

Match algorithm description

Bruno Crotman

18/04/2020

1 Introduction

The goal is to analyze two different versions of a code (an old version and a new version) and decide if

- each PMD alert generated for the old version is *not fixed* or *fixed* in the new version. When an alert is *not fixed*, it means that it remains in the new version. When an alert is *fixed*, it means that it does not exist anymore in the new version.
- each PMD alert generated for the new version is *not fixed* or *new*. When an alert is *not fixed* it means that the same alert was identified in the old version. When an alert is *new*, it means that the same alert was not identified in the old version.

The alerts identified as *not fixed* are the same in both new and old versions. And the intersection between the *fixed* alerts, the *new* alerts and the *not fixed* alerts is empty.

In order to decide if an alert is *not fixed*, *fixed* or *new*, I must identify if an alert in the old version is equivalent to an alert in the new version.

I describe here two algorithms. The first is a naive algorithm based on matches by lines of code, it was the first to be implemented. The second is a more sophisticated algorithm, based on matches by blocks of code. The second one is not implemented yet.

2 Matches by line of code

In this first algorithm, I match the lines of code of the old version with the lines of code of the new version using information from the output of git's diff command. When an alert of the same kind occurs in both matched lines, this alert is declared *not fixed*. The alerts that occur in a not matched line of the old version is declared *fixed* and

The old version, with the alerts generated by PMI, is the following code

```
/* 1-
/* 2-
/* 3-
/* 4-
/* 5-
/* 6-
/* 7-
/* 8-
/* 9-
/* 10-
/* 11-
/* 12-
/* 13-
/* 14-
/* 15-
/* 16-
/* 17-
/* 18-
/* 19-
/* 20-
/* 21-
/* 22-
/* 23-
/* 24-UnusedImports
/* 25-

/**
 * Copyright (C) 2007 Yusuke Yamamoto
 * Copyright (C) 2011 Twitter, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package twitter4j;

import twitter4j.auth.Authorization;
import twitter4j.conf.Configuration;

import java.util.*;
import java.util.concurrent.ConcurrentHashMap;
```

```

/* 26-
/* 27-
/* 28-
/* 29-
/* 30-
/* 31-
/* 32-
/* 33-
/* 34-
/* 35-
/* 36-UnusedPrivateField
/* 37-
/* 38-
/* 39-
/* 40-
/* 41-
/* 42-
/* 43-
/* 44-
/* 45-
/* 46-
/* 47-
/* 48-
/* 49-
/* 50-
/* 51-
/* 52-
/* 53-
/* 54-
/* 55-
/* 56-
/* 57-
/* 58-
/* 59-
/* 60-
/* 61-
/* 62-
/* 63-
/* 64-
/* 65-
/* 66-
/* 67-
/* 68-
/* 69-
/* 70-
/* 71-
/* 72-
/* 73-
/* 74-
/* 75-
/* 76-
/* 77-
/* 78-OptimizableToArrayCall
/* 79-
/* 80-
/* 81-
/* 82-
/* 83-
/* 84-
/* 85-
/* 86-
/* 87-
/* 88-
/* 89-
/* 90-
/* 91-
/* 92-FormalParameterNamingConventions
/* 93-FormalParameterNamingConventions(2)
/* 94-FormalParameterNamingConventions(2)
/* 95-
/* 96-
/* 97-
/* 98-
/* 99-
/* 100-
/* 101-
/* 102-
/* 103-
/* 104-
/* 105-
/* 106-
/* 107-
/* 108-
/* 109-
/* 110-
/* 111-
/* 112-
/* 113-
/* 114-
/* 115-
/* 116-OptimizableToArrayCall
/* 117-
/* 118-
/* 119-
/* 120-

/**
 * A Java representation of the <a href="https://dev.twitter.com/docs/api">Twitter REST API</a><br>
 * This class is thread safe and can be cached/re-used and used concurrently.<br>
 * Currently this class is not carefully designed to be extended. It is suggested to extend this class only for mock test...
 */
 * @author Yusuke Yamamoto - yusuke at mac.com
 */
class TwitterImpl extends TwitterBaseImpl implements Twitter {
    private static final long serialVersionUID = 9170943084096085770L;
    private static final Logger logger = Logger.getLogger(TwitterBaseImpl.class);

    /**package*/
    TwitterImpl(Configuration conf, Authorization auth) {
        super(conf, auth);
        INCLUDE_MY_RETWEET = new HttpParameter("include_my_retweet", conf.isIncludeMyRetweetEnabled());
        if (implicitParamsMap.containsKey(conf)) {
            this.IMPLICIT_PARAMS = implicitParamsMap.get(conf);
            this.IMPLICIT_PARAMS_STR = implicitParamsStrMap.get(conf);
        } else {
            String implicitParamsStr = conf.isIncludeEntitiesEnabled() ? "include_entities=" + true : "";
            boolean contributorsEnabled = conf.getContributingTo() != -1L;
            if (contributorsEnabled) {
                if (!"".equals(implicitParamsStr)) {
                    implicitParamsStr += "&";
                }
                implicitParamsStr += "contributingto=" + conf.getContributingTo();
            }

            if (conf.isTweetModeExtended()) {
                if (!"".equals(implicitParamsStr)) {
                    implicitParamsStr += "&";
                }
                implicitParamsStr += "tweet_mode=extended";
            }

            List<HttpParameter> params = new ArrayList<HttpParameter>(3);
            if (conf.isIncludeEntitiesEnabled()) {
                params.add(new HttpParameter("include_entities", "true"));
            }
            if (contributorsEnabled) {
                params.add(new HttpParameter("contributingto", conf.getContributingTo()));
            }
            if (conf.isTrimUserEnabled()) {
                params.add(new HttpParameter("trim_user", "1"));
            }
            if (conf.isIncludeExtAltTextEnabled()) {
                params.add(new HttpParameter("include_ext_alt_text", "true"));
            }
            if (conf.isTweetModeExtended()) {
                params.add(new HttpParameter("tweet_mode", "extended"));
            }
            HttpParameter[] implicitParams = params.toArray(new HttpParameter[params.size()]);

            // implicitParamsMap.containsKey() is evaluated in the above if clause.
            // thus implicitParamsStrMap needs to be initialized first
            implicitParamsStrMap.putIfAbsent(conf, implicitParamsStr);
            implicitParamsMap.putIfAbsent(conf, implicitParams);
        }
        this.IMPLICIT_PARAMS = implicitParams;
        this.IMPLICIT_PARAMS_STR = implicitParamsStr;
    }

    @Override
    public AccountSettings updateAccountSettings(Integer trend_locationWoeid,
        Boolean sleep_timeEnabled, String start_sleepTime,
        String end_sleepTime, String time_zone, String lang)
        throws TwitterException {
        List<HttpParameter> profile = new ArrayList<HttpParameter>(6);
        if (trend_locationWoeid != null) {
            profile.add(new HttpParameter("trend_location_woeid", trend_locationWoeid));
        }
        if (sleep_timeEnabled != null) {
            profile.add(new HttpParameter("sleep_time_enabled", sleep_timeEnabled.toString()));
        }
        if (start_sleepTime != null) {
            profile.add(new HttpParameter("start_sleep_time", start_sleepTime));
        }
        if (end_sleepTime != null) {
            profile.add(new HttpParameter("end_sleep_time", end_sleepTime));
        }
        if (time_zone != null) {
            profile.add(new HttpParameter("time_zone", time_zone));
        }
        if (lang != null) {
            profile.add(new HttpParameter("lang", lang));
        }
        return factory.createAccountSettings(post(conf.getRestBaseUrl() + "account/settings.json",
            profile.toArray(new HttpParameter[profile.size()]));
    }
}

```

The new version is the following code:

- line 24 of the old version “import java.util.concurrent.ConcurrentHashMap”, that was a “Unused Import”, was removed.
- line 92 of the old version (91 of the new version) was fixed by changing the name of the parameter
- line 119 was included in the new version, with an unused private field.

```

/* 1-
/* 2-
/* 3-
/* 4-
/* 5-
/* 6-
/* 7-
/* 8-
/* 9-
/* 10-
/* 11-
/* 12-
/* 13-
/* 14-
/* 15-
/* 16-
/* 17-
/* 18-
/* 19-
/* 20-
/* 21-
/* 22-
/* 23-
/* 24-
/* 25-
/* 26-
/* 27-
/* 28-
/* 29-
/* 30-
/* 31-
/* 32-
/* 33-
/* 34-
/* 35-UnusedPrivateField
/* 36-
/* 37-
/* 38-
/* 39-
/* 40-
/* 41-
/* 42-
/* 43-
/* 44-
/* 45-
/* 46-
/* 47-
/* 48-
/* 49-
/* 50-
/* 51-
/* 52-
/* 53-
/* 54-
/* 55-
/* 56-
/* 57-
/* 58-
/* 59-
/* 60-
/* 61-
/* 62-
/* 63-
/* 64-
/* 65-
/* 66-
/* 67-
/* 68-
/* 69-
/* 70-
/* 71-
/* 72-
/* 73-
/* 74-
/* 75-
/* 76-
/* 77-OptimizableToArrayCall
/* 78-
/* 79-
/* 80-
/* 81-
/* 82-
/* 83-
/* 84-
/* 85-
/* 86-
/* 87-
/* 88-
/* 89-
/* 90-
/* 91-
/* 92-FormalParameterNamingConventions(2)
/* 93-FormalParameterNamingConventions(2)
/* 94-
/* 95-
/* 96-
/* 97-
/* 98-
/* 99-
/* 100-
/* 101-
/* 102-
/* 103-
/* 104-
/* 105-
/* 106-
/* 107-

/**
 * Copyright (C) 2007 Yusuke Yamamoto
 * Copyright (C) 2011 Twitter, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package twitter4j;

import twitter4j.auth.Authorization;
import twitter4j.conf.Configuration;

import java.util.*;

/**
 * A Java representation of the <a href="https://dev.twitter.com/docs/api">Twitter REST API</a><br>
 * This class is thread safe and can be cached/re-used and used concurrently.<br>
 * Currently this class is not carefully designed to be extended. It is suggested to extend this class only for mock testi...
 *
 * @author Yusuke Yamamoto - yusuke at mac.com
 */
class TwitterImpl extends TwitterBaseImpl implements Twitter {
    private static final long serialVersionUID = 9170943084096085770L;
    private static final Logger logger = Logger.getLogger(TwitterBaseImpl.class);

    /*package*/
    TwitterImpl(Configuration conf, Authorization auth) {
        super(conf, auth);
        INCLUDE_MY_RETWEET = new HttpParameter("include_my_retweet", conf.isIncludeMyRetweetEnabled());
        if (implicitParamsMap.containsKey(conf)) {
            this.IMPLICIT_PARAMS = implicitParamsMap.get(conf);
            this.IMPLICIT_PARAMS_STR = implicitParamsStrMap.get(conf);
        } else {
            String implicitParamsStr = conf.isIncludeEntitiesEnabled() ? "include_entities=" + true : "";
            boolean contributorsEnabled = conf.getContributingTo() != -1L;
            if (contributorsEnabled) {
                if (!"".equals(implicitParamsStr)) {
                    implicitParamsStr += "&";
                }
                implicitParamsStr += "contributingto=" + conf.getContributingTo();
            }

            if (conf.isTweetModeExtended()) {
                if (!"".equals(implicitParamsStr)) {
                    implicitParamsStr += "&";
                }
                implicitParamsStr += "tweet_mode=extended";
            }

            List<HttpParameter> params = new ArrayList<HttpParameter>(3);
            if (conf.isIncludeEntitiesEnabled()) {
                params.add(new HttpParameter("include_entities", "true"));
            }
            if (contributorsEnabled) {
                params.add(new HttpParameter("contributingto", conf.getContributingTo()));
            }
            if (conf.isTrimUserEnabled()) {
                params.add(new HttpParameter("trim_user", "1"));
            }
            if (conf.isIncludeExtAltTextEnabled()) {
                params.add(new HttpParameter("include_ext_alt_text", "true"));
            }
            if (conf.isTweetModeExtended()) {
                params.add(new HttpParameter("tweet_mode", "extended"));
            }
            HttpParameter[] implicitParams = params.toArray(new HttpParameter[params.size()]);

            // implicitParamsMap.containsKey() is evaluated in the above if clause.
            // thus implicitParamsStrMap needs to be initialized first
            implicitParamsStrMap.putIfAbsent(conf, implicitParamsStr);
            implicitParamsMap.putIfAbsent(conf, implicitParams);
        }
        this.IMPLICIT_PARAMS = implicitParams;
        this.IMPLICIT_PARAMS_STR = implicitParamsStr;
    }

    @Override
    public AccountSettings updateAccountSettings(Integer trendlocationWooid,
        Boolean sleep_timeEnabled, String start_sleepTime,
        String end_sleepTime, String time_zone, String lang)
        throws TwitterException {
        List<HttpParameter> profile = new ArrayList<HttpParameter>(6);
        if (trendlocationWooid != null) {
            profile.add(new HttpParameter("trend_location_wooid", trendlocationWooid));
        }
        if (sleep_timeEnabled != null) {
            profile.add(new HttpParameter("sleep_time_enabled", sleep_timeEnabled.toString()));
        }
        if (start_sleepTime != null) {
            profile.add(new HttpParameter("start_sleep_time", start_sleepTime));
        }
        if (end_sleepTime != null) {
            profile.add(new HttpParameter("end_sleep_time", end_sleepTime));
        }
    }
}

```

```

/*108-          */      if (time_zone != null) {
/*109-          */      profile.add(new HttpParameter("time_zone", time_zone));
/*110-          */      }
/*111-          */      if (lang != null) {
/*112-          */      profile.add(new HttpParameter("lang", lang));
/*113-          */      }
/*114-          */      return factory.createAccountSettings(post(conf.getRestBaseUrl() + "account/settings.json"
/*115-OptimizableToArrayCall */      , profile.toArray(new HttpParameter[profile.size()]));
/*116-          */      }
/*117-          */      }
/*118-          */      }
/*119-UnusedPrivateField */      private int not_used = 0;
/*120-          */      }
/*121-          */      }

```

The result of the git diff operation upon this two versions is this:

```

5 4 j/match_algorithm_description/{old => new}/code.java

diff --git a/j/match_algorithm_description/old/code.java b/j/match_algorithm_description/new/code.java
index 33a26b4..ed77b15 100644
--- a/j/match_algorithm_description/old/code.java
+++ b/j/match_algorithm_description/new/code.java
@@ -24,23,0 @@ import java.util.*;
-import java.util.concurrent.ConcurrentHashMap;
@@ -92,91 @@ class TwitterImpl extends TwitterBaseImpl implements Twitter {
- public AccountSettings updateAccountSettings(Integer trend_locationWoeid,
+ public AccountSettings updateAccountSettings(Integer trendlocationWoeid,
@@ -97,2 +96,2 @@ class TwitterImpl extends TwitterBaseImpl implements Twitter {
- if (trend_locationWoeid != null) {
+ if (trendlocationWoeid != null) {
- profile.add(new HttpParameter("trend_location_woeid", trend_locationWoeid));
+ profile.add(new HttpParameter("trend_location_woeid", trendlocationWoeid));
+ if (trendlocationWoeid != null) {
+ profile.add(new HttpParameter("trend_location_woeid", trendlocationWoeid));
@@ -119,0 +119,2 @@ class TwitterImpl extends TwitterBaseImpl implements Twitter {
+ private int not_used = 0;
+

```

Using this information from git's diff, it's possible to create a map between the lines of the old version and the new version:

```

examples_crossed$lines_map[[1]] %>%
  select(old = map_remove, new = map_add) %>%
  gt() %>%
  tab_header(title = "Lines map", subtitle = "Map between lines in the old version and in te new version")

```

Lines map

Map between lines in the old version and in te new version

old	new
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	NA
25	24
26	25
27	26
28	27
29	28
30	29
31	30
32	31
33	32
34	33
35	34
36	35
37	36
38	37
39	38
40	39
41	40
42	41
43	42
44	43
45	44
46	45
47	46
48	47
49	48
50	49
51	50
52	51
53	52

54	53
55	54
56	55
57	56
58	57
59	58
60	59
61	60
62	61
63	62
64	63
65	64
66	65
67	66
68	67
69	68
70	69
71	70
72	71
73	72
74	73
75	74
76	75
77	76
78	77
79	78
80	79
81	80
82	81
83	82
84	83
85	84
86	85
87	86
88	87
89	88
90	89
91	90
NA	91
92	NA
93	92
94	93
95	94
96	95
NA	96
NA	97
97	NA
98	NA
99	98
100	99
101	100
102	101
103	102
104	103
105	104
106	105
107	106
108	107
109	108
110	109
111	110
112	111
113	112
114	113
115	114
116	115
117	116
118	117
119	118
NA	119
NA	120
120	121

Now we can connect the to versions and plugin the alerts

```

/* 1-
/* 2-
/* 3-
/* 4-
/* 5-
/* 6-
/* 7-
/* 8-
/* 9-
/* 10-
/* 11-
/* 12-
/* 13-
/* 14-
/* 15-
/* 16-
/* 17-
/* 18-
/* 19-
/* 20-
/* 21-
/* 22-
/* 23-
/* 24-UnusedImports
/* 25-
/* 26-
/* 27-
/* 28-
/* 29-

/**
 * Copyright (C) 2007 Yusuke Yamamoto
 * Copyright (C) 2011 Twitter, Inc.
 *
 * Licensed under the Apache License, Version 5-
 * you may not use this file except in compli 6-
 * You may obtain a copy of the License at 7-
 *
 * http://www.apache.org/licenses/LICENSE/ 8-
 *
 * Unless required by applicable law or agree 11-
 * distributed under the License is distribut 12-
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KI 13-
 * See the License for the specific language 14-
 * limitations under the License. 15-
 */
package twitter4j;
import twitter4j.auth.Authorization;
import twitter4j.conf.Configuration;
import java.util.*;
import java.util.concurrent.ConcurrentHashMap
/*
/*
/**
 * A java representation of the <a href="http/* 27-
 * This class is thread safe and can be cache/* 28-

/**
 * Copyright (C) 2007 Yusuke Yamamoto
 * Copyright (C) 2011 Twitter, Inc.
 *
 * Licensed under the Apache License, Version
 * you may not use this file except in compli
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE
 *
 * Unless required by applicable law or agree
 * distributed under the License is distribut
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KI
 * See the License for the specific language
 * limitations under the License.
 */
package twitter4j;
import twitter4j.auth.Authorization;
import twitter4j.conf.Configuration;
import java.util.*;
/**XXXXXXXXXXXXXXXXXXXX*/
/*
/**
 * A java representation of the <a href="http
 * This class is thread safe and can be cache

```

6