

Paradigm Independent Software Complexity Metrics

Dr. Zoltán Porkoláb
Department of Programming Languages and Compilers
Eötvös Loránd University, Faculty of Informatics

e-mail: gsd@elte.hu

The structure of the presentation

- ◆ The role of software metrics
- ◆ Metrics: an overview
- ◆ Object-oriented software metrics
- ◆ Multiparadigm programming
- ◆ The AV-graph
- ◆ Results

Role of software metrics

- Cost of the software
 - Specification
 - Design
 - Implementation
 - Testing & Bug-fixing
 - Maintenance
- More than 70% of cost is for testing and maintenance (Zuse 1998)
- Software quality

SQI 2005.04.11.

3

Software metrics

- Software metrics:
 - Measuring the development process (process metrics)
 - Measuring the product (product metrics)
- Product metrics:
 - External metrics:
 - Reliability metrics
 - Functional metrics
 - Efficiency metrics
 - Internal product metrics:
 - Size
 - Complexity
 - Style

SQI 2005.04.11.

4

Product metrics

- Size metrics:
 - LOC, eLOC
 - Ignore the semantic
- Structural metrics:
 - McCabe 1976
 - Motivation: predict testing efforts
 - For structured programs: $V(G) = p + 1$
 - Howatt and Baker 1989
 - Motivation: involve nesting level
 - $SN(G) = |N| + ND(G)$

SQI 2005.04.11.

5

Object-oriented metrics in theory

- Chidamber-Kemerer (1994) OO metrics suite:
 - WMC (Weighted Methods per Class)
 - DIT (Depth of Inheritance Tree, DIT)
 - NOC (Number of Child Classes)
 - CBO (Coupling Between Object Classes) fan-in and fan out
 - RFC (Response for Class)
 - LCOM (Lack of Cohesion in Methods)
 - Chidamber - Kemerer
 - Henderson - Sellers

SQI 2005.04.11.

6

Object-Oriented metrics in practice

- Eclipse (www.eclipse.org)
 - McCabe cyclomatic complexity
 - Efferent coupling (≠ fan out!)
 - Lack of Cohesion
 - LOC (Lines of Code)
 - Number of Fields in Classes
 - Nesting Depth
 - Number of Method Parameters
 - Number of Statements
 - Weighted Methods per Class

SQI 2005.04.11.

7

Software paradigm

- Evolution of Software paradigm
 - Structured programming
 - Object-oriented programming
 - Classes, inheritance, virtual function
 - Generative programming
 - Aspect-Oriented (Kitzales 1994)
 - Intentional (Simonyi 1995)
 - Template metaprogramming (Veldhuizen 1994)
 - Multiparadigm programming (Coplien 1998)
 - Simultaneous usage of paradigms

SQI 2005.04.11.

8

Paradigm-independent Software Metrics

Applicable for programs written in different paradigms or in mixed-paradigm environment

Based on general programming language features which are paradigm- and language-independent. The paradigm-dependent attributes are derived from these features.

SQI 2005.04.11.

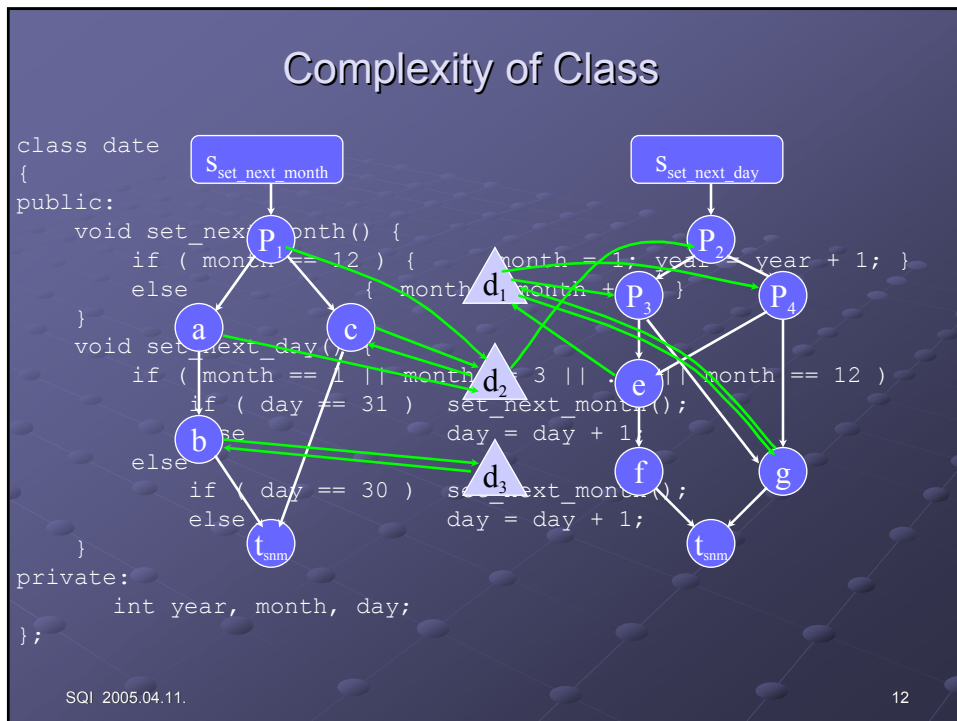
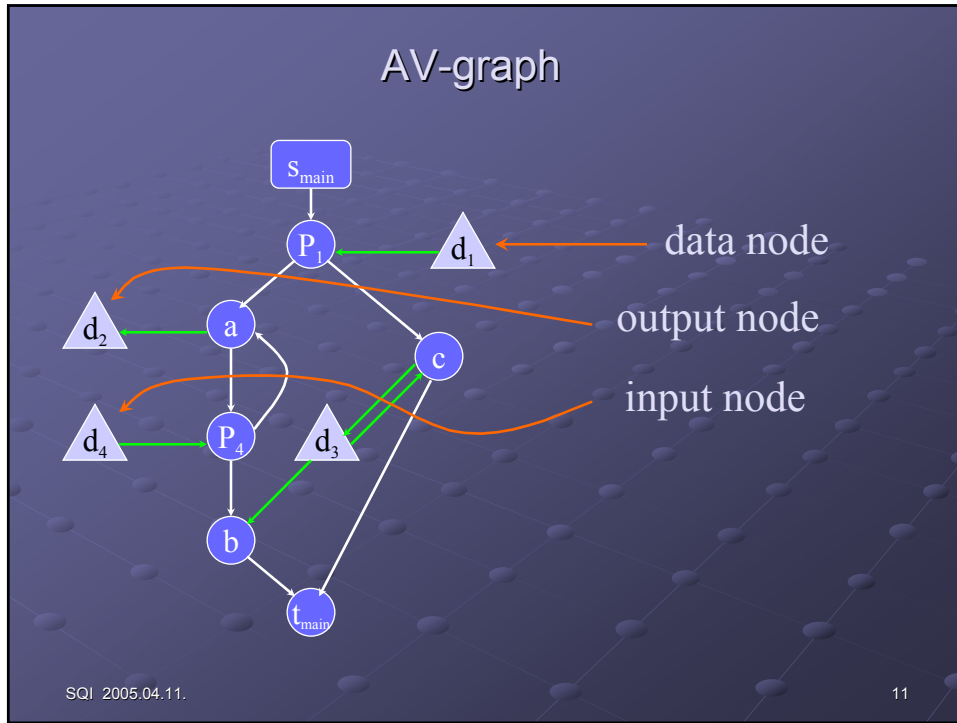
9

Components

- Control Structure of Program
 - Most of the programs share the same control statements
- Complexity of Data Types
 - Reflects the complexity of data types used (like classes)
- Complexity of Data Access
 - Connection between control structure and data
 - Direction of data flow
 - Nesting depth

SQI 2005.04.11.

10



Our Measuring tool

- Supported languages
 - Java 1.3
 - Java 1.4 (assert)
 - Java 5 (generics)
- Implementation
 - standalone application and Eclipse Plug-in
- Output
 - CSV
 - XML

SQI 2005.04.11.

13

Test Data

- Java Standard Library 1.4.2
 - 367.000 eLOC
- jBOSS 3.2.3
 - 300.000 eLOC
- Omg.org.CORBA
 - 5.000 eLOC
- The measure tool (with mostly generated parser)
 - 7.000 eLOC
- Eclipse 3.0M6
 - 900.000 eLOC
- **17.000 class – more than 1.5 million lines**

SQI 2005.04.11.

14

Results

- No statistical correlation between the OO and multi paradigm metrics:
 - OO metrics only measure the big picture
 - MPM consider more properties: higher density
 - The structural complexity of methods are extremely increase the overall complexity.