



Fixing Feature Mappings

Rahel Sundermann, Paul Bittner, Lennart Pape, Timo Kehrer, Thomas Thüm | March 27, 2026

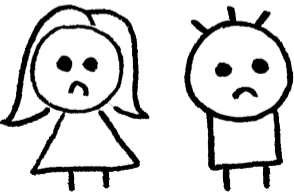
Fixing Feature Mappings - In Theory

```
def createMySandwich():  
    ...  
    #ifdef Vegetarian  
        addParmesan()  
    #endif  
    ...
```

```
def createMySandwich():  
    ...  
    #ifdef Meat  
        addParmesan()  
    #endif  
    ...
```



What kinds of mistakes are actually made in practice?



We can tell you how to work with feature mappings!
[6, 5]

We investigate keeping feature mapping correctness!
[8, 7]

We describe changes in preprocessor annotations!
[2, 4]

We just assume that they are right!
[1]

Fixing Feature Mappings - In Gimp

What happened?

- Mistake has been made in the first commit
- `#ifdef` was added in the second commit to fix the mistake

Commit message:

app: Fix localization language in Windows

```
class app/language.c:  
  ...  
  g_setenv ("LANG", language, TRUE);  
  ...
```

Commit message (239 days later):

app: fix localization of plug-ins on Linux.

```
class app/language.c:  
  ...  
  #ifdef G_OS_WIN32  
    g_setenv ("LANG", language, TRUE);  
  #endif  
  ...
```

Fixing Feature Mappings - In Godot

What happened?

- `#ifdef` was wrongly a `#ifndef` in the first commit
- `#ifndef` was changed to `#ifdef` in the second commit to fix the mistake

Commit message (90 files changed):

Create a dedicated 2D navigation server

Commit message (116 days later):

Fix flipped clipper2 ifdef

```
class ../godot_navigation_server_2d.cpp:
...
#ifndef CLIPPER2_ENABLED
    NavMeshGenerator2D:[...]parsers);
#endif
...
```

```
class ../godot_navigation_server_2d.cpp:
...
#ifdef CLIPPER2_ENABLED
    NavMeshGenerator2D:[...]parsers);
#endif
...
```

Fixing Feature Mappings - In GNU Compiler Collection (GCC)

What happened?

- Modifier + return type wrongly included in `#if`
- Moved modifier + return type outside of the `#if`

Commit message:

ada: Add SIGPROT handler for CheriBSD

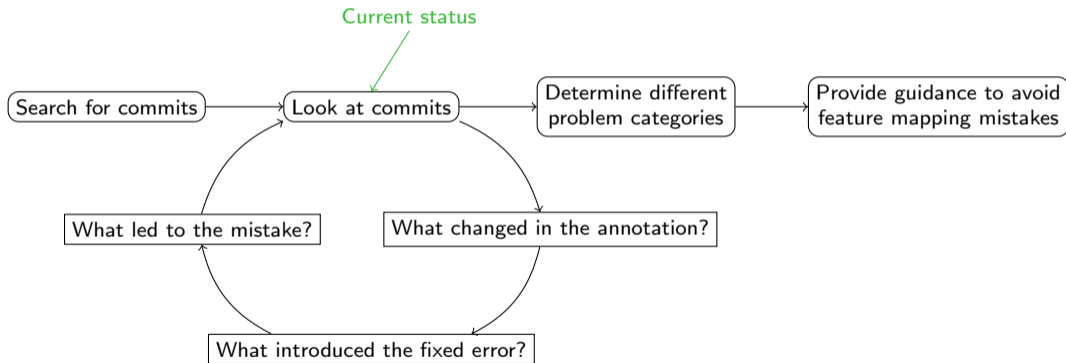
```
class gcc/ada/init.c:  
...  
#if __CHERI__  
    static void  
    __gnat_error_handler(...)  
#else  
    __gnat_error_handler(...)  
#endif  
...
```

Commit message (139 days later):

Fix compilation failure on FreeBSD

```
class ../avx512bwintrin.h:  
...  
static void  
#if __CHERI__  
    __gnat_error_handler(...)  
#else  
    __gnat_error_handler(...)  
#endif  
...
```

Methodology

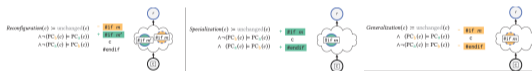


Search for commits

- Analyze 41 git repositories of real product line repositories with DiffDetective [3]



- Filter with Classification of Bittner et al. [2]



- String filter on commit message for specific keywords

Regex: `(fix)|(problem)|(issue)|(solve)|(error)|((?<!e)bug(?!=b:))`

Result: 29,596 commits

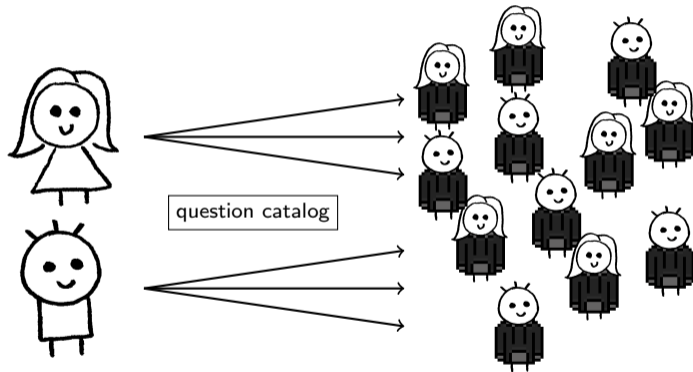
Look at commits - Checklist - First Results

- Version incompatibility** Version incompatibility
4/68
- if/ifdef mistake** if/ifdef mistake 3/68
- Mutations according to Al-Hajjaji et al. [4]**
Mutations according to Al-Hajjaji et al. [4] 35/68
- Extension of Mutations** Extension of Mutations
19/68
- Typo in feature name** Typo in feature name
6/68
- unused warning** unused warning x/68
- Wrongly included** Wrongly included 12/68

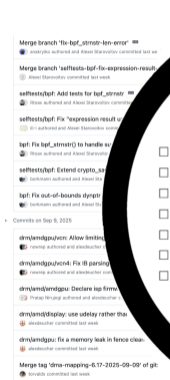
```
class modules/md/md_curl.c:  
    ...  
    #ifdef CURLSSLOPT_NO_REVOKE  
        ssl_options |= CURLSSLOPT_NO_REVOKE;  
    #endif  
    ...
```

Determine Different Problem Categories

Contact developers for more information.



Determine Different Problem Categories



- Version incompatibility
- if/ifdef mistake
- Mutations according to Al-Hajjaji et al. [4]
- Extension of Mutations
- Typo in feature name
- unused warning
- Wrongly included



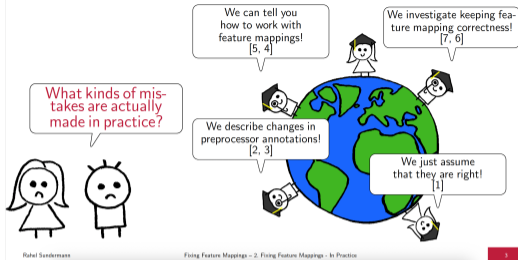
Provide Guidance to Avoid Feature Mapping Mistakes

Let us determine what kinds of mistakes need to be addressed.

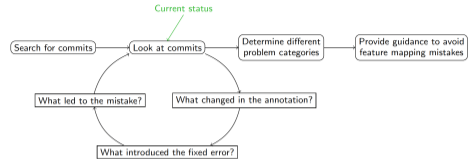


And maybe some tool support or assistance can be found to address the different problems.





Methodology



Look at commits - Checklist

- ✓ Version incompatibility
- ✗ if/ifdef mistake
- ✓ Mutations according to Al-Hajjaji et al. [4]
- ✗ Extension of Mutations
- ✗ Typo in feature name
- ✗ unused warning
- ✗ Wrongly included

```

class modules/md/md_curl.c:
...
#ifdef CURLSSLOPT_NO_REVOKE
ssl_options |= CURLSSLOPT_NO_REVOKE;
#endif
...
  
```

Provide Guidance to Avoid Feature Mapping Mistakes

Let us determine what kinds of mistakes need to be addressed.



And maybe some tool support or assistance can be found to address the different problems.



Fixing Feature Mappings

1. Fixing Feature Mappings - In Theory

2. Fixing Feature Mappings - In Practice

3. Methodology

Search for commits

Look at commits

Determine Different Problem Categories

Provide Guidance to Avoid Feature Mapping Mistakes

4. Summary

Content Overview

- [1] Paul Bittner et al. “Feature Trace Recording”. In: Athens, Greece, Aug. 2021, pp. 1007–1020. ISBN: 9781450385626. DOI: 10.1145/3468264.3468531.
- [2] Paul Maximilian Bittner et al. “Classifying Edits to Variability in Source Code”. In: Singapore, Nov. 2022, pp. 196–208. ISBN: 9781450394130. DOI: 10.1145/3540250.3549108.
- [3] Paul Maximilian Bittner et al. “Variability-Aware Differencing with DiffDetective”. In: Porto de Galinhas, Brazil, July 2024, pp. 632–636. ISBN: 9798400706585. DOI: 10.1145/3663529.3663813.
- [4] Mustafa Al-Hajjaji et al. “Mutation Operators for Preprocessor-Based Variability”. In: Salvador, Brazil, Jan. 2016, pp. 81–88. ISBN: 978-1-4503-4019-9. DOI: 10.1145/2866614.2866626.
- [5] Flávio Medeiros et al. “Discipline Matters: Refactoring of Preprocessor Directives in the #ifdef Hell”. In: 44.5 (2018), pp. 453–469. DOI: 10.1109/TSE.2017.2688333.
- [6] Gabriela Karoline Michelon et al. “Mining Feature Revisions in Highly-Configurable Software Systems”. In: Montreal, QC, Canada, Oct. 2020, pp. 74–78. ISBN: 9781450375702. DOI: 10.1145/3382026.3425776.
- [7] Leonardo Passos et al. “Coevolution of Variability Models and Related Artifacts: A Case Study from the Linux Kernel”. In: Tokyo, Japan, Aug. 2013, pp. 91–100. ISBN: 978-1-4503-1968-3. DOI: 10.1145/2491627.2491628.
- [8] Christoph Seidl, Florian Heidenreich, and Uwe Aßmann. “Co-Evolution of Models and Feature Mapping in Software Product Lines”. In: Salvador, Brazil, 2012, pp. 76–85. DOI: 10.1145/2362536.2362550.