Meta-Programming, Program Analysis, and Software Analytics

Prof. Mark Hills
Department of Computer Science
East Carolina University
May 25, 2016

http://www.rascal-mpl.org

Who am I

- Undergrad: CS at Western Illinois University
- 8 years experience as a software engineer
- PhD: University of Illinois at Urbana-Champaign, formal semantics of languages for language prototyping, program analysis, and formal methods
- Postdoc: Centrum Wiskunde & Informatica, Software Analysis & Transformation group, meta-programming for program analysis
- Assistant Prof at ECU for 3 years

What do I work on?

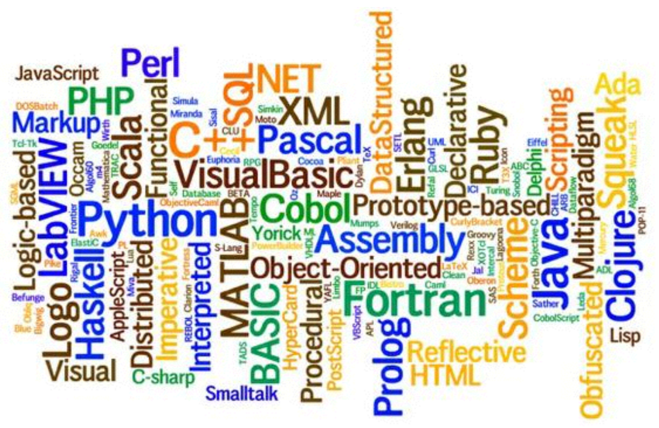
- Program Analysis
- Meta-Programming Languages
- Software Analytics/Empirical Software Engineering
- Software Evolution
- Formal Semantics of Programming Languages
- Formal Methods

Digging deeper…

- Defining and working with programming languages
- Applying meta-programming languages
- Analyzing programs
- Introducing software analytics

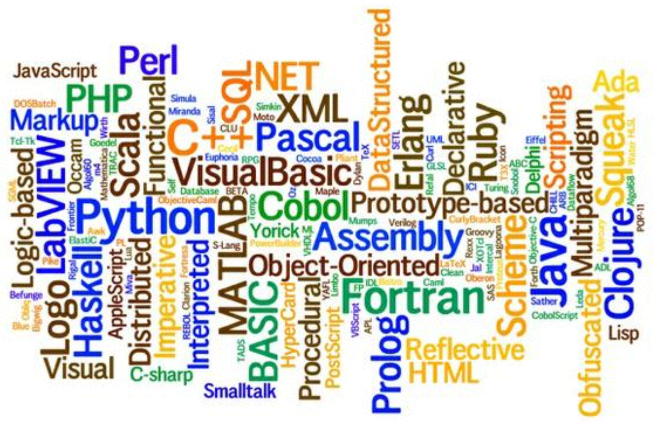
Programming Languages

Important Concepts: Syntax



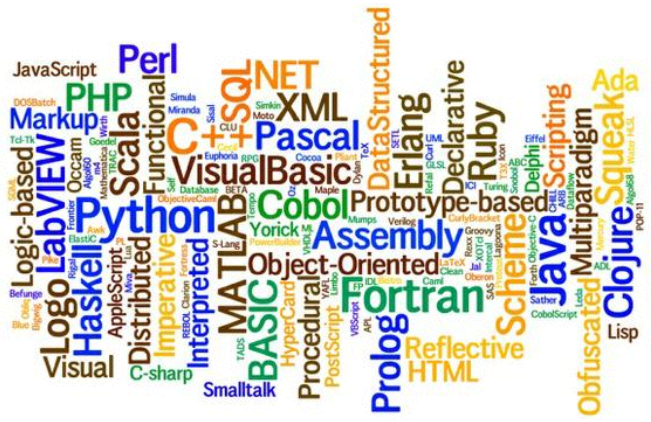
- Scanners and regular expressions
- Parsers and grammars
- Scannerless parsers
- Parse trees and ASTs

Important Concepts: Semantics



- What do we mean by semantics?
- How do we define semantics?
  - Informal techniques
  - Formal techniques
  - By implementation
- Why define a semantics at all?

Important Concepts: Tools



- What kinds of tools do we use for programming languages?
  - Compilers
  - Interpreters
  - Refactoring Tools
  - Metrics
  - Analysis and analytics

What is Meta-Programming?

Defining Meta-Programming

- What do we mean by “meta”?
  - “meta-cognition”
  - “meta-data”
  - “this is so meta…”

Defining Meta-Programming

- What do we mean by “meta”?
  - “meta-cognition”
  - “meta-data”
  - “this is so meta…”
- Meta-programs = programs that manipulate other programs
- Terms: meta-level, object level

Meta-Programming Usage Scenarios

- Program analysis (see later!)
- Program transformation/refactoring
- Fact extraction
- Language implementation

Meta-Programming Usage Scenarios

- Program analysis (see later!)
- Program transformation/refactoring
- Fact extraction
- Language implementation

- This covers many of the tool areas we discussed earlier

What is Program Analysis?

Important Concepts: Static Analysis

- What makes an analysis static?

Important Concepts: Static Analysis

- What makes an analysis static?
  - You do it *without* running the program
- Why?

Important Concepts: Static Analysis

- What makes an analysis static?
  - You do it *without* running the program
- Why?
  - You can capture all the possible behaviors, not just those you see
  - Running a program may be expensive
  - Many uses (e.g., refactoring) are inherently static

Important Concepts: Static Analysis

- What makes an analysis static?
  - You do it *without* running the program
- What are possible problems?

Important Concepts: Static Analysis

- What makes an analysis static?
  - You do it *without* running the program
- What are possible problems?
  - Some programs are very dynamic, hard to statically capture behavior
  - Some programs make heavy use of libraries, reflection, FFIs, may be hard to actually get the code you are analyzing

Important Concepts: Dynamic Analysis

- What makes an analysis dynamic?

Important Concepts: Dynamic Analysis

- What makes an analysis dynamic?
  - You do it by *running* the program (and maybe extrapolating based on that)
- Why?

Important Concepts: Dynamic Analysis

- What makes an analysis dynamic?
  - You do it by *running* the program (and maybe extrapolating based on that)
- Why?
  - guaranteed that the problems you find are real problems (if you don't extrapolate)
  - deals with libraries, reflection, FFIs, etc, we can at least see what we give them and what they give back

Important Concepts: Dynamic Analysis

- What makes an analysis dynamic?
  - You do it by *running* the program (and maybe extrapolating based on that)
- What are possible problems?

Important Concepts: Dynamic Analysis

- What makes an analysis dynamic?
  - You do it by *running* the program (and maybe extrapolating based on that)
- What are possible problems?
  - we may miss a problem we are looking for if we don’t take that program path
  - too expensive to use to back tools like IDEs

Important Concepts: Sound/Complete

- Soundness: we don’t miss what we are looking for
- Completeness: if we find something it’s an actual case of what we are looking for
- Why may these conflict?

Important Concepts: Sound/Complete

- Soundness: we don't miss what we are looking for
- Completeness: if we find something it's an actual case of what we are looking for
- Why may these conflict?
  - may be hard to be sound without also finding possible errors where there are none
  - may be hard to remove spurious errors without also removing some real ones

Common examples

- IDEs
  - refactoring, code completion, …
- Compiler optimizations
  - common subexpression elimination, LICM, …
- Security analysis
  - finding SQL injection vulnerabilities, …

What is Software Analytics?

Defining Software Analytics

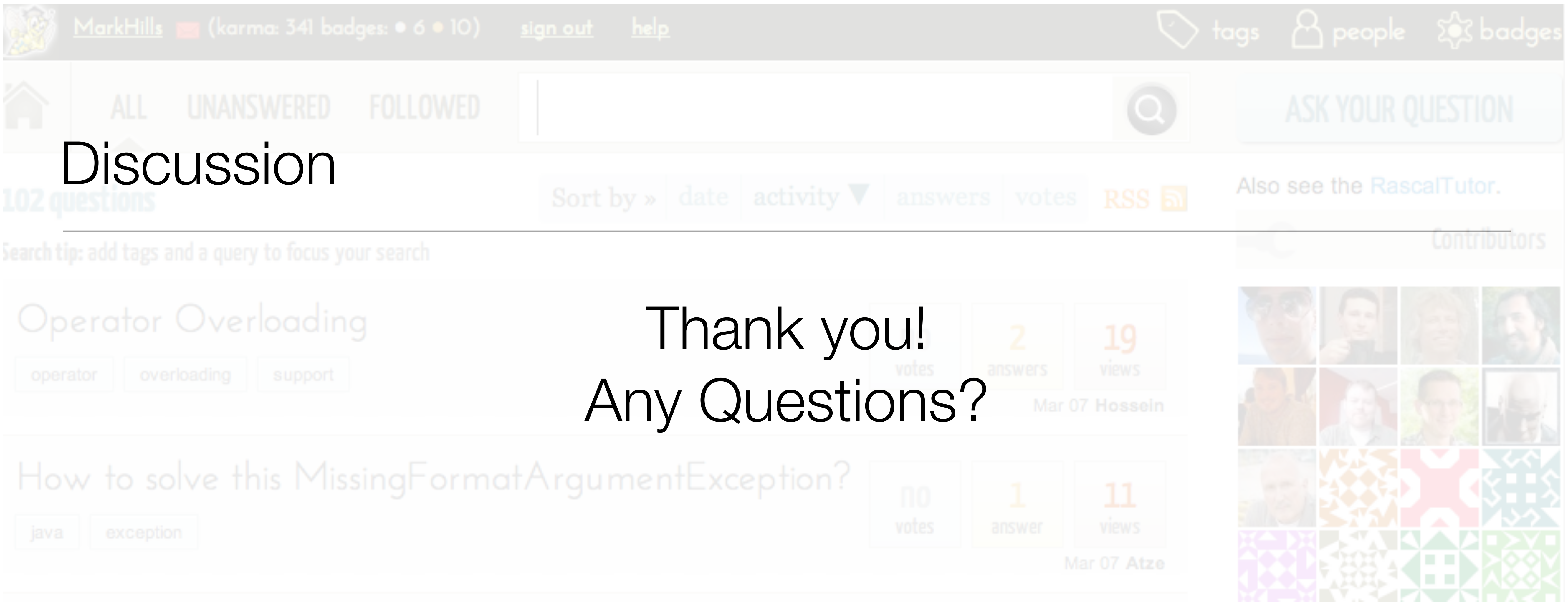
"Software analytics is analytics on software data for managers and software engineers with the aim of empowering software development individuals and teams to gain and share insight from their data to make better decisions."

R. Buse and T. Zimmermann, "Information Needs for Software Development Analytics," Proc. Int'l Conf. Software Eng. (ICSE), IEEE CS, 2012; http://thomas-zimmermann.com/publications/details/buse-icse-2012.

Use Cases

- Explore and analyze large systems and their "ecosystem" (code repositories, bug repositories, project management information, etc)
- Inform decision making
- Make predictions based on existing data
- Bring together various sources of project data (see first point)

Discussion


MarkHills (karma: 341 badges: ● 6 ● 10) sign out help
tags people badges
ALL UNANSWERED FOLLOWED
ASK YOUR QUESTION
102 questions
Sort by » date activity ▼ answers votes RSS
Also see the RascalTutor.
Search tip: add tags and a query to focus your search
Contributors

Operator Overloading
operator overloading support
2 answers 19 views
Mar 07 Hossein

How to solve this MissingFormatArgumentException?
java exception
no votes 1 answer 11 views
Mar 07 Atze

Thank you!
Any Questions?

- Rascal: http://www.rascal-mpl.org
- Me: http://www.cs.ecu.edu/hillsma