



ADMINISTRACIÓN Y GESTIÓN DE PROYECTOS DE SOFTWARE - (7500)

ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE - (7502)

Segundo cuatrimestre de 2017

TRABAJO PRÁCTICO 9

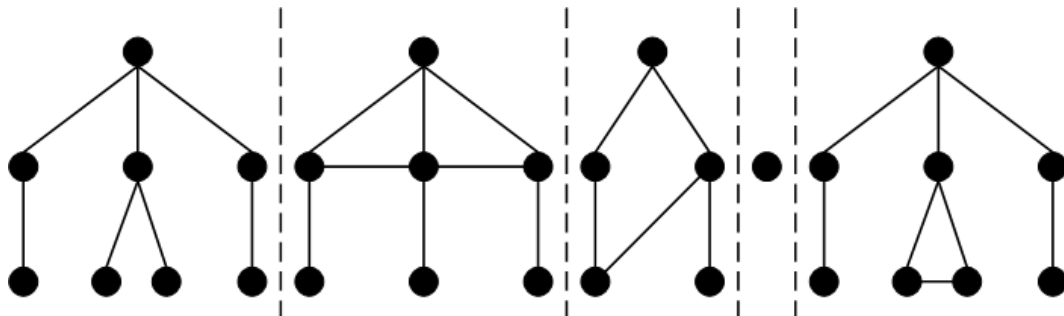
Métricas III: Atributos Internos - Métricas OO

Métricas de atributos internos

1. ¿Cuál es la diferencia entre analizar la complejidad de un problema y analizar la complejidad de una solución?
2. ¿Cómo se analiza la complejidad de una solución? ¿Cómo se puede medir la eficiencia de una solución?

Medidas morfológicas

3. ¿Qué se entiende por módulo? ¿Con qué herramientas se cuenta para representar información acerca de las relaciones entre módulos?
4. ¿Qué atributos de la morfología de un diseño pueden medirse?
5. ¿Qué mide la “medida de impuridad de árbol”? ¿Qué propiedades debe satisfacer?
6. Defina una medida de impuridad de árbol. Luego ordene los siguientes diseños de acuerdo a su grado de impuridad.



Reuso

7. ¿Qué información aporta medir el reuso de un sistema? ¿Cómo se puede medir?

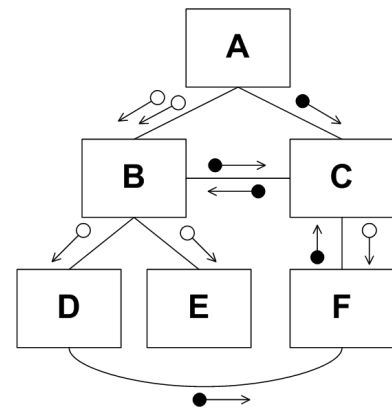
Acoplamiento

8. Explique y dé ejemplos de cada uno de los siguientes tipos de acoplamiento entre módulos:
 - R1: de datos
 - R2: de molde
 - R3: de control
 - R4: común
9. Explique la métrica propuesta por Fenton y Melton para medir el acoplamiento.

10. Considere los módulos que se muestran en la figura.

Aparte de los flujos de dato y control que se muestran asuma que B y C acceden a tres items de dato globales en común y que D y E aceptan como parámetro el mismo tipo de registro.

Indique, para cada par de módulos i y j , la medida de acoplamiento de Fenton & Melton, $c(i, j)$ y para el sistema la medida de acoplamiento general.



Cohesión

11. ¿A qué se denomina cohesión?

12. a) Describa los siguientes tipos/niveles de cohesión:

- Coincidental
- Lógica
- Temporal
- Procedimiento
- Comunicacional
- Secuencial
- Funcional

b) Un módulo puede tener más de un tipo de cohesión, ¿cómo se mide la cohesión en ese caso?

13. Para un sistema íntegro, ¿qué mide el ratio: #Módulos con cohesión funcional / #Total de módulos?

Flujos de información

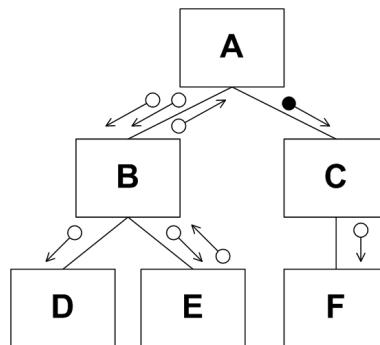
14. Defina los siguientes conceptos:

- Flujo directo local
- Flujo indirecto local
- Flujo Global
- Fan_in

15. ¿Cómo puede medirse el flujo de información?

16. Considerando el siguiente dibujo,

a) ¿qué módulos son -intuitivamente- más complejos con respecto al flujo de información en el que se ven involucrados?



b) Indique cuál es la medida de complejidad Shepperd para cada módulo.

Métricas Orientadas a Objetos

Para todas las métricas orientadas a objetos que utilice, de la definición completa mencionando la fuente.

17. Revise las métricas OO estudiadas e identifique cuál de ellas aborda en forma directa o indirecta:

- el ocultamiento de información
- la abstracción

18. Enuncie dos métricas OO para el tamaño de una clase. Explique cómo se calculan.

19. Una clase C tiene 12 operaciones. Se ha calculado la complejidad ciclomática para todas las operaciones del sistema OO en cuestión y el valor promedio de la complejidad de módulo es 4. Para la clase C la complejidad de sus operaciones de la 1 a la 12 es 5, 4, 3, 6, 8, 2, 2, 5, 5, 4, 4, 3, respectivamente. Calcule WMC (Métodos ponderados por clase)¹.

20. Analice el conjunto de clases que se presenta en el material adjunto.

a) Determine la jerarquía de clases que conforman.

b) Para todas las clases de la jerarquía calcule las siguientes métricas:

- Profundidad del árbol de herencia (DIT: Depth of Inheritance Tree)
- Número de Descendientes (NOC: Number of Children)
- Tamaño de la Clase (CS: Class Size)

c) Para las clases *Rectangle*, *RightTriangle*, *EquilateralTriangle* y *Circle* calcule:

- Número de Operaciones Invalidadas por una Subclase (NOO: Number of Operations Overridden by a Subclass)
- Número de Operaciones Agregadas por una Subclase (NOA: Number of Operations Added by a Subclass)
- Índice de Especialización (SI: Specialization Index)

d) Para las clases *Triangle*, *RightTriangle* y *EquilateralTriangle* calcule:

- Falta de Cohesión en los métodos (LCOM: Lack of Cohesion in Methods)
- Respuesta para una clase (RFC: Response For a Class)
- Acoplamiento entre Clases (Coupling Between Objects)

e) Para la clase *Triangle* calcule:

- Métodos Ponderados por Clase (WMC: Weighted Methods per Class)

En este último caso utilice como métrica de complejidad la *Complejidad Ciclométrica*.

21. La métrica para proyectos orientados a objetos "Número de guiones de escenario", ¿qué idea nos puede dar sobre un proyecto/programa?

22. El Centro de Tecnología de Seguridad del Software de la NASA (NASA-SATC) propone entre otras métricas la que llama "Porcentaje de Comentarios" y la define como: el número de comentarios dividido por el total de líneas de código menos el número de líneas en blanco. Donde el número de comentarios está definido como el número de líneas de código que tienen comentarios: tanto los comentarios que comparten la línea con código como los comentarios que ocupan una línea en sí mismos.

¿Qué significado tiene un número alto de esta métrica? ¿Qué características del código analizado refleja?

Referencias

- [1] T. DeMarco. *Controlling Software Projects: Management, Measurement, and Estimates*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1986.
- [2] Norman E. Fenton and Shari L. Pfleeger. *Software Metrics: A Rigorous and Practical Approach*. 2 edition.

¹WMC: Weighted Methods per Class