SEXTANT: A TOOL TO SPECIFY AND VISUALIZE SOFTWARE METRICS FOR JAVA SOURCE-CODE

Victor Winter, Carl Reinke, and Jonathan Guerrero
SETTING THE STAGE

- Metrics is the science of indirect measurement
  - Metrics leverage “that which can be measured (e.g., complexity)” to make predictive statements about “that which cannot be measured (e.g., correctness)”
- As a field of study, metrics is heavily influenced by:
  - Analytical capabilities/limitations of the “tools of the day”
  - Ability to represent information obtained from such tools in a comprehensible fashion
- Metrics include
  - Standard metrics (SLOC, NOM)
  - Complexity measures (McCabe and Halstead)
  - Design measures (coupling and cohesion)
  - Source code analysis rules (PMD and FindBugs)
SEXTANT

- A source-code analysis tool under development at the University of Nebraska at Omaha.
- A non-trivial extension of the TL system specialized to the domain of the Java programming language.
- Primary Design Goal:

  to provide a tool facilitating the **specification** and **visualization** of custom analysis rules.....In particular, rules that are domain specific or even application specific.
ON THE TECHNICAL SIDE

• Sextant currently uses two software models
  • Parse tree
  • Compound Attributed Graph (CAG)

• Analysis:
  • Accomplished primarily through semantically informed program transformation
  • Tables and sets can be used to store information associated with analysis rules
  • Sets can be used to model arbitrary predicates
VISUALIZATION AND ANALYSIS

- Accomplished through third party tools
  - GraphViz
    - dot format
  - TreeMap
    - csv format
  - Cytoscape
    - Java-centric Sextant plugins
    - json format
  - SolidSX
    - xml format
Some of the things that can be done

IMAGES
Impact of overriding method declarations on package structure
Impact of overriding method declarations on subtype hierarchy
Unsafe uses of generic types in the CDK Molecular package
Dependencies on the native method `fillInStackTrace`
SolidSX: CLINIT Test Suite
Usage Analysis of Generic Types

APPLICATION
MOTIVATION

• At Sandia National Laboratories, a hardware implementation of the JVM, called the SCore processor, has been developed for use in high-consequence embedded systems.

• **Task:** Develop a tool capable of migrating selected Java Core APIs to the SCore platform.
  • java.io, java.lang, java.util

• Use software metrics to focus development efforts.
  • Focus: **generic types**
GENERIC TYPE DECLARATIONS
TYPE VARIABLES
USES OF GENERIC TYPES
INTERSECTION WITH REMOVED ELEMENTS
THE END

Questions?