CWI Centrum Wiskunde & Informatica

Rascal: A One-Stop-Shop for Program Analysis and Transformation

Mark Hills

SET Seminar
November 20, 2012
Eindhoven, The Netherlands

http://www.rascal-mpl.org

Rascal: A Meta-Programming One-Stop-Shop

- Context: wide variety of programming languages (including dialects) and meta-programming tasks

- Typical solution: many different tools, lots of glue code

- Instead, we want this to all be in one language, i.e., the “one-stop-shop”

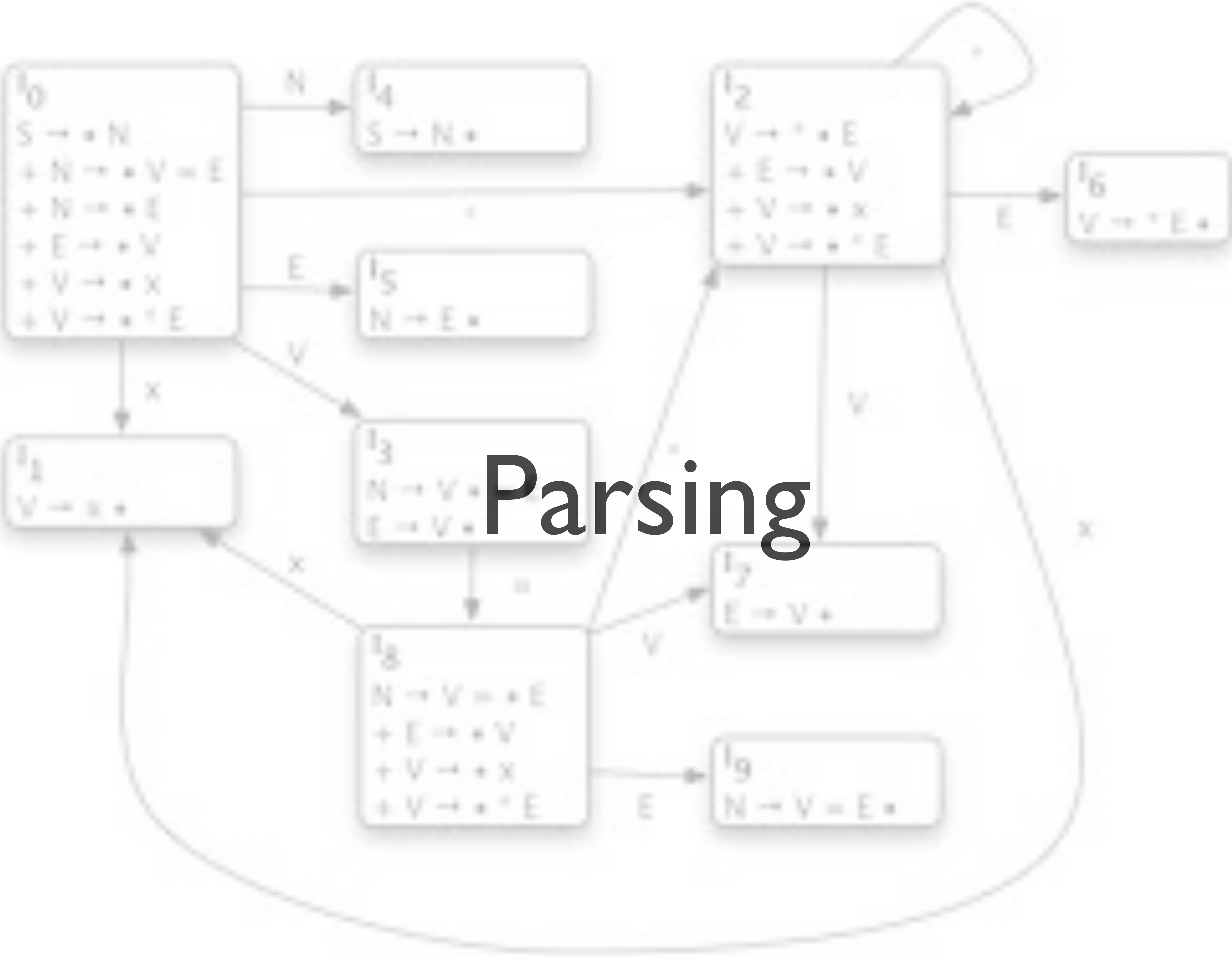
- Rascal: domain specific language for program analysis, program transformation, DSL creation

Picture from: http://www.mountainhighlands.com/listings/colabrese.html

Usage Scenarios

- Parsing (briefly!)
- DSLs
- Software Repository Mining
- Program Analysis
- Visualization
- Many others...

Parsing


I0
S → • N
+ N → • V = E
+ N → • E
+ E → • V
+ V → • x
+ V → • * E
I4
S → N •
I5
N → E •
I1
V → x •
I3
N → V • = E
E → V •
I2
V → * • E
+ E → • V
+ V → • x
+ V → • * E
I6
V → * E •
I7
E → V •
I8
N → V = • E
+ E → • V
+ V → • x
+ V → • * E
I9
N → V = E •
N
*
E
V
x
=

Parsing

- Rascal grammar definition language, GLL parsing
- Filtering rules written in Rascal provide disambiguation
    - Example: C’s famous T *b, need a symbol table
- Other features: implode to AST, track source locations
- Parsing integrates with IDE support: provides parse trees needed by IDE functionality, annotations on tree trigger IDE functionality

Domain-Specific Languages

The Addison-Wesley Signature Series
Domain-Specific Languages
A Martin Fowler Signature Book

Domain-Specific Languages

- DSLs support domain-level concepts, syntax familiar to practitioners
- Many familiar examples from tech space: SQL for database access, HTML for web pages, ATL for model transformations
- Some not so familiar: S3QL in Bioinformatics, Cg for graphics programming

Another Domain: Digital Forensics (Jeroen van den Bos)

- From Wikipedia: “Digital forensics is a branch of forensic science encompassing the recovery and investigation of material found in digital devices, often in relation to computer crime.”
- Challenges: need custom software, engineered to specific requirements (including for legal reasons), that performs well
- Research Question: can model-driven techniques be used to create fast, maintainable digital forensics software?

File carving

- File carving is the process of recovering files without the help of (file system) storage metadata.
- A file carver typically consists of two parts:
  - The carver itself, which selects and/or combines blocks of data from the input as candidate files.
  - A set of format validators that determine whether a candidate file is of any of the formats they validate.

Slide from Jeroen van den Bos

Describing File Formats in Derric

1. Header

Name and encoding/ type defaults

```
format PNG

strings ascii
size 1
unit byte
sign false
type integer
order lsb0
endian little
```

2. Sequence

Data structure ordering

```
sequence

Signature
IHDR
(ITXT ICMT)*
PLTE?
IDAT
IDAT*
IEND
```

3. Structures

Layout of individual data structures

```
structures

IHDR {
 l: lengthOf(d)
      size 4;
 n: “IHDR”;
 d: { ... }
 c: checksum
(...) size 4; }
```

Slide from Jeroen van den Bos

Describing File Formats in Derric

```
structures

Chunk {
  length: lengthOf(chunkdata) size 4;
  chunktype: type string size 4;
  chunkdata: size length;
  crc: checksum(algorithm="crc32-ieee",
                fields=chunktype+chunkdata) size 4;
  end: 0xFF3F;
}
IHDR = Chunk {
  chunktype: "IHDR";
  chunkdata: {
    width: !0 size 4;
    height: !0 size 4;
    bitdepth: 1|2|4|8|16;
    imagesize: (width*height*bitdepth)/8 size 4;
    interlace: 0|1;
  }
}
```

Slide from Jeroen van den Bos

Applying Derric

- Each format has one/several descriptions.
- Code generator uses descriptions:
    - Applies (domain-specific) optimizations/transformations.
    - Runs quickly, so easy to rerun after changes.
    - May output for multiple targets.

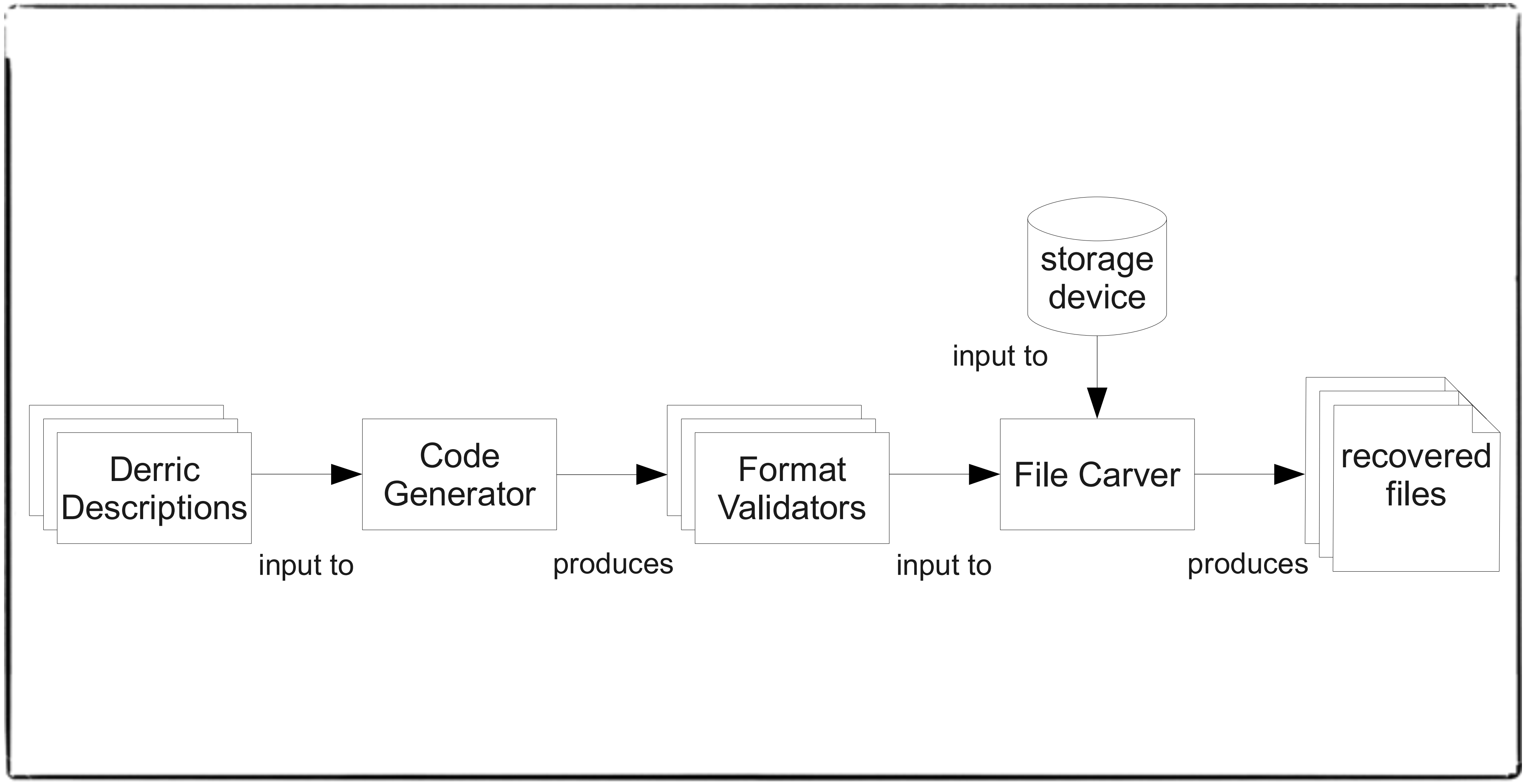
Slide from Jeroen van den Bos

Applying Derric

- Runtime system uses generated validators:
  - Recognizes, extracts or ignores files.
  - Implements algorithms and common optimizations.

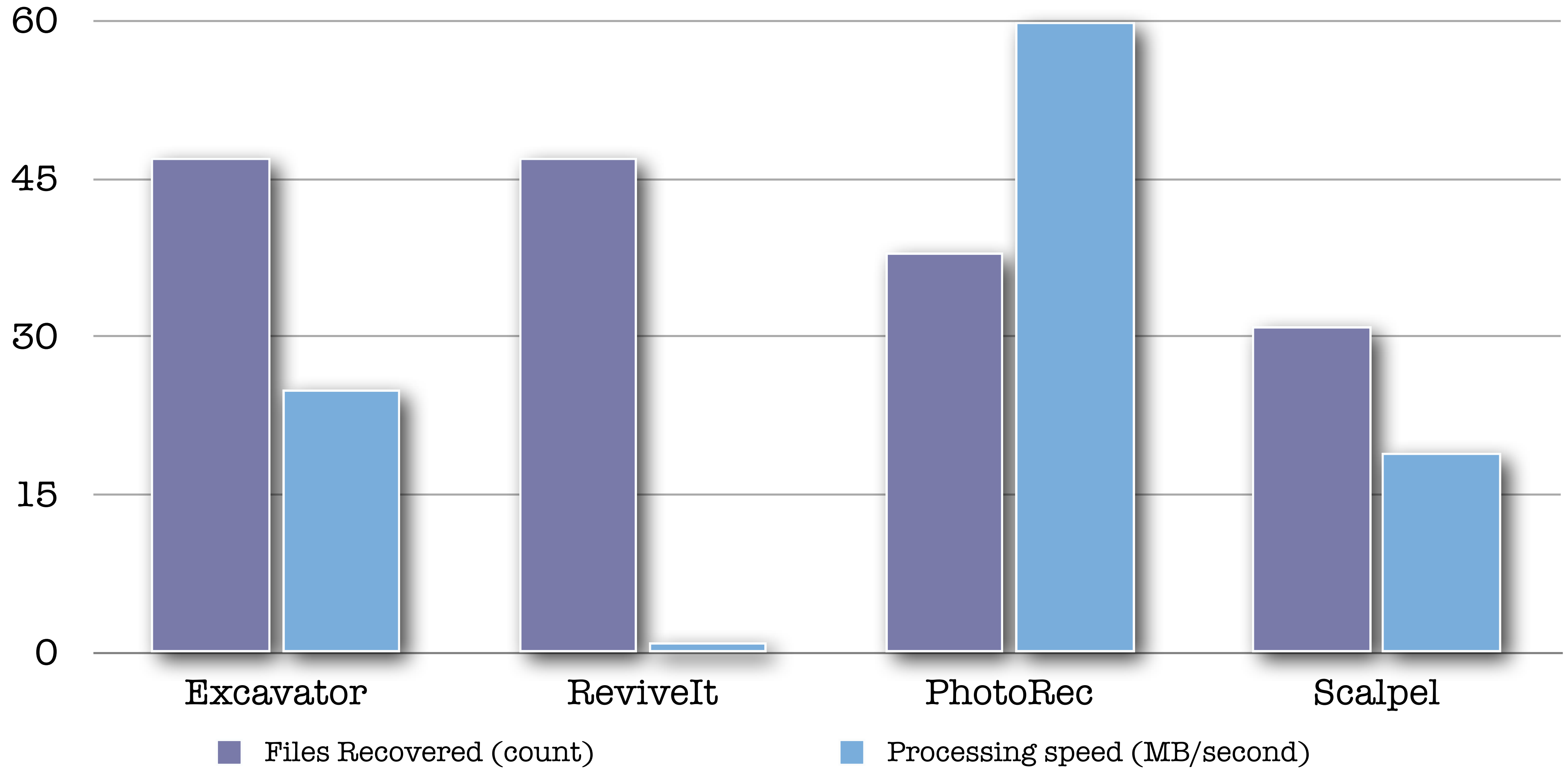
Slide from Jeroen van den Bos

Excavating Architecture



Slide from Jeroen van den Bos

Comparing to Existing Tools on a Set of Benchmarks



“Bringing Domain-Specific Languages to Digital Forensics”, van den Bos/van der Storm, ICSE’11.

Slide from Jeroen van den Bos

Software Repository Mining

Repository Mining

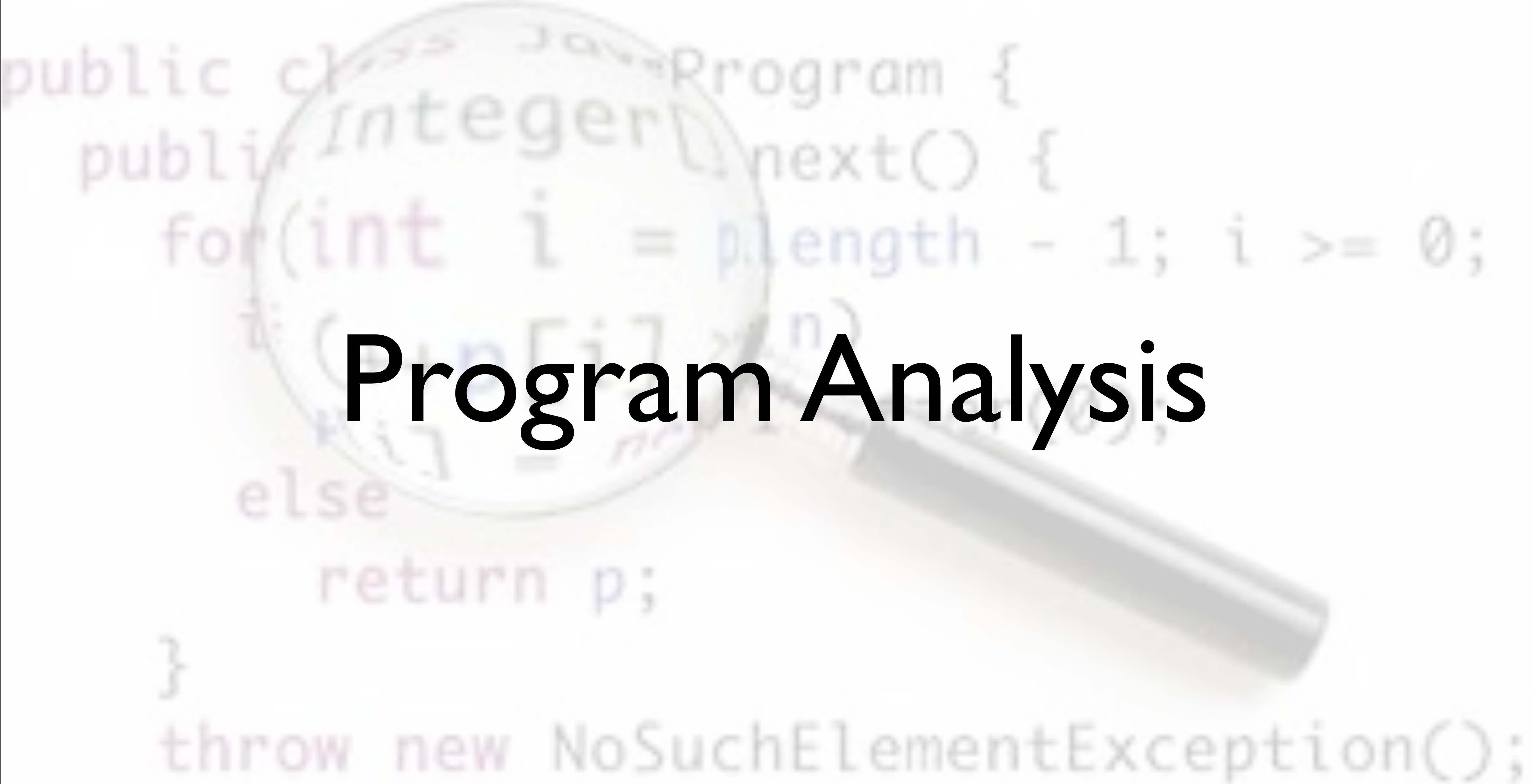
- “The Mining Software Repositories (MSR) field analyzes the rich data available in software repositories to uncover interesting and actionable information about software systems and projects.” -- MSR 2013 Homepage
- Repositories: version control, defect tracking, communications between team members
- Uses: support maintenance, improve design, facilitate reuse, empirical validation, prediction and planning

Example: What Features are Used in PHP?

- Goal: determine which features are used in PHP programs, what usage patterns are visible
- Special focus: which features are hard to analyze?
- Technique: extract system source from PHP repositories, perform statistical analysis over code bases of systems, use Rascal to identify interesting parts of code that we can look at more closely
- Corpus: 19 systems, close to 3.4 million lines of PHP

Results

- Of 109 language features, 7 are never used in the corpus, while 35 are not used in 80% of the files
- Most PHP files are below 1300 lines of code
- Most uses of variable-variables can be resolved statically (75% with systems that support PHP5)
- And more! (ask for the paper if you are interested...)



Program Analysis

Program Analysis

- Program analysis is an umbrella for a large number of techniques to programmatically discover information about programs
- Two camps: static and dynamic (with some mixing at the borders)
- Many techniques: abstract interpretation, control-flow analysis, data-flow analysis, augmented type systems (including type and effect systems), constraints
- Many uses: bug finding, optimization, verification

Example: Analysis of Lua Code (Reimer van Rozen)

- Lua is a popular scripting language, including for games
- Standard dynamic language challenges: no types, checks at runtime, high flexibility can lead to unexpected results
- Solution: static analysis of Lua in Rascal, with IDE tooling

EQuA

Lua AiR Framework

Contextual Analysis Example

Slide from Riemer van Rozen

Lua AiR Framework

Contextual Analysis Example

```
function f(c)      -- assign function to f
```

Slide from Riemer van Rozen

Lua AiR Framework

Contextual Analysis Example

```
function f(c)     -- assign function to f
  a       = 1     -- creates global a
```

Slide from Riemer van Rozen

Lua AiR Framework

Contextual Analysis Example

```
function f(c)     -- assign function to f
  a       = 1     -- creates global a
  local b = true  -- creates local b
```

Slide from Riemer van Rozen

EQuA

Lua AiR Framework

Contextual Analysis Example

```
function f(c)     -- assign function to f
  a       = 1     -- creates global a
  local b = true  -- creates local b
  a, b    = b, a  -- swap a and b
```

Slide from Riemer van Rozen

EQuA

Lua AiR Framework

Contextual Analysis Example

```
function f(c)     -- assign function to f
  a       = 1     -- creates global a
  local b = true  -- creates local b
  a, b    = b, a  -- swap a and b
  a, b    = 1,2,3 -- discards 3
```

Slide from Riemer van Rozen

Lua AiR Framework

Contextual Analysis Example

```
function f(c)     -- assign function to f
  a       = 1     -- creates global a
  local b = true  -- creates local b
  a, b    = b, a  -- swap a and b
  a, b    = 1,2,3 -- discards 3
  a, b    = c     -- implicitly deletes b
```

Slide from Riemer van Rozen

Lua AiR Framework

Contextual Analysis Example

```
function f(c)     -- assign function to f
  a       = 1     -- creates global a
  local b = true  -- creates local b
  a, b    = b, a  -- swap a and b
  a, b    = 1,2,3 -- discards 3
  a, b    = c     -- implicitly deletes b
  print(b)        -- nil, undeclared b
```

Slide from Riemer van Rozen

Lua AiR Framework

Contextual Analysis Example

```
function f(c)         -- assign function to f
  a        = 1        -- creates global a
  local b = true      -- creates local b
  a, b     = b, a     -- swap a and b
  a, b     = 1,2,3 -- discards 3
  a, b     = c        -- implicitly deletes b
  print(b)            -- nil, undeclared b
end                   -- close scope
```

Slide from Riemer van Rozen

Lua AiR Framework

Contextual Analysis Example

```
function f(c)     -- assign function to f
  a       = 1     -- creates global a
  local b = true  -- creates local b
  a, b    = b, a  -- swap a and b
  a, b    = 1,2,3 -- discards 3
  a, b    = c     -- implicitly deletes b
  print(b)        -- nil, undeclared b
end               -- close scope
f("4")            -- call f, bind c to "4"
```

Slide from Riemer van Rozen

EQuA

Lua AiR Framework

Contextual Analysis Example

```
function f(c)     -- assign function to f
  a       = 1     -- creates global a
  local b = true  -- creates local b
  a, b    = b, a  -- swap a and b
  a, b    = 1,2,3 -- discards 3
  a, b    = c     -- implicitly deletes b
  print(b)        -- nil, undeclared b
end               -- close scope
f("4")            -- call f, bind c to "4"
print(a)          -- 4, read global a
```

Slide from Riemer van Rozen

EQuA

Lua AiR Framework

Contextual Analysis Example

```lua
function f(c)     -- assign function to f
  a       = 1     -- creates global a
  local b = true  -- creates local b
  a, b    = b, a  -- swap a and b
  a, b    = 1,2,3 -- discards 3
  a, b    = c     -- implicitly deletes b
  print(b)        -- nil, undeclared b
end               -- close scope
f("4")            -- call f, bind c to "4"
print(a)          -- 4, read global a
d = 2 .. a        -- coerces 2 to string
```

Slide from Riemer van Rozen

EQuA

Lua AiR Framework

Contextual Analysis Example

```
function f(c)     -- assign function to f
  a         = 1   -- creates global a
  local b = true  -- creates local b
  a, b      = b, a -- swap a and b
  a, b      = 1,2,3 -- discards 3
  a, b      = c   -- implicitly deletes b
  print(b)        -- nil, undeclared b
end               -- close scope
f("4")            -- call f, bind c to "4"
print(a)          -- 4, read global a
d = 2 .. a        -- coerces 2 to string
d = d / "12"      -- coerces 12 to number
```

Slide from Riemer van Rozen

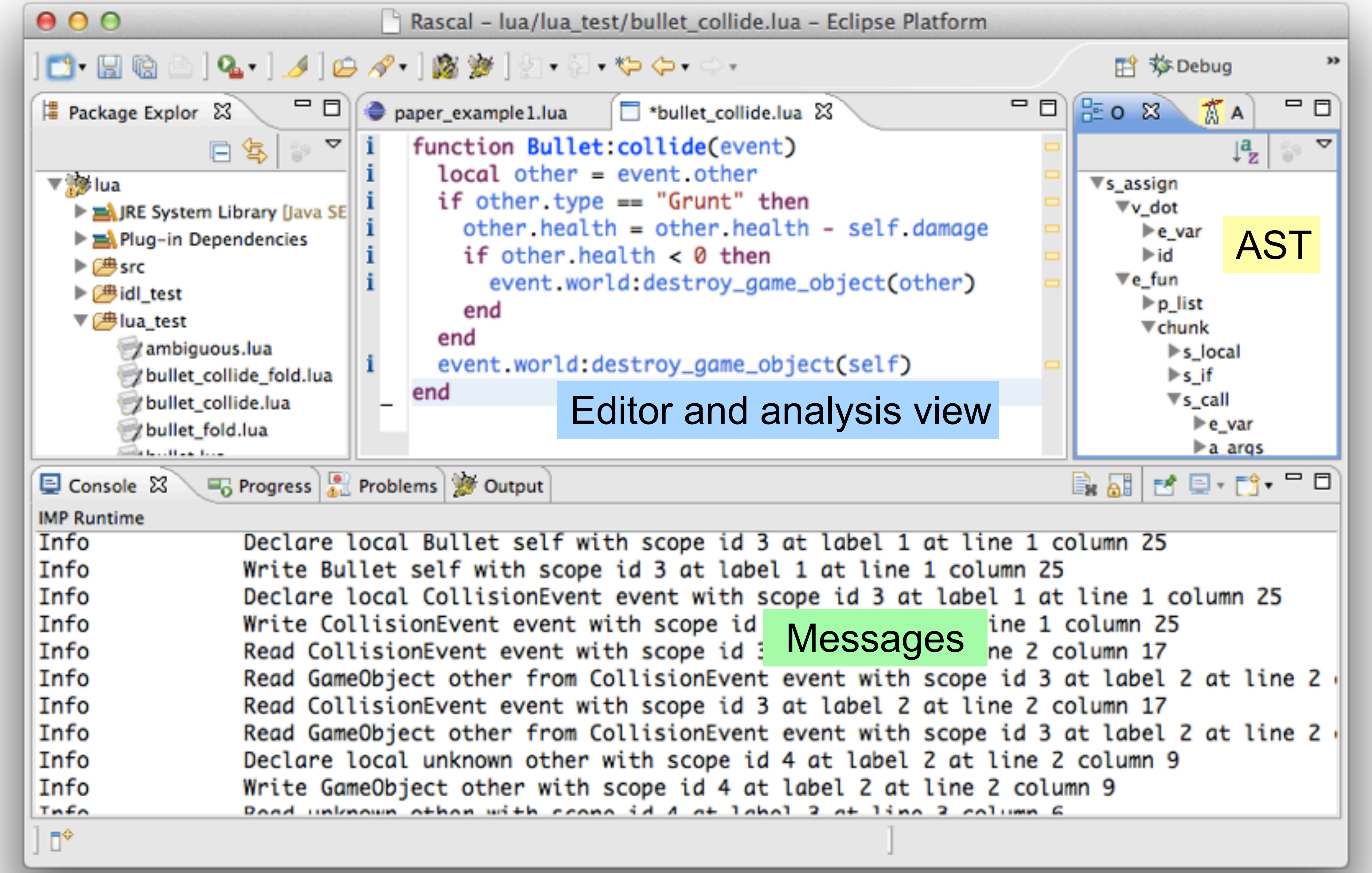
Lua AiR Framework

Contextual Analysis Example

```
function f(c)     -- assign function to f
  a       = 1     -- creates global a
  local b = true  -- creates local b
  a, b    = b, a  -- swap a and b
  a, b    = 1,2,3 -- discards 3
  a, b    = c     -- implicitly deletes b
  print(b)        -- nil, undeclared b
end               -- close scope
f("4")            -- call f, bind c to "4"
print(a)          -- 4, read global a
d = 2 .. a        -- coerces 2 to string
d = d / "12"      -- coerces 12 to number
print(c, d)       -- nil 2, undeclared c
```

Slide from Riemer van Rozen

Example: Lua IDE



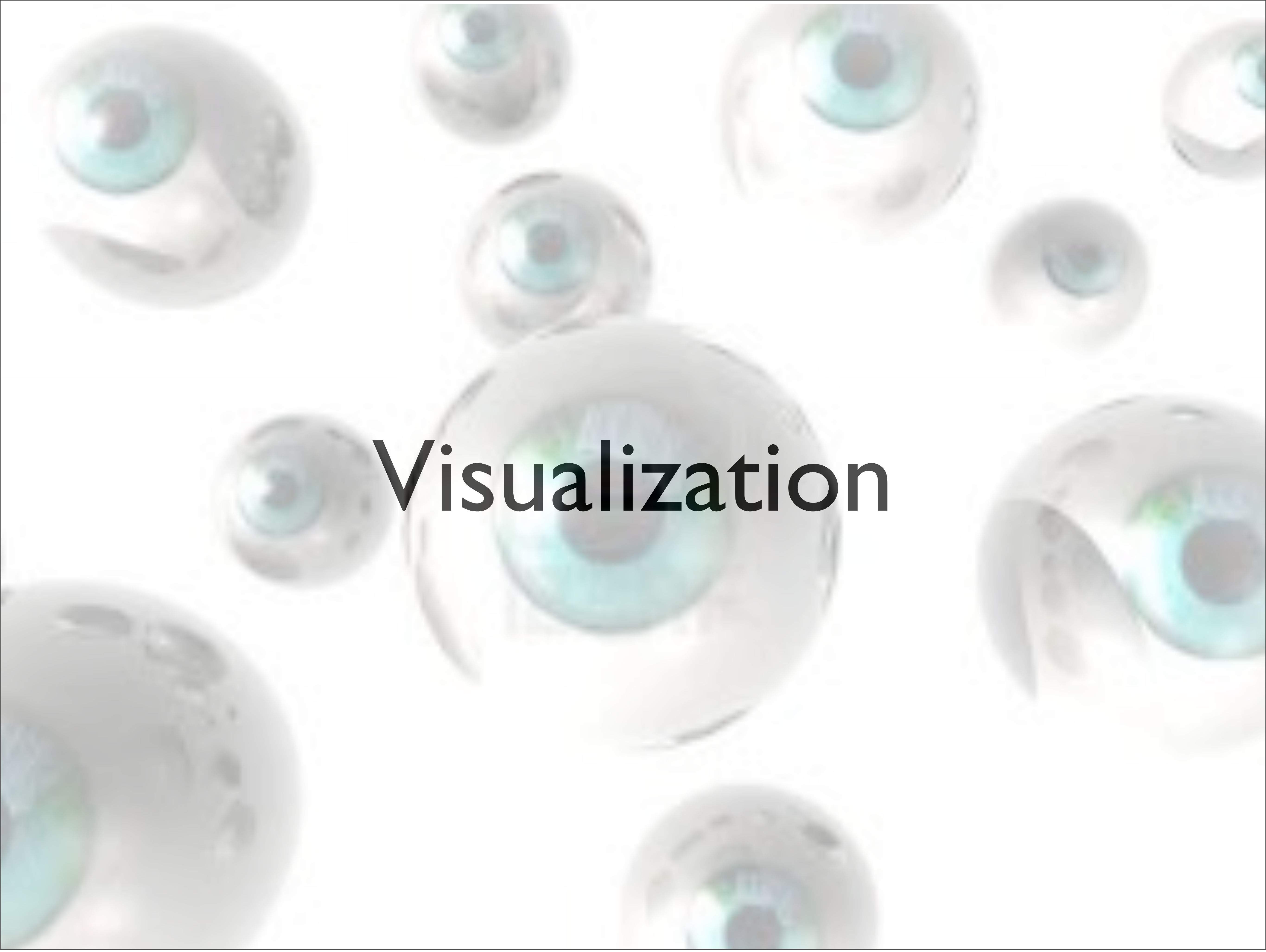
```
function Bullet:collide(event)
  local other = event.other
  if other.type == "Grunt" then
    other.health = other.health - self.damage
    if other.health < 0 then
      event.world:destroy_game_object(other)
    end
  end
  event.world:destroy_game_object(self)
end
```

```
IMP Runtime
Info        Declare local Bullet self with scope id 3 at label 1 at line 1 column 25
Info        Write Bullet self with scope id 3 at label 1 at line 1 column 25
Info        Declare local CollisionEvent event with scope id 3 at label 1 at line 1 column 25
Info        Write CollisionEvent event with scope id                 ine 1 column 25
Info        Read CollisionEvent event with scope id 3                ne 2 column 17
Info        Read GameObject other from CollisionEvent event with scope id 3 at label 2 at line 2
Info        Read CollisionEvent event with scope id 3 at label 2 at line 2 column 17
Info        Read GameObject other from CollisionEvent event with scope id 3 at label 2 at line 2
Info        Declare local unknown other with scope id 4 at label 2 at line 2 column 9
Info        Write GameObject other with scope id 4 at label 2 at line 2 column 9
Info        Read unknown other with scope id 4 at label 3 at line 3 column 6
```

Slide from Riemer van Rozen

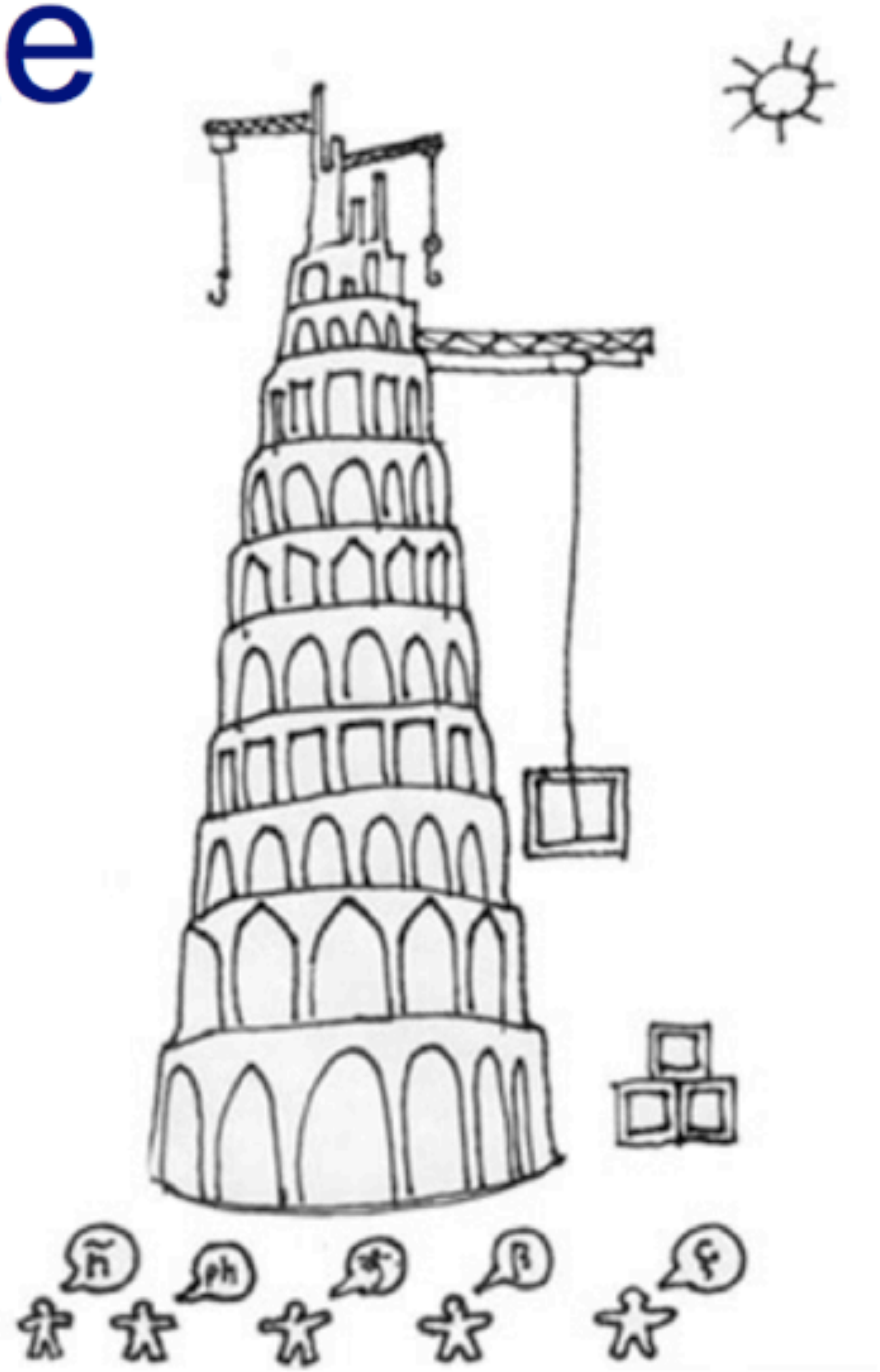
Example: Analysis of PHP (Ongoing Work)

- Eventual goal: full suite of PHP analysis tools
- Current work: the basics!
  - Analysis of file includes
  - Type inference
  - Alias analysis
- Some promising initial work on statically resolving includes, which are a dynamic property

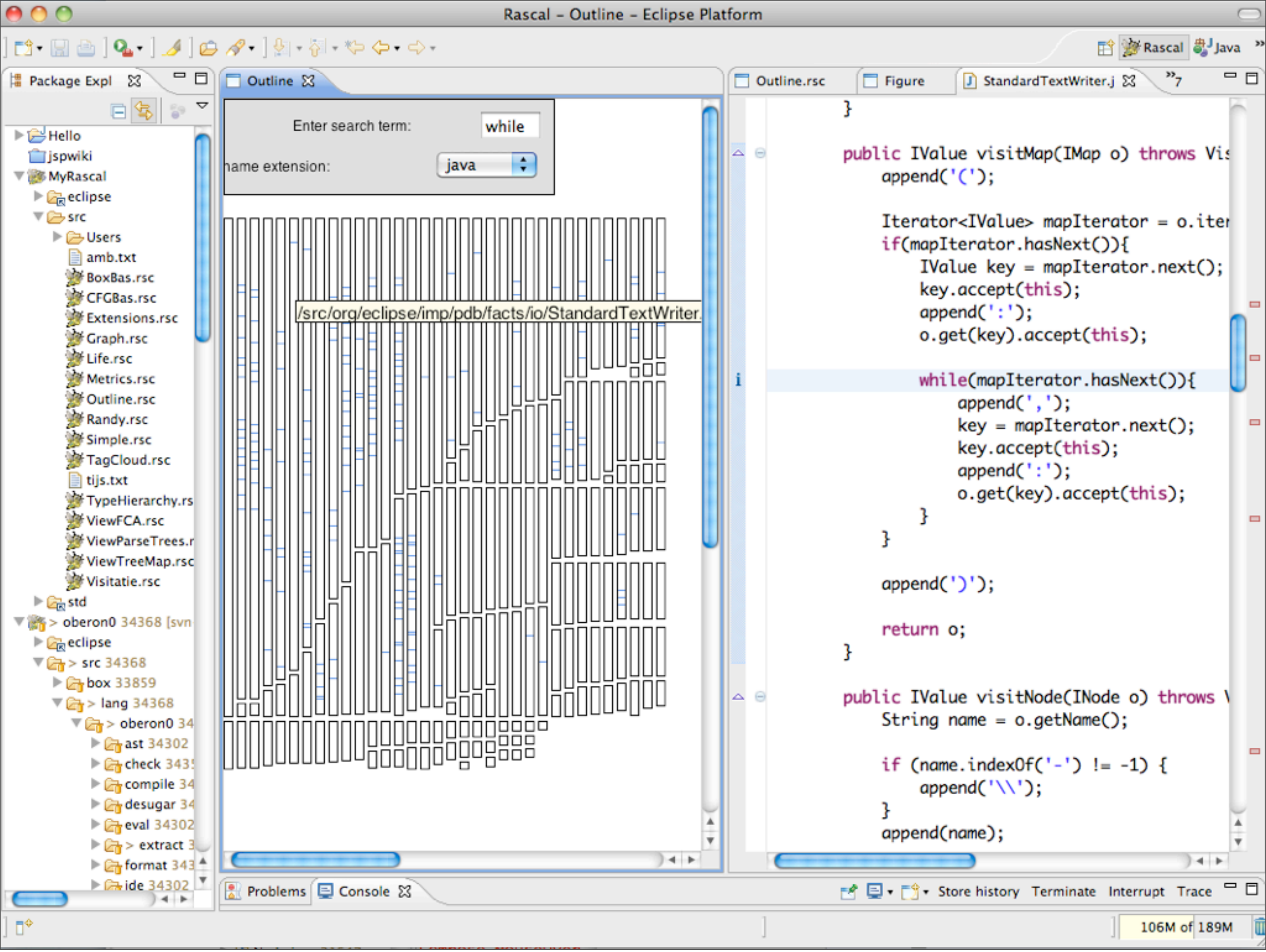


Visualization

How to integrate Software Visualization in Rascal?



Slide from Paul Klint



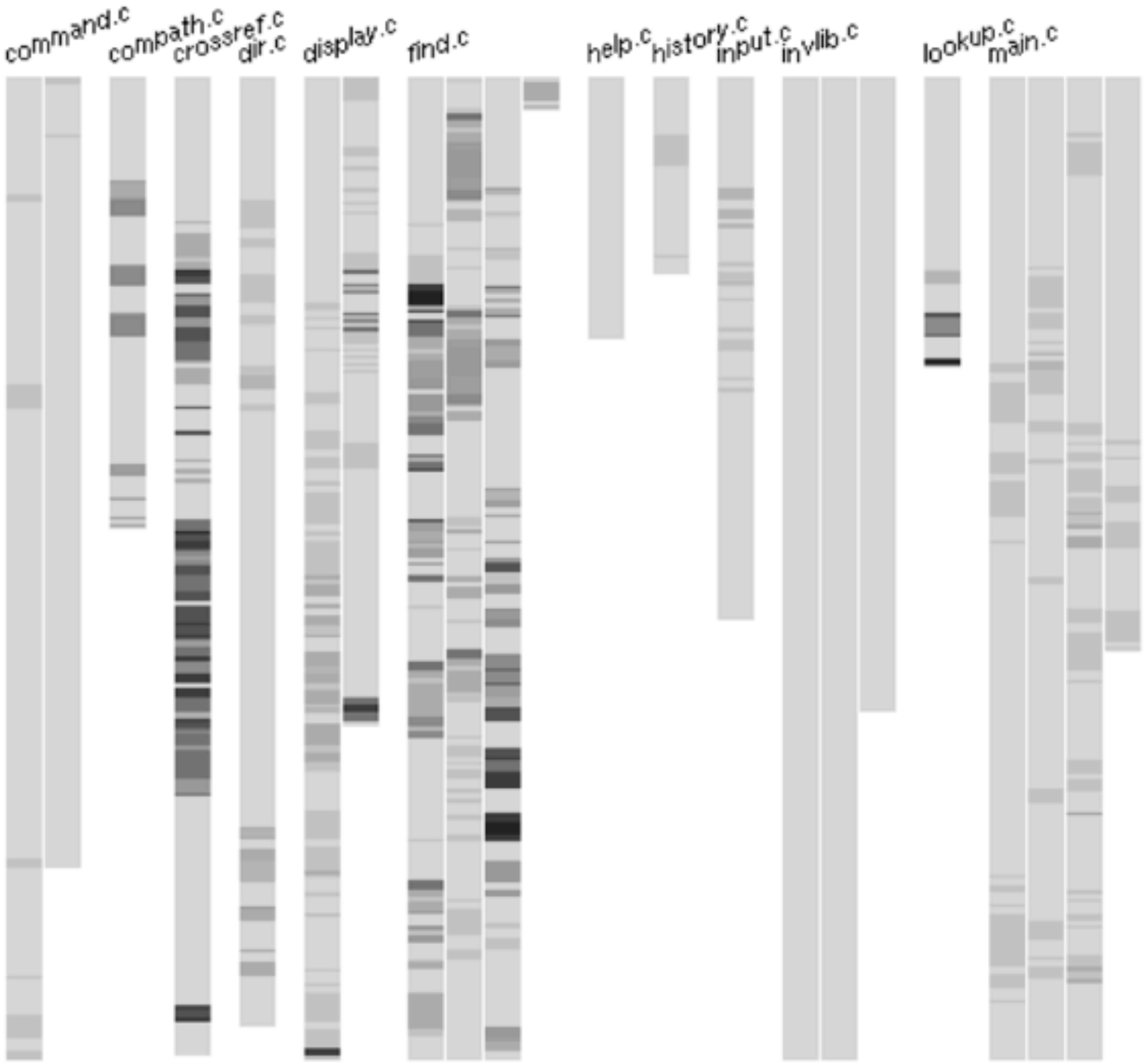
Enter search term: while

name extension: java

/src/org/eclipse/imp/pdb/facts/io/StandardTextWriter.

```
        }

        public IValue visitMap(IMap o) throws Vis
            append('(');

            Iterator<IValue> mapIterator = o.iter
            if(mapIterator.hasNext()){
                IValue key = mapIterator.next();
                key.accept(this);
                append(':');
                o.get(key).accept(this);

                while(mapIterator.hasNext()){
                    append(',');
                    key = mapIterator.next();
                    key.accept(this);
                    append(':');
                    o.get(key).accept(this);
                }
            }

            append(')');

            return o;
        }

        public IValue visitNode(INode o) throws
            String name = o.getName();

            if (name.indexOf('-') != -1) {
                append('\\');
            }
            append(name);
```

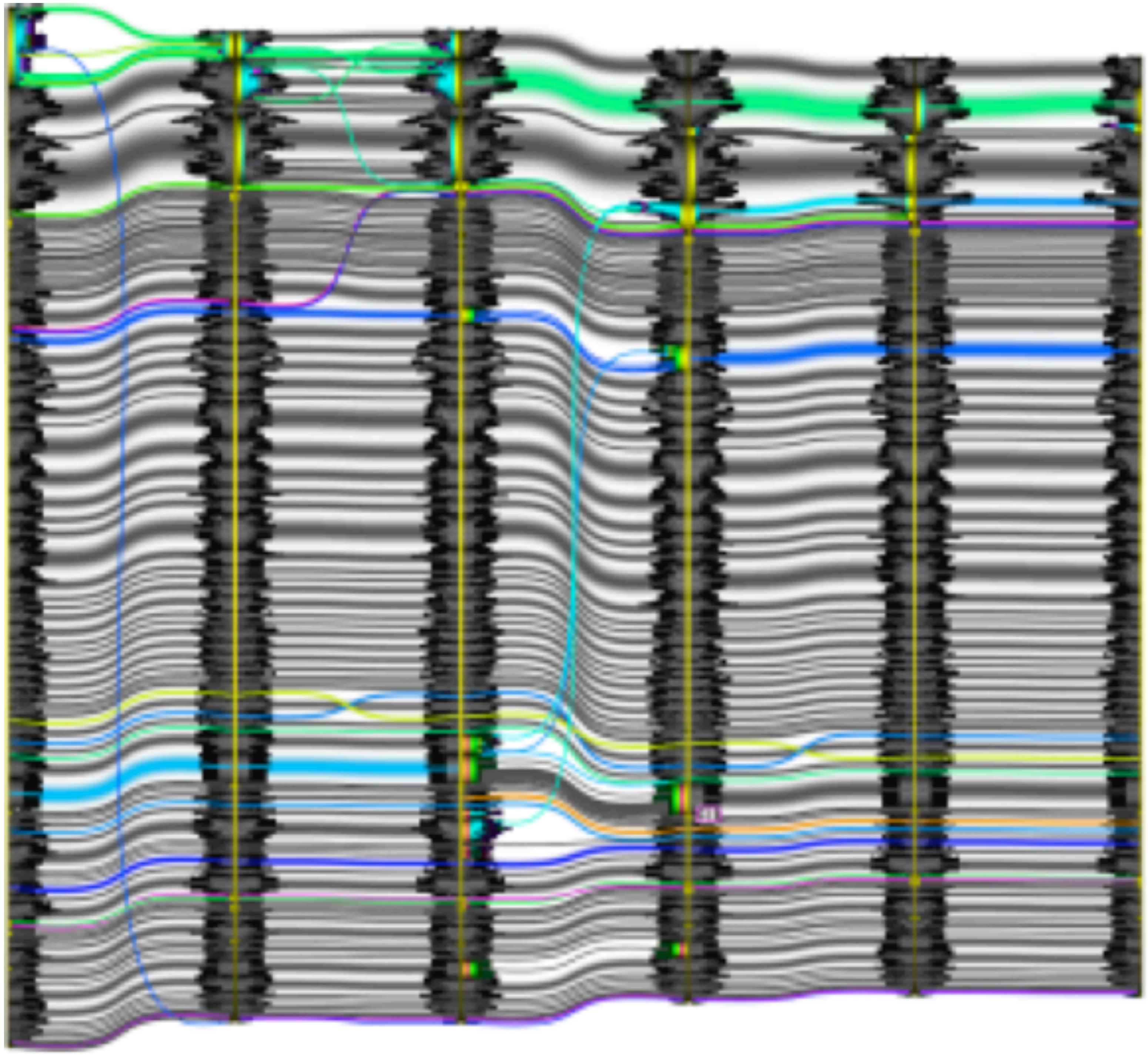
Software Visualization: Execution Frequency



Credits: **Steven Eick**

Slide from Paul Klint

Software Visualization: Revision Histories



Credits: Alex Telea, RUG

Slide from Paul Klint

Software Visual Analytics

- Emerging field where data extracted from software artifacts are visualized in order to
  - **Understand** the software: Architecture? Component dependencies?
  - **Identify** parts with special properties: Most complex? Most revisions? Test coverage?
  - **What if** questions: What happens if we adapt this part?

Slide from Paul Klint

Rascal Visualization Design Principles

- Automatic & Domain-Specific: reduce low-level issues (layout, size), automate mappings (e.g., axis, color scale, ...), automate interaction support

- Reuse: treat figures and visual attributes as ordinary values; can be parameters/result of functions, arbitrary nesting of figures, well-defined composition of visual attributes

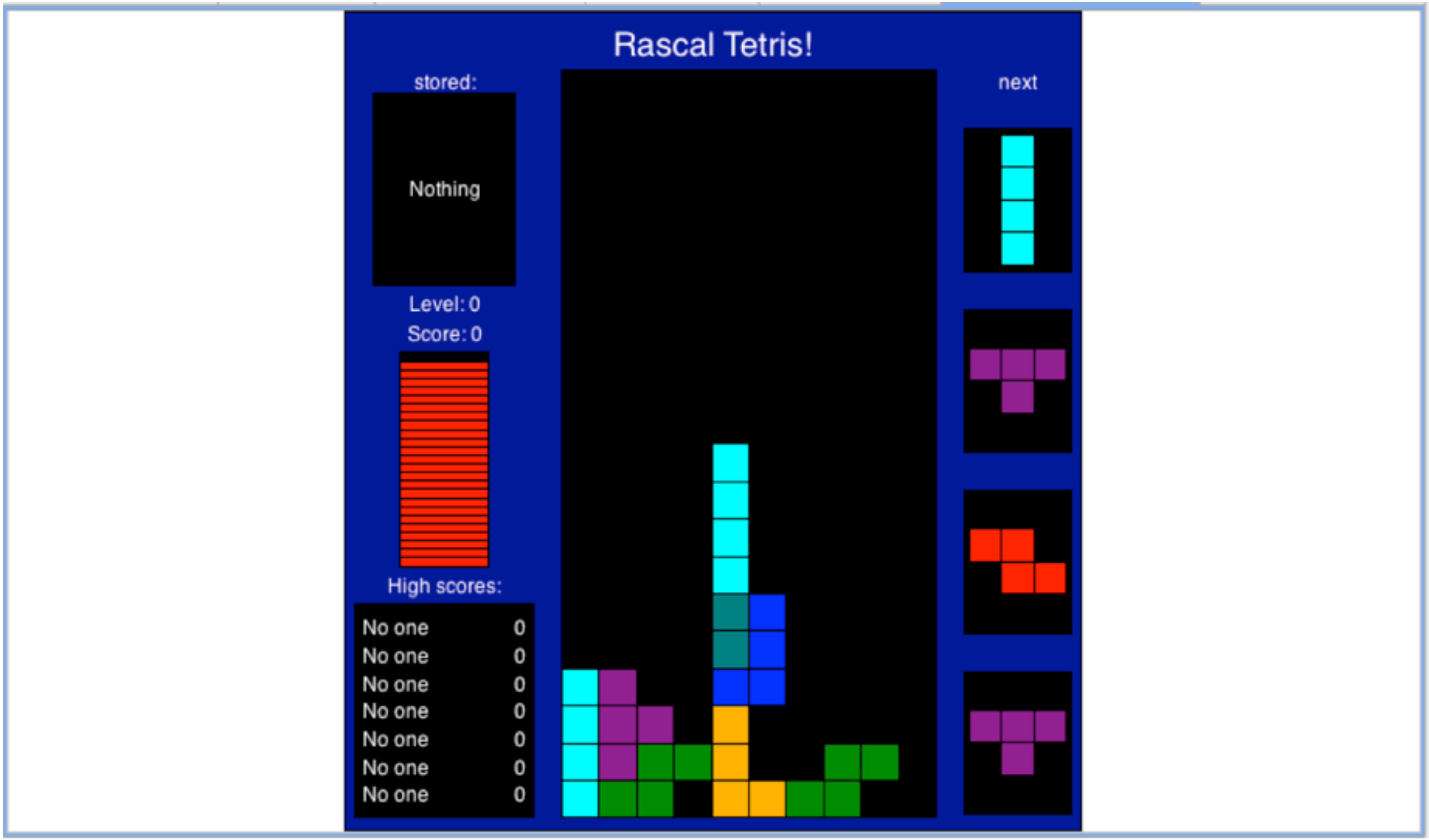
Slide from Paul Klint

Rascal Visualization Design Principles

- Compositionality: global visualization state (e.g. Pen color) hinders composition, self-contained, composable, visualizations
- Interactivity: enable Schneidermann's Mantra of Overview First, Zoom and Filter, then Details-on-demand, provide the GUI-elements (buttons, text fields, ...) to achieve this.

Slide from Paul Klint

And, of course, the ultimate goal...



Rascal Tetris!

stored:

Nothing

Level: 0

Score: 0

High scores:

No one	0
No one	0
No one	0
No one	0
No one	0
No one	0
No one	0

Slide from Paul Klint

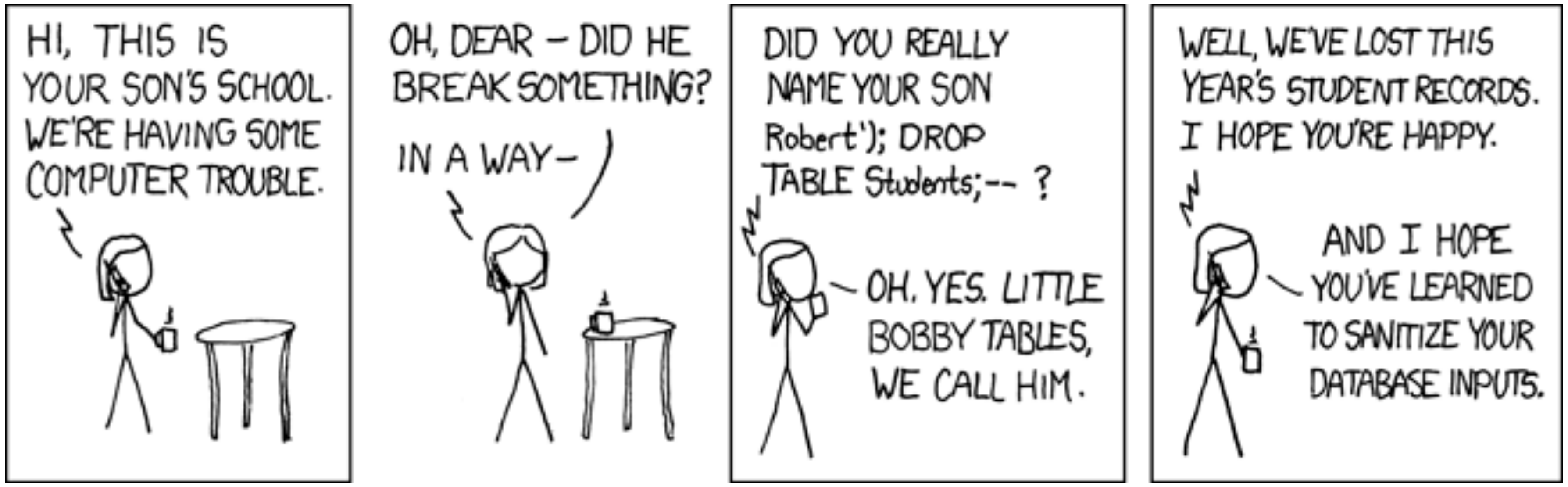
Ideas for Assignments

Assignment Ideas: Grab-bag, needs work...

- Parsing -- see Ali Afroozeh's talk from November 13
- DSL construction -- challenge here is coming up with something novel and useful in the limited timeframe
- Data-rich programming: add support for new formats, like RDF -- challenge is you really need something useful to do with it
- IDE support: can we use information in IDEs for other languages to provide support similar to what we have with Java?

Assignment Idea #1: Taint Analysis in PHP

- Problem: user inputs in GET and POST should not be used directly in database queries
- Solution: http://www.php.net/manual/en/security.database.sql-injection.php
- Analysis: verify that, along all paths, steps are taken to sanitize strings before they are used in queries


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

http://xkcd.com/327/

Assignment Idea #2: MSR

- Context: major changes from PHP4 to PHP5, many upgraded systems

- Question 1: How have OO features been adopted?

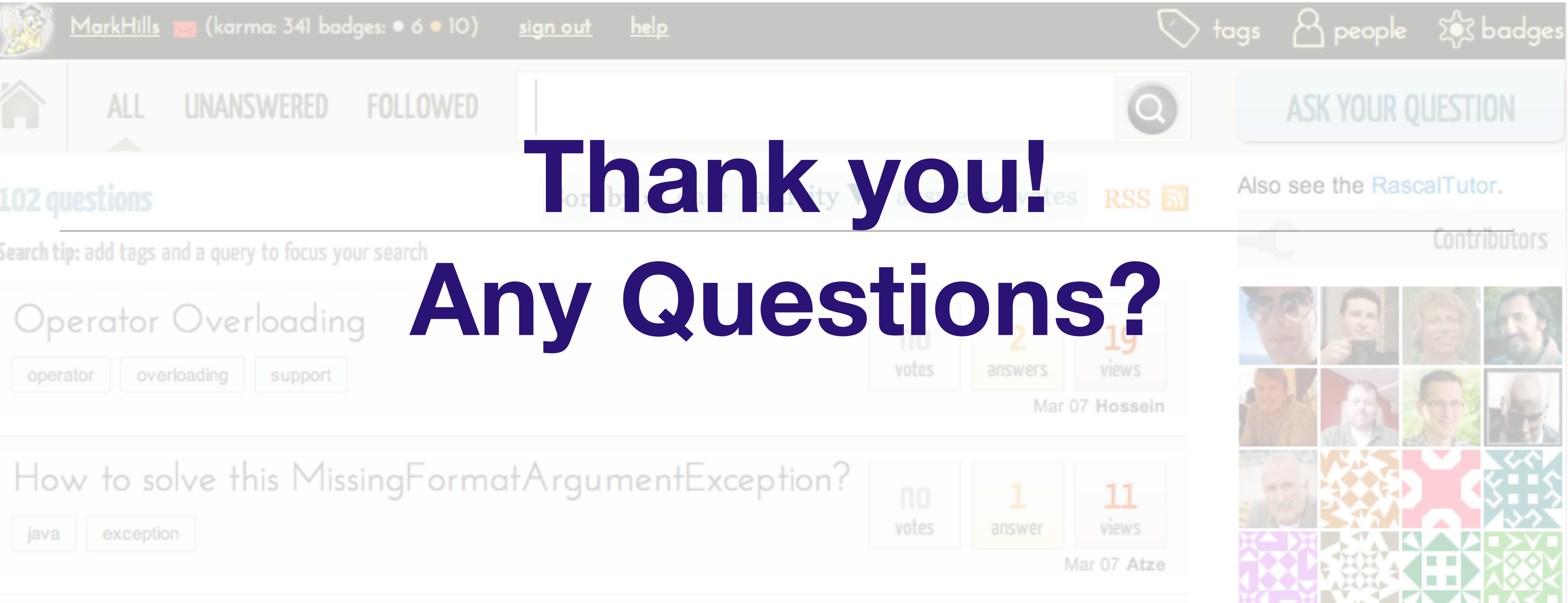
- Question 2: Does this lead to differences in popular code quality metrics?

- Question 3: Can information in the repository be tied into support of new features and language changes?

- Question 4: Can we identify committers that are improving quality metrics?

Thank you!

Any Questions?



102 questions

Also see the RascalTutor.

Search tip: add tags and a query to focus your search

Operator Overloading

operator overloading support

no votes 2 answers 19 views

Mar 07 Hossein

How to solve this MissingFormatArgumentException?

java exception

no votes 1 answer 11 views

Mar 07 Atze

- Rascal: http://www.rascal-mpl.org
- SEN1: http://www.cwi.nl/sen1
- Me: http://www.cwi.nl/~hills