Advanced Typing for the Kieker Instrumentation Languages

SSP 2016, Kiel

Reiner Jung  Christian Wulf

8th November 2016
Motivation

Instrumentation of software systems

- Multiple technologies and languages
- Vast set of event types (currently 50)

Kieker instrumentation languages [Jung et al. 2013]

- Provide language and technology independent instrumentation
- Support
  - type declaration
    instrumentation record language (IRL)
  - probe specification & aspect weaving
    instrumentation aspect language (IAL)
Introduction

Motivation

Instrumentation of software systems
- Multiple technologies and languages
- Vast set of event types (currently 50)

Kieker instrumentation languages
- Provide language and technology independent instrumentation
- Support
  - type declaration
    instrumentation record language (IRL)
  - probe specification & aspect weaving
    instrumentation aspect language (IAL)

Two issues
- Extending multiple event types is cumbersome
- Instrumentation advice construction is error prone
Extending event types
Extending event types

Example - Extending trace records
Extending event types

Example - Extending trace records

IClassSignature    IOperationSignature    IFlowRecord

IConstructorRecord    IOperationRecord    ITraceRecord    IEwishEvent    IExceptionRecord    TraceMetadata

ICallRecord

AbstractEvent

AbstractTraceEvent

AbstractOperationEvent

CallOperationEvent    BeforeOperationEvent    AfterOperationEvent

CallConstructorEvent    BeforeConstructorEvent    AfterConstructorEvent    AfterOperationFailedEvent

AfterConstructorFailedEvent

IRequestSize

Reiner Jung
Extending event types

Example - Extending trace records
Extending event types

Extending trace records - using the IRL

template IRequestSize {
    long size
}

entity RS_CallOperationEvent extends CallOperationEvent : IRequestSize
entity RS_BeforeOperationEvent extends BeforeOperationEvent : IRequestSize
entity RS_AfterOperationEvent extends AfterOperationEvent : IRequestSize
entity RS_AfterOperationFailedEvent extends AfterOperationFailedEvent : IRequestSize
entity RS_CallConstructorEvent extends CallConstructorEvent : IRequestSize
entity RS_BeforeConstructorEvent extends BeforeConstructorEvent : IRequestSize
entity RS_AfterConstructorEvent extends AfterConstructorEvent : IRequestSize
entity RS_AfterConstructorFailedEvent extends AfterConstructorFailedEvent : IRequestSize
Extending event types

Extending trace records with model types

based on model types introduced by Steel et al. 2007
Extending event types

Extending trace records with model types

based on model types introduced by Steel et al. 2007
Extending event types

Extending trace records with model types

based on model types introduced by Steel et al. 2007
Extending event types

Extending trace records with model types

KiekerTraceEvents

based on model types introduced by Steel et al. 2007

Reiner Jung
Extending event types

Extending trace records with model types

Template based

model KiekerTraceEvents AbstractTraceEvent

template IRequestSize {
  long size
}

sub RS KiekerTraceEvents IRequestSize
Extending event types

Extending trace records with model types

**Template based**

```java
model KiekerTraceEvents AbstractTraceEvent

template IRequestSize {
  long size
}

sub RS KiekerTraceEvents IRequestSize
```

**Attribute based**

```java
model KiekerTraceEvents AbstractTraceEvent

sub RS KiekerTraceEvents {
  long size
}
```
Instrumentation advice construction
**Instrumentation advice construction**

**Old advice declaration**

---

**Event type declaration**

```java
entity BeforeOperationEvent {
    long timestamp
    string classSignature
    string operationSignature
    long traceId
    int orderIndex
}
```

**Advice declaration**

```java
advice LogOperationInvocation() {
    before BeforeOperationEvent(time, $clsSignature, $opSignature, traceId, orderIndex)
    after AfterOperationEvent(time, $opSignature, $clsSignature, traceId, orderIndex)
}
```
Event type declaration

```java
entity BeforeOperationEvent {
    long timestamp
    string classSignature
    string operationSignature
    long traceId
    int orderIndex
}
```

Advice declaration

```java
advice LogOperationInvocation() {
    before BeforeOperationEvent(time, $clsSignature, $opsSignature, traceId, orderIndex)
    after AfterOperationEvent(time, $opsSignature, $clsSignature, traceId, orderIndex)
}
```
Instrumentation advice construction

Utilizing semantic annotations

Event type declaration

```java
entity BeforeOperationEvent {
    long timestamp : time
    string classSignature : class-signature
    string operationSignature : operation-signature
    long traceId : trace-id
    int orderIndex : order-index
}
```

Advice declaration

```java
advice LogOperationInvocation() {
    before BeforeOperationEvent(time, $clsSignature, $opsSignature, traceId, orderIndex)
    after AfterOperationEvent(time, $opsSignature, $clsSignature, traceId, orderIndex)
}
```
Instrumentation advice construction

Utilizing semantic annotations

Event type declaration

```java
entity BeforeOperationEvent {
    long timestamp : time
    string classSignature : class-signature
    string operationSignature : operation-signature
    long traceId : trace-id
    int orderIndex : order-index
}
```

Advice declaration

```java
advice LogOperationInvocation() {
    before BeforeOperationEvent
    after AfterOperationEvent
}
```
Realization
Realization

Architecture
Realization
Architecture

- Generator Provider
- Generator IAL
- Model Mapper
- Probe Gen. Provider
- Configuration Gen. Provider
- Semantics Extension
- C Generator
- Perl Generator
- Java Generator
- C Semantics
- Java Semantics
- Perl Semantics
Realization
Architecture

Reiner Jung
Realization

Architecture
Realization

Model transformation

IRL with Model Types \rightarrow IAL
Realization

Model transformation

IRL with Model Types

Semantic Annotations

IAL
Realization

Model transformation

- IRL with Model Types
- Semantic Annotations
- IAL

T_{Model Types}

IRL

T_{IRL(lang)}

Event Type
Model transformation
Conclusion
Conclusion

Summary

- Entity type extensions mechanism based on model types
- Semantic properties for save probe construction

Future Work

- Complete PCM and Java IAL support
- Adaptation of IAL generators to new API

Github

https://github.com/kieker-monitoring/instrumentation-languages

Eclipse update site - snapshot

https://build.se.informatik.uni-kiel.de/eus/mdm/snapshot/
References