

Vaci:

Variation Analysis of Context-Sharing Identifiers with Code Clones

Toshihiro Kamiya

Service Innovation Management Research Team

National Institute of Advanced Industrial Science and Technology, Japan

t-kamiya@aist.go.jp

Naming, an old and new problem

Everyday developers give names to types, functions, variables, enums, ...

Adequate (descriptive, consistent) name is important.

- contributes to maintainability and understandability
- → naming convention

However, it's not always possible.

- Changelogs contain “rename”, “replace”...

Some tools have been proposed to support naming / renaming.

Tool support

- Support for renaming
 - Name **restructuring** tool (overhauling names in a product)
 - B. Caprile and P. Tonella, “Restructuring Program Identifier Names”, *Proc. IEEE ICSM’00*, pp. 97-107, Oct. 2000.
 - Rename **refactoring** menu (such as Eclipse IDE)
 - “@deprecated” JavaDoc, **annotation** (Java)
 - Source **code converter** (Python 2.5 → 3.0)
- Support for checking name
 - Consistency checker (name \Leftrightarrow type)
 - F. Deißeböck and M. Pizka, “Concise and Consistent Naming”, *Proc. IWPC 2005*, pp. 97-106. May 2005.
 - D. Lawrie, et al, “Syntactic Identifier Conciseness and Consistency”, *Proc. IEEE SCAM 2006*, pp. 139-148, Sep. 2006.
 - Spell checker (such as Eclipse IDE)

Tool Vaci

The tool collects and analyzes relations between each name and its contexts (i.e., code fragments where the name appears).

- Finding Variation of Names: to check consistency among the names that are used in the similar contexts.
- Tracking Change of Names: to track how names changes between versions.

Snapshot #1, finding variation of names

The viewer works as Eclipse plugin.

It reads: the name “metricFile” and “cloneSetMetricFile” are used in the similar context.

I should rename the former name when I coded the latter...

The screenshot shows the Eclipse IDE with the following components:

- Editor:** Displays the code for `MainWindow.java`. A yellow circle highlights the `add_file_metrics()` method. A line of code is highlighted in blue: `String metricFile = cloneDataFile + ".fn.tmp"; //SNON-NLS-1$`.
- Package Explorer:** Shows the project structure on the left, including packages like `gemx`, `gemx.dia.ogs`, and `resources`.
- Problems View:** At the bottom, a list of search results for the name `metricFile` is shown. The results include:
 - `a. metricFile : abbreviated`
 - `metricFile`
 - `cloneSetMetricFile`
 - `a. getSelection : abbreviated`
 - `getSelection`
 - `getSelection: index`
 - `a. indexOf : abbreviated`
 - `indexOf`
 - `lastIndexOf`
 - `a. selectFilesByCloneClassID : abbreviated`
- Vaci Search Dialog:** A dialog box titled "Vaci search" is open, showing filtering options:
 - nearly-identical
 - 123 distinct-numbers
 - short-name
 - same-prefix
 - x same-postfix
 - Cc distinct-c
 - a. abbreviated
 - minority
 - othersThe "Find:" field is empty, and "Ignore" and "Regex" checkboxes are also present.

Snapshot #2, tracking change of names

The analysis result is rendered into html document, including graphics (Scalable Vector Graphics).

It reads: the identifiers named "HWMODE_A" is renamed to "IEEE80211_BAND_5GHZ".

Also, some identifiers named "MODE_IEEE80211A" is renamed to "IEEE80211_BAND_5HZ." The others is renamed to "AR5K_MODE_11A".

Mozilla Firefox

ファイル(E) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(I) ヘルプ(H)

file:///C:/experiments/linuxkernel_versions/translation_map_17.html

よく見るページ Firefox を使ってみよう 最新ニュース

#930 #935 #980 #1023 #1026 #1030 #1033 #1035 #1045 #1051
 #1055 #1060 #1061 #1065 #1066 #1067 #1070 #1079 #1082 #1120
 #1124 #1138 #1142 #1145 #1149 #1161 #1165 #1170 #1172 #1183
 #1185 #1186 #1190 #1196 #1204 #1206 #1207 #1213 #1215 #1219
 #1220 #1227 #1228 #1230 #1232 #1237 #1238 #1241 #1256 #1258
 #1261 #1263 #1267 #1280 #1285 #1288 #1289 #1314 #1360 #1364
 #1372 #1382 #1408 #1420 #1461 #1465 #1469 #1475 #1481 #1484
 #1487 #1584 #1585 #1594 #1595 #1596 #1597 #1622 #1639 #1648
 #1649 #1655 #1678 #1695 #1704 #1716 #1745 #1746

Type MtoN,single

#8 #256 #257 #420 #421 #422 #423 #519 #623 #800
 #809 #975 #1001 #1017 #1029 #1113 #1376 #1706 #1707 #1711
 #1742

Type MtoN,multiple

Translation map #17

Graph

```

graph LR
    HWMODE_A --> IEEE80211_BAND_5GHZ
    MODE_IEEE80211A --> AR5K_MODE_11A
  
```

→ merge nodes

Edges (name changed)

HWMODE_A → IEEE80211_BAND_5GHZ

c:\experiments\linuxkernel_versions\linux-2.6.25\drivers\net\wireless\rt2x00
 \rt61pci.c:430 → c:\experiments\linuxkernel_versions\linux-2.6.26\drivers\net\wireless
 \rt2x00\rt61pci.c:518

Basic idea:

“Identifiers share a context”

The cases where two identifiers of distinct names appear in the identical code fragments.


identifier

File.PathSeparator

identifier

File.PathSeparatorChar

context

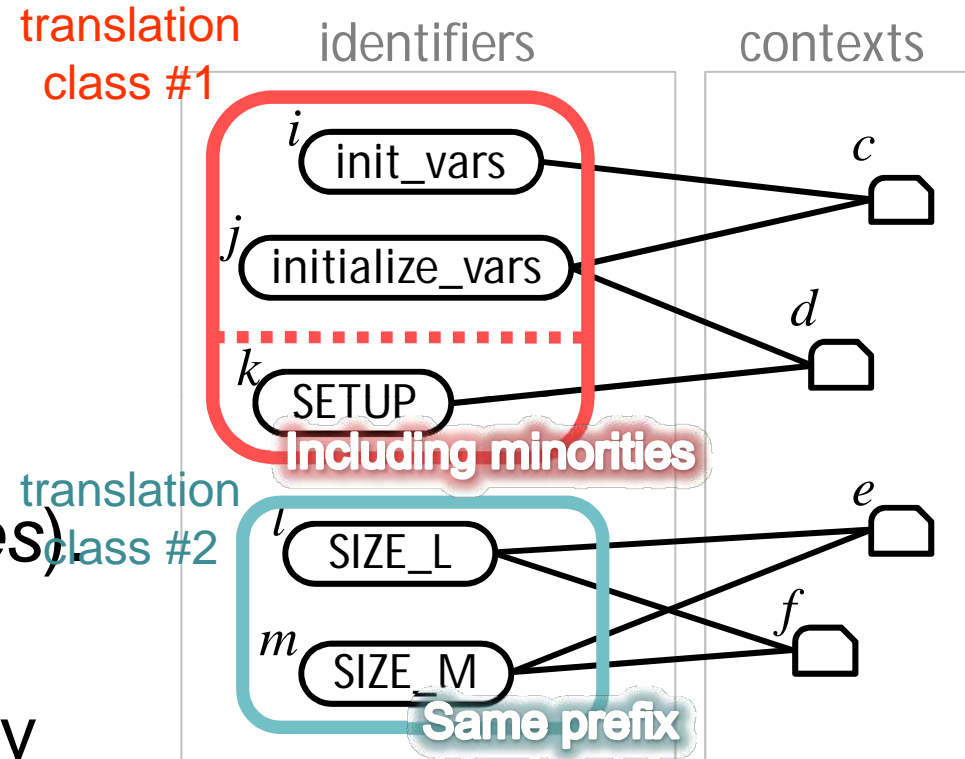
```
int pos = fpath.indexOf(  );
if (pos >= 0)
    fpath = fpath.substring(0, pos);
return fpath;
```

```
....
String getFirstPartOfPath(String fpath) {
    int pos = fpath.indexOf(File.pathSeparator);
    if (pos >= 0)
        fpath = fpath.substring(0, pos);
    return fpath;
}
....
```

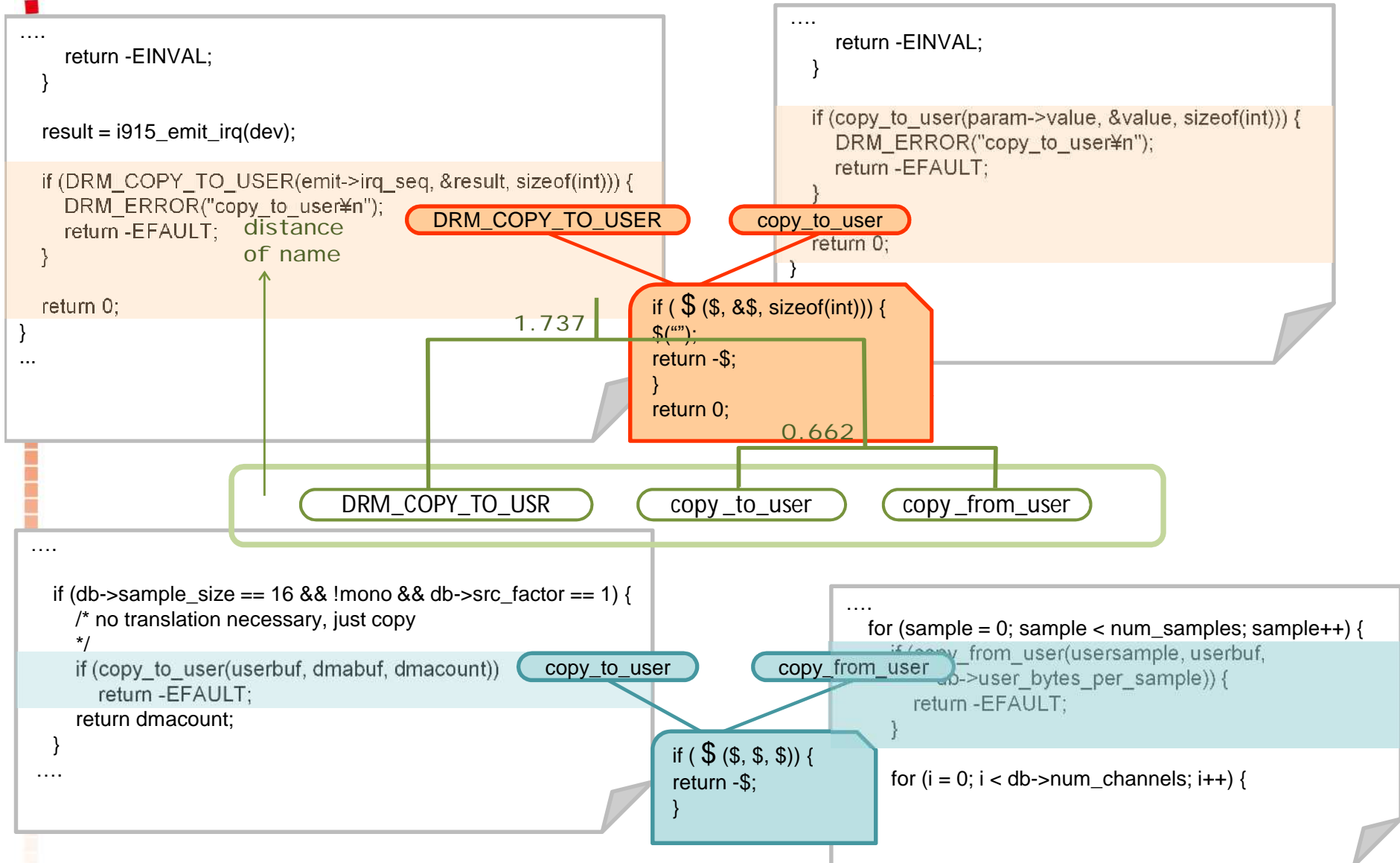
```
....
String split1stPart(String fpath) {
    int pos = fpath.indexOf(File.pathSeparatorChar);
    if (pos >= 0)
        fpath = fpath.substring(0, pos);
    return fpath;
}
....
```

Finding Variation of Names

1. Detect “context-sharing” identifier pairs.
2. Generate their transitive closures (*translation classes*)
3. Classify each translation class by the names of identifiers in the class

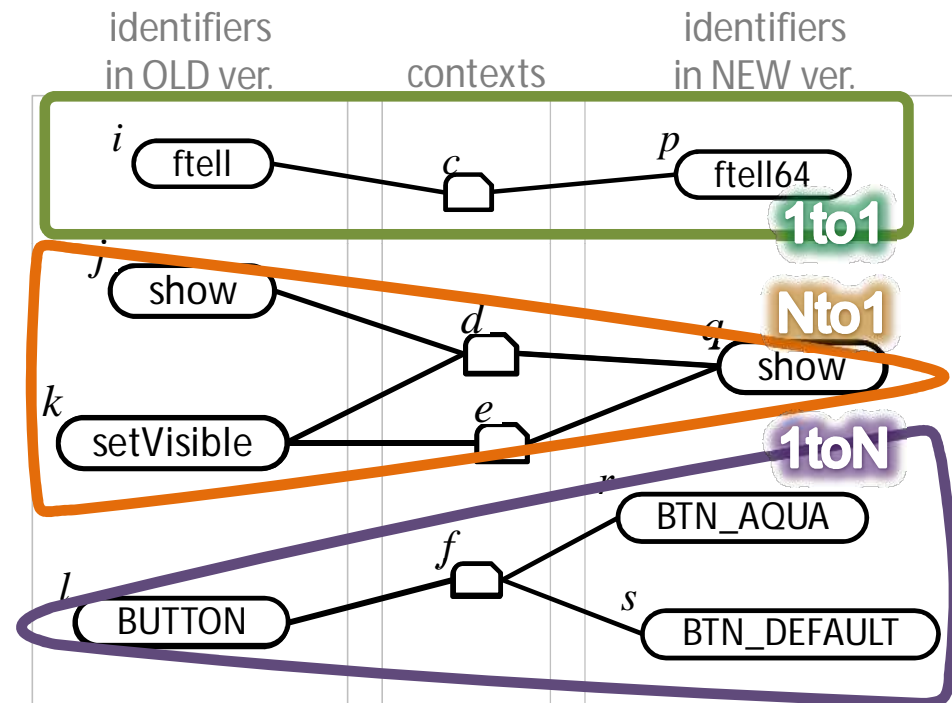


Example from Linux Kernel



Tracking Change of Names

1. Detect context-sharing identifiers **between two versions.**
2. Generate their transitive closures (*translation maps*).
3. Classify each translation class by **number of identifiers**



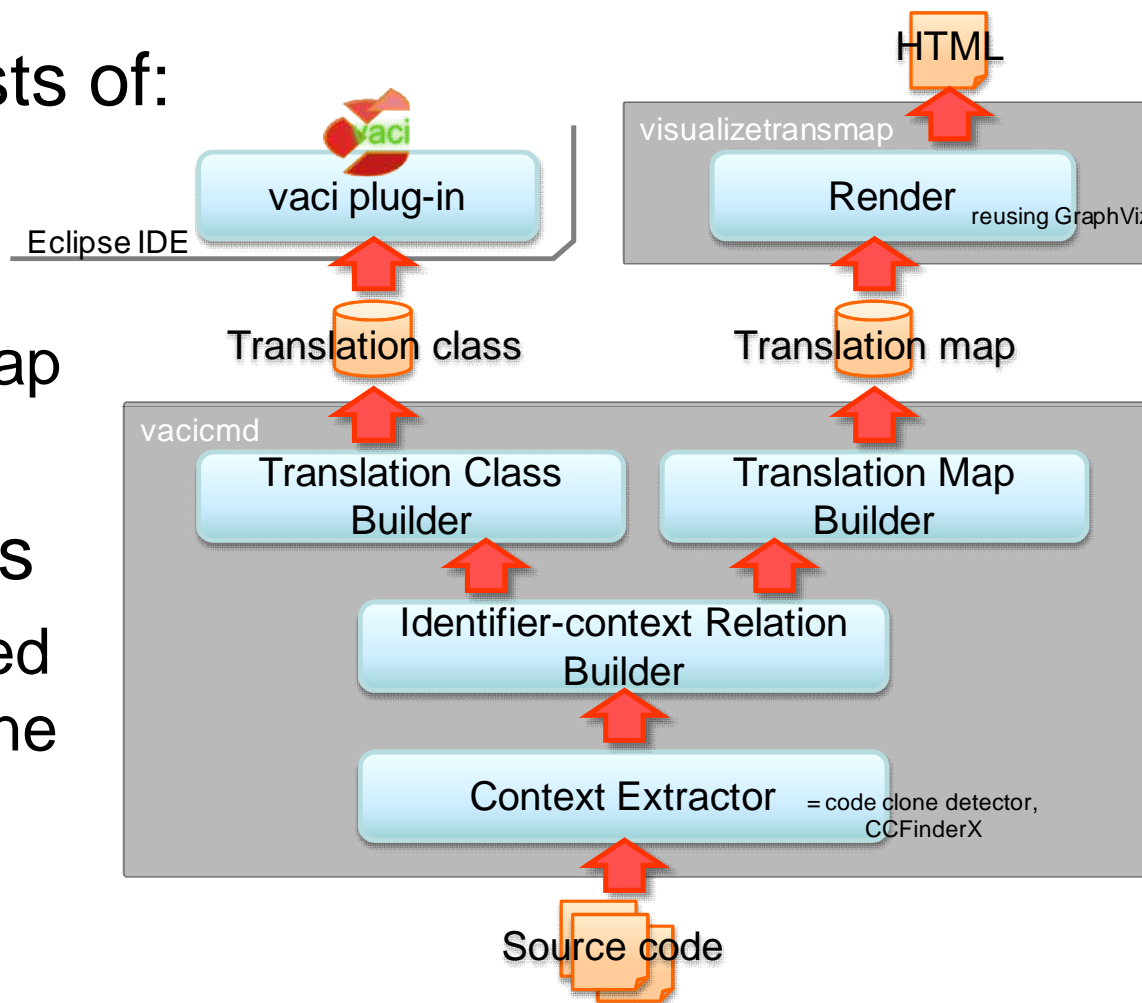
Implementation

Tool Vaci consists of:

- vacicmd
- vaci plug-in
- visualizetransmap

Total 5k lines of
Python/C++ lines

- Except for reused code from a clone detection tool “CCFinderX”



www.ccfinder.net/vaci.html

The latest version of Vaci (used in this presentation) will be available soon.

Other useful materials

- Observations, theories
 - Derek M. Jones, The New C Standard, Sentence 792 (identifiers). Taken from <http://www.coding-guidelines.com/cbook/sent792.pdf>, 2008/09/11.
- Keyword → Code
 - Greg Little and Robert C. Miller, “Keyword Programming in Java”, Proc. the 22th IEEE/ACM Int’l Conf. Automated Software Engineering, pp. 84-93, 2007