

《Automatically Repairing Dependency-Related Build Breakage》 Review

Paper Info

《Automatically Repairing Dependency-Related Build Breakage》 Christian Macho, Shane McIntosh, Martin Pinzger

Major Contribution

The paper introduces an approach to automatically repair Maven builds that break due to dependency-related issues. In order to demonstrate the feasibility of their work, the authors investigate 37 broken Maven builds in 23 open source Java projects manually. Based on the conclusion of the investigation, they design three strategies on 84 additional broken builds from 23 studied projects.

The major contributions of their work include two part: MLA, a tool to analyze the changes in build files, and BUILDMEDIC, an approach to automatically repair Maven builds.

In my personal view, the paper is an empirical study more than a research technical work. They find some useful results and their conclusion can guide both researchers and developers, but their tool is limited in the application.

Main Work

As explained above, the main contributions of this work are MLA and BUILDMEDIC. Based on the result extracted by MLA, the authors find the ways developers repair dependency-related build breakage. Then they propose three strategies to repair the breakage automatically, and design a tool BUILDMEDIC.

MLA extracts details from the execution log of a Maven build, and study these changes to find out which types of changes have been performed by developers. The authors group the changes into seven categories, including PROPERTY_CHANGE, VERSION_CHANGE, REPOSITORY_CHANGE, DEPENDENCY_DELETE, DEPENDENCY_ID_CHANGE, DEPENDENCY_INSERT, OTHERS. They find that changes of the category VERSION_CHANGE occurred most frequently in revisions followed by the category PROPERTY_CHANGE. 27/37 revisions can be fixed with a single change. Developers fix dependency issues by update the version in 17/37 cases. A common update is to remove the "-SNAPSHOT" suffix to promote the dependency(13/17).

Based on the results above, the authors consider three categories: VERSION_UPDATE, DEPENDENCY_DELETE, ADD_REPOSITORY. Other three kinds of patterns are complex and contribute much less to breakage, so they will be considered in their future work. In their repair tool BUILDMEDIC, they combine these three strategies. Assume that a repair plan consists of n repair strategies. Because plans starting with ADD_REPOSITORY are placed first, and ADD_REPOSITORY does not occurs in the later $(n-1)$ strategies, we have $2^{n-1} + 2^n$ repair plans. The later steps are simple and naive. They just execute the build files after repair, find out whether build succeeds, and record all possible efficient repair plans. BUILDMEDIC is able to repair 54% of dependency-related build breakage adding an average overhead of 8.6 minutes.

Their work has some applications and implications on research and on development. They figure out the common ways for developers to repair the build breakage, and this is quite useful in practice. The shining points of their work are perfect experimental design and analysis. They choose the open-source projects that fulfill some strict criteria in order to minimize false positives, and analyze the results in different way to get more detailed answers of their research questions.

Some Criticism

This paper is a good work of breakage repair for Maven build system crashes. However, it has several limitations and needs to be discussed and improved.

- The author find six patterns of breakage repair. However, they only consider three of them in their repair tool. Although their approach can handle most of cases, there are some breakages they cannot repair.
- When they generate the repair plan, they only consider to repair the breakage in n repair steps. This limits the repair applicability of tool. Some breakages need more than n steps to be repaired.
- BUILDMEDIC can generate all possible efficient repair plans, but some plans are not proper enough, and developers would never choose them to repair in practice. Hence they need to find the repair patterns of developers and only generate practical repair plans.