

Detection Strategies

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

Motivation

- A metric alone cannot answer all questions about the system
 - Metrics must be used in combination
- A metric is too fine grained to be understood
 - We need to provide engineers with means to work with metrics at a more abstract level

Detection Strategies

- A detection strategy is a composed logical condition, based on metrics, by which code fragments with specific properties are detected
- It relies on two mechanisms
 - Filtering
 - Composition

Filtering Mechanism

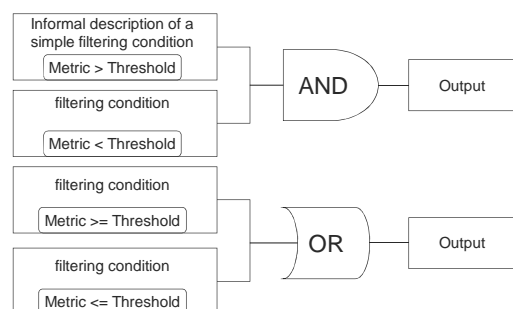
- Its goal is to reduce the data set
- Two types of filters
 - Absolute comparator uses classical comparison of numbers ($>$, $<$, $>=$ e $<=$)
 - Relative comparator specifies the number of entities to be retrieved (i.e., relative to the original data)

Composition Mechanism

- It supports a interpretation of multiple result sets based on the AND and OR operators
- Notation



Abstract Example



[Detection Strategies]

- God Class
- God Method
- Feature Envy
- Shotgun Surgery
- Refused Bequest

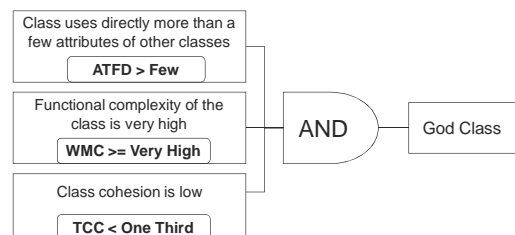
[Work Definition of God Class]

- God Class ...
 - performs too much work
 - lacks internal relationships between its methods
 - accesses a lot of data from other classes
- In other words, God Class has
 - High complexity
 - Low cohesion
 - Access foreign data

[Selecting Metrics for God Class]

- High complexity
 - Weighted Method per Class (WMC)
- Low cohesion
 - Tight Class Cohesion (TCC)
- Access foreign data
 - Access to Foreign Data (ATFD)

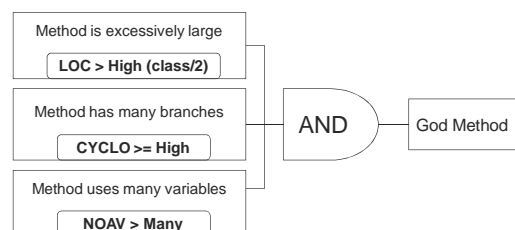
[Strategy for God Class]



[God Method]

- It is a long method (size)
 - Line of Code (LOC)
- It makes intensive use of branches
 - McCabe's Cyclomatic Complexity (CYCLO)
- It defines many local variables and uses many instances variables
 - Number of Accessed Variables (NOAV) (counts attributes, local variables and parameters)

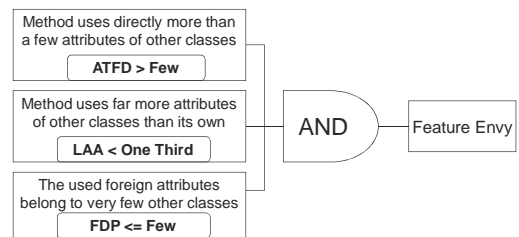
[Strategy for God Method]



[Feature Envy]

- Method uses directly more than a few attributes of other classes
 - Access to Foreign Data (ATFD)
- Method uses far more attributes from other classes than its own class.
 - Locality of Attribute Accesses (LAA)
- The used foreign attributes belong to very few other classes
 - Foreign Data Providers (FDP)

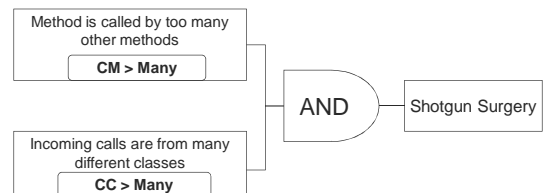
[Strategy for Feature Envy]



[Shotgun Surgery]

- Operation is called by too many other operations
 - Changing Methods (CM)
- Incoming calls are from many classes
 - Changing Classes (CC)
- CC and CM count coupling connections from other classes outside the inheritance tree

[Strategy for Shotgun Surgery]



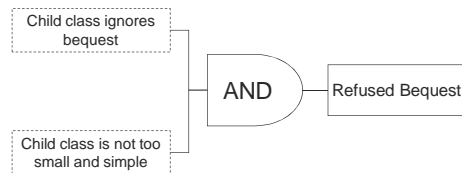
[Refused Bequest]

- Child class ignores bequest
 - Parent provides more than a few protected methods
 - Child uses only little of parent's bequest
 - Overriding methods are rare in child
- Child class is not too small and simple
 - Functional complexity above average
 - Class complexity not lower than average
 - Class size is above average

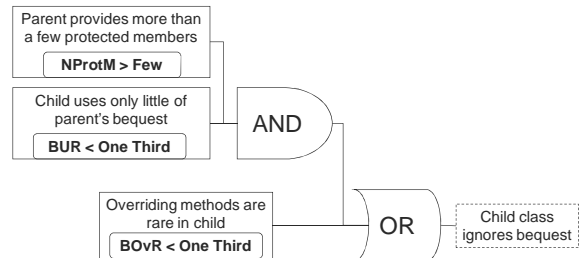
[Used Metrics]

- Child class ignores bequest
 - Number of Protected Members (NProtM)
 - Base-class Usage Ratio (BUR)
 - Base-class Overriding Ratio (BOvR)
- Child class is not too small and simple
 - Average Method Weight (AMW)
 - Weighted Methods per Class (WMC)
 - Number of Methods (NOM)

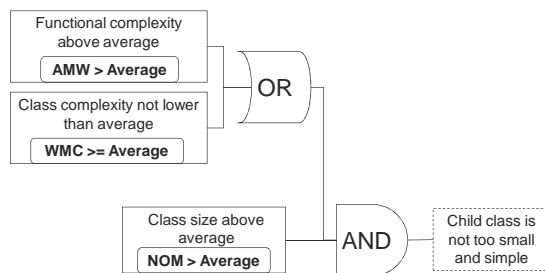
Strategy for Refused Bequest



Child Class Ignores Bequest



Child Class is not Too Small



Research Question

- Do detection strategies accurately identify the bad smells of a system?
 - Tenho particular interesse em saber a resposta para esta pergunta.
 - Sugestão de tema (estudo) para o trabalho prático da disciplina.



E. Figueiredo et al. **Applying and Evaluating Concern-Sensitive Design Heuristics**. Brazilian Symposium on Software Engineering (SBES), 2009.

Bibliography

- M. Lanza e R. Marinescu. **Object-Oriented Metrics in Practice**. Springer, 2006.
 - Section 4.1 Detection Strategies
 - Section 5.3 God Class
 - Section 5.4 Feature Envy
 - Section 5.6 Brain Method
 - Section 6.5 Shotgun Surgery
 - Section 7.3 Refused Parent Bequest