Goal: Practice with the frequent itemset mining technique with source code data
Author: Martin Monperrus

Preparation

Task : Download `MethodCallCollector.java` on Moodle

Task: Compile it using Soot (http://www.sable.mcgill.ca/soot/), either the SVN head version or the latest stable version at your choice.

Training

Task: (Data Preparation): Download the source code of Tomcat 6.0 on Moodle, create a project and compile it.

Task: (Data Collection): Use MethodCallCollector to analyze the resulting bytecode. What do the resulting traces (further called all.dat) contain? How many traces do you obtain?

Task: (Data Filtering): Change MethodCallCollector so that you keep only traces of application-specific types (i.e. remove java.*). The resulting dataset is called app.dat. How many traces do you obtain?

Task: Download Borgelt's apriori implementation at http://www.borgelt.net/apriori.html

Task: (Analysis): Use apriori to compute the maximal frequent sets on the traces of tomcat. What is the flag for obtaining maximal frequent sets?

- What is the maximum support to obtain at least 1 frequent set?
- What is the maximum support to obtain at least 1 frequent set of size 3?
- What does the most frequent itemset of app.dat mean?

Exercise

Task: Choose an open-source library from the list in appendix. Use code.ohloh.net to identify 10 client applications. Download the Jar files of those 10 client applications. Decompress the .class files from all Jar files in a single directory

Task: Run the method call collector on this directory to extract the list of method calls (using the *-allow-phantom*-refs flag). Filter the output to only keep the method calls of the library under study.

Task: For 10 library-specific frequent sets of the top-50 (support) maximal itemsets of size >2, answer to the following questions (if relevant):

- What's the meaning of the itemset?
- Are the elements of the frequent set documented in Javadoc as especially important?
- Fnd and comment a code snippet which supports the frequent set.
- Would it be a bug not to have one of the method calls of the frequent set?
- Does the involved itemset fill a lack in the default API (I.e. is it an application-specific pattern which could not be reused elsewhere?)

Appendix: list of possible Java applications

1. Castor
2. JDOM
3. Piccolo
4. Saxon
5. XBean
6. XOM
7. XPP
8. XStream
9. Xalan-J
10. Xerces-J
11. Batik
12. BluePrints UI
13. CGLib
14. Ganymed ssh
15. Genericra
16. HOWL
17. Hibernate
18. JGroups
19. JarJar Links
20. Log
21. MOF
22. MX
23. OGNL
24. OpenSAML
25. Shale Remoting
26. TranQL
27. Trove
28. XML Security
29. Codec
30. Collections
31. DBCP
32. Digester
33. Discovery
34. EL
35. FileUpload
36. HttpClient
37. Lang
38. Modeler
39. Net
40. Pool
41. Validator