André van Hoorn$^1$ and Wilhelm Hasselbring$^2$

$^1$University of Stuttgart
Institute of Software Technology
Reliable Software Systems Group

$^2$Kiel University
Department of Computer Science
Software Engineering Group

Joint Kieker/Palladio Days (KP DAYS ’13)
November 27, 2013 @ Karlsruhe
Agenda

1. Motivation: Performance, Monitoring, Dynamic Analysis, ...

2. What is Kieker?

3. What is Kieker Good For?

4. Who has been/is Working on Kieker?
Dynamic Analysis with Kieker

What is Kieker?

Software system with monitoring instrumentation

Monitoring probe

Software system with monitoring instrumentation
Dynamic Analysis with Kieker

What is Kieker?

- Monitoring records
- Monitoring log/stream
- Measurement

Software system with monitoring instrumentation

A. van Hoorn, W. Hasselbring
Kieker 101
Nov. 27, 2013 @ KP DAYS, Karlsruhe
Dynamic Analysis with Kieker

What is Kieker?

- Monitoring records
- Analysis configuration (Web GUI)
- Analysis
- Plugins
- Monitoring log/stream
- Measurement
- Monitoring probe
- Software system with monitoring instrumentation
Dynamic Analysis with Kieker

What is Kieker?

Monitoring records

Measurement

Monitoring log/stream

Software system with monitoring instrumentation

Analysis configuration (Web GUI)

Analysis

Plugins

Visualizations

Workload Anomaly Detection

A. van Hoorn, W. Hasselbring

Kieker 101

Nov. 27, 2013 @ KP DAYS, Karlsruhe
Core Kieker Framework Components

What is Kieker?

- Monitoring Probe
- Monitoring Controller
- Monitoring Writer
- Monitoring Log/Stream
- Kieker.Analysis

Java probes/samplers:
- Manual instrumentation
- Spring
- AspectJ
- CXF/SOAP
- Servlet
- Sigar
- CPU utilization
- Memory usage

+ basic adapters for
  - C#/.NET
  - Visual Basic 6/COM
  - COBOL
Core Kieker Framework Components

What is Kieker?

Java probes/samplers:
- Manual instrumentation
- AspectJ
- Spring
- Servlet
- CXF/SOAP
- <your interception technology>
- <your monitoring probe>

+ basic adapters for
  - C#/.NET
  - Visual Basic 6/COM
  - COBOL

Kieker.Monitoring

Kieker.Analysis

Monitoring Log/Stream

A. van Hoorn, W. Hasselbring
Kieker 101
Nov. 27, 2013 @ KPDAYS, Karlsruhe
What is Kieker?

Core Kieker Framework Components

- **Kieker.Monitoring**
  - Monitoring Probe
  - Monitoring Controller
  - Logging
  - Periodic Sampling
  - JMX Interface
  - Time Source

- **Monitoring Log/Stream**
  - Monitoring Record

- **Kieker.Analysis**
  - Pipe & Filter Configuration
  - Monitoring Reader
  - Analysis / Visualization Plugin
  - Analysis Controller
What is Kieker?

Kieker.Monitoring
- Monitoring Probe
- Monitoring Controller
- Monitoring Writer
- JMX Interface
- Periodic Sampling
- Logging
- Time Source

Monitoring Log/Stream
- Monitoring Record

Kieker.Analysis
- Pipe & Filter Configuration
- Monitoring Reader
- Analysis / Visualization Plugin
- Analysis Controller

Monitoring Readers/Writers
- File system
- Java Messaging Service (JMS)
- Java Management Ext. (JMX)
- Database (SQL)
- Named pipe
- <your monitoring reader/writer>
Core Kieker Framework Components

What is Kieker?

Kieker.Monitoring

- Monitoring Probe
- Monitoring Controller
- Monitoring Writer
- JMX Interface
- Periodic Sampling
- Logging
- Time Source

Monitoring Log/Stream

- Monitoring Record

Kieker.Analysis

- Pipe & Filter Configuration
- Monitoring Reader
- Analysis / Visualization Plugin
- Analysis Controller

Monitoring Readers/ Writers

- File system
- Java Messaging Service (JMS)
- Java Management Ext. (JMX)
- Database (SQL)
- Named pipe
- <your monitoring reader/writer>
Core Kieker Framework Components

What is Kieker?

Kieker.Monitoring
- Monitoring Probe
- Monitoring Controller
- Monitoring Writer
- JMX Interface
- Periodic Sampling
- Logging
- Time Source

Monitoring Log/Stream
- Monitoring Record

Kieker.Analysis
- Monitoring Reader
- Pipe & Filter Configuration
- Analysis / Visualization Plugin
- Analysis Controller

Monitoring Readers/Writers
- File system
- Java Messaging Service (JMS)
- Java Management Ext. (JMX)
- Database (SQL)
- Named pipe
- <your monitoring reader/writer>
Core Kieker Framework Components

What is Kieker?

- **Kieker.Monitoring**
  - Monitoring Probe
  - Monitoring Controller
  - Monitoring Writer
  - Monitoring Log/Stream
  - JMX Interface
  - Periodic Sampling
  - Logging
  - Time Source

- **Kieker.Analysis**
  - Monitoring Reader
  - Analysis / Visualization Plugin
  - Pipe & Filter Configuration
  - Trace analysis
  - Architecture reconstruction
  - Visualization
  - Dependency graphs
  - Sequence diagrams
  - Call graphs
  - <your visualization>
  - <your trace analysis>
  - <your reconstruction plugin>
  - <your analysis plugin/tool>

- Analysis Controller
Example Pipe-and-Filter Configuration

What is Kieker?

A. van Hoorn, W. Hasselbring
Kieker 101
Nov. 27, 2013 @ KPDAYS, Karlsruhe
Example Pipe-and-Filter Configuration

What is Kieker?

Kieker.Analysis example pipes-and-filters configuration
Performance anomaly detection and visualization
... FS reader
<<Filter>>
: Performance anomaly filter
operationExecutions
systemModel
anomalyRatings

<<Reader>>
: FS reader
outputPort

A. van Hoorn, W. Hasselbring
Kieker 101
Nov. 27, 2013 @ KPDAYS, Karlsruhe
What is Kieker?

Example Pipe-and-Filter Configuration

What is Kieker?

Kieker.Analysis example pipes-and-filters configuration
     Performance anomaly detection and visualization
     ... Performance anomaly filter
operationExecutions
systemModel
anomalyRatings
<<Repository>>
: System model repository

<<Reader>>
: FS reader
outputPort

<<Filter>>
: Performance anomaly filter

<<Repository>>
: System model repository

operationExecutions
systemModel
anomalyRatings
Example Pipe-and-Filter Configuration

What is Kieker?

- Kieker.Analysis example pipes-and-filters configuration
- Performance anomaly detection and visualization
- Architecture and trace reconstruction/visualization

<<Reader>>: FS reader

<<Filter>>: Performance anomaly filter

operationExecutions

systemModel

<<Repository>>: System model repository

anomalyRatings

<<Filter>>: Anomaly graph plotter

anomalyRatings

A. van Hoorn, W. Hasselbring

Kieker 101

Nov. 27, 2013 @ KPDAWS, Karlsruhe
Kieker Analysis example pipes-and-filters configuration

- Performance anomaly detection and visualization
- Architecture and trace reconstruction/visualization
Example Pipe-and-Filter Configuration

What is Kieker?

Kieker.Analysis example pipes-and-filters configuration
Performance anomaly detection and visualization

Architecture and trace reconstruction/visualization

outputPort<<Reader>>: FS reader
<<Repository>>: System model repository

<<Filter>>: Trace reconstruction filter
traceEvents systemModel
messageTraces
executionTraces

<<Filter>>:
: Performance anomaly filter
operationExecutions
anomalyRatings systemModel
anomalyRatings

<<Filter>>:
: Anomaly graph plotter

A. van Hoorn, W. Hasselbring
Kieker 101
Nov. 27, 2013 @ KP DAYS, Karlsruhe
What is Kieker?

Kieker.Analysis example pipes-and-filters configuration
- Performance anomaly detection and visualization
- Architecture and trace reconstruction/visualization

```
<<Reader>>
: FS reader

<<Repository>>
: System model repository

<<Filter>>
: Trace reconstruction filter
- messageTraces
- systemModel
- executionTraces

<<Filter>>
: Sequence diagram visualization
- messageTraces
- systemModel

<<Filter>>
: Dependency graph visualization
- messageTraces
- systemModel
```

A. van Hoorn, W. Hasselbring
Kieker 101
Nov. 27, 2013 @ KPDAYS, Karlsruhe
Kieker Analysis example pipes-and-filters configuration
- Performance anomaly detection and visualization
- Architecture and trace reconstruction/visualization
Welcome to the Kieker WebGUI
This is an early beta version of the Kieker WebGUI. Therefore it may contain bugs and some functionality may have not been implemented yet. Just click “Login” to continue.

Welcome to the Kieker WebGUI
Username: Guest
Password: Guest
Login

Kieker > Bookstore-Example
File > Help

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassPath</td>
<td>Kiekerexamples.userUBLIC.3.bookstore.MyJavaPkg.myFilter</td>
</tr>
<tr>
<td>Name</td>
<td>MyJavaPkg.myFilter</td>
</tr>
</tbody>
</table>

Analysis Plugins

- Reader
  - MyPkgReader
  - DTraceReader
  - JProfilerReader
  - PMDReader
  - JavaMemoryReader
- Filter
  - MyPkgFilter
  - MyJavaPkgFilter
  - MyJavaPkg.DTraceReaderFilter
  - MyJavaPkg.PMDReaderFilter
  - MyJavaPkg.JavaMemoryReaderFilter
  - MyJavaPkg.MyFilter
  - StringReaderFilter

Control

- Initialize Analysis
- Clean Analysis
- Start Analysis
- Stop Analysis

Kieker 101
Nov. 27, 2013 @ KP DAYS, Karlsruhe
1. Motivation: Performance, Monitoring, Dynamic Analysis, ...

2. What is Kieker?

3. What is Kieker Good For?

4. Who has been/is Working on Kieker?
What is Kieker Good For?

1. **Architecture Discovery** (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)
   - Trace-based architecture analysis

2. **Application Performance Management**
   - Continuous QoS monitoring + feedback (self-*)
   - Distributed tracing and trace-based analysis
   - Architecture-based performance analysis
   - Automatic problem detection and diagnosis
   - Extraction of usage profiles (workload intensity, navigational patterns)

Characteristics (cross-cutting)
- Modular, flexible, and extensible architecture
- Non-intrusive instrumentation
- Low performance overhead
- Model-driven instrumentation and analysis
- Evaluated in lab and industrial case studies
What is Kieker Good For?

1. **Architecture Discovery** *(Dynamic/Hybrid Analysis)*
   - Extraction of *architectural models* *(structure, behavior)*

2. **Application Performance Management**
Kieker Use Cases/Characteristics

What is Kieker Good For?

1. **Architecture Discovery** (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems

2. **Application Performance Management**

---

A. van Hoorn, W. Hasselbring
Kieker 101
Nov. 27, 2013 @ KPDAWS, Karlsruhe
### What is Kieker Good For?

1. **Architecture Discovery** *(Dynamic/Hybrid Analysis)*
   - Extraction of *architectural models* *(structure, behavior)*
   - Reverse engineering of legacy systems
   - Software *visualization* *(2D/3D)*

2. **Application Performance Management**
What is Kieker Good For?

1. **Architecture Discovery** (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)
   - Trace-based architecture analysis

2. **Application Performance Management**

   • Continuous QoS monitoring + feedback (self-*)
   • Distributed tracing and trace-based analysis
   • Architecture-based performance analysis
   • Automatic problem detection and diagnosis
   • Extraction of usage profiles (workload intensity, navigational patterns)
### What is Kieker Good For?

1. **Architecture Discovery** *(Dynamic/Hybrid Analysis)*
   - Extraction of *architectural models* *(structure, behavior)*
   - Reverse engineering of legacy systems
   - Software *visualization* *(2D/3D)*
   - Trace-based *architecture analysis*

2. **Application Performance Management**
   - Continuous *QoS monitoring* + feedback *(self-*)

### Characteristics (cross-cutting)
- Modular, flexible, and extensible architecture
- Non-intrusive instrumentation
- Low performance overhead
- Model-driven instrumentation and analysis
- Evaluated in lab and industrial case studies
1. **Architecture Discovery** (Dynamic/Hybrid Analysis)
   - Extraction of *architectural models* (structure, behavior)
   - Reverse engineering of legacy systems
   - Software *visualization* (2D/3D)
   - Trace-based architecture analysis

2. **Application Performance Management**
   - Continuous QoS monitoring + feedback (*self-* )
   - Distributed tracing and trace-based analysis
Kieker Use Cases/Characteristics

What is Kieker Good For?

1. **Architecture Discovery** (Dynamic/Hybrid Analysis)
   - Extraction of **architectural models** (structure, behavior)
   - Reverse engineering of legacy systems
   - Software **visualization** (2D/3D)
   - Trace-based **architecture analysis**

2. **Application Performance Management**
   - Continuous **QoS monitoring** + feedback (**self-***) 
   - Distributed **tracing** and trace-based analysis
   - Architecture-based **performance analysis**
What is Kieker Good For?

1. **Architecture Discovery** (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)
   - Trace-based architecture analysis

2. **Application Performance Management**
   - Continuous QoS monitoring + feedback (self-*)
   - Distributed tracing and trace-based analysis
   - Architecture-based performance analysis
   - Automatic problem detection and diagnosis

Characteristics (cross-cutting)
- Modular, flexible, and extensible architecture
- Non-intrusive instrumentation
- Low performance overhead
- Model-driven instrumentation and analysis
- Evaluated in lab and industrial case studies
Kieker Use Cases/Characteristics

What is Kieker Good For?

1. **Architecture Discovery** (Dynamic/Hybrid Analysis)
   - Extraction of **architectural models** (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)
   - Trace-based architecture analysis

2. **Application Performance Management**
   - Continuous QoS monitoring + feedback (**self-***)  
   - Distributed tracing and trace-based analysis
   - Architecture-based performance analysis
   - Automatic problem detection and diagnosis
   - Extraction of **usage profiles** (workload intensity, navigational patterns)

Characteristics (cross-cutting)

- Modular, flexible, and extensible architecture
- Non-intrusive instrumentation
- Low performance overhead
- Model-driven instrumentation and analysis
- Evaluated in lab and industrial case studies
What is Kieker Good For?

1. **Architecture Discovery** (Dynamic/Hybrid Analysis)
   - Extraction of architectural models (structure, behavior)
   - Reverse engineering of legacy systems
   - Software visualization (2D/3D)
   - Trace-based architecture analysis

2. **Application Performance Management**
   - Continuous QoS monitoring + feedback (self-*)
   - Distributed tracing and trace-based analysis
   - Architecture-based performance analysis
   - Automatic problem detection and diagnosis
   - Extraction of usage profiles (workload intensity, navigational patterns)

3. **Characteristics (cross-cutting)**
   - Modular, flexible, and extensible architecture
   - Non-intrusive instrumentation
   - Low performance overhead
   - Model-driven instrumentation and analysis
   - Evaluated in lab and industrial case studies
What is Kieker Good For?

Architecture Discovery: Model Extraction + Visualization

[van Hoorn et al. 2009]
What is Kieker Good For?

Architecture Discovery: Model Extraction + Visualization (cont’d)

[Richter 2012]
Architecture Discovery: Model Extraction + Visualization (cont’d)

[Magedanz 2011]
What is Kieker Good For?

Architecture Discovery: Model Extraction + Visualization (cont’d)
Architecture Discovery: Model Extraction + Visualization (cont’d)

[Döhring 2012] (based on [Wulf 2010])
What is Kieker Good For?

**APM**: Anomaly Detection + Diagnosis

[Bielefeld 2012, Frotscher 2013]
What is Kieker Good For?

**APM: Anomaly Detection + Diagnosis (cont’d)**

---

[Marwede et al. 2009]
1 Motivation: Performance, Monitoring, Dynamic Analysis, ...

2 What is Kieker?

3 What is Kieker Good For?

4 Who has been/is Working on Kieker?
Looking back ... 2006–2009

Who has been/is Working on Kieker?

2006

2007

2008

2009

Performance Monitoring von Middleware-basierten Applikationen

31. März 2006

Diplomarbeit

Performance Monitoring von Middleware-basierten Applikationen

31. März 2006

Diplomarbeit

Performance Monitoring von Middleware-basierten Applikationen

31. März 2006

Diplomarbeit
Looking back… 2006–2009

Who has been/is Working on Kieker?

2006

May

June

July

Aug.

Sept.

2007

May

June

July

Aug.

Sept.

2008

May

June

July

Aug.

Sept.

2009

May

June

July

Aug.

Sept.

A. van Hoorn, W. Hasselbring

Kieker 101

Nov. 27, 2013 @ KP DAYS, Karlsruhe  12 / 14
Looking back . . . 2006–2009

Who has been/is Working on Kieker?

A. van Hoorn, W. Hasselbring

Kieker 101

Nov. 27, 2013 @ KPDAYS, Karlsruhe
Looking back ... 2006–2009

Who has been/is Working on Kieker?

2006
- Nokia Siemens Networks

2007
- Trustworthy Software Systems

2008
- Trustworthy Software Systems
- SourceForge

2009
- Trustworthy Software Systems
- sourceforge
- cewe color

A. van Hoorn, W. Hasselbring

Kieker 101

Nov. 27, 2013 @ KPDAYS, Karlsruhe
Looking back ... 2010–2013

Who has been/is Working on Kieker?

2010

MENGES
KoSSe

JUnit

1.1

1.2

2011

2012

2013

May

May

May

May

Sept.

Sept.

Sept.

A. van Hoorn, W. Hasselbring

Kieker 101

Nov. 27, 2013 @ KPDAYS, Karlsruhe
Looking back . . . 2010–2013

Who has been/is Working on Kieker?

- 2010
  - May
  - Sept.

- 2011
  - May
  - Sept.

- 2012
  - May
  - Sept.

- 2013
  - May
  - Sept.

A. van Hoorn, W. Hasselbring
Looking back ... 2010–2013

Who has been/is Working on Kieker?

2010
- MENGES

2011
- DynaMod
- PubFlow
- JUnit
- FindBugs

2012
- DynaMod
- PubFlow
- git

2013
- DynaMod

regular meetings

FindBugs

1.3 1.4

A. van Hoorn, W. Hasselbring

Kieker 101

Nov. 27, 2013 @ KPDAYS, Karlsruhe
Looking back . . . 2010–2013

Who has been/is Working on Kieker?

2010

May
Sept.

2011

May
Sept.

2012

May
Sept.

2013

May
Sept.

Kieker 101

Nov. 27, 2013 @ KPDAYS, Karlsruhe

A. van Hoorn, W. Hasselbring
Looking back... 2010–2013

Who has been/is Working on Kieker?

For additional details on Kieker’s evolution, see the Kieker Days ’12 talk “Kieker: Overview, Review, and Outlook”
Various people contributed to Kieker in the past years.


—Alphabetic list of people who contributed in different form (source code, bug reports, promotion, etc) and intensity
Visit 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

About Kieker

The internal behavior of large-scale software systems cannot be determined on the basis of static (e.g., source code) analysis alone. Kieker provides complementary dynamic analysis capabilities, i.e., monitoring and analyzing a software system's runtime behavior — enabling Application Performance Monitoring and Architecture Discovery.

Kieker 101

A. van Hoorn, W. Hasselbring

Nov. 27, 2013 @ KPDAYS, Karlsruhe


For a comprehensive list of publications, talks, and theses about Kieker, visit: http://kieker-monitoring.net/research/