

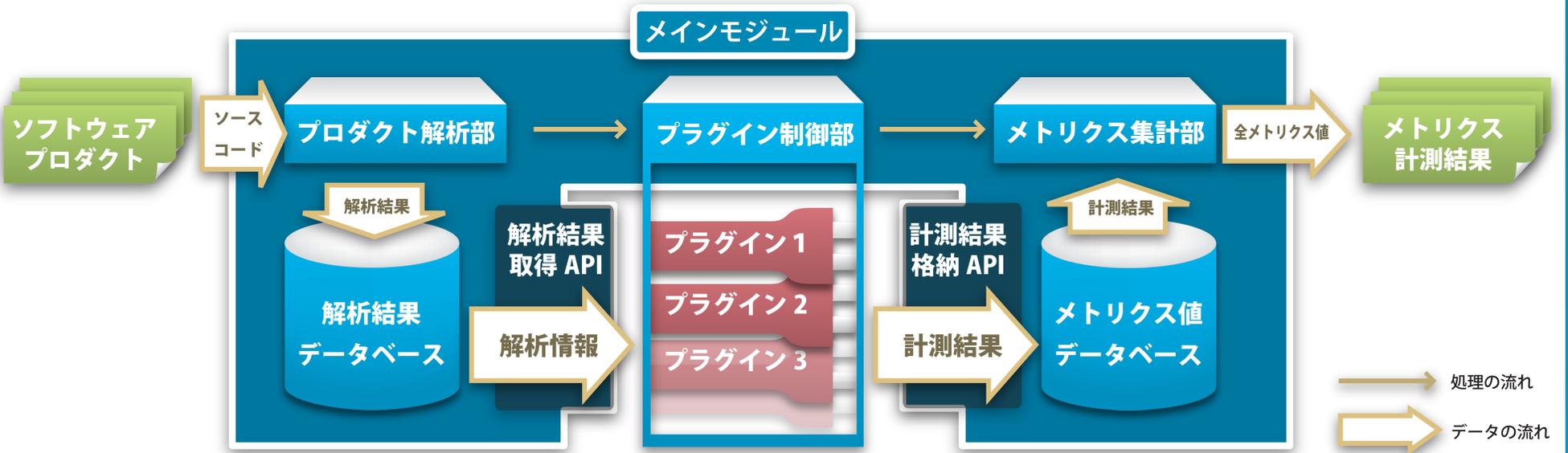
# メトリクス計測ツール MASU の Eclipse プラグイン

山田吾郎 齋藤晃 三宅達也 肥後芳樹 楠本真二 井上克郎

† 大阪大学 大学院情報科学研究科



プラグイン形式の多言語対応メトリクス計測プラットフォーム



## プラグイン形式のメトリクス計測ツール

- ファイル、ディレクトリ単位でメトリクス計測対象を指定
- メトリクス計測部はプラグインとして実装
  - メトリクス計測ロジックの記述のみで実装可能
- 1つのプラグインで複数言語のソースコードから計測可能
- CKメトリクスの1つ、CBO計測プラグインの実装例

```
import jp.ac.osaka_u.ist.sel.metricstool.main.plugin.AbstractClassMetricPlugin;
import ...;

public class CBOPlugin extends AbstractClassMetricPlugin {
    protected Number measureClassMetric(TargetClassInfo targetClass) {
        Set<ClassTypeInfo> classes = new HashSet<ClassTypeInfo>();
        for (final TargetFieldInfo field : targetClass.getDefinedFields()) {
            final TypeInfo type = field.getType();
            if (type instanceof ClassTypeInfo) classes.add((ClassTypeInfo) type);
        }
        for (final TargetMethodInfo method : targetClass.getDefinedMethods()) {
            final TypeInfo returnType = method.getReturnType();
            if (returnType instanceof ClassTypeInfo) classes.add((ClassTypeInfo) returnType);
            for (final ParameterInfo parameter : method.getParameters()) {
                final TypeInfo parameterType = parameter.getType();
                if (parameterType instanceof ClassTypeInfo) classes.add((ClassTypeInfo) parameterType);
            }
            for (final VariableInfo? extends UnitInfo variable : LocalVariableInfo
                .getLocalVariables(method.getDefinedVariables())) {
                final TypeInfo variableType = variable.getType();
                if (variableType instanceof ClassTypeInfo) classes.add((ClassTypeInfo) variableType);
            }
        }
        classes.remove(targetClass);
        return classes.size();
    }
    protected String getDescription() { return "Measuring the CBO metric."; }
    protected String getMetricName() { return "CBO"; }
    protected boolean useFieldInfo() { return true; }
    protected boolean useMethodInfo() { return true; }
}
```

## 静的解析ライブラリ

- 解析結果取得 API は外部プログラムから利用できる
- MASU が構築したデータベースを元に静的解析を容易に行える
- 以下は、./src 以下の Java ソースコードの全メソッドについて呼び出しているメソッドと呼び出されているメソッドを列挙するプログラムの実装例

```
import jp.ac.osaka_u.ist.sel.metricstool.main.MetricsTool;
import ...;

public class MASUTest extends MetricsTool {
    public static void main(java.lang.String[] args) {
        Settings.getInstance().setTargetDirectory("./src");
        Settings.getInstance().setLanguage("java");

        final MASUTest test = new MASUTest();
        test.readTargetFiles();
        test.analyzeTargetFiles();

        final ClassInfoManager classInfoManager = DataManager.getInstance().getClassInfoManager();
        for (final TargetClassInfo classInfo : classInfoManager.getTargetClassInfos()){
            for (final TargetMethodInfo methodInfo : classInfo.getDefinedMethods()){
                System.out.println("** " + methodInfo.getMeasuredUnitName() + " calls...");
                for (final CallInfo? call : methodInfo.getCalls()){
                    System.out.println("  - " + call.getText());
                }
                System.out.println("  " + methodInfo.getMeasuredUnitName() + " is called by...");
                for (final CallableUnitInfo caller : methodInfo.getCallers()){
                    System.out.println("    - " + caller.getSignatureText());
                }
            }
        }
    }
}
```

## 開発した Eclipse プラグインの概要

- MASU の GUI フロントエンド
- Eclipse から手軽にメトリクス計測を行える
- 機能
  - ファイル、ディレクトリ単位で計測対象を指定
  - 計測結果は 2 通りの出力
    - テキストファイル
    - Eclipse 内のビュー
  - Eclipse 内ビューに出力した場合、過去の計測結果と比較ができる
- MASU プラグインのソースコードのテンプレートを出力する機能などを実装予定

