Automatic Recovery of Impacted Method Sets from Resolved Bug Reports

Christopher Corley
University of North Alabama
REU-SLE 2010

THE UNIVERSITY OF ALABAMA ENGINEERING

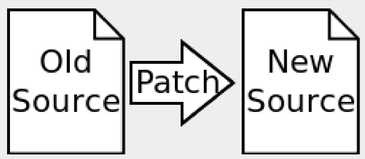
Abstract

Open source software development teams often use a revision control system, such as CVS, and a bug tracking system, such as Bugzilla, to manage software evolution tasks. Utilizing patch data retrieved from resolved Bugzilla bug reports, impacted method sets in source code can be extracted.

In the past, manual inspection of the patch was required to recover the impacted methods; however, this is a time consuming process. Our tool uses an automated approach to recover impacted method sets from resolved bug reports.

Patch File

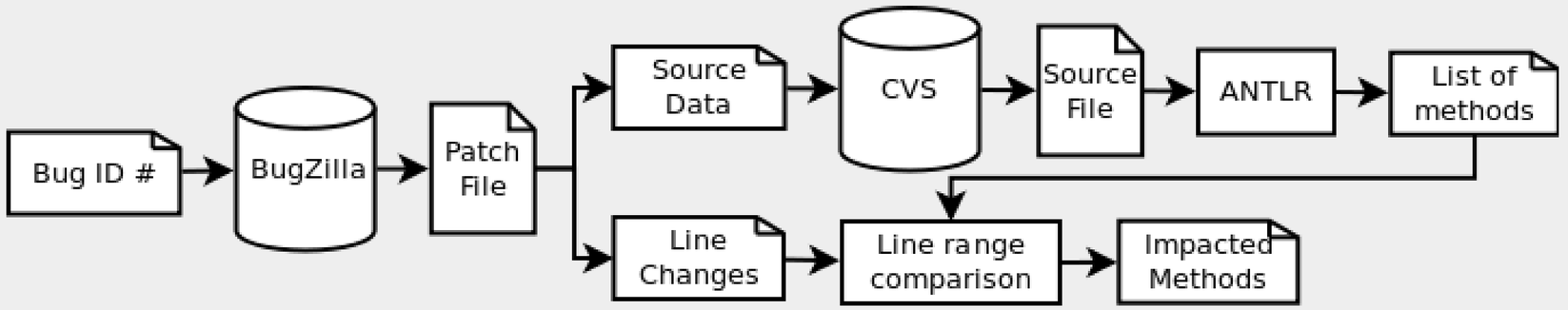
A patch file contains data which describes the differences between two sets of files. The comparison of files is done line by line. In software development, patches are used to list the changes between old source code and new source code. Applying a patch file will bring any unchanged files up to date. Bug tracking systems allow users to submit patches as solutions to bug reports.



Patches contain much usable data:

- File names of old and new sources
- Dates of last changes to each source
- Line ranges where changes occur
- Context lines that were unchanged
- Each line change with special denotation

Methodology



Experiments

In order to test the accuracy of our methodology, we performed case studies on two software projects. Our goal was to measure our automatic approach in comparison to manual inspection. Our data set consists of resolved Bugzilla bug reports for Mozilla Rhino and Eclipse. We considered 104 Rhino bug reports and 203 Eclipse bug reports.

Software	Manual Results	Automatic Results	Agreed	Disagreed	Task Mismatch
Rhino	655	866	600	242	79
Eclipse	953	1938	847	461	736
Total	1608	2804	1447	703	815

Figure : Number of impacted methods

Comparing results between manual and automatic inspection, both agreements and disagreements in which methods were changed were found. Of the 703 methods in disagreement, our automatic approach was incorrect for 24. These can be classified in one of the following categories:

- Changes outside method body range
  - Lines in the global scope of the class
  - Lines on or after the original method's ending brace (exclusive range)
- Fixes for files that do not appear in any milestone releases we inspected
- Patches embedded in bug report comments which were corrupted

Applications

Impacted method sets are utilized in many areas:

- Fault prediction models
- Bug localization
- Concept location

Acknowledgments

The work presented in this poster is supported in part by the National Science Foundation under Grant No. 0851824.

cscorley@una.edu