Striffs: Architectural Component Diagrams for Code Reviews

Muntazir Fadhel

www.mfadhel.com

HADII

Dealing With Uncertainty In Software Architecture

"... building software exists in a world of uncertainty unknown to the physical world."[1] - Martin Fowler

[1] *"Is High Quality Software Worth the Cost?"* - martinfowler.com

Code Reviews

An Essential Software Maintenance Activity

Code Review: A process in which a reviewer protects the quality of the code repository from incoming code submissions.

Reviewers *typically* have three types of artifacts at their disposal:

1. Source Code
2. Continuous Integration (CI) Artifacts
3. A description of the code submission itself

Line-Wise Diffs in Code Reviews

*A **Microscopic** View of Code*

Discussion 22 | Commits 7 | Pipelines 9 | **Changes 10**

Show all activity | 5/5 discussions resolved

Filter files

...mponents/tree_list.vue	+95 -21
...onents/file_row.vue	+33 -1
...stylesheets/pages/diff.scss	+12 -2
...s/unreleased/mr-file-list.yml	+5 -0
locale/gitlab.pot	+6 -0
package.json	+1 -1
...onents/tree_list_spec.js	+70 -2
...pts/diffs/store/utils_spec.js	+22 -0
...nts/file_row_spec.js	+36 -0
yarn.lock	+6 -1

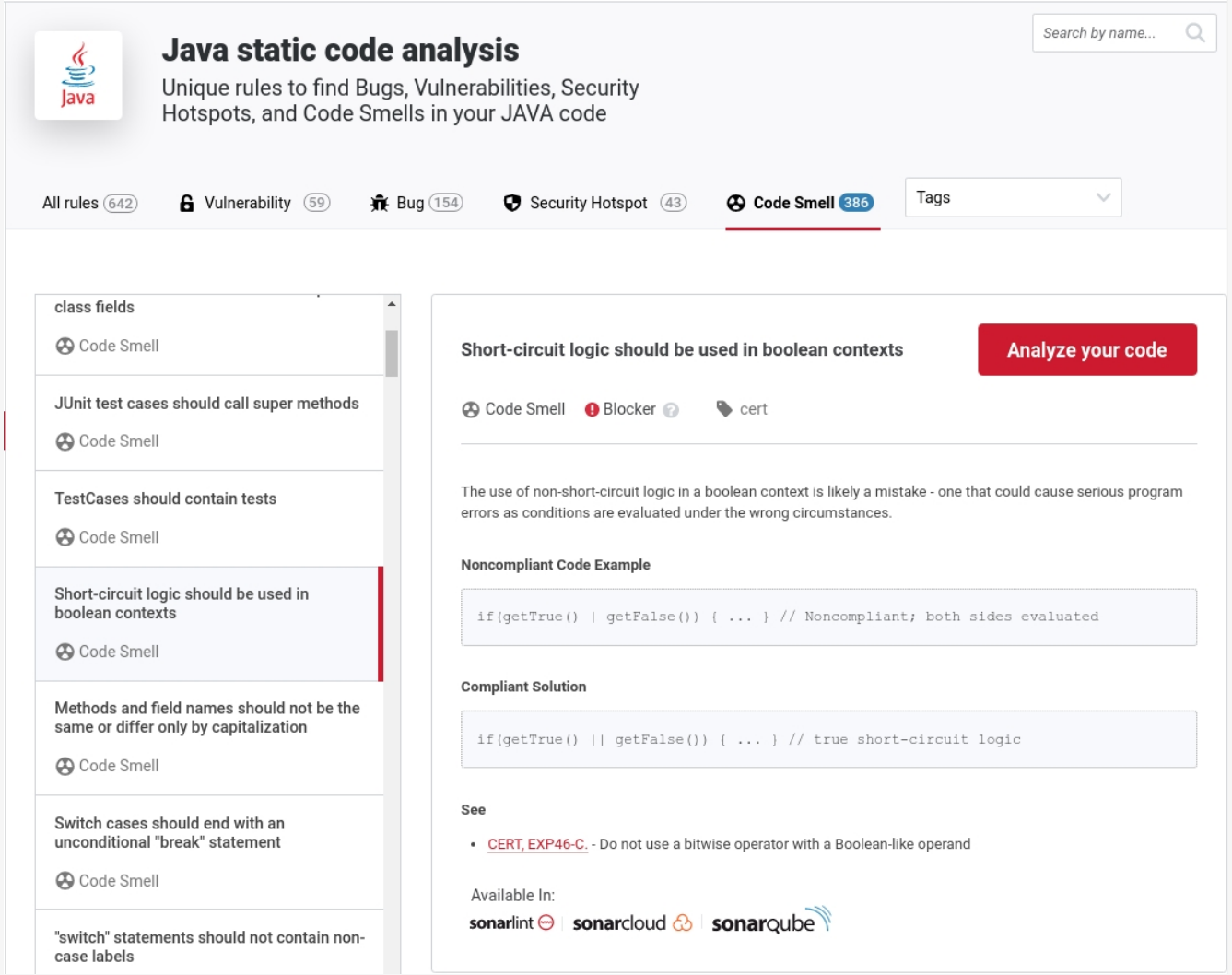
10 changed files
286 additions 28 deletions

app/assets/javascripts/diffs/components/tree_list.vue | Edit | View file @ 3b3aa28c

```
1   1    <script>
2   2    import { mapActions, mapGetters, mapState } from 'vuex';
    3  + import { TooltipDirective as Tooltip } from '@gitlab-org/gitlab-ui';
    4  + import { convertPermissionToBoolean } from '~/lib/utils/common_utils';
3   5    import Icon from '~/vue_shared/components/icon.vue';
4   6    import FileRow from '~/vue_shared/components/file_row.vue';
5   7    import FileRowStats from './file_row_stats.vue';
6   8
    9  + const treeListStorageKey = 'mr_diff_tree_list';
    10 +
7   11   export default {
    12 +   directives: {
    13 +     Tooltip,
    14 +   },
8   15     components: {
9   16       Icon,
10  17       FileRow,
11  18     },
12  19     data() {
    20 +     const treeListStored = localStorage.getItem(treeListStorageKey);
```

Static Analysis Tools in Code Reviews

Microscopic Problems of Code



Reducing The Cognitive Burden of Assessing Design

Existing Solutions

- **DesignDIFF** [3]: captures the high-level design differences through a set of 17 basic design change operators.
- **LSDiff** [2]: computes the set difference of extracted facts about code elements and their structural dependencies between old and new program versions.

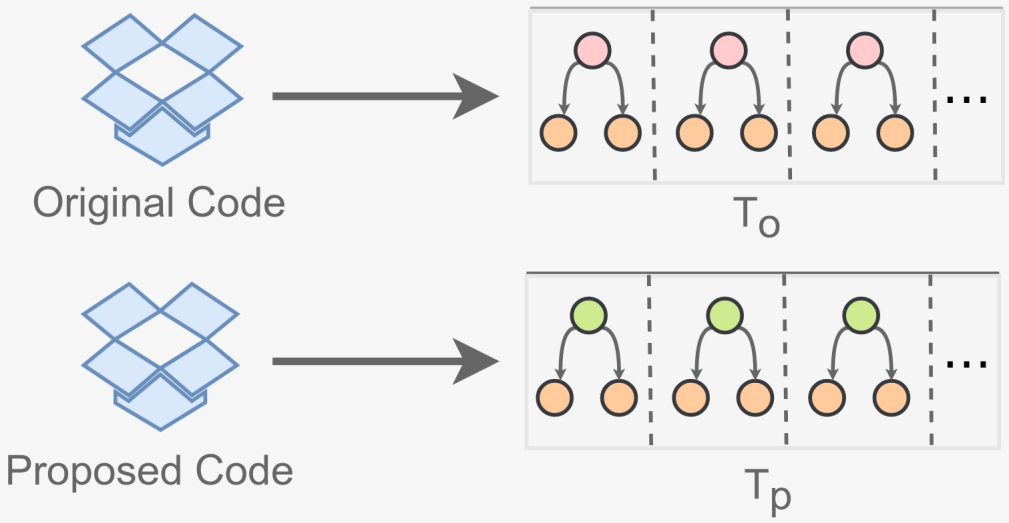
Figure 1: A screen snapshot of the LSdiff Eclipse plug-in and its features

Striff Algorithm

*Stage 1: **Parse Source Code***

Generates a list of **high-level** components (classes, interfaces, etc..) from source code[2].

- Each component is represented as a high level, filtered-down Abstract Syntax Tree (AST)



[2]Implemented using the hadii-tech/clarpse GitHub Java library.

Striff Algorithm

Stage 2: ***Generate Graphs***

For each list of components in the previous stage, generate a graph where:

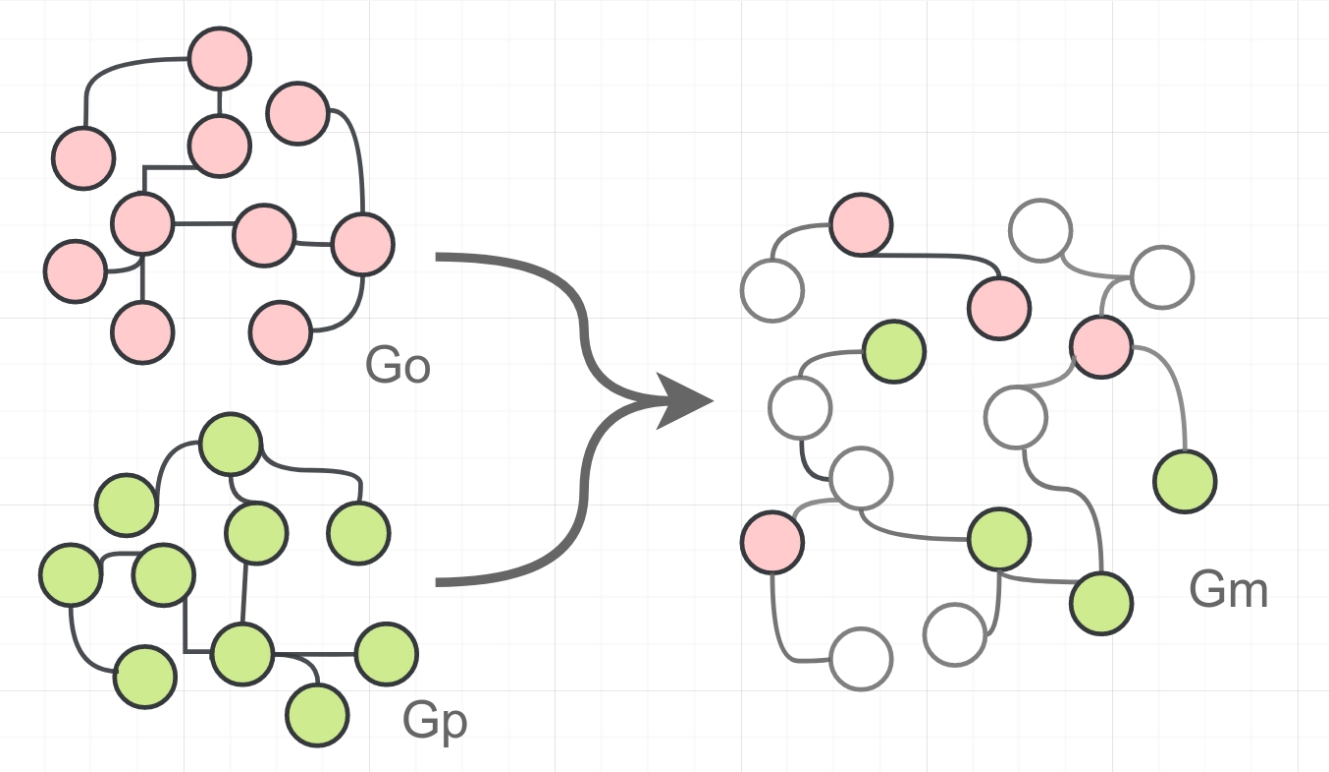
- **Vertices** correspond to high level components in the code base.
- **Edges** correspond to UML class relationships between components.
  - Edges are **weighted** according to the *importance* of the UML relationship that exists between the connected components.

Type	Realization	Generalization	Composition	Aggregation	Association
Weight	6	6	4	3	1

Striff Algorithm

Stage 3: ***Diff Graphs***

Compare components in current and proposed graph representations to form a list of all the **new** and **old** components and relationships.



Striff Algorithm

Stage 4: ***Filter Graph***

Leverage the Louvain [1] algorithm to produce subgraphs that maximize edge density with respect to other graphs.

Procedure: Recursively execute Louvain over subgraphs to eventually produce diagrams that:

1. Are small enough to be readable.
2. Contain logically related components from an Object Oriented (OO) point of view.
3. Show essential context alongside modified components.

Striff Demo

https://github.com/junit-team/junit5/pull/2581/files

Current State of Development

Currently supports **Java** and **Go**. Support for Python and C# is planned next.

Visit and subscribe at www.striffs.io to stay up to date with the latest ways to consume striff:

- **Chrome Integration**: Available on GitHub, soon to be published on the Chrome Marketplace.
- **Striff GitHub Action** (Coming Soon): Posts diagrams directly to GitHub pull requests as new commits are pushed.

References

[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre.
Fast unfolding of communities in large networks.
Journal of Statistical Mechanics: Theory and Experiment, 2008(10):P10008, Oct 2008.

[2] M. Kim and D. Notkin.
Discovering and representing systematic code changes.
In *Proceedings of the 31st International Conference on Software Engineering*, ICSE '09, page 309–319, USA, 2009. IEEE Computer Society.

[3] X. Wang, L. Xiao, K. Huang, B. Chen, Y. Zhao, and Y. Liu.
Designdiff: Continuously modeling software design difference from code revisions.
In *2020 IEEE International Conference on Software Architecture (ICSA)*, pages 179–190, 2020.