

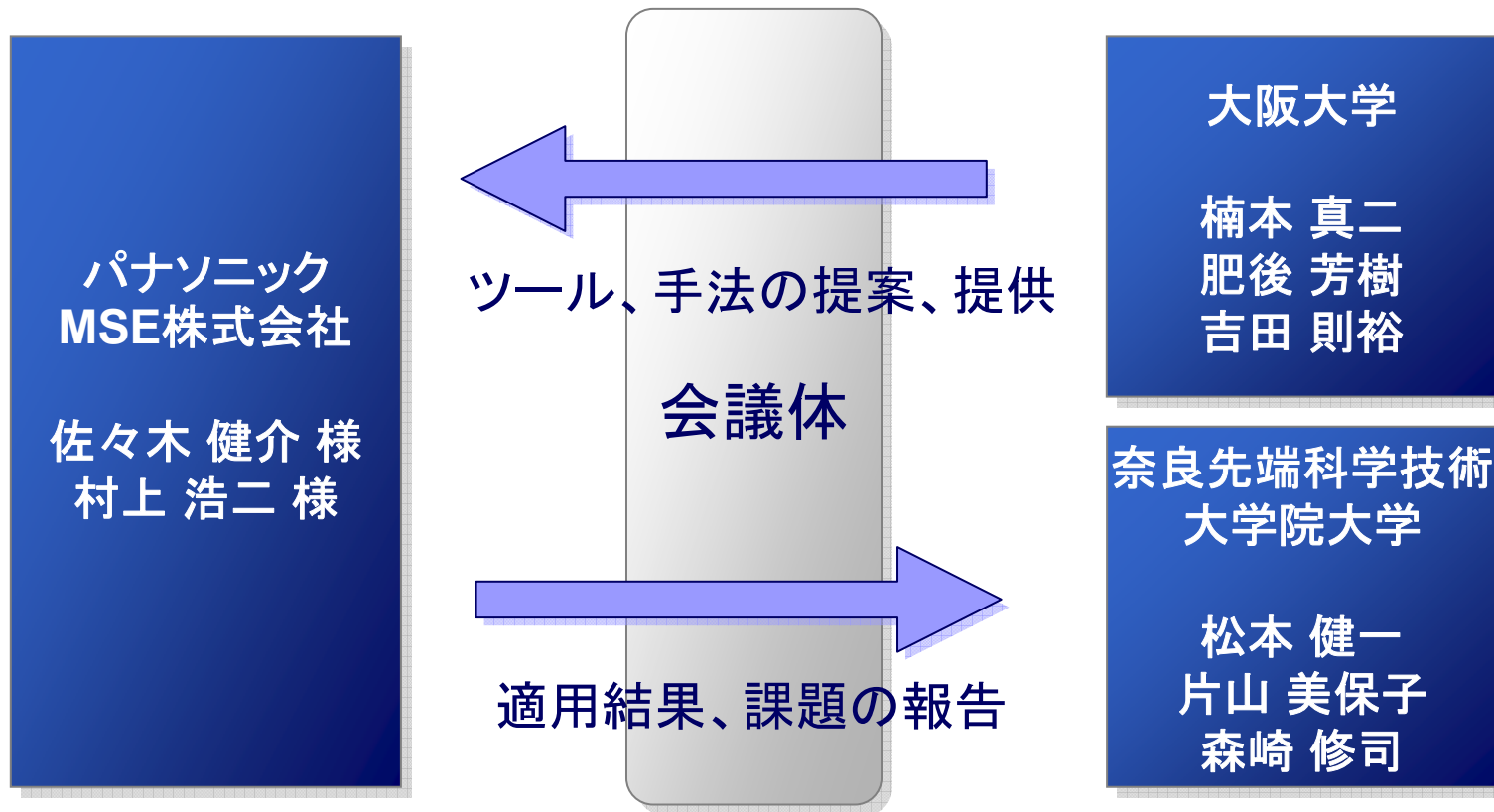
コードクローンによる類似バグ検出 - パナソニックMSE株式会社様での適用事例 -

森崎 修司

EASEプロジェクト

大阪大学/奈良先端科学技術大学院大学

本検討の体制図



本研究の一部は、文部科学省「eSociety基盤ソフトウェアの総合開発」の委託に基づいて行われた。2

概要

- 背景
 - 類似バグ
- コードクローン分析
 - コードクローンによる類似バグの検索
 - キーワード検索との比較
- 試行
 - 方法
 - ツール
 - 結果

類似バグの修正 - 一般的な流れ -

- 発見されたバグに対応するコードを修正する。

```
...  
for (int i = 0; i < 10; i++){  
    stream.writeLine(lines[i]);  
    if (lines[i].equalsTo("EOF")){  
        continue;  
    }  
    tokens = parse(lines[i]);  
}  
...
```

CustomerRequest.java

修正

```
...  
for (int i = 0; i < 10; i++){  
    stream.writeLine(lines[i]);  
    if (lines[i].equalsTo("EOF")){  
        break;  
    }  
    tokens = parse(lines[i]);  
}  
...
```

CustomerRequest.java

類似バグの修正 - 一般的な流れ -

- 担当者やリーダーの記憶の範囲内で、他に修正の必要のある部分を修正

```
...  
for (int i = 0; i < 10; i++){  
    stream.writeLine(lines[i]);  
    if (lines[i].equalsTo("EOF")){  
        continue;  
    }  
    tokens = parse(lines[i]);  
}  
...  
CustomerRequest.java
```

修正

```
...  
for (int i = 0; i < 10; i++){  
    stream.writeLine(lines[i]);  
    if (lines[i].equalsTo("EOF")){  
        break;  
    }  
    tokens = parse(lines[i]);  
}  
... CustomerRequest.java
```

修正すべき?

```
...  
for (int i = 0; i < 10; i++){  
    stream.writeLine(lines[i]);  
    if (lines[i].equalsTo("EOF")){  
        continue;  
    }  
    tokens = parse(lines[i]);  
}  
... StaffRequest.java
```

類似バグの修正

- 記憶に鮮やかでない部分でバグ修正が必要かどうかをチェックしなければならない。

```
...
for (int i = 0; i < 10; i++){
    stream.writeLine(lines[i]);
    if (lines[i].equalsTo("EOF")){
        break;
    }
    tokens = parse(lines[i]);
}
...

StaffRequest.java
CustomerRequest.java
```

他に修正すべき箇所は？

コードクローンによる類似バグの検索 キーワード検索との比較

- 簡単なキーワード検索では、関係のない場所が多く検索結果に現れてしまう。
例) 「grep break *.java」
- 前後の行を含めたり、異なる変数名を考慮するとキーワードの設定が煩雑である。

```
if (lines[i].equalsTo("EOF")){  
    break;  
}
```

修正前コード

```
if (lines[i].equalsTo("EOF")){  
    continue;  
}
```

修正後コード

コードクローンによる類似バグの検索 変数の匿名化

- 文法を解析し変数部分を匿名化(ワイルドカード化)することにより、異なる変数名を使った似通ったロジックを検索できる。

```
if (lines[i].equalsTo("EOF")){  
    break;  
}
```

検索キーとなるコード片

```
if (stream[i].equalsTo("EOF")){  
    break;  
}
```

検索結果

- 過去のプロジェクトのソースコードを利用する。
 - A: Windowsアプリ C++ 約13.5kstep
 - B: 68系組込みサブシステム C 約16.7kstep
 - C: Windows C/S サブシステム Java: 約7.2kstep
- コード修正の規則を利用する。
 - 同一の不具合に対する修正に対して同一の修正IDが記述されている。
 - 修正後のコードを検索キーとして修正ID部分が検索結果に含まれるかを調べる。
- あくまで実験であり、実際には修正前のコードに対して検索を実施する必要がある。

修正箇所の記述ルール

```
if (ptr != NULL) {  
    *ptr++;  
}
```

修正前

```
if (ptr != NULL) {  
    *ptr += 8;  
}
```

修正後

```
if (ptr != NULL) {  
// 変更部分開始 ID0001 2007/7/9  
//    *ptr++;  
// 変更部分終了 ID0001  
    *ptr += 8;  
}
```

同一の不具合の修正には同一のIDをつけ、変更記録を残す。

ソースコードへの記述

修正箇所の記述ルール

```
if (ptr != NULL) {  
// 変更部分開始 ID0001  
//   *ptr++;  
// 変更部分終了 ID0001  
   *ptr += 8;  
}
```

client.c

```
if (ptr != NULL) {  
// 変更部分開始 ID0001  
//   *ptr++;  
// 変更部分終了 ID0001  
   *ptr += 8;  
}
```

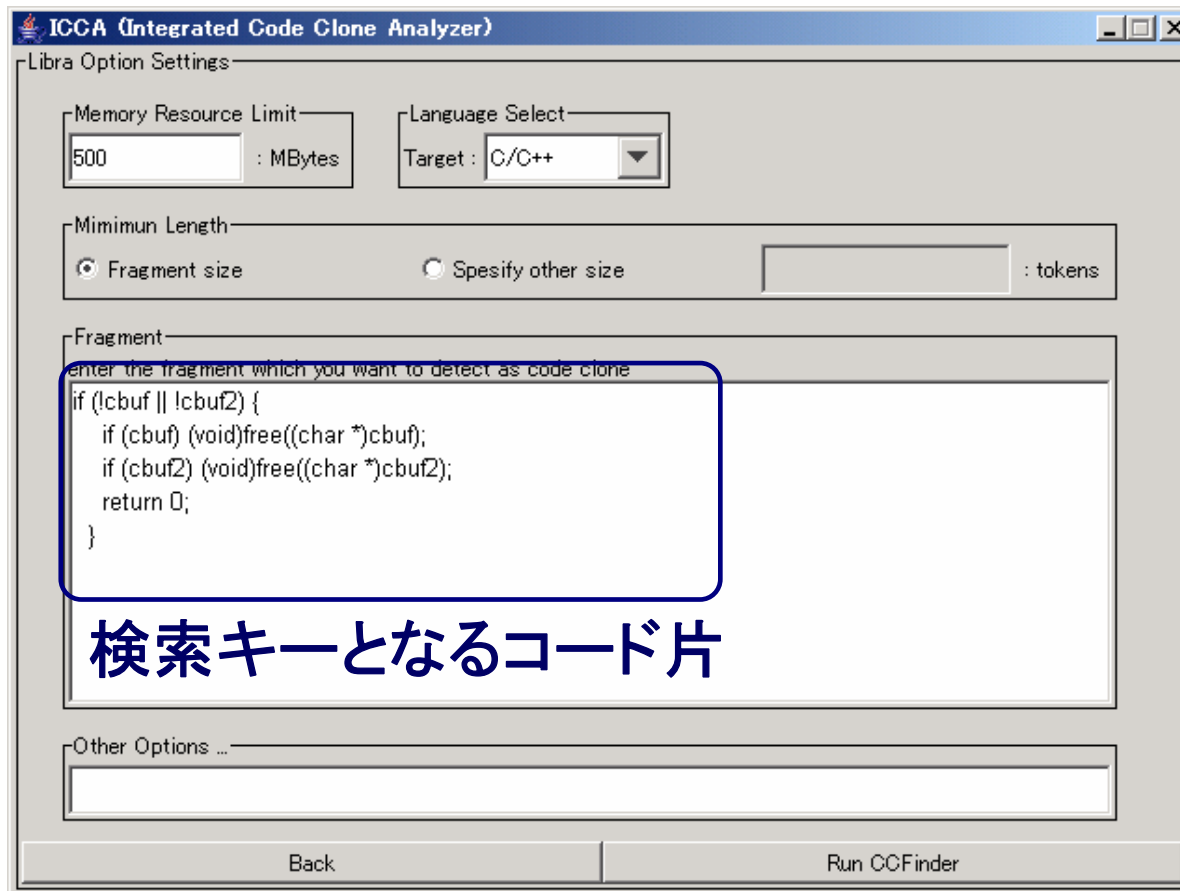
server.c

異なる修正箇所、ソースコードファイルであっても、
同一の不具合には同一のIDを記録する。

試行

コードクローン検索ツールLibra

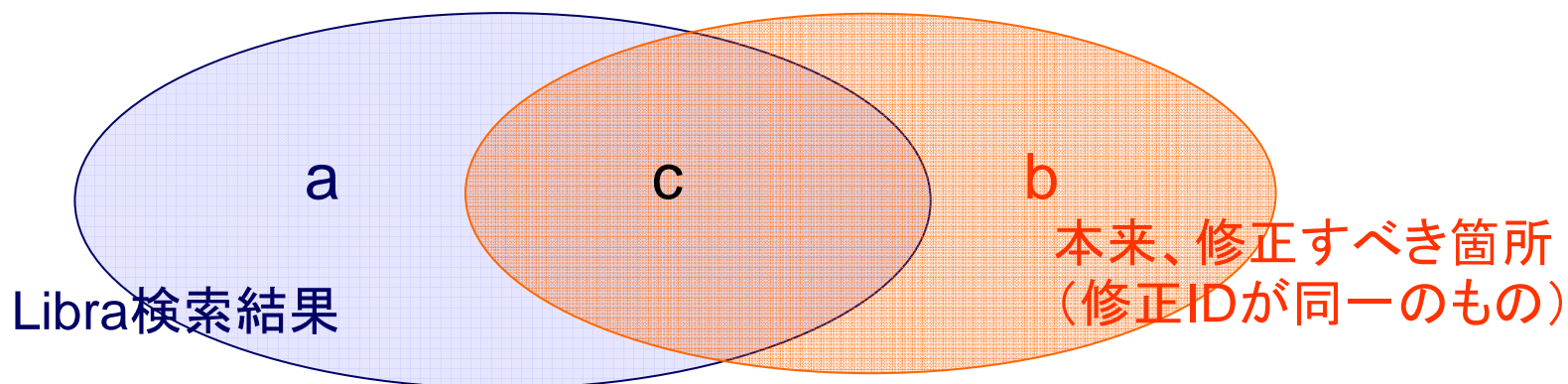
- 対象ソースコードを指定し、バグ修正の対象となったソースコード片を検索キーとして指定する。



12

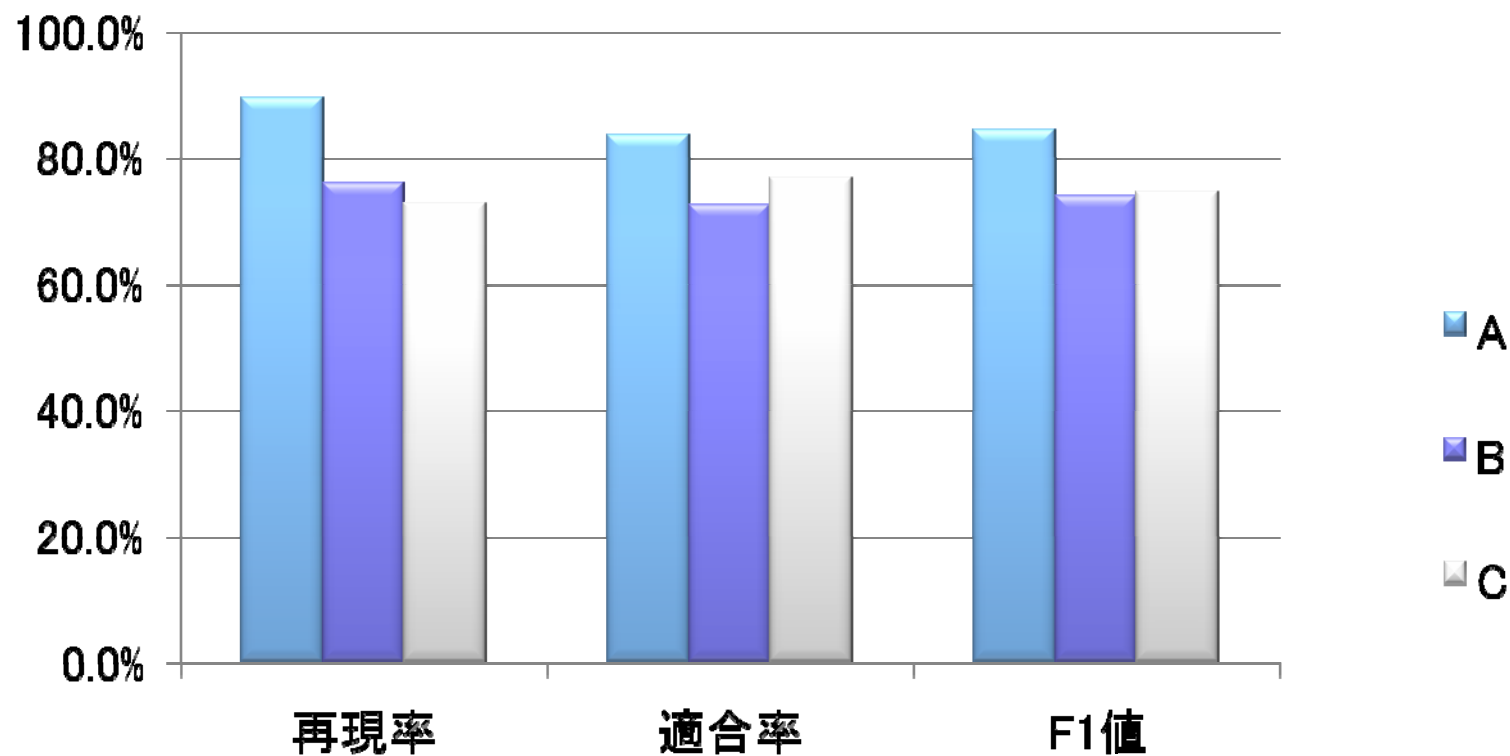
再現率と適合率

- 適合率: c/a (検索結果に修正すべき箇所が含まれていた割合)
- 再現率: c/b (修正すべき箇所のうち検索結果が含まれていた割合)
- F1値: 適合率と再現率のバランス (値が高いほど再現率と適合率のバランスがよい。)



試行 試行結果

- A: Windowsアプリ C++ 約13.5kstep
- B: 68系組込みサブシステム C 約16.7kstep
- C: Windows C/S サブシステム Java: 約7.2kstep



試行

A: Windowsアプリ(言語:C++のみ抽出) 約13.5kstep

検索結果の例

検索コード片: `sys->usSysStat |= SYSST_UNYO_U;`

検索結果: 2件

本来修正すべき箇所(同一ID)	検索結果に含まれていた
<code>sys->usSysStat = SYSST_SYSTEM_ON;</code>	○
<code>sys->usSysStat = SYSST_UNYO_S;</code>	○
<code>sys->usSysStat &= ~SYSST_UNYO_S;</code>	×

検索結果が多かった上位3件のコード片

順位	与えたコード片	件数
1	<code>pStrRes = (InfB_AnrmSig *)pby_wk;</code>	410
2	<code>CSysCom::TimerCntl(this, enTimer_Cntcnd, FALSE, bCntcndTimeFlg);</code>	48
3	<code>for(int i=0; i<5; ++i)</code>	29

15

試行

B: 68系組込みサブシステム C 約16.7kstep

検索結果の例

検索コード片:

```
if((( ncuno > 0 )&&( ncuno < NCUMAX+1 ))&&  
    (( chno > 0 )&&( chno < NCUCH+1 ))) {  
    vs_ncuctl( ncuno, chno, CODEC_ON );  
}
```

検索結果: 2件

本来修正すべき箇所(同一ID)	検索結果に含まれていた
<pre>if(((ncuno > 0)&&(ncuno < NCUMAX+1))&& ((chno > 0)&&(chno < NCUCH+1))) { vs_ncuctl(ncuno, chno, CODEC_OFF); }</pre>	○
<pre>if(((ncuno > 0)&&(ncuno < NCUMAX+1))&& ((chno > 0)&&(chno < NCUCH+1))) { cm_ncuctl(ncuno, chno, CODEC_OFF); }</pre>	○

16

試行

B: 68系組込みサブシステム C 約16.7kstep

検索結果が多かった上位3件のコード片

順位	検索キー	件数
1	seicnt++;	136
2	c_delay(100);	86
3	if(p_cdctl->d_vinibn != 0){ return; }	8

試行

C: Windows C/S サブシステム Java: 約7.2kstep

検索コード片:

```
ret = ( m_main.req_device != 1 ) ? IsCom.S_NET_NG : IsCom.MO_DAT_NG;
```

検索結果: 5件

検索結果の例

本来修正すべき箇所(同一ID)	検索結果に含まれていた
ret = (m_main.req_device != 1) ? IsCom.S_NET_NG : IsCom.MO_DAT_NG;	○
ret = (m_main.req_device != 1) ? IsCom.S_NET_NG : IsCom.MO_DAT_NG;	○
ret = (m_main.req_device != 1) ? IsCom.S_NET_NG : IsCom.MO_DAT_NG;	○
ret = (m_main.req_device != 1) ? IsCom.S_NET_NG : IsCom.MO_DAT_NG;	○
ret = (m_main.req_device != 1) ? IsCom.S_NET_NG : IsCom.MO_DAT_NG;	○

検索結果が多かった上位3件のコード片

順位	検索キー	件数
1	if(output != IsCom.RTN_OK)	424
2	output = m_reset.getTermStatus(request);	246
3	Line2 += MachAdd;	60

- 完全な網羅性はないが、高い精度で類似バグを検出でき、適用コストも低い。
 - コーディング～単体テストで担当者レベルで発見したバグを見つけるために利用できる。
 - 結合～システムテストで修正したバグが派生プロダクトや他の担当者の部分にないかをさがすために利用できる。
- 検索コード片
 - 複数行にまたがるようなコード片の場合、よい結果が得られる。
 - 1行に収まるような修正の場合、検索結果が大量となり、その全てを追跡するのは現実的ではない。

コードクローン分析ツールのご案内

- CCFinderX
 - 作者: 産業総合研究所 神谷年洋氏
 - 詳細: <http://www.ccfinder.net/ccfinderx-j.html>
- ICCA (Libraを含むツール群)
 - 作者: 大阪大学 肥後芳樹氏
 - 詳細: <http://sel.ics.es.osaka-u.ac.jp/icca/index.html>