What Topics do Firefox and Chrome Contributors Discuss?

Mario Luca Bernardi
Dept. of Engineering,
University of Sannio, Italy
mlbernar@unisannio.it

Carmine Sementa
Dept. of Engineering,
University of Sannio, Italy
csementa@unisannio.it

Quirino Zagarese
Dept. of Engineering,
University of Sannio, Italy
quirino.zagarese@unisannio.it

Damiano Distante
Fac. of Economics, Unitelma
Sapienza Univ., Italy
distante@unitelma.it

Massimiliano Di Penta
Dept. of Engineering,
University of Sannio, Italy
dipenta@unisannio.it

ABSTRACT

Firefox and Chrome are two very popular open source Web browsers, implemented in C/C++. This paper analyzes what topics were discussed in Firefox and Chrome bug reports over time. To this aim, we indexed the text contained in bug reports submitted each semester of the project history, and identified topics using Latent Dirichlet Allocation (LDA). Then, we investigated to what extent Firefox and Chrome developers/contributors discussed similar topics, either in different periods, or over the same period. Results indicate a non-negligible overlap of topics, mainly on issues related to page layouting, user interaction, and multimedia contents.

Categories and Subject Descriptors

D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement—*Corrections, Enhancement*

General Terms

Measurement

Keywords

Text Mining, Bug reports, Topic Mining, Co-Evolution

1. INTRODUCTION

In the Web 2.0 era, browsers are among the most widely used pieces of software, since many applications are directly accessed through the Web. Nowadays, applications such as email clients, office productivity tools, chat/ communication forums, but also more sophisticated applications are entirely Web-based, avoiding the users to install heavy clients on their machines, and giving them the capability to access an application (and their data) from any available machine simply equipped with a Web browser.

Two very popular browsers are Firefox[1] and Chrome[2]. Firefox is an open source browser built at the end of 2004 from the Mozilla suite. At the time of writing, Firefox is considered as the second most popular browser in the world. More recently (in 2008) Google released the Chrome browser, based on WebKit[3]. The intended points-of-strength of this browser are its performances, security level, and light-weightiness if compared with other browsers. At the time of writing, Chrome is considered to be the third-most popular browser in the world. Over their histories, both browsers underwent changes to fix issues, as well as to introduce new features to cope with user needs, with the continuous evolution of Web technologies, and to compete with features provided by similar products.

The *goal* of this study is to investigate on the topics discussed in issue reports of the two projects, Firefox and Chrome, with the *purpose* of understanding how the discussed topics evolved over time and to what extent—in the same or in different periods—the two projects discussed related topics. Since the two projects belong to the same domain, it is likely that during their lifetime both Firefox and Chrome encountered similar problems and/or introduced similar features—e.g., related to page layouting, handling of HTML language and multimedia content. The *quality focus* concerns what are the areas of interest—identified by topics—in which most of the development and maintenance work is focused in a given period of time. For example, it would be useful to understand what is the relevance multimedia-related features are having during recent years. The *context* consists in 88,538 Firefox issue reports, and 49,986 Chrome issue reports observed in the periods January 2005–June 2010 and August 2008-June 2010 respectively.

To analyze topics being discussed, we indexed the text of issue reports and used an information retrieval technique, Latent Dirichlet Allocation (LDA) [**?**], to identify topics discussed in periods of six months. Recently, topic models have been used in software engineering, for instance by Thomas *et al.* [**?**], who extracted topics from source code changes adopting topic models to describe software evolution, and by Linstead and Baldi [**?**], who introduced a metric based on LDA to measure the coherence of issue reports. Once extracted the topics, we analyze and compare them with the aim of answering the following research questions:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MSR '11, May 21-22 2011, Honolulu, Hawaii
Copyright 2011 ACM ...$10.00.

[1] http://www.mozilla.org/firefox
[2] http://www.google.com/chrome
[3] http://www.webkit.org

- **RQ1:** What are the topics discussed by issue reports of the two browsers over time?
- **RQ2:** Is there any overlap between topics discussed in Firefox and in Chrome issue reports (even if in different periods of time)?
- **RQ3:** Is there any overlap between topics discussed in Firefox and in Chrome issue reports in the same period of time?

The paper is organized as follows. Section 2 describes the approach we followed to extract data from issue tracking systems. Section 3 reports and discusses results of the study we conducted. Section 4 concludes the paper and outlines directions for future work.

2. THE MINING APPROACH

This section describes the approach we followed to mine topics from issue reports of Firefox and Chrome.

2.1 Identification of the Relevant Issue Reports

Firefox and Chrome issue reports are stored into two different kinds of issue tracking system, i.e., *Bugzilla*[4], and *Chromium* (Google Code)[5] respectively. Before performing the analysis, we identify the subset of issue reports relevant for our study. For Chrome, we consider all issues in the observed time period 2008-2010. For Firefox, we consider the following Mozilla products that are directly related to the browser: *Core*, *Firefox*, *Directory*, *JSS*, *NSPR*, *NSS*, *Plugins*, and *Rhino*. Also, we only consider issues opened since the first semester of 2005, i.e., since the stand-alone Firefox browser was released.

2.2 Term Extraction and Indexing

After having identified the relevant issue reports, we extract terms from each of them. For Firefox, we consider the issue title and description (excluding all other comments), while for Chrome we consider the issue title and the text contained in the *pre* tag following the *div* named *author*.

Since we download issue reports as HTML pages, we only extract their textual content, pruning out tags. We exclude numeric values and special characters, considering only terms beginning with a letter and at least two characters long. Then, we remove the following stop words (i) English stop words (ii) C/C++ keywords and (iii) HTML tag names. After stop words removal, we bring back terms to their stem using a Perl implementation of the Porter stemmer [**?**]. Finally, we perform a further pruning, by removing terms appearing in more than 5% of issue reports for that semester (thus not helping to discriminate different reports).

The result of this step is, for each issue report, a vector of terms weighted by their occurrences in the issue report.

2.3 Topic Mining

To identify topics discussed by issue reports, we group issues in semesters (S1: Jan-June, and S2: July-December) based on their opening date. Alternative choices could have been considering groups composed of the same number of issue reports, or grouping issues around major releases. However, we prefer to use a temporal grouping to allow a direct comparison between the Firefox and Chrome histories.

Then, we use Latent Dirichlet Allocation (LDA) [**?**] to identify which topics are described in issues reported in a semester. LDA allows to fit a generative probabilistic model from the term occurrences in a corpus of documents. In our case documents are the set of indexed issue reports opened in a given semester. Specifically, LDA outputs a set of topics, each one consisting of a list of terms.

When applying LDA, the number of topics—in the following referred as k—has to be fixed *a priori*. We determine it with a data-driven approach. In other words, we start by fixing a large number of topics (30 in our case), and reduce it if we find duplicate topics, i.e., topics described by the same set of terms, until no more duplicate topics are found. In this study we consider—for both Firefox and Chrome—20 topics. To apply LDA, we rely on the *R*[6] library *lda*.

2.4 Topic Co-Evolution Analysis

To analyze commonalities between topics discussed by the two projects, we perform two kinds of analyses, a *coarse-grained* analysis and a *fine-grained* analysis. Given i a semester of the Firefox history and j a semester of the Chrome history, the coarse-grained analysis considers the Jaccard overlap among the top n terms (in this study we used $n = 15$) of all topics identified in semester i for Firefox and in semester j for Chrome, defined as follows:

$$\frac{\left(\bigcup_{x=1}^{k} Tf_{x,i}\right) \bigcap \left(\bigcup_{y=1}^{k} Tc_{y,j}\right)}{\left(\bigcup_{x=1}^{k} Tf_{x,j}\right) \bigcup \left(\bigcup_{y=1}^{k} Tc_{y,j}\right)}$$

where $Tf_{x,i}$ ($Tc_{y,j}$) is the set of terms describing the x-th (y-th) topic identified in semester i (j) of the Firefox (Chrome) history. The Jaccard index is a well-known Information Retrieval similarity measure, and varies between 0 (no overlap) and 1 (total overlap).

The *fine-grained* analysis is performed by identifying, among topics discussed in both projects, those having the largest overlap and thus checking whether, in two semesters i and j, Firefox and Chrome issues discuss topics that largely overlap. In other words, the *fine-grained* analysis identifies a common discussion between semester i of the Firefox history and semester j of the Chrome history if $\exists x, y$ such that:

$$\frac{Tf_{x,i} \bigcap Tc_{y,j}}{Tf_{x,i} \bigcup Tc_{y,j}} > th$$

where *th* is a threshold we set equal to 0.3, i.e., topics must overlap for at least 8 out of $n = 15$ terms.

3. STUDY: TOPICS DISCUSSED IN FIREFOX AND CHROME

To address **RQ1**, we identify the most popular topic for each semester of both projects, by assigning to each issue the topic having with it highest term overlap, and then considering the topic associated to the highest number of issues. Results are shown in Table 1. For Firefox, the earlier period is mostly related to issues about page formatting/layouting

[4] http://www.bugzilla.org

[5] http://code.google.com/chromium

[6] http://www.r-project.org

Table 1: Most popular topics for Firefox and Chrome.

Firefox

Semester	Most popular topic
2005 S1	width; px; border; height; align; xhtml; margin; background; dtd; cell; tag; black; awuest; statustext; href
2005 S2	width; background; px; height; border; row; margin; tag; xhtml; align; cell; overflow; dtd; pad; posit
2006 S1	px; width; border; cell; posit; leak; tag; height; row; background; margin; block; hidden; dtd; pad
2006 S2	enabl; disabl; usr; warn; gmake; gcc; leak; ac; configur; seamonkei; pc; wno; bin; argum; target
2007 S1	home; match; bin; zbyszek; crt; ed; xpdf; ae; hour; leak; cvsqueri; maxdat; mindat; assertion; sid
2007 S2	ctype; lc; match; moz; nsiframe; maxdat; cvsqueri; mindat; anemitz; dc; bd; sortbi; hour; filetyp; fcvsroot
2008 S1	leak; zoom; extension; addon; total; install; leaked; level; malloccount; mfreecount; mrealloccount; madoptcount; madoptfreecount; msharecount; nsstringstat
2008 S2	width; px; background; border; size; height; left; posit; align; black; elem; famili; pad; white; margin
2009 S1	ircategoryopt; home; warn; unexpected; tinderbox; unit; info; mochitest; gz; showlog; convers; deprec; seamonkei; mochikit; pass
2009 S2	elem; width; height; px; tag; size; background; xhtml; border; docum; moz; dtd; abda; pixel; left
2010 S1	slave; reftest; unexpected; mochitest; pass; info; serial; outer; moz; tinderbox; failur; gz; showlog; assertion; unittest

Chrome

Semester	Most popular topic
2008 S2	slave; messageloop; offici; signatur; pump; sec; uptim; cpu; wchar; messagepumpwin; id; inform; dispatch; info; fault
2009 S1	pump; messagepumpwin; wchar; dispatch; ntdll; reportid; hwnd; product; messagepumpforui; deleg; kernel; widgetwin; anonym; signatur; messagepump
2009 S2	player; quicktim; mb; ram; ghz; hostnam; unknown; thumbnail; playerx; output; screen; rohmac; revis; sheet; modal
2010 S1	info; player; quicktim; mb; signatur; ram; product; ghz; hostnam; playerx; sec; chromeo; favicon; unknown; uptim

Table 2: Common terms in Firefox and Chrome topics for different semesters.

Firefox	Chrome	Common terms	Jaccard
2009 S1	2010 S1	mous; left; move; bin; zn; languag; framework; plai; drag; past; doc; xb; sound; video; pdf; disabl; dylib; comput; ctrl; notif; login; entri; byte; px; libsystem; preview; cancel; foo; failur; width; youtub; drop; api; size; watch; jpg; warn; home; usr; keyboard; hang; restart; info; quicktim; account; zoom; ex; background; player; height; leak; border; product; block; focu	0.19
2009 S2	2010 S1	mous; left; move; bin; languag; framework; read; plai; drag; past; doc; slave; xb; video; pdf; disabl; dylib; comput; ctrl; entri; byte; px; libsystem; word; preview; shift; foo; failur; ffff; width; send; drop; size; signatur; warn; home; usr; keyboard; hang; restart; info; pass; zoom; ex; background; height; anim; border; unittest; product; block; fast; focu	0.19
2009 S1	2009 S2	mous; left; zn; brows; languag; email; framework; plai; drag; past; doc; xb; sound; video; pdf; dylib; ctrl; login; entri; xf; byte; px; libsystem; cancel; youtub; width; drop; api; size; enabl; reload; watch; redirect; warn; jaunti; highlight; usr; home; info; stop; quicktim; account; ex; background; player; bottom; leak; unit; screen; product; focu	0.19
2008 S2	2010 S1	mous; left; bin; es; framework; read; plai; drag; appkit; xb; sound; video; pdf; dylib; comput; ctrl; avail; login; entri; byte; px; libsystem; preview; shift; failur; width; youtub; resiz; drop; api; size; jpg; warn; upload; home; usr; keyboard; hang; restart; info; pass; account; zoom; ex; background; player; task; height; leak; border; product; block; focu	0.19
2007 S2	2010 S1	mous; bin; framework; plai; drag; past; xb; video; pdf; disabl; dylib; comput; ctrl; avail; entri; byte; px; libsystem; preview; foo; failur; width; resiz; send; requir; drop; lwp; hit; size; la; unknown; warn; upload; home; usr; keyboard; websit; hang; restart; info; debian; quicktim; zoom; background; launch; player; height; leak; border; product; block; focu	0.18

issues (*width*, *border*, *height*, *align*, *margin*, *background*, etc.), and to (X)HTML compatibility issues (*dtd*, *xhtml*). It must be noted that, while some of the above terms are HTML attributes, they were considered as appearing in the text and not as part of HTML code. Then (second semester of 2006 until 2008) topics seem to mostly refer low level and building issues (e.g., *bin*, *configur*, *gcc*, *gmake*). During the last period—i.e., since the second semester of 2008—the discussion is brought back to page formatting/layouting issues.

In Chrome, the first two semesters seem to mostly refer low-level and performance issues (*cpu*, *sec*). Later on (second semester of 2009 and first of 2010), the discussion moves towards issues related to multimedia (*player*, *playerx*, *quicktim*), while performance-related terms continue to emerge (*ram*, *ghz*, *sec*). Overall, most of the Chrome discussion looks more oriented towards performance issues and support to multimedia, reflecting the scope Chrome was conceived for.

To address **RQ2**, we first perform a coarse-grain analysis, determining the set of overlapping terms among Firefox and Chrome topics. Table 2 shows the list of terms (stems) shared between Firefox and Chrome topics in five pairs of semesters for which the overlap was higher. We compare all semesters of Chrome with all semesters of Firefox, filtering out cases where the Jaccard score is below 0.15. As one can notice by looking at terms in overlapping topics, there is a large set of terms concerning the various media being handled by the browsers (e.g., *pdf*, *doc*, *jpg*, *quicktim*, *video*, *play*, *preview*, *sound*, *youtube*). Other terms are, instead, related to more generic problems (e.g., *resiz*, *hang*, *background*, *border*, *upload*, *restart*, *redirect*). A graphical view of some noticeable topic correspondences we identified is also shown

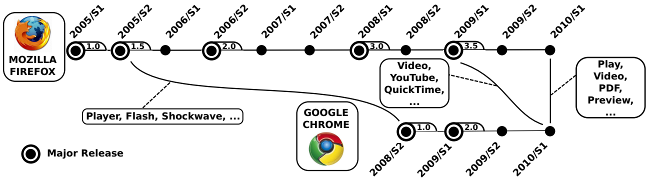

Figure 1: An excerpt of topics shared among Firefox and Chrome issues.

Table 3: Overlapping topics.

	Firefox	Chrome	Overlapping terms in the topic
T1	2008 S2	2008 S2	movi; youtub; stop; video; player; game; plai; flash; sound
T2	2009 S1	2008 S2	youtub; stop; video; player; watch; game; plai; flash; sound
T3	2010 S1	2010 S1	left; width; resiz; bottom; height; size; border; posit; px
T4	2005 S2	2008 S2	movi; stop; video; shockwav; player; game; plai; flash
T5	2006 S1	2009 S2	mous; resiz; ctrl; size; posit; screen; keyboard; focu
T6	2006 S2	2009 S1	width; resiz; background; bottom; size; visibl; posit; scrollbar
T7	2007 S2	2008 S2	youtub; movi; video; player; watch; plai; sound; flash
T8	2007 S2	2010 S1	width; resiz; bottom; height; border; size; posit; px
T9	2007 S2	2010 S1	mous; hit; ctrl; reload; shift; activ; focu; keyboard
T10	2008 S1	2010 S1	width; resiz; bottom; height; size; border; posit; px
T11	2009 S1	2010 S1	youtub; stop; video; player; watch; plai; ogg; sound
T12	2009 S1	2010 S1	width; left; bottom; height; size; border; posit; px
T13	2010 S1	2008 S2	left; width; resiz; bottom; height; border; px; pixel

in Figure 1. It is worthwhile to point out that in all cases we found overlap between a Firefox semester and a more recent Chrome semester, i.e., the topics emerged before in Firefox, likely because of its longer history, higher popularity and adoption.

Since, as discussed above, there is, indeed, a non-negligible overlap of terms in Firefox and Chrome topics, we search for specific topics overlapping for at least 8 terms. Results are shown in Table 3 (for convenience overlapping topics are numbered as T1-T13). As it can be seen, in this case the terms belong to the same topics and the results definitely look more cohesive than in Table 2. In fact, there are several cases (T1, T2, T4, T7, T11) where the topic really relates to multimedia (as it contains terms such as *mov, youtub, stop, video, player, game, flash, sound*) while in other cases the topic refers to page layouting (e.g., *left, width, resiz, bottom, height, size, bor*) or to the interaction with the browser (*mous, resiz, ctrl, siz; posit, screen, keyboard, focu*).

By browsing issue reports related to the matching topics, we have found several related reports for example (i) concerning multimedia content, Firefox issue # 317201 (*"Plugins do not render at all"*) and Chrome issue # 1361 *"flash chops a lot"*); (ii) concerning layouting, Firefox issue # 302151 *"horizontal scrollbars not shown with absolute positioned elements inside an overflow: auto element"* and Chrome issue # 46203 *"layout isn't stable for absolutely positioned divs with a bottom percentage. Rounding error??"*; and (iii) concerning interaction, Firefox issue # 399330 *"modal dialogs are not capturing keyboard events"*) and Chrome issue 45940 *"focus is lost when reload post form"*.

Finally, to address **RQ3**, we identified overlapping topics within the same semester. There are only two cases—also shown in Table 3 for which there are topics, in the same semester, overlap for at least 8 terms, i.e., in 2008 S2 (*movi, youtub, stop, video, player, game, plai, flash, sound*) and 2010 S1 (*left, width, resiz, bottom, height, size, border, posit, px*). By looking to cases with a lower overlap (i.e., at least 5 terms), we have found other cases of overlap, shown in Table 4. In this case, although once again the "emerging topics" discussed in Table 3 (multimedia, layouting, interaction with the browser) are still evident, other topics emerge, e.g., related to authentication (*usernam, account, login, email, authent*) or to performance issues (*cpu, task, usag, slow, hang*).

Table 4: Overlapping topics in the same semester.

Semester	Overlapping terms in the topic
2008 S2	movi; youtub; stop; video; player; game; plai; flash; sound
2008 S2	left; width; height; border; px
2008 S2	cpu; task; usag; slow; hang
2008 S2	hit; shift; tag; keyboard; focu
2008 S2	usernam; account; login; email; authent
2009 S1	youtub; video; player; plai; flash
2009 S1	width; background; bottom; size; posit
2009 S2	left; mous; bottom; posit; screen
2010 S1	left; width; resiz; bottom; height; size; border; posit; px
2010 S1	restart; visit; login; comput; websit; hang

4. CONCLUSIONS

This paper reported a study aimed at analyzing—by using an Information Retrieval technique named Latent Dirichlet Allocation—the topics discussed by developers/contributors/users in issue reports of two popular browsers, Firefox and Chrome.

The study revealed that, while for Firefox the discussion was quite heterogeneous, in some cases related to low-level issues, and in some other cases to layouting issues, for Chrome the discussion mainly related to performance issues and to handling multimedia contents, reflecting the specific purposes of this browser. Also, there is a non-negligible overlap among topics identified for the two browsers. Such an overlap is, in some cases, related to generic topics related to HTML browsers, e.g., page layouting or user interaction through keyboard or mouse. In many other cases, and especially during recent periods, there are common topics related to multimedia contents.

Work-in-progress aims at identifying further commonalities/differences between issue report discussions of the two projects, e.g., the presence of duplicate issue reports across projects (e.g., due to compatibility issues with new versions of scripting languages, or with specific plugins).

5. REFERENCES