Live Trace Visualization for System and Program Comprehension in Large Software Landscapes

PhD Thesis Defense
Kiel University, Software Engineering Group
Florian Fittkau — November 30, 2015
Motivation

Live trace visualization of large software landscapes for comprehension of systems and applications

Selected Challenges:
- Possible huge monitoring data amount (performance/cost efficiency)
- Finding abstractions to understand huge landscapes but also application-level details
- Live visualization of thousands or even millions of traces

[Fittkau et al. 2013a, Fittkau et al. 2015b]
Major Scientific Contributions

- SC1: An approach named ExplorViz for **enabling live trace visualization** of large software landscapes

- SC2: A **monitoring and analysis approach** capable of logging and processing the huge amount of conducted method calls in large software landscapes

- SC3: **Display and interaction concepts** for the software city metaphor beyond classical 2D displays and 2D pointing devices
SC1 – ExplorViz Approach
The ExplorViz Method

Legend
A1: Monitoring
A2: Preprocessing
A3: Aggregation
A4: Transformation
A5: Navigation

[Fittkau et al. 2013a]
Monitoring & Trace Processing

Monitoring

Analysis

[Fittkau et al. 2013b, Fittkau et al. 2015c, Beye 2013, Matthiessen 2014, Weißenfels 2014]
Monitoring & Trace Processing

Kieker 1.8 with Analysis: 0.5k traces per second
ExplorViz with Analysis: 112.6k traces per second (limited by network speed)
Speedup of about factor 250

Kieker 1.8:
16.6k traces per second
ExplorViz:
141.2k traces per second
Speedup of about factor 9

[Fittkau et al. 2013b]

[Waller et al. 2014]
Landscape Perspective

[ExplorViz]

Landscape-Level Perspective

[Fittkau et al. 2013a, Fittkau et al. 2015b]
Landscape Perspective

VERSUS

Visualization Capabilities:

- Usable for system comprehension
- Abstractions provide a valuable addition

29 subjects solved system comprehension tasks

140 applications

Table:

<table>
<thead>
<tr>
<th></th>
<th>Time Spent</th>
<th>Correctness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Flat</td>
<td>Hierarchical</td>
</tr>
<tr>
<td>mean</td>
<td>23.49</td>
<td>23.45</td>
</tr>
<tr>
<td>difference</td>
<td>-0.17%</td>
<td></td>
</tr>
<tr>
<td>sd</td>
<td>3.87</td>
<td>3.29</td>
</tr>
<tr>
<td>min</td>
<td>15.03</td>
<td>15.93</td>
</tr>
<tr>
<td>median</td>
<td>24.64</td>
<td>23.14</td>
</tr>
<tr>
<td>max</td>
<td>29.68</td>
<td>33.16</td>
</tr>
</tbody>
</table>

Shapiro-Wilk W: 0.9232 vs. 0.9605
Levene F: 2.1048 vs. 1.2307
Student’s t-test: df = 27, t = 2.24105, p-value = 0.02303

[Fittkau et al. 2015d] Best Paper Award
Landscape Meta-Model

[Fittkau et al. 2015b]
Application Perspective

Application-Level Perspective

[Fittkau et al. 2013a]
Selected ExplorViz Features

**Clustering**

**Performance Analysis**

**Architecture Modeling & Conformance Checking**

[Fittkau et al. 2014b, Simolka 2015]

**Trace Replayer**

**Database Monitoring**

[Barzel 2014]

[Finke 2014]

[Jähde 2015]

[Zirkelbach 2015]
Application Perspective

First Controlled Experiment
Visualization Capabilities:

- Usable for solving program comprehension tasks
- Interactive concept for method calls works
- More efficient and effective than competitor (for the chosen tasks)

[Fittkau et al. 2015a]
SC2 – Elastic Monitoring and Analysis Approach
Scalable Trace Processing

[Fittkau et al. 2014b, Fittkau et al. 2015c, Koppenhagen 2013, Stelzer 2014]
Scalable Trace Processing

Processing Capabilities:

- Cost efficient
- Scalable to millions of monitored methods

[Fittkau et al. 2014b, Fittkau et al. 2015c, Stelzer 2014]
SC3 – New Display and Interaction Concepts
Physical 3D Models

[Fittkau et al. 2015e]
Physical 3D Models

112 participants solved program comprehension tasks in pairs

[Fittkau et al. 2015e]
Virtual Reality

[Fittkau et al. 2015f, Krause 2015]
Related Work & Outlook
Closely Related Work

Software Visualizations

APM Tools

Monitoring

[van Hoorn et al. 2012]
[Souza et al. 2012]
[Alam and Dugerdil 2007]
[Greevy et al. 2006]

Panas et al. 2003
Wettel and Lanza 2007
www.appdynamics.com
[www.appdynamics.com]
Publications & Theses

21 publications in 3.5 years
(15 peer-reviewed)

Selected Publications

- **VISSOFT 2015**: 3 papers (Acceptance Rate: 43% and 30%) and Best Paper Award
- **ESOCC 2015**: Fittkau and Hasselbring, *Elastic Application-Level Monitoring for Large Software Landscapes in the Cloud* (Acceptance Rate: 37.5%)
- **ICPC 2015**: Fittkau, Finke, Hasselbring, and Waller, *Comparing Trace Visualizations for Program Comprehension through Controlled Experiments* (Acceptance Rate: 31.5%)
- **ECIS 2015**: Fittkau, Roth, and Hasselbring, *ExplorViz: Visual Runtime Behavior Analysis of Enterprise Application Landscapes* (Acceptance Rate: 31%)
- **VISSOFT 2013**: Fittkau, Waller, Wulf, and Hasselbring, *Live Trace Visualization for Comprehending Large Software Landscapes: The ExplorViz Approach*
- **ICSE 2013**: Frey, Fittkau, and Hasselbring, *Search-Based Genetic Optimization for Deployment and Reconfiguration of Software in the Cloud* (Acceptance Rate: 18.5%)

15 Students’ Theses in the Context of ExplorViz

Outlook & Future Work

Live trace visualization for large software landscapes available as open-source software (Apache License 2.0)

All results available online
- Raw results, R scripts, code, ratings, ...
- ExplorViz versions used in the experiments
- All screen and camera recordings about 160 hours material
- Long-time archival on Zenodo.org

Future Work:
- More controlled experiments (e.g., comparison with more visualization metaphors; professionals as subjects)
- Layout of the Application-Level Perspective [Barbie 2014]
Bibliography


Bibliography (cont’d)


[Stelzer 2014] P. Stelzer. Scalable and Live Trace Processing in the Cloud, Bachelor thesis, Kiel University


Backup Slides
<table>
<thead>
<tr>
<th>ID</th>
<th>Category</th>
<th>Description</th>
<th>Score</th>
</tr>
</thead>
</table>
| T1 | A{4,8}  | *Context: Identifying refactoring opportunities*  
Name three classes (from different packages) that have high fan-in (at least 4 incoming communications) and almost no fan-out (outgoing communication). | 3    |
| T2.1 | A{3,4,5} | *Context: Understanding the checking process*  
Write down all constructor/method calls between RuleChain and JavaRuleChainVisitor. | 3    |
| T2.2 | A{1,2,5,6} | *Context: Understanding the violation reporting process*  
In general terms, describe the lifecycle of GodClassRule: Who creates it, what does it do (on a high level)? | 3    |
| T3.1 | A{1,5}  | *Context: Understanding the violation reporting process*  
Which rules are violated by the input file using the design rule set? Hint: Due to dynamic analysis the violation object is created only for those cases. | 2    |
| T3.2 | A{1,3}  | *Context: Understanding the violation reporting process*  
How does the reporting of rule violations work? Where does a rule violation originate and how is it communicated to the user? Write down the classes directly involved in the process. Hint: The output format is set to HTML. | 4    |
| T4 | A{1,7,9} | *Context: Gaining a general understanding*  
Starting from the Mainclass PMD – On high level, what are the main abstract steps that are conducted during a PMD checking run. Stick to a maximum of five main steps. Hint: This is an exploration task to get an overview of the system. One strategy is to follow the communication between classes/packages. Keep the handout of PMD in mind. | 5    |
<table>
<thead>
<tr>
<th>ID</th>
<th>Category</th>
<th>Description</th>
<th>Score</th>
</tr>
</thead>
</table>
| RT1 | A{4,8} | *Context: Identifying refactoring opportunities*  
Name three classes that have high fan-in (at least 3 incoming communications) and almost no fan-out (outgoing communication). | 3     |
| RT2.1 | A{3,4,5} | *Context: Understanding the login process*  
Write down all constructor/method calls between gui.MainActivity and comm.Sync. | 3     |
| RT2.2 | A{1,2,5,6} | *Context: Understanding the antibiotics display process*  
In general terms, describe the lifecycle of data.User: Who creates it, how is it used? Write down the method calls. | 3     |
| RT3 | A{1,3} | *Context: Understanding the antibiotics display process*  
How does the display of antibiotics work? Where and how are they created? Write down the classes directly involved in the process. | 6     |
| RT4 | A{1,7,9} | *Context: Gaining a general understanding*  
Starting from the Mainclass gui.MainActivity - What are the user actions (e.g., Login and Logout) that are performed during this run of Babsi. Write down the classes of the activities/fragment for each user action. Stick to a maximum of seven main steps (excluding Login and Logout). Hint: This is an exploration task to get an overview of the system. One strategy is to follow the communication between classes. | 7     |
## Tasks 3D Print Experiment

<table>
<thead>
<tr>
<th>ID</th>
<th>Category</th>
<th>Description</th>
<th>Score</th>
</tr>
</thead>
</table>
| T1 | A{3,5,6,8} | Context: Metric-Based Analysis  
Find the package containing the one class having the most instances in the application. How is the package named? How many classes (and subpackages if existing) does it contain? Please write down the full package path. | 2 |
| T2 | A{6,8} | Context: Structural Understanding  
What are the names of the three packages directly containing the most classes (without their subpackages)? Please order your answer by beginning with the package containing the most classes and write down the full path. | 4 |
| T3 | A{1,3,7} | Context: Concept Location  
Assuming a good design, which package could contain the Main class of the application? Give reasons for your answer. | 2 |
| T4 | A{3,4} | Context: Structural Understanding  
Which package name occurs the most in the application? In addition, shortly describe the distribution of these packages in the system. Hint: Have a look at the different levels of the packages. There are exactly two types of distribution. | 3 |
| T5 | A{1,2,3,9} | Context: Design Understanding  
What is the purpose of the lang package and what can you say about its content regarding PMD? Are there any special packages? Do they differ by size? Ignore the xpath and dfa packages and name three facts in your answer. Hint: Remember the received paper about the introduction to PMD. | 3 |
### Tasks Landscape Experiment

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Context: Identification of Critical Dependencies</strong></td>
<td></td>
</tr>
<tr>
<td>T1</td>
<td>Name three applications that have a high fan-in (at least two incoming communication lines). The two incoming communication lines should be on one node and not distributed over multiple nodes.</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td><strong>Context: Potential Bottleneck Detection</strong></td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td>Name the Top 3 communications with the highest request count in descending order. Write down the start application and the end application.</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td><strong>Context: Scalability Evaluation</strong></td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td>Which applications are duplicated on multiple nodes? The answer should contain all 8 duplicated applications which are all named differently. Hint: The hostname of the nodes, where the applications are running, are numbered, e.g., Server 1, Server 2,...</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td><strong>Context: Service Analysis</strong></td>
<td></td>
</tr>
<tr>
<td>T4</td>
<td>What is the purpose of the WWWPRINT application in your opinion? How does the process might work to achieve the functionality for the user?</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td><strong>Context: Risk Management</strong></td>
<td></td>
</tr>
<tr>
<td>T5</td>
<td>What are the consequences of a failure of the LDAP application? Name all affected applications and briefly describe their purposes. Hint: Remember the received paper about the introduction to the university landscape.</td>
<td>7</td>
</tr>
</tbody>
</table>
Gestures

Translation

Rotation

Zoom in and out

[Fittkau et al. 2015f, Krause 2015]
Extensibility (Control Center)

Symptoms

Diagnosis

Planning

✅ Successful

[Fittkau et al. 2014a, Gill 2015, Michaelis 2015, Mannstedt 2015]