Challenges in Mining Whole Software Universe

Katsuro Inoue
Osaka University
Analyzing Evolution of **kern_malloc**

Results by G(Google Code Search) and K(Koders):

- **File in New BSD License**
- **File in Original BSD License**

- 1. Lites 1.0 (G)
  - Kernel Source (G)
  - 28-33 Kame (G)
- 2. Archive - CMU Mach 3.0 (K)
  - 34-36 SimOS (K)
- 3. Lites 1.1.13 (G)
  - 27-46 Kame (G)
- 4. Lites 1.1.950808 (G)
  - 47 Netnice (G)
- 5. The Rio (RAM I/O) Project (K)
  - 48 Kame (G)
- 6. ftp in The University of Edinburgh (G)
  - 1998/07/24 2001/04/19 2004/01/14 2006/10/10 2009/07/06
- 7. Mip-summer98 (G)
  - 49-50 Psunip (G)
- 8. FreeBSD/SPARC (G)
  - 51 Netnice (G)
- 9. ftp in Stockholm University (G)
  - 52 Reflexprotocol (G)
- 10. FreeBSD-cam2.1.5R (G)
  - 53 Netnice (G)
- 11. SonicOSX (G)
  - 54 NetBSD v1.105 (K)
- 12. Labyrinth BSD (labyrinthos) (K)
  - 55 OpenBSD PV Xen (K)
- 13. OsKit (G)
  - 56 OpenBSD v1.73 (K)
- 14. Psunip (G)
  - 57 Pmon (G)
- 15. Mach (G)
  - 58 Proyecto A.T.L.D. GNU/hurd(extremelinux) (K)
- 16. Savannah (G)
  - 59 774 (G)
- 17. Unofficial OSKit source (K)
  - 60 OpenBSD v1.74 (K)
- 18. Unofficial OSKit source(oskit) (K)
  - 61 Pmon (G)
- 19. 20-22 (K)
  - 62 Openbsd-loongson-vc (G)
- 20. ftp in Stockholm University (G)
  - 63 OpenBSD v1.74 (K)
- 21. Unofficial OSKit source (oskit) (K)
  - 64 Pmon (G)
- 22. Chord-ns3 (G)
  - 65 774 (G)
- 23. Unofficial OSKit source (oskit) (K)
  - 66 Openbsd-loongson-vc (G)
- 24. 20-22 (K)
  - 67 Openbsd-loongson-vc (G)
- 25. 20-22 (K)
  - 68 Openbsd-loongson-vc (G)
Analyzing Reuse of Outdated Libraries

Vulnerability of 50 OSS Projects Using libpng

Vulnerabilities reported

No defects reported

Result from Google Code Search and Koders
Experience and Concern

Mining source code repositories, e.g., SourceForge, Github, Open Hub, Google Code, Marven, ...

(BlackDuck)

– Outcomes heavily depend on repository contents
– Aren't we mining a small world?
– There may be many other source code contents in the universe
Whole Software Universe \( U \)

- Whole Software Universe

\[ U \equiv \text{Collection of All Software Developed by Human in the Past} \]
  - Open source software
  - Personally-developed software
  - Proprietary software
  - any others

- \( P \): Set of all meaningful software
  (a countable infinite set)

- \( U \subseteq P \)
Questions for $U$

A) How do we get $U$?

B) What do we mine from $U$?

C) How do we mine $U$?

D) Why do we mine $U$?
A) How Do We Get $U$?

• No one knows actual $U$
• So we would collect many repositories, and construct a subset $U' \subseteq U$
• $U'$ should be as large as possible, of course
• $U'$ should reflect characteristics of $U$
• Challenges
  – Collecting and unifying different repositories into $U'$
    • Duplication, coherence, ...
  – Performance and capacity for $U'$
  – Updating and maintaining $U'$
B) What Do We Mine from $U$?

Examples

- Simple metrics of $U$ over history
  - Size $|U|_{t1}, |U|_{t2}, ...$
  - Language usage

- Density of $U$ with respect to $P$

- History and evolution of code $c$ in $U$
  - Origin version of $c$
  - Closely related code $c'$ (clone, variation, family, ...)
  - Future prediction for $c$
C) How Do We Mine $U (U')$?

1. Direct mining
   - Good model
   - Powerful machine

2. Indirect mining
   - Use external services
   - Reconstruct mining result from those external services
Indirect Mining

Want to know about $U'$

Mashup Engine

Query Decomposition and Result Composition
D) Why Do We Mine $U$?

Objectives of mining $U$

• Reuse and knowledge transfer
  – We do not want to reinvent the wheel

• Historical Archive
  – Frontier's wisdom

…
Discussion!

• Is it interesting research topics?

• Can we get useful research results?

• Is it feasible research target?
Thank you