An Empirical Investigation Into a Large-Scale Java Open Source Code Repository

Mark Grechanik, Collin McMillan, Luca Deferrari, Marco Comi, Stefano Crespi, Denys Poshylvanyk, Chen Fu, Qing Xie, Carlo Ghezzi
Joint work between Accenture Technology Labs, University of Illinois at Chicago, College of William & Mary, and Politecnico di Milano
Motivation

- Getting insight into different aspects of source code artifacts
  - One trillion lines of code have been written
  - 35 billion lines of code added / year
- Getting empirical evidence of common patterns and facts of how programmers write code

Typical usage
- Provide guidance for commonly used techniques, patterns
- Validate assumptions
- Find matched subjects for empirical studies
Challenges

- Source code files in large code repositories are treated as unstructured text by search engines and utilities
- Source code files are contained in compressed project files in the repositories
- Many repositories are polluted with poorly functioning projects
  - Fault tolerance mechanism
- Users should be able to form declarative queries
  - No low-level programs that traverse parse trees
  - SQL
SourceForge

- Largest open-source software development website
  - Over 240,000 projects
  - Over 30,000 Java projects

- Widely used software
  - eMule – 539,287,695 downloads
  - 7-zip – 103,139,981 downloads
  - jEdit – 5,931,227 downloads
Infrastructure

INTERNET

Download

ARCHIVER

Project Directory

Database

CRAWLER

Threads

PARSER

Syntax-tree

SYSTEn

Analyze

Write

Query 1

Query 2

Query 3
Components

- Archiver
  - Crawl Sourceforge to retrieve Java projects
  - Populate project folders (250GB)

- Walker
  - Traverse project folders
  - Extract source files from zipped archive
  - Apply parser to the extracted source code

- Parser
  - Use JavaCompiler to Parse source code to build parse trees
  - Content of nodes are traversed and stored in databases

- Database
  - 71 tables and 278 attributes
  - Schema matches (non)terminals of the Java grammar
  - 9.2GB data (962MB compressed)
  - Publicly available at http://www.cs.wm.edu/semeru/treasure/
Query

- SQL query to state research questions

Knowledge of
- Database schema
- Relations between schema and Java grammar
- How to translate plain English to SQL

Can be simple or complicated
- SELECT c.name AS class, COUNT(m.id) AS number_methods FROM method m JOIN class c ON m.class = c.id GROUP BY c.id HAVING COUNT(m.id) >=100
Empirical Evidence

- 2080 Java Applications
- 32 research questions
  - Classes and interfaces
  - Methods and constructors
  - Fields
  - Statements
  - Exceptions
  - Variables
  - Evolution and Maintenance
Classes and Interfaces

- 270,973 classes
  - 5,827 declared as abstract
  - 7,368 static classes
  - 29,237 anonymous classes
  - 14,270 nested classes

- 116,194 classes that are in some inheritance hierarchy
  - Maximum depth is 5

- 2026 interfaces extend hierarchies
  - Maximum depth is 4
Number of Classes per Application

Max 2017
Min 1
Median 33
Mean 97
Total 270,973
Inheritance Hierarchies Depth

Max 5
Min 1
Median 1
Mean 1
Total 116,194

AdminIPAccess->IPAccessControl->LockssServlet->HttpServlet->GenericServelet
Methods and Constructors

- 938,779 methods in classes
  - 35,846 occurrences in recursive method calls
  - 231,647 static methods (excluding main)
  - 414,953 return void vs 523,826 return non-void
  - 840,937 use “this”
  - 544,324 have at least one argument

- 84,130 methods in interfaces

- 145,124 classes do not define constructors
- 106,892 classes have overloaded constructors
Number of Arguments per Method

Max 30
Min 1
Total 544,324
Number of Overloaded Constructors per Class

Max 25
Min 1
Total 106,892
Fields

- 448,898 fields in classes
  - 492 volatile
  - 2,305 transient
  - 154,067 static
  - 231,647 of type String

- 831 out of 29,907 assignments to a static field is null
  - Signal garbage collection

- Correlation coefficient is 0.99 for number of methods and number of fields in classes
Number of Fields per Class

![Bar chart showing the number of classes by number of fields.](chart.png)
The number of fields in a class is strongly correlated with the number of methods in the same class.
**Statements**

- 620,419 conditional statements
  - If–else/switch/for/while/do–while
  - 4,956 using simple boolean variables as conditions
  - 42% of switch statement do not contain default path
- 397,605 methods don’t have conditional statements
- 2,047 assert statements
Assert Statements per Method

![Bar chart showing the number of methods with different numbers of assert statements. The chart indicates a significant number of methods with 0 assert statements, followed by a peak at 1 assert statement, and a decline as the number of assert statements increases.](chart.png)
Exceptions

- 93,714 try/catch statements
  - Finally is used 6.8%
- 19,181 exceptions thrown using keyword throws
- 110,740 propagated exceptions
Try/Catch Statements per Method

![Bar chart showing the distribution of try/catch statements per method. The x-axis represents the range of methods (1-5, 6-10, 11-15, etc.), and the y-axis represents the number of try/catch statements. The chart displays a significant peak for methods 1-5, with a decrease in frequency as the method range increases.](image-url)
Local Variables and Types

- 818,358 local variables
  - 10% final
- Use primitive types much more than corresponding class-based types
Type Classes

![Bar chart showing the total number of type classes for different data types. The chart indicates that integer has the highest total number of type classes, followed by long and boolean. Double, float, byte, and short have significantly fewer type classes.]
Primitive Type

![Bar chart showing the number of basic types]

<table>
<thead>
<tr>
<th>Type</th>
<th>Total # of basic types</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>2500000</td>
</tr>
<tr>
<td>double</td>
<td>225000</td>
</tr>
<tr>
<td>float</td>
<td>200000</td>
</tr>
<tr>
<td>long</td>
<td>175000</td>
</tr>
<tr>
<td>boolean</td>
<td>150000</td>
</tr>
<tr>
<td>byte</td>
<td>25000</td>
</tr>
<tr>
<td>short</td>
<td>25000</td>
</tr>
</tbody>
</table>
Evolution and Maintenance

- Select applications which have at least 2 versions
  - Total 2,427 versions range from 2–24
    - 6,249 removed/added fields between versions
    - 7,861 removed/added methods between versions
    - 5,713 removed/added classes between versions
Added/Removed Number of Fields across Versions

The chart shows the number of field pairs for different increments in the number of fields. The increments are as follows:

-143, -95, -48, 0, 47, 95, 143, -190, -142, -94, -47, 0, 48, 96
Related Work

- Infrastructure
  - FLOSSMole
    - Metadata on collaboration purpose
  - SourcererDB

- Empirical study
Conclusions

- Built the infrastructure
- Obtained insights into 2,080 Java applications
- Posed 32 research questions

Future work
- Deep dive
  - Extreme cases
  - Correlations
  - Rationale
Thanks

- http://www.cs.wm.edu/semereu/treasure/