Marcus Denker

Universität Bern
Roadmap

> Research

> Teaching and Advising Students

> Open Source Involvement

> Future
Roadmap

> Research

> Teaching and Advising Students

> Open Source Involvement

> Future
The Systems of the Future...

> ... are getting larger and more complex

> ... are getting more and more dependent on each other

> ... thus are working in a changing environment
Three Research Directions

> Reflective System

> Context Oriented Programming

> Virtual Machines
1. Reflective Systems

Query and Change
Sub-Method 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
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
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,
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 for a Dynamic Language,” (Journal of Computer Languages, Systems and Structures, 2009)
Example: Partial Behavioral Reflection

> Joined Project with University of Chile

> Realized Reflex for a dynamic languages

> Support unanticipated use.

Marcus Denker, Stéphane Ducasse and Éric Tanter: “Runtime Bytecode Transformation for Smalltalk” (Journal of Computer Languages, Systems and Structures, 2006)

Using Sub-method Structure

> Realized Reflex model on top of sub-method reflection

> Very good performance
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)
2. Context Oriented Programming

> Software needs to work in changing environments

> Code to deal with context specific behavior spread all over the system

> No representation of context

Martin von Löwis, Marcus Denker and Oscar Nierstrasz: *Context-Oriented Programming: Beyond Layers* (ICDL 2007)
Context-Aware Aspects

- Aspects: modularizing crosscutting concerns
- Context code crosscuts application code

Éric Tanter, Kris Gybels, Marcus Denker and Alexandre Bergel: “Context-Aware Aspects” (Software Composition (SC 2006), LNCS, vol. 4089)
ChangeBoxes

> Programming Languages do not model change

> Changeboxes:
  — All changes to code are recorded
  — Execution of old versions

Marcus Denker, Tudor Gîrba, Adrian Lienhard, Oscar Nierstrasz, Lukas Renggli and Pascal Zumkehr:
“Encapsulating and Exploiting Change with Changeboxes”
(ICDL 2007)
Problem: applying behavioral reflection on system classes (Numbers, Array…).

Reason: meta-object call recursion

Solution: meta-execution context

3. Virtual Machines

- Late-bind compilation for dynamic languages
- Provide virtual, easy to change “hardware”
Squeak J3

> Just-In-Time Compiler for Squeak Smalltalk

> Ported from PowerPC to Intel
   — 2000/2001 while at Disney

Marcus Denker:
Entwurf von Optimierungen für Squeak
(Studienarbeit, University of Karlsruhe, 2002)
Virtual Machines are very complex
- Especially dynamic Optimizers

Idea: Move optimization out of the VM

My Master’s Thesis, 2004
- Prototype for Squeak.

Interpreter Inline Caches

Bring some ideas of high-performance JIT Compilers back to simple Interpreters

Interesting for Bytecode-to-Bytecode optimization

Michael Haupt, Robert Hirschfeld and Marcus Denker: 
Type feedback for bytecode interpreters
(ICOOOLPS'2007)
Roadmap

- Research
- Teaching and Advising Students
- Open Source Involvement
- Future
Students

> 5 Master Projects, 1 Bachelor co-advised
  ─ Students contributed to research
  ─ 4 published as papers at conferences
  ─ 2 later extended to Journals

> Google Summer of Code
Courses and Lectures

> Assistant
  — Smalltalk (2004). (Prof. Stephane Ducasse)

> Lectures given as part of courses:
  — HPI Germany
  — Bern
  — University of Chile

> Current: Compiler Construction, University Bern
  — Two Lectures on SSA and Optimizations
Roadmap

> Research

> Teaching and Advising Students

> Open Source Involvement

> Future
Squeak

> Open Source Smalltalk System
  — Alan Kay, Dan Ingalls

> Responsible for Version 3.9

> Co-Founder Squeak Foundation
  — Board Member till 2007

60,000 Downloads
eToys

> Programming for Children

> On the OLPC ($100 Laptop)

> Based on Squeak

— Worked with Alan Kay’s group on Squeak
— Founder, Squeak Germany e.V.
— Translated Book “Powerful Ideas in the Classroom”
Roadmap

> Research

> Teaching and Advising Students

> Open Source Involvement

> Future
Future Research

- Context and Reflection
- Bio-Inspired Reflective Languages
- Compiler for Dynamic Languages
Context and Reflection

- Reflection is powerful, but dangerous.
- How can we control reflective structural change?
- Continue work started with ChangeBoxes and MetaContext
Runtime Compilers

> Back to Bytecode-to-Bytecode Optimization

> SIStA: Son-of-AOSTA
  — revisit work done on AOSTA

> Research: Integrate with Reflection Framework
Complex, ever-evolving, dynamic Systems?  
— They are everywhere!

Idea: <CENSORED> as a concept for reflective languages
Future Teaching

> Continue with advising Master and Bachelor Students

> Possible Courses at University of Chile:
  – Programming Languages
  – Software Engineering
  – Compiler Construction
  – Virtual Machines
Future Open Source

> Pharo: Squeak, rebooted
  – Professional Development
  – Stable basis for Research
  – INRIA, France

> Squeak by Example, OLPC Edition
  – SBE Focusing on Squeak for OLPC
  – Programming beyond eToys
Questions?