

品質保証活動における CCFinderの適用事例報告

2007年 2月7日

株式会社NTTデータ
基盤システム事業本部
町田 欣史



- **はじめに**
- **コードクローン検出サービスの導入**
 - **コードクローンメトリクスに関する情報の提供**
 - **コードクローン箇所に関する情報の提供**
- **まとめ**



はじめに



コードクローン検出サービスの導入



コードクローンメトリクスに関する情報の提供



コードクローン箇所に関する情報の提供



まとめ

● 所属

- 株式会社NTTデータ 基盤システム事業本部
 - テスト・品質保証に関する技術支援、研究

● 活動

- JSTQB(テスト技術者資格認定)技術委員会
- SQuBOK(ソフトウェア品質知識体系)策定部会
- 執筆・講演歴
 - ソフトウェア・テスト PRESS Vol.1～Vol.4
 - JSTQB教科書 JSTQB認定テスト技術者 Foundation Level試験
 - @IT「Eclipseで使えるテストツールカタログ」
 - JavaWorld DAY 2005
 - 第25回 ソフトウェア品質シンポジウム

コードクローンを紹介しています



NTTデータグループ

基盤システム事業本部

システム開発のプラットフォームにおける技術、品質、生産性を追求し、NTTデータグループ全体に展開する

ソリューション
の提供

公共

金融

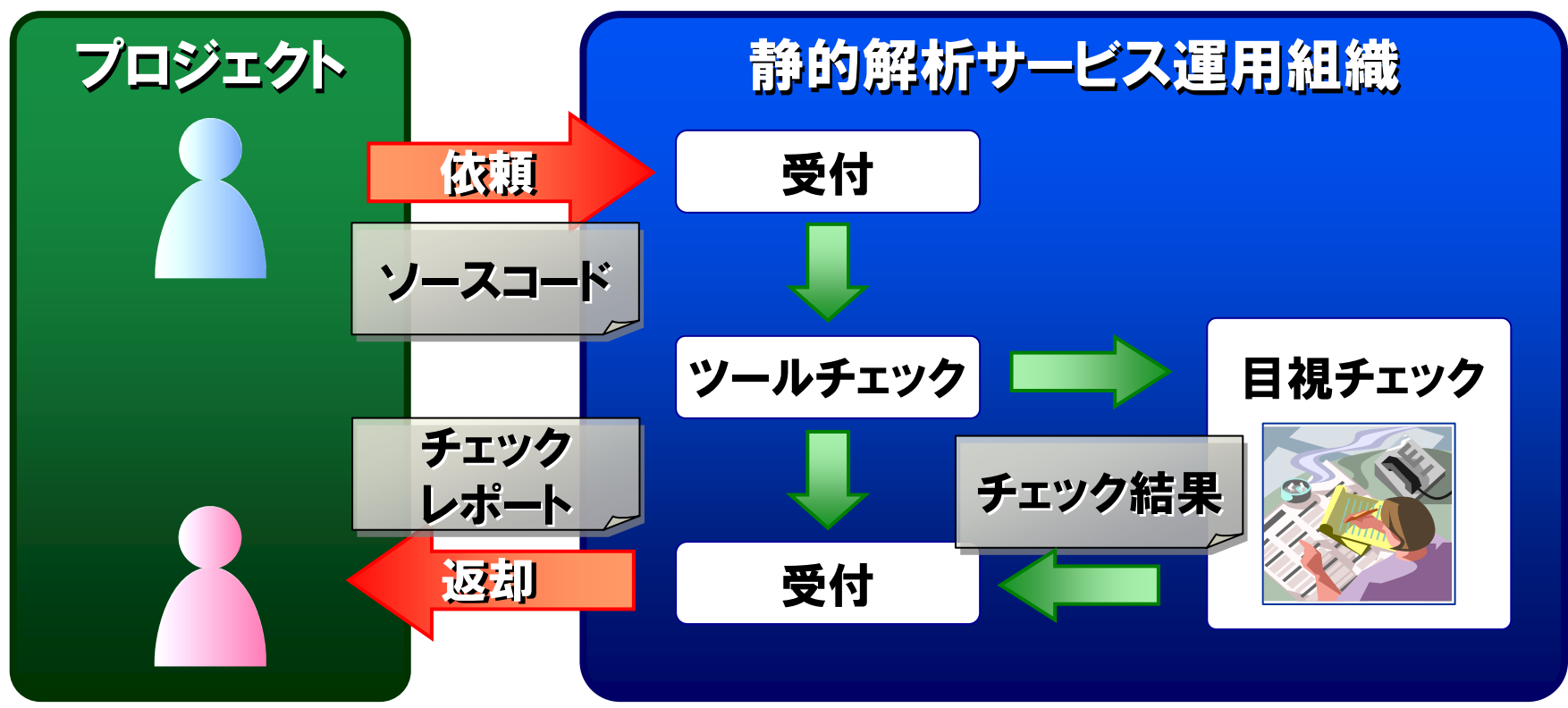
法人

事業本部

NTTデータグループ会社

Webアプリケーション開発の総合的ソリューション
「TERASOLUNA」の社内・グループ会社内展開

● 社内プロジェクト向け静的解析サービス



静的解析ツールによるチェック結果と目視によるソースコードレビューの結果を合わせてレポートする

静的解析

プログラムを実行せずにソースコードを検査して、エラーの可能性のある記述を見つけ出す手法

コーディング規約に基づくチェック

パターン化されたバグの検出
コーディング規約違反の検出

➡ 記述の誤り(顕在しているバグ、エラー)を見つける

ソースコード・メトリクス計測

ソースコードの構造的な特徴を数値で示す

➡ バグが“潜在”する可能性のあるコードを見つける

コードクローンの検出

類似したコード対の検出

➡ 保守を困難にするコードを見つける



- はじめに
- **コードクローン検出サービスの導入**
 - コードクローンメトリクスに関する情報の提供
 - コードクローン箇所に関する情報の提供
- まとめ

- 静的解析サービスで、**コードクローン**に関する情報を提供する
 1. **コードクローンメトリクスに関する情報**
 2. **コードクローン箇所に関する情報**

- **使用するツール**

- ICCA/CCFinder
 - Gemini
 - Aries

通常の利用形態



- GUI上で解析および結果の確認を行う

静的解析サービスでの利用形態

- リモート環境で解析し、結果のみを伝える



CCFinderの出力ファイルをもとにレポートを生成



- はじめに
- コードクローン検出サービスの導入
- **コードクローンメトリクスに関する情報の提供**
- コードクローン箇所に関する情報の提供
- まとめ



ソースコード・メトリクス

- ソースコードの状態を数値で表す
- ↓
- ソースコードの品質を可視化する
- ↓
- 低品質なソースコードを検出する

コードクローンに関する
メトリクスもレポートに
追加

保守性

可読性やテスト容易性が低い

信頼性

複雑でバグが潜在する

可能性が高い

ソースコード・メトリクスの例

- クラスやメソッドなどの数
- クラスやメソッドの内部構造
 - 規模、サイクロマチック数、凝集度、など
- パッケージ間やクラス間の関係
 - 結合度

● Geminiが計測するメトリクス(ファイル≒クラス単位)

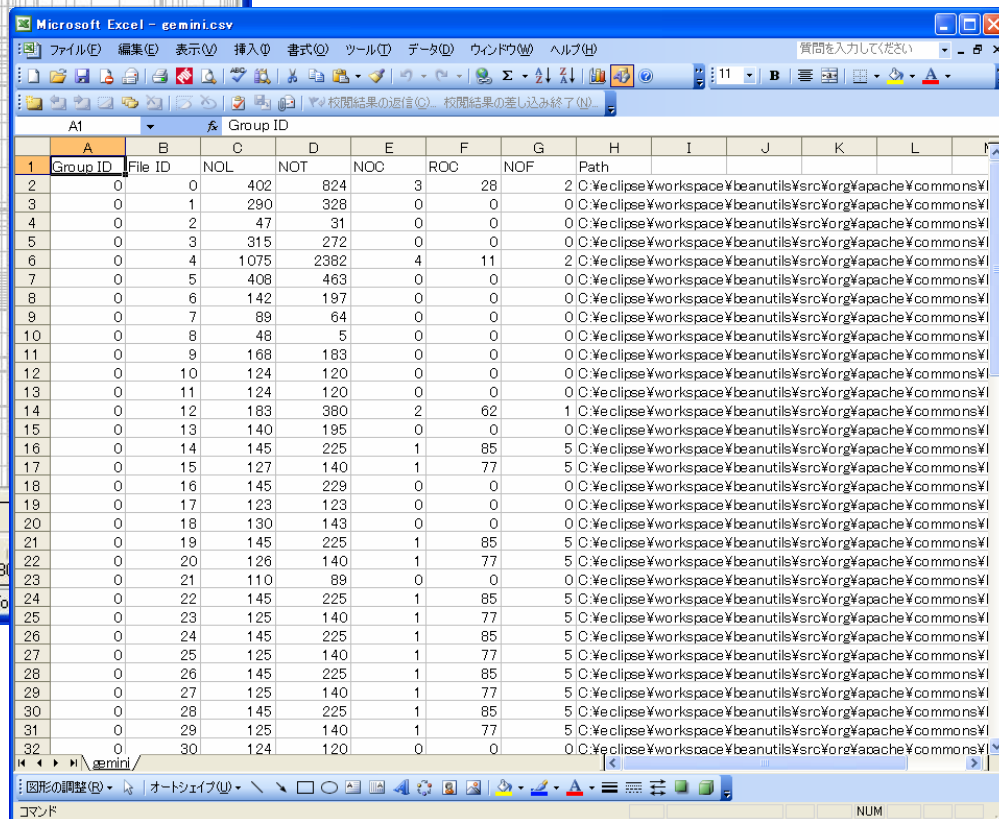
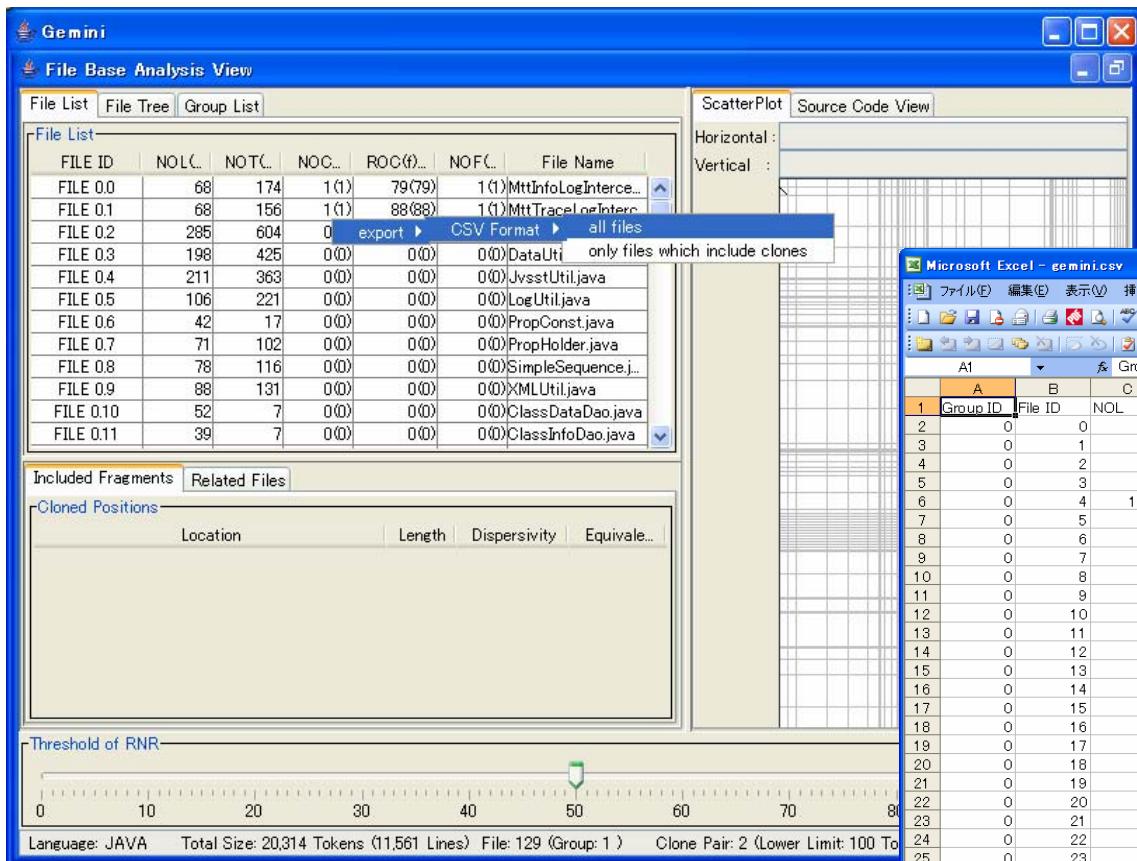
- NOL そのファイルの行数 **不採用**
 ➔ 社内ツールで計測する
- NOT そのファイルのトークン数 **不採用**
 ➔ 規模を表す意味でNOLと大きな違いはない
- NOC そのファイルに含まれるクローン数 **不採用**
 ➔ クローンの数は規模に依存する
- ROC そのファイルの重複度 **採用**
 ➔ クローン片となっている割合が多いほどメンテナンス性が低い
- NOF そのファイルとクローンを共有しているファイルの数 **採用**
 ➔ 多くのファイルとクローンを共有しているほどメンテナンス性が低い

静的解析サービスでは「クラス単位」での品質評価を実施



ROC(類似コード率)とNOF(類似コード共有クラス数)を採用

- GeminiのGUIからCSV形式のファイルを出力



他のメトリクスもあわせて
レポートを作成(次ページ)

ソースコード品質の定量化の例

項目		類似コード数	類似コード率	類似コード共有クラス数		類似コード率		類似コード共有クラス数		類似コード率		類似コード共有クラス数		類似コード率		類似コード共有クラス数		類似コード率		
11	測定クラス数	81	81	75	75	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81
12	注意クラス数	0	4	8	9	8	7	2	3	3	3	3	3	3	3	3	3	3	3	3
13	警告クラス数	0	0	2	0	12	13	1	5	5	5	5	5	5	5	5	5	5	5	5
14	合計	7,922		1,798																
15	最小値	4	33%	2	0%	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	最大値	964	92%	220	90%	5	1	29	32	32	32	32	32	32	32	32	32	32	32	32
17	平均値	97.8	63.4%	24.0	15.4%	0.9	0.1	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6
59	org.apache.commons.beanutils.converter.s.DoubleArrayConverter	57	50%	13	0%	5	85%	2	1	1	1	1	1	1	1	1	1	1	1	1
60	org.apache.commons.beanutils.converter.s.IntegerArrayConverter	57	50%	13	0%	5	85%	2	1	1	1	1	1	1	1	1	1	1	1	1
61	org.apache.commons.beanutils.converter.s.FloatArrayConverter	57	50%	13	0%	5	85%	2	1	1	1	1	1	1	1	1	1	1	1	1
62	org.apache.commons.beanutils.converter.s.ShortArrayConverter	57	50%	13	0%	5	85%	2	1	1	1	1	1	1	1	1	1	1	1	1
63	org.apache.commons.beanutils.converter.s.LongArrayConverter	57	50%	13	0%	5	85%	2	1	1	1	1	1	1	1	1	1	1	1	1
64	org.apache.commons.beanutils.converter.s.BooleanArrayConverter	57	50%	13	0%	5	85%	2	1	1	1	1	1	1	1	1	1	1	1	1
65	org.apache.commons.beanutils.converter.s.ByteArrayConverter	57	50%	13	0%	5	85%	2	1	1	1	1	1	1	1	1	1	1	1	1
66	org.apache.commons.beanutils.converter.s.CharacterArrayConverter	57	50%	13	0%	5	85%	2	1	1	1	1	1	1	1	1	1	1	1	1
67	org.apache.commons.beanutils.converter.s.BigDecimalConverter	57	50%	13	0%	5	85%	2	1	1	1	1	1	1	1	1	1	1	1	1
68	org.apache.commons.beanutils.converter.s.BigIntegerConverter	57	50%	13	0%	5	85%	2	1	1	1	1	1	1	1	1	1	1	1	1
69	org.apache.commons.beanutils.converter.s.DateConverter	57	50%	13	0%	5	85%	2	1	1	1	1	1	1	1	1	1	1	1	1
70	org.apache.commons.beanutils.converter.s.TimeConverter	57	50%	13	0%	5	85%	2	1	1	1	1	1	1	1	1	1	1	1	1

類似コード数
類似コード率
類似コード共有クラス数

類似コード共有クラス数
類似コード率

5 85%
5 77%
0 0%
5 85%
5 77%
5 85%
5 77%
5 85%
5 77%
5 85%
5 77%

コードクローンの多いクラスは赤色で警告を表示

(参考)採用したソースコード・メトリクス

規模

行数(LOC)

コメント

コメント率

複雑度

サイクロマチック数
ネストの深さ

凝集度

凝集度の欠如

結合度

参照クラス数
被参照クラス数
呼び出しメソッド数
呼び出し元メソッド数

類似度

コードクローンを利用

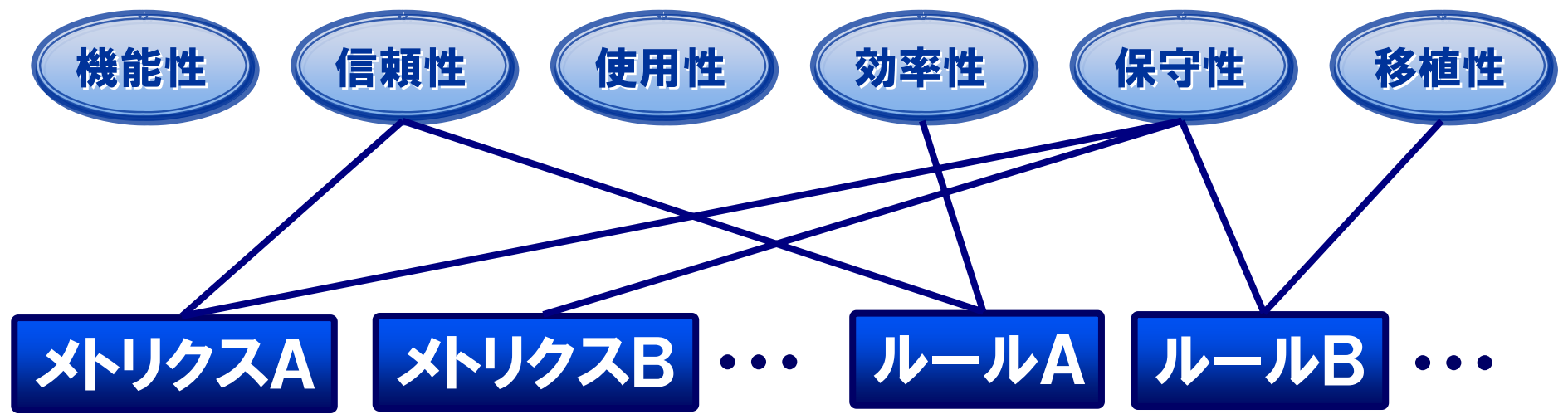
類似コード率
類似コード共有クラス数

データ数

パラメータ数

- ソースコード・メトリクス と チェックルール(コーディング規約) を品質特性と対応づける

ISO/IEC9126 (JIS X0129) の 6つの品質特性

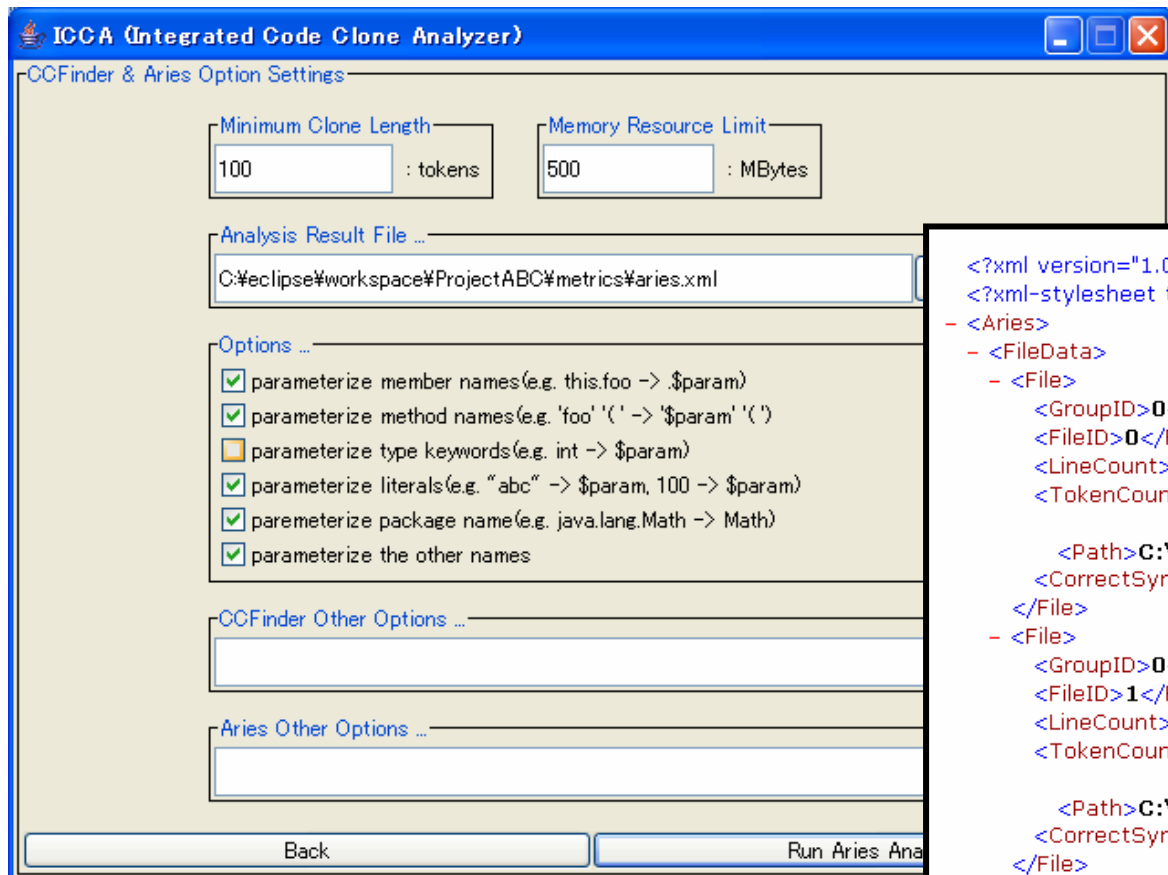


品質特性ごとに定量化



- はじめに
- コードクローン検出サービスの導入
 - コードクローンメトリクスに関する情報の提供
 - **コードクローン箇所に関する情報の提供**
- まとめ

- AriesのGUIからXML形式のファイルを出力



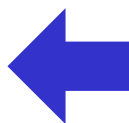
```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xsl" ?>
- <Aries>
- <FileData>
  - <File>
    <GroupID>0</GroupID>
    <FileID>0</FileID>
    <LineCount>402</LineCount>
    <TokenCount>1235</TokenCount>

    <Path>C:\eclipse\workspace\beanutils\src\org\apache\commons\b
    <CorrectSyntax>true</CorrectSyntax>
  </File>
- <File>
  <GroupID>0</GroupID>
  <FileID>1</FileID>
  <LineCount>290</LineCount>
  <TokenCount>513</TokenCount>

  <Path>C:\eclipse\workspace\beanutils\src\org\apache\commons\b
  <CorrectSyntax>true</CorrectSyntax>
</File>
- <File>
  <GroupID>0</GroupID>
  <FileID>2</FileID>
  <LineCount>47</LineCount>
  <TokenCount>39</TokenCount>

  <Path>C:\eclipse\workspace\beanutils\src\org\apache\commons\b
```

必要な情報を抽出
(次ページ)



● Ariesの出力ファイルから必要な情報を抽出

ファイル(クラス)情報

```

- <Class>
  <ID>61</ID>
  - <Package>
    <Name>org</Name>
    <Name>apache</Name>
    <Name>commons</Name>
    <Name>beanutils</Name>
    <Name>locale</Name>
    <Name>LocaleBeanUtils</Name>
  </Package>
  - <Location>
    <GroupID>0</GroupID>
    <FileID>61</FileID>
    <FromLine>45</FromLine>
    <FromColumn>8</FromColumn>
    <ToLine>531</ToLine>
    <ToColumn>1</ToColumn>
  </Location>
  
```

クラス名

対応するファイルに
存在するクローン
(IDで関連づけ)

クローンセット情報

```

- <Fragments>
  - <Class>
    <ID>61</ID>
    - <Location>
      <GroupID>0</GroupID>
      <FileID>61</FileID>
      <FromLine>474</FromLine>
      <FromColumn>22</FromColumn>
      <FromToken>1192</FromToken>
      <ToLine>530</ToLine>
      <ToColumn>5</ToColumn>
      <ToToken>1387</ToToken>
    </Location>
    <VariableData />
  </Class>
  - <Class>
    <ID>62</ID>
    - <Location>
      <GroupID>0</GroupID>
      <FileID>62</FileID>
      <FromLine>898</FromLine>
      <FromColumn>15</FromColumn>
      <FromToken>2283</FromToken>
      <ToLine>954</ToLine>
      <ToColumn>5</ToColumn>
      <ToToken>2478</ToToken>
    </Location>
    <VariableData />
  </Class>
  
```

クローンの範囲

```

    <Name>locale</Name>
    <Name>LocaleBeanUtilsBean</Name>
  </Package>
  - <Location>
    <GroupID>0</GroupID>
    <FileID>62</FileID>
    <FromLine>44</FromLine>
    <FromColumn>8</FromColumn>
    <ToLine>955</ToLine>
    <ToColumn>1</ToColumn>
  </Location>
  
```

- コードクローンとなっている箇所の情報をレポート

```
エラー詳細レポート.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

org.apache.commons.beanutils.locale.LocaleBeanUtilsBean
-----
+++++++
類似コード : 2種類
+++++++
1:
  org/apache/commons/beanutils/BeanUtilsBean.java:307-332
```

org/apache/commons/beanutils/locale/LocaleBeanUtils.java:474-530
org/apache/commons/beanutils/locale/LocaleBeanUtilsBean.java:898-954

```
org.apache.commons.beanutils.locale.converters.BigDecimal
-----
+++++++
類似コード : 1種類
+++++++
1:
  org/apache/commons/beanutils/locale/converters/BigDecimalLocaleConverter.java:33-198
  org/apache/commons/beanutils/locale/converters/BigIntegerLocaleConverter.java:32-198
  org/apache/commons/beanutils/locale/converters/ShortLocaleConverter.java:33-198
```

クローンセットとなっているクラス名とその範囲(開始行・終了行)



- はじめに
- コードクローン検出サービスの導入
 - コードクローンメトリクスに関する情報の提供
 - コードクローン箇所に関する情報の提供
- **まとめ**

効果

- CCFinderを持っていない人、使い方を知らない人にもコードクローンの情報を示すことができた
- 規約チェックでは気づかない低品質なコードを示すことができた

参考

- チェックされたクラスの割合
 - 規約チェック 約40%
 - コードクローン 約15%
 - ➡ コードクローンのみ 約5%



問題点・課題

- GeminiとAriesの2つのツールを使っている
 - ➡ メトリクスとクローン情報が一致しない場合がある
- 最新ツール(CCFinderX)を導入できていない
- コードの具体的な修正方法まで示せていない
- 最小クローン長の最適な設定値を見つける
 - ➡ 現在は100トークンに設定
- メモリ使用量が多く、解析不能となることがある



- Eclipseプラグインとして使えるとよい
 - 規約チェックやメトリクス計測はEclipseプラグインツールを用いて開発者が容易に実行できる
 - ➡ コードクローンも同様にできるとよい
- リファクタリングのパターンもレポートできるとよい
 - AriesのGUI上では確認できるが、ファイルには出力されない
- GUI実行時の性能向上

コードクローン検出は高品質なコードを作る上で非常に有効であり、現場導入も含めて推進していきたい

お問い合わせ先

株式会社NTTデータ
基盤システム事業本部
システム方式技術ビジネスユニット
第一技術統括部 第一システム方式技術担当

TEL:050-5546-2482

町田 欣史:machidays@nttdata.co.jp

品質保証活動における CCFinderの適用事例報告

2007年 2月7日

株式会社NTTデータ
基盤システム事業本部
町田 欣史