Universidade Federal do Rio Grande do Sul
Programa de Pós-Graduação em Computação

Swarm Debugging

The Collective Debugging Intelligence of the Crowd

Fábio Petrillo

Orientadores
Prof. Marcelo Pimenta
Profa. Carla Freitas

Porto Alegre, 19 outubro de 2016

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf INSTITUTO DE INFORMÁTICA UFRGS

Outline

- Motivation
- Background and Related Work
- Swarm Debugging
- Swarm Debug Infrastructure
- Evaluation
- Conclusion and Future Work

Is **debugging** important?

In software maintenance, debugging is an **everyday** activity

(TANENBAUM; BENSON, **1973**)

Is debugging an important activity?

"Developers spend over two-thirds of their time (68%) investigating code, and the majority of this time is spent **debugging** code (**33%**)."

LaToza, T. D., & Myers, B. a. (2010). Developers ask reachability questions. *2010 ACM/IEEE 32nd International Conference on Software Engineering*, *1*, 185–194.

Is debugging an important activity?

- **100%** of developers **execute** a system to understand its source code
- **80%** debugging
- Developers **avoid understanding** software systems (MAALEJ et al., 2014a): they want **only to fix** bugs
- Debugging is still a “hot” research topic

Maalej, W., Tiarks, R., Roehm, T., & Koschke, R. (2014). On the Comprehension of Program Comprehension. ACM Transactions on Software Engineering and Methodology, 23(4), 1–37.

Debugging is hard

Debugging is a **tedious** task and a **huge** effort!

Fleming, S. D., Scaffidi, C., Piorkowski, D., Burnett, M., Bellamy, R., Lawrance, J., & Kwan, I. (2013). An Information Foraging Theory Perspective on Tools for Debugging, Refactoring, and Reuse Tasks. ACM Transactions on Software Engineering and Methodology, 22(2), 1–41.

Problem

1. Debugging **paradigm** is basically the same
   - Breakpoints
   - Stepping
   - Tracing
2. Debugging is usually considered as an **individual** activity
3. Context information is **implicit** and not **captured**
4. Debugging knowledge is **not used** in next debugging sessions

Objectives

1. Proposing Swarm Debugging (SD) approach:
   - debugging as a **collective** activity
   - **explicit** context information
   - debugging knowledge can be **reused**

“Swarm Debugging is an approach that use collective intelligence to collect and share interactive debugging data, providing visualizations and searching tools to support software maintenance activities.”

2. Define and implements SDI as a support for SD

“We propose the Swarm Debug Infrastructure (SDI), with which practitioners and researchers can collect and share data about developers’ interactive debugging activities.”

Outline

- Motivation
- Background and Related Work
- Swarm Debugging
- Swarm Debug Infrastructure
- Evaluation
- Conclusion and Future Work

Debugging - Definition

"***Debug.*** *Set of techniques to detect, locate, and correct faults in a computer program. Techniques include the use of breakpoints, desk checking, dumps, inspection, reversible execution, single-step operations, and traces.*"

—IEEE Standard Glossary of SE Terminology—

Interactive debugging

- Debugging using an **interactive** tool -> debuggers!!!!
  - Navigate through the code
  - Stepping
  - State of variables
- Interactive debugging -> **gain of knowledge** (TIARKS; RöHM, 2013)
- First task -> **define** breakpoints
- Toggle breakpoints is an **“extremely difficult”** task (TIARKS; RöHM, 2013)
- (TIARKS; RöHM, 2013) claim that developers often simplistically toggle several irrelevant breakpoints

Debugging tools

- Debugging tools are **essentials** (CHI; NIERSTRASZ; GÎRBA, 2013)
- Automated debugging tools are not **helpful** (PARNIN; ORSO, 2011)
- Hipikat (ČUBRANIĆ et al., 2005)
  - content of artefacts
- Jive (GESTWICKI; JAYARAMAN, 2005)
  - execution trace, does not work with breakpoints, no sharing
- DebugAdvisor (ASHOK et al., 2009)
  - similar issues, no uses fine-grained data (breakpoints, events, invocations, paths, etc)
- Collaborative Debugging (ESTLER et al., 2013)
  - **synchronous approach**: after session, debugging **data are not reused**
  - global software

Information Foraging Theory

- Inspired by biological sciences, **pray/predator** metaphor
- Information foraging theory (IFT) - Pirolli and Card (PIROLLI; CARD, 1999) to understand how **individuals search information** in Web
- (LAWRANCE; BELLAMY; BURNETT, 2007) ITF - how professional **developers explore** on source code during **maintenance**.
- (FLEMING et al., 2013): an Information Foraging Theory Perspective on Tools for **Debugging**

Swarm Intelligence

- simple **agents** interacting locally, following very **simple rules** without central **coordination** simple: repeated interactions between individuals can produce complex adaptive **patterns** at the level of the group, since individual units do **not have a complete picture**
- Previously used for sw teams: Software teams have used collaborative and self-organisation approaches because software projects have some **analogies with collective behaviors** (CHOW; CAO, 2008)
- We proposed Swarm Intelligence **as a new metaphor** for Interactive Debugging : Swarm Debugging!

Task Context Model (KERSTEN; MURPHY, 2006)

- Task context is created by **monitoring** a programmer's activity and extracting the structural **relationships** of program artifacts.
- Operations on task contexts **integrate** with development **environment** features
- **Mylyn (Mylar)**
- Degree-of-interest (DOI): based on the **frequency** of interactions with the element and a measure of the interactions' recency.

Sharing (debugging) information...

- A new tendence of **collaboration** in SE - Crowd (STOREY et al., 2014)
- **Lack of tool support for collective activities** (crowd) in SE (STOREY et al., 2014)
- **Developers are willing to share information** about work collected by IDE automatically. (MAALEJ et al., 2014a)
- (MAALEJ et al., 2014a) suggests implementation of instrumentation of the IDE and **continuous observation** of developers' work.
- Unfortunately context information is typically implicit and not captured: for example, association issues/tasks and debugging sessions.

if **interactive debugging** is important to create **knowledge** about a software project and a huge effort, why **waste** it?

Outline

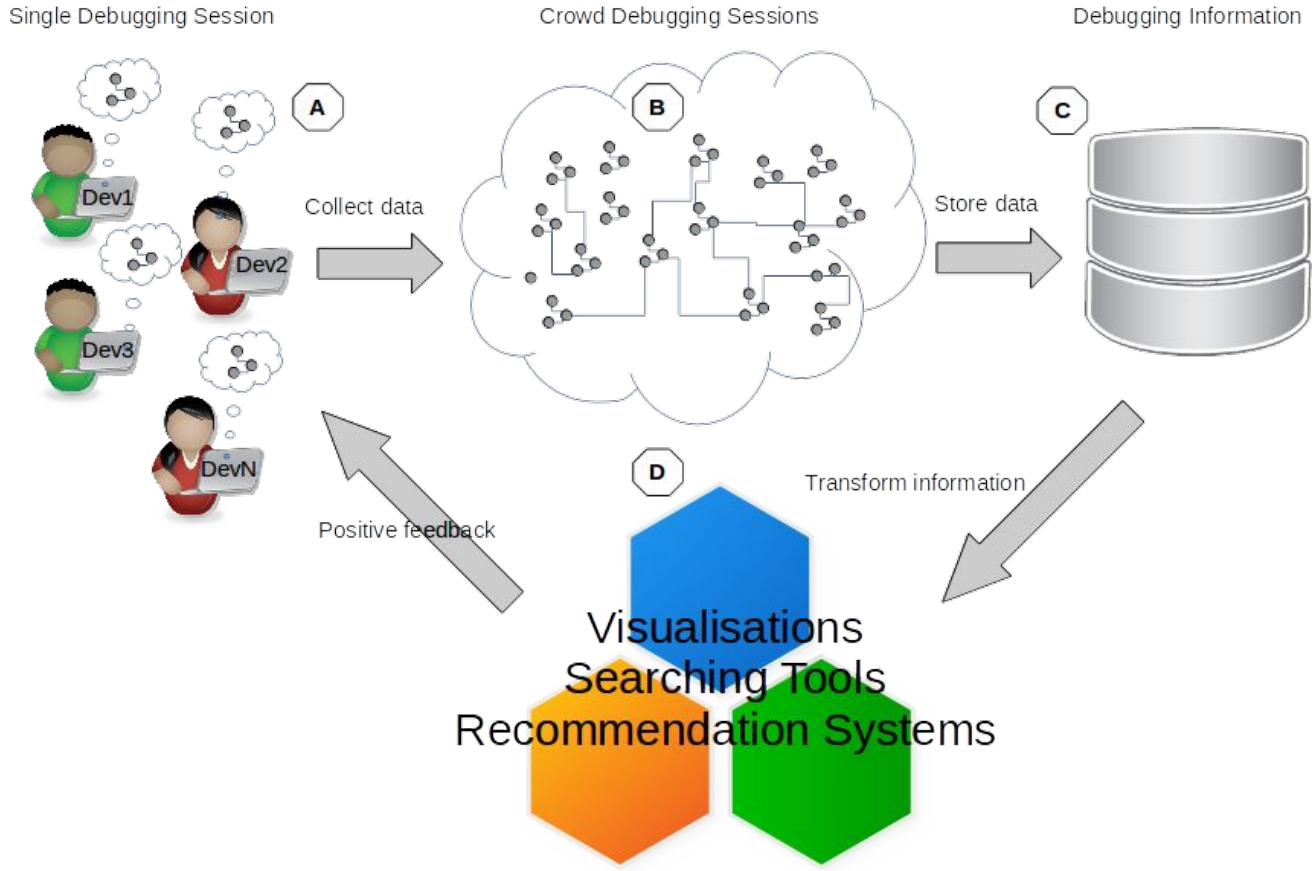
- Motivation
- Background and Related Work
- Swarm Debugging
- Swarm Debug Infrastructure
- Evaluation
- Conclusion and Future Work

Swarm Debugging

Swarm Debugging

- SD is a different way to doing debugging: original idea inspired on combination of IFT and swarm intelligence, providing **collective context** and data **automatically** captured to interactive debugging
- **Central idea: many developers on different tasks working independently make knowledge, creating a swarm intelligence environment.**
- IFT -> one prey/one predator
- Swarm Debugging -> **many prey/many predators**
- Explore breakpoints and debugging paths in a collective way
- Use previous data to support debugging tasks

Swarm Debugging



Swarm Debugging vs. Traditional Analysis

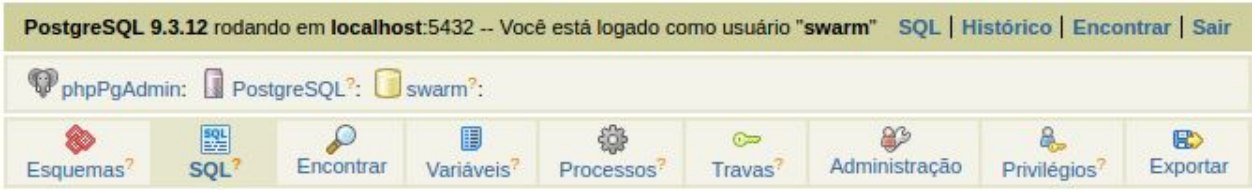
- Static analysis
  - examine a piece of code without running, using parsers.
  - Identify violations, metrics or structures (patterns)
- Tradicional Dynamic Analysis
  - all interactions, states and events are collected by tools
  - tracing all data without any developers' decision control or context
  - intrusive infrastructure to collect data
  - considered as a exceptional testing task, NOT a regular development tasks
  - Approach *collect-all-data-mining-after ,* typically generating a huge quantity of data
- **Swarm Debugging**
  - **Explore the gap -> static and dynamic analysis**
  - **Collect only paths intentionally explored by developers**
  - **Fundamental difference**
  - **Collecting methods invocations**

Outline

- Motivation
- Background and Related Work
- Swarm Debugging
- Swarm Debug Infrastructure
- Evaluation
- Conclusion and Future Work

Swarm Debug Infrastructure (SDI)

- SDI is an infrastructure to provide support to our approach
- Provides a set of tools to collect, store, share, retrieve and visualise interactive debugging sessions
- SDI has three main modules:
  - **Tracer**: listeners to **collect automatically** interactive debugging data
  - **Services**: servers to **store and share** debugging session data
  - **Views**: visualizations and searching tools



Informe o SQL a executar abaixo:

SQL

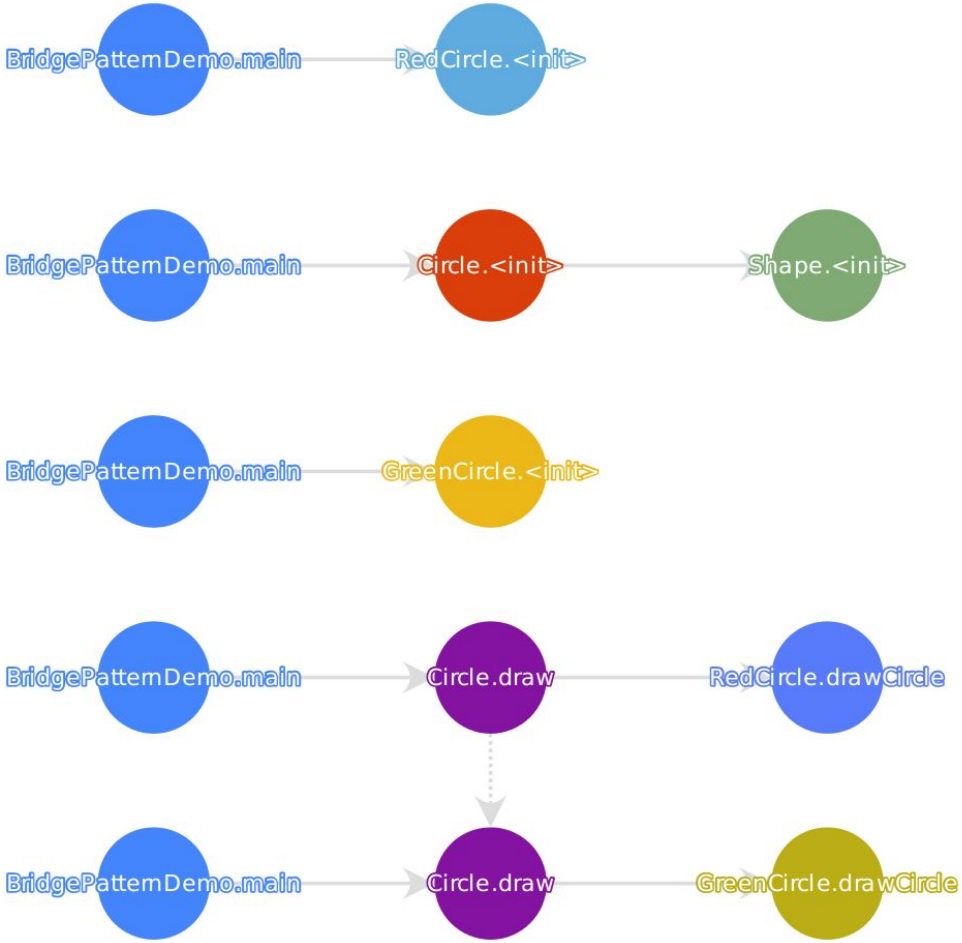
```
SELECT * FROM breakpoint WHERE type = 200
```

ou carregue o script SQL de um arquivo: Choose file No file chosen

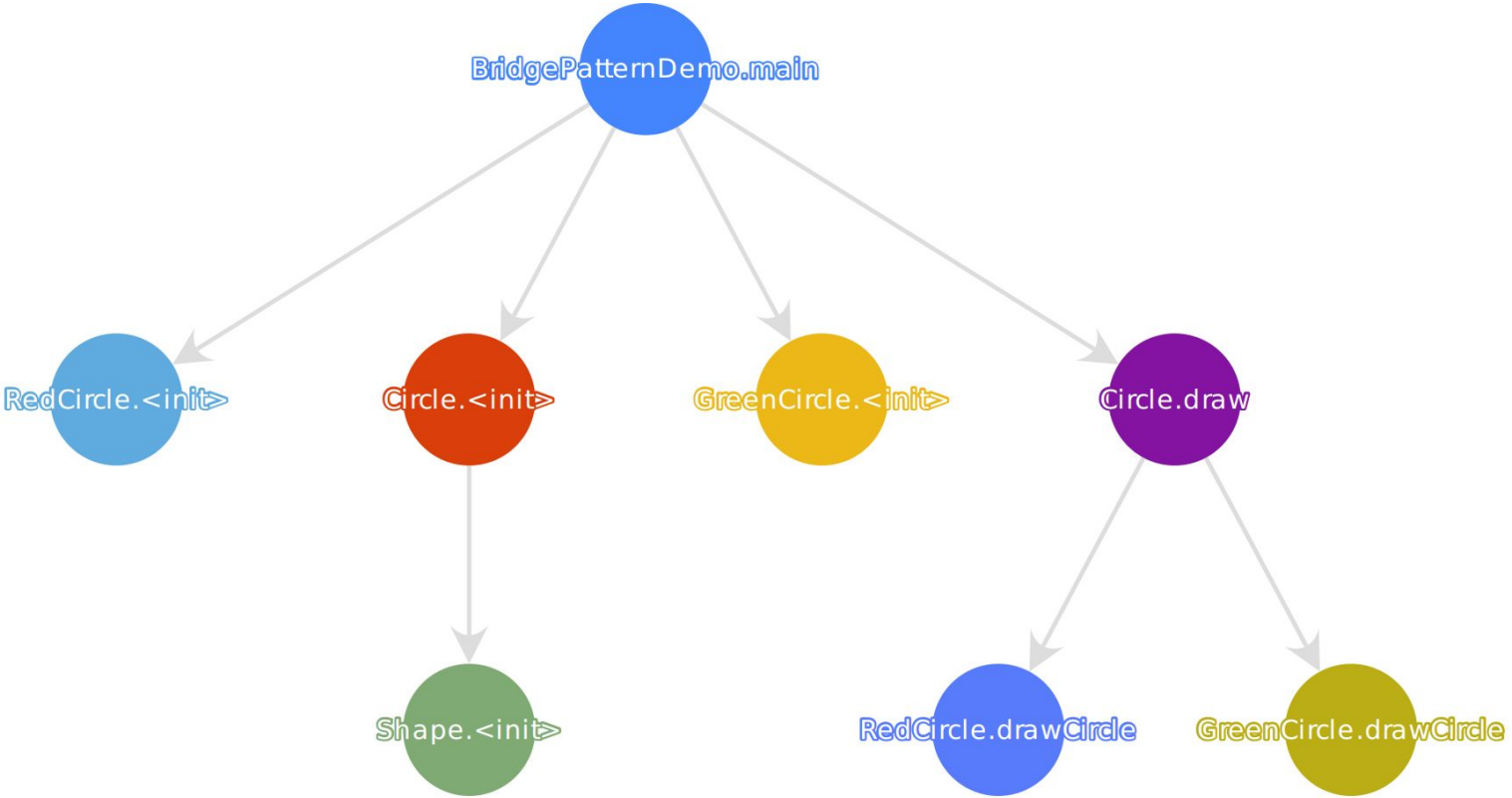
Paginar resultados

Executar Reiniciar

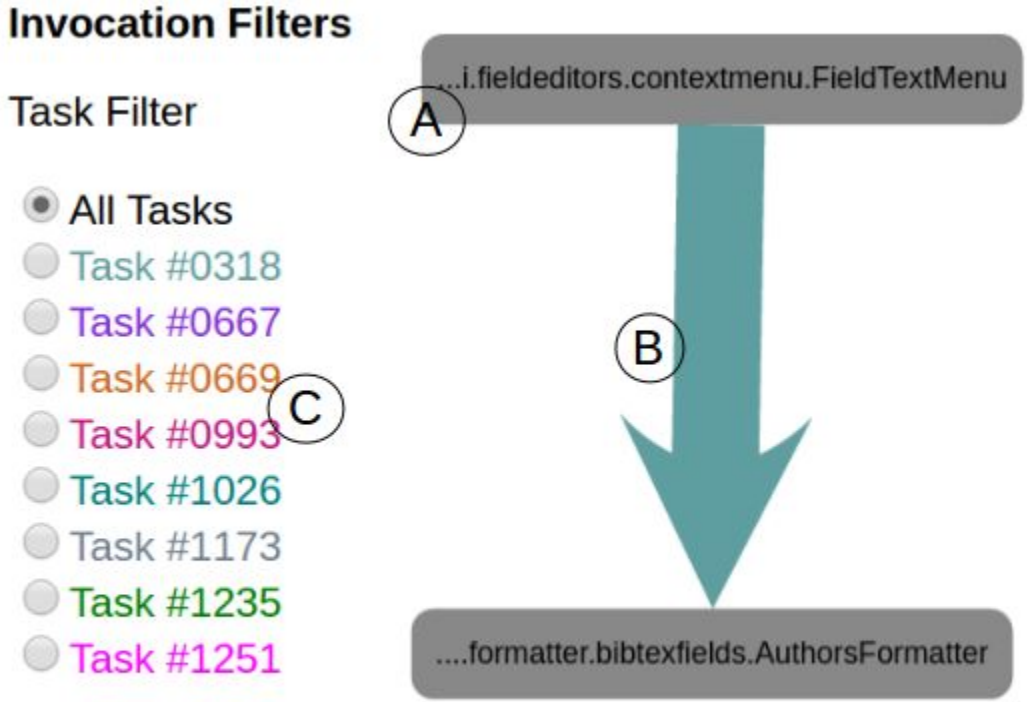
Sequence stack diagram



Dynamic method call graphs



Debug Global View

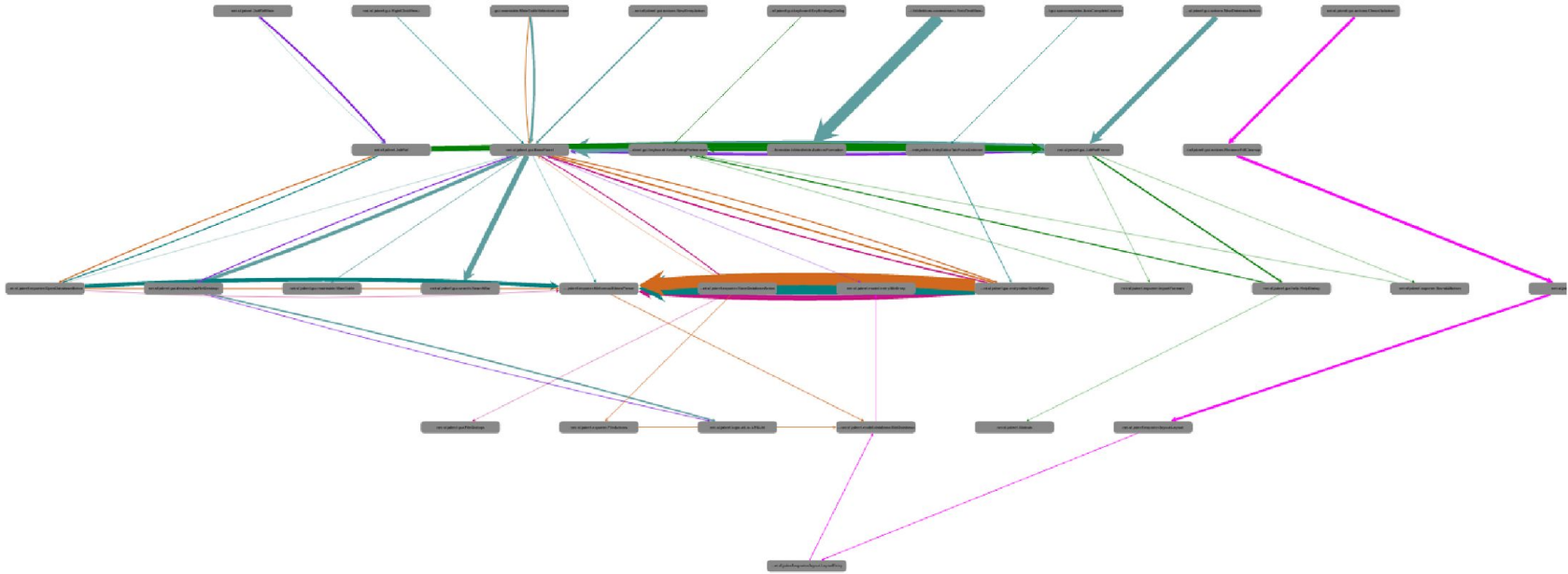


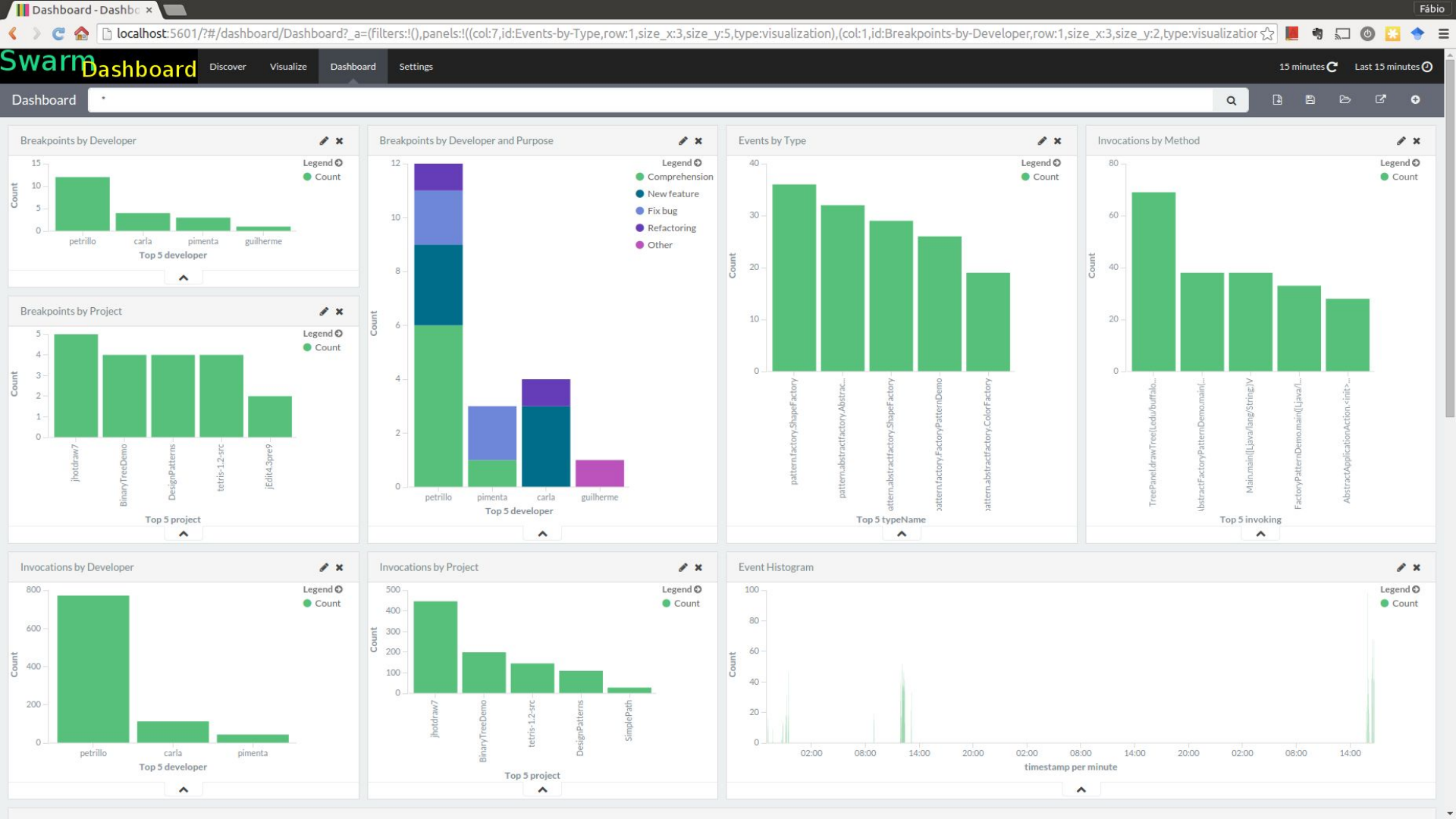
Debug Global View

Invocation Filters

Task Filter

- All Tasks
- Task #0318
- Task #0667
- Task #0669
- Task #0993
- Task #1026
- Task #1173
- Task #1235
- Task #1251




Dashboard - Dashbo
Fábio
localhost:5601/?#/dashboard/Dashboard?_a=(filters:!(),panels:!((col:7,id:Events-by-Type,row:1,size_x:3,size_y:5,type:visualization),(col:1,id:Breakpoints-by-Developer,row:1,size_x:3,size_x:3,size_y:2,type:visualizatior
Swarm Dashboard
Discover
Visualize
Dashboard
Settings
15 minutes
Last 15 minutes
Dashboard
*
Breakpoints by Developer
Legend
Count
Count
petrillo
carla
pimenta
guilherme
Top 5 developer
Breakpoints by Developer and Purpose
Legend
Comprehension
New feature
Fix bug
Refactoring
Other
Count
petrillo
pimenta
carla
guilherme
Top 5 developer
Events by Type
Legend
Count
Count
pattern.factory.ShapeFactory
pattern.abstractfactory.Abstrac...
pattern.abstractfactory.ShapeFactory
pattern.factory.FactoryPatternDemo
pattern.abstractfactory.ColorFactory
Top 5 typeName
Invocations by Method
Legend
Count
Count
TreePanel.drawTree(Ledu/buffalo...
Main.main([Ljava/lang/String;)V
FactoryPatternDemo.main([Ljava/l...
AbstractApplicationAction.<init>...
Top 5 invoking
Breakpoints by Project
Legend
Count
Count
jhotdraw7
BinaryTreeDemo
DesignPatterns
tetris-1.2-src
jEdit4.3pre9
Top 5 project
Invocations by Developer
Legend
Count
Count
petrillo
carla
pimenta
Top 5 developer
Invocations by Project
Legend
Count
Count
jhotdraw7
BinaryTreeDemo
tetris-1.2-src
DesignPatterns
SimplePath
Top 5 project
Event Histogram
Legend
Count
Count
02:00
08:00
14:00
20:00
02:00
08:00
14:00
20:00
02:00
08:00
14:00
timestamp per minute

Swarm Debug Infrastructure is Free!

- It is **free** and **open** research data
  - https://github.com/SwarmDebugging
  - http://server.swarmdebugging.org/

Outline

- Motivation
- Background and Related Work
- Swarm Debugging
- Swarm Debug Infrastructure
- Evaluation
- Conclusion and Future Work

Experimental Study

- Evaluation of the Swarm Debugging
- Three experiments
  - Experiment #1: towards understanding interactive debugging
  - Experiment #2: mining debugging data to recommend breakpoints
  - Experiment #3: supporting maintenance tasks using shared debugging visualisations
- Some interesting findings
- Also a contribution towards debugging phenomenon comprehension

General experimental details

- Participants: professional freelancers and students
- Target system: Open source project JabRef 3.2
- 5 **actual** bug location tasks: focus on breakpoints
  a. Breakpoints are essential for interactive debugging!!
- Warm-up task
- Video recording (screencast during sessions) and video analysis
- Using SDI to collect debugging session data
- Protocol:
  a. Profile survey
  b. Warm-up
  c. Task execution (bug location)
  d. Questionnaires: qualitative feedback
  e. Analysis

Main Results

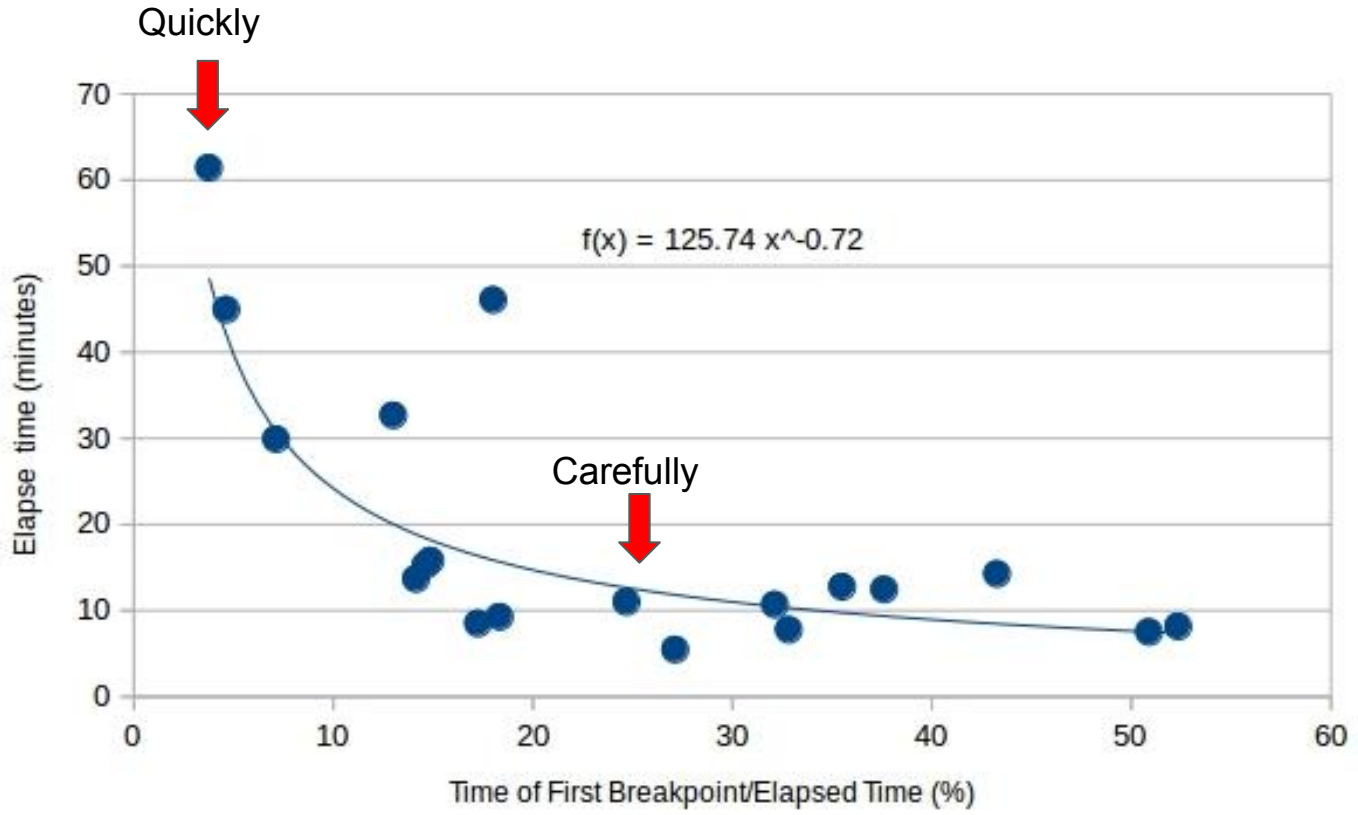
Main results

RQ: Is there a correlation between time of first breakpoint and task's elapsed time?

- There is a strong correlation ($\rho = -0.637$)

$$f(x) = \frac{\alpha}{x^{\beta}}$$

Main results



Main results

RQ:Is there a correlation between time of first breakpoint and task's elapsed time?

- whether developers toggle breakpoints carefully, they complete tasks faster than developers who toggle breakpoints too quickly

Main results

RQ: How much time do developers spend between **start** a debugging session and toggling the **first breakpoint**?

- In average, participants spent **27% of task time** to toggle the first breakpoint
- Toggling the first breakpoint is **not an easy task** and developers **need tools to assist them** in locating the places to toggle breakpoints.

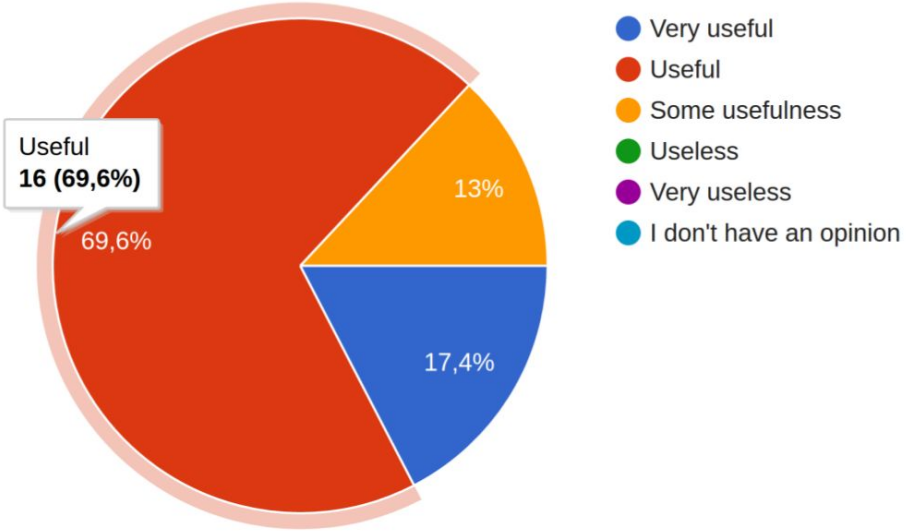
Main results

RQ: Do different developers toggle breakpoints at the same location (line of code) for the same task?

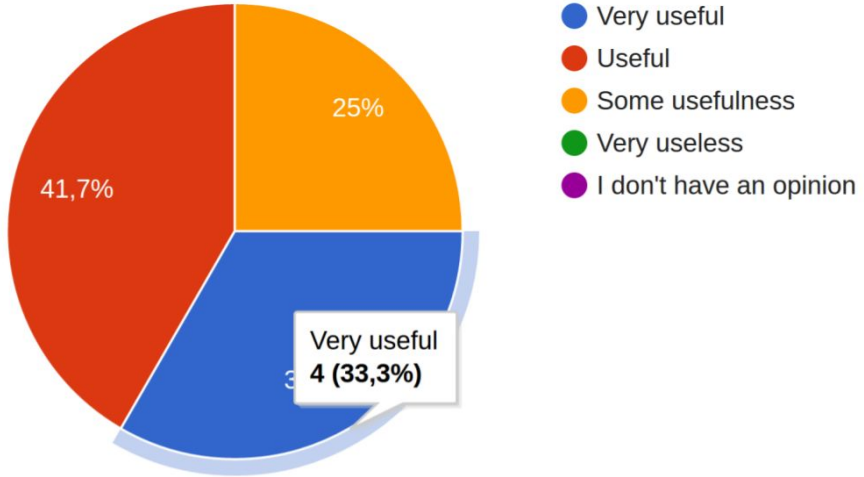
- Yes, 39 breakpoints out of 207 (near to 20%) were toggled in **exactly the same line of code** for the same task toggled by different developers
- **There is evidence of a rational choice of breakpoints**

Main results

RQ: Does **sharing and visualizing** debugging data support software maintenance tasks?



Phase 1



Phase 2

Outline

- Motivation
- Background and Related Work
- Swarm Debugging
- Swarm Debug Infrastructure
- Evaluation
- Conclusion and Future Work

Contributions

- SD, a new approach to collect, share and retrieve information from debugging sessions
- SDI, an infrastructure to support our approach, providing several visualisations and searching tools.

Good for...

- Developers
- Debugger’s developers
- Researchers
- Educators

Limitations

- Versioning
- Platform: Java, Eclipse
- Dependency on collecting
- Visualisation scaling

Future Work

- Versioning (evolution)
- SDI Tracer to new platforms
  - C++ (GDB - in progress)
  - Python (PyDev - in progress)
  - Javascript (Firebug - in progress)
  - PHP, Ruby, .Net
  - Intellij, Netbeans, Pharo (in progress)
- Mylyn integration
- New recommendation systems
- Improve and create visualisations
- New controlled experiments
- Foragers and Builders approach (new applied project)

Publications

- Directly related to Thesis: **nine** submitted papers
  - SANER 2017 (submitted) Today!! :-)
  - IEEE Software - Special Issue - Crowdsourcing on SE (revision - 2nd round)
  - QRS 2016 (accepted)
  - ICPC 2016 (accepted)
  - VISSOFT 2015 (accepted)
  - VEM 2015 (accepted)
  - ICSME 2016 (rejected)
  - VISSOFT 2016 (rejected)
  - ICSME 2015 (rejected)
- During PhD
  - SAC 2017 (Cloud Lexicon - submitted)
  - WBMA 2016 (Kanban - accepted)
  - ICSOC 2016 (Cloud REST API - accepted)
  - GAS 2016 (Video game dev. process - accepted)
  - VEM 2014 (Polymorphism - accepted)
  - CibSE 2012 (Software Visualisation - accepted)
  - IHC 2011 (Likert Scale Visualisation - accepted)

Final Remarks

- Swam Debugging is an approach uses Swarm Intelligence and Information Foraging Theory to provide knowledge from interactive debugging session information
- Swarm Debug Infrastructure (SDI) provides an infrastructure to create tools for context-aware debugging
- Improving debugging phenomenon comprehension:
  - When developers toggle breakpoints carefully, they complete tasks faster than developers who toggle breakpoints too quickly
  - There is evidence of a rational choice of breakpoints
- 75% of developers claim that visualise **shared debugging data** is useful or very **useful** on supporting maintenance tasks
- Many open questions on interactive debugging….

Universidade Federal do Rio Grande do Sul
Programa de Pós-Graduação em Computação

Swarm Debugging

The Collective Debugging Intelligence of the Crowd

Fábio Petrillo

Thanks a lot!!
Questions????

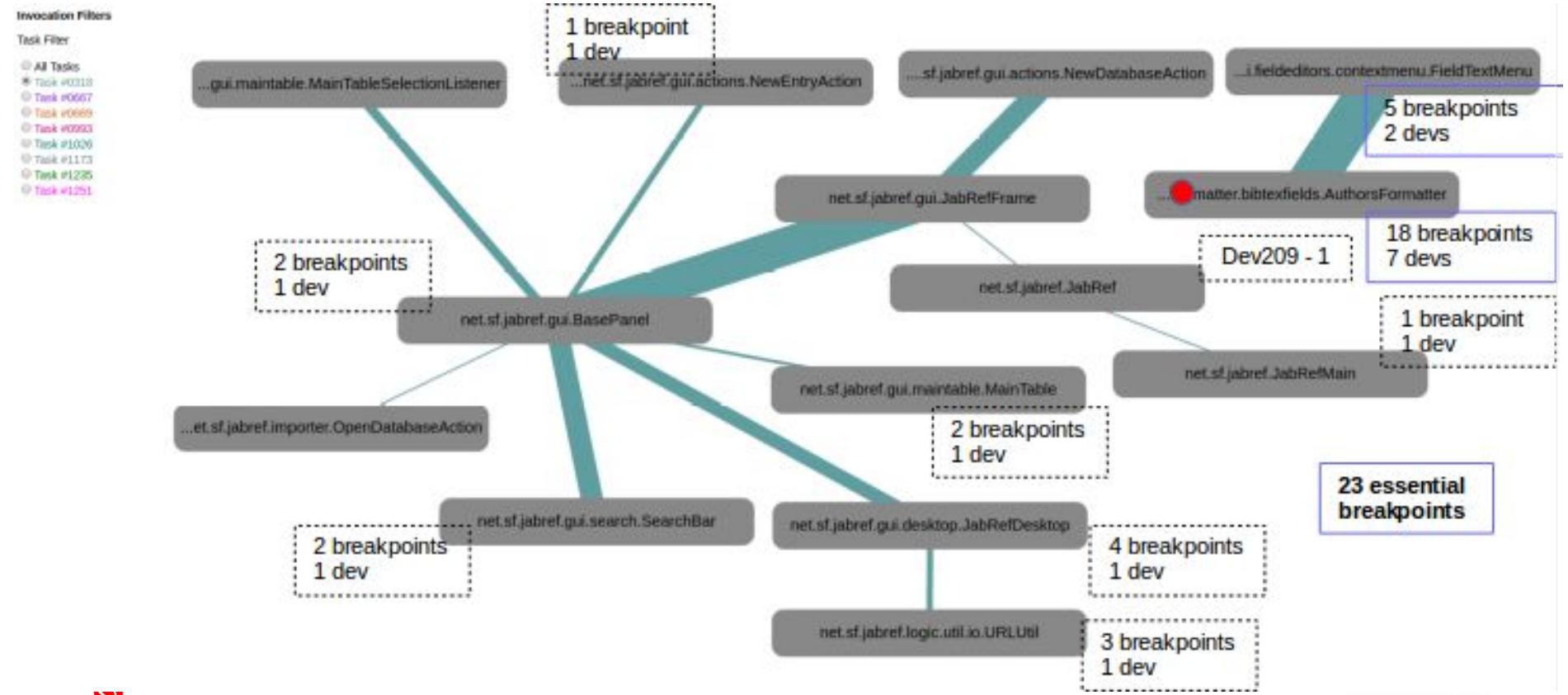
Porto Alegre, 19 outubro de 2016

Main results (not in thesis text)

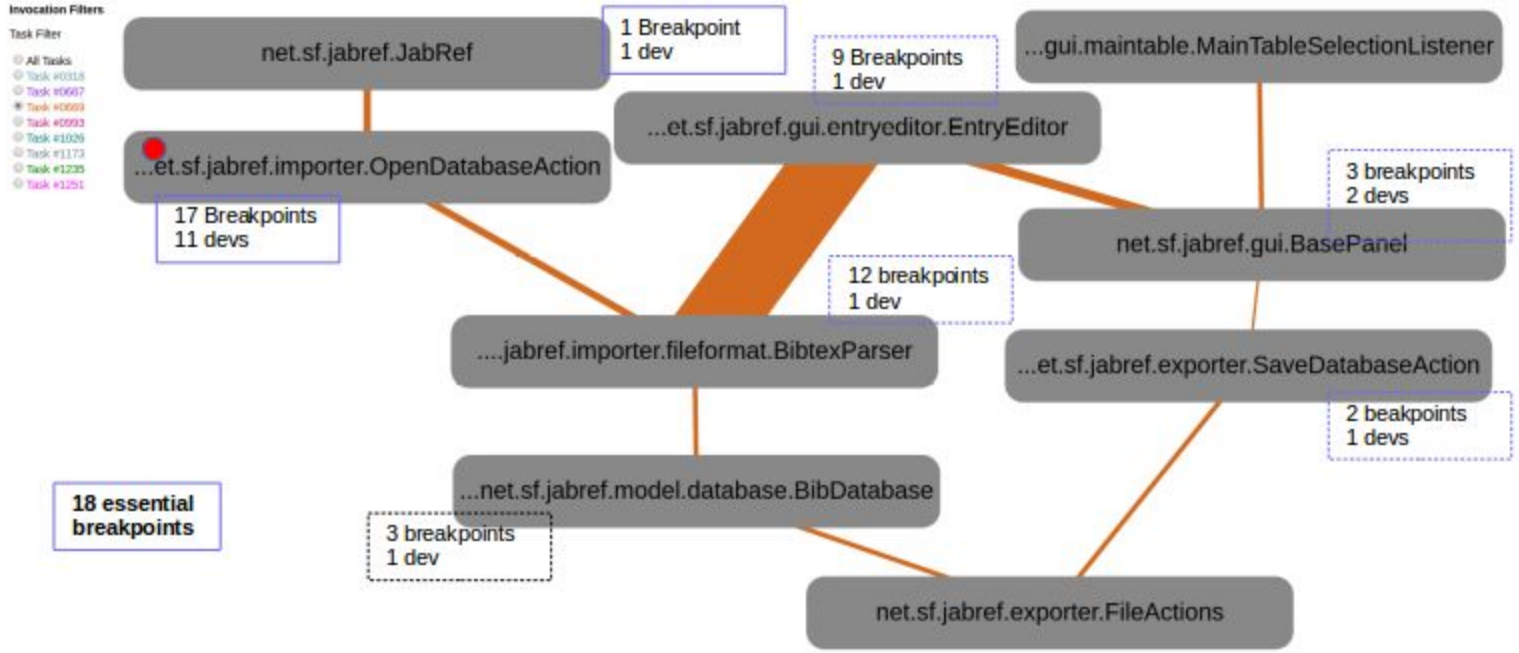
RQ: How many **essential breakpoints** are toggled by developers on a task?

- essential breakpoint: breakpoint toggled on path of the fault

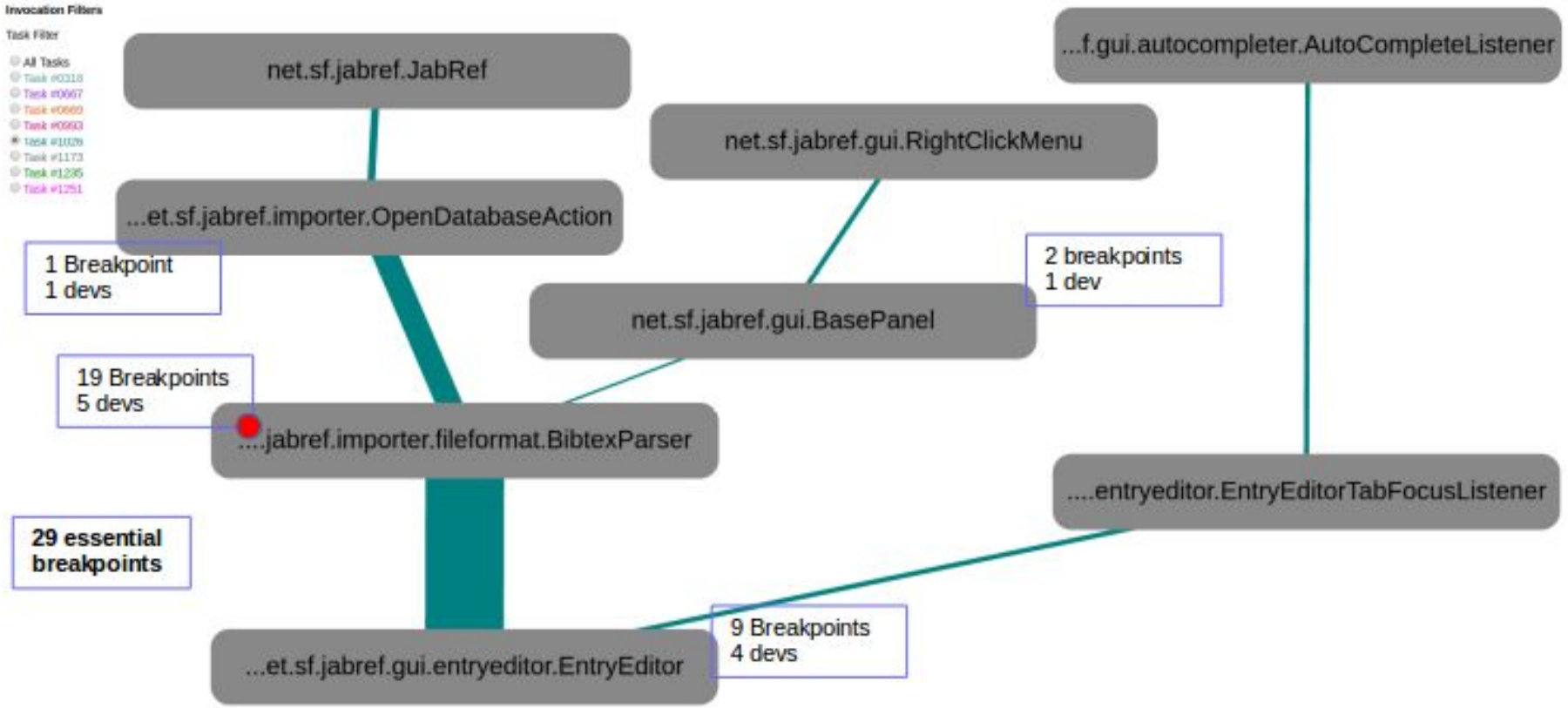
Global View - Task #0318



Global View - Task #0669



Global View - Task #1026



Main results (not in thesis text)

How many essential breakpoints are toggled by developers on a task?

We found 118 essential breakpoints toggled by developers, or 57% (118/207) of breakpoints are essential to achieve the fault.

Comentários dos revisores (IEEE Software)

"All of the reviewers agreed that the paper is interesting and promising, and is well within the scope of crowdsourcing." Issue's Editor

"I appreciate this work, very much. The direction is innovative, fun (in a way), and challenging. This is a nice research problem to be working on, and I encourage the authors to continue." Reviewer #1

"This manuscript addresses an important problem with a thought-provoking metaphor and solution." Reviewer #2

"The authors propose a very interesting idea for software debugging. I encourage the authors to further push forward to identify and solve the key problems on this topic." Reviewer #3

Comentários dos revisores (VISSOFT 2015)

“The authors identify a pain point in software engineering: debugging is a human activity performed individually by developers, and these developers accumulate knowledge that is either lost or simply not easily shared between developers on the same project."

“An important problem is being addressed. Good use of collective intelligence. Builds on prior work.”

"Debugging is a specific and distinct enough activity that specific exploration and support of the topic is worth exploring.”

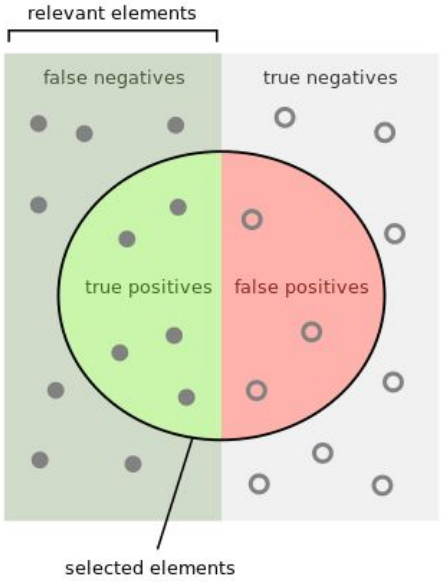
Co-breakpoint

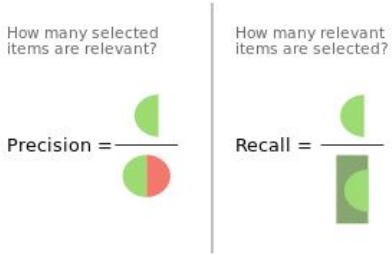
- Recommendation system example

Table 4.2: Approach for breakpoint prediction

Tasks	Co-breakpoint Database (DB)	Recommendation Candidates
T_1	$\{\}$	$(e_1, \varnothing) - (e_2, \varnothing) - (e_3, \varnothing)$
T_2	$\{\{e_1, e_2, e_3\}\}$	$(e_4, \varnothing) - (e_5, \varnothing) - (e_2, \{e_1, e_3\}) - (e_6, \varnothing)$
T_3	$\{\{e_1, e_2, e_3\}, \{e_4, e_5, e_2, e_6\}\}$	$(e_1, \{e_2, e_3\}) - (e_7, \varnothing)$
T_4	$\{\{e_1, e_2, e_3\}, \{e_4, e_5, e_2, e_6\}, \{e_1, e_7\}\}$	$(e_2, \{e_1, e_3, e_4, e_5, e_6\}) - (e_5, \{e_4, e_6\}) - (e_3, \{e_1\})$

Precision and Recall





Precision: how many selected item are relevant?

Recall: how many relevant items are selected?

https://en.wikipedia.org/wiki/Precision_and_recall

Static Analysis

- Static analysis examine a piece of code without running, using parsers.
- Identify violations, metrics or structures (patterns)

Dynamic Analysis

- Running program
  - behavior
  - instrumentation
- Benefits
  - precision
  - context/scenario
- Limitations
  - small fraction of execution
  - difficulty to determinate scenarios
  - scalability
-
- Identify violations, metrics or structures (patterns)

Pygmalion effect

- High leader expectation increase follow performance
- If I pay, freelancers “increase” responses to thank the researchers

Use of IDEs (GU, 2012)

- .NET - 97% - Visual Studio
- Java - 73% - Eclipse

Information Foraging Theory

- In another study, Kuttal et al. (KUTTAL; SARMA; ROTHERMEL, 2013) showed that the stronger scents available within mashup programming environments could im prove users' foraging success, leading to a new model for debugging activities framed concerning information foraging theory to support debugging.
- Fleming et al. (FLEMING et al., 2013) without environment support, foraging during debugging may be tedious and costly, and in IFT terms, setting breakpoints enriches the environment by creating low-cost links.

Data Frugality (Fowler, 2016)

- Handle, capture and store only data that we need.

Collective Intelligence

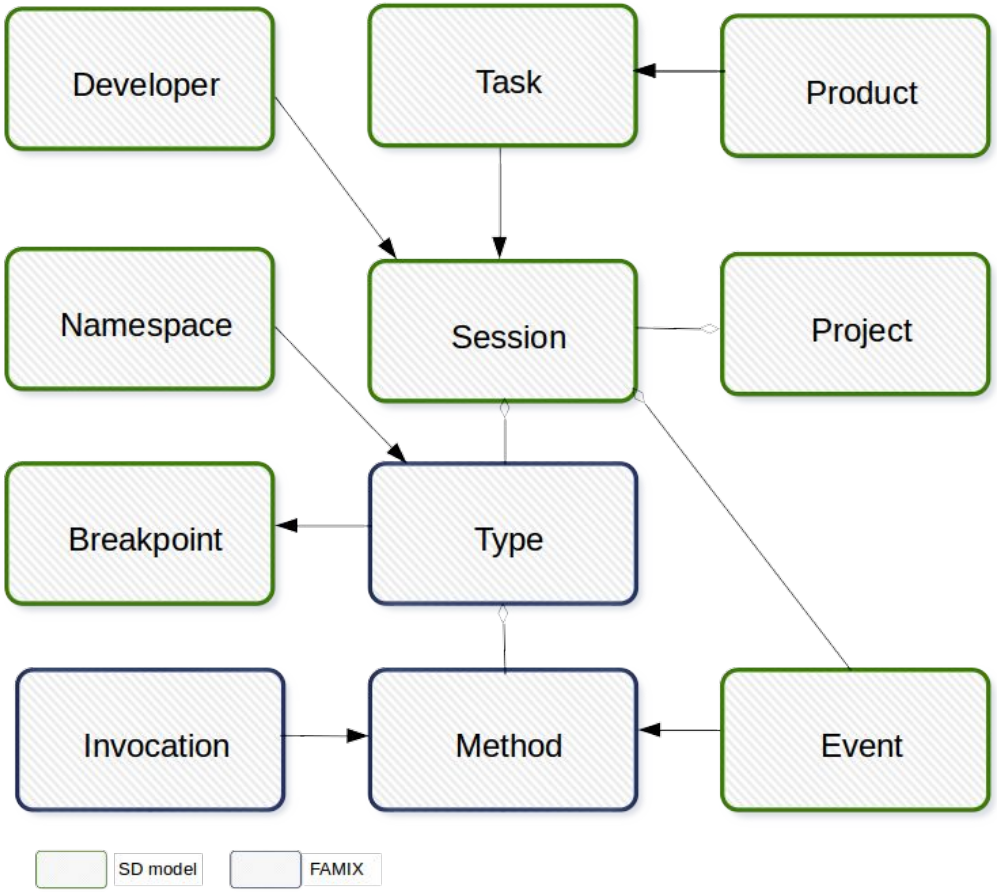
- Integrated Development Environments (IDEs) only integrate the tools and knowledge of a single user and workstation. (BRUCH et al.,2010)
- After they found an answer, the newly gained knowledgeis usually lost inside the IDEs.
- IDE provides a rich source of information that can help ourselves and other programmers to avoid mistakes in the future
- collective intelligence is an open-field for new software development tools (STOREY et al., 2014)
- new developers expect collaborations
- Rise of social programmers
  - cooperate in on-line communities open to contributes
- They are opened, transparent, and expect to share their knowledge

Automated debugging criticism

- Navigation pattern was not linear
- Give a statement is not enough!
- Automated debugging tools benefit
  - point developers in the right direction
  - not the exactly fault statement
- Devs want alternative views
- Different ways of interactions
- Seamlessly integrated activities

Parnin, C., & Orso, A. (2011). Are automated debugging techniques actually helping programmers? *Proceedings of the 2011 International Symposium on Software Testing and Analysis ISSTA 11*, 199.

Swarm Debugging Meta-model



Swarm Restful API

- Implemented on Spring Boot
- Operations
  - Create
  - Retrieve
  - Update
  - Delete

Reopen and Next bugs

- Developers spend significant time looking for similar bugs that have been resolved in the past (BUGDE et al., 2008)
- Developers benefit from knowing this previous bug data
- About 20-40% of total fixing changes appear repeatedly (KIM; PAN; WHITEHEAD, 2006).
- Next Bug (2016)
  - 67% of Mozilla developers in the field study indicated interest in a Bugzilla extension with recommendations
- Finding and ranking relevant classes is a current/open research topic

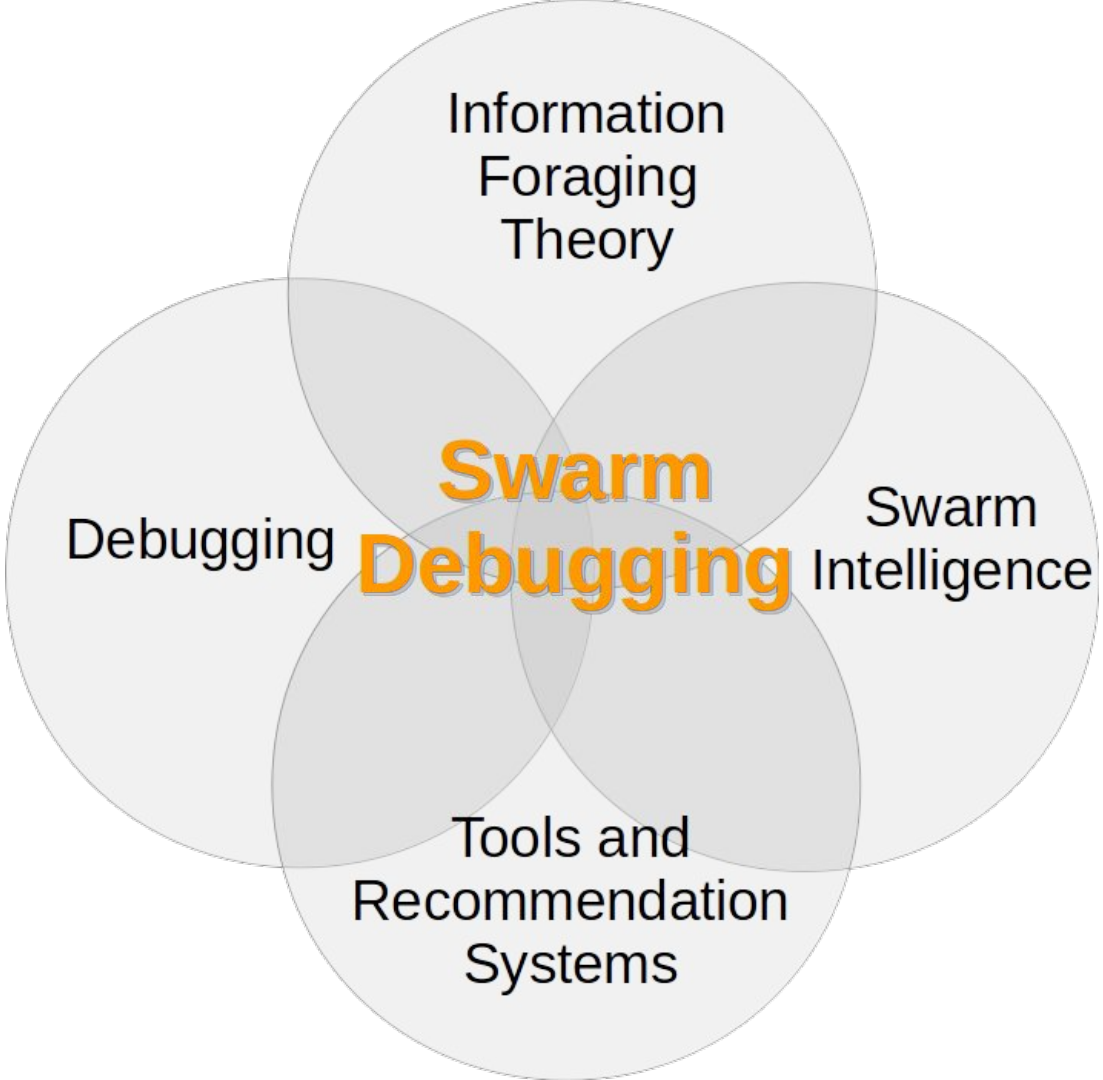
Comentários dos revisores (ICSME 2015)

"The idea itself is definitely promising and I would like to encourage more research in the area, but this work is still in a very early stage."

Debugging is hard

"We observed that performing path simulation **manually** was nearly **impossible** for statements with high branching factors as there were simply too many paths to consider".

LaToza, T. D., & Myers, B. a. (2010). Developers ask reachability questions. *2010 ACM/IEEE 32nd International Conference on Software Engineering*, *1*, 185–194.


Information Foraging Theory
Debugging
Swarm Debugging
Swarm Intelligence
Tools and Recommendation Systems

Crowd in SE

- A new tendence of collaboration in SE
- Large group of individuals together
- In SE -> fluid workforces automatically arranged by the environment to perform micro tasks within a workflow (LATOZA; HOEK, 2015)
- What is the difference?
    - Awareness of influence of own activities in someone else's activities

Maintaining software is currently an incredibly complex activity

(BOOCH, 2015)

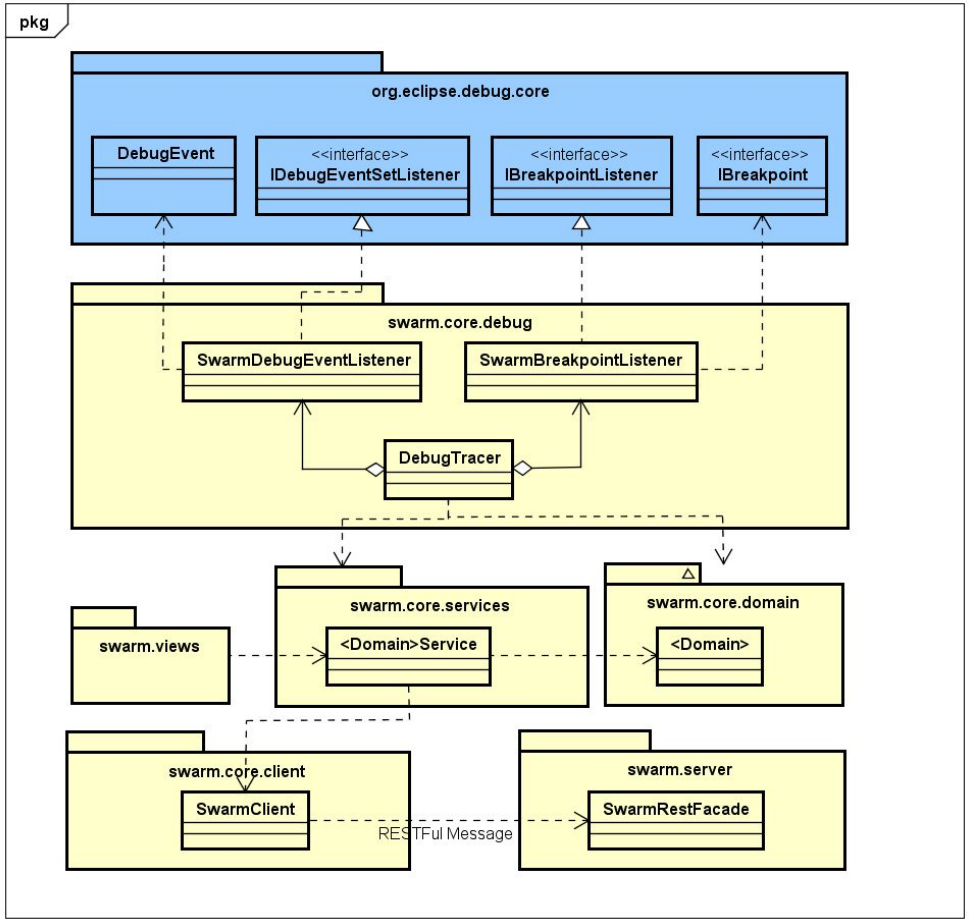
What is lacking?

- Static analysis is not able to achieve all software paths correctly
- Tradicional dynamic analysis
  - high instrumentation effort
  - collect irrelevant data
  - out of everyday activity
- Debugging is seen as an isolated activity

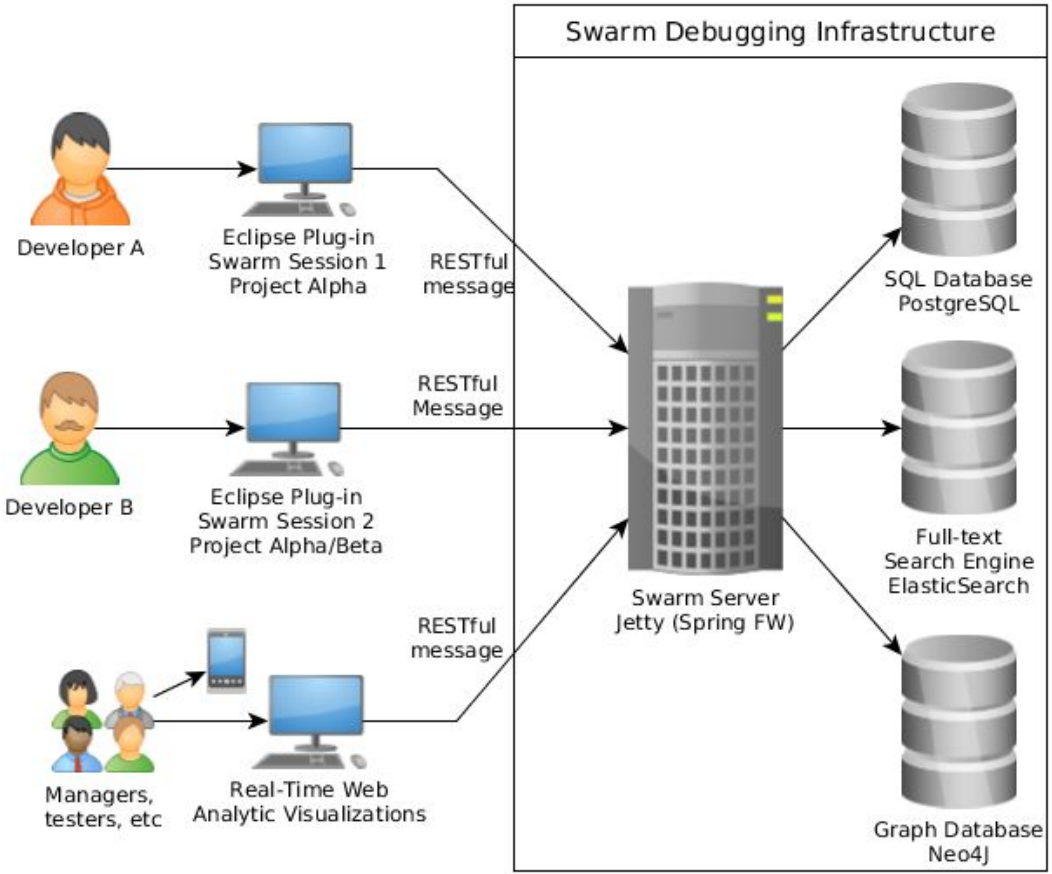
Collaborations

- École Poly de Montréal - Prof. Yann-Gael Gueheneuc, Prof. Foutse Kohm
- UQAM - Profa. Naouel Moha
- INRIA/Lille - Dr. Phillipe Merle
- UFMG - Prof. Marco Túlio Valente, Prof. Guilherme Avelino
- Drew University - Profa. Emily Hill
- UFSM - Sr. Cristiano Politowski (Master student)
- Serpro/UECE - Sr. Francisco Lopes (Master student)
- UniRitter - Prof. Guilherme Lacerda
- UFRGS - Sr. Gabriel Veras (Master student)
- UERN - Profa. Carla Monteiro

Swarm Debug Tracer



Swarm Debug Infrastructure



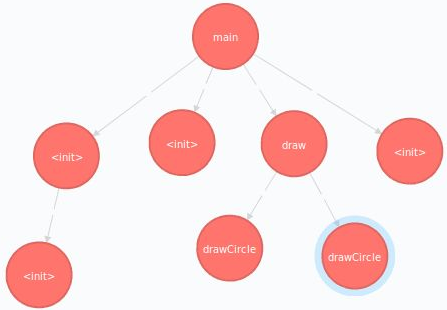
Neo4J Browser

```
$ MATCH p=(m:Method)-[i:INVOKES*]->(n:Method) where m.session = 21951 RETURN p
```

$ MATCH p=(m:Method)-[i:INVOKES*]->(n:Method) where m.session = 21951 RETURN p

Graph
Rows

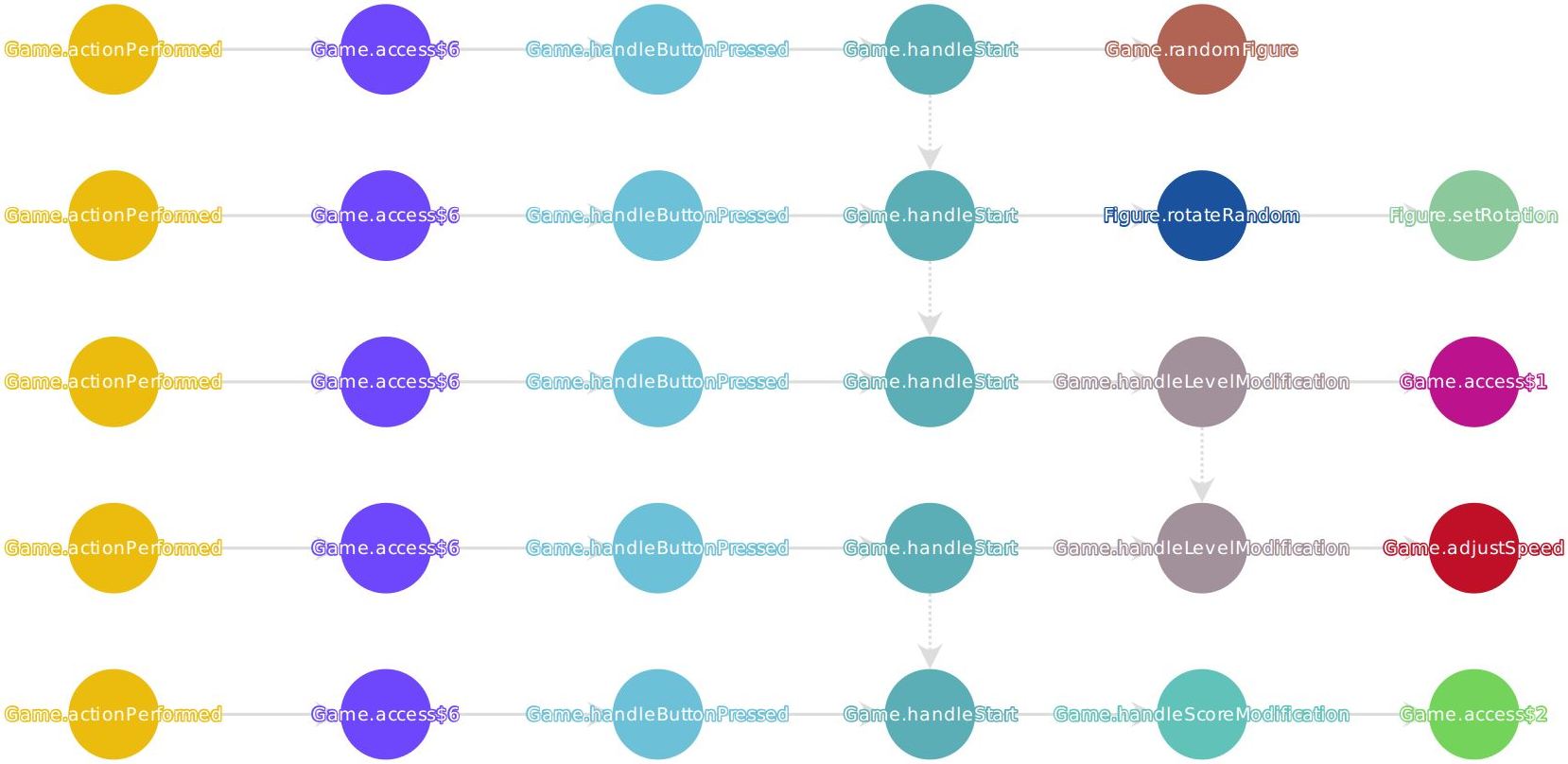
*(8) Method(8)
*(7) INVOKES(7)



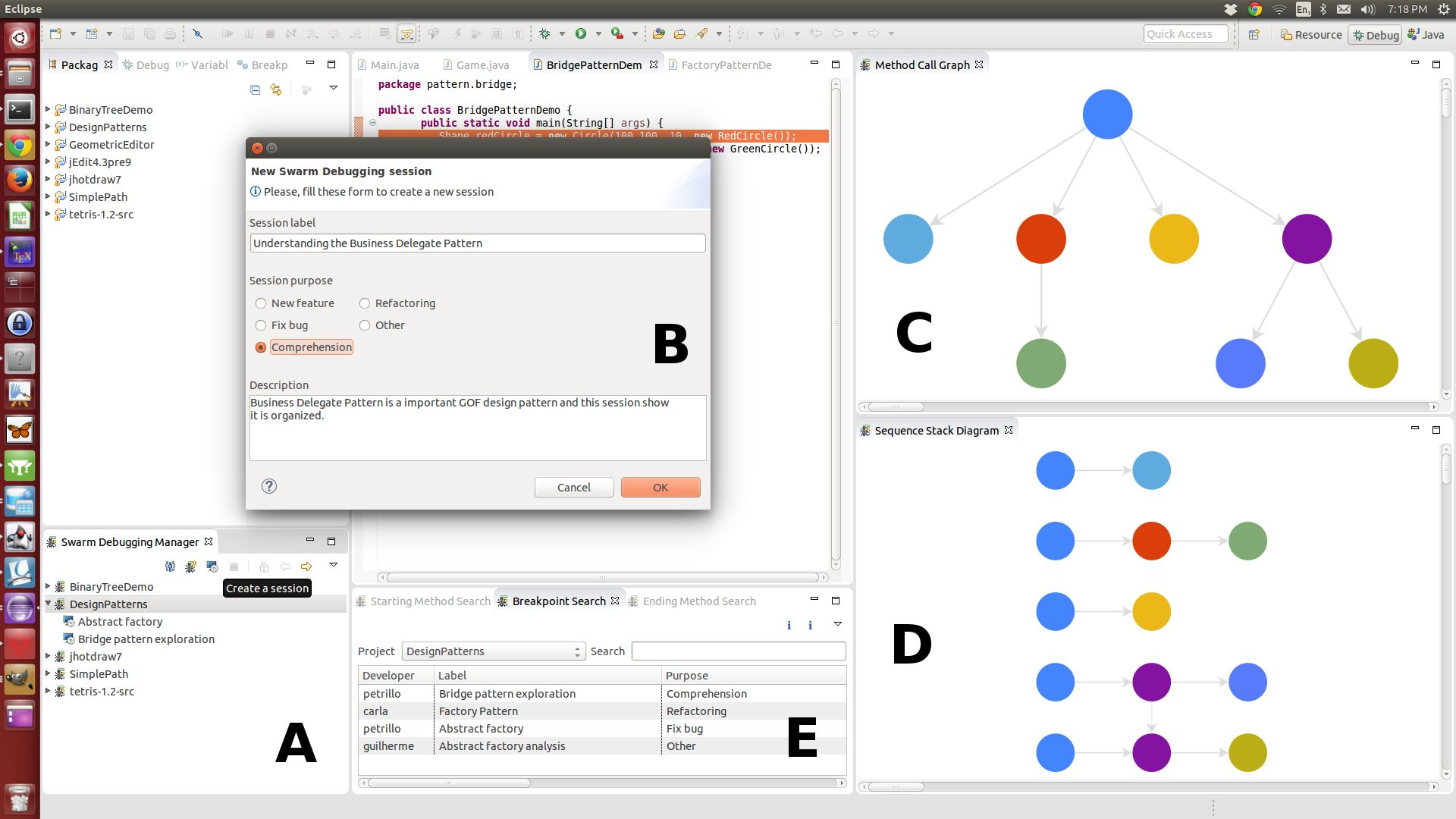
Method **id:** 22008 **name:** drawCircle **key:** Lpattern.bridge.RedCircle;.drawCircle(III)V **typeName:** pattern.bridge.RedCircle **session:** 21951 **developer:** petrillo

Java - Eclipse Platform

Sequence Stack Diagram



SDI in action


Eclipse
7:18 PM
Quick Access
Resource
Debug
Java
Packag
Debug
Variabl
Breakp
BinaryTreeDemo
DesignPatterns
GeometricEditor
jEdit4.3pre9
jhotdraw7
SimplePath
tetris-1.2-src
Main.java
Game.java
BridgePatternDem
FactoryPatternDe
package pattern.bridge;
public class BridgePatternDemo {
public static void main(String[] args) {
Shape redCircle = new Circle(100,100, 10, new RedCircle());
ew GreenCircle());
New Swarm Debugging session
Please, fill these form to create a new session
Session label
Understanding the Business Delegate Pattern
Session purpose
New feature
Refactoring
Fix bug
Other
Comprehension
B
Description
Business Delegate Pattern is a important GOF design pattern and this session show it is organized.
Cancel
OK
Method Call Graph
C
Sequence Stack Diagram
D
Swarm Debugging Manager
BinaryTreeDemo
Create a session
DesignPatterns
Abstract factory
Bridge pattern exploration
jhotdraw7
SimplePath
tetris-1.2-src
A
Starting Method Search
Breakpoint Search
Ending Method Search
Project
DesignPatterns
Search
Developer
Label
Purpose
petrillo
Bridge pattern exploration
Comprehension
carla
Factory Pattern
Refactoring
petrillo
Abstract factory
Fix bug
guilherme
Abstract factory analysis
Other
E