Bringing Reflection to Life with First-Class Interpreters

Toon Verwaest
Camillo Bruni
David Gurtner
Adrian Lienhard
Oscar Nierstrasz

Software Composition Group
University of Bern
Switzerland
debugging is hard
developing debuggers is even harder
System.out.println
Object-Flow Debugger

Lienhard '07
modifications to the Virtual Machine
A programming language is a notational system for describing computation in a machine-readable and human-readable form.

— Louden
What if we could build a specialized debugger in just a few hours?
modify the interpretation in the language itself in terms of the source code
<table>
<thead>
<tr>
<th>Interpreter</th>
</tr>
</thead>
<tbody>
<tr>
<td>environment</td>
</tr>
<tr>
<td>interpret:</td>
</tr>
<tr>
<td>send:to:class:</td>
</tr>
<tr>
<td>visitSend:</td>
</tr>
<tr>
<td>visit...</td>
</tr>
</tbody>
</table>
**PINOCCHIO**

```
Interpreter
environment
interpret:
send:to:class:
visitSend:
visit...

Stepping
stepBlock
send:to:class:
defaultStepBlock

Debugger
defaultStepBlock
```
Debugger interpret: [ Person new ]
Debugger interpret: [ Person new ]

structural reflection
continuous
behavioral
reflection
send: message to: receiver: class: class

self print:
  receiver class name, '>>', message.

^ self debugShellWithAction:[
  super
    send: message
    to: receiver
    class: class ]
### Interpreter

- environment
- interpret:
  - send:to:object:
  - visitSend:
  - visit...

### Stepping

- stepBlock

### Debugger

- defaultStepBlock

---

recursive AST visitors

garbage collection

object model
Alias Interpreter

person := Person new

person name: 'John'

person name: 'Doe'

AliasInterpreter

interpret: [
  p := Person new.
  p name: 'John'.
  p name: 'Doe'.
]

origin

predecessor
interpretMethod: method
 | result | result := super interpretMethod: method.
 ^ (ReturnAlias alias: result)
   environment: environment
Performance (fib)

- Pinocchio: 2x slower than Pharo
- 2x slower than Ruby 1.9
- 2x faster than Python 2.6.4
- 5x faster than Ruby 1.8

- Metacircular: 160x slower than Pinocchio
Performance \((fib)\)

**Pinocchio**  
- 2x slower than Pharo  
- 2x slower than Ruby 1.9  
- 2x faster than Python 2.6.4  
- 5x faster than Ruby 1.8

**Metacircular**  
- 160x slower than Pinocchio

**Java**  
- 160x faster than Ruby 1.8
• recursive AST visitors
• extensible using OO techniques
• implemented practical debuggers

Future work
• performance is not addressed yet