Kieker in Eclipse
A Plug-in for Application Performance Monitoring and Dynamic Analysis in Eclipse

Symposium on Software Performance 2016

Florian Echternkamp and Christian Wulf

08.11.2016

Software Engineering Group
Kiel University, Germany
• Aspect-oriented programming (AspectJ)
• META-INF/aop.xml: excludes, aspects, pointcuts

```xml
<!-- DOCTYPE aspectj PUBLIC "-//AspectJ//DTD//EN" "http://www.aspectj.org/dtd/aspectj_1_5_0.dtd">
<aspectj>
  <weaver options="-showWeaveInfo">
    <exclude within="org.apache.commons.logging..*" />
    <exclude within="org.slf4j..*" />
    <exclude within="java.util.logging..*" />
    <exclude within="org.apache.log4j..*" />
  </weaver>

  <aspects>
    <aspect name="kieker.monitoring.probe.aspectj.flow.operationExecution.FullInstrumentationNoGetterAndSetter" />

    <concrete-aspect name="my.TargetedAspect"
      extends="kieker.monitoring.probe.aspectj.flow.operationExecution.AbstractAspect">
      <pointcut name="monitoredOperation" expression="execution(int kiekerFlow.TestFlow.getA())" />
    </concrete-aspect>
  </aspects>
</aspectj>
```
Monitoring with Kieker

- META-INF/kieker.properties: writer configuration

```properties
kieker.monitoring.writer=kieker.monitoring.writer.tcp.TCPWriter
kieker.monitoring.writer.tcp.TCPWriter.hostname=localhost
kieker.monitoring.writer.tcp.TCPWriter.port1=10133
kieker.monitoring.writer.tcp.TCPWriter.port2=10134
kieker.monitoring.writer.tcp.TCPWriter.bufferSize=65535
kieker.monitoring.writer.tcp.TCPWriter.flush=false
kieker.monitoring.writer.tcp.TCPWriter.QueueSize=100
```

- Command line invocation:

```bash
java
   -javaagent:kieker-1.12_aspectj.jar
   -cp .; lib/*; ...
   ./application.Main
```
Agenda

• Introduction
• Monitoring with the Kieker Plugin
• Analyzing with the Kieker Plugin
• Work in progress
• Conclusions
Configuration in Eclipse

Kieker nature

New tab in Eclipse run configurations
Context Menu - Monitoring

- Run As
- Debug As
- Restore from Local History...
- Kieker Monitoring
  - Probes Configuration
  - Properties
- Kieker Analysis
- Team
- Compare With
- Source
- Properties
- Alt+Enter
Monitoring – Probe Configuration

**AspectU**

**Weaver**

Options
- [ ] verbose
- [ ] showWeaverInfo
- [ ] Xjoinpointasyncronization

**Includes**

Packages to instrument.
One line per include! When Includes is empty, everything will be included!

```
*
```

**Excludes**

Sub packages (from above) to exclude.
One line per exclude! Important: The logger library that is configured to be used by Kieker must always be excluded, so the following packages are automatically excluded:
- org.apache.commons.logging..
- org.slf4j..
- java.util.logging..
- org.apache.log4j..

**Probes**

- [ ] Predefined Probes
  - kieker.monitoring.probe.aspectj.flow.operationExecution.FullInstrumentation
- [ ] Custom Probe

[Save] [Reset]
Context Menu - Analysis

Kieker Monitoring

Kieker Analysis

Show Traces
Show Aggregated Traces
Show Operation Calls
Show Aggregated Operation Calls
Show Monitoring Log Statistics
Analysis – Aggregated Traces

Kieker Analysis

- Traces (48)
- Aggregated Traces (8)
- Operation Calls (55)
- Aggregated Operation Calls

Filter Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Component</th>
<th>Container</th>
<th>Calls</th>
<th>St</th>
</tr>
</thead>
<tbody>
<tr>
<td>onReceive(...)</td>
<td>StaticAnalyzer</td>
<td>Florian-NB</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>onReceive(...)</td>
<td>JavaFile2AstWorker</td>
<td>Florian-NB</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>onReceive(...)</td>
<td>HtmlTableBuilderWorker</td>
<td>Florian-NB</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>exportHtml(...)</td>
<td>HtmlTableBuilderWorker</td>
<td>Florian-NB</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>onReceive(...)</td>
<td>FileWriterWorker</td>
<td>Florian-NB</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>onReceive(...)</td>
<td>Dir2JavaFilesWorker</td>
<td>Florian-NB</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>onReceive(...)</td>
<td></td>
<td>Florian-NB</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Open File in Editor

HtmTableBuilderWorker.java

```java
public void exportHtml() {
    try {
        html._table();
    } catch (Exception e) {
        e.printStackTrace();
    }
    fileWriter.tell(new FileWriterMessage(html.toHtml(), html));
}
```
WIP: Trace Visualization

![Trace Visualization Diagram]

- Entry
- SE::@3.kieker.examples.bookstore.Bookstore searchBook(...)
- SE::@1.kieker.examples.bookstore.Catalog getBook(...)
- SE::@2.kieker.examples.bookstore.CRM getOffers(...)
- SE::@1.kieker.examples.bookstore.catalog getBook(...)

Florian Echternkamp & Christian Wulf — 08.11.2016
Colored mean execution time per component execution
Colored overall execution time per component execution
Time scale showing a filter interval in red
WIP: Timing Behavior Diagram

: FileHandler

: FileEncrypter

: FSWriter

Begin Time: 2872ms
End Time: 3876ms
Execution Time: 1004ms
Conclusions

- Open source Eclipse plugin\(^1\) to ease monitoring and analysis with Kieker
- Abstracts from properties’ file location and format
- Embeds analysis views

http://kieker-monitoring.net

Work in progress
- Trace visualization
- Execution behavior diagram
- Timing behavior diagram

http://teetime.sourceforge.net

\(^1\) https://build.se.informatik.uni-kiel.de/kieker/kieker-eclipse-plugin