Call Graphs for Live Programming

Implementing Call Tracing in Babylonian/S based on a Survey of Property Extraction Techniques for Dynamic Analysis

Christian Flach

Software Architechture Group
Hasso Plattner Institute, University of Potsdam

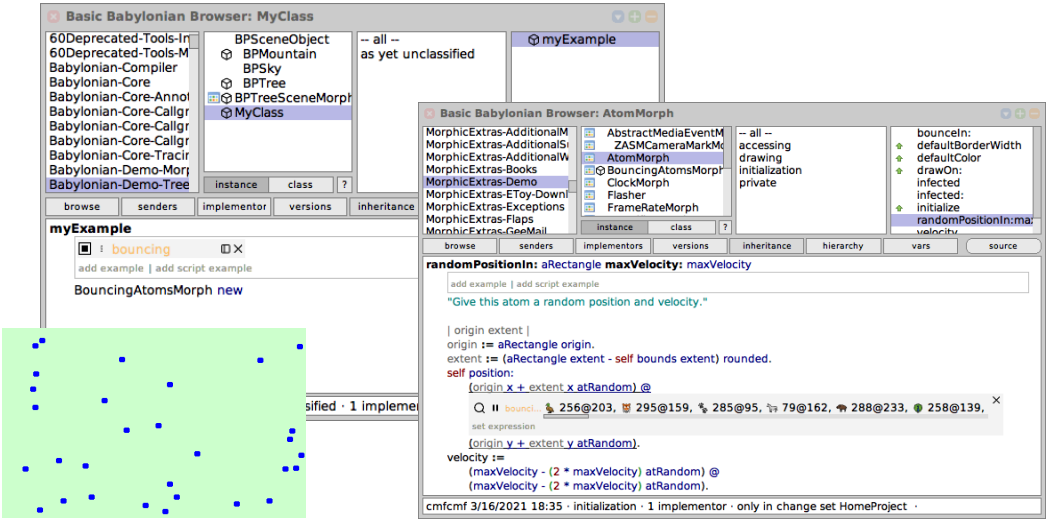
—

HPI Research Symposium 2021

April 19, 2021

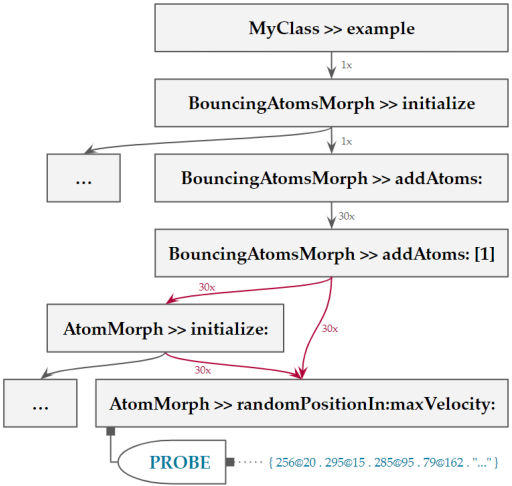
Live Programming with Squeak/Smalltalk

Microview



Selection of Possible Use Cases

Macroview



Possible Use Cases

As part of an example, . . .

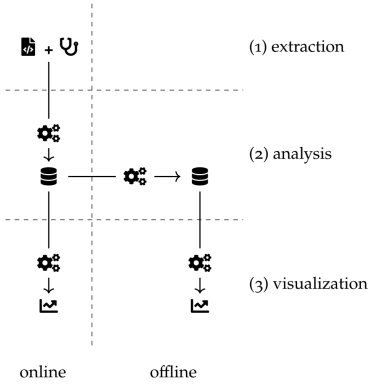
- given two or more procedure/probe executions, how do they relate to each other?
- given a procedure execution, what other procedures/probes are called from it?
- which procedures/probes are executed?

Dynamic Analysis

"the analysis of the **properties** of a **running** program"[1]
(emphasis mine)

Process[2][3]

1. Property Extraction
2. Property Analysis
3. Property Visualization



[1] Thomas Ball. "The Concept of Dynamic Analysis". In: *Software Engineering - ESEC/FSE'99, 7th European Software Engineering Conference, Held Jointly with the 7th ACM SIGSOFT Symposium on the Foundations of Software Engineering, Toulouse, France, September 1999, Proceedings*. Ed. by Oscar Nierstrasz and Michel Lemoine. Vol. 1687. Lecture Notes in Computer Science. Springer, 1999, pp. 216–234. DOI: 10.1007/3-540-48166-4_14.

[2] Mireille Ducassé and Jacques Noyé. "Logic Programming Environments: Dynamic Program Analysis and Debugging". In: *The Journal of Logic Programming* 19/20 (1994), pp. 351–384. DOI: 10.1016/0743-1066(94)90030-2.

[3] Margaret-Anne D. Storey. "Theories, Tools and Research Methods in Program Comprehension: Past, Present and Future". In: *Software Quality Journal* 14.3 (2006), pp. 187–208. DOI: 10.1007/s11219-006-9216-4.

Approach

- Exploratory search for "dynamic analysis" and related topics ("debugger", "malware analysis", "program comprehension", etc.), including
- Well-known publications for Squeak/Smalltalk
- *Workshop on Dynamic Analysis* (WODA, 2003 to 2017)

Corpus

- ~40 surveys/survey-like publications, ~20 tool frameworks, ~60 tools, ~90 other publications, ~20 Smalltalk papers
- 13 publications that explicitly list more than two Property Extraction techniques

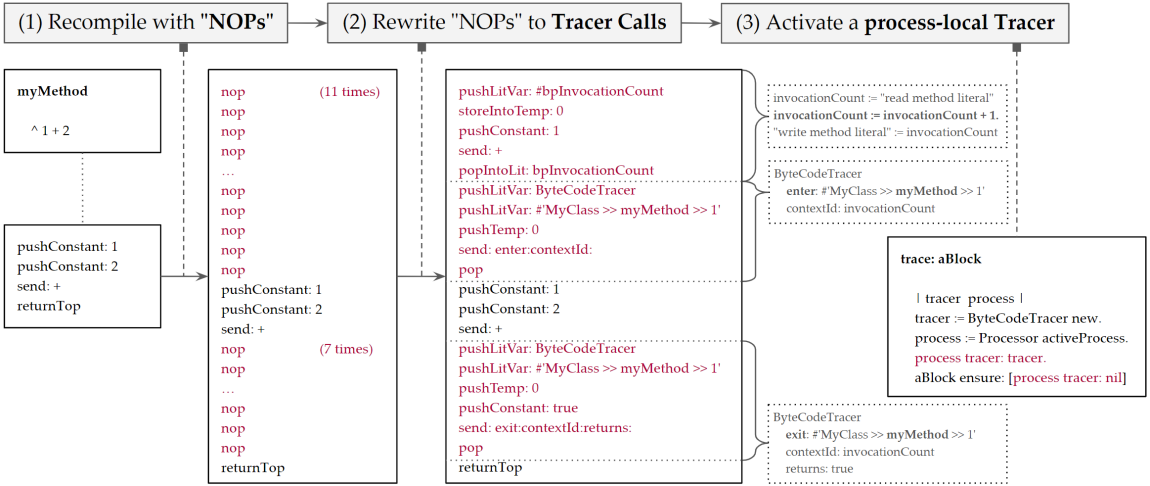
Survey on Property Extraction Techniques for Dynamic Analysis

Hierarchy of Property Extraction Implementation Techniques

1. **Program Level**
   - 1.1 Changing Input, Observing Output
   - 1.2 Source Code Rewriting
   - 1.3 Bytecode/Machine Code Rewriting
2. **Runtime/Interpreter Level**
   - 2.1 Analysis Interfaces Provided by the Runtime/Interpreter
   - 2.2 Runtime/Interpreter Modification
3. **Operating System Level**
4. **System/Processor Software-based**
   - 4.1 Processor Tracing
   - 4.2 Hardware Performance Counters
   - 4.3 Hardware Sensors
5. **Virtualization and Emulation**
   - 5.1 System Virtual Machine
   - 5.2 Processor Emulation
   - 5.3 System Emulation
6. **System/Processor Hardware-based**
   - 6.1 Hardware Processor Debugging
   - 6.2 System/Processor Hardware Instrumentation
7. **System External**

Call Tracing Implementation

Bytecode Rewriting — Process

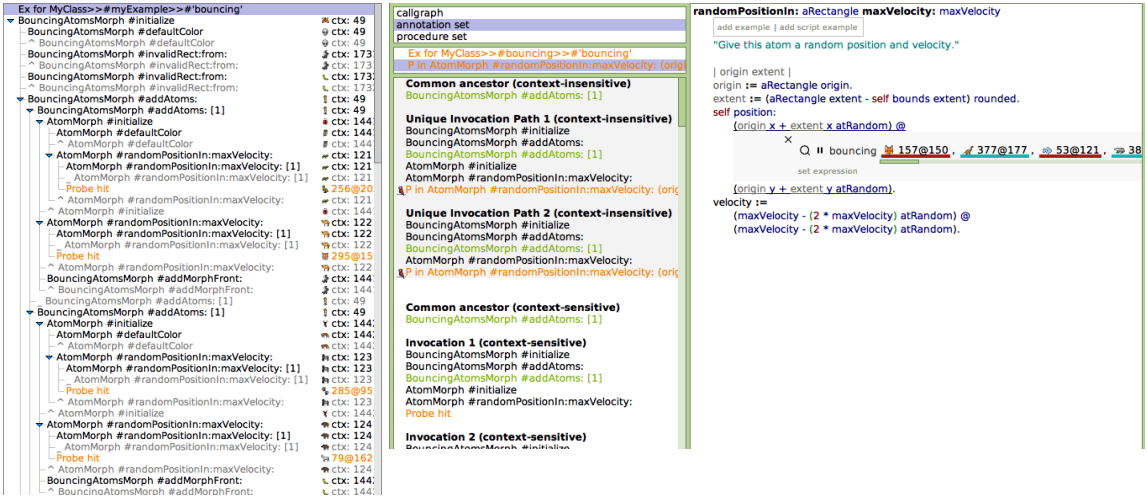


Benchmarks

(lower is better)

Squeak/Smalltalk (Babylonian/S) Integration

Macroview



Call Graphs for Live Programming

Implementing Call Tracing in Babylonian/S based on a Survey of Property Extraction Techniques for Dynamic Analysis

Exploratory Survey on **Property Extraction for Dynamic Analysis**

~200 publications

Hierarchy of Property Extraction Implementation Techniques

1. Program Level
2. Runtime/Interpreter Level
3. *Operating System Level*
4. *System/Processor Software-based*
5. *Virtualization and Emulation*
6. *System/Processor Hardware-based*
7. *System External*

Call Tracing **Implementations** for Squeak/Smalltalk

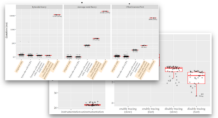
Bytecode Rewriting

Simulation

***Shared* Call Graph Generation Implementation**
(Call Trace → Call Graph)

```
methodA enter
methodB enter
…
methodB exit
methodA exit
…
```

Evaluation and Benchmarks



Squeak/Smalltalk (Babylonian/S) Integration