



A Framework for Application Performance Monitoring and Dynamic Software Analysis

—*Preview for Invited Demo/Poster Presentation*—

André van Hoorn, J. Waller, and W. Hasselbring

Software Engineering Group
University of Kiel, Germany

April 24, 2012 @ ICPE '12, Boston, MA



Kieker: Example Workflow and Use Cases

Preview for Invited Demo/Poster Presentation

The screenshot shows the Eclipse IDE interface with a Java file named Catalog.java open. The code contains a try-catch block that logs a message to a logger if an exception occurs. A red box highlights the logger call, and a black box highlights the word "Monitoring probe". To the right of the code editor is the Kieker AspectJ editor, which displays a tree structure of monitoring probes and their contexts. A blue box highlights the "Monitoring probe" node in this tree.

```
28:     @Aspect
29:     @Log(when = Log.Level.USER)
30:     public void logUserOperation() {
31:         try {
32:             // ...
33:         } catch (InterruptedException ex) {
34:             logger.error(ex);
35:         }
36:     }
37: }
```

Java - ch5-trace-monitoring-aspect/inc/bookstoreTracing/Catalog.java - Eclipse

Catalog.java

Monitoring probe

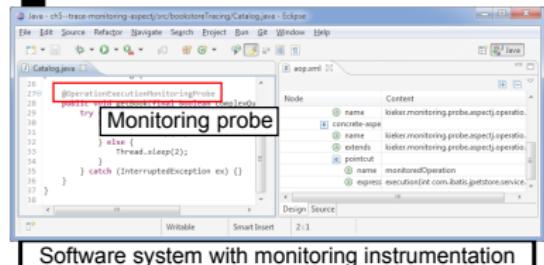
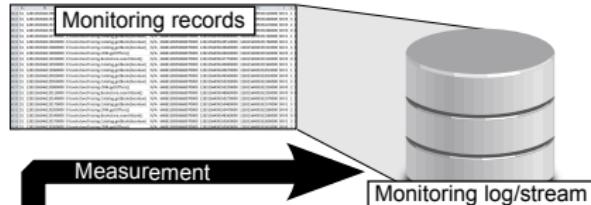
aspects

Node	Context
④ name	kieker.monitoring.probe.aspectj.operation..
④ extends	kieker.monitoring.probe.aspectj.operation..
④ pointcut	kieker.monitoring.probe.aspectj.operation..
④ name	monitoredOperation
④ express	execution(int com.ibatis.jdbcstore.service..

Software system with monitoring instrumentation

Kieker: Example Workflow and Use Cases

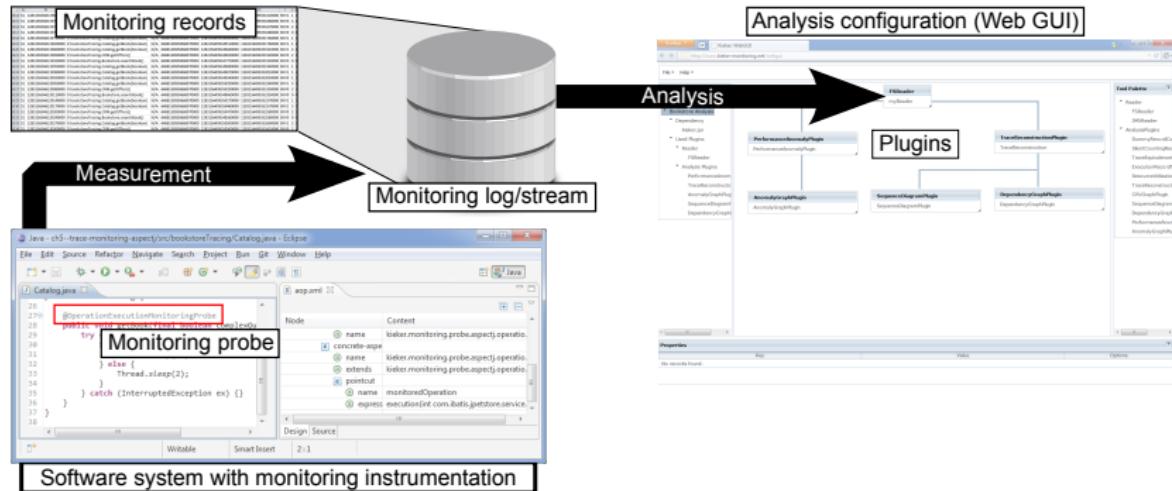
Preview for Invited Demo/Poster Presentation



Software system with monitoring instrumentation

Kieker: Example Workflow and Use Cases

Preview for Invited Demo/Poster Presentation

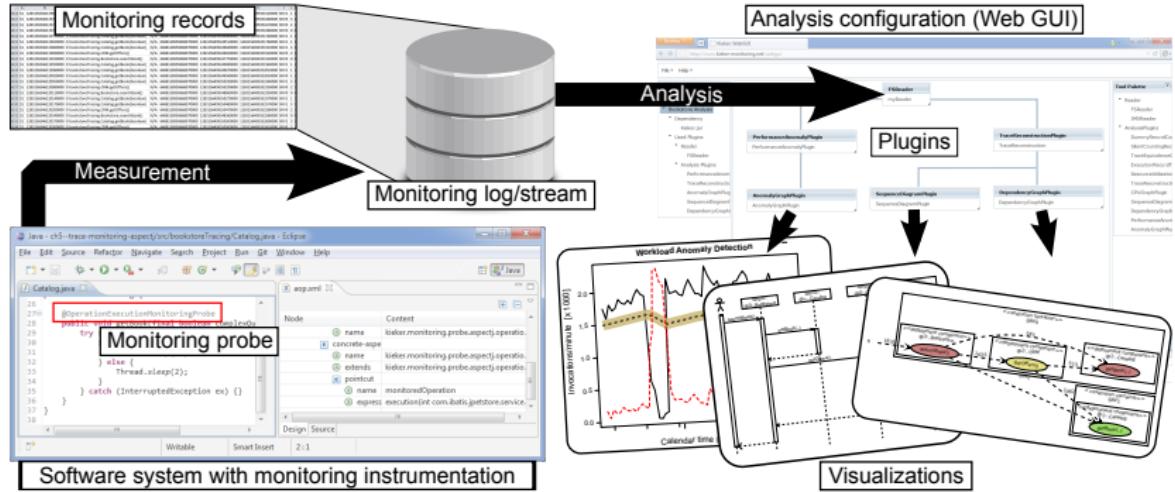


Kieker: Example Workflow and Use Cases

Preview for Invited Demo/Poster Presentation

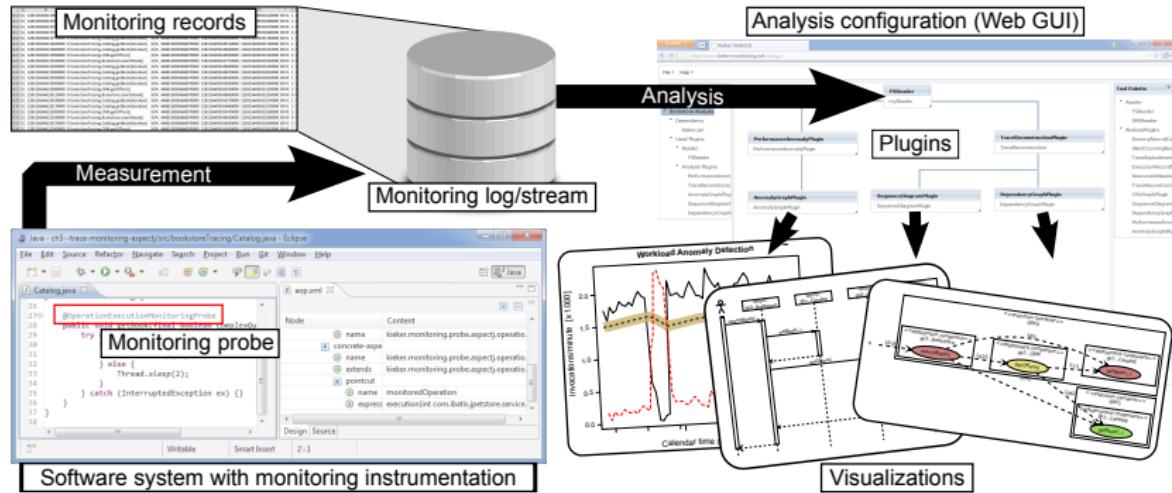


Christian-Albrechts-Universität zu Kiel



Kieker: Example Workflow and Use Cases

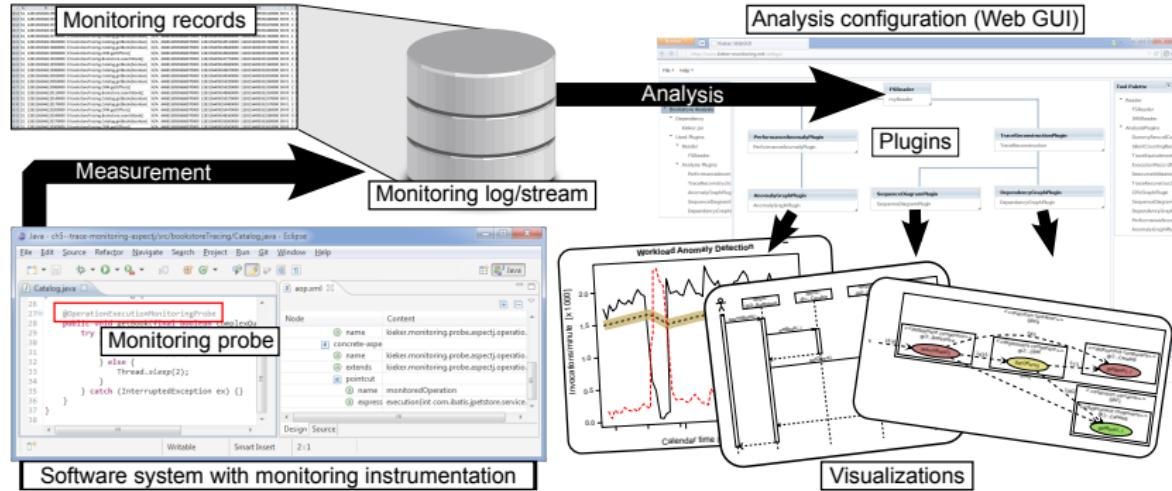
Preview for Invited Demo/Poster Presentation



Use cases in research and practice:

Kieker: Example Workflow and Use Cases

Preview for Invited Demo/Poster Presentation

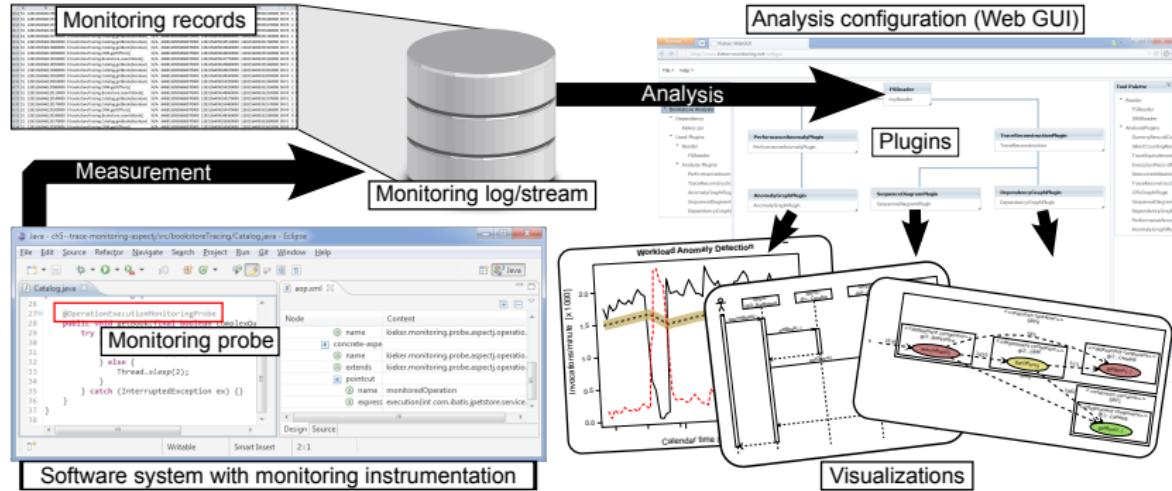


Use cases in research and practice:

- Online/offline performance evaluation and feedback, e.g.,

Kieker: Example Workflow and Use Cases

Preview for Invited Demo/Poster Presentation

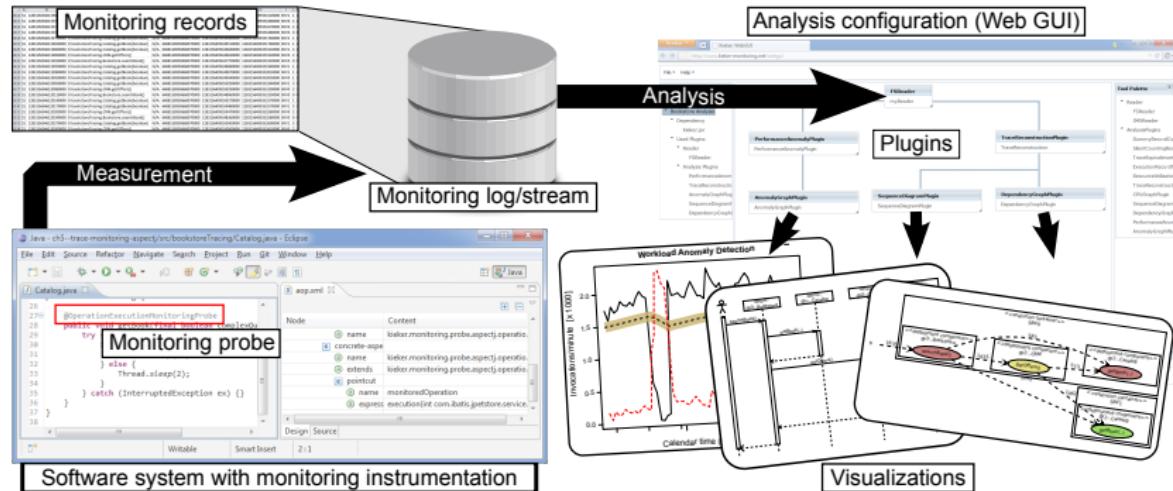


Use cases in research and practice:

- Online/offline performance evaluation and feedback, e.g.,
 - Continuous monitoring of application behavior and usage

Kieker: Example Workflow and Use Cases

Preview for Invited Demo/Poster Presentation

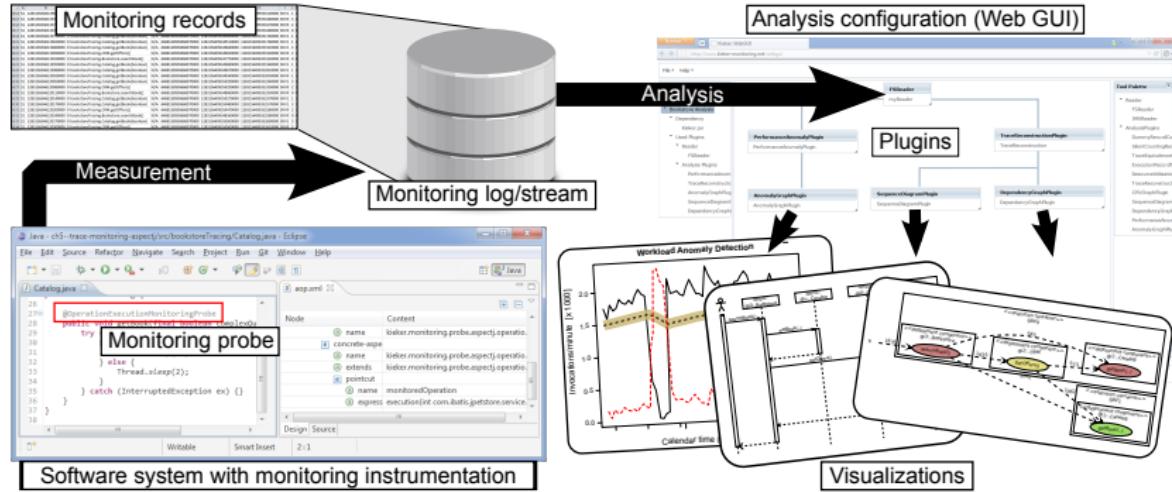


Use cases in research and practice:

- Online/offline performance evaluation and feedback, e.g.,
 - Continuous monitoring of application behavior and usage
 - Performance anomaly detection and diagnosis

Kieker: Example Workflow and Use Cases

Preview for Invited Demo/Poster Presentation

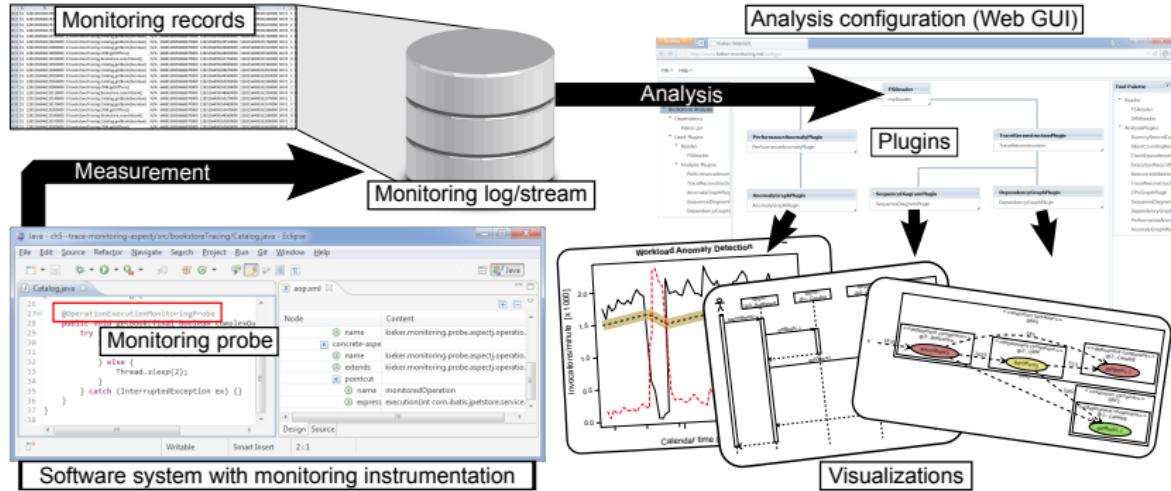


Use cases in research and practice:

- Online/offline performance evaluation and feedback, e.g.,
 - Continuous monitoring of application behavior and usage
 - Performance anomaly detection and diagnosis
 - (Self-)adaptation control

Kieker: Example Workflow and Use Cases

Preview for Invited Demo/Poster Presentation

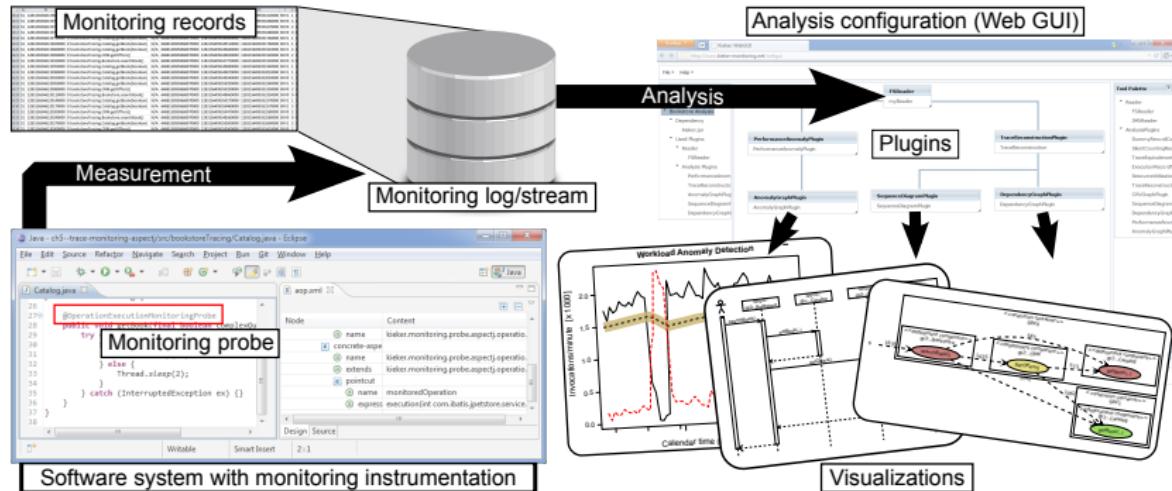


Use cases in research and practice:

- Online/offline performance evaluation and feedback, e.g.,
 - Continuous monitoring of application behavior and usage
 - Performance anomaly detection and diagnosis
 - (Self-)adaptation control
- Extraction of software architectural (performance) models and visualizations

Kieker: Example Workflow and Use Cases

Preview for Invited Demo/Poster Presentation



Use cases in research and practice:

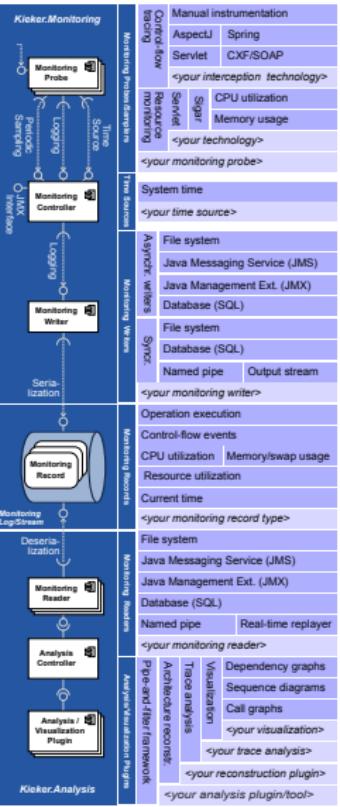
- Online/offline performance evaluation and feedback, e.g.,
 - Continuous monitoring of application behavior and usage
 - Performance anomaly detection and diagnosis
 - (Self-)adaptation control
- Extraction of software architectural (performance) models and visualizations
- Simulation (replaying previously monitored stimuli; measurement, logging, and analysis)

Characteristics, Features, Extension Points

Preview for Invited Demo/Poster Presentation

Kieker Framework

- Modular, flexible, and extensible architecture
(Probes, records, readers, writers, filters etc.)
- Pipes-and-filters framework for analysis configuration
- Distributed tracing (logging, reconstruction, visualization)
- Low overhead (designed for continuous operation)
- Evaluated in lab and industrial case studies (since 2006)



Characteristics, Features, Extension Points

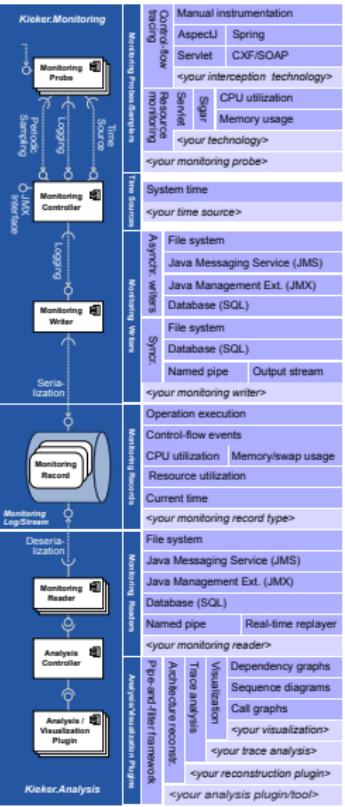
Preview for Invited Demo/Poster Presentation

Kieker Framework

- Modular, flexible, and extensible architecture
(Probes, records, readers, writers, filters etc.)
- Pipes-and-filters framework for analysis configuration
- Distributed tracing (logging, reconstruction, visualization)
- Low overhead (designed for continuous operation)
- Evaluated in lab and industrial case studies (since 2006)

Kieker is open-source software (Apache License, V. 2.0)

<http://kieker-monitoring.net>



Characteristics, Features, Extension Points

Preview for Invited Demo/Poster Presentation

Kieker Framework

- Modular, flexible, and extensible architecture
(Probes, records, readers, writers, filters etc.)
- Pipes-and-filters framework for analysis configuration
- Distributed tracing (logging, reconstruction, visualization)
- Low overhead (designed for continuous operation)
- Evaluated in lab and industrial case studies (since 2006)

Kieker is open-source software (Apache License, V. 2.0)

<http://kieker-monitoring.net>

Kieker is distributed as part of SPEC® RG's repository of peer-reviewed tools for quantitative system evaluation and analysis

<http://research.spec.org/projects/tools.html>

