Sub-method Structural and Behavioral Reflection

Marcus Denker

Universität Bern

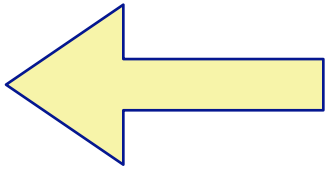
The Systems of the future...

- ... are getting larger and more complex
- ... are getting more and more dependent on each other
- **The demands are changing**

Examples of New Demands

- Dynamic Analysis
  - Fine-grained selection
  - Install / retract at runtime
  - Complete system
- Development Environment
  - Complete representation of the system
  - Extensible

Reflection

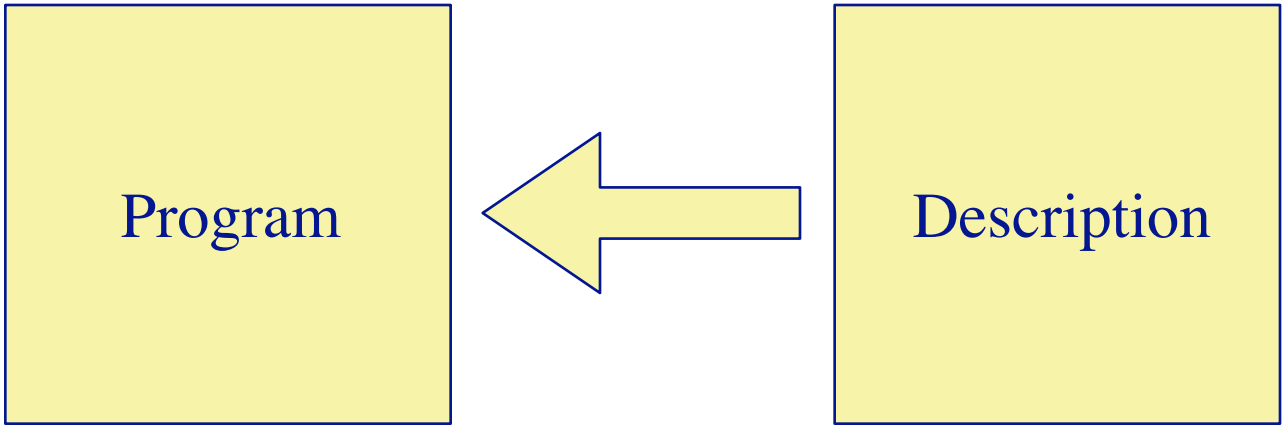


004-StringZippedAndTest-md.1.cs

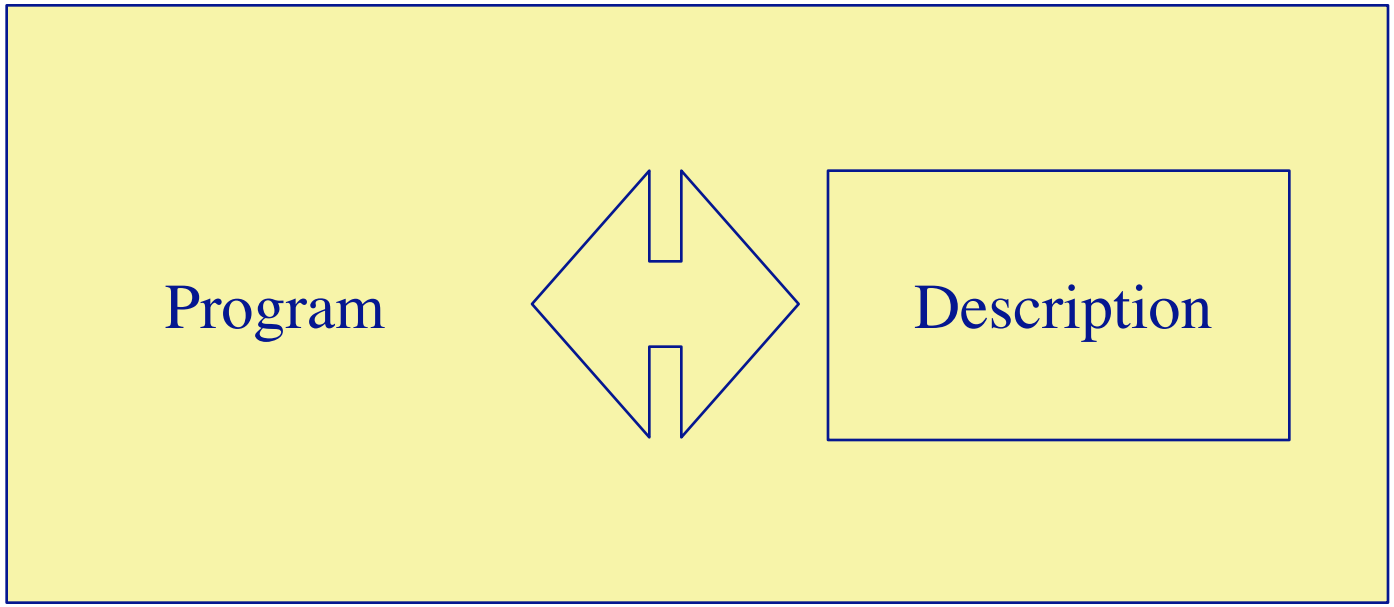
```
'From Squeak3.9 of 7 November 2006 [latest update: #7067] on 5 February 2007 at 3:25:56 pm'!
"Change Set:        StringZippedAndTest-md
Date:           5 February 2007
Author:         Marcus Denker

String has #unipped, but no #zipped. This
cs addes String>>zipped (originally from Diego Gomez Deck) and a test for unzipped/zipped.

This changeset is neutral to the question of zipped/unzipped  beeing in String, but if there is
#unzipped, there should be #zipped. And there should be a test.
"!


!String methodsFor: 'converting' stamp: 'dgd 11/26/2005 21:19'!
zipped
    | stream gzstream |

    stream := RWBinaryOrTextStream on: String new.

    gzstream := GZipWriteStream on: stream.
    gzstream nextPutAll: self.
    gzstream close.
    stream reset.

    ^ stream contents.
! !


!StringTest methodsFor: 'tests - converting' stamp: 'md 2/5/2007 15:21'!
testZipped
    | compressed |

    compressed := 'hello' zipped.
    self assert: (compressed unzipped = 'hello').! !
```

Line: 1 Column: 1 Plain Text Tab Size: 4

Reflection



Reflection



Query and **Change**

Repeat: Demands

- Dynamic Analysis
  - Fine-grained Selection
  - Install / retract at runtime
  - Complete System
- Development Environment
  - Complete representation of the system
  - Extensible

Reflection to the Rescue

- Where?
- At runtime!
- Complete Structure
- Everywhere!

Reflection to the Rescue

- Where? **Solved** (*Partial Behavioral Reflection, Eric Tanter)*
- **At runtime!**
- **Complete structure**
- **Everywhere!**

Three Problems of Reflection

1. Partial behavioral reflection needs to be anticipated

2. Structural reflection is limited to the granularity of a method

3. Behavioral reflection cannot be applied to the whole system

Three Problems of Reflection

1. Anticipation

2. Structural reflection is limited to the granularity of a method

3. Behavioral reflection cannot be applied to the whole system

Three Problems of Reflection

1. Anticipation

2. Sub-method Structure

3. Behavioral reflection cannot be applied to the whole system

Three Problems of Reflection

1. Anticipation
2. Sub-method Structure
3. Context

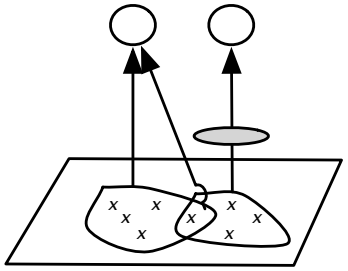
Thesis

To support **unanticipated** behavioral reflection, reflection needs to be extended with **sub-method structure** and with the concept of **context**.

Contributions of the Dissertation

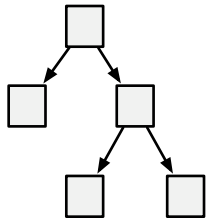
- *Unanticipated* partial behavioral reflection
- Sub-Method Structural Reflection
- Partial Behavioral Reflection *using Annotations*
- *Contextual Reflection*

Roadmap



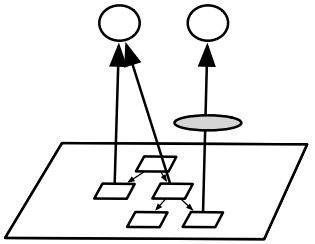
1. **Unanticipated** partial behavioral reflection
2. Sub-Method Structure
3. Revisit Partial Reflection
4. Context

Roadmap



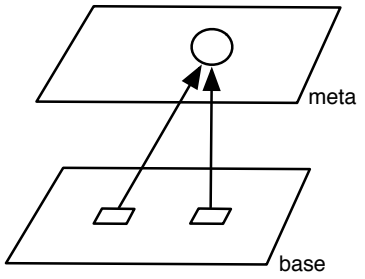
1. Unanticipated partial behavioral reflection
2. **Sub-Method Structure**
3. Revisit Partial Reflection
4. Context

Roadmap



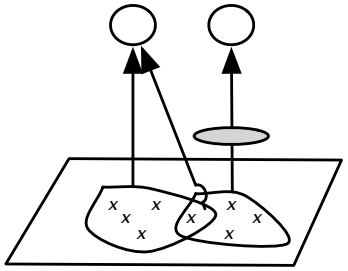
1. Unanticipated partial behavioral reflection
2. Sub-Method Structure
3. **Revisit Partial Reflection**
4. Context

Roadmap



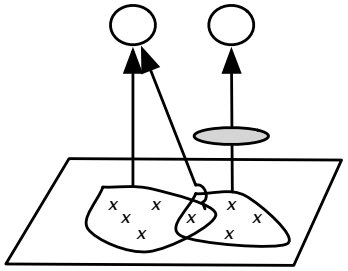
1. Unanticipated partial behavioral reflection
2. Sub-Method Structure
3. Revisit Partial Reflection
4. **Context**

Roadmap

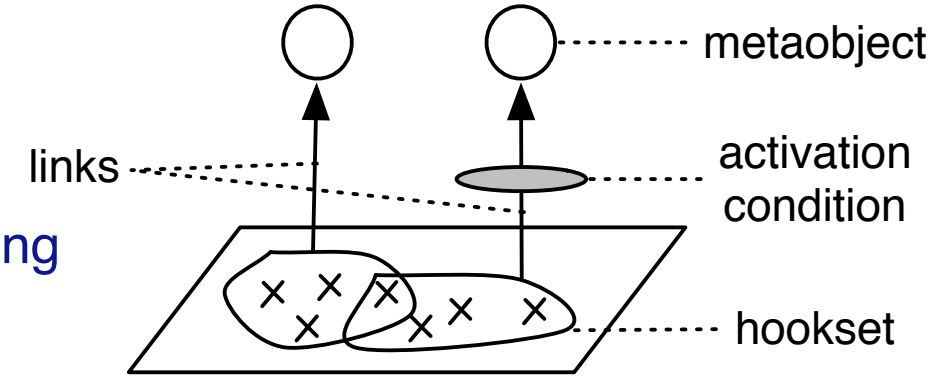


1. **Unanticipated** partial behavioral reflection
2. Sub-Method Structure
3. Revisit Partial Reflection
4. Context

Reflex: Partial Behavioral Reflection

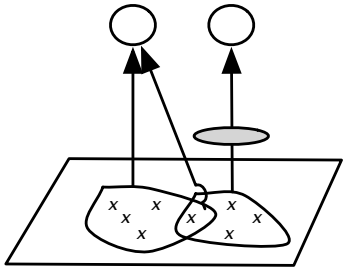


- Hooksets: collection of operation occurrences
- Links
  - Bind hooksets to meta-objects
  - Define protocol between base and meta
- Goals
  - Highly selective reification
  - Flexible meta-level engineering
    - *Protocol specification*
    - *Cross-cutting hooksets*



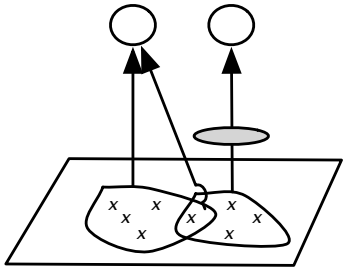
Tanter, OOPSLA03

Example: Live Analysis



- Typical Web application (e.g. Wiki)
- Shows performance problem under high load
- Goals:
  - Profile and fix the problem
  - No restart / interruption of service

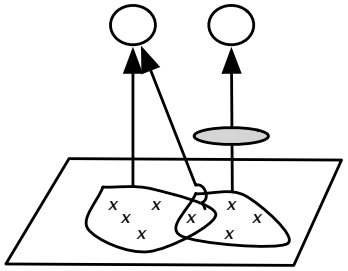
Live Analysis



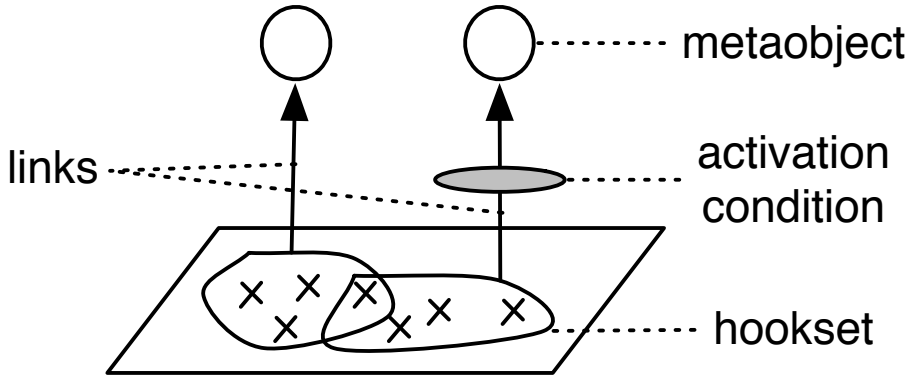
- **Install** profiler
- Analyze
- **Retract** profiler

... while the system is running!

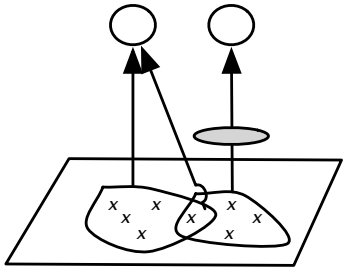
Live Profiling



- Operation:
  - Method Execution (around)
- Hookset:
  - All execution operations in the wiki package
- Metaobject:
  - A profiling tool



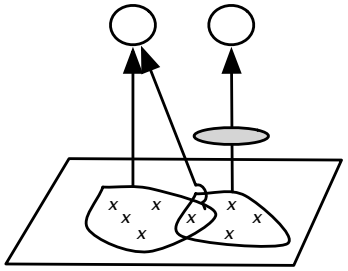
Unanticipated Partial Behavioral Reflection



- Geppetto: **Unanticipated** Partial Behavioral Reflection
  - For Squeak 3.9 with Bytecode transformation

David Röthlisberger, Marcus Denker and Éric Tanter:
Unanticipated Partial Behavioral Reflection:
Adapting Applications at Runtime
Journal of Computer Languages, Systems and Structures, vol. 34, no. 2-3, July 2008, pp. 46-65.

Good Results



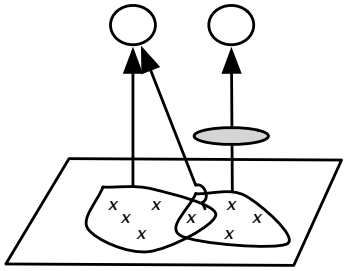
- Completely dynamic
- Simpler
- High performance

Benchmarks Geppetto

> Slowdown for reification of message send

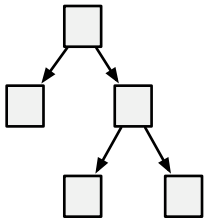
System	Slowdown
Geppetto	10.85
Iguana/J	24
Metaclasstalk	20

Missing Sub-method Structure

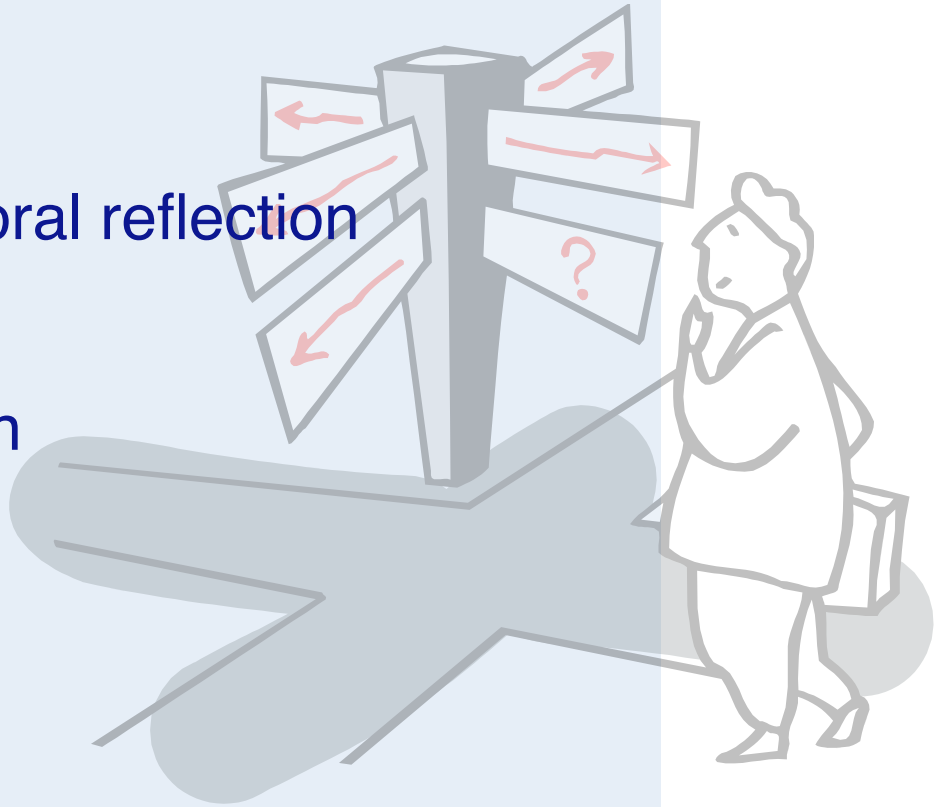


- Semantic Mismatch
- Code Quality
- Synthesized Code

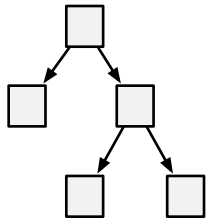
Roadmap



1. Dynamic partial behavioral reflection
2. **Sub-Method Structure**
3. Revisit Partial Reflection
4. Context

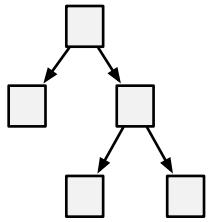


Methods and Reflection



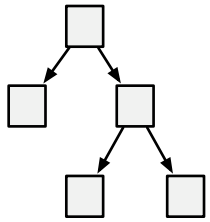
- Method are Objects
  - e.g in Smalltalk
- No high-level model for sub-method elements
  - Message sends
  - Assignments
  - Variable access
- Structural reflection stops at the granularity of methods

Sub-Method Reflection



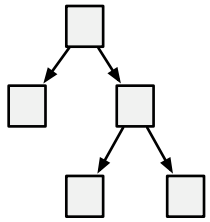
- Many tools work on sub method level
  - Profiler, Refactoring Tool, Debugger, Type Checker
- Communication between tools needed
  - Example: Code coverage
- All tools use different representations
  - Tools are harder to build
  - Communication not possible

Sub-method Representation Requirements



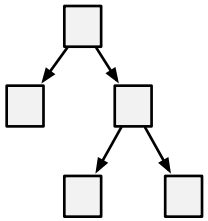
- Causal Connection
- Abstraction Level
- Extensibility
- Persistent
- Size and Performance

Existing Method Representations



- Existing representations for Methods
  - Text
  - Bytecode
  - AST

Text



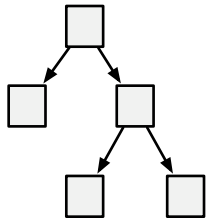
- Low level abstraction
- Not causally connected

004-StringZippedAndTest-md.1.cs

```
'From Squeak3.9 of 7 November 2006 [latest update: #7067] on 5 February 2007 at 3:25:56 pm'!
"Change Set:        StringZippedAndTest-md
Date:           5 February 2007
Author:         Marcus Denker

String has #unipped, but no #zipped. This
cs addes String>>zipped (originally from Diego Gomez Deck) and a test for unzipped/zipped.

This changeset is neutral to the question of zipped/unzipped  beeing in String, but if there is
#unzipped, there should be #zipped. And there should be a test.
"!


!String methodsFor: 'converting' stamp: 'dgd 11/26/2005 21:19'!
zipped
    | stream gzstream |

    stream := RWBinaryOrTextStream on: String new.

    gzstream := GZipWriteStream on: stream.
    gzstream nextPutAll: self.
    gzstream close.
    stream reset.

    ^ stream contents.
! !


!StringTest methodsFor: 'tests - converting' stamp: 'md 2/5/2007 15:21'!
testZipped
    | compressed |

    compressed := 'hello' zipped.
    self assert: (compressed unzipped = 'hello').! !
```

Line: 1 Column: 1 Plain Text Tab Size: 4

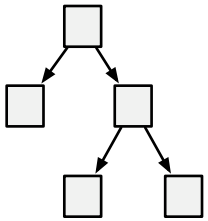
Bytecode



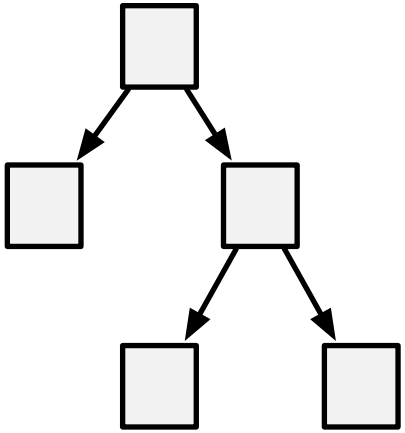
- Low level abstraction
- Missing extensibility
- Mix of base- and meta-level code

```
1 16 0 0 160 197 59
17 172 94 7 17 204
122 70 17 92 94 7 17
65 112 224 135 120 88
141 0 252
```

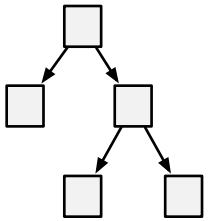
Abstract Syntax Tree



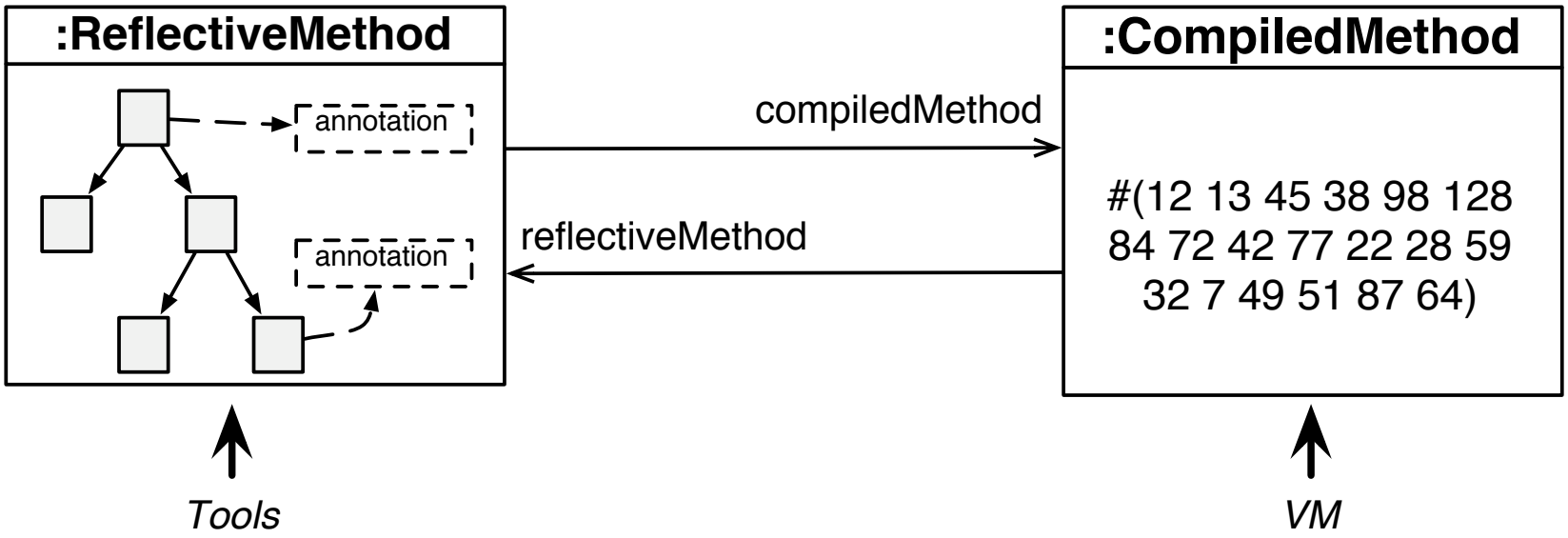
- Not causally connected
- Not extensible
- Not persistent



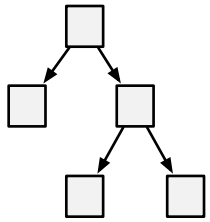
Solution: Reflective Methods



- Annotated, persistent AST
- Bytecode generated on demand and cached



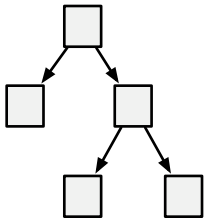
Implementation: Persephone



- Implementation of Reflective Methods for Squeak
- Smalltalk compiler generates *Reflective Methods*
  - Translated to bytecode on demand
- Open Compiler: Plugins
  - Called before code generation
  - Transform a copy of the AST

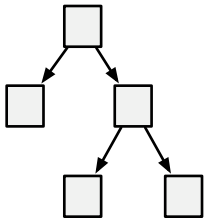
Marcus Denker, Stéphane Ducasse, Adrian Lienhard
Philippe Marschall: **Sub-Method Reflection**
Journal of Object Technology, vol. 6, no. 9,

Requirements revisited



- Abstraction Level ✓
- Causal Connection ✓
- Extensibility ✓
- Persistency ✓
- Size and Performance ✓

Extensible with Annotations



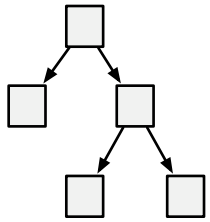
- Source visible annotations
  - extended Smalltalk syntax

```
(9 raisedTo: 10000) <:evaluateAtCompiletime:>
```

- Source invisible annotations
  - Reflective API
  - Can reference any object

- Every node can be annotated
- Semantics: Compiler Plugins

Example: Pluggable Type-System



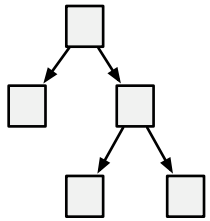
- Example for textual annotations

```
bitFromBoolean: aBoolean <:type: Boolean :>
^ (aBoolean ifTrue: [1] ifFalse: [0]) <:type: Integer :>
```

- Optional, pluggable type-system
- Types stored as annotations in the Reflective Methods

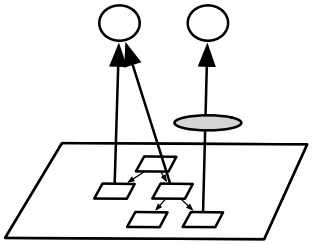
Niklaus Haldiman, Marcus Denker, Oscar Nierstrasz: “Practical, Pluggable Types,” (ICDL 2007)

Memory Requirements

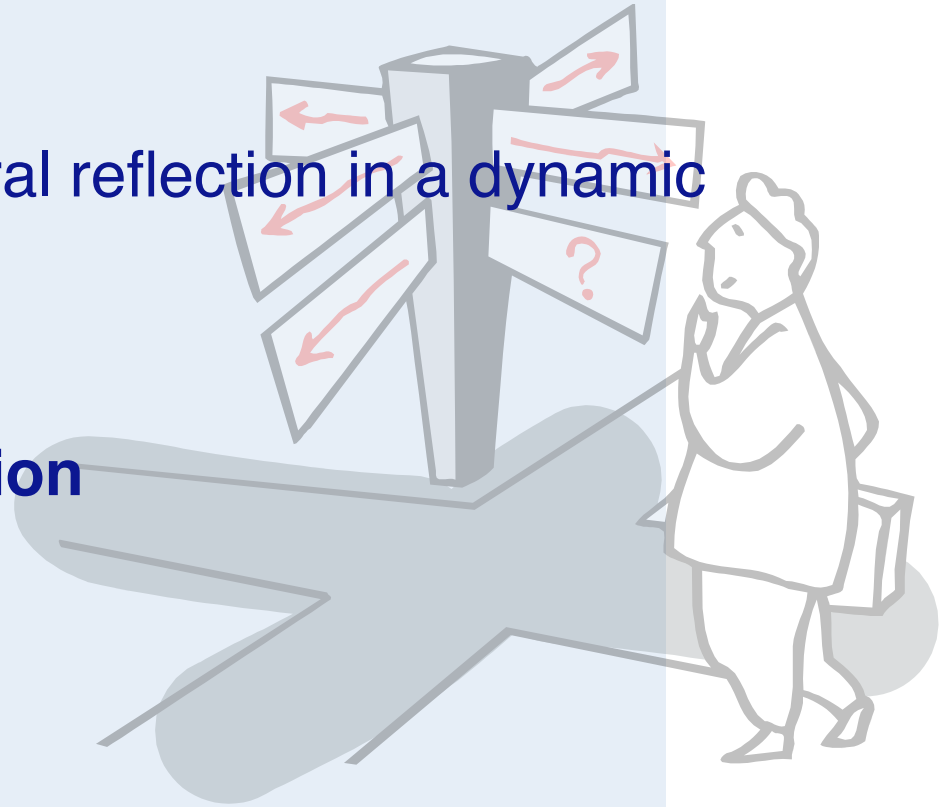


	number of classes	*memory*
Squeak 3.9	*2040*	*15.7 MB*
Persephone *no reflective methods*	*2224*	*20 MB*
Persephone *reflective methods*	*2224*	*123 MB*

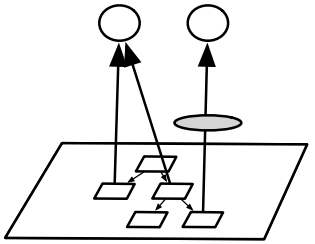
Roadmap



1. Realize partial behavioral reflection in a dynamic language
2. Sub-Method Structure
3. **Revisit Partial Reflection**
4. Context

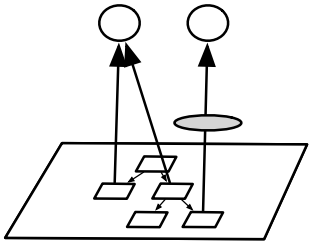


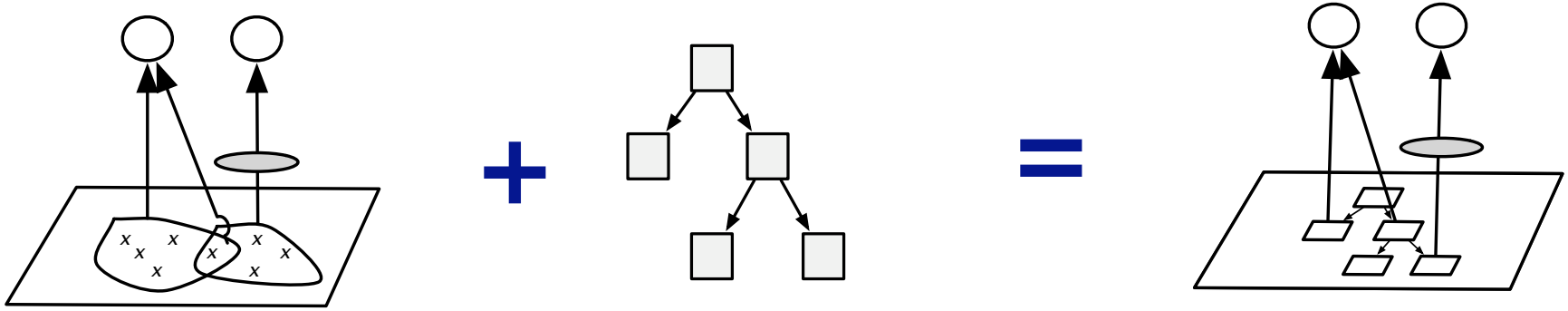
Partial Behavioral Reflection Revisited



- Problems of Bytecode:
  - Semantic Mismatch
  - Code Quality
  - Synthesized Code
- With Sub-method Reflection, we can do better!

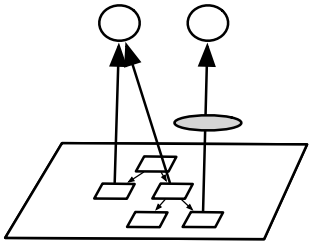
Sub-method Structure





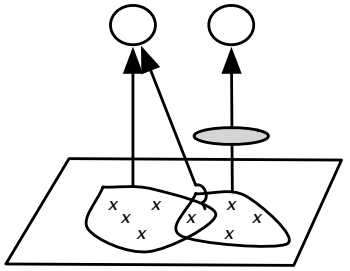
- Links can be annotations on the AST

Performance Properties



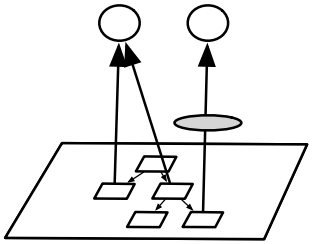
- Very fast annotations
  - No decompile!
- On-the-fly code generation
  - Only code executed gets generated
- Generated code is fast
  - Better then working on bytecode level

Repeat: Missing Sub-method Structure



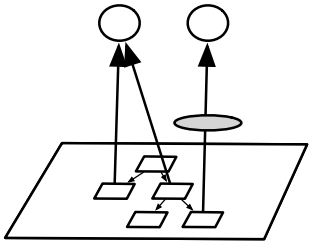
- Semantic Mismatch
- Code Quality
- Synthesized Code

Sub-method Structure

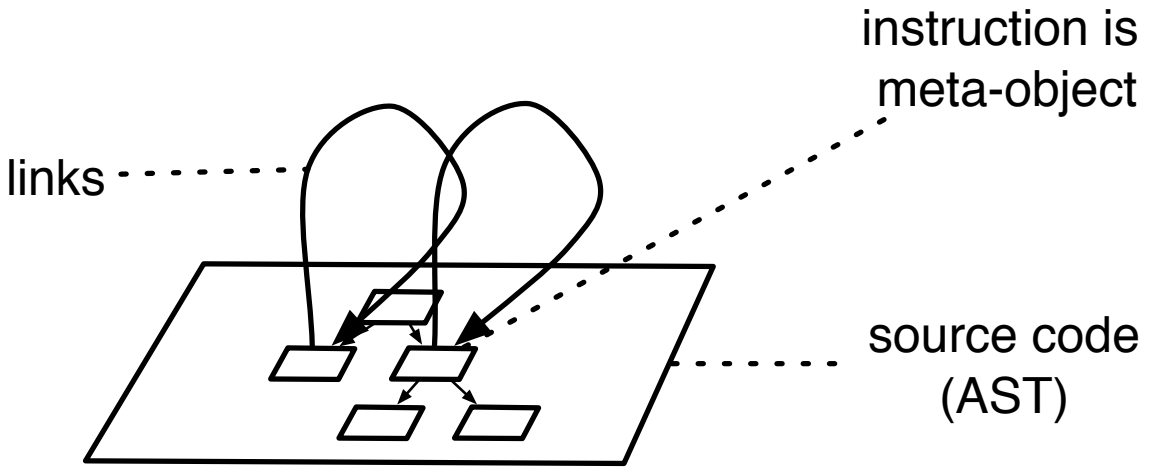


- Semantic Mismatch ✓
- Code Quality ✓
- Synthesized Code ✓

Example: Feature Annotations

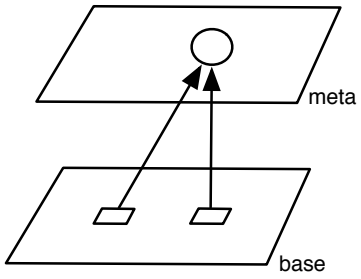


- Features modeled as traces
- Many Problems
  - Space
  - Merging Traces
- Solution: annotate structure

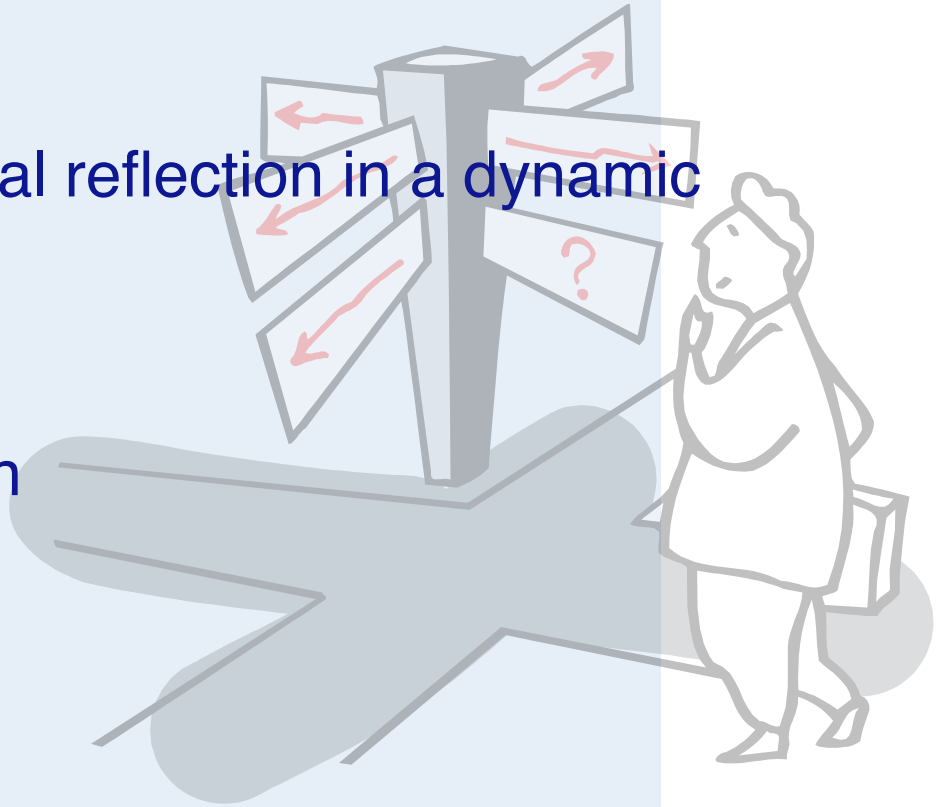


Marcus Denker, Orla Greevy, Oscar Nierstrasz: **Supporting Feature Analysis with Runtime Annotations** (PCODA 2007)

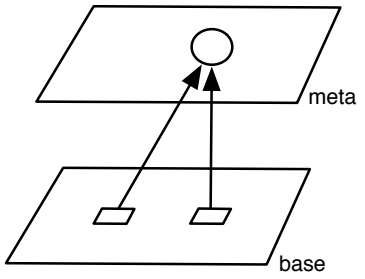
Roadmap



1. Realize partial behavioral reflection in a dynamic language
2. Sub-Method Structure
3. Revisit Partial Reflection
4. **Context**

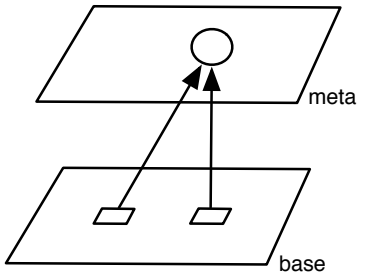


Problem: Whole System



- Behavioral reflection cannot be applied to the whole system
- Problem: recursion
  - System classes
  - Meta-objects

The Problem of Recursion

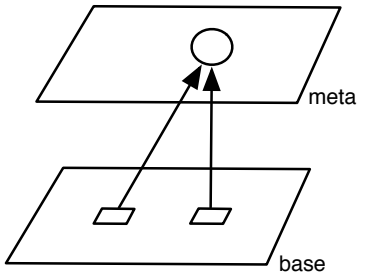


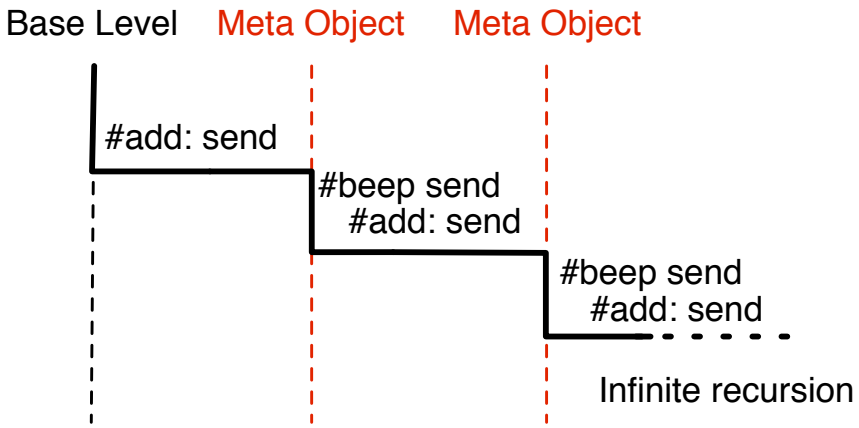
- Call the Beeper from OrderedCollection>>#add

```
beepLink := Link new metaObject: Beeper.
beepLink selector: #beep.
```

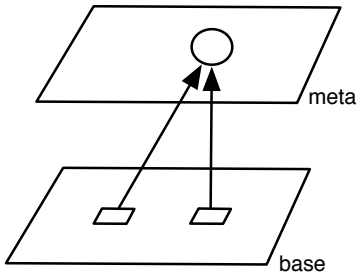
```
(OrderedCollection>>#add:) methodNode link: beepLink.
```

Meta-object Call Recursion

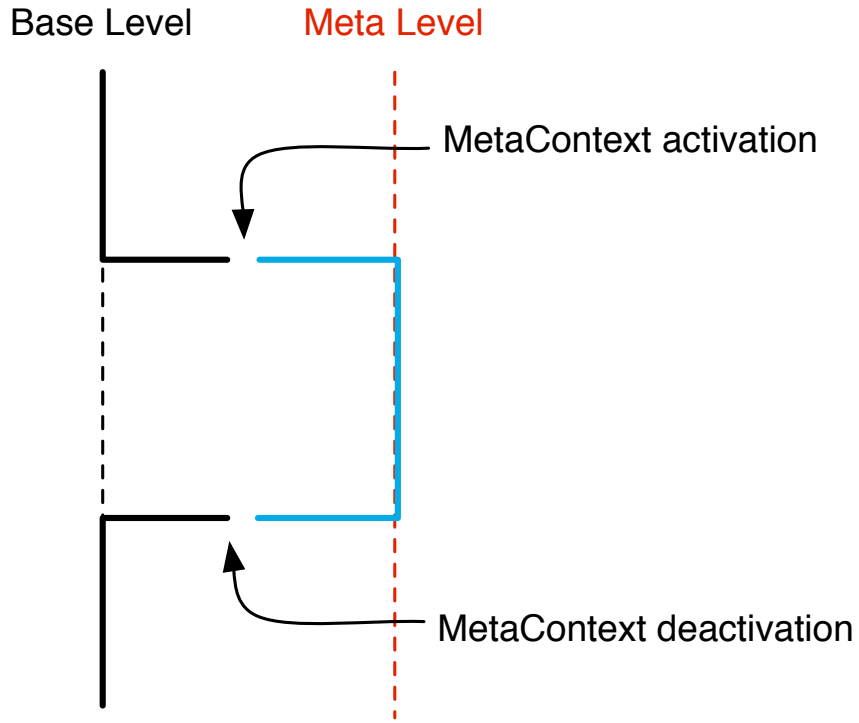




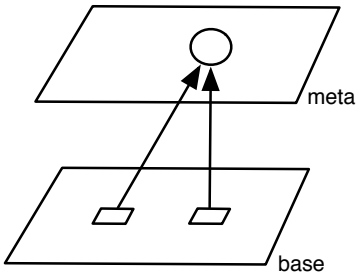
Representing Meta-level Execution



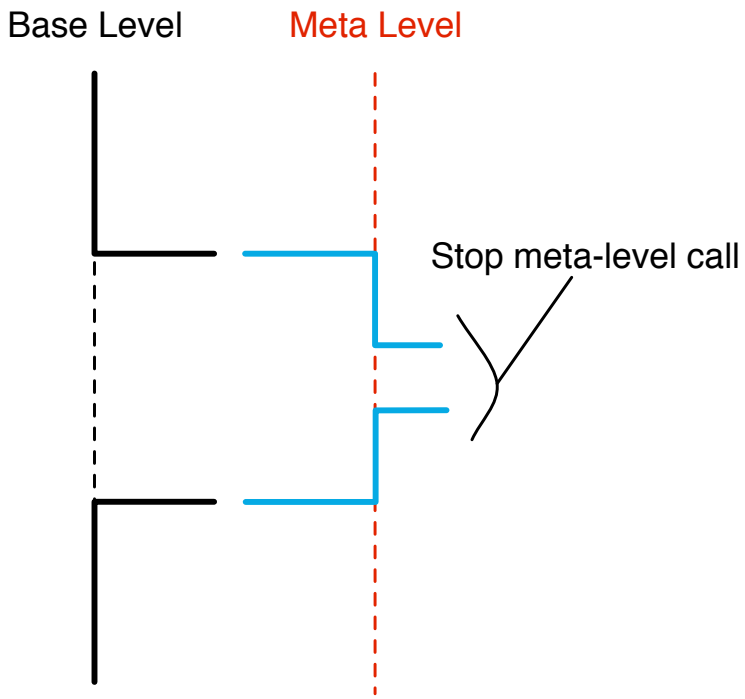
> Link enables **MetaContext**



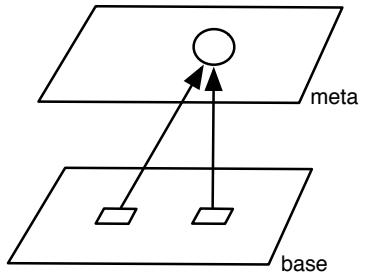
Context-aware Links



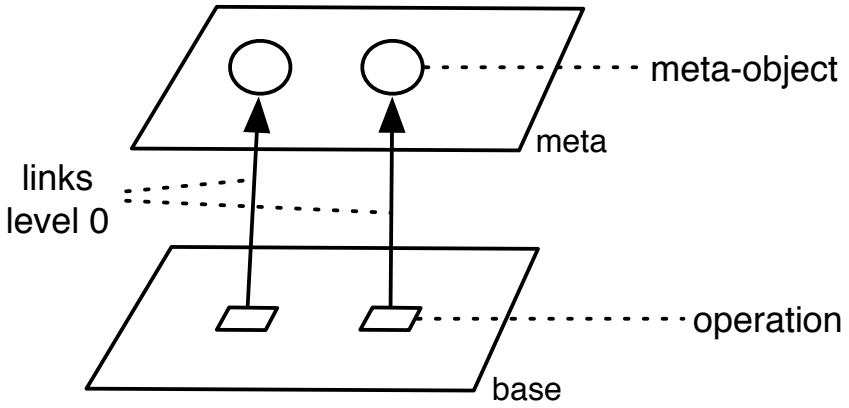
- Disable call when already on the meta-level



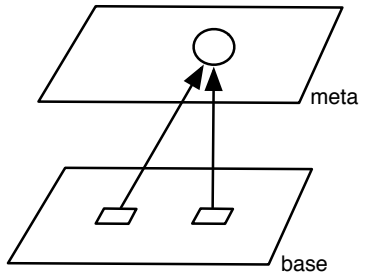
MetaContext: Conclusion



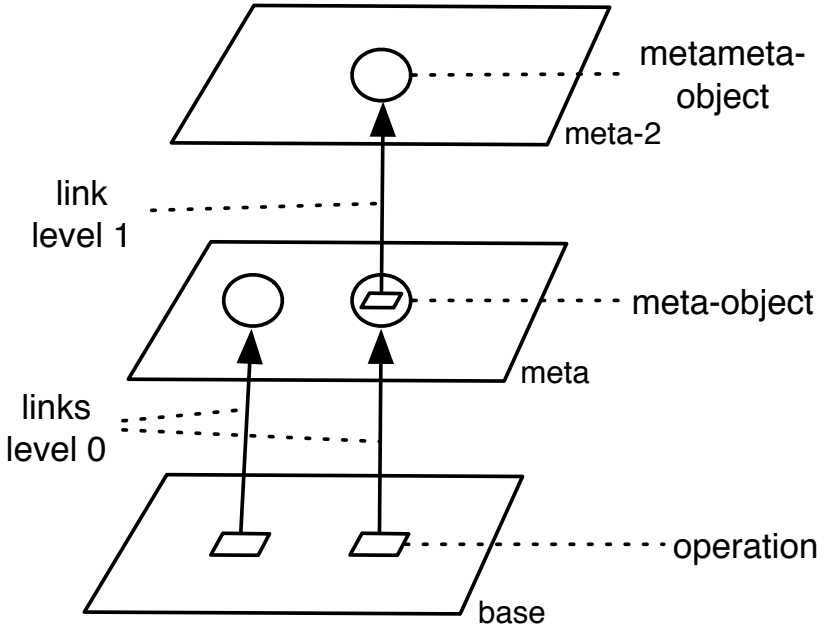
> Recursion problem solved



MetaContext: Conclusion



> Meta-level Analysis



Marcus Denker, Mathieu Suen,Stéphane Ducasse:
The Meta in Meta-object Architectures
TOOLS EUROPE 2008

Thesis Revisited

To support **unanticipated** behavioral reflection, reflection needs to be extended with **sub-method structure** and with the concept of **context**.

Future Work

- Sub-method Structure
  - Simpler AST
  - AST compression
  - Replace text with sub-method representation
- Behavioral Reflection
  - Composition of Links
  - Generalize context model: beyond the MetaContext

Contributions of the Dissertation

- *Unanticipated* partial behavioral reflection
- Sub-Method Structural Reflection
- Partial Behavioral Reflection *using Annotations*
- *Contextual Reflection*

Questions

- *Unanticipated* partial behavioral reflection
- Sub-Method Structural Reflection
- Partial Behavioral Reflection *using Annotations*
- *Contextual Reflection*

Questions?