

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 is the following code

```
/* 1*/  
/* 2*/ * Copyright (C) 2007 Yusuke Yamamoto  
/* 3*/ * Copyright (C) 2011 Twitter, Inc.  
/* 4*/ *  
/* 5*/ * Licensed under the Apache License, Version 2.0 (the "License");  
/* 6*/ * you may not use this file except in compliance with the License.  
/* 7*/ * You may obtain a copy of the License at  
/* 8*/ *  
/* 9*/ *     http://www.apache.org/licenses/LICENSE-2.0  
/* 10*/ *  
/* 11*/ * Unless required by applicable law or agreed to in writing, software  
/* 12*/ * distributed under the License is distributed on an "AS IS" BASIS,  
/* 13*/ * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
/* 14*/ * See the License for the specific language governing permissions and  
/* 15*/ * limitations under the License.  
/* 16*/ */  
/* 17*/  
/* 18*/ package twitter4j;  
/* 19*/  
/* 20*/ import twitter4j.auth.Authorization;  
/* 21*/ import twitter4j.conf.Configuration;  
/* 22*/  
/* 23*/ import java.util.*;  
/* 24*/ import java.util.concurrent.ConcurrentHashMap;  
/* 25*/
```

```

/* 26*/
/* 27*//**
/* 28*/ * A java representation of the <a href="https://dev.twitter.com/docs/api">Twitter REST...
/* 29*/ * This class is thread safe and can be cached/re-used and used concurrently.<br>
/* 30*/ * Currently this class is not carefully designed to be extended. It is suggested to ex...
/* 31*/ *
/* 32*/ * @author Yusuke Yamamoto - yusuke at mac.com
/* 33*/ */
/* 34*/ class TwitterImpl extends TwitterBaseImpl implements Twitter {
/* 35*/     private static final long serialVersionUID = 9170943084096085770L;
/* 36*/     private static final Logger logger = Logger.getLogger(TwitterBaseImpl.class);
/* 37*/
/* 38*/     /*package*/
/* 39*/     TwitterImpl(Configuration conf, Authorization auth) {
/* 40*/         super(conf, auth);
/* 41*/         INCLUDE_MY_RETWEET = new HttpParameter("include_my_retweet", conf.isIncludeMyRe...
/* 42*/         if (implicitParamsMap.containsKey(conf)) {
/* 43*/             this.IMPLICIT_PARAMS = implicitParamsMap.get(conf);
/* 44*/             this.IMPLICIT_PARAMS_STR = implicitParamsStrMap.get(conf);
/* 45*/         } else {
/* 46*/             String implicitParamsStr = conf.isIncludeEntitiesEnabled() ? "include_entit...
/* 47*/             boolean contributorsEnabled = conf.getContributingTo() != -1L;
/* 48*/             if (contributorsEnabled) {
/* 49*/                 if (!"".equals(implicitParamsStr)) {
/* 50*/                     implicitParamsStr += "&";
/* 51*/                 }
/* 52*/                 implicitParamsStr += "contributingto=" + conf.getContributingTo();
/* 53*/             }
/* 54*/
/* 55*/             if (conf.isTweetModeExtended()) {
/* 56*/                 if (!"".equals(implicitParamsStr)) {
/* 57*/                     implicitParamsStr += "&";
/* 58*/                 }
/* 59*/                 implicitParamsStr += "tweet_mode=extended";
/* 60*/             }
/* 61*/
/* 62*/             List<HttpParameter> params = new ArrayList<HttpParameter>(3);
/* 63*/             if (conf.isIncludeEntitiesEnabled()) {
/* 64*/                 params.add(new HttpParameter("include_entities", "true"));
/* 65*/             }
/* 66*/             if (contributorsEnabled) {
/* 67*/                 params.add(new HttpParameter("contributingto", conf.getContributingTo()));
/* 68*/             }
/* 69*/             if (conf.isTrimUserEnabled()) {
/* 70*/                 params.add(new HttpParameter("trim_user", "1"));
/* 71*/             }
/* 72*/             if (conf.isIncludeExtAltTextEnabled()) {
/* 73*/                 params.add(new HttpParameter("include_ext_alt_text", "true"));
/* 74*/             }
/* 75*/             if (conf.isTweetModeExtended()) {
/* 76*/                 params.add(new HttpParameter("tweet_mode", "extended"));
/* 77*/             }
/* 78*/             HttpParameter[] implicitParams = params.toArray(new HttpParameter[params.si...
/* 79*/
/* 80*/             // implicitParamsMap.containsKey() is evaluated in the above if clause.
/* 81*/             // thus implicitParamsStrMap needs to be initialized first
/* 82*/             implicitParamsStrMap.putIfAbsent(conf, implicitParamsStr);
/* 83*/             implicitParamsMap.putIfAbsent(conf, implicitParams);
/* 84*/
/* 85*/             this.IMPLICIT_PARAMS = implicitParams;
/* 86*/             this.IMPLICIT_PARAMS_STR = implicitParamsStr;
/* 87*/         }
/* 88*/     }
/* 89*/
/* 90*/
/* 91*/     @Override
/* 92*/     public AccountSettings updateAccountSettings(Integer trend_locationWoeid,
/* 93*/                                                     Boolean sleep_timeEnabled, String star...
/* 94*/                                                     String end_sleepTime, String time_zone...
/* 95*/
/* 96*/         throws TwitterException {
/* 97*/         List<HttpParameter> profile = new ArrayList<HttpParameter>(6);
/* 98*/         if (trend_locationWoeid != null) {
/* 99*/             profile.add(new HttpParameter("trend_location_woeid", trend_locationWoeid));
/* 100*/         }
/* 101*/         if (sleep_timeEnabled != null) {
/* 102*/             profile.add(new HttpParameter("sleep_time_enabled", sleep_timeEnabled.toStr...
/* 103*/         }
/* 104*/         if (start_sleepTime != null) {
/* 105*/             profile.add(new HttpParameter("start_sleep_time", start_sleepTime));
/* 106*/         }
/* 107*/         if (end_sleepTime != null) {
/* 108*/             profile.add(new HttpParameter("end_sleep_time", end_sleepTime));
/* 109*/         }
/* 110*/         if (time_zone != null) {
/* 111*/             profile.add(new HttpParameter("time_zone", time_zone));
/* 112*/         }
/* 113*/         if (lang != null) {
/* 114*/             profile.add(new HttpParameter("lang", lang));
/* 115*/         }
/* 116*/         return factory.createAccountSettings(post(conf.getRestBaseUrl() + "account/sett...
/* 117*/             , profile.toArray(new HttpParameter[profile.size()]));
/* 118*/     }
/* 119*/
/* 120*/ }

```

The old version with alerts

```

/* 1-
/* 2-
/* 3-
/* 4-
/* 5-
/* 6-
/* 7-
/* 8-
*/
/* * Copyright (C) 2007 Yusuke Yamamoto
/* * Copyright (C) 2011 Twitter, Inc.
/* *
/* * Licensed under the Apache License, Version 2.0 (the ...
/* * you may not use this file except in compliance with ...
/* * You may obtain a copy of the License at
/* *

```

```

/* 9-
/* 10-
/* 11-
/* 12-
/* 13-
/* 14-
/* 15-
/* 16-
/* 17-
/* 18-
/* 19-
/* 20-
/* 21-
/* 22-
/* 23-
/* 24-UnusedImports
/* 25-
/* 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*/
/* 93-FormalParameterNamingConventions*/
/* 94-FormalParameterNamingConventions*/
/* 94-FormalParameterNamingConventions*/
/* 95-
/* 96-
/* 97-
/* 98-
/* 99-
/* 100-
/* 101-
/* 102-
/* 103-
/* 104-
/* 105-
/* 106-
/* 107-
/* 108-
/* 109-
/* 110-
/* 111-
/* 112-
/* 113-
*/
    http://www.apache.org/licenses/LICENSE-2.0
    */
    */
    * Unless required by applicable law or agreed to in wr...
    * distributed under the License is distributed on an "...
    * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either...
    * See the License for the specific language governing ...
    * limitations under the License.
    */
    */
    */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="https://dev.tw...
    */ * This class is thread safe and can be cached/re-used ...
    */ * Currently this class is not carefully designed to be...
    */ *
    */ * @author Yusuke Yamamoto - yusuke at mac.com
    */ */
    */class TwitterImpl extends TwitterBaseImpl implements Tw...
    */ private static final long serialVersionUID = 917094...
    */ private static final Logger logger = Logger.getLogg...
    */
    */
    *//*package*/
    */ TwitterImpl(Configuration conf, Authorization auth) {
    */ super(conf, auth);
    */ INCLUDE_MY_RETWEET = new HttpParameter("include...
    */ if (implicitParamsMap.containsKey(conf)) {
    */ this.IMPLICIT_PARAMS = implicitParamsMap.ge...
    */ this.IMPLICIT_PARAMS_STR = implicitParamsSt...
    */ } else {
    */ String implicitParamsStr = conf.isIncludeEn...
    */ boolean contributorsEnabled = conf.getContr...
    */ if (contributorsEnabled) {
    */ if (!"".equals(implicitParamsStr)) {
    */ implicitParamsStr += "&";
    */ }
    */ implicitParamsStr += "contributingto=" ...
    */ }
    */
    */ if (conf.isTweetModeExtended()) {
    */ if (!"".equals(implicitParamsStr)) {
    */ implicitParamsStr += "&";
    */ }
    */ implicitParamsStr += "tweet_mode=extend...
    */ }
    */
    */ List<HttpParameter> params = new ArrayList<...
    */ if (conf.isIncludeEntitiesEnabled()) {
    */ params.add(new HttpParameter("include_e...
    */ }
    */ if (contributorsEnabled) {
    */ params.add(new HttpParameter("contribut...
    */ }
    */ if (conf.isTrimUserEnabled()) {
    */ params.add(new HttpParameter("trim_user...
    */ }
    */ if (conf.isIncludeExtAltTextEnabled()) {
    */ params.add(new HttpParameter("include_e...
    */ }
    */ if (conf.isTweetModeExtended()) {
    */ params.add(new HttpParameter("tweet_mod...
    */ }
    */ HttpParameter[] implicitParams = params.toA...
    */
    */ // implicitParamsMap.containsKey() is evalu...
    */ // thus implicitParamsStrMap needs to be in...
    */ implicitParamsStrMap.putIfAbsent(conf, impl...
    */ implicitParamsMap.putIfAbsent(conf, implici...
    */
    */ this.IMPLICIT_PARAMS = implicitParams;
    */ this.IMPLICIT_PARAMS_STR = implicitParamsStr;
    */ }
    */ }
    */
    */ @Override
    */ public AccountSettings updateAccountSettings(Intege...
    */ Boolean...
    */ Boolean...
    */ String...
    */ String...
    */
    */ throws TwitterException {
    */ List<HttpParameter> profile = new ArrayList<Htt...
    */ if (trend_locationWeid != null) {
    */ profile.add(new HttpParameter("trend_locati...
    */ }
    */ if (sleep_timeEnabled != null) {
    */ profile.add(new HttpParameter("sleep_time_e...
    */ }
    */ if (start_sleepTime != null) {
    */ profile.add(new HttpParameter("start_sleep...
    */ }
    */ if (end_sleepTime != null) {
    */ profile.add(new HttpParameter("end_sleep_ti...
    */ }
    */ if (time_zone != null) {
    */ profile.add(new HttpParameter("time_zone", ...
    */ }
    */ if (lang != null) {
    */ profile.add(new HttpParameter("lang", lang));
    */ }

```

```

/*114-                */    }
/*115-                */    return factory.createAccountSettings(post(conf....
/*116-OptimizableToArrayCall    */    , profile.toArray(new HttpParameter[pro...
/*117-                */
/*118-                */    }
/*119-                */
/*120-                */}

```

The new version is the following code. I removed the line 24 “import java.util.concurrent.ConcurrentHashMap”.

```

/* 1*/
/* 2*/ * Copyright (C) 2007 Yusuke Yamamoto
/* 3*/ * Copyright (C) 2011 Twitter, Inc.
/* 4*/ *
/* 5*/ * Licensed under the Apache License, Version 2.0 (the "License");
/* 6*/ * you may not use this file except in compliance with the License.
/* 7*/ * You may obtain a copy of the License at
/* 8*/ *
/* 9*/ *     http://www.apache.org/licenses/LICENSE-2.0
/*10*/ *
/*11*/ * Unless required by applicable law or agreed to in writing, software
/*12*/ * distributed under the License is distributed on an "AS IS" BASIS,
/*13*/ * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
/*14*/ * See the License for the specific language governing permissions and
/*15*/ * limitations under the License.
/*16*/ */
/*17*/
/*18*/ package twitter4j;
/*19*/
/*20*/ import twitter4j.auth.Authorization;
/*21*/ import twitter4j.conf.Configuration;
/*22*/
/*23*/ import java.util.*;
/*24*/
/*25*/
/*26*/ /**
/*27*/ * A java representation of the <a href="https://dev.twitter.com/docs/api">Twitter REST...
/*28*/ * This class is thread safe and can be cached/re-used and used concurrently.<br>
/*29*/ * Currently this class is not carefully designed to be extended. It is suggested to ex...
/*30*/ *
/*31*/ * @author Yusuke Yamamoto - yusuke at mac.com
/*32*/ */
/*33*/ class TwitterImpl extends TwitterBaseImpl implements Twitter {
/*34*/     private static final long serialVersionUID = 9170943084096085770L;
/*35*/     private static final Logger logger = Logger.getLogger(TwitterBaseImpl.class);
/*36*/
/*37*/     /*package*/
/*38*/     TwitterImpl(Configuration conf, Authorization auth) {
/*39*/         super(conf, auth);
/*40*/         INCLUDE_MY_RETWEET = new HttpParameter("include_my_retweet", conf.isIncludeMyRe...
/*41*/         if (implicitParamsMap.containsKey(conf)) {
/*42*/             this.IMPLICIT_PARAMS = implicitParamsMap.get(conf);
/*43*/             this.IMPLICIT_PARAMS_STR = implicitParamsStrMap.get(conf);
/*44*/         } else {
/*45*/             String implicitParamsStr = conf.isIncludeEntitiesEnabled() ? "include_entit...
/*46*/             boolean contributorsEnabled = conf.getContributingTo() != -1L;
/*47*/             if (contributorsEnabled) {
/*48*/                 if (!"".equals(implicitParamsStr)) {
/*49*/                     implicitParamsStr += "&";
/*50*/                 }
/*51*/                 implicitParamsStr += "contributingto=" + conf.getContributingTo();
/*52*/             }
/*53*/
/*54*/             if (conf.isTweetModeExtended()) {
/*55*/                 if (!"".equals(implicitParamsStr)) {
/*56*/                     implicitParamsStr += "&";
/*57*/                 }
/*58*/                 implicitParamsStr += "tweet_mode=extended";
/*59*/             }
/*60*/
/*61*/             List<HttpParameter> params = new ArrayList<HttpParameter>(3);
/*62*/             if (conf.isIncludeEntitiesEnabled()) {
/*63*/                 params.add(new HttpParameter("include_entities", "true"));
/*64*/             }
/*65*/             if (contributorsEnabled) {
/*66*/                 params.add(new HttpParameter("contributingto", conf.getContributingTo()));
/*67*/             }
/*68*/             if (conf.isTrimUserEnabled()) {
/*69*/                 params.add(new HttpParameter("trim_user", "1"));
/*70*/             }
/*71*/             if (conf.isIncludeExtAltTextEnabled()) {
/*72*/                 params.add(new HttpParameter("include_ext_alt_text", "true"));
/*73*/             }
/*74*/             if (conf.isTweetModeExtended()) {
/*75*/                 params.add(new HttpParameter("tweet_mode", "extended"));
/*76*/             }
/*77*/             HttpParameter[] implicitParams = params.toArray(new HttpParameter[params.si...
/*78*/
/*79*/             // implicitParamsMap.containsKey() is evaluated in the above if clause.
/*80*/             // thus implicitParamsStrMap needs to be initialized first
/*81*/             implicitParamsStrMap.putIfAbsent(conf, implicitParamsStr);
/*82*/             implicitParamsMap.putIfAbsent(conf, implicitParams);
/*83*/
/*84*/             this.IMPLICIT_PARAMS = implicitParams;
/*85*/             this.IMPLICIT_PARAMS_STR = implicitParamsStr;
/*86*/         }
/*87*/     }
/*88*/
/*89*/
/*90*/     @Override
/*91*/     public AccountSettings updateAccountSettings(Integer trendLocationWoeid,
/*92*/                                                  Boolean sleep_time_enabled, String sta...
/*93*/                                                  String end_sleepTime, String time_zone...
/*94*/                                                  throws TwitterException {
/*95*/         List<HttpParameter> profile = new ArrayList<HttpParameter>(6);
/*96*/         if (sleep_time_enabled != null) {

```

```

/* 97*/      profile.add(new HttpParameter("sleep_timeEnabled", sleep_time_enabled.toStr...
/* 98*/    }
/* 99*/    if (start_sleepTime != null) {
/*100*/      profile.add(new HttpParameter("start_sleep_time", start_sleepTime));
/*101*/    }
/*102*/    if (end_sleepTime != null) {
/*103*/      profile.add(new HttpParameter("end_sleep_time", end_sleepTime));
/*104*/    }
/*105*/    if (time_zone != null) {
/*106*/      profile.add(new HttpParameter("time_zone", time_zone));
/*107*/    }
/*108*/    if (lang != null) {
/*109*/      profile.add(new HttpParameter("lang", lang));
/*110*/    }
/*111*/    return factory.createAccountSettings(post(conf.getRestBaseUrl() + "account/sett...
/*112*/      , profile.toArray(new HttpParameter[profile.size()]));
/*113*/
/*114*/  }
/*115*/
/*116*/}

```