MAMBA
Model-Based Software Analysis
Utilizing OMG’s SMM
WSR 2012

Sören Frey  André van Hoorn  Reiner Jung
Benjamin Kiel  Wilhelm Hasselbring

Christian-Albrechts-University, Kiel
Software Engineering Group

3.05.2012
Contents

1. Motivation

2. Structured Metrics Meta-Model

3. MAMBA

4. Conclusions
**Goal:** Model-Based Software Analysis with SMM [Obj12]
Goal: Model-Based Software Analysis with SMM [Obj12]

- Re-usable metrics
Goal: Model-Based Software Analysis with SMM [Obj12]

- Re-usable metrics
- Use of metrics across technology boundaries
Goal: Model-Based Software Analysis with SMM [Obj12]

- Re-usable metrics
- Use of metrics across technology boundaries
- Combination of static and dynamic analysis
Motivation

**Goal:** Model-Based Software Analysis with SMM [Obj12]
- Re-usable metrics
- Use of metrics across technology boundaries
- Combination of static and dynamic analysis

Solution
Motivation

Goal: Model-Based Software Analysis with SMM [Obj12]
- Re-usable metrics
- Use of metrics across technology boundaries
- Combination of static and dynamic analysis

Solution
- OMG’s Structured Metrics Meta-Model (SMM)
**Goal:** Model-Based Software Analysis with SMM [Obj12]

- Re-usable metrics
- Use of metrics across technology boundaries
- Combination of static and dynamic analysis

**Solution**

- OMG’s Structured Metrics Meta-Model (SMM)
- Measurement Architecture for Model-Based Analysis (MAMBA) [FvHJ+11]

http://mamba-framework.sf.net/
Motivation

Bookstore - Example

- Catalog
  - `getBook : Book`

- Bookstore
  - `searchBook : List<Book>`

- CRM
  - `getOrders : List<Order>`

Example Metrics

- Number of classes
- Average response time of methods (here `searchBook`)

Frey, van Hoorn, Jung, Kiel, Hasselbring
Bookstore - Example

Motivation

Example Metrics

- **Catalog**
  - `getBook : Book`

- **Bookstore**
  - `searchBook : List<Book>`

- **CRM**
  - `getOrders : List<Order>`

Example Metrics

- Number of classes
- Average response time of methods

Frey, van Hoorn, Jung, Kiel, Hasselbring
Example Metrics

- Number of classes
Bookstore - Example

Motivation

Example Metrics

- Number of classes
- Average response time of methods
Bookstore - Example

Motivation

Example Metrics

- Number of classes
- Average response time of methods (here `searchBook`)

```
<<component>>
Catalog
getBook : Book

<<component>>
Bookstore
searchBook : List<Book>

<<component>>
CRM
getOrders : List<Order>

AverageResponseTime
```
Composition

- Definition of measures
Composition

- Definition of measures
- Definition of observations
Composition

- Definition of measures
- Definition of observations
- Collection of measurement results
Composition
- Definition of measures
- Definition of observations
- Collection of measurement results

Measures
- Direct measures, like a ClassRecognizer
Composition

- Definition of measures
- Definition of observations
- Collection of measurement results

Measures

- Direct measures, like a ClassRecognizer
- Derived measures, like a NumberOfClasses
Composition
- Definition of measures
- Definition of observations
- Collection of measurement results

Measures
- Direct measures, like a ClassRecognizer
- Derived measures, like a NumberOfClasses
- Closed: rely only on SMM and observed model
Composition
- Definition of measures
- Definition of observations
- Collection of measurement results

Measures
- Direct measures, like a ClassRecognizer
- Derived measures, like a NumberOfClasses
- Closed: rely only on SMM and observed model
- Open: require additional external data (e.g. monitoring data)
Measuring with SMM

Structured Metrics Meta-Model

bookstore.smm :MeasureLibrary

:CollectiveMeasure
name="NumberOfClasses"
accumulator=sum
Measuring with SMM

Structured Metrics Meta-Model

bookstore.smm

:MeasureLibrary

:CollectiveMeasure
name="NumberOfClasses"
accumulator=sum

:Scope
class="kdm.code.CodeModel"
Measuring with SMM

Structured Metrics Meta-Model

```
bookstore.smm

:MeasureLibrary

:CollectiveMeasure
name="NumberOfClasses"
accumulator=sum

:BaseMeasureRelationship
from
:Counting
name="ClassRecognizer"
to
:Scope
class="kdm.code.CodeModel"

Frey, van Hoorn, Jung, Kiel, Hasselbring
MAMBA
3.05.2012 6 / 16
```
Measuring with SMM

Structured Metrics Meta-Model

bookstore.smm

:MeasureLibrary

:CollectiveMeasure
name="NumberOfClasses"
accumulator=sum

:BaseMeasureRelationship
from

:Counting
name="ClassRecognizer"

to

:Scope
class="kdm.code.CodeModel"

:Scope
class="kdm.code.Package"
Measuring with SMM

Structured Metrics Meta-Model

```
:書store.smm :MeasureLibrary

from

:CollectiveMeasure
name="NumberOfClasses"
accumulator=sum

:BaseMeasureRelationship

:Counting
name="ClassRecognizer"

:Scope
class="kdm.code.CodeModel"

:Scope
class="kdm.code.Package"

:Operation
language="OCL"
body="isOCLTypeOf(kdm.code.ClassUnit)"
```
Measuring with SMM

Structured Metrics Meta-Model

**Operation**

```ocl
isOCLTypeOf(kdm.code.ClassUnit)
```

**Scope**

- **class**: kdm.code.Package

**Counting**

**name**: ClassRecognizer

**CollectiveMeasure**

- **name**: NumberOfClasses
- **accumulator**: sum

**CollectiveMeasure**

- **name**: AverageMethodResponseTime
- **accumulator**: average

**BaseMeasureRelationship**

**from**: bookstore.smm

**to**: kdm.code.CodeModel

**MeasureLibrary**

- **name**: NumberOfClasses
- **accumulator**: sum
- **name**: AverageMethodResponseTime
- **accumulator**: average

---

Frey, van Hoorn, Jung, Kiel, Hasselbring

MAMBA

3.05.2012
Measuring with SMM

Structured Metrics Meta-Model

```
bookstore.smm

:CollectiveMeasure
name="NumberOfClasses"
accumulator=sum
from
:BaseMeasureRelationship
name="ClassRecognizer"
scope
:Counting
name="ClassRecognizer"

:CollectiveMeasure
name="AverageMethodResponseTime"
accumulator=average
from
:BaseMeasureRelationship
class="kdm.code.CodeModel"
to
:Scope
class="kdm.code.Package"

:NamedMeasure
name="kieker.KDMMethodResponseTime"
operation
language="OCL"
body="isOCLTypeOf(kdm.code.ClassUnit)"
```

Frey, van Hoorn, Jung, Kiel, Hasselbring

MAMBA

3.05.2012
Measuring with SMM

Structured Metrics Meta-Model

bookstore.smm

:MeasureLibrary

:CollectiveMeasure
name="NumberOfClasses"
accumulator=sum

:CollectiveMeasure
name="AverageMethodResponseTime"
accumulator=average

:BaseMeasureRelationship
from
:Counting
name="ClassRecognizer"
scope
class="kdm.code.ClassUnit"

:BaseMeasureRelationship
to
:Scope
class="kdm.code.CodeModel"

:BaseMeasureRelationship
from
:Scope
class="kdm.code.ClassUnit"
scope
:MeasureLibrary
:BaseMeasureRelationship
to
:BaseMeasureRelationship
from
:Operation
language="OCL"
body="isOCLTypeOf(kdm.code.ClassUnit)"

name="kieker.KDMMethodResponseTime"

:NamedMeasure

Frey, van Hoorn, Jung, Kiel, Hasselbring
Measuring with SMM

Structured Metrics Meta-Model

```ocl
definition language="OCL"
body="isOCLTypeOf(kdm.code.ClassUnit)"
end definition

definition language="OCL"
body="kdm.code.ClassUnit.codeElement->select( method : kdm.code.MethodUnit | method.name = ${methodName})"
end definition
```

```ocl
definition language="OCL"
body="kdm.code.ClassUnit.codeElement->select( method : kdm.code.MethodUnit | method.name = ${methodName})"
end definition
```

```ocl
definition language="OCL"
body="kdm.code.ClassUnit.codeElement->select( method : kdm.code.MethodUnit | method.name = ${methodName})"
end definition
```
Measuring with SMM

Structured Metrics Meta-Model

```
bookstore.smm

:MeasureLibrary

:BaseMeasureRelationship

:Counting

:Scope

:Operation

language="OCL"
body="isOCLTypeOf(kdm.code.ClassUnit)"
```

```
:CollectiveMeasure
name="NumberOfClasses"
accumulator=sum
```

```
:CollectiveMeasure
name="AverageMethodResponseTime"
accumulator=average
```

```
:Scope
class="kdm.code.CodeModel"
```

```
:Scope
class="kdm.code.ClassUnit"
```

```
:NamedMeasure
name="kieker.KDMMethodResponseTime"
```

```
:NamedMeasure
name="ClassRecognizer"
```

```
:BaseMeasureRelationship
from
:Scope
class="kdm.code.Package"
scope
:Counting
name="ClassRecognizer"
scope
:Operation
language="OCL"
body="isOCLTypeOf(kdm.code.ClassUnit)"
```

```
:Operation
language="OCL"
body="kdm.code.ClassUnit.codeElement->select( method : kdm.code.MethodUnit | method.name = ${methodName})"
```

```
:String
value="{ String methodName }
```

---

Frey, van Hoorn, Jung, Kiel, Hasselbring

MAMBA

3.05.2012 6 / 16
Measuring in SMM

Structured Metrics Meta-Model

Declaring an Observation

- Observation
Measuring in SMM

Structured Metrics Meta-Model

Declaring an Observation

- Observation
- Observed measures
Measuring in SMM

Structured Metrics Meta-Model

Declaring an Observation

- Observation
- Observed measures
- Referenced measures (selection of the measures)
Declaring an Observation

- Observation
- Observed measures
- Referenced measures (selection of the measures)

Advantages

- Addresses all aspects of measuring
Measuring in SMM

Structured Metrics Meta-Model

Declaring an Observation

- Observation
- Observed measures
- Referenced measures (selection of the measures)

Advantages

- Addresses all aspects of measuring
- Meta-model agnostic (works with MOF and Ecore-models)
Declaring an Observation

- Observation
- Observed measures
- Referenced measures (selection of the measures)

Advantages

- Addresses all aspects of measuring
- Meta-model agnostic (works with MOF and Ecore-models)

Disadvantages

- No tool support
Measuring in SMM

Structured Metrics Meta-Model

Declaring an Observation

- Observation
- Observed measures
- Referenced measures (selection of the measures)

Advantages

- Addresses all aspects of measuring
- Meta-model agnostic (works with MOF and Ecore-models)

Disadvantages

- No tool support
- Cumbersome and error prone process to develop manually
Declaring an Observation

- Observation
- Observed measures
- Referenced measures (selection of the measures)

Advantages

- Addresses all aspects of measuring
- Meta-model agnostic (works with MOF and Ecore-models)

Disadvantages

- No tool support
- Cumbersome and error prone process to develop manually
- Limited aggregate functions
Measuring in SMM

Structured Metrics Meta-Model

Declaring an Observation

- Observation
- Observed measures
- Referenced measures (selection of the measures)

Advantages

- Addresses all aspects of measuring
- Meta-model agnostic (works with MOF and Ecore-models)

Disadvantages

- No tool support
- Cumbersome and error prone process to develop manually
- Limited aggregate functions
- No support for periodic measures
A Measurement Architecture for Model-Based Analysis

- Framework
A Measurement Architecture for Model-Based Analysis

- Framework
- Eclipse-based Tooling
A Measurement Architecture for Model-Based Analysis

- Framework
- Eclipse-based Tooling
- Specification Languages
MAMBA Framework

Measurement
Controller
Execution
Engine

MDL2SMM
MQL2SMM
SMM

Measurement Providers

Kieker Measurement Provider
JDepend Measurement Provider

MDL
SMM
MQL
UML
KDM

SMM
MAMBA Framework

- MDL
- SMM
- MQL
- UML
- KDM

**Measurement Providers**

- Kieker Measurement Provider
- JDepend Measurement Provider

**Measurement Controller**

**Execution Engine**

**Input Files**

- MDL
- SMM
- MQL
- UML
- KDM
Framework

MAMBA Framework

Measurement Controller

Execution Engine

Measurement Providers

SMM

Kieker Measurement Provider

JDepend Measurement Provider

MDL2SMM

MQL2SMM

MDL

SMM

MQL

UML

KDM

...
Framework

MAMBA Framework

Measurement Controller

Execution Engine

Measurement Providers:
- Kieker Measurement Provider
- JDepend Measurement Provider

MHDL2SMM

MQL2SMM

MDL

SMM

MQL

UML

KDM

SMM
**Framework**

- **MAMBA Framework**
- **Measurement Controller**
- **Execution Engine**
- **MDL2SMM**
- **MQL2SMM**
- **SMM**
- **Kieker Measurement Provider**
- **JDepend Measurement Provider**
- **Measurement Providers**

**Tools:**
- **MDL**
- **SMM**
- **MQL**
- **UML**
- **KDM**
MAMBA Framework

- MDL
- SMM
- MQL
- UML
- KDM

MAMBA Framework:
- Measurement Controller
- MDL2SMM
- MQL2SMM
- Execution Engine
- Measurement Providers
  - Kieker Measurement Provider
  - JDepend Measurement Provider

Measurement Providers:
- SMM
- MDL
- SMM
- MQL
- SMM

Frey, van Hoorn, Jung, Kiel, Hasselbring
Measurement Provider

MAMBA

Purpose

- Transforms external data to measurement results
Measurement Provider

Purpose

- Transforms external data to measurement results
- Feed results to Measurement Controller
Measurement Provider

**Purpose**
- Transforms external data to measurement results
- Feed results to Measurement Controller

**Workflow**

```mermaid
diagram flowchart LR
bookstore.smm[bookstore.smm] --> :SmmModel[ :SmmModel ]
:SmmModel[ :SmmModel ] --> :Observation[ :Observation ]
:Observation[ :Observation ] --> observedMeasures[observedMeasures]
observedMeasures[observedMeasures] --> :CollectiveMeasure[ CollectiveMeasure ]
:CollectiveMeasure[ CollectiveMeasure ] --> name="AverageMethodResponseTime"(name="AverageMethodResponseTime")
:CollectiveMeasure[ CollectiveMeasure ] --> accumulator=average(accumulator=average)
:CollectiveMeasure[ CollectiveMeasure ] --> :BaseMeasureRelationship[ BaseMeasureRelationship ]
:BaseMeasureRelationship[ BaseMeasureRelationship ] --> from(from)
from(from) --> :Scope[ Scope ]
:Scope[ Scope ] --> class="kdm.code.ClassUnit"(class="kdm.code.ClassUnit")
:Scope[ Scope ] --> :Argument[ Argument ]
:Argument[ Argument ] --> name="methodName"(name="methodName")
:Argument[ Argument ] --> type="String"(type="String")
:Argument[ Argument ] --> value="searchBooks"(value="searchBooks")
:Argument[ Argument ] --> arguments(arguments)
arguments(arguments) --> to(to)
to(to) --> :NamedMeasure[ NamedMeasure ]
:NamedMeasure[ NamedMeasure ] --> name="Kieker2KDMProvider:KDMMethodResponseTime"(name="Kieker2KDMProvider:KDMMethodResponseTime")
:NamedMeasure[ NamedMeasure ] --> requestedMeasures[requestedMeasures]
requestedMeasures[requestedMeasures] --> provide list of named measures(provide list of named measures)


```

Frey, van Hoorn, Jung, Kiel, Hasselbring
Measurement Provider

Purpose

- Transforms external data to measurement results
- Feed results to Measurement Controller

Workflow

```
bookstore.smm :SmmModel
  :Observation
    observedMeasures
      :CollectiveMeasure
        name="AverageMethodResponseTime"
        accumulator=average
      :Scope
        class="kdm.code.ClassUnit"
      :BaseMeasureRelationship
      :NamedMeasure
        name="Kieker2KDMProvider:KDMMethodResponseTime"

requestedMeasures
  measure
    :Argument
      name="methodName"
      type="String"
      value="searchBooks"
```

KDM

MAMBA Framework

Measurement Controller

Execution Engine

Measurement Providers

Frey, van Hoorn, Jung, Kiel, Hasselbring
Measurement Provider

Purpose

- Transforms external data to measurement results
- Feed results to Measurement Controller

Workflow
Measurement Provider

**Purpose**
- Transforms external data to measurement results
- Feed results to Measurement Controller

**Workflow**

```
bookstore.smm :SmmModel
    
    :Observation
        observedMeasures
    
    :CollectiveMeasure
        name="AverageMethodResponseTime"
        accumulator=average
    
    :Scope
        class="kdm.code.ClassUnit"
    
    :BaseMeasureRelationship
        to
    
    :NamedMeasure
        name="Kieker2KDMProvider:KDMMethodResponseTime"

requestedMeasures

:Argument
    name="methodName"
    type="String"
    value="searchBooks"

MAMBA Framework

Measurement Controller

Execution Engine

Measurement Providers

Kieker

JDepend

SMM

add measurement results to SMM
```
library example

model kdm "http://schema.omg.org/spec/KDM/1.2"

collect NumberOfClasses sum ClassRecognizer
   scope kdm.code.CodeModel

count ClassRecognizer
   select isOCLTypeof(kdm.code.ClassUnit)
   scope kdm.code.Package

collect AverageMethodResponseTime ( String methodName )
   average KDMMethodResponseTime
   scope ( kdm.code.ClassUnit.codeElement method |
          method.name == methodName && method instanceof kdm.code.MethodUnit)

measure KDMMethodResponseTime
library bookstore

bookstore.smm : MeasureLibrary
```mambasml
library bookstore

org/spec/KDM/1.2"
```
library bookstore

model kdm "http://schema.omg.org/spec/KDM/1.2"

collect NumberOfClasses
  sum ...

bookstore.smm :MeasureLibrary
  :CollectiveMeasure
  name="NumberOfClasses"
  accumulator=sum
library bookstore

model kdm "http://schema.omg.org/spec/KDM/1.2"

collect NumberOfClasses
  sum ...
  scope kdm.code.CodeModel
library bookstore

model kdm "http://schema.omg.org/spec/KDM/1.2"

collect NumberOfClasses
  sum ClassRecognizer
  scope kdm.code.CodeModel
library bookstore

model kdm "http://schema.omg.org/spec/KDM/1.2"

collect NumberOfClasses
  sum ClassRecognizer
  scope kdm.code.CodeModel

count ClassRecognizer
library bookstore

model kdm "http://schema.omg.org/spec/KDM/1.2"

collect NumberOfClasses
  sum ClassRecognizer
  scope kdm.code.CodeModel

count ClassRecognizer
  ...
  scope kdm.code.Package
library bookstore

model kdm "http://schema.omg.org/spec/KDM/1.2"

collect NumberOfClasses
  sum ClassRecognizer
  scope kdm.code.CodeModel

count ClassRecognizer
  select isOCLTypeOf(kdm.code.ClassUnit)
  scope kdm.code.Package
collect AverageMethodResponseTime ( String methodName )
    average KDMMethodResponseTime
    scope ( kdm.code.ClassUnit.codeElement method |
        method.name == methodName && method instanceof kdm.code.MethodUnit)

measure KDMMethodResponseTime
MAMBA Query Language

Defines measurement process

- Definition of the observations

Example query over the collected measurements:

```sql
use bookstore
model appModel "bookstore-model.kdm"
select AverageMethodResponseTime("searchBook") as avgrt
from appModel where avgrt > 500
```
Defines measurement process

- Definition of the observations
- If necessary, extension of the measure library
  Example can be found in [FvHJ⁺11]
MAMBA Query Language

Defines measurement process

- Definition of the observations
- If necessary, extension of the measure library
  Example can be found in [FvHJ+11]
- Query over the collected measurements

```sql
use bookstore
model appModel "bookstore-model.kdm"
select AverageMethodResponseTime("searchBook") as avgrt
from appModel where avgrt > 500
```
MAMBA Query Language

Defines measurement process

- Definition of the observations
- If necessary, extension of the measure library
  Example can be found in [FvHJ+11]
- Query over the collected measurements

Average response time

use bookstore
Defines measurement process
- Definition of the observations
- If necessary, extension of the measure library
  Example can be found in [FvHJ+11]
- Query over the collected measurements

Average response time

use bookstore

model appModel "bookstore-model.kdm"
MAMBA Query Language

Defines measurement process

- Definition of the observations
- If necessary, extension of the measure library
  Example can be found in [FvHJ+11]
- Query over the collected measurements

Average response time

use bookstore

model appModel "bookstore-model.kdm"

select AverageMethodResponseTime("searchBook") as avgrt
   from appModel where avgrt > 500
Features

- Computation engine for SMM and its MAMBA extensions
Conclusions

Features

- Computation engine for SMM and its MAMBA extensions
- Languages to define measures and queries
Conclusions

Features

- Computation engine for SMM and its MAMBA extensions
- Languages to define measures and queries
- Meta-Model independent
Conclusions

Features

- Computation engine for SMM and its MAMBA extensions
- Languages to define measures and queries
- Meta-Model independent
- Incorporation of static and dynamic analysis
Conclusions

Features

- Computation engine for SMM and its MAMBA extensions
- Languages to define measures and queries
- Meta-Model independent
- Incorporation of static and dynamic analysis

Case Studies [FvHJ+ 11]
Conclusions

Features

- Computation engine for SMM and its MAMBA extensions
- Languages to define measures and queries
- Meta-Model independent
- Incorporation of static and dynamic analysis

Case Studies [FvHJ⁺11]

- DynaMod: Re-engineering project [vHFG⁺11]
Conclusions

Features
- Computation engine for SMM and its MAMBA extensions
- Languages to define measures and queries
- Meta-Model independent
- Incorporation of static and dynamic analysis

Case Studies [FvHJ+11]
- DynaMod: Re-engineering project [vHFG+11]
- MENGES: DSL for embedded systems and tooling [GHH+12]
Conclusions

Features

- Computation engine for SMM and its MAMBA extensions
- Languages to define measures and queries
- Meta-Model independent
- Incorporation of static and dynamic analysis

Case Studies [FvHJ⁺11]

- DynaMod: Re-engineering project [vHFG⁺11]
- MENGES: DSL for embedded systems and tooling [GHH⁺12]
- CouldMIG: Framework for cloud migration [FH10, FHS12]


