

Development of Plug-in Platform for Metrics Measurement

Akira Saito, Goro Yamada, Tatsuya Miyake, Yoshiki Higo, Shinji Kusumoto, Katsuro Inoue

† Graduate School of Information Science and Technology, Osaka University



Metrics Assessment plugin-platform for Software Unit of multiple programming languages

Problems

Many metrics tools have been proposed

However, these are three problems

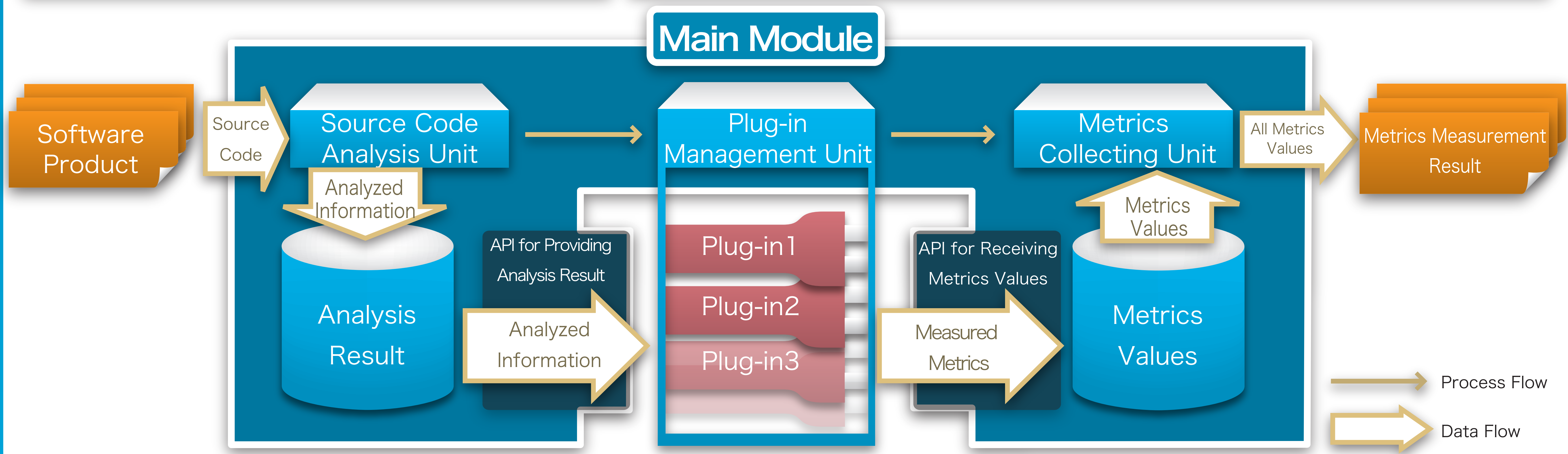
- Ambiguous definitions
 - ▶ A metric has several definitions
 - ▶ Each tool has its own measurement logic
- High cost of implementation
 - ▶ In case of new metrics, source code analysis has to be implemented in addition to logic of measurement
- Lack of compatibility
 - ▶ Existing tools can hardly handle multiple programming languages

Solutions

- Metrics measurement is completely separated from source code analysis
 - ▶ Users easily implement their original logic
 - ▶ MASU stores the result of analysis to internal database
 - ▶ Users only have to implement measurement logic as a plug-in
- MASU handles multiple programming languages
 - ▶ Java and C#
- For instance, a RFC plug-in is presented as below:

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

public class RfcPlugin extends AbstractClassMetricPlugin {
    protected Number measureClassMetric(TargetClassInfo targetClass) {
        final Set<CallableUnitInfo> rfcMethods = new HashSet<CallableUnitInfo>();
        // gets defined methods
        final Set<TargetMethodInfo> localMethods = targetClass.getDefinedMethods();
        rfcMethods.addAll(localMethods);
        // gets called methods from local methods
        for (final TargetMethodInfo m : localMethods) {
            rfcMethods.addAll(MethodCallInfo.getCalleees(m.getCalls()));
        }
        return new Integer(rfcMethods.size());
    }
    protected String getDescription() { return "Measuring the RFC metric."; }
    protected String getMetricName() { return "RFC"; }
    protected boolean useFieldInfo() { return true; }
    protected boolean useMethodInfo() { return true; }
}
```



Development of Eclipse Front-end

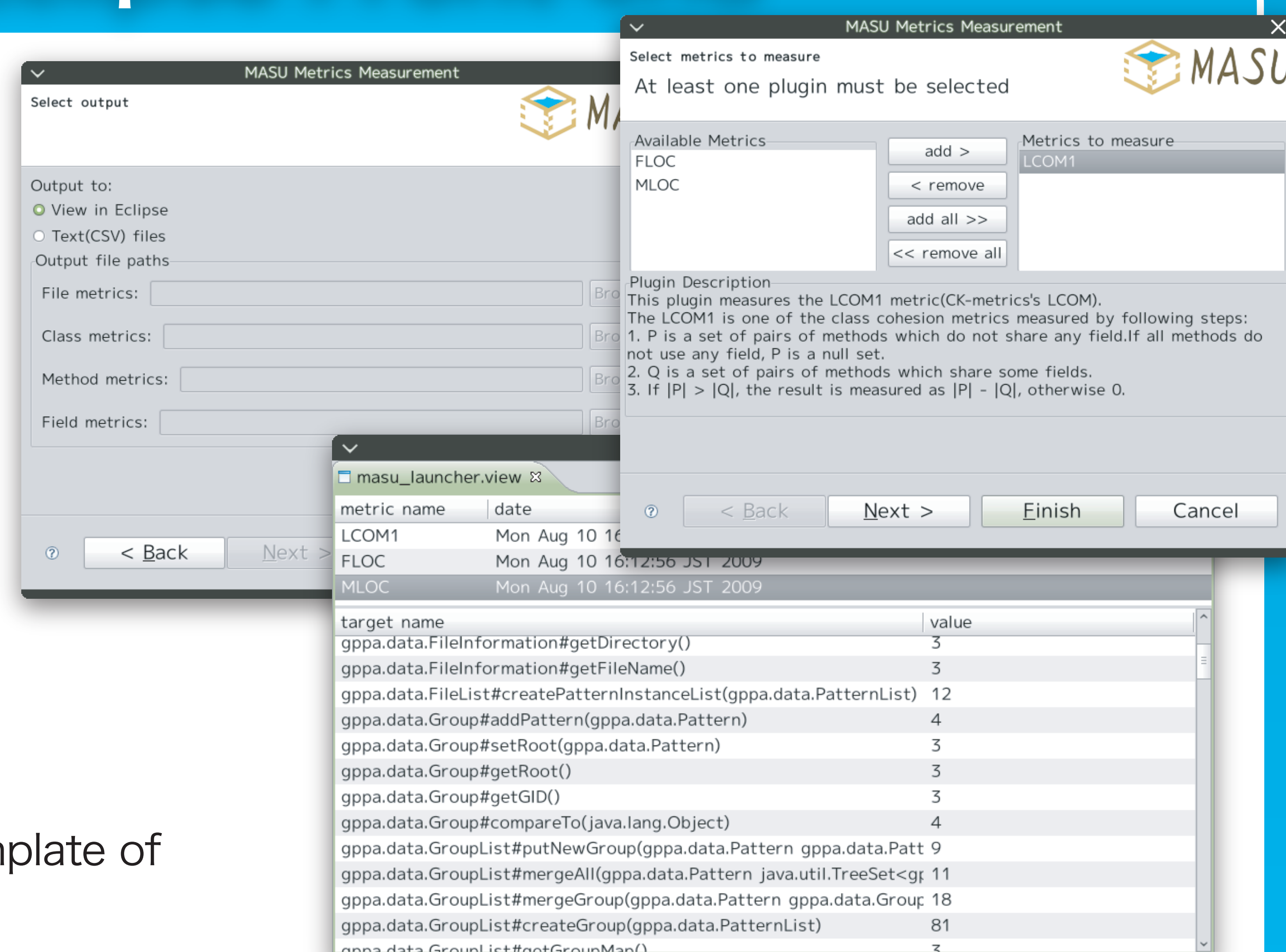
GUI front-end of MASU

- Easily measure the metrics from Eclipse

Features

- Selects files or directories that you want to measure the metrics
- Two types of output format
 - ▶ Outputs the result to view in Eclipse
 - ▶ Also shows difference of the result measured before and after
 - ▶ Exports the result to a text file

In the future work, we will implement the function of generating a template of MASU plug-ins



<http://sourceforge.net/projects/masu/>