eds. Damiani, E., Fitzgeralg, B., Scacchi, W., Scotto, M., Succi, G., (Boston:Springer), pp 65-75.

[6] Luis López-Fernández, Gregorio Robles, Jesús M. González-Barahona and Israel Herraiz, *Applying Social Network Analysis Techniques to Community-Driven Libre Software Projects*, Proceedings: International Journal of Information Technology and Web Engineering, Vol. 1, Issue 3, September 1st – 2006.

[7] Jin Xu, Yongqin Gao, Christley S, Madey G, *A Topological Analysis of the Open Source Software Development Community*, System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference, 03-06 Jan. 2005.

[8] Luis Lopez-Fernandez, Gregorio Robles, Jesus M. Gonzalez-Barahona, *Applying Social Network Analysis to the Information in CVS Repositories*, Proceedings of the Mining Software Repositories Workshop. 26th International Conference on Software Engineering (Edinburgh, Scotland), May 25th – 2004.