

Supporting situation awareness in FLOSS projects by semantical aggregation of tools feeds

Quang Vu DANG, Christian BAC
Olivier BERGER, Valentin VLASCEANU
Institut TELECOM, Télécom SudParis

9, rue Charles Fourier, 91011 Evry Cedex, France

{quang_vu.dang; christian.bac; olivier.berger; ion_valentin.vlasceanu} @it-sudparis.eu

ABSTRACT

It is rather difficult to monitor or visualize what can be the contribution of a member in a collaboration project, especially when the project uses multiple tools to produce its results. This is the case for collaborative development of FLOSS software, that uses 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, FOAF and EvoOnt. Resulting data can then be processed, re-published or displayed to project members. This approach was used to implement a supervision module that is integrated into the PicoForge platform. This module is able to draw a live graph of the social community out of the different sources of data, and in turn exports 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, DOAP, EvoOnt, Helios BT.

1. INTRODUCTION

Free libre and open source software (FLOSS) projects often use development platforms called “software forges” (such as SourceForge, Savannah, Gforge/FusionForge, Trac, PicoForge...). A *forge* helps them organize their community and provides collaborative tools to the members (such as source versioning, mailing lists, wikis, bug trackers, forums ...).

In order to help researchers conduct analysis on FLOSS development, there are already many tools that retrieve information about FLOSS. These 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 as described in [1]. This allows only differed studies on the projects and doesn'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 in such cases, one needs to integrate project data from multiple sources [3].

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

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

In this paper we propose an approach for 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 work is conducted in the frame of a PhD thesis on quality in FLOSS projects. In this respect, we plan to also be 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 draws the framework for supporting a situation awareness tool, while Section 4 describes our approach and methodology. Section 5 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 FLOSSmole¹ project provides public data about FLOSS development for academic research. It includes data and analysis from SourceForge, Freshmeat, RubyForge, ObjectWeb... [1]. The FLOSSMetrics² 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-OSS³ 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].

We also have a large scale facts databases from GNU/Linux distributions like Ultimate Debian Database (UDD) in the Debian distribution or Doc4 in the Mandriva distribution.

There are many tools which were used in the above projects to retrieve the information from libre software projects hosted in

¹ <http://ossmole.sourceforge.net/>

² <http://www.flossmetrics.org/>

³ <http://www.sqo-oss.eu/>

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 content⁴. This standard is usually used to publish information updates whose nature changes frequently, typically in forges, which 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 1.0⁵ (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 Core⁶ 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.

FOAF⁷ (Friend Of A Friend) is an 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 DOAP⁸ (Description Of A Project) schema that is an 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, RDFohloh⁹ 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 metadata in communities. The SIOC¹⁰ project defines an ontology that contains concepts necessary to express

information contained in online community sites (for instance boards, blogs, etc). Baetle¹¹ is an ontology to describe software bugs and trouble tickets that aimed at becoming the standard used by Bugzilla and other repositories to enable people to query for bugs across repositories. EvoOnt¹² is a set of software ontologies which provide the means to store all elements necessary for software analyses including the software design itself (SOM) as well as its release (VOM) and bug-tracking (BOM) information. Helios_bt is also an ontology that describes software bugs, based on EvoOnt BOM ontology, but as a complementary addition to it, in the context of the use of the Helios project. It 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. SITUATION AWARENESS SUPPORT TOOL

Situation awareness is the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future [8].

We desire a support situation awareness tool that helps members in community to be aware of what is happening around them to understand how information, events, and their own activities impact their goals and objectives, both now and in the near future. This tool is based on the most common theoretical framework of SA provided by Endsley [9]. Endsley's model illustrates three stages or steps of SA formation: perception, comprehension, and projection.

3.1 Perception

The first step in achieving SA is to perceive the status, attributes, and dynamics of relevant elements in the environment. Thus, Level 1 SA, the most basic level of SA, involves the processes of monitoring, cue detection, and simple recognition, which lead to an awareness of multiple situational elements (objects, events, people, systems, environmental factors) and their current states (locations, conditions, modes, actions).

This step is supported by the data collection phase of our tool. We need to collect data from internal sources, provided by forge's tools, and also from external ones (for example: information of distributions).

Time is an important concept in SA, as a situation is dynamic, changing at a tempo dictated by the activities of individuals, task characteristics, and the surrounding environment. Hence, the interesting data for SA are recent events and activities that are in relation to the community's current situation.

3.2 Comprehension

The next step in SA formation involves a synthesis of disjointed Level 1 SA elements through the processes of pattern recognition, interpretation and evaluation. Level 2 SA requires integrating this information to understand how it will impact upon the individual's goals and objectives. This includes

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 <http://rdfohloh.wikier.org/>

¹⁰ Semantically-Interlinked Online Communities (<http://sioc-project.org/>)

¹¹ Bug And Enhancement Tracking LanguageE (<http://code.google.com/p/baetle/>)

¹² <http://www.ifi.uzh.ch/ddis/evo/>

developing a comprehensive picture of the world or of that portion of the world of concern to the individual.

This step is implemented in the data processing phase in our tool. 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.

3.3 Projection

The third and highest level of SA involves the ability to project the future actions of the elements in the environment. Level 3 SA is achieved through knowledge of the status, dynamics of the elements and comprehension of the situation (Levels 1 and 2 SA), and then extrapolating this information forward in time to determine how it will affect future states of the operational environment.

The present initial work for level 3 AS 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?

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

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

Semantic Web technologies provide new opportunities for synthesizing information from numerous, disparate and often heterogeneous information sources and can be used to better support complex knowledge fusion. In our longer term vision, the data of the two steps described above can be integrated with historical databases like Flossmetrics or UDD, represented using RDF schemas. It could be helpful to build prediction models for this step.

Figure 1 shows the architecture of a SA support tool for software forges. There are multiple level of data which is accessible by users. Section 4 will present in detail our solution to implement this tool.

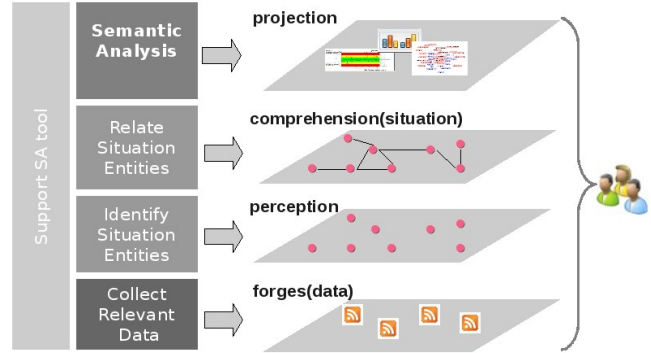


Figure 1: Framework of SA support tool for software forges

4. APPROACH AND METHODOLOGY

From the description of the situation awareness support tool above, we identify some data properties for our solution. The data is extracted from multiple sources, disparate and heterogeneous. Then the data must be integrated together for analysis. Moreover the data is fresh information. Consequently our solution is aggregating feeds provided by forge's tools and using RDF schema: RSS/RDF, FOAF, DOAP, EvoOnt schemas ... to represent data.

4.1 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 members 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 [6,7], for instance.

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

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.

4.2 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 syntactic 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, FOAF and EvoOnt, 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. There are also references to VOM, BOM documents when the item refers to a *commit* or a *bug*.

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.

An example of such a RSS/RDF channel description with a non-semantic reference (rss:link) to project's web page (human-understandable content) and semantic reference (doap:Project) to a DOAP document (machine-understandable content) of its *project* and also semantic references to its *items* is:

```
<http://forgedemo.org/projectA/feed>
  rdf:type rss:channel .
  rss:title 'projectA'.
  rss:description '...' .
  rss:link 'http://forgedemo.org/pjtA.html'.
  doap:Project <http://forgedemo.org/pjtA>.
  rss:item <http://forgedemo.org/projectA/item1> .
  rss:item <http://forgedemo.org/projectA/item2> .
  rss:item <http://forgedemo.org/projectA/itemn> .
```

Next, here is an example of project's description using the DOAP schema:

```
<http://forgedemo.org/pjtA>
  rdf:type doap:Project .
  doap:name 'projectA' .
  doap:Developer <http://forgedemo.org/dev/toto>.
```

The following example is of an RSS item's description with semantic references to FOAF, VOM, BOM resources. This item describes the *commit* number 2463 which is made by the "toto" user to fix the *bug* number 236.

```
<http://forgedemo.org/projectA/item1>
  rdf:type rss:item .
  rss:title 'minor fixes in version 2 branch' .
  rss:link 'http://forgedemo.org/projectA' .
  rss:description 'Rev 2463 - toto (3 file(s)
modified) fix bug #236' .
  foaf:Person <http://forgedemo.org/dev/toto> .
  vom:Version <http://svndemo.org/projectA/v2463>.
  bom:Issue <http://bugdemo.org/projectA/bug236> .
  dc:date 'Mon, 02 Jun 2008 22:18:02 +0100' .
```

This example of a VOM fragment gives details on commit number 2463:

```
<http://svndemo.org/projectA/v2463>
  rdf:type vom:Version .
  vom:number '2463' .
  vom:hasAuthor <http://forgedemo.org/dev/toto> .
```

The example of a BOM fragment for details of bug number 236 is :

```
<http://bugdemo.org/projectA/bug236>
  rdf:type bom:Issue .
  bom:number '236' .
  bom:isFixedBy <http://svndemo.org/pjtA/v2463>.
```

The last example is of a FOAF fragment for details of the "toto" user:

```
<http://forgedemo.org/dev/toto>
  rdf:type foaf:Person .
  foaf:nick 'toto' .
  foaf:inbox <toto@forgedemo.org> .
  foaf:currentProject <http://forgedemo.org/pjtA>.
```

5. CASE STUDY ON PICOFORGE

5.1 Adding a supervision tool in the forge

Started in order to use it as a pedagogical platform, PicoForge¹³ 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.

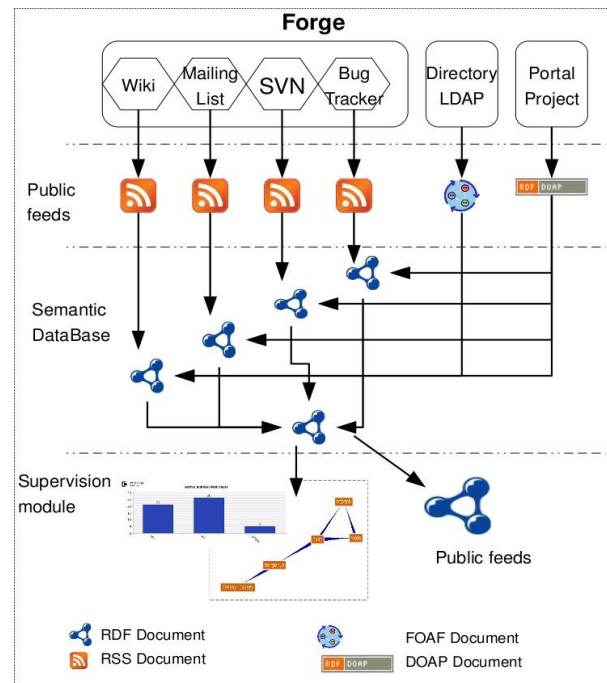


Figure 2: Supervision on PicoForge

¹³ [Http://www.picoforge.org/](http://www.picoforge.org/)

In order to have a vision of the communities 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.

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

- *RDF API for PHP*¹⁴ (aka RAP) : a Semantic Web toolkit written in PHP. It allows to parse, store, query, manipulate, serialize and serve RDF. It supports for the RDQL query language.
- *RSS PHP*¹⁵: 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*¹⁶ : 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*¹⁷ : 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.

5.2 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, SudParis¹⁸. 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 managers know what is the kind of common activities in recent times. Figure 4 shows the active users with the number of actions.

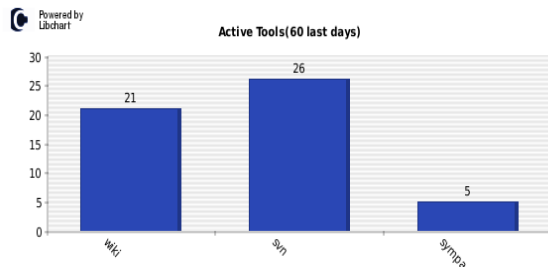


Figure 3: Total activities in project in 60 last days

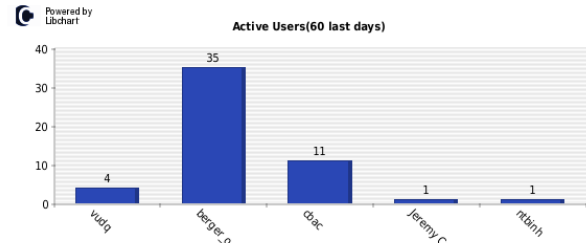


Figure 4: Active users in 60 last days

Moreover, based on members interaction 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

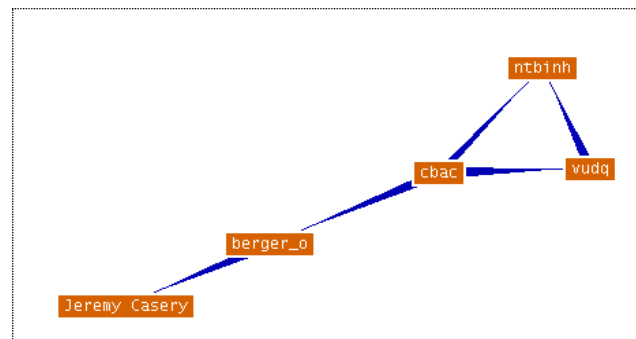


Figure 5: Members network

6. CONCLUSION

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

We extended the RSS 1.0 RDF schema by integrating FOAF, DOAP, VOM, BOM 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. Since visualization capabilities are clearly useful in terms of enabling improved situation awareness. So 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 members activity patterns 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

¹⁴ <http://www4.wiwiss.fu-berlin.de/bizer/rdfapi/>

¹⁵ <http://rssphp.net/>

¹⁶ <http://naku.dohcrew.com/libchart/>

¹⁷ <http://www.touchgraph.com/>

¹⁸ <http://picoforge.int-evry.fr/>

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.

7. 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., Fitzgerald, 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] 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.
- [8] Endsley, M.R., 1995a. *Towards a theory of Situation Awareness in Dynamic Systems*, Human Factors, Vol. 37, pp. 32-64.
- [9] Endsley, M.R., 1995b. *Measurement of Situation Awareness in Dynamic Systems*, Human Factors, Vol. 37, pp. 65-84.