Extraction of Contributor Information from Software Repositories

Omar Alonso
Premkumar T. Devanbu
Michael Gertz
Dept. of Computer Science
University of California, Davis
{oralonso, mgertz, ptdevanbu}@ucdavis.edu

ABSTRACT

Open-source projects derive their vitality and dynamism through the contributions of many volunteers. While only a relatively small number of people are developers, who have commit privileges, many others actually contribute to the source code. In this paper, we examine the relationship of contributors to developers, and how much help the contributors actually provide. To discover this information we mine a CVS data source to find bugs, submissions, and contributors per transaction. The output is the inner network of a particular developer or the contributors to the project through a developer. We enhance the output using a couple of information visualization metaphors that allow a better exploration of the contributors to a project. We present a prototype implementation that describes our research using the Apache HTTP Web server project as a case study.

Keywords

Text mining, information extraction, contributor credit, expertise identification, information visualization, databases.

1. INTRODUCTION

It is common belief that everyone can participate in an open source project. Another common belief is that hundreds of people contribute in an open source project. In this paper, we examine in some detail the mechanics by which people contribute to open-source projects. By now, it has been well established (*e.g.,* in the Apache project [4]) that there is a relatively small group of developers who actually make those changes. This would indicate that the structure is very similar to a traditional industrial development team. But in fact, there are others who actually contribute source code, bug fixes, patches etc. There contributions play a significant role in the success of these projects. Naturally, several questions arise about the contributors. Are they in the hundreds? Do they work alone or do they participate in the discussions? What are the mechanisms for motivating and rewarding these contributors?

Research in mining software repositories has been very active lately with many projects working beyond just source code. Email and CVS sources contain rich data for a wide range of analysis [10], [13], [14]. The social aspect is also an important component of the mining process like the identification of active participants, and owners in a project, and overall structure of a team [7], [8], and [12].

2. FRAMEWORK

We use *Minero,* a framework for the analysis and exploration of software repositories [1]. Minero provides database and mining techniques for the integration, processing, analysis, and management of different types of open source repository data. Data integration enables us to manipulate different information sources within a single conceptual framework. A database query language like SQL and associated loading, optimization and evaluation tools let us process the data to answer complex queries, even analytical questions to extract trends and correlations.

Each data source is unique, both in model and representations so different techniques have to be applied to extract meaningful information from the source data. We illustrate this below with an example; in our case, we need to combine relational querying, XML path querying, and regular expression matching to extract information about contributors and committers.

3. DATA PREPARATION AND MINING

Apart from the source code, the most prominent data sources for an open source project are email archives (where developers communicate with each other), the CVS repository, the bug database, and the documentation. The content, except source code, is mainly free text with some rudimentary level of structure, if any.

3.1 CVS characteristics

The CVS log file usually has a field where a developer is expected to enter detailed information about the transaction, such as which bug was fixed (if there is one open), who submitted the patch (if there is a submission) and who has reviewed it. From the data characteristic perspective, it is *semi-structured data*: good structure for some items (like author, file name) and unstructured for the messages and comments part. For example:

MSR 06.

```
<entry>
<date>2004-08-11</date>
<weekday>Wednesday</weekday>
<time>15:44</time>
<author>wrowe</author>
<file>
<name>modules/proxy/proxy_ajp.c</name>
<revision>1.6</revision>
</file>
<msg>Close only when needed.
Submitted by: jfclere
</msg>
</entry>
```

In practice, developers don't always enter all the different elements that are expected; the mining tools must be robust in cases where data is missing. However, when available, this is potentially very useful information: the identity of the person who is credited for a contribution to a particular commit. This data gives us useful information about the relationship of contributors and developers, and the effects of the contributor/committer relationship on the productivity of the committer. We can ask questions such as the following:

1. *Are developers who are the most productive, i.e., entering the most amount of code and also the ones who credit the most contributors?*
2. *Are the developers who are most active on the email lists also the ones who issue the most credited submissions?*

We present a preliminary analysis of the data in this respect. The analysis suggests that the answers to the above questions are affirmative.

3.2 Information Extraction and Mining

Given the structure of the textual message in CVS, using regular expressions we can identify some patterns and write simple extraction tasks. As long as there is some sort of formatting regularity, automated extraction is feasible [9]. Ideally, we expect to find the following although in practice not all the data is available.

```
<msg>message
PR:
Obtained from:
Submitted by:
Reviewed by:
</msg>
```

We can easily pull out names of contributors that provided fixes for bugs (identified as "PR") and populate part of the database schema for later mining. The schema contains information about authors (developers), entries, files, bugs (PR), and contributors. Using standard SQL queries we can group, sort, and count the data.

3.3 Presentation Layer

Information visualization is defined as "visual representations of abstract data to amplify cognition". In the context of large data sets, visualization can help users navigate as well as provide a summary of the data collection [2]. To enhance the schema for discovery of patterns and a close examination, we added a presentation layer that consists of views for information visualization support.

We have equipped the database with a package that takes an XML representation of the data to be visualized and generates data exploration views as the main interface to modularize the implementation. Data exploration views are constructed using the traditional view mechanism available in a database system. The advantage of data exploration views is that they are always the same while the data sources and visualization metaphors can change.

4. IMPLEMENTATION

Our prototype extends our existing Minero framework (an Oracle 10*g* database [5]). We obtained a dump of the CVS repository in XML that contains 15,589 entries between 1996 and 2004 for the Apache 2.0 release.

The second step was to manipulate XML, using the parsed SAX representation, to process the entire file and populate the database schema. The advantage is that one can use SQL, XPath, regular expressions or any combination of existing query languages to perform information extraction. The following script is an example of some rules for detecting a contributor in a message text. The query illustrates the convenience of being able to refer to relational and semi-structured data into a single query.

```
select extractValue(entry,'/entry/author') a,
   count( extractValue(entry,'/entry/msg') ) sub
from cvs_table where regexp_like(
extractValue(entry,'/entry/msg'),'Submitted by:
[a-zA-Z]+')
group by extractValue(entry,'/entry/author')
```

The following table shows, as an illustration, a few records with authors (developers who have a committer id), the number of entries in the CVS log file (transactions) and of those transactions how many have a "submitted by" comment.

```
AUTHOR                ENTRY SUBMITTED
-------------------- ------ ---------
nd                     1814        89
wrowe                  1792        87
trawick                1634        30
stoddard                789        23
```

With this information we can now retrieve all the names of contributors for a given developer.

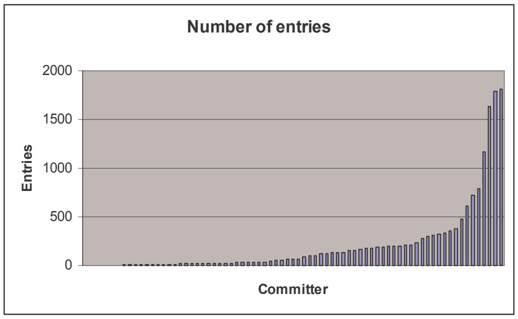
For presentation purposes, the visualization component issues SQL queries to the data exploration view and returns results in an intermediate XML representation format. A transformer then modifies the data for the particular visualization metaphor (a tree map or force graph). Finally, the applet generator produces a Web page with all Java applet parameters. This applet reads the query output stream and produces the visualization in the browser. We implemented two visualization metaphors, one using a commercial tree map [3] and a second one based on an open source toolkit [6].

The prototype implements the three main components of any knowledge discovery system: data acquisition, text mining, and information presentation. The next section provides some preliminary results using the tools described.

5. PRELIMINARY ANALYSIS

Our preliminary analysis indicates that in the case of Apache, a small group of developers perform all commits to the project. These people have been around for a number of years and are the owners of significant portions of the code. For example developers wrowe, nd, trawick, rbb, stoddard, slive, jerenkratz, and bnicholes account for the bulk of transactions, and all of them have been working on the project for several years. This is consistent with the project meritocracy philosophy [11].

The CVS log file shows that there are 75 unique developers that account for 15,589 entries. As we can see from the chart, a small group did over hundreds of transactions. This is consistent with similar findings about individual author contributions on the Linux project reported in [8].



The force graph visualization (figure 1) allows us to get an idea of the team structure by identifying clusters of contributors. The central node represents the name of the project (Apache) and each committer (represented as a node) has a direct path to it. As one can see from the picture, some clusters of people are visible. At the center of a cluster is the committer name, highlighted in red. The inner network consists of people who have contributed submissions, highlighted in orange. The structure indicates a similarity with the onion-like structure as proposed in [12].

Some people prefer one visualization metaphor in favor of another for information discovery, so we offer a tree map as a second choice that provides more structural information about a person. The tree map (figure 2) shows the relative contributions of all the developers in a single window.

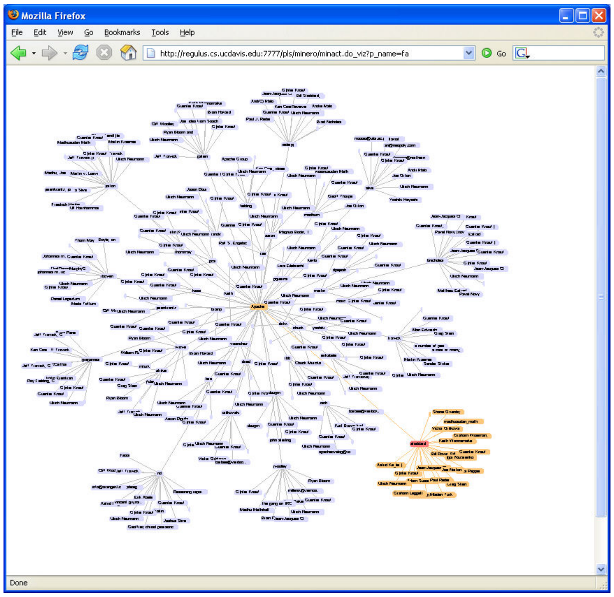
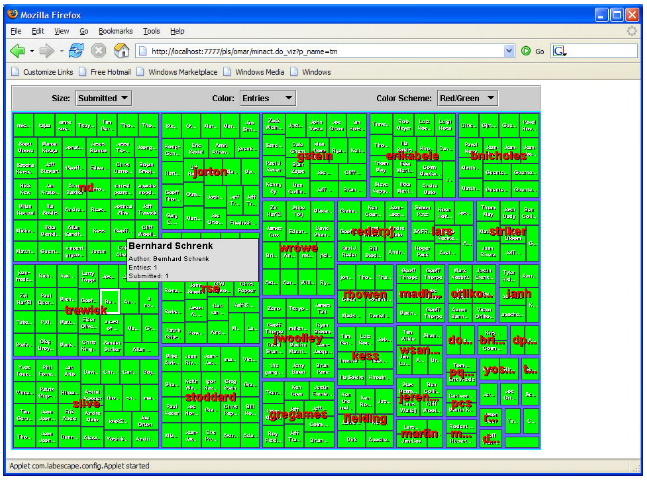

Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://regulus.cs.ucdavis.edu:7777/pls/minero/minact.do_viz?p_name=fa
Go
Apache
Done

Figure 1. Clusters of contributors in the force graph

Each developer's contributions are shown in tiles within a window; there are as many tiles as developers that have at least one contributor. Within a developer's tile, there are individual cells indicating the identities of the contributors who are credited in that developer's submissions. One can read the map by starting on the top left, where developers have large number of submitters; ending at the bottom right, where few submitters are available per developer. One can see that developer trawick, has benefited from contributors like Larry, Oleg, and Mark, to name a few. The tree map helps to drill down information per contributor, therefore exploring the comments field in more detail. When one hovers the mouse over a cell, one can see more data on pop-up box like in the case of Bernhard Schrenk.



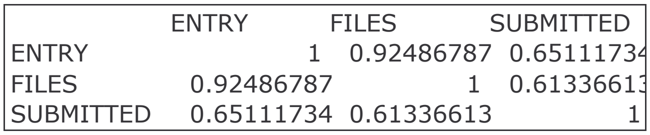
Size: Submitted
Color: Entries
Color Scheme: Red/Green

Bernhard Schrenk
Author: Bernhard Schrenk
Entries: 1
Submitted: 1

Applet com.labescape.config.Applet started

Figure 2. Exploration of contributors using a tree map

It is also important to point out that not every entry in CVS is well commented, therefore indicating that probably important information is missing.



	ENTRY	FILES	SUBMITTED
ENTRY	1	0.92486787	0.65111734
FILES	0.92486787	1	0.61336613
SUBMITTED	0.65111734	0.61336613	1

Figure 3. Spearman's Rank Correlation Table showing that developers who are more active tend to give more credit to contributors

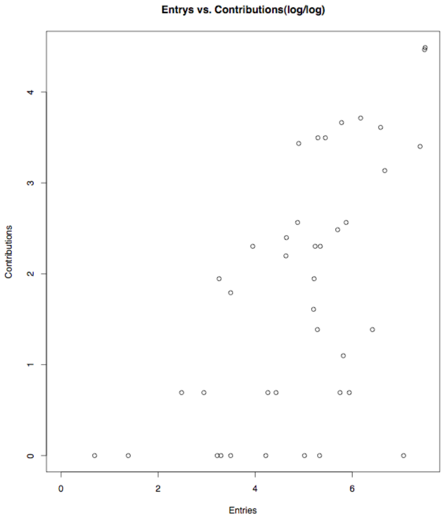


Figure 4. Scatterplot showing relationship between number of Entries (transactions) and number of contribution credits (log/log plot).

6. Data Analysis

We were able to extract data regarding contributions for 75 developers who have entries in CVS. Out of these, 39 gave no credit in their commit logs to contributors. The most credits were given by developer *nd,* 89 separate credits in total. Using this data, we tabulated the data, including 1) total number of entries (or transactions), 2) the total number of files touched (note that more than one file can be touched per transaction) and 3) the number of contributions listed. Using these entries, we ran Spearman's rank correlation between the different data values, and the table shown in Figure 3 was obtained. Not surprisingly, per developer, the number of files and the number of entries are strongly correlated. The rank correlation between developers who executed the most transactions and the developers who gave the most credit is noteworthy. This relationship can also be seen in Figure 4, which is a scatterplot of the values for different developers, showing the relationship between the number of entries and the number contributor credits.

7. CONCLUSIONS AND FUTURE WORK

Clearly, the CVS data source has rich information for discovering contributors and activities in the Apache project. Using information extraction we were able to identify developers and contributors through developers, which gives important insight information about how the work is performed. A preliminary data analysis provides an indication that there is a good connection between a developer's openness to contributions and their productivity.

We have developed a prototype using an Oracle DBMS, a commercial tree map visualization SDK, and an open source toolkit to demonstrate the main ideas. We plan to continue working on mining assets with different technique to extract useful information.

8. ACKNOWLEDGMENTS

We thank Brian Behlendorf and Brian Fitzpatrick from the Apache team for answering questions and sharing the CVS data set.

9. REFERENCES

[1] O. Alonso, P. Devanbu, and M. Gertz "Database Techniques for the Analysis and Exploration of Software Repositories" *First MSR Workshop,* ICSE 2004, Scotland UK.

[2] S. Card, J. Mackinlay, B. Shneiderman. *Readings in Information Visualization* Morgan-Kaufmann, 1999.

[3] Lab Escape, www.labescape.com

[4] Apache Web server project, httpd.apache.org

[5] Oracle 10gR2 Reference documentation, tahiti.oracle.com

[6] J. Heer, S. Card, and J. Landay "prefuse: A Toolkit for Interactive Information Visualization" *CHI 2005*, Portland OR

[7] S. Huang and K. Liu. "Mining Version Histories to Verify the Learning Process of Legitimate Peripheral Participants". *Second MSR Workshop*, ICSE 2005, Saint Louis, USA.

[8] B. Dempsey *et al*. "Who Is An Open Source Software Developer?" *CACM*, February 2002, Vol. 45, No. 2.

[9] P. Jackson and I. Moulinier *Natural Language Processing for Online Applications: Text Retrieval, Extraction, and Categorization.* John Benjamins Publishing (2002).

[10] Y. Kidane and P. Gloor "Correlating Temporal Communication Patterns of the Eclipse Open Source Community with Performance and Creativity", *NAACSOS* (2005).

[11] R. Fielding "Shared Leadership in the Apache Project" *CACM,* April 1999, Vol 42, No. 4.

[12] K. Crowston and J. Howison "The Social Structure of Free and Open Source Software Development", *First Monday*, Vol. 10, No. 2 (February 2005).

[13] A. Mockus, R. Fielding, and J. Herbsleb "Two Case Studies Of Open Source Software Development: Apache and Mozilla". *ACM TOSEM* 2002, Vol. 11, No. 3.

[14] L. Lopez J. Gonzalez-Barahona, and G. Robles, "Applying Social Network Analysis to the Information in CVS Repositories". *First MSR Workshop,* ICSE 2004, Scotland UK