V & V

Métodos formales

Dra. Lizbeth Alejandra Hernández González
lizhernandez@uv.mx

Sistemas de seguridad crítica

- Demostrar la corrección de un software es muy importante en algunas aplicaciones:
  - Controladores de rectores nucleares
  - Sistemas de frenado de autos
  - Equipos médicos controlados por software

- La demostración de corrección formal proporciona una manera de establecer la ausencia de fallos en el software cuando una validación exhaustiva mediante pruebas no se puede hacer.

  - [Manuel Capel]

Métodos formales

- *Especificación formal*. Se utiliza una notación matemática para proporcionar una descripción precisa de lo que debe hacer un programa.
- *Verificación formal*. Se utilizan reglas precisas para demostrar matemáticamente que un programa satisface una especificación formal.
- Desarrollo formal (o Sistemático). Se desarrollan programas de una forma tal que se asegura matemáticamente que satisfacen sus especificaciones formales.
- Los métodos formales son complementarios en la validación de programas mediante pruebas.
  - NO DEBE PENSARSE EN ELLOS COMO LA PANACEA PARA LA VALIDACIÓN DE CUALQUIER TIPO DE SOFTWARE
  - Podría darse que un diseño formalmente verificado no llegase a funcionar.
  - Los métodos formales pueden dar una sensación falsa de seguridad.

[Manuel Capel]

Métodos formales (MF)

- Un método formal es una técnica basada en matemáticas, usada para describir sistemas de hardware o software [Wing, Jeannette M., 1990].
- Los métodos formales permiten representar la especificación del software, verificación y diseño de componentes mediante notaciones matemáticas.
- Para los procesos de especificación se reconocen las siguientes clasificaciones:
  1. Lenguajes basados en modelos y estados.
  2. Lenguajes basados en especificaciones algebraicas.
  3. Lenguajes de especificación de comportamiento.

1. Lenguajes basados en modelos y estados

- Permiten especificar el sistema mediante un concepto formal de estados y operaciones sobre estados.
- Los datos y relaciones/ funciones se describen en detalle y sus propiedades se expresan en lógica de primer orden (lógica de predicados).
- La semántica de los lenguajes está basada en la teoría de conjuntos.
- Ejemplos:
  - VDM, **Z**, B, OCL

2. Lenguajes basados en especificaciones algebraicas

- Proponen una descripción de estructuras de datos estableciendo tipos y operaciones sobre esos tipos.
- Para cada tipo se define un conjunto de valores y operaciones sobre dichos valores.
- Las operaciones de un tipo se definen a través de un conjunto de axiomas o ecuaciones que especifican las restricciones que deben satisfacer las operaciones.
- Ejemplos:
  - Larch, OBJ, TADs

3. Lenguajes de especificación de comportamiento

- Métodos basados en álgebra de procesos:
  - modelan la interacción entre procesos concurrentes. Su difusión en la especificación de sistemas de comunicación (protocolos y servicios de telecomunicaciones) y de sistemas distribuidos y concurrentes.
  - Ejemplos : CCS, CSP, Pi Calculus y LOTOS.
- Métodos basados en Redes de Petri:
  - una red de Petri es un formalismo basado en autómatas, es decir, un modelo formal basado en flujos de información.
  - Permiten expresar eventos concurrentes.
  - Los formalismos basados en redes de Petri establecen la noción de estado de un sistema mediante lugares que pueden contener marcas.
  - Un conjunto de transiciones (con pre y post condiciones) describe la evolución del sistema entendida como la producción y consumo de marcas en varios puntos de la red.

- Métodos basados en lógica temporal:
  - se usan para especificar sistemas concurrentes y reactivos.
  - Los sistemas reactivos son aquellos que mantienen una continua interacción con su entorno respondiendo a los estímulos externos y produciendo salidas en respuestas a los mismos.
  - El orden de los eventos en el sistema no es predecible y su ejecución no tiene por qué terminar.

Proceso de verificación

- Gracias al correcto proceso de especificación, se pueden verificar propiedades derivadas de cada módulo mediante técnicas de razonamiento asociadas a los modelos formales.
- Dos enfoques:
  1. **Verificación de modelos:** trabajan mediante una búsqueda exhaustiva en los estados posibles de un modelo para encontrar errores en la especificación.

  2. **Prueba de teoremas:** donde a partir de un conjunto de axiomas se trata de probar si la especificación, extendida con algunos teoremas, es válida.

10 mandamientos de los métodos formales

1. Elegirás la notación apropiada.
2. Formalizarás pero no en exceso.
3. Estimarás los costos (entrenamiento, adquisición de herramientas, consultores).
4. Tendrás un experto en métodos formales a tu disposición.
5. No abandonarás los métodos tradicionales de desarrollo.
6. Documentarás suficientemente. Es recomendable incluir comentarios en lenguaje natural.
7. No comprometerás los estándares de calidad.
8. No serás dogmático. No hay garantía de recorrección.
9. Probarás, probarás y probarás de nuevo.
10. Reutilizarás. (Los MF son un enfoque apropiado cuando se han de crear componentes).

Fracasos de la aplicación de métodos formales

Algoritmo de división del procesador Pentium (1993)

Costo: 475 millones de dólares, credibilidad de Intel

Explosión de la misión espacial Ariane 5 (1996)

Costo: 500 millones de dólares

Orbitador climático de Marte (1998)

Costo: 125 millones de dólares

Logros de la aplicación de métodos formales

Metéor, Matra Transport, France:

Herramienta: Atelier-B

Resultado: No se encontró ningún error durante el testing de 80000 líneas de código.

SPOT 4, CS-CI, France:

Herramienta: IFAD VDM-SL Toolbox

Resultado: Se redujo un 38% menos de código fuente y un 36% menos de esfuerzo total.

Bibliografía

- Métodos formales aplicados en la industria del software. Ensayo. Carlos A. Fernández y Fernández, Miriam Ambrosio, Gabriel Andrade, Galileo Cruz, Martín José, Román Ortiz, Hernán Sánchez.

- Ingeniería de Software, un enfoque práctico Pressman, sexta edición.