VMware, Inc. Invention Disclosure Form

Overview

File #: C162.ID

Title*

A METHOD FOR RETRIEVING DUPLICATE REPORT IN CLOSED BUG TRACKING SYSTEM

Technology Group:

Technology Group Category*

SDDC Management

Product or Technology*

SDDC Management - Miscellaneous

Received Date

7/25/2014

vmware

VMware, Inc. Invention Disclosure Form

Disclosure

Previous Public Disclosure:

Has this invention been made known to anyone outside of VMware and not subject to an NDA?*

No

When?

How?

Anticipated Public Disclosure:

Outside of a non-disclosure agreement (NDA), is there any planned disclosure of this invention or release of a product incorporating this invention to anyone outside of VMware?

No

When?

How?

Supporting Documents:

Documents and Attachments:	
Document Name	**Subject**

vmware

VMware, Inc. Invention Disclosure Form

Third Party Interest

Development Funding:

Is the development of this invention being funded by an agency of the U.S. Government?*

No

Agency

Contract Number

Special Contract Limitations:

Are you under a duty to maintain any aspect of this invention in confidence or to assign all or any part of it to any other individual (for example, business partner), company (for example, previous employer), organization, or educational institution?*

No

Describe

Have you developed all or any part of this invention using the equipment, consulting services, or facilities of any other individual, company, organization or educational institution?*

No

Describe

Do you know of any other potential limitations to VMware's exclusive right to own or develop this invention?*

No

Describe

VMware Project:

Is the development of this invention related to a current VMware Project?*

No

Enter Name (or Codename) of Project:

Is this a project that VMware is collaborating on with a third party (or parties)?*

Enter Name of Third Parties:

Supporting Documents:

Documents and Attachments	
Document Name	**Subject**

vmware

VMware, Inc. Invention Disclosure Form

Invention Description

Problem to be Solved:

What problem does the invention solve?*

In software industry, bug tracking system is widely used in every company, helping engineers track and fix product defects. For a complex product that contains many components, there are a large number of bug reports coming in every day and in order to efficiently and rapidly fixing one new bug, engineers need spend lots of time and effort on evaluating whether this bug is a duplicated one, i.e., a report about the same bug was filed before. Besides finding exactly the same bug, engineer could also benefit from finding similar bugs in the bug tracking system, getting some clues about the new bug.
If there is an efficient and automatic way to retrieving duplicate bug report from a closed bug tracking system(like VMware bugzilla), it would significantly reduce consuming of engineering resource and enhance bug fixing efficiency. Here closed bug tracking system is like VMware bugzilla which is managed by VMware company itself engineers, engineers out of VMware can not access this bug tracking system.

Summary of the Invention:

Briefly, summarize the invention and how it solves the stated problem.*

One bug report in a tracking system often has specific structure and a commenting system where people could post their findings and discuss the bug with others. This proposed method is designed to analyze the structure information and comments of each bug and calculate similarities between different reports, by taking bugs different type information into consideration and combining it with a set of common algorism.
More specifically, this method contains four calculating factors, natural language information, person information, execution information and structured information. In order to improve the accuracy of information similarity measures, we set up reference dictionaries to achieve accurate result. The basic workflow is demonstrated bellow:
1. Based on company closed bug tracking system, and usually one kind of bug is handled by constant group persons, when bug and comment owner (person) belong to the same PCC(PCC stands for product, category and component), the person's comment should be more valuable, and then if two bug reports have same PCC and same persons who updated comments, the similarity can be higher. Then a Person Dictionary is automatically trained that represents the relevance between employees and products components.
2. Also based on company's product/components, we could define a set of keywords attached to each component and set up another dictionary: Keywords Dictionary.The reason of building keyword dictionary is that one kind of bug can always be described by a set of keywords, which either can express one class topic.
3. For different bug reports, similarity calculation is performed
3.1 Natural Language Similarity (NL-S): we process the comments for each bug report and extract the top-5 most important comments via statistics analysis such as one that have been replied/referenced more than others, one that contains many keywords related to the bug's product/components field based on the Keywords Dictionary, etc. Then we can calculate the similarity of two bug reports based on their top-5 list of comments

as well as summary description of the reports. Textual similarity calculation can be applied here.

3.2 Person Similarity (P-S): for each bug report, we can get the top-5 persons who have the most contribution to this bug report. It is also measured by statistics analysis like who has post the most number of comments, who is the owner/assignee of this bug, whose name been asked many times, etc. To calculate P-S of two bugs, we compare their top-5 list and leverage the Person Dictionary information that two persons from the same product/component are highly relevant.

3.3 Execution Similarity (E-S): for bug report discussion inside a company system, there are many software execution message been posted as comments, like Call-stack, warning/error log, source code, etc. Textual similarity calculation will be applied to get the E-S of two bugs.

3.4 Structured Similarity (S-S): different from a general online discussion system, bug tracking system provides many structured field for each bug report. Each column/field is highly related to certain product/component feature and valued ones are included in to a textual calculation to generate the structured similarities of two bugs.

4. Based on the above algorithm, those four factors will be combined together and weight for each other is trained to comply with logistic regression model by processing existing bug reports.

5. Finally, once a new bug report is coming, engineers could apply the established model and determine whether there are duplicate bug reports filed, or retrieve a list of reports that has the most relevancies to this new bug.

More detailed processing approach can be found in our attached paper.

This paper is reviewed on RADIO 2014.
All reviewer thought it might be patentable.

Here are the comments from them.

First comment:
==========

I thought that the combination of previously published techniques used to detect duplicate bug reports might be patentable. The first paragraph of section 2 seems to sum it up:

" Basic idea of the approach is as follows. First, based on
information retrieval, we calculate feature similarity between the
target bug report and each existing bug report for natural language
information, person relevance information, execution information,
and structured information respectively. Second, each feature
weight is trained based on logistic regression model. Finally,
based on trained weight, each two bug reports similarity is
determined. Figure 1 depicts the basic architecture and workflow
of our approach. "

That pipeline of methods seemed to me to be the essence of the paper's innovation.

Second comment:
==========

I think that the paper suggests an algorithm to find duplicate bugs in bugzilla and I think that the formula that he suggests there can be a patent to a serious problem.

==========
The last reviewer said:

"I thought their text-mining approach for detecting duplicate bug reports was interesting. There's a lot of interest in building machine learning pipelines currently, and putting together an appropriate pipeline for a given learning objective is one of the non-trivial tasks."

Prior Art:

How have others tried to solve the problem and in what ways have their solutions been inadequate?

So far some researches have been proposed to find the top-N similarity bugs. Some approaches adopt information-retrieval techniques to measure the similarity between bug reports using natural language information. Like "Detection of duplicate defect reports using natural language processing. In International Conference on Software Engineering (ICSE), pages 499–510, 2007"

Some of them considered execution information to retrieve duplicated bugs. Like "An Approach to Detecting Duplicate Bug Reports using Natural Language and Execution Information", in Proc. of 30th International Conference on Software Engineering (ICSE'08), Leipzig, Germany, 10 - 18 May 2008 2008"

And recently some of them extracted structured information from bugzilla non-textual fields to get the similarity between bug reports. Like "Extracting Structural
Information from Bug Reports. MSR'08, May 10-11, 2008. "

Although these approaches already provide positive effect on duplicate bug report retrieval, there is still need to improve them due to their approaches have different limitation for closed bug tracking system like VMware bugzilla system.

Supporting Documents:

Documents and Attachments	
Document Name	**Subject**
Dup-Bugzilla_IDF.docx	dup-bug
Duplicate Bug Reports Retrieval Based On VMware Bugzilla Comments Quantitative Analytics.pdf	RADIO paper for dup bug

vmware

VMware, Inc. Invention Disclosure Form

Inventors

Inventor [1]	
Full Name:	Zhao, Yimin (Hill)
Home Address:	, US
Work Email:	hillzhao@vmware.com
Citizenship:	China
Office Location:	8th Floor South Wing Tower C, Raycom InfoTech Park No. 2 Kexueyuan South Road Haidian District Beijing Beijing, Beijing 100190
Work Phone Number:	8601058746639

VMware, Inc. Invention Disclosure Form

Inventors

Inventor [2]	
Full Name:	Wang, Fangchi
Home Address:	, US
Work Email:	fangchiw@vmware.com
Citizenship:	N/A
Office Location:	Level 8, South Wing of Tower C Raycom Info Tech Park No. 2 Kexueyuan South Road, Haidian District Beijing, 100190
Work Phone Number:	650-427-5000

VMware, Inc. Invention Disclosure Form

Inventors

Inventor [3]	
Full Name:	Xie, Hongsheng
Home Address:	, US
Work Email:	hxie@vmware.com
Citizenship:	N/A
Office Location:	Level 8, South Wing of Tower C Raycom Info Tech Park No. 2 Kexueyuan South Road, Haidian District Beijing, 100190
Work Phone Number:	650-427-5000