Preprint: van Hoorn, A., Knoche, H., Goerigk, W. and Hasselbring, W. (2011) Model-Driven Instrumentation for Dynamic Analysis of Legacy Software Systems [Paper] In: 13th Workshop Software-Reengineering(WSR 2011), May 2-4, 2011, Bad Honnef, Germany.

Model-Driven Instrumentation for Dynamic Analysis of Legacy Software Systems

André van Hoorn[1], Holger Knoche[2], Wolfgang Goerigk[2], and Wilhelm Hasselbring[1]

[1]Software Engineering Group, University of Kiel, Germany

[2]b+m Informatik AG, Melsdorf, Germany

May 03, 2011 @ WSR 2011, Bad Honnef

C | A | U
Christian-Albrechts-Universität zu Kiel
Technische Fakultät

b+m
business IT management

DynaMod

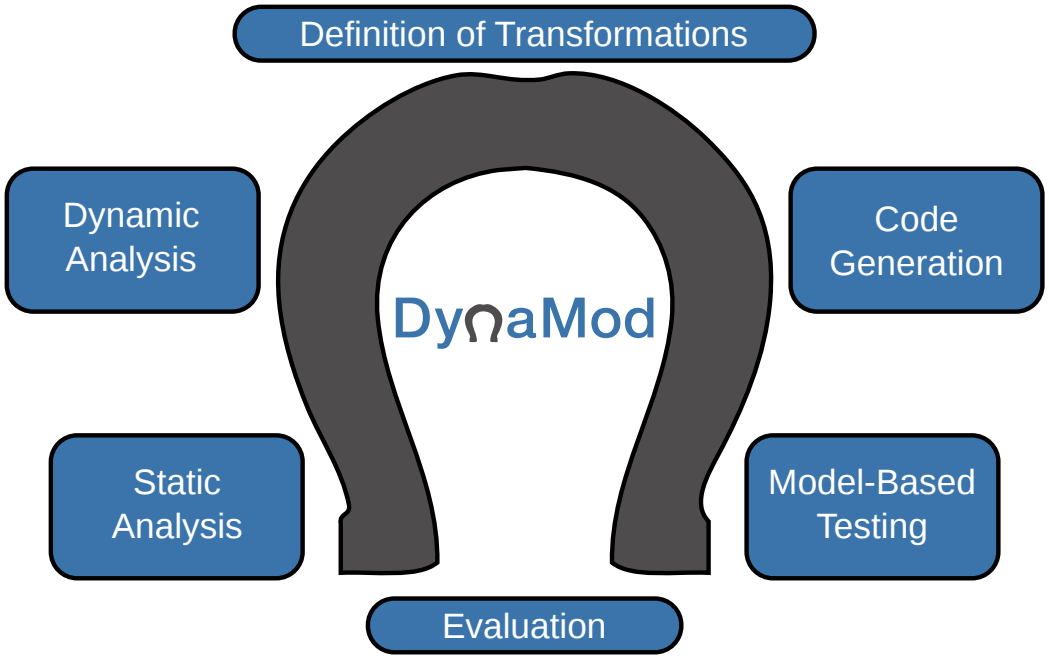
GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Christian-Albrechts-Universität zu Kiel

DynaMod

Dynamic Analysis for
Model-Driven Software Modernization



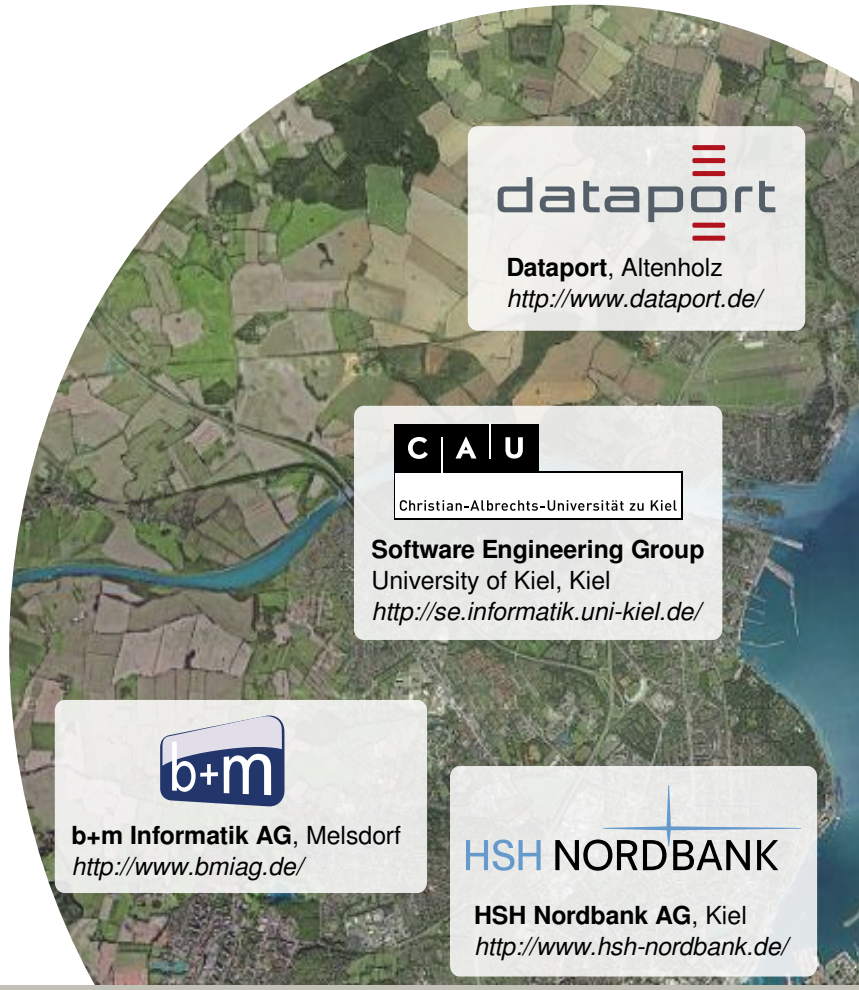
Christian-Albrechts-Universität zu Kiel

DynaMod

Dynamic Analysis for
Model-Driven Software Modernization

Project Consortium:

1. **b+m Informatik AG**
   (Development partner, consortium leader)
   - Comprehensive MDSD know-how
   - Initiated openArchitectureWare (oAW)

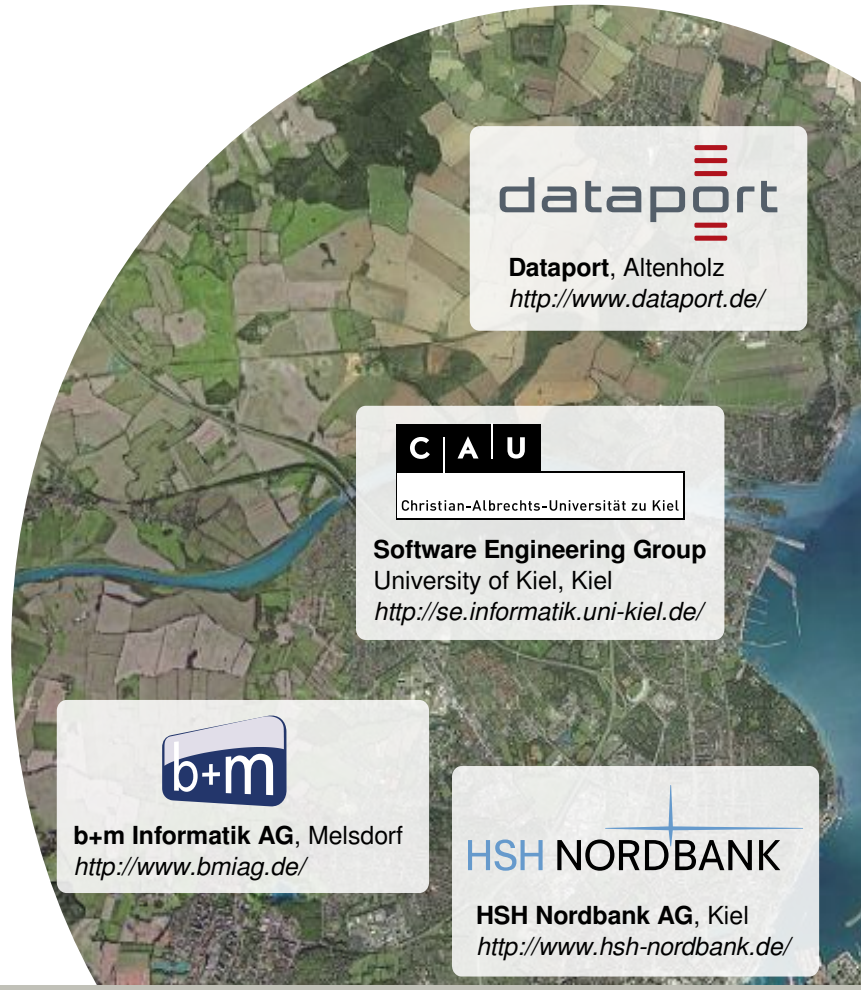


DynaMod

Dynamic Analysis for
Model-Driven Software Modernization

Project Consortium:

1. **b+m Informatik AG**
   (Development partner, consortium leader)
   - Comprehensive MDSD know-how
   - Initiated openArchitectureWare (oAW)
2. **Software Engineering Group, Univ. Kiel**
   (Research partner)
   - Model-driven engineering, operation, and evolution of software systems
   - Emphasis on software quality (of service)



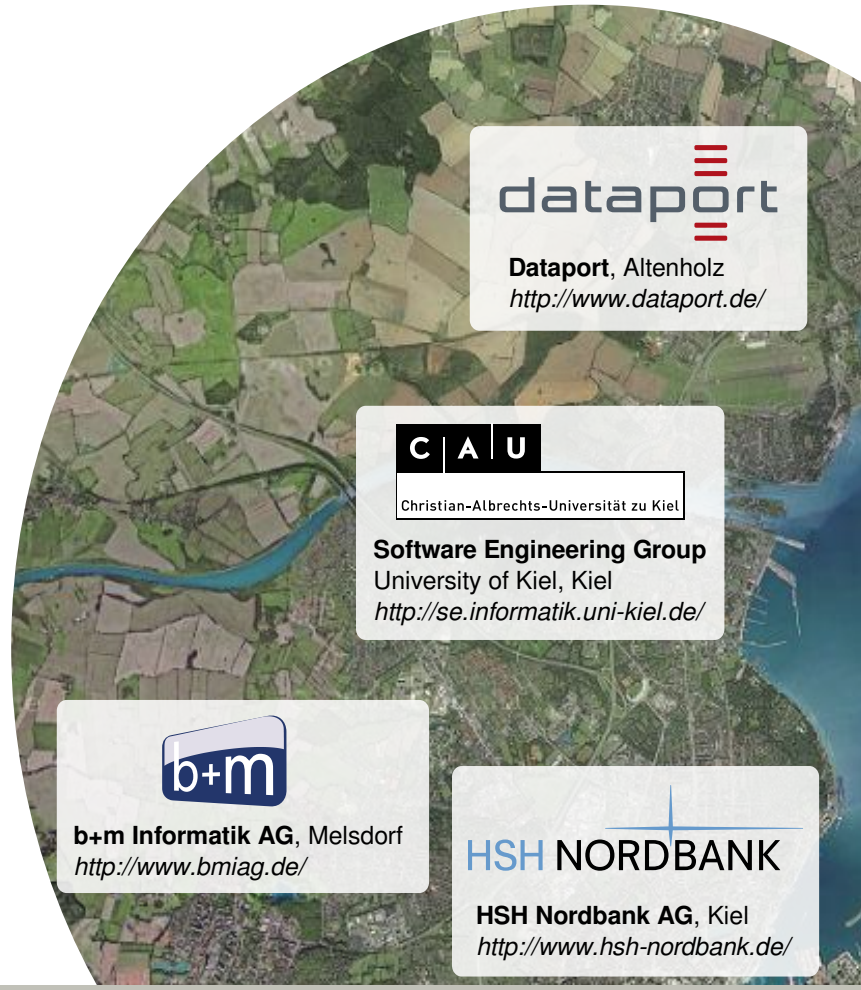
Christian-Albrechts-Universität zu Kiel

DynaMod

Dynamic Analysis for Model-Driven Software Modernization

Project Consortium:

1. **b+m Informatik AG**
   (Development partner, consortium leader)
   - Comprehensive MDSD know-how
   - Initiated openArchitectureWare (oAW)
2. **Software Engineering Group, Univ. Kiel**
   (Research partner)
   - Model-driven engineering, operation, and evolution of software systems
   - Emphasis on software quality (of service)
3. **Dataport**
   (Associated partner)
   - Provides ICT services for public/tax administrations

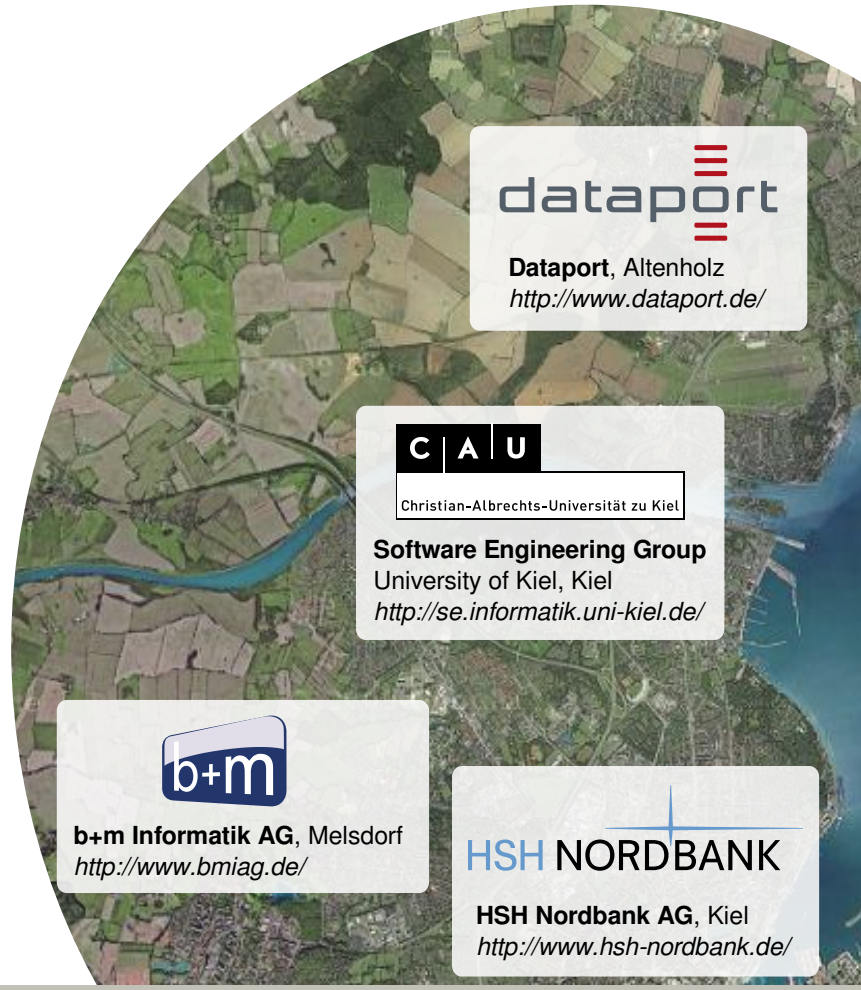


DynaMod

Dynamic Analysis for Model-Driven Software Modernization

Project Consortium:

1. **b+m Informatik AG**
   (Development partner, consortium leader)
   - Comprehensive MDSD know-how
   - Initiated openArchitectureWare (oAW)
2. **Software Engineering Group, Univ. Kiel**
   (Research partner)
   - Model-driven engineering, operation, and evolution of software systems
   - Emphasis on software quality (of service)
3. **Dataport**
   (Associated partner)
   - Provides ICT services for public/tax administrations
4. **HSH Nordbank AG**
   (Associated partner)
   - Leading bank for corporate and private clients in northern Germany



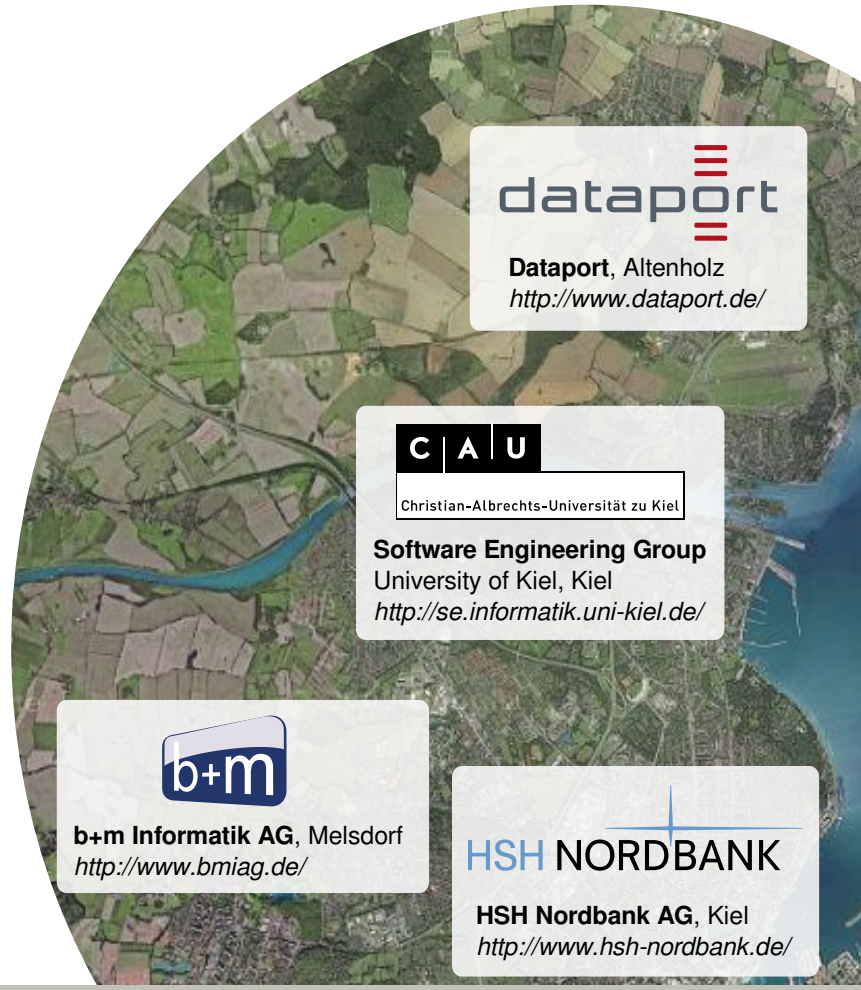
Christian-Albrechts-Universität zu Kiel

DynaMod

Dynamic Analysis for Model-Driven Software Modernization

Project Consortium:

1. **b+m Informatik AG**
   (Development partner, consortium leader)
2. **Software Engineering Group, Univ. Kiel**
   (Research partner)
3. **Dataport**
   (Associated partner)
4. **HSH Nordbank AG**
   (Associated partner)



Funding:

- BMBF “KMU-innovativ”
- 2 years (01/11–12/12)

SPONSORED BY THE
Federal Ministry of Education and Research

Under grant no. 01IS10051

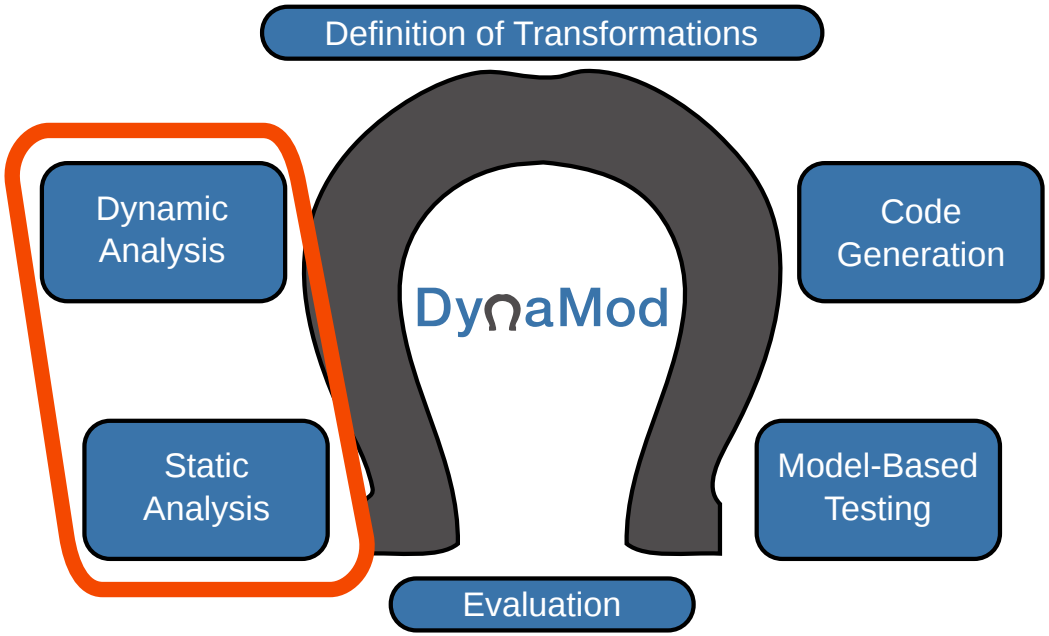
Read More:

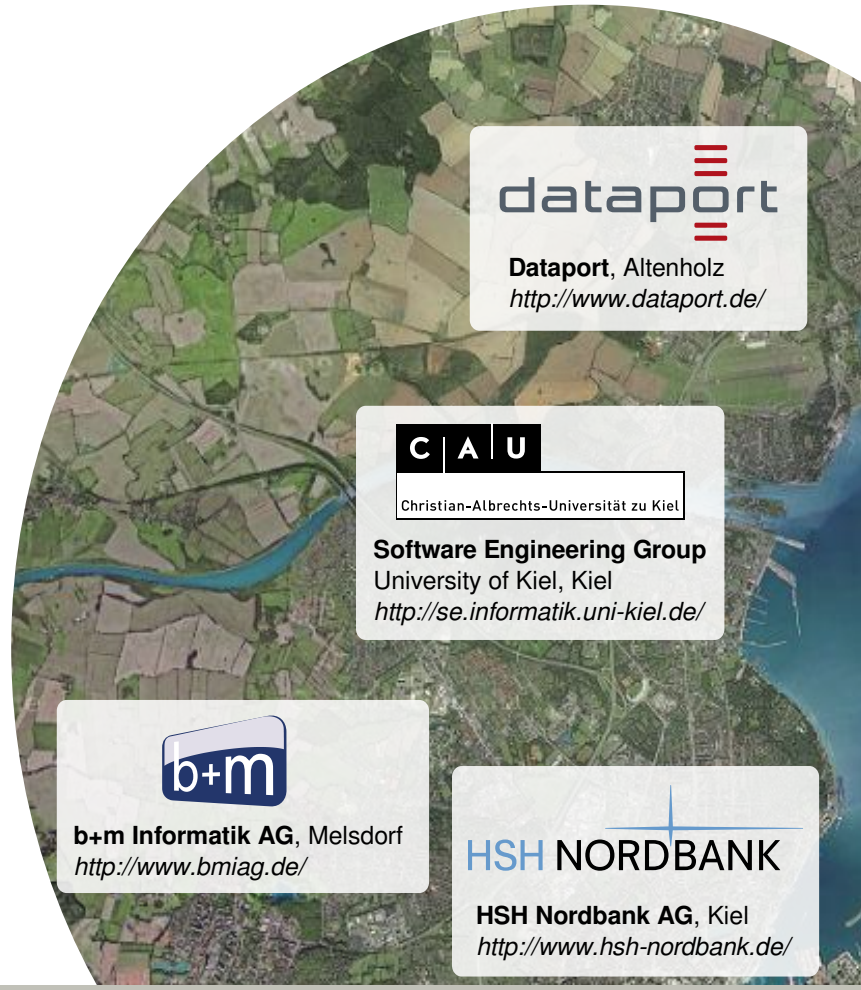
- `http://kosse-sh.de/dynamod/`
- MDSM @ CSMR 2011 [vHFG+11]

Christian-Albrechts-Universität zu Kiel

DynaMod

Dynamic Analysis for Model-Driven Software Modernization





Motivation

Static analysis is not sufficient to study a system's architecture comprehensively

Motivation

Static analysis is not sufficient to study a system's architecture comprehensively

How to Gather Runtime Data from Executing Systems? — Instrumentation

- Profiling — employed in development environments; considerable performance overhead
- Monitoring — employed in production environments; captures real usage profile

Motivation

Static analysis is not sufficient to study a system's architecture comprehensively

How to Gather Runtime Data from Executing Systems? — Instrumentation

- Profiling — employed in development environments; considerable performance overhead
- Monitoring — employed in production environments; captures real usage profile

Use Cases — Online & Offline Analysis

The obtained monitoring data can, for instance, be used for

- Performance evaluation (e.g., bottleneck detection)
- (Self-)adaptation control (e.g., capacity management)
- Application-level failure detection and diagnosis
- Service-level management

Motivation

Static analysis is not sufficient to study a system's architecture comprehensively

How to Gather Runtime Data from Executing Systems? — Instrumentation

- Profiling — employed in development environments; considerable performance overhead
- Monitoring — employed in production environments; captures real usage profile

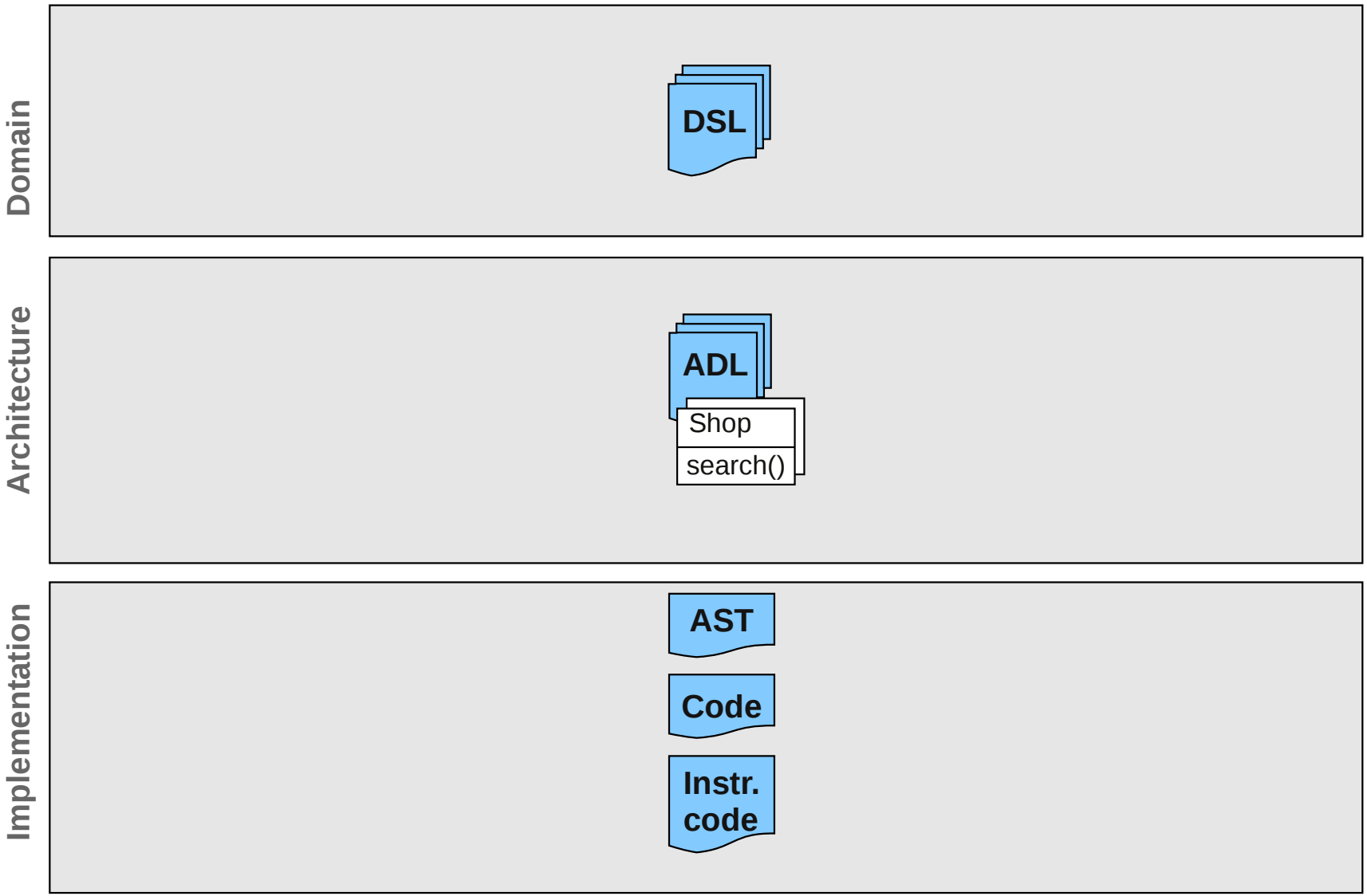
Use Cases — Online & Offline Analysis

The obtained monitoring data can, for instance, be used for

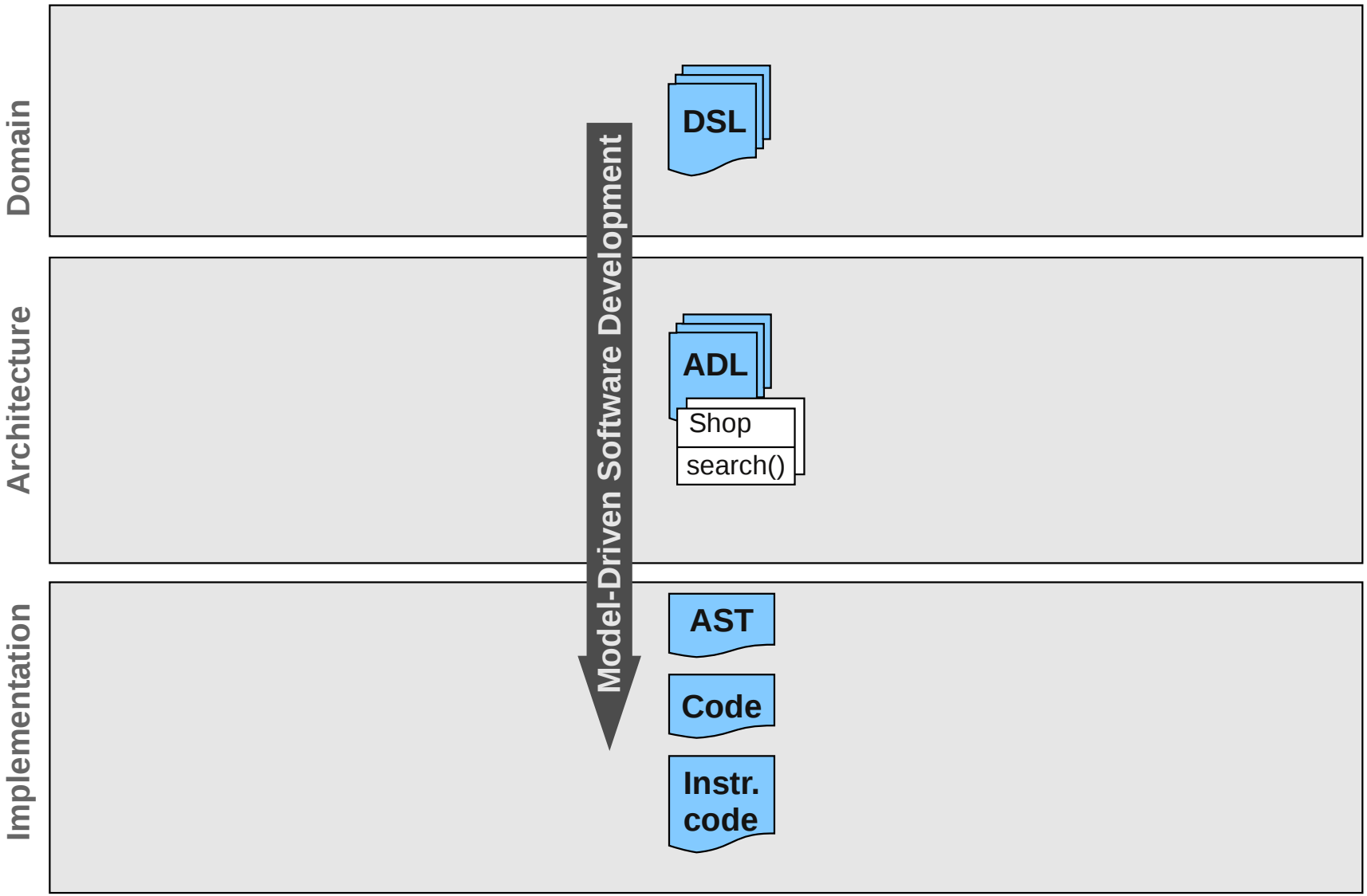
- Performance evaluation (e.g., bottleneck detection)
- (Self-)adaptation control (e.g., capacity management)
- Application-level failure detection and diagnosis
- Service-level management
- **Software reengineering:** (architecture) reconstruction, modernization

Agenda

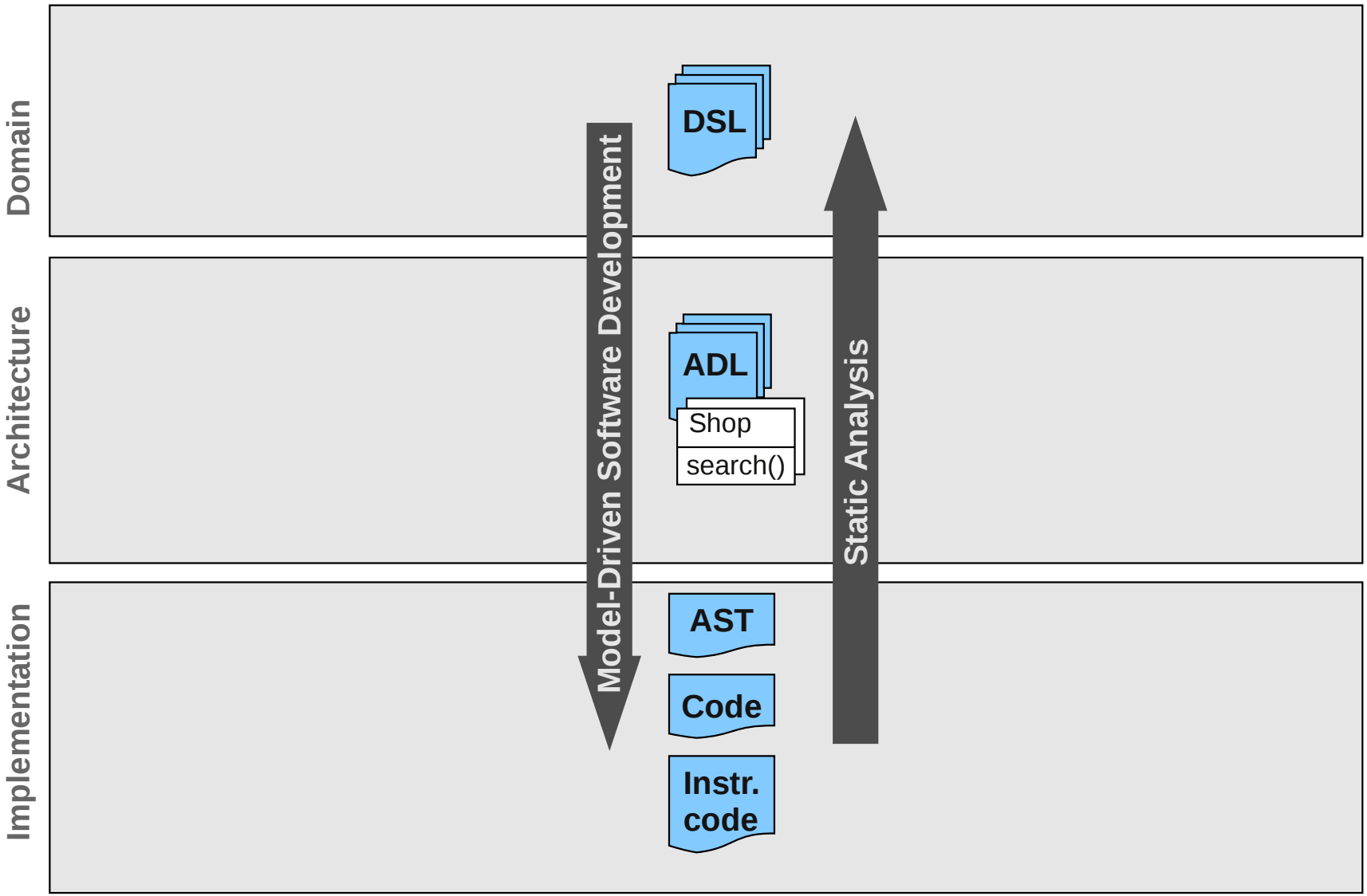
Overview


Domain
DSL
Architecture
ADL
Shop
search()
Implementation
AST
Code
Instr.
code

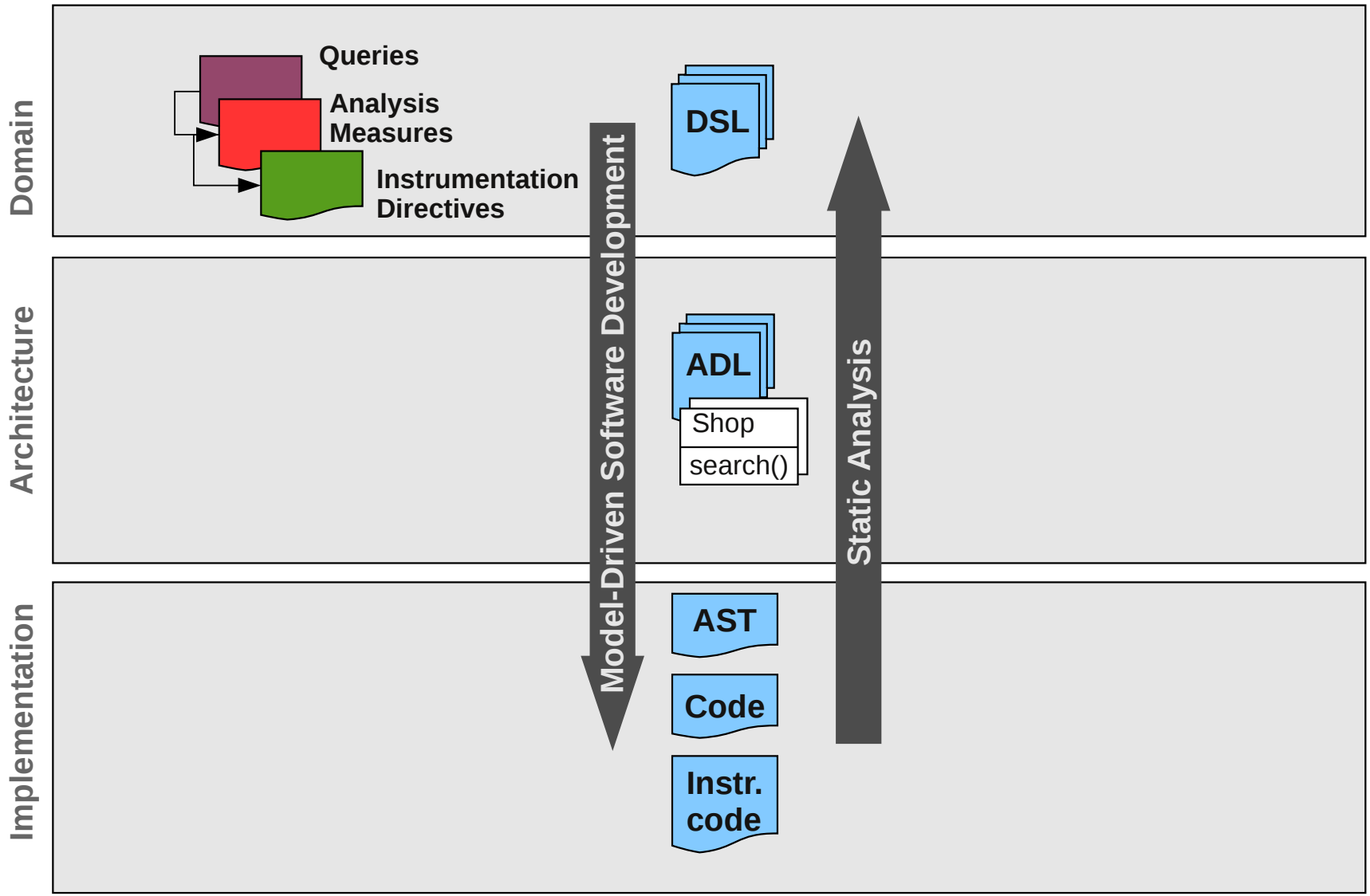
Overview


Domain
Architecture
Implementation
Model-Driven Software Development
DSL
ADL
Shop
search()
AST
Code
Instr. code

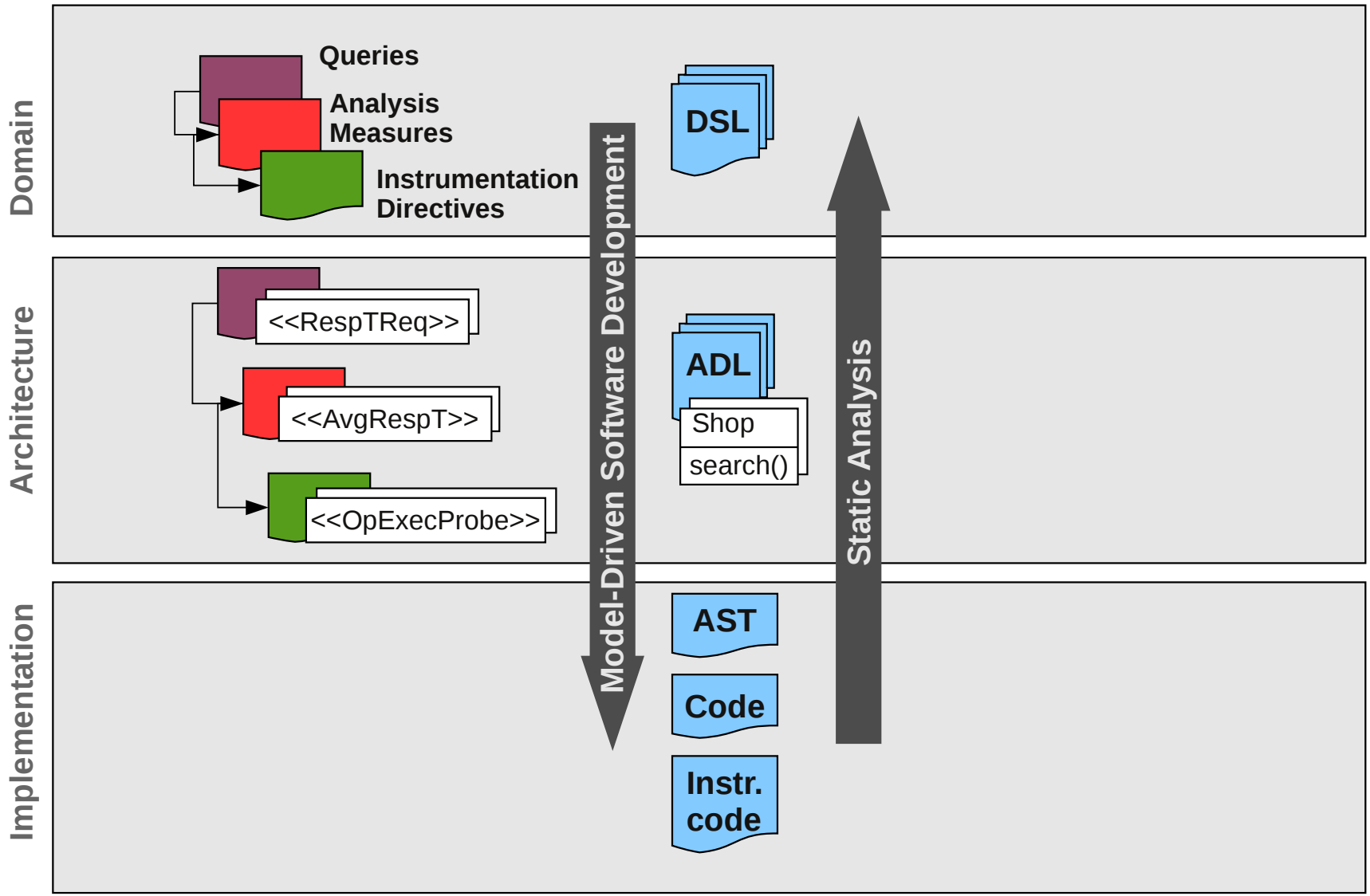
Overview


Domain
Architecture
Implementation
DSL
ADL
Shop
search()
AST
Code
Instr. code
Model-Driven Software Development
Static Analysis

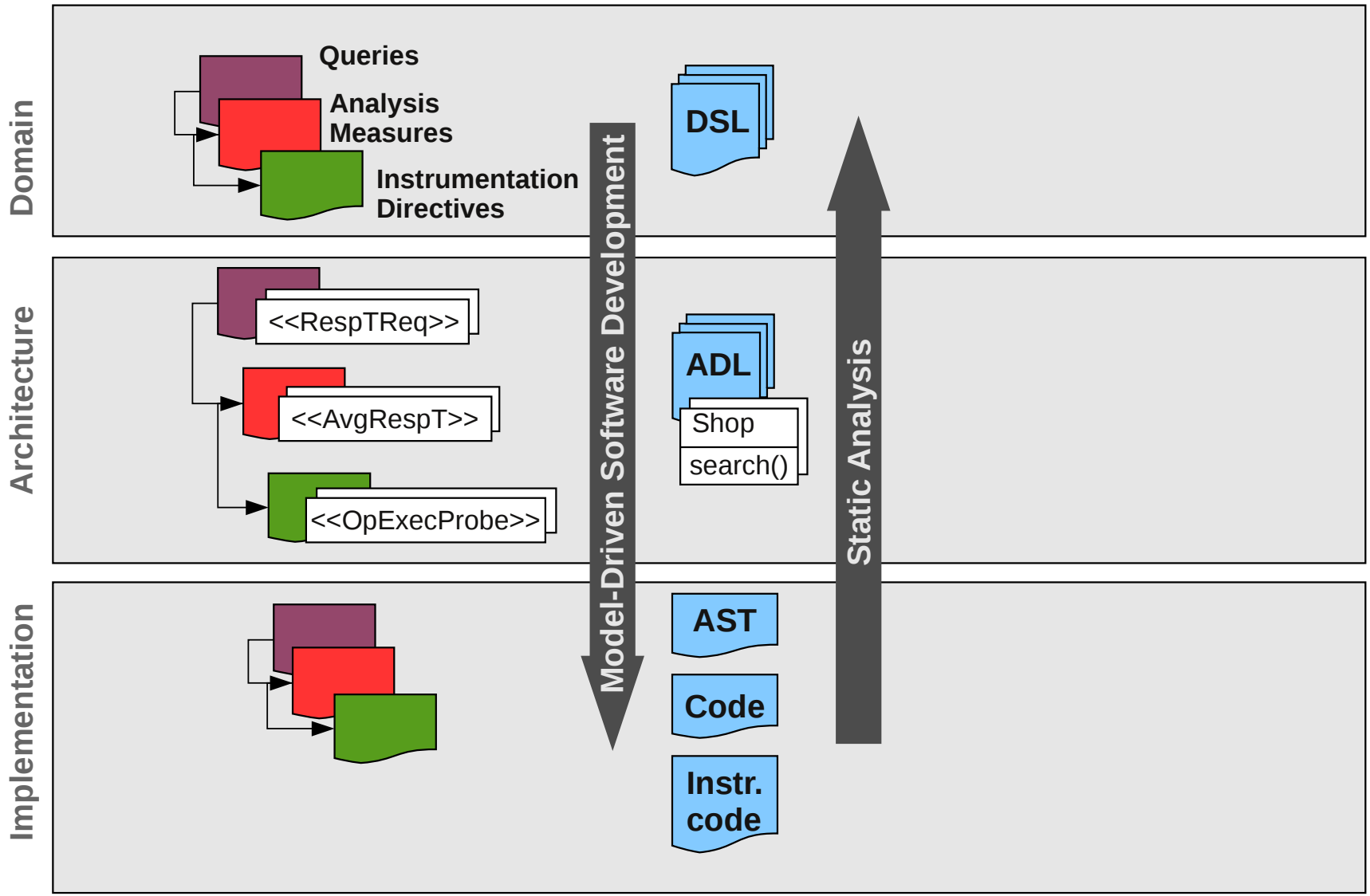
Overview


Domain
Architecture
Implementation
Queries
Analysis
Measures
Instrumentation
Directives
DSL
ADL
Shop
search()
AST
Code
Instr.
code
Model-Driven Software Development
Static Analysis

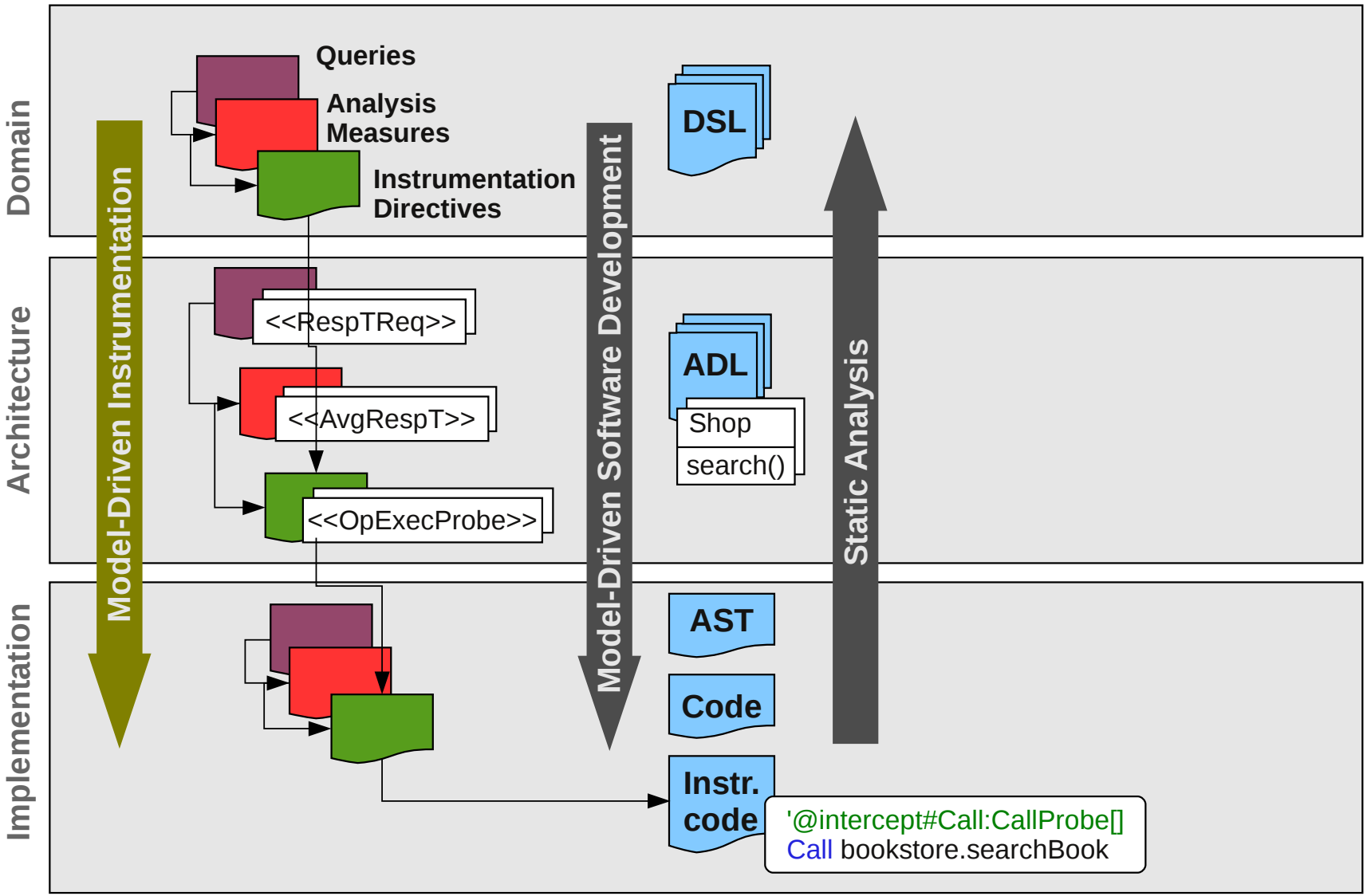
Overview

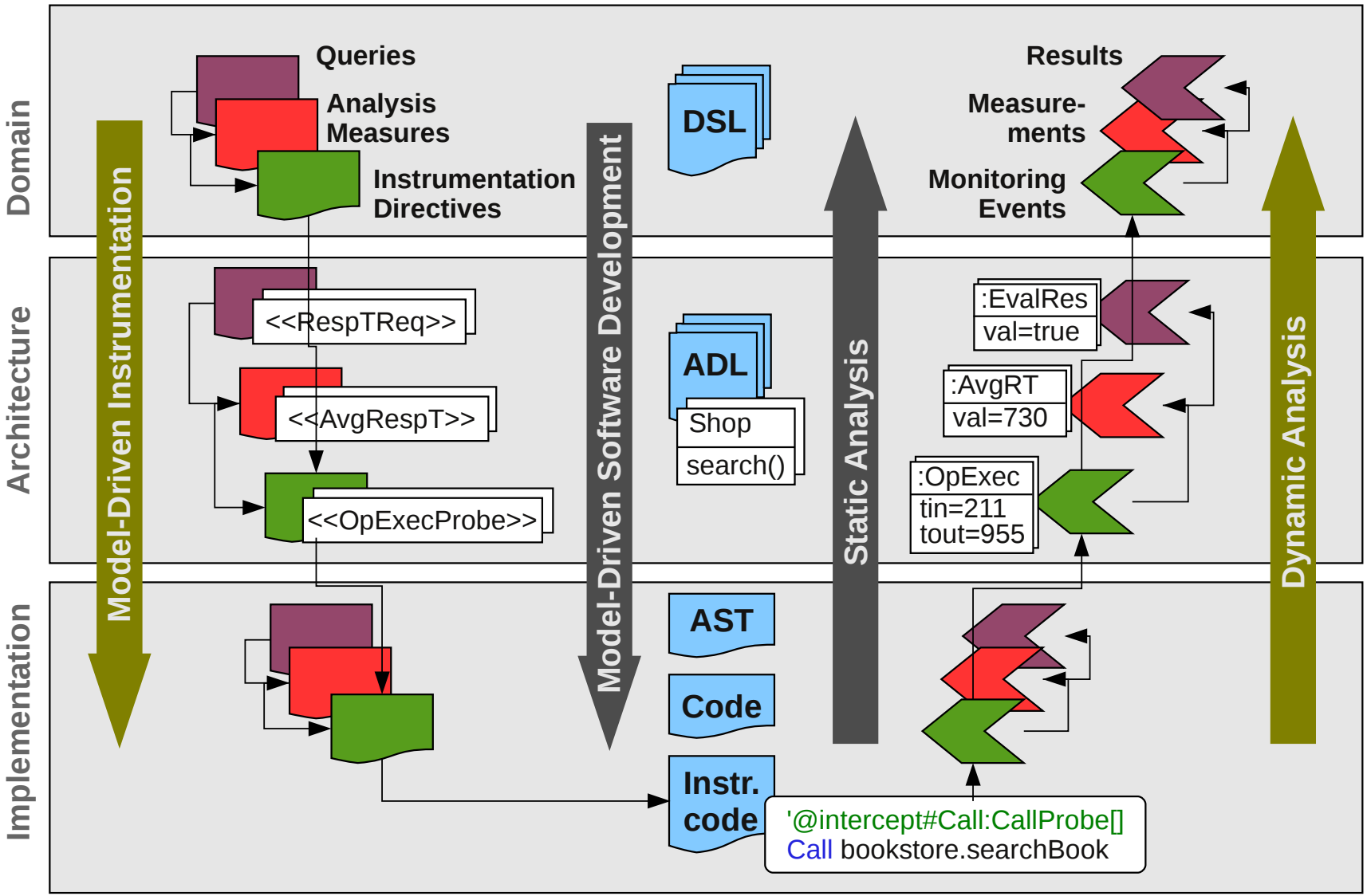

Domain
Queries
Analysis Measures
Instrumentation Directives
DSL
Architecture
<<RespTReq>>
<<AvgRespT>>
<<OpExecProbe>>
ADL
Shop
search()
Implementation
AST
Code
Instr. code
Model-Driven Software Development
Static Analysis

Overview

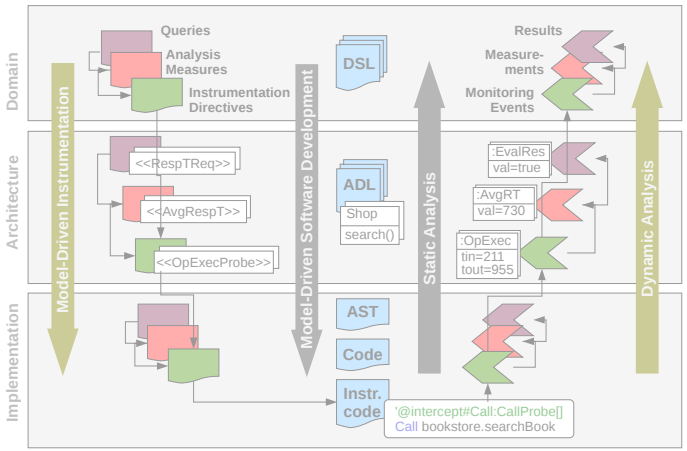

Domain
Queries
Analysis
Measures
Instrumentation
Directives
DSL
Architecture
<<RespTReq>>
<<AvgRespT>>
<<OpExecProbe>>
ADL
Shop
search()
Implementation
AST
Code
Instr.
code
Model-Driven Software Development
Static Analysis

Overview

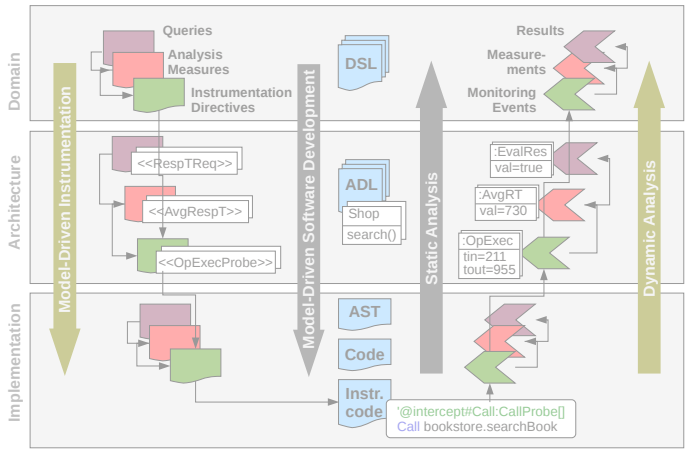

Domain
Architecture
Implementation
Queries
Analysis Measures
Instrumentation Directives
DSL
Model-Driven Instrumentation
Model-Driven Software Development
Static Analysis
<<RespTReq>>
<<AvgRespT>>
<<OpExecProbe>>
ADL
Shop
search()
AST
Code
Instr. code
'@intercept#Call:CallProbe[]
Call bookstore.searchBook


Domain
Architecture
Implementation
Queries
Analysis Measures
Instrumentation Directives
DSL
Results
Measure-ments
Monitoring Events
Model-Driven Instrumentation
Model-Driven Software Development
Static Analysis
Dynamic Analysis
<<RespTReq>>
<<AvgRespT>>
<<OpExecProbe>>
ADL
Shop
search()
:EvalRes
val=true
:AvgRT
val=730
:OpExec
tin=211
tout=955
AST
Code
Instr. code
'@intercept#Call:CallProbe[]
Call bookstore.searchBook

Overview—DynaMod Examples

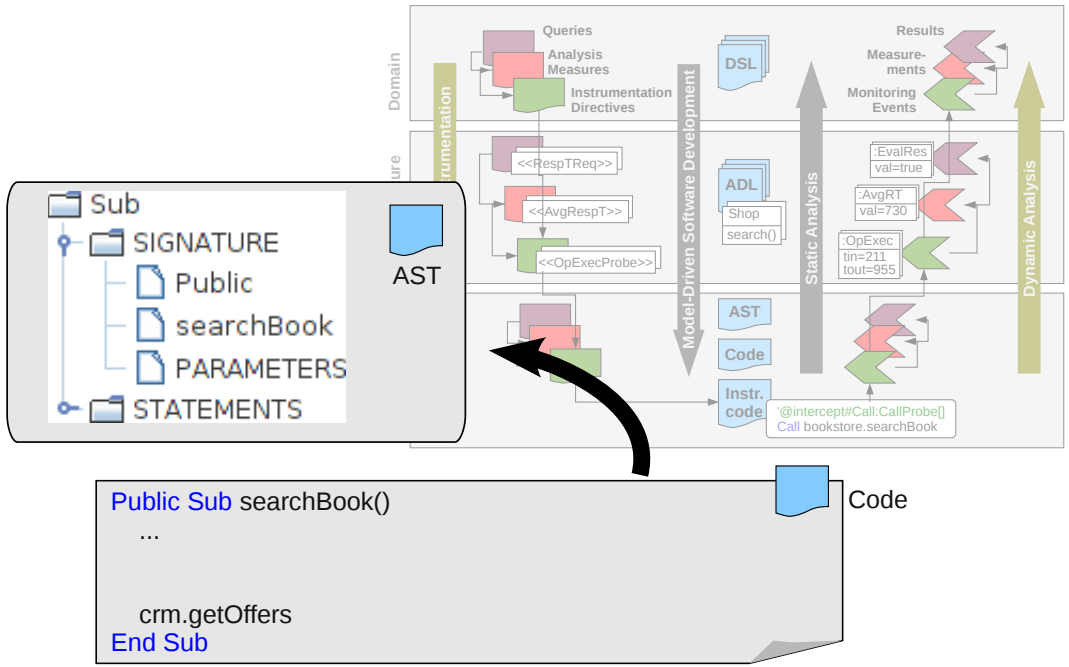


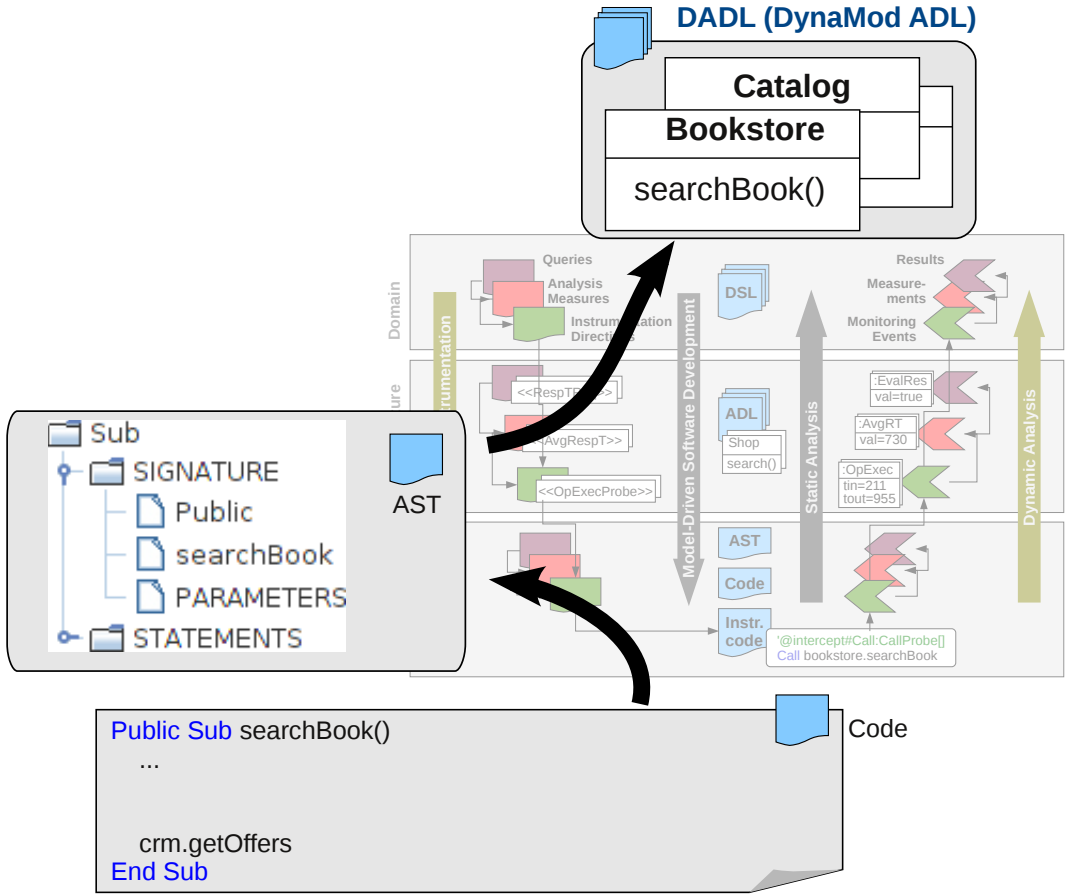
Overview—DynaMod Examples

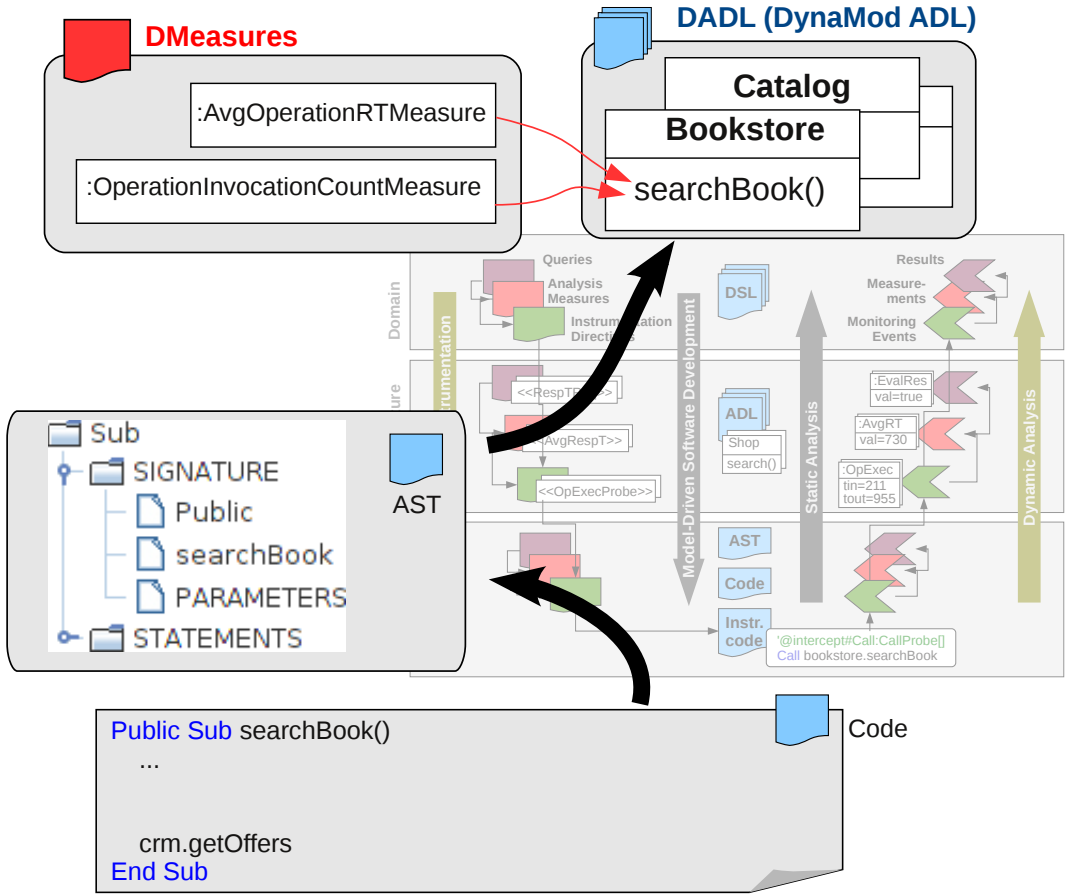


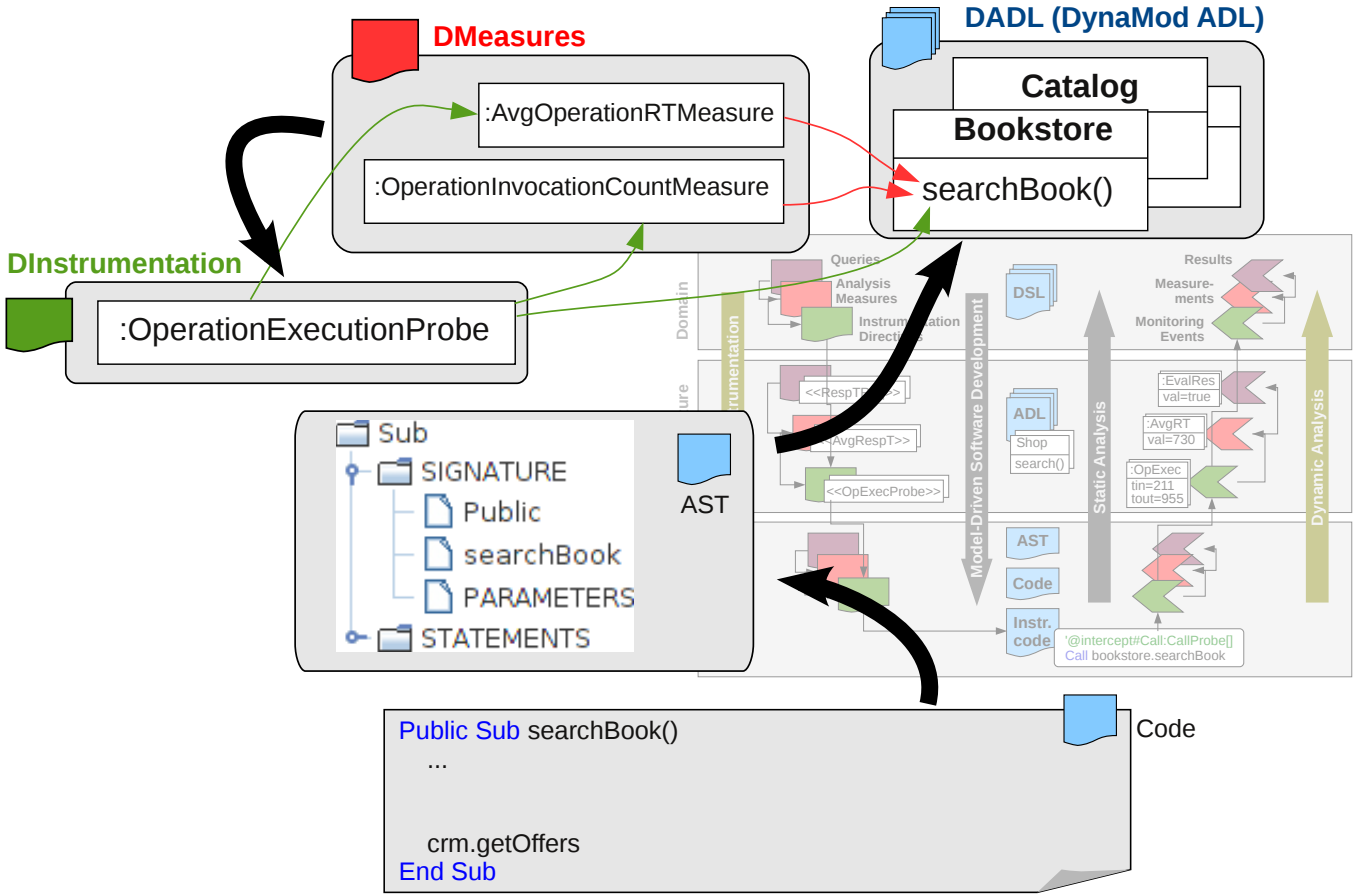

```
Public Sub searchBook()
   ...


   crm.getOffers
End Sub
```

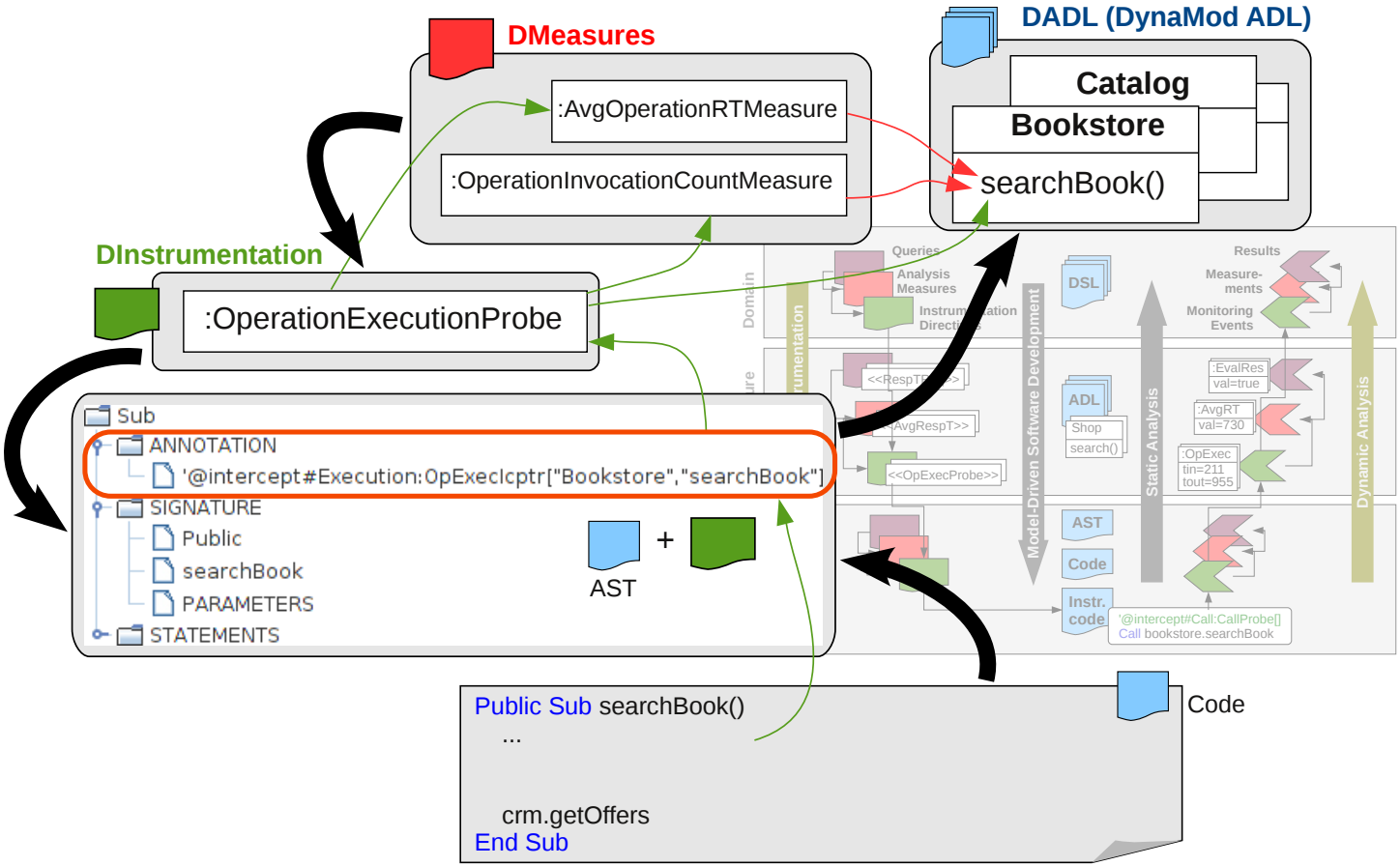
Code

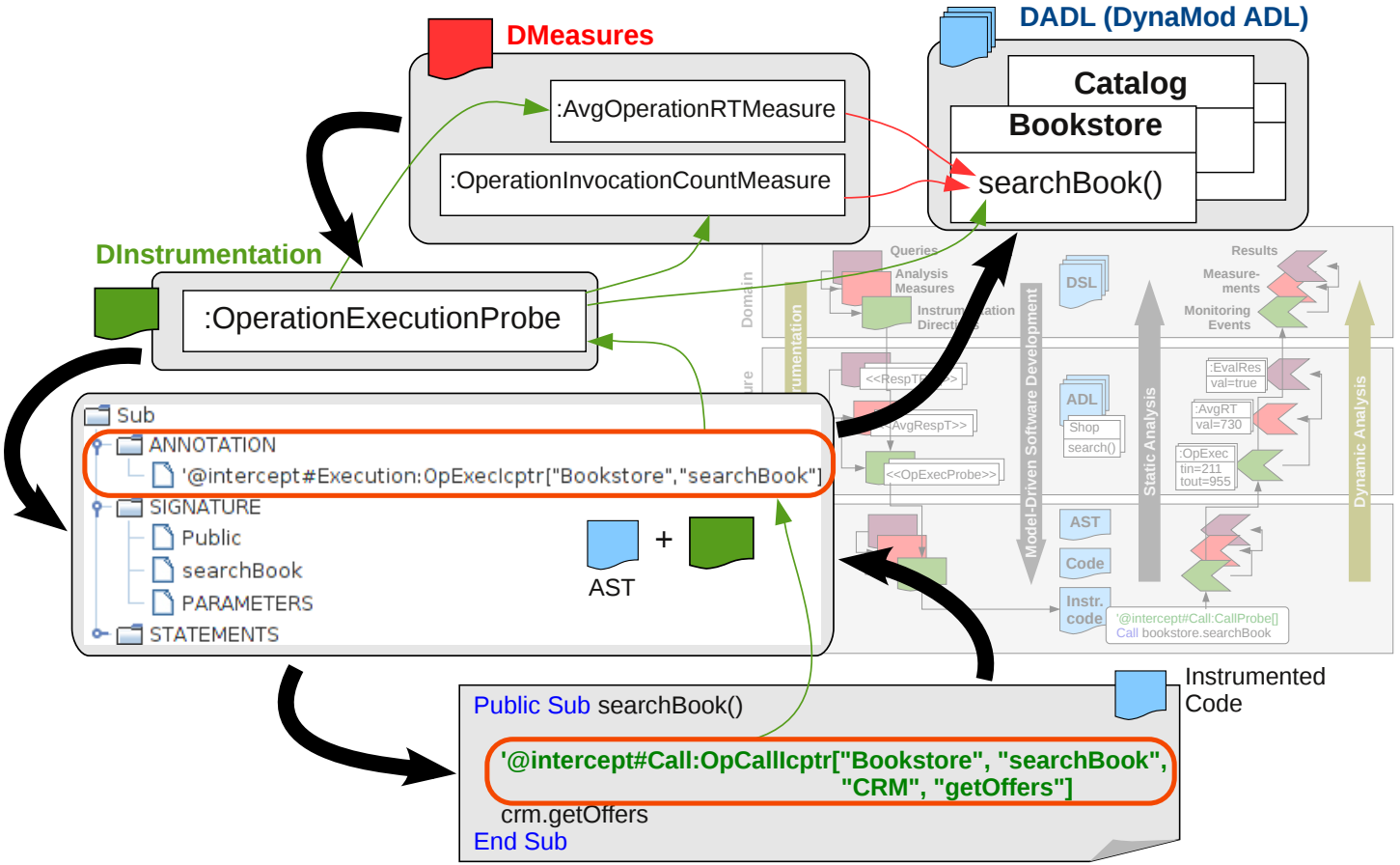

Domain
Queries
Analysis Measures
Instrumentation Directives
DSL
Results
Measure-ments
Monitoring Events
<<RespTReq>>
<<AvgRespT>>
<<OpExecProbe>>
ADL
Shop
search()
:EvalRes
val=true
:AvgRT
val=730
:OpExec
tin=211
tout=955
Model-Driven Software Development
Static Analysis
Dynamic Analysis
AST
Code
Instr. code
'@intercept#Call:CallProbe[]
Call bookstore.searchBook
Sub
SIGNATURE
Public
searchBook
PARAMETERS
STATEMENTS
AST
Public Sub searchBook()
...
crm.getOffers
End Sub
Code

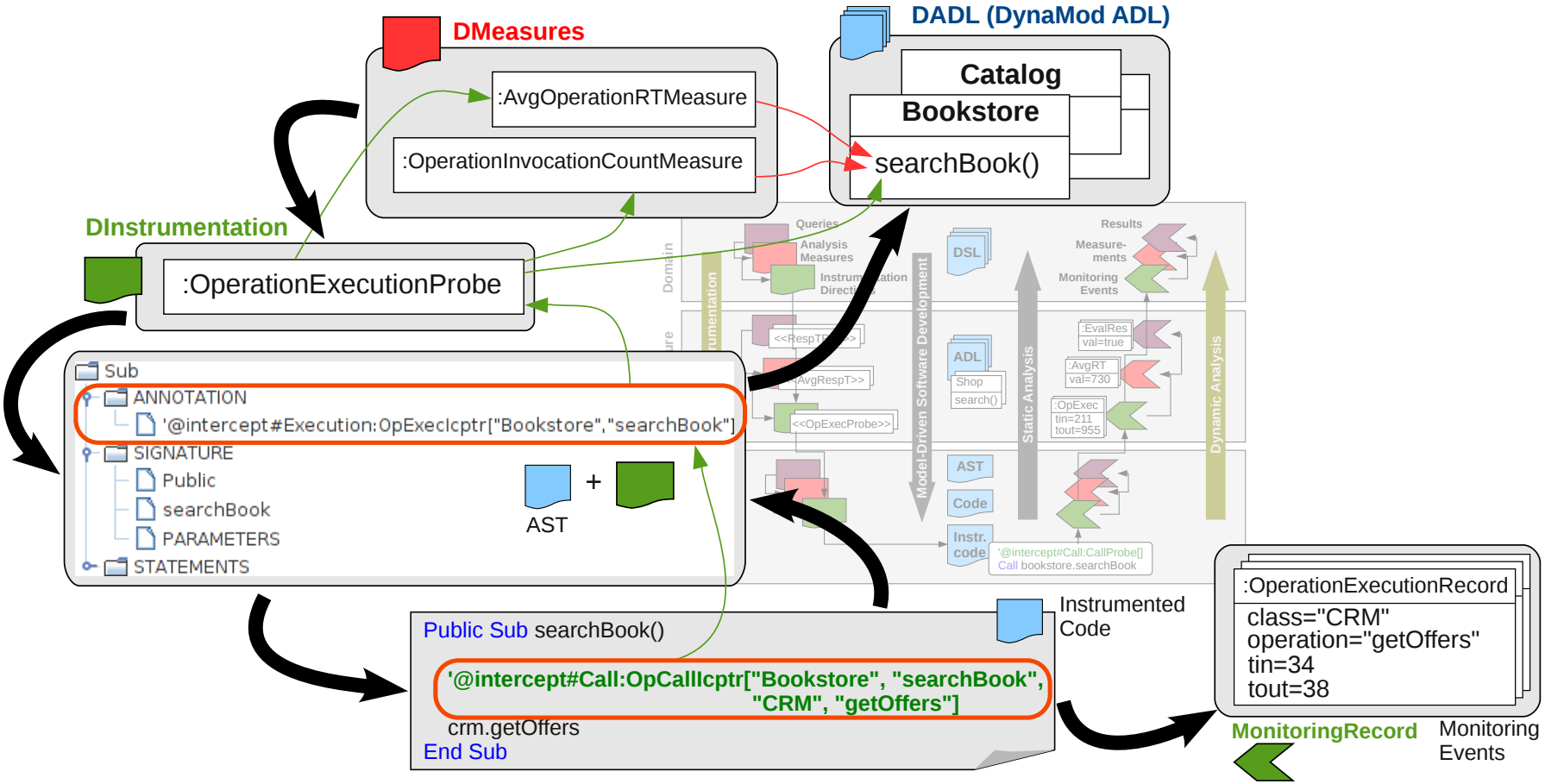

DADL (DynaMod ADL)
Catalog
Bookstore
searchBook()
Queries
Analysis Measures
Instrumentation Directives
DSL
Results
Measure-ments
Monitoring Events
Domain
Model-Driven Software Development
Static Analysis
Dynamic Analysis
ADL
Shop
search()
:EvalRes val=true
:AvgRT val=730
:OpExec tin=211 tout=955
<<OpExecProbe>>
AST
Code
Instr. code
'@intercept#Call:CallProbe[]
Call bookstore.searchBook
Sub
SIGNATURE
Public
searchBook
PARAMETERS
STATEMENTS
AST
Public Sub searchBook()
...
crm.getOffers
End Sub
Code

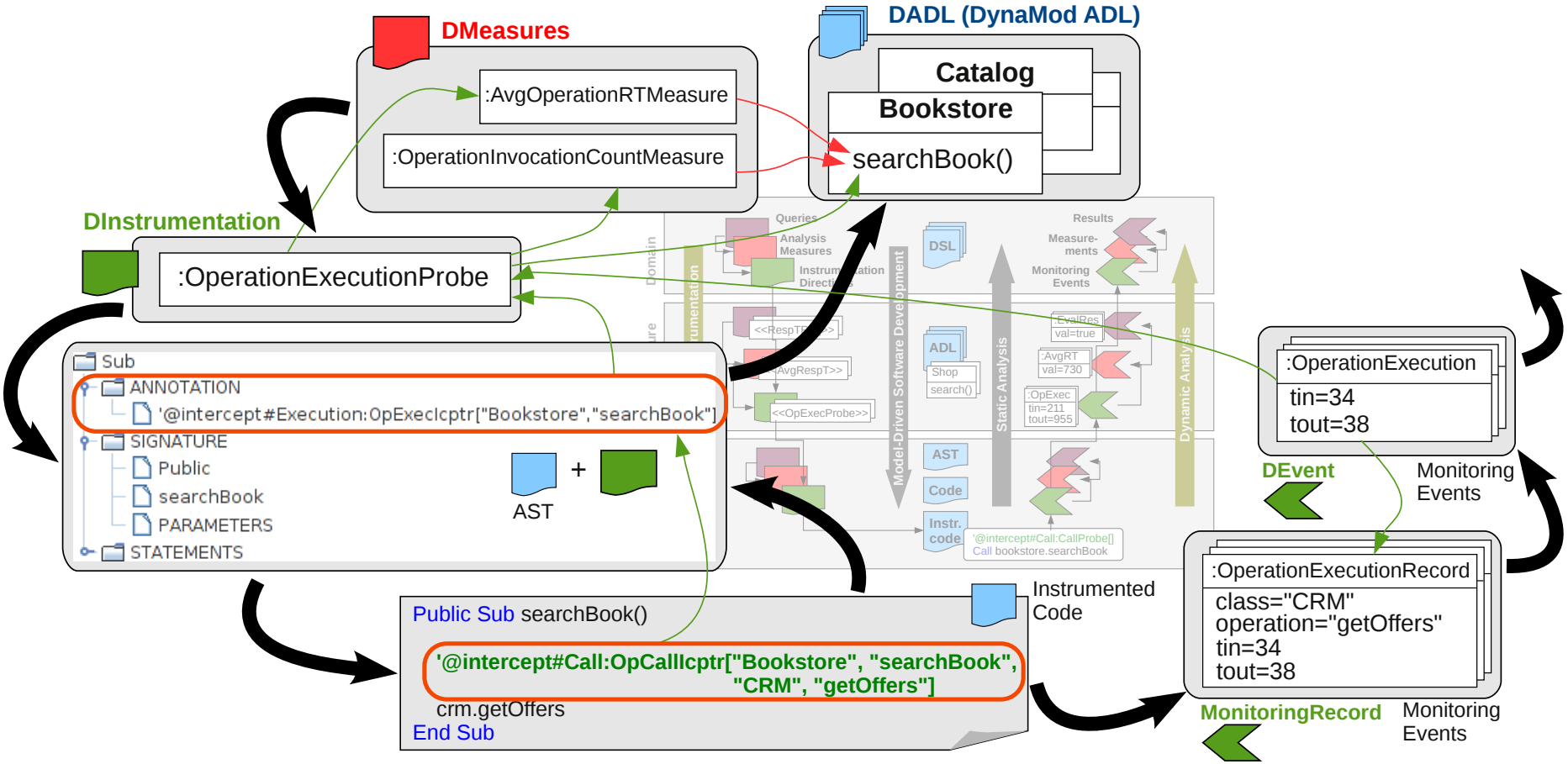

DMeasures
:AvgOperationRTMeasure
:OperationInvocationCountMeasure
DADL (DynaMod ADL)
Catalog
Bookstore
searchBook()
Queries
Analysis Measures
Instrumentation Directives
DSL
Results
Measure-ments
Monitoring Events
Domain
Model-Driven Software Development
Static Analysis
Dynamic Analysis
ADL
Shop
search()
:EvalRes val=true
:AvgRT val=730
:OpExec tin=211 tout=955
<<AvgRespT>>
<<OpExecProbe>>
AST
Code
Instr. code
'@intercept#Call:CallProbe[]
Call bookstore.searchBook
Sub
SIGNATURE
Public
searchBook
PARAMETERS
STATEMENTS
AST
Public Sub searchBook()
...
crm.getOffers
End Sub
Code

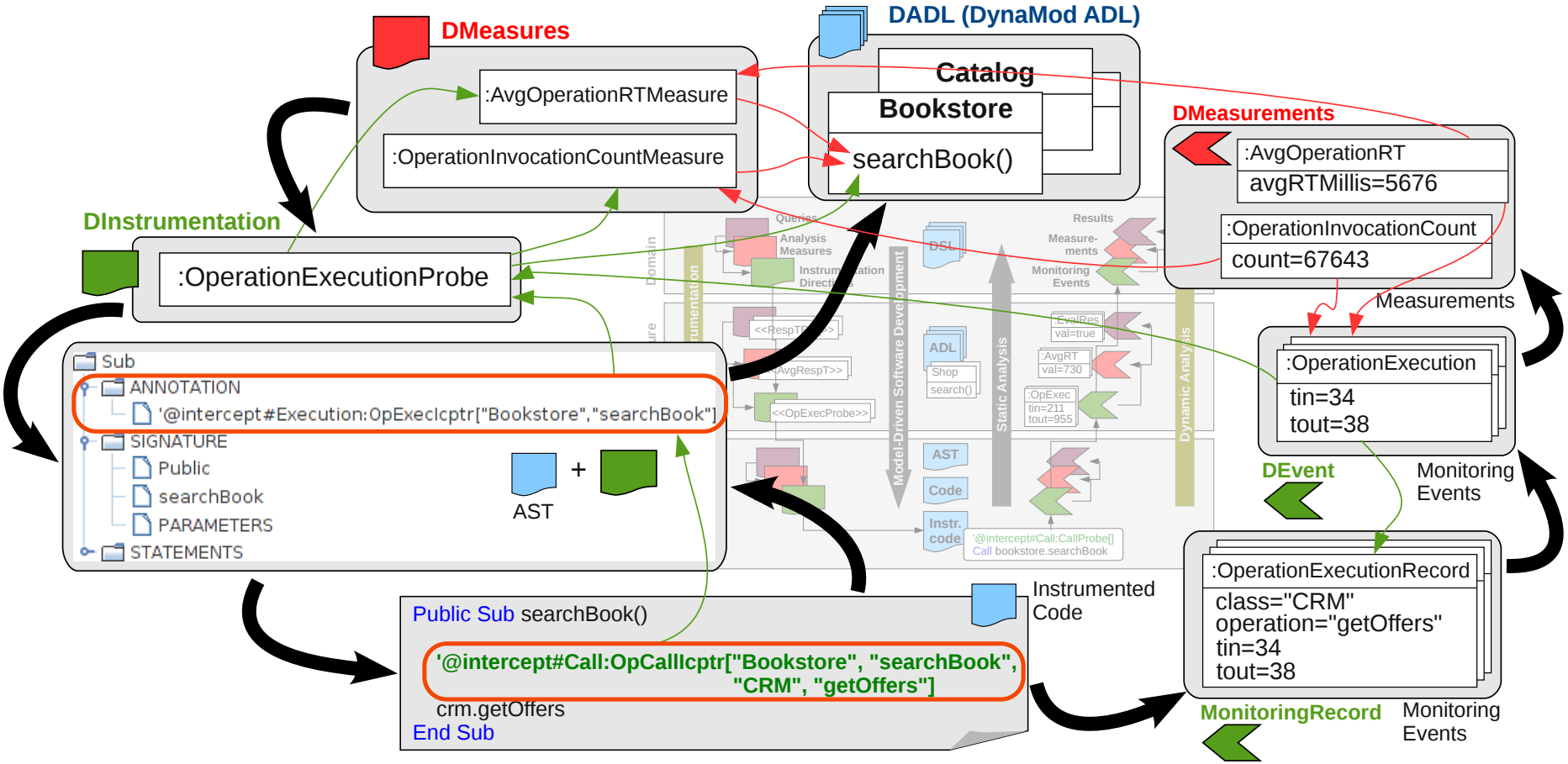

DMeasures
:AvgOperationRTMeasure
:OperationInvocationCountMeasure
DADL (DynaMod ADL)
Catalog
Bookstore
searchBook()
DInstrumentation
:OperationExecutionProbe
Sub
SIGNATURE
Public
searchBook
PARAMETERS
STATEMENTS
AST
Queries
Analysis Measures
Results
Measure-ments
Monitoring Events
DSL
ADL
Shop
search()
AST
Code
Instr. code
Model-Driven Software Development
Static Analysis
Dynamic Analysis
:EvalRes val=true
:AvgRT val=730
:OpExec tin=211 tout=955
<<OpExecProbe>>
'@intercept#Call:CallProbe[]
Call bookstore.searchBook
Public Sub searchBook()
...
crm.getOffers
End Sub
Code


DMeasures
:AvgOperationRTMeasure
:OperationInvocationCountMeasure
DADL (DynaMod ADL)
Catalog
Bookstore
searchBook()
DInstrumentation
:OperationExecutionProbe
Sub
ANNOTATION
'@intercept#Execution:OpExecIcptr["Bookstore","searchBook"]
SIGNATURE
Public
searchBook
PARAMETERS
STATEMENTS
AST
Public Sub searchBook()
...
crm.getOffers
End Sub
Code
Queries
Analysis Measures
DSL
Results
Measure-ments
Monitoring Events
Model-Driven Software Development
Static Analysis
Dynamic Analysis
ADL
Shop
search()
AST
Code
Instr. code
:EvalRes
val=true
:AvgRT
val=730
:OpExec
tin=211
tout=955
'@intercept#Call:CallProbe[]
Call bookstore.searchBook


DMeasures
:AvgOperationRTMeasure
:OperationInvocationCountMeasure
DADL (DynaMod ADL)
Catalog
Bookstore
searchBook()
DInstrumentation
:OperationExecutionProbe
Sub
ANNOTATION
'@intercept#Execution:OpExecIcptr["Bookstore","searchBook"]
SIGNATURE
Public
searchBook
PARAMETERS
STATEMENTS
AST
Public Sub searchBook()
'@intercept#Call:OpCallIcptr["Bookstore", "searchBook",
"CRM", "getOffers"]
crm.getOffers
End Sub
Instrumented Code
Queries
Analysis Measures
DSL
Results
Measure-ments
Monitoring Events
ADL
Shop
search()
AST
Code
Instr. code
Model-Driven Software Development
Static Analysis
Dynamic Analysis


DMeasures
:AvgOperationRTMeasure
:OperationInvocationCountMeasure
DADL (DynaMod ADL)
Catalog
Bookstore
searchBook()
DInstrumentation
:OperationExecutionProbe
Sub
ANNOTATION
'@intercept#Execution:OpExecIcptr["Bookstore","searchBook"]
SIGNATURE
Public
searchBook
PARAMETERS
STATEMENTS
AST
Public Sub searchBook()
'@intercept#Call:OpCallIcptr["Bookstore", "searchBook",
"CRM", "getOffers"]
crm.getOffers
End Sub
Instrumented Code
:OperationExecutionRecord
class="CRM"
operation="getOffers"
tin=34
tout=38
MonitoringRecord
Monitoring Events
Queries
Analysis Measures
DSL
Results
Measure-ments
Monitoring Events
ADL
Shop
search()
AST
Code
Instr. code
Model-Driven Software Development
Static Analysis
Dynamic Analysis
:EvalRes
val=true
:AvgRT
val=730
:OpExec
tin=211
tout=955


DMeasures
:AvgOperationRTMeasure
:OperationInvocationCountMeasure
DADL (DynaMod ADL)
Catalog
Bookstore
searchBook()
DInstrumentation
:OperationExecutionProbe
Sub
ANNOTATION
'@intercept#Execution:OpExecIcptr["Bookstore","searchBook"]
SIGNATURE
Public
searchBook
PARAMETERS
STATEMENTS
AST
Public Sub searchBook()
'@intercept#Call:OpCallIcptr["Bookstore", "searchBook",
"CRM", "getOffers"]
crm.getOffers
End Sub
Instrumented Code
Queries
Analysis Measures
Results
Measure-ments
Monitoring Events
DSL
ADL
Shop
search()
AST
Code
Instr. code
Model-Driven Software Development
Static Analysis
Dynamic Analysis
:OperationExecution
tin=34
tout=38
DEvent
Monitoring Events
:OperationExecutionRecord
class="CRM"
operation="getOffers"
tin=34
tout=38
MonitoringRecord
Monitoring Events


DMeasures
:AvgOperationRTMeasure
:OperationInvocationCountMeasure
DADL (DynaMod ADL)
Catalog
Bookstore
searchBook()
DMeasurements
:AvgOperationRT
avgRTMillis=5676
:OperationInvocationCount
count=67643
DInstrumentation
:OperationExecutionProbe
Measurements
Sub
ANNOTATION
'@intercept#Execution:OpExecIcptr["Bookstore","searchBook"]
SIGNATURE
Public
searchBook
PARAMETERS
STATEMENTS
AST
:OperationExecution
tin=34
tout=38
DEvent
Monitoring Events
Public Sub searchBook()
'@intercept#Call:OpCallIcptr["Bookstore", "searchBook", "CRM", "getOffers"]
crm.getOffers
End Sub
Instrumented Code
:OperationExecutionRecord
class="CRM"
operation="getOffers"
tin=34
tout=38
MonitoringRecord
Monitoring Events
Model-Driven Software Development
Static Analysis
Dynamic Analysis

Agenda

AOP Framework—VB6 Example

Christian-Albrechts-Universität zu Kiel

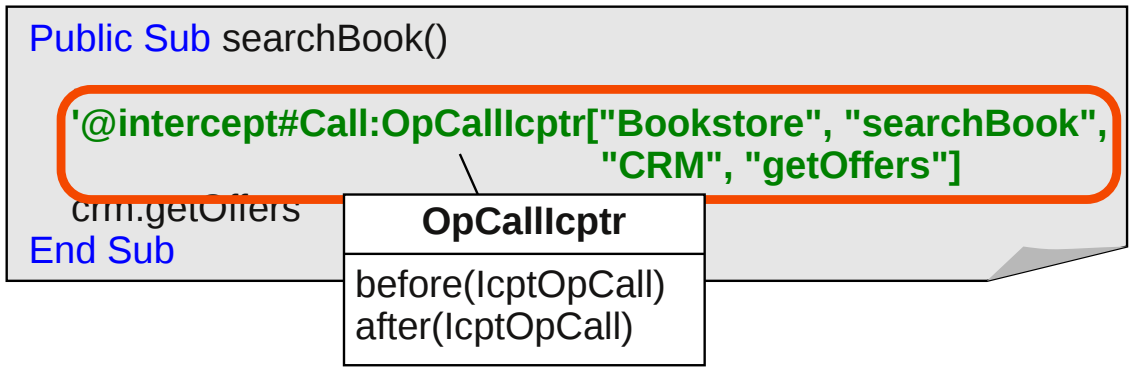
```
Public Sub searchBook()
    ...


    crm.getOffers
End Sub
```

AOP Framework—VB6 Example

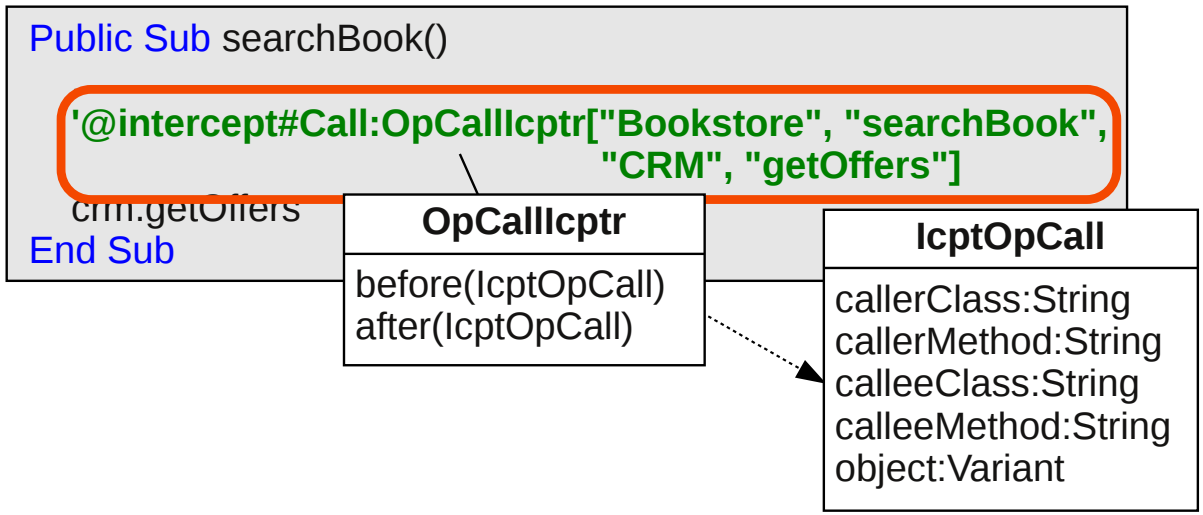
Christian-Albrechts-Universität zu Kiel

```
Public Sub searchBook()
    '@intercept#Call:OpCallIcptr["Bookstore", "searchBook",
                                 "CRM", "getOffers"]
    crm.getOffers
End Sub
```

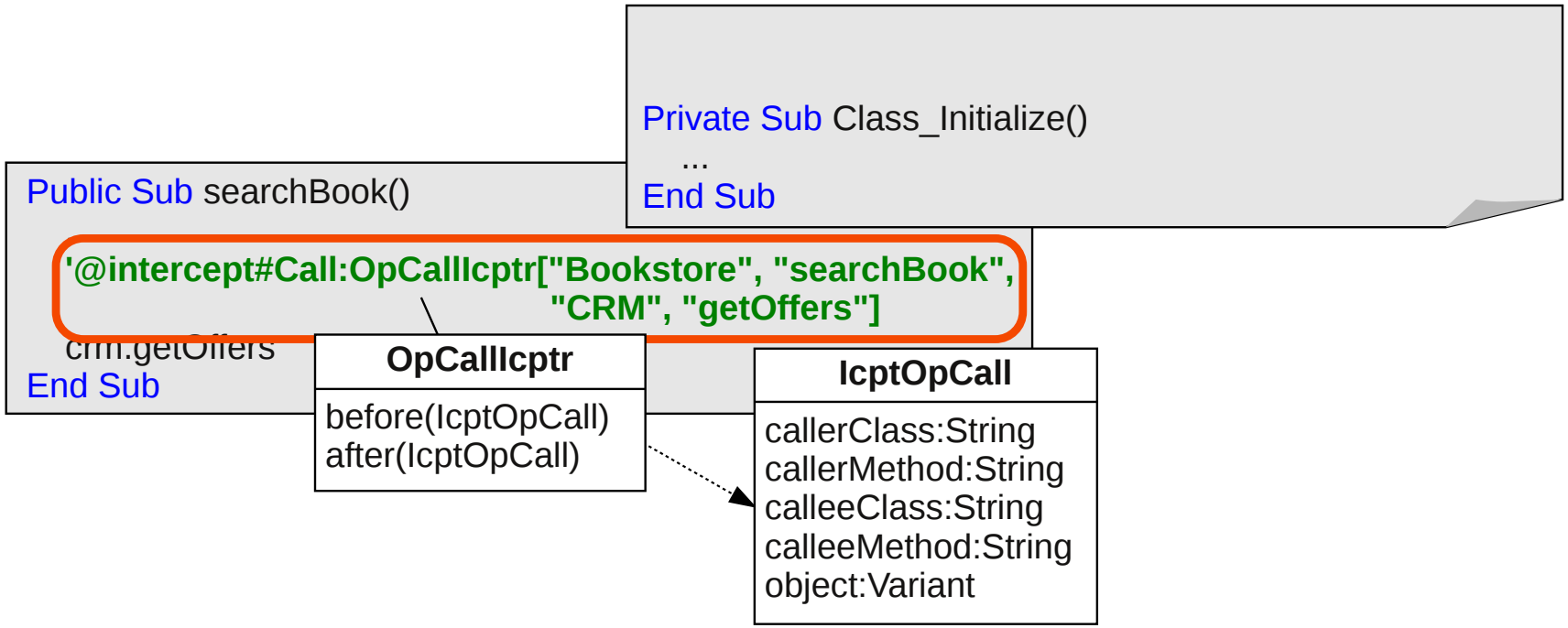
AOP Framework—VB6 Example



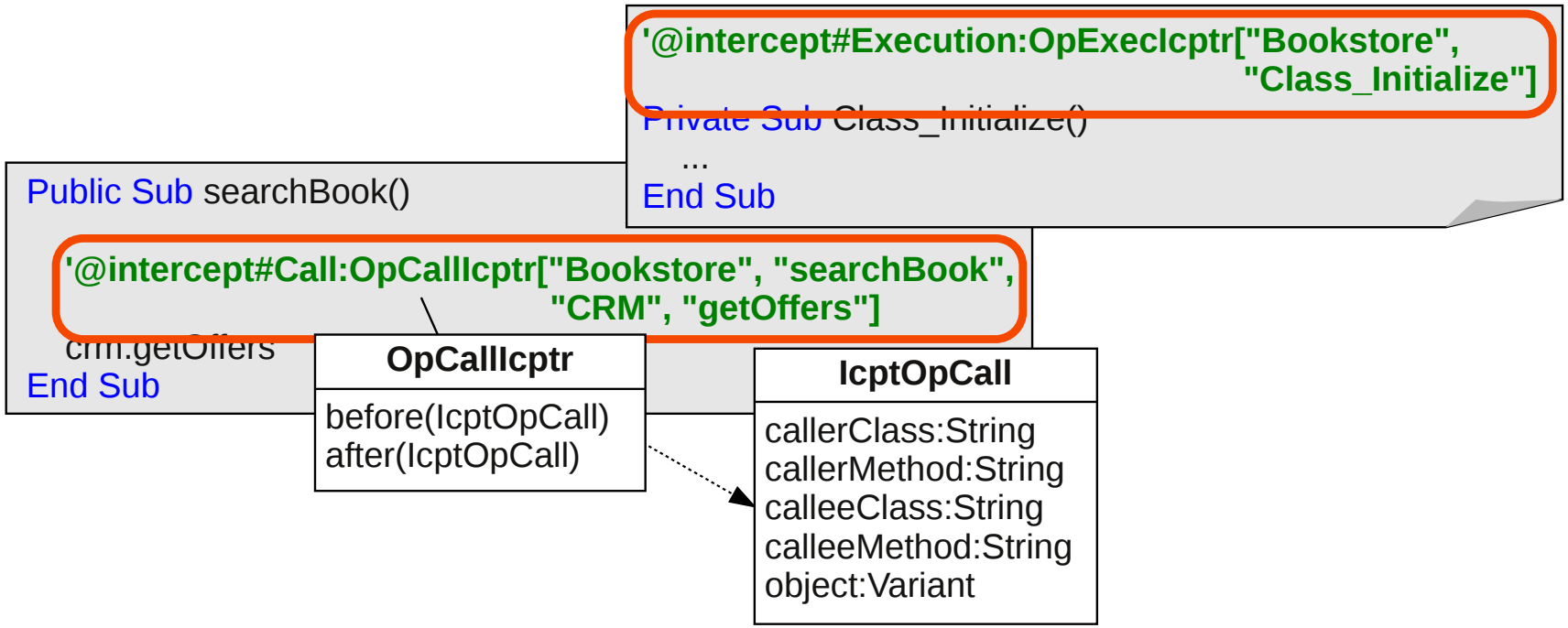
AOP Framework—VB6 Example



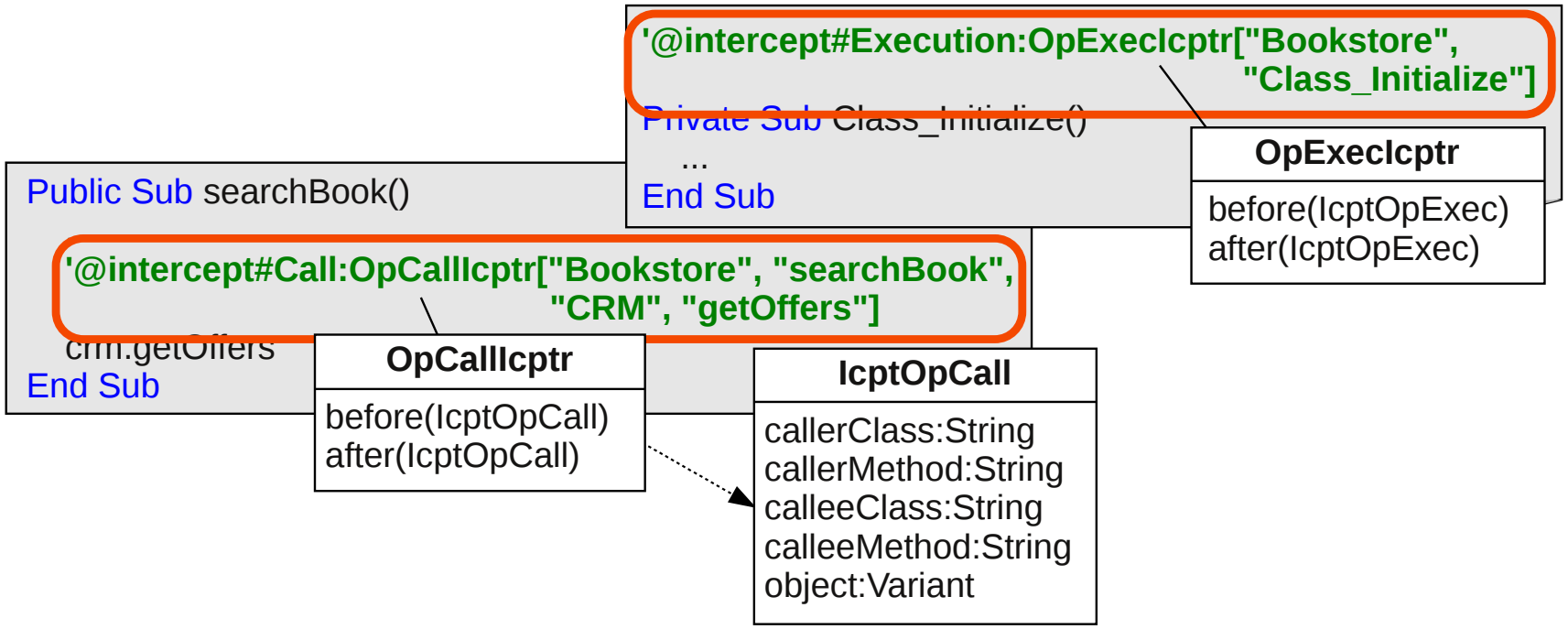
AOP Framework—VB6 Example



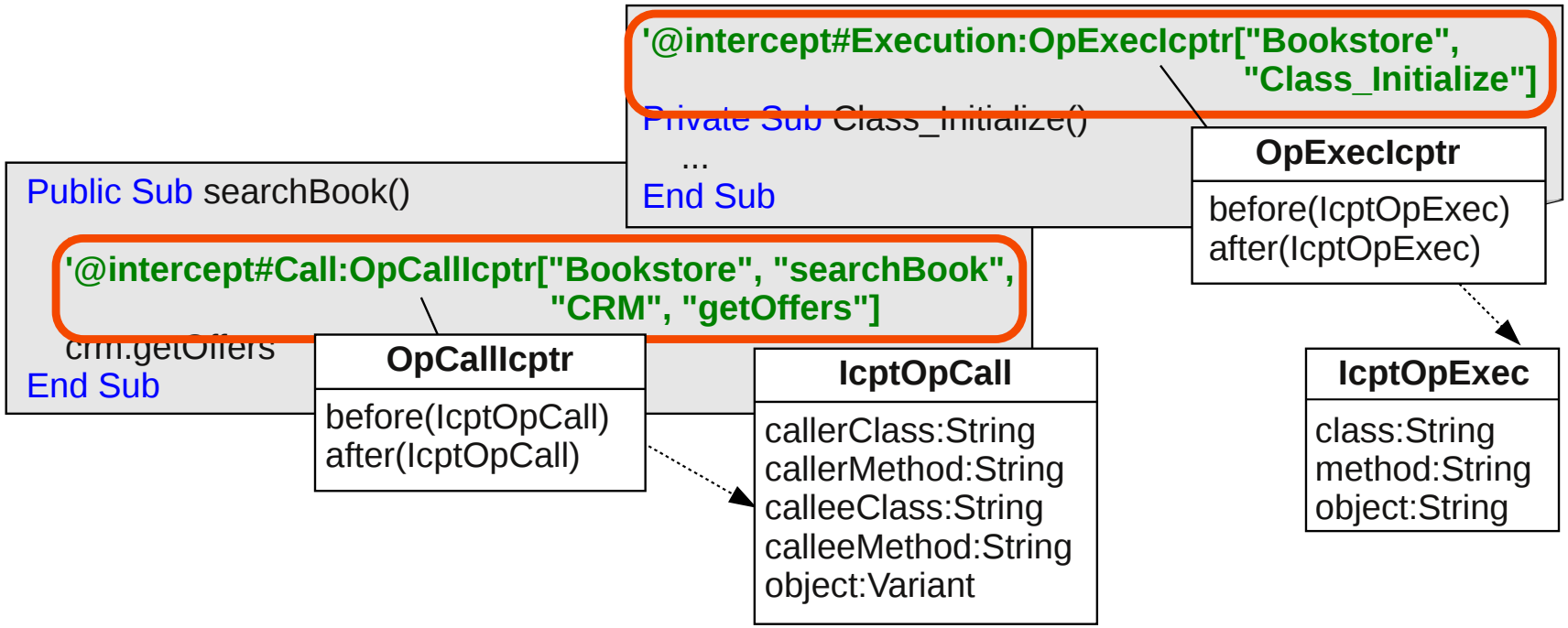
AOP Framework—VB6 Example



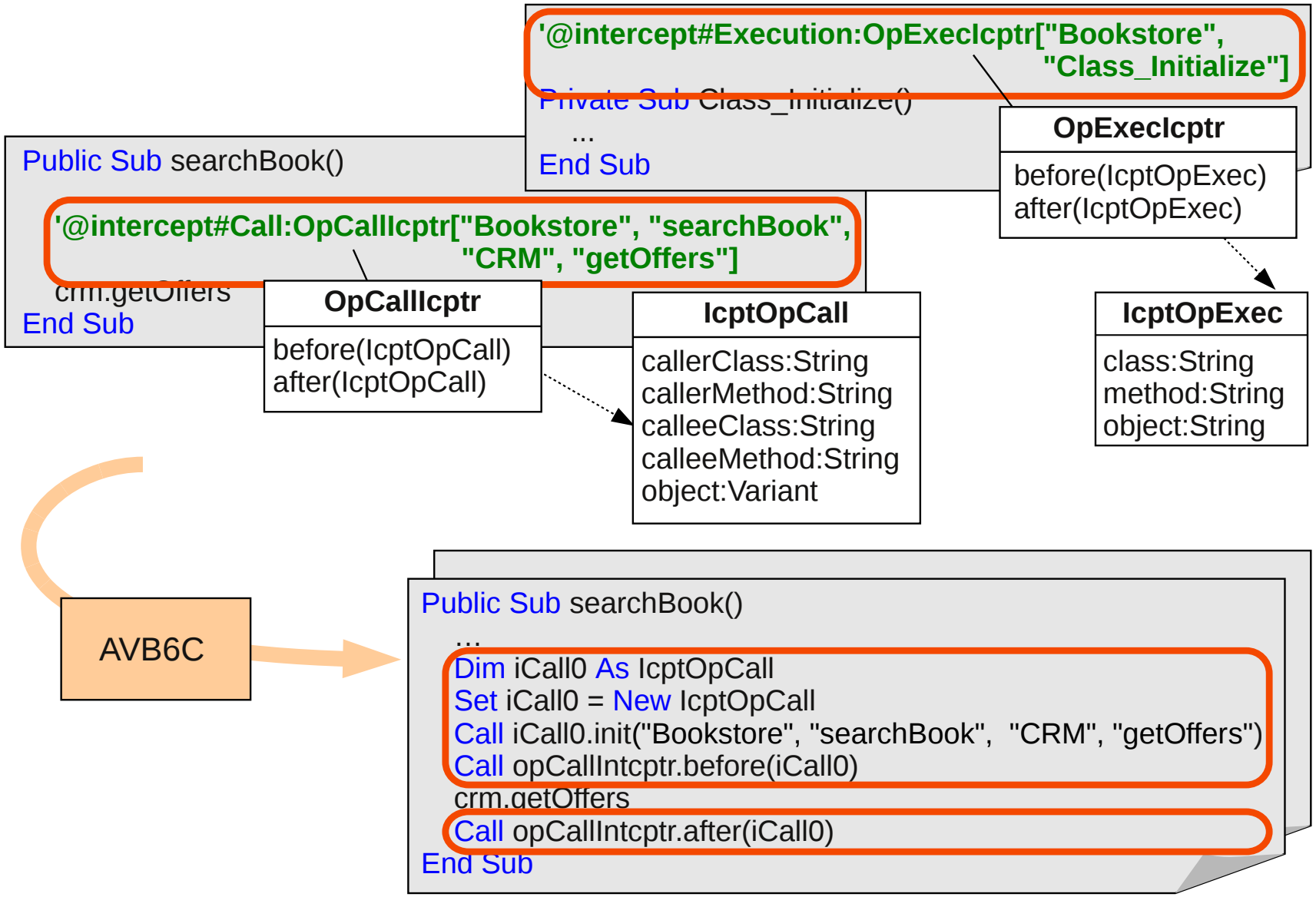
AOP Framework—VB6 Example



AOP Framework—VB6 Example

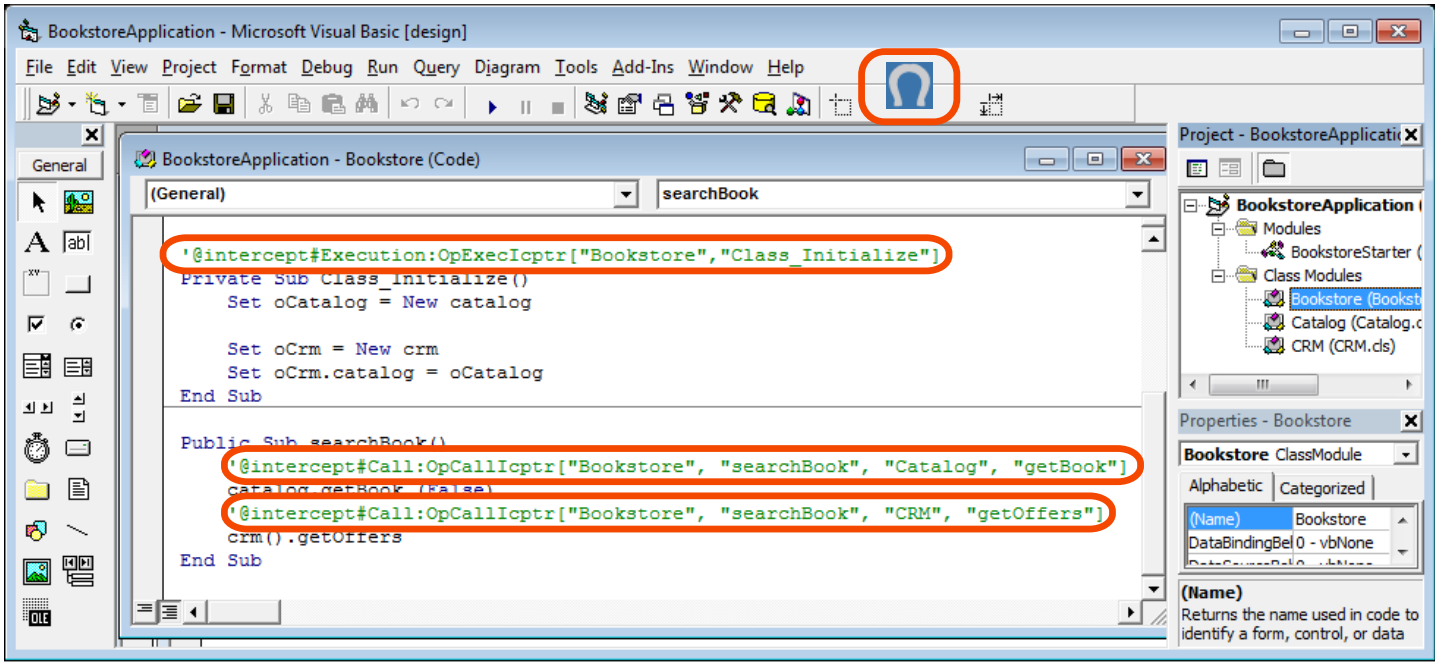


AOP Framework—VB6 Example



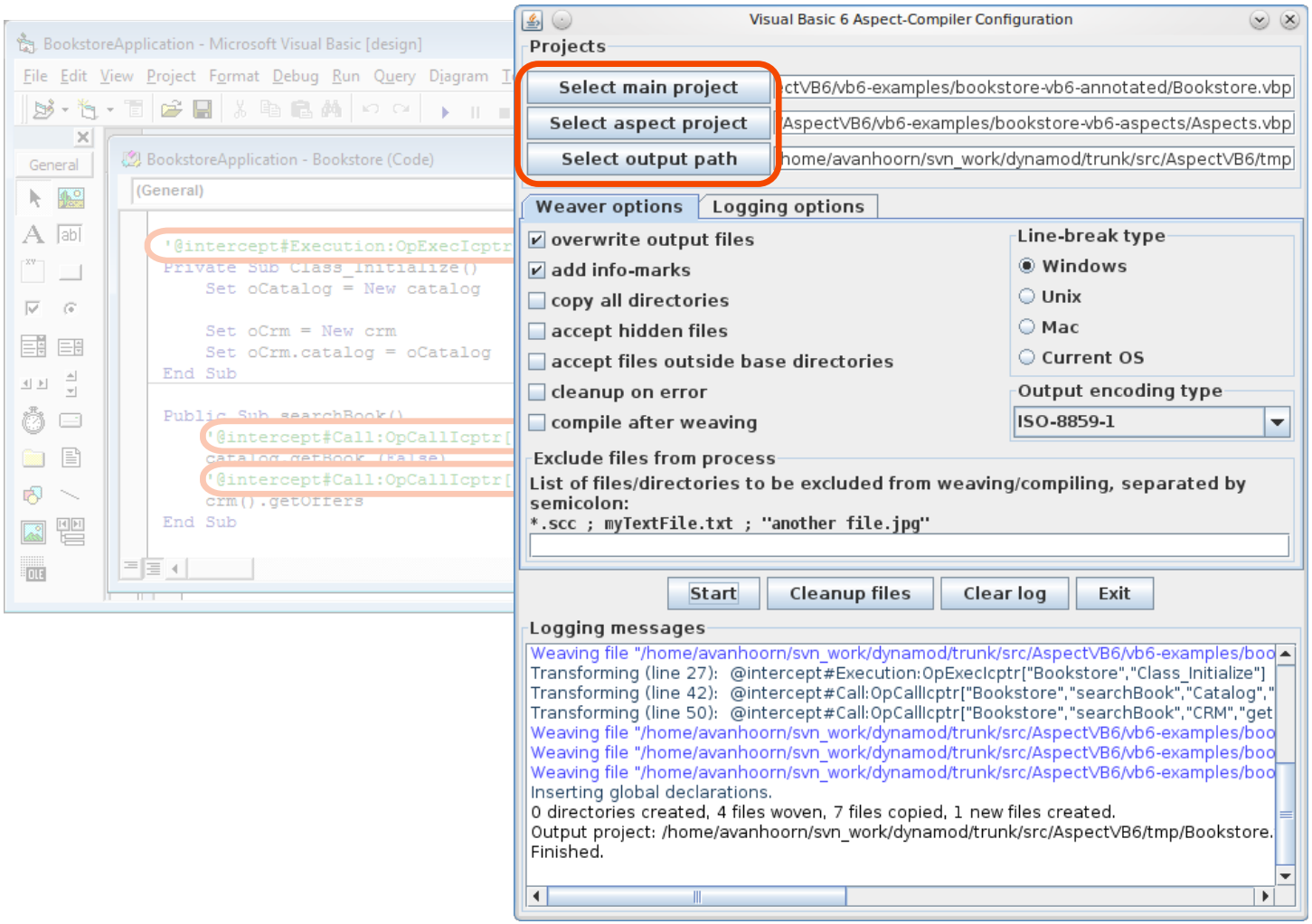
AVB6C — IDE Integration

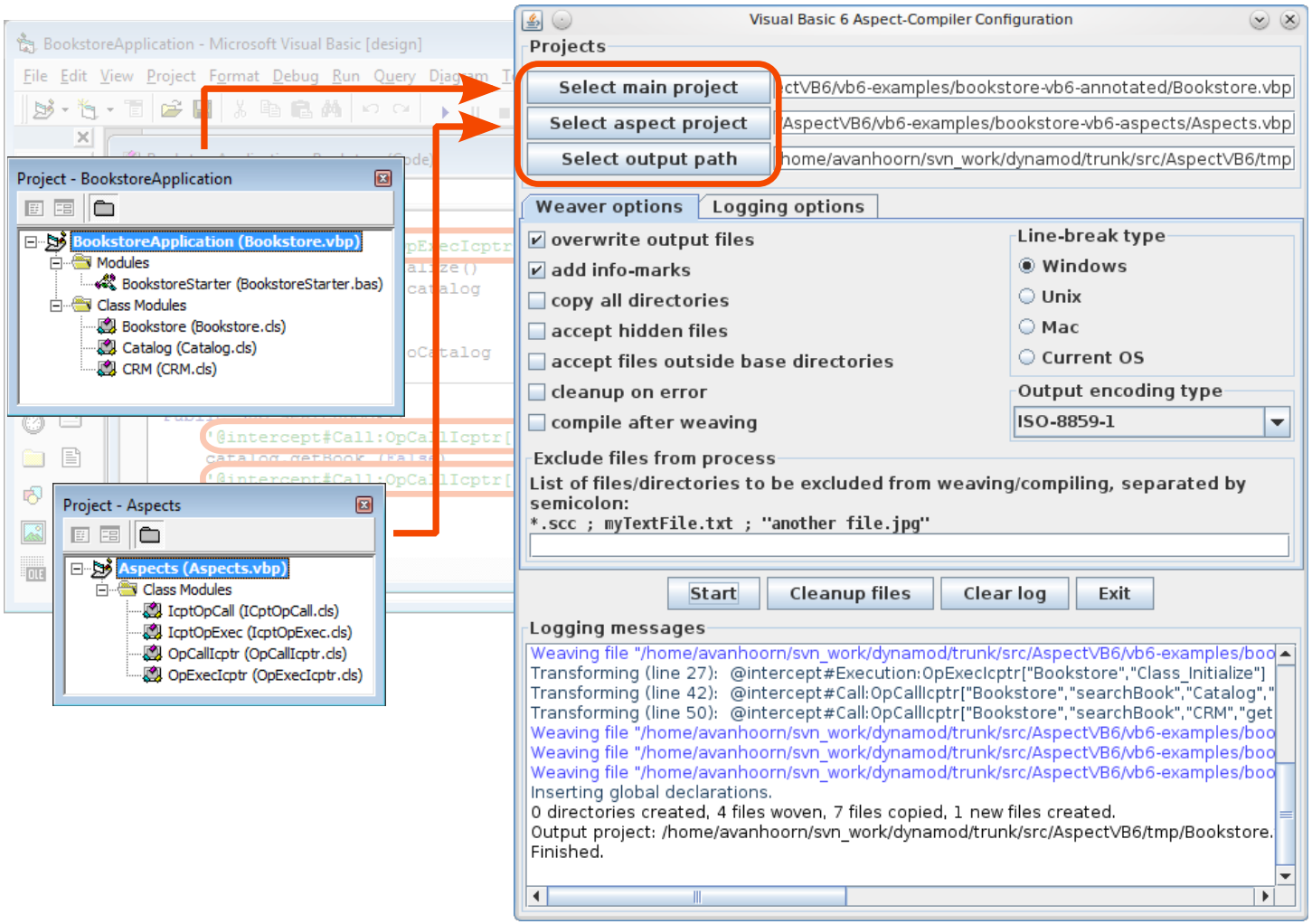
Christian-Albrechts-Universität zu Kiel

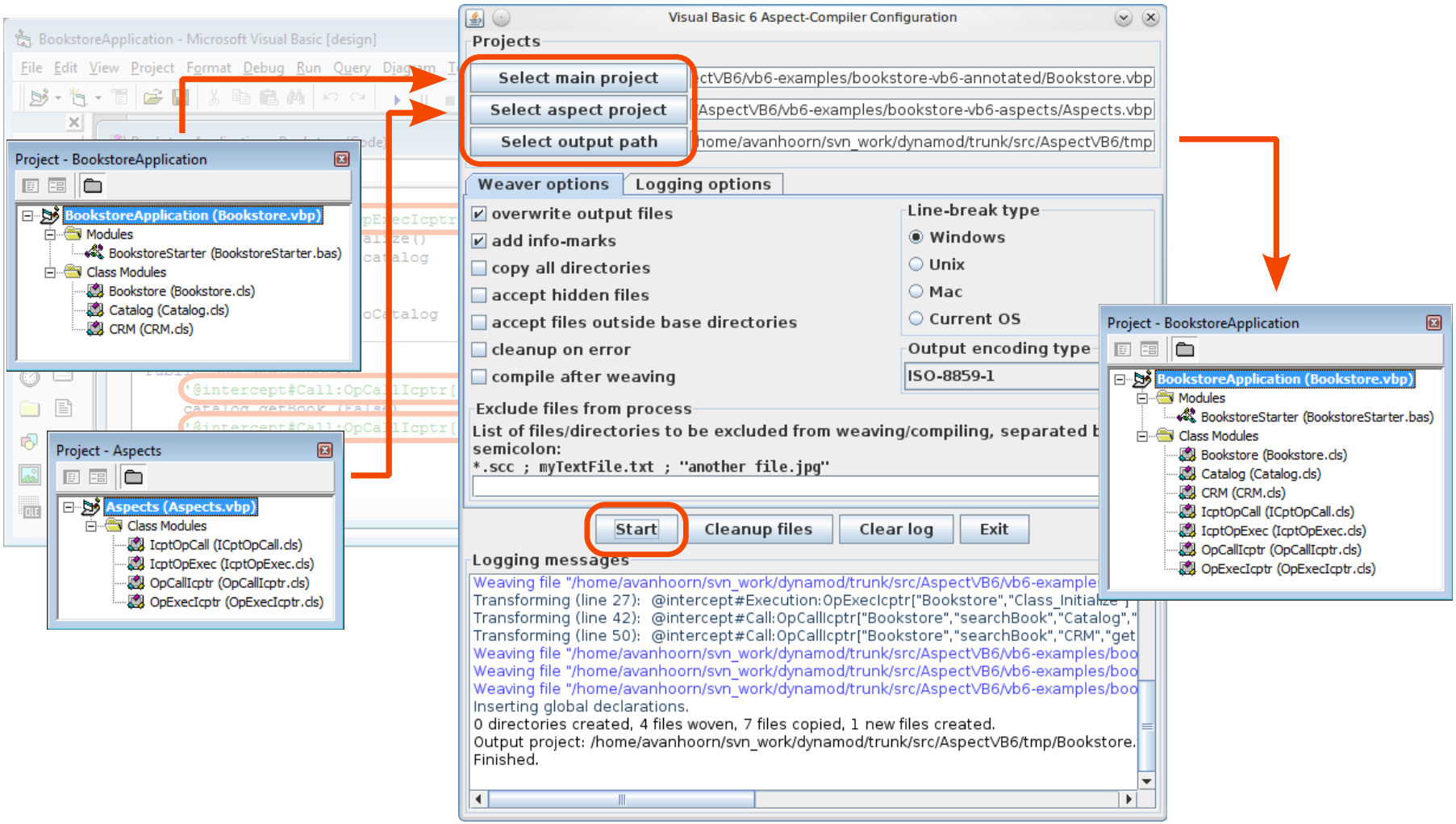


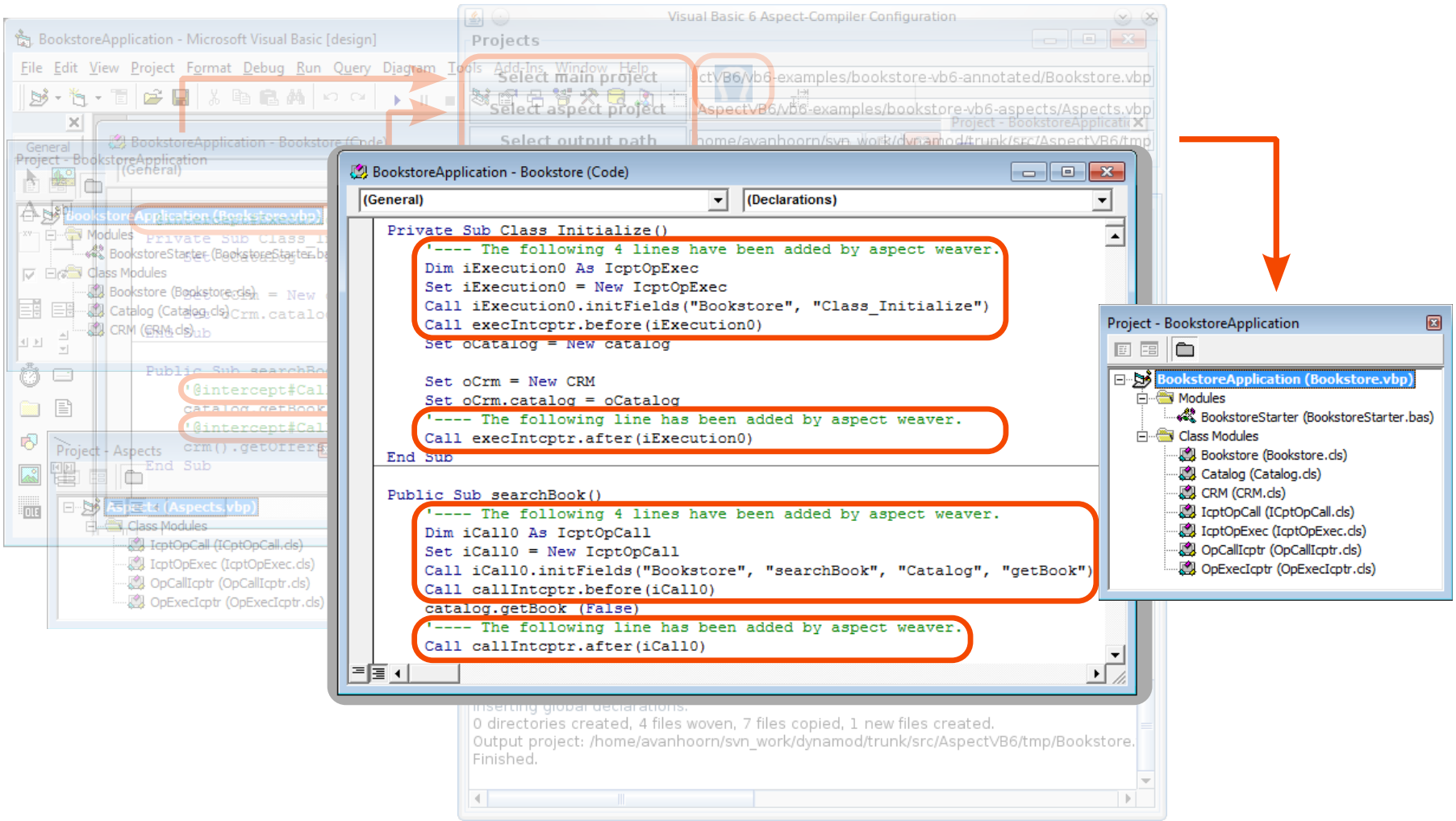
AVB6C — IDE Integration

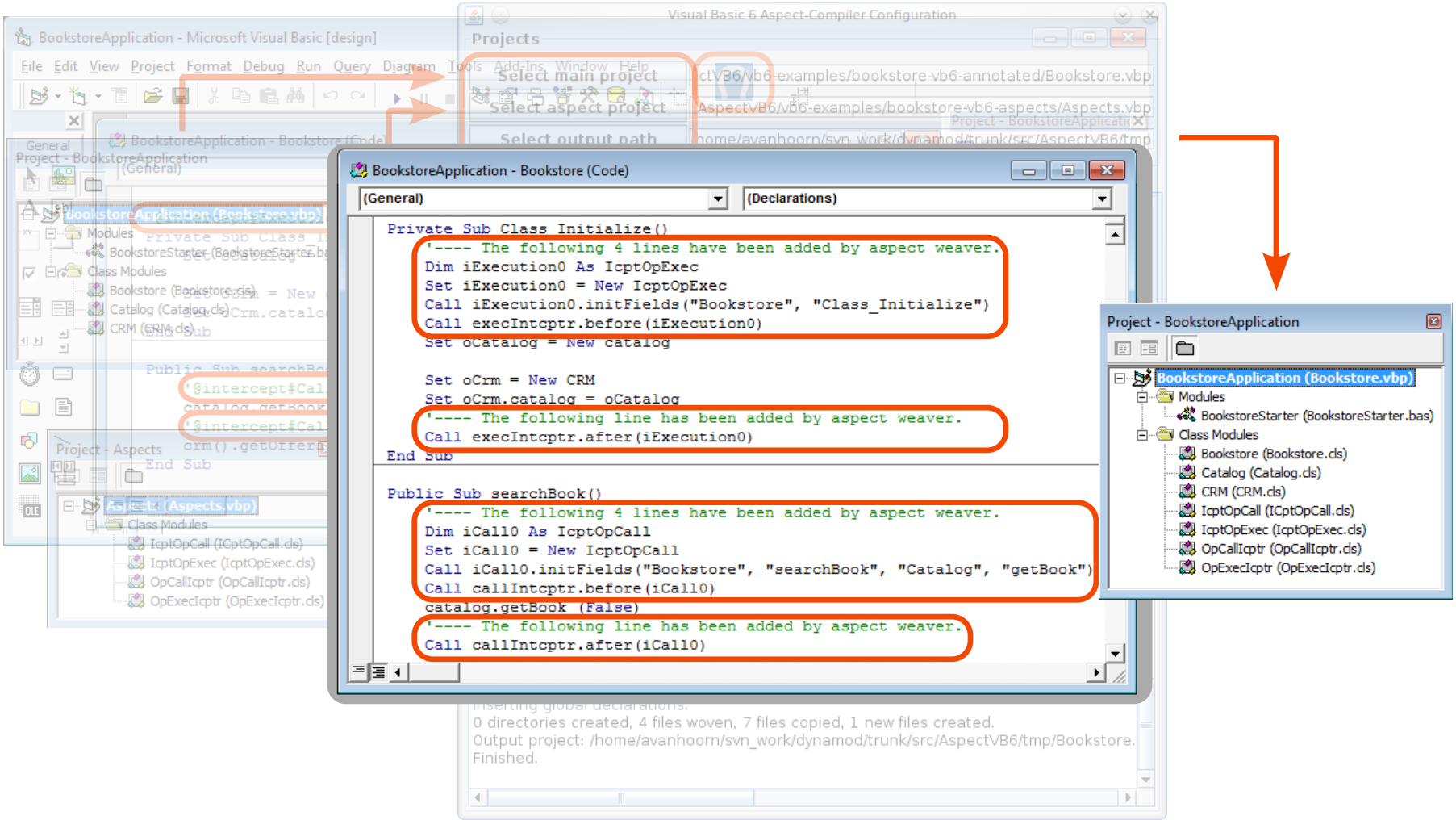
Christian-Albrechts-Universität zu Kiel




BookstoreApplication - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Query
Project - BookstoreApplication
BookstoreApplication (Bookstore.vbp)
Modules
BookstoreStarter (BookstoreStarter.bas)
Class Modules
Bookstore (Bookstore.cls)
Catalog (Catalog.cls)
CRM (CRM.cls)
Project - Aspects
Aspects (Aspects.vbp)
Class Modules
IcptOpCall (ICptOpCall.cls)
IcptOpExec (IcptOpExec.cls)
OpCallIcptr (OpCallIcptr.cls)
OpExecIcptr (OpExecIcptr.cls)
Visual Basic 6 Aspect-Compiler Configuration
Projects
Select main project
ctVB6/vb6-examples/bookstore-vb6-annotated/Bookstore.vbp
Select aspect project
AspectVB6/vb6-examples/bookstore-vb6-aspects/Aspects.vbp
Select output path
home/avanhoorn/svn_work/dynamod/trunk/src/AspectVB6/tmp
Weaver options
Logging options
overwrite output files
add info-marks
copy all directories
accept hidden files
accept files outside base directories
cleanup on error
compile after weaving
Line-break type
Windows
Unix
Mac
Current OS
Output encoding type
ISO-8859-1
Exclude files from process
List of files/directories to be excluded from weaving/compiling, separated by semicolon:
*.scc ; myTextFile.txt ; "another file.jpg"
Start
Cleanup files
Clear log
Exit
Logging messages
Weaving file "/home/avanhoorn/svn_work/dynamod/trunk/src/AspectVB6/vb6-examples/boo
Transforming (line 27): @intercept#Execution:OpExecIcptr["Bookstore","Class_Initialize"]
Transforming (line 42): @intercept#Call:OpCallIcptr["Bookstore","searchBook","Catalog","
Transforming (line 50): @intercept#Call:OpCallIcptr["Bookstore","searchBook","CRM","get
Weaving file "/home/avanhoorn/svn_work/dynamod/trunk/src/AspectVB6/vb6-examples/boo
Weaving file "/home/avanhoorn/svn_work/dynamod/trunk/src/AspectVB6/vb6-examples/boo
Weaving file "/home/avanhoorn/svn_work/dynamod/trunk/src/AspectVB6/vb6-examples/boo
Inserting global declarations.
0 directories created, 4 files woven, 7 files copied, 1 new files created.
Output project: /home/avanhoorn/svn_work/dynamod/trunk/src/AspectVB6/tmp/Bookstore.
Finished.

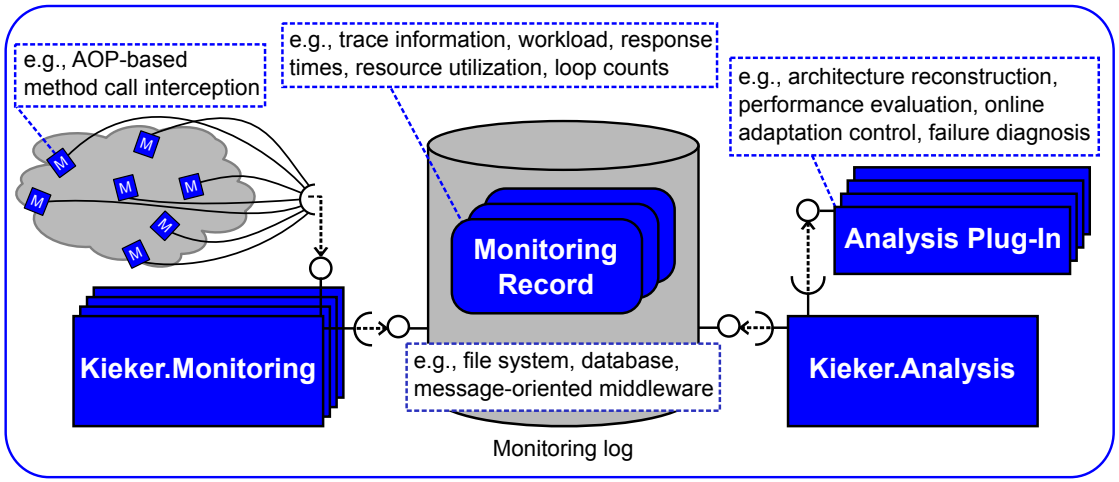

Visual Basic 6 Aspect-Compiler Configuration
Projects
Select main project
Select aspect project
Select output path
Weaver options
Logging options
overwrite output files
add info-marks
copy all directories
accept hidden files
accept files outside base directories
cleanup on error
compile after weaving
Line-break type
Windows
Unix
Mac
Current OS
Output encoding type
ISO-8859-1
Exclude files from process
List of files/directories to be excluded from weaving/compiling, separated b
semicolon:
*.scc ; myTextFile.txt ; "another file.jpg"
Start
Cleanup files
Clear log
Exit
Logging messages
Weaving file "/home/avanhoorn/svn_work/dynamod/trunk/src/AspectVB6/vb6-example
Transforming (line 27): @intercept#Execution:OpExecIcptr["Bookstore","Class_Initialize
Transforming (line 42): @intercept#Call:OpCallIcptr["Bookstore","searchBook","Catalog","
Transforming (line 50): @intercept#Call:OpCallIcptr["Bookstore","searchBook","CRM","get
Weaving file "/home/avanhoorn/svn_work/dynamod/trunk/src/AspectVB6/vb6-examples/boo
Weaving file "/home/avanhoorn/svn_work/dynamod/trunk/src/AspectVB6/vb6-examples/boo
Weaving file "/home/avanhoorn/svn_work/dynamod/trunk/src/AspectVB6/vb6-examples/boo
Inserting global declarations.
0 directories created, 4 files woven, 7 files copied, 1 new files created.
Output project: /home/avanhoorn/svn_work/dynamod/trunk/src/AspectVB6/tmp/Bookstore.
Finished.
Project - BookstoreApplication
BookstoreApplication (Bookstore.vbp)
Modules
BookstoreStarter (BookstoreStarter.bas)
Class Modules
Bookstore (Bookstore.cls)
Catalog (Catalog.cls)
CRM (CRM.cls)
Project - Aspects
Aspects (Aspects.vbp)
Class Modules
IcptOpCall (ICptOpCall.cls)
IcptOpExec (IcptOpExec.cls)
OpCallIcptr (OpCallIcptr.cls)
OpExecIcptr (OpExecIcptr.cls)
Project - BookstoreApplication
BookstoreApplication (Bookstore.vbp)
Modules
BookstoreStarter (BookstoreStarter.bas)
Class Modules
Bookstore (Bookstore.cls)
Catalog (Catalog.cls)
CRM (CRM.cls)
IcptOpCall (ICptOpCall.cls)
IcptOpExec (IcptOpExec.cls)
OpCallIcptr (OpCallIcptr.cls)
OpExecIcptr (OpExecIcptr.cls)


BookstoreApplication - Bookstore (Code)
(General)
(Declarations)
Private Sub Class_Initialize()
'---- The following 4 lines have been added by aspect weaver.
Dim iExecution0 As IcptOpExec
Set iExecution0 = New IcptOpExec
Call iExecution0.initFields("Bookstore", "Class_Initialize")
Call execIntcptr.before(iExecution0)
Set oCatalog = New catalog
Set oCrm = New CRM
Set oCrm.catalog = oCatalog
'---- The following line has been added by aspect weaver.
Call execIntcptr.after(iExecution0)
End Sub
Public Sub searchBook()
'---- The following 4 lines have been added by aspect weaver.
Dim iCall0 As IcptOpCall
Set iCall0 = New IcptOpCall
Call iCall0.initFields("Bookstore", "searchBook", "Catalog", "getBook")
Call callIntcptr.before(iCall0)
catalog.getBook (False)
'---- The following line has been added by aspect weaver.
Call callIntcptr.after(iCall0)
Project - BookstoreApplication
BookstoreApplication (Bookstore.vbp)
Modules
BookstoreStarter (BookstoreStarter.bas)
Class Modules
Bookstore (Bookstore.cls)
Catalog (Catalog.cls)
CRM (CRM.cls)
IcptOpCall (ICptOpCall.cls)
IcptOpExec (IcptOpExec.cls)
OpCallIcptr (OpCallIcptr.cls)
OpExecIcptr (OpExecIcptr.cls)
Visual Basic 6 Aspect-Compiler Configuration
Projects
Select main project
Select aspect project
Select output path
0 directories created, 4 files woven, 7 files copied, 1 new files created.
Output project: /home/avanhoorn/svn_work/dynamod/trunk/src/AspectVB6/tmp/Bookstore.
Finished.



Acknowledgment: Our DynaMod student assistants, **Eike Schulz and Benjamin Schnoor**, are responsible for large parts of the implementation!

Agenda

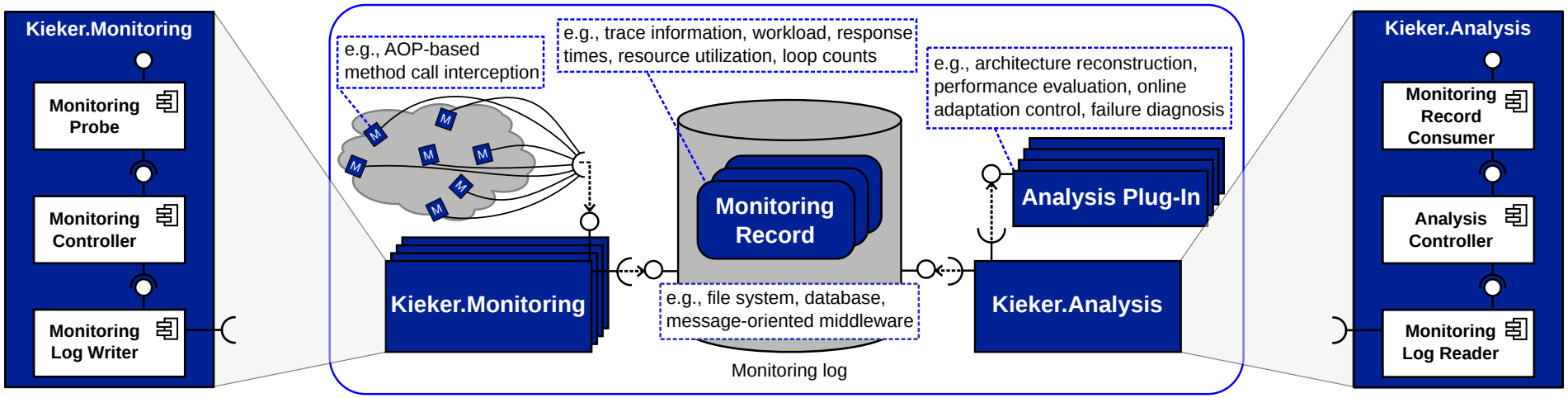


Kieker Framework — Core Characteristics [vHRH+09]

- Flexible architecture (custom *probes*, *readers*, *writers*, *analysis plug-ins*)
- Integrated & extensible *record* type model for monitoring & analysis
- Logging, reconstruction, analysis/visualization of (distributed) traces
- Low overhead (designed for continuous operation in multi-user systems)
- Evaluated in industry case studies

Kieker

http://kieker.sourceforge.net

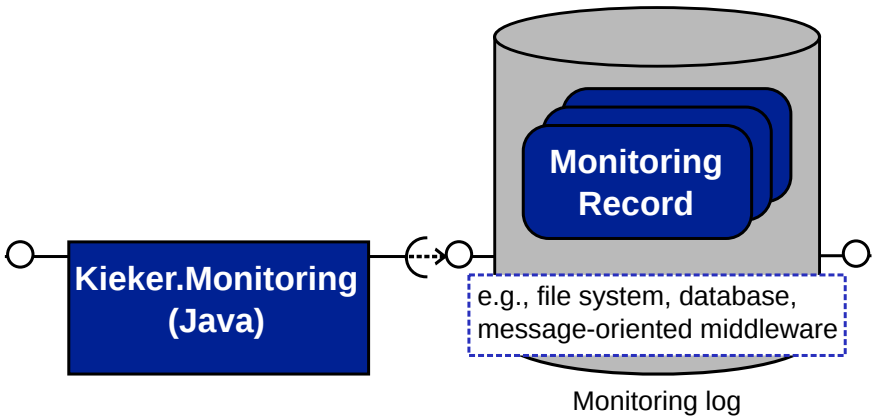


Kieker Framework — Core Characteristics [vHRH+09]

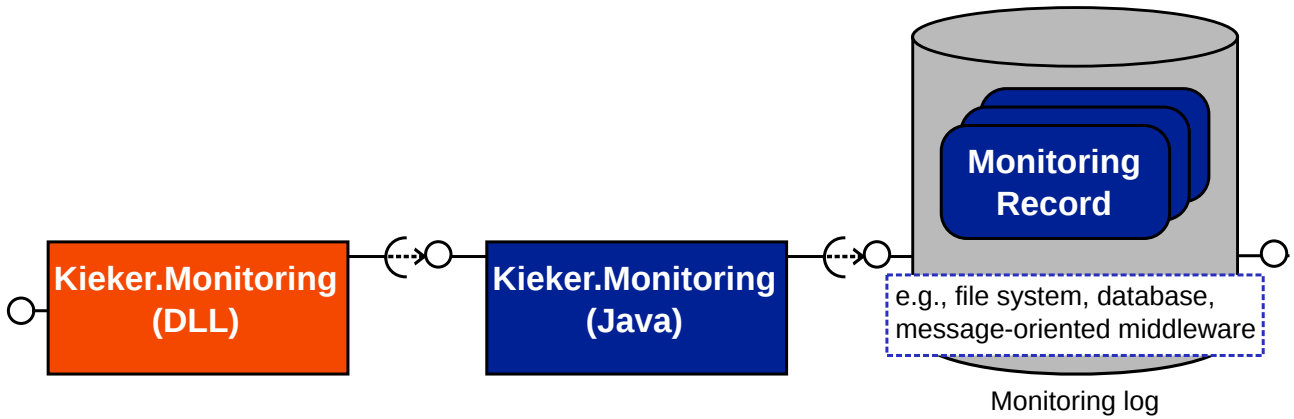
- Flexible architecture (custom *probes*, *readers*, *writers*, *analysis plug-ins*)
- Integrated & extensible *record* type model for monitoring & analysis
- Logging, reconstruction, analysis/visualization of (distributed) traces
- Low overhead (designed for continuous operation in multi-user systems)
- Evaluated in industry case studies

Kieker

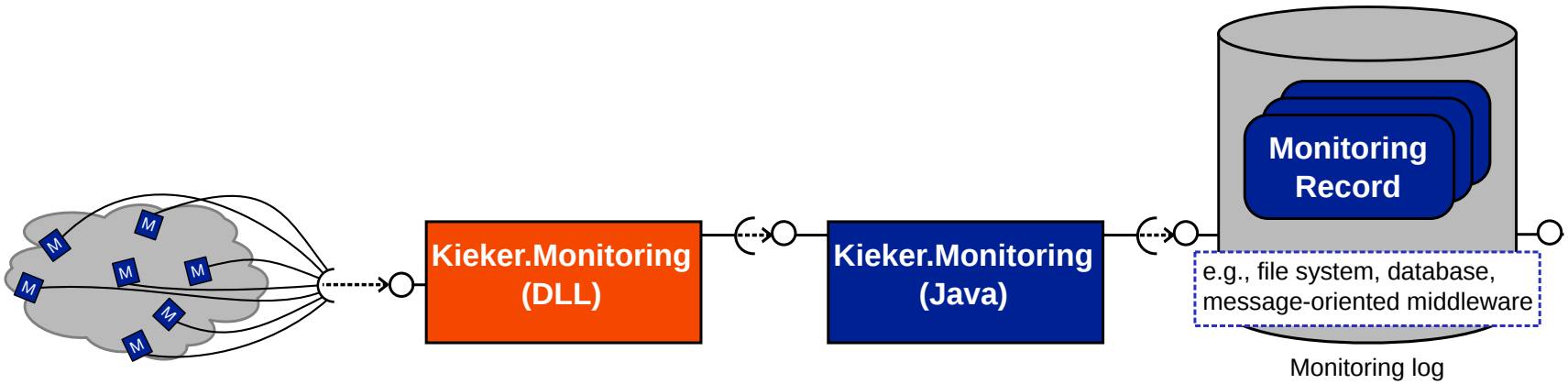
http://kieker.sourceforge.net



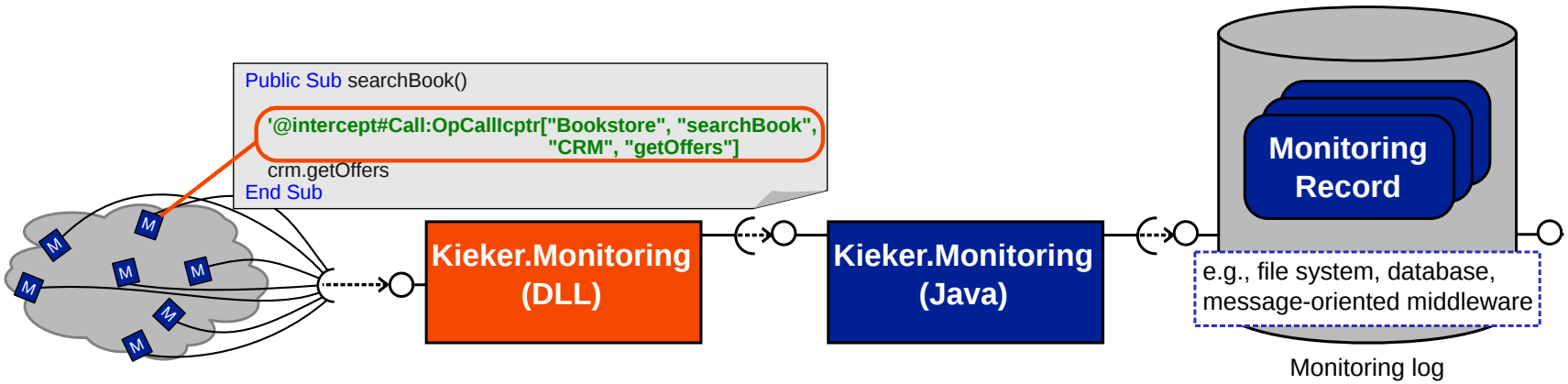
- Goal: Reuse existing Java-based Kieker.Monitoring component



- Goal: Reuse existing Java-based Kieker.Monitoring component
- Requires communication bridge among legacy language and Java



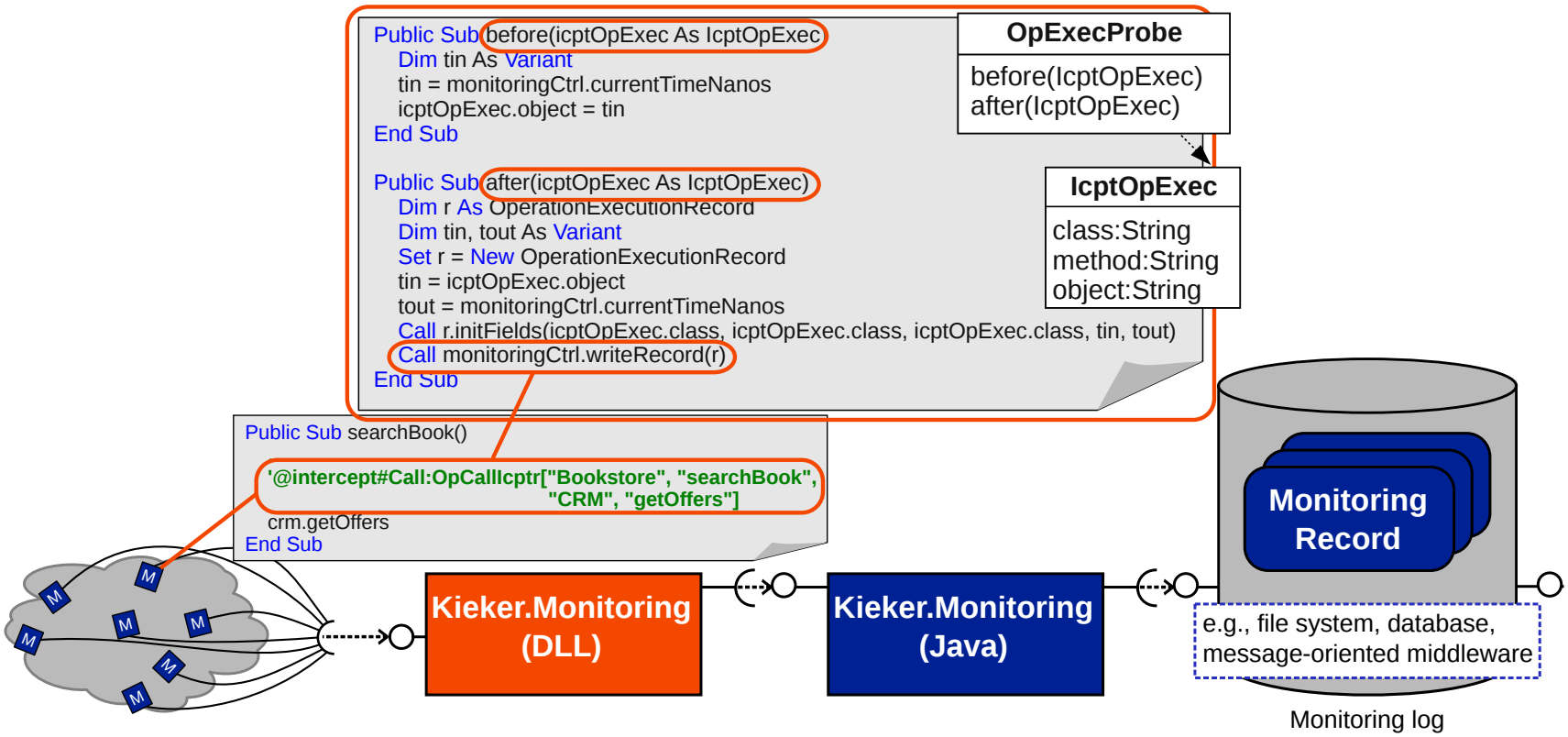
- Goal: Reuse existing Java-based Kieker.Monitoring component
- Requires communication bridge among legacy language and Java
- Instantiation of presented AOP framework for monitoring probes



- Goal: Reuse existing Java-based Kieker.Monitoring component
- Requires communication bridge among legacy language and Java
- Instantiation of presented AOP framework for monitoring probes
- Developed proof-of-concept implementation for VB6

Instrumentation of Legacy Languages

Christian-Albrechts-Universität zu Kiel



- Goal: Reuse existing Java-based Kieker.Monitoring component
- Requires communication bridge among legacy language and Java
- Instantiation of presented AOP framework for monitoring probes
- Developed proof-of-concept implementation for VB6

Summary

- Context: DynaMod project for model-driven modernization
- Model-driven instrumentation for dynamic analysis
- AOP for legacy languages
- Monitoring instrumentation employing AOP framework and Kieker
- Current focus on Visual Basic 6

Summary

- Context: DynaMod project for model-driven modernization
- Model-driven instrumentation for dynamic analysis
- AOP for legacy languages
- Monitoring instrumentation employing AOP framework and Kieker
- Current focus on Visual Basic 6

Future Work

- Refine and extend meta-models for queries, measures, instrumentation etc.
- Development of corresponding tool support — e.g., M2M/M2Code transformations
- Evaluate approach on other abstraction layers — e.g., DSLs, ASTs

Summary

- Context: DynaMod project for model-driven modernization
- Model-driven instrumentation for dynamic analysis
- AOP for legacy languages
- Monitoring instrumentation employing AOP framework and Kieker
- Current focus on Visual Basic 6

Future Work

- Refine and extend meta-models for queries, measures, instrumentation etc.
- Development of corresponding tool support — e.g., M2M/M2Code transformations
- Evaluate approach on other abstraction layers — e.g., DSLs, ASTs
- Additional AOP features — e.g., loops, branches
- Additional programming languages — e.g., COBOL, Natural, Structured Text
- Study performance overhead

Literature

Thomas Stahl and Markus Völter.

Model-Driven Software Development – Technology, Engineering, Management.
Wiley & Sons, 2006.

André van Hoorn, Sören Frey, Wolfgang Goerigk, Wilhelm Hasselbring, Holger Knoche, Sönke Köster, Harald Krause, Marcus Porembski, Thomas Stahl, Marcus Steinkamp, and Norman Wittmüss.
DynaMod project: Dynamic analysis for model-driven software modernization.
In Andreas Fuhr, Wilhelm Hasselbring, Volker Riediger, Magiel Bruntink, and Kostas Kontogiannis, editors, *Joint Proceedings of the 1st International Workshop on Model-Driven Software Migration (MDSM 2011) and the 5th International Workshop on Software Quality and Maintainability (SQM 2011)*, volume 708 of *CEUR Workshop Proceedings*, pages 12–13, March 2011.
Invited paper.

André van Hoorn, Holger Knoche, Wolfgang Goerigk, and Wilhelm Hasselbring.

Model-driven instrumentation for dynamic analysis of legacy software systems.
In *Proceedings of the 13. Workshop Software-Reengineering (WSR '11)*, 2011.

André van Hoorn, Matthias Rohr, Wilhelm Hasselbring, Jan Waller, Jens Ehlers, Sören Frey, and Dennis Kieselhorst.

Continuous monitoring of software services: Design and application of the Kieker framework.
Technical Report TR-0921, Department of Computer Science, University of Kiel, Germany, November 2009.