UNIVERSITÄT
BERN

Unanticipated Partial Behavioral Reflection

David Röthlisberger
Marcus Denker
Eric Tanter

Roadmap

- Example
- Behavioral Reflection
  - Smalltalk / Metaclasstalk
  - Unanticipated Reflection
  - Partial Reflection
- Unanticipated Partial Behavioral Reflection
- Example revisited
- Benchmarks
- Conclusion

Running Example

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

Towards a Solution

- Analyse the problem
  - Install profiler
  - Analyze
  - Retract profiler
- Solve the problem
  - Introduce a caching mechanism
  - Experiment with multiple solutions

Reflection

- Reflection: computation about computation
  - base level / meta level
  - causally connected
- Structural Reflection
  - Reification of structure
- Behavioral Reflection
  - Reification of execution

Unanticipated Reflection

> “Unanticipated Use” means:
  - No need to know in advance (and plan for) reflection
  - Possible to be done at runtime

> Java: reflection only at load time
  - No runtime change in general
  - Need to compile in hooks for reflection

> Smalltalk reflection is unanticipated
  - taken for granted. But it’s quite cool!

Reflection in Smalltalk

> Structural Reflection
  - Classes / Methods are objects
  - Can be changed at runtime

> Behavioral Reflection
  - No model of execution below method body
  - message sending / variable access hard coded by VM
  - #doesNotUnderstand / MethodWrappers

> Reflective capabilities of Smalltalk should be improved!

MetaclassTalk

- Extends the Smalltalk metaclass model

- Metaclass defines
  - message lookup
  - access to instance variables

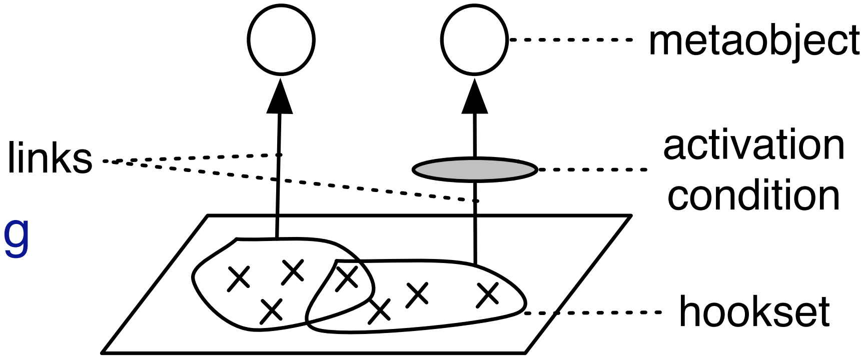
- Problems:
  - Reflection only controllable at class boundaries
  - No fine-grained selection (e.g. single operations)
  - Protocol between base and meta level is fixed

Partial Reflection

UNIVERSITÄT BERN

- Hooksets: collection of operation occurrences
- Links
  - Bind hooksets to metaobjects
  - Define Protocol between base and meta

- Goals
  - Highly selective reification
  - Flexiblel metalevel engineering
    - *Protocol specification*
    - *Cross-cutting hooksets*



Geppetto

> Partial Behavioral Reflection pioneered in Java
  - Code transformation at load time
  - Not unanticipated (it's Java...)

> Geppetto: Partial Behavioral Reflection for Smalltalk

> For Squeak 3.9 with ByteSurgeon
  - but portable to other dialects

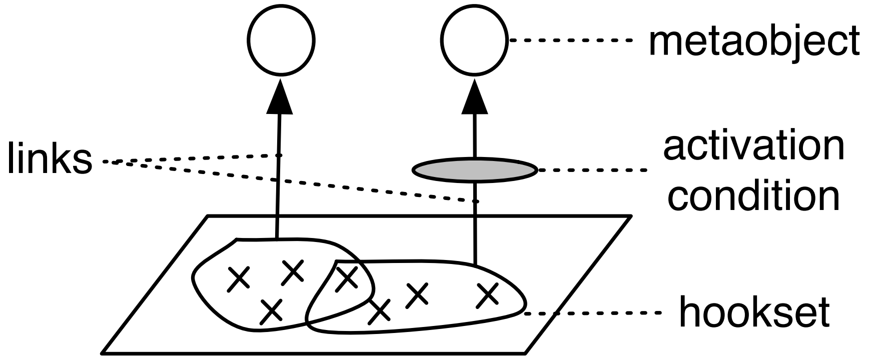
> Let's see an example!

Solving the Problem

> Operation:
  - Method Execution (around)

> Hookset:
  - All execution operations in the wiki package

> Metaobject:
  - A profiling tool



Solving the Problem: Hookset + Link

> Hookset

```
allExecs := Hookset new.
allExecs inPackage: ’Wiki’; operation: MethodEval.
```

> Link

```
profile := Link id: #profiler
                hookset: allExecs
                metaobject: Profiler new.
profile control: Control around.
```

Solving the Problem: Protocol

- Protocol

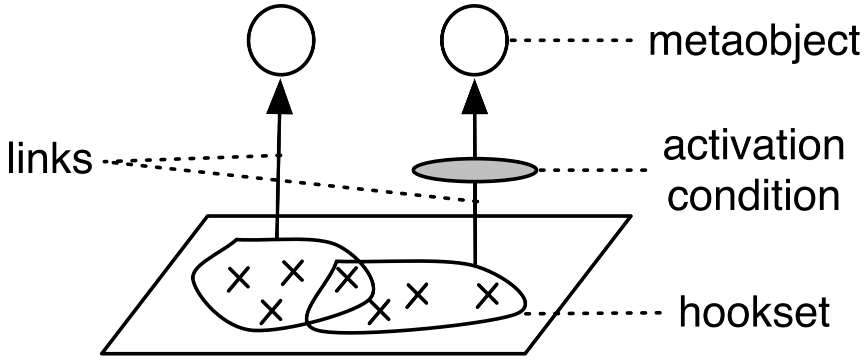
```
profile callDescriptor:
   (CallDescriptor
     selector: #profileMethod:in:withArguments:
     parameters: {Parameter selector.
                  Parameter self.
                  Parameter arguments.}
     passingMode: PassingMode plain).
```

- Install / Retract

```
profile install.
profile uninstall.
```

Solving the Problem: Caching

- Operation:
  - Method Execution (around)
- Hookset:
  - The one slow method (#toughWorks:)
- Metaobject:
  - A Cache



Caching II

- Hookset:

```
toughWorks := Hookset new.
toughWorks inClass: Worker; inMethod: #toughWork:;
          operation: MethodEval.
```

- Link:

```
cache := Link id: #cache
           hookset: toughWorks
           metaobject: Cache new.
cache control: Control around.

cache callDescriptor:(CallDescriptor
    selector: #cacheFor:
    parameters: {Parameter arg1}
    passingMode: PassingMode plain).
```

Geppetto

- Operations:
  - MethodEval (like MethodWrappers)
  - MsgSend, InstVarAccess, TempAccess
- Control
  - Before, After, Around, Replace
- Activation condition per Link

Benchmarks I

- Slowdown for reification of message send

System	Slowdown
Geppetto	10.85
Iguana/J	24
Metaclasstalk	20

Benchmarks II

> Geppetto Vs. Metaclasstalk

	Metaclasstalk (ms)	Geppetto (ms)	Speedup
message send	108	46	2.3x
instance var read	272	92	2.9x

Future Work

- Pluggable backends
  - Bytecode
  - AST based transformation
  - VM Support
- Use for typical ByteSurgeon based projects
  - e.g. Tracing, Unstuck-Debugger
- Experiment with advanced Scoping

Conclusion

- Example for the need of Partial Behavioral Reflection
- Overview of Geppetto
- Solution of the Example
- Benchmarks

Conclusion

- Example for the need of Partial Behavioral Reflection
- Overview of Geppetto
- Solution of the Example
- Benchmarks

Questions?