Métodos para la Construcción de Software Fiable

María del Mar Gallardo
Pedro Merino

Dpto. de Lenguajes y Ciencias de la Computación
Universidad de Málaga

(gallardo, pedro)@lcc.uma.es

Métodos para la Construcción de Software Fiable
Programa/Evaluación

1. Introducción
2. Model Checking con TDFs
3. Análisis Estático
4. Abstract Model Checking
5. Herramientas (Laboratorio)
6. Model Checking Software
7. Conferencia
8. Realización de trabajos (3 semanas)
9. Exposición de trabajos

Métodos para la Construcción de Software Fiable
Programa/Evaluación

1. Introducción

Motivación-Algunos errores software

Revisión de técnicas/herramientas de análisis

Tendencia:

Integración de técnicas y Model Checking Software

Referencias

Algunos errores software

- Industria espacial
  - Mariner 1
  - Ariane 501
  - Mars PathFinder
  - Mars Polar Lander
- Armamento
  - Misiles
- Comunicaciones
  - Fallos en servicio AT&T
- Salud
  - Therac-25
- Aviación comercial
  - Peter Ladkin, Universidad de Bilelefed

Mariner 1

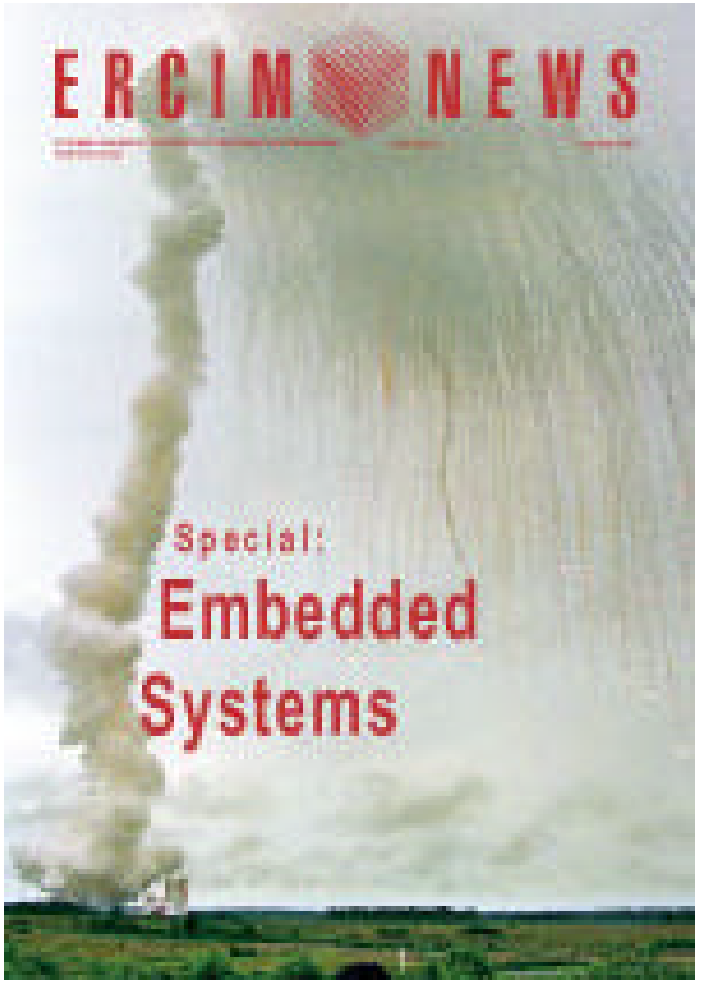
- Proyecto de la NASA para enviar sonda a Venus (1962)
  - G.J. Myers, *Software Reliability: Principles & Practice*
- Fallo:
  - Error en una sentencia de código FORTRAN
- Efecto
  - Perdida de la sonda

Mariner 1

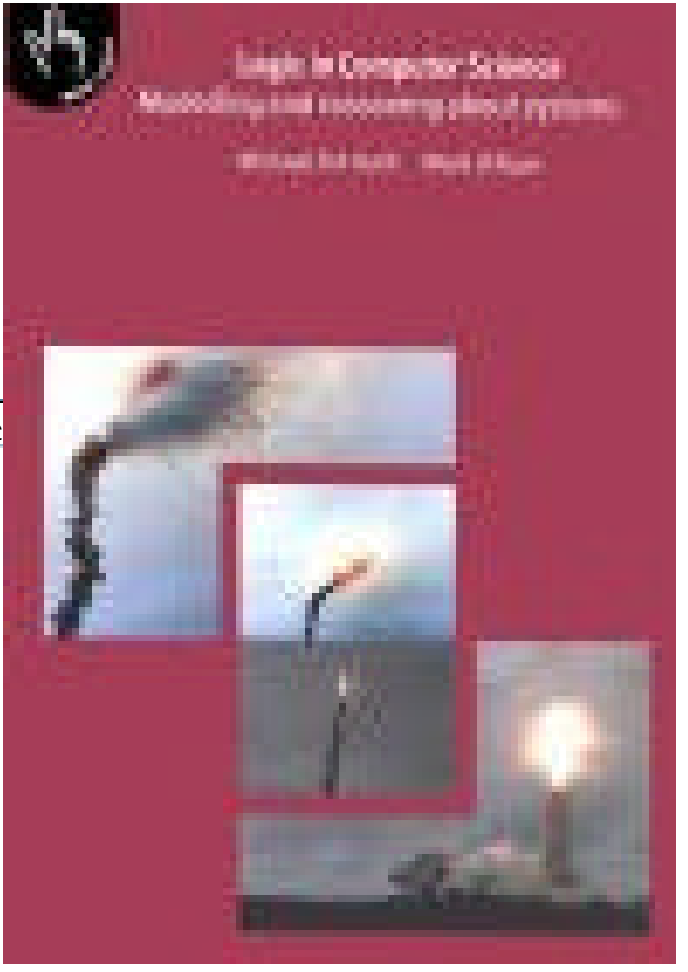
```
...
IF (TVAL .LT. 0.2E-2) GOTO 40
DO 40 M = 1, 3
W0 = (M-1)*0.5
X = H*1.74533E-2*W0
DO 20 N0 = 1, 8
EPS = 5.0*10.0**(N0-7)
CALL BESJ(X, 0, B0, EPS, IER)
IF (IER .EQ. 0) GOTO 10
20 CONTINUE
DO 5 K = 1. 3          DO 5 K = 1, 3
T(K) = W0
Z = 1.0/(X**2)*B1**2+3.0977E-4*B0**2
D(K) = 3.076E-2*2.0*(1.0/X*B0*B1+3.0977E-4*
*(B0**2-X*B0*B1))/Z
E(K) = H**2*93.2943*W0/SIN(W0)*Z
H = D(K)-E(K)
5 CONTINUE
10 CONTINUE
Y = H/W0-1
40 CONTINUE
...
```

Ariane 501

- Proyecto de la ESA para poner satélites en órbita (1996)
  - Nota de prensa ESA (http://www.esrin.esa.it/htdocs/tidc/Press/Press96/ariane5rep.html)
- Fallo:
  - Conversión de un flotante de 64 bits relativo a la velocidad horizontal a un entero de 16 bits. El número era mayor de 32,768
  - Reutilización de código del Ariane 4
- Efecto
  - Explosión a los 40 segundos del despegue
  - Valor estimado con la carga: 500 millones de dolares

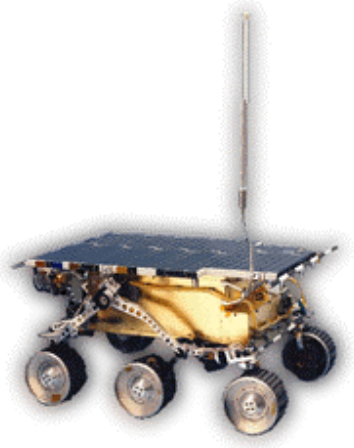


cust-rec



Mars PathFinder

- Proyecto de la NASA para analizar la superficie de Marte (1997)
  - Vehículo para toma de muestras
- Fallo:
  - Implementación errónea mecanismo de acceso a memoria compartida junto con prioridades. “Inversión de prioridades”
- Efecto:
  - “Reset” del software de forma inesperada



Mars Climate Orbiter

- Proyectos de la NASA para analizar la superficie de Marte (1999)
  - Informe NASA (ftp://ftp.hq.nasa.gov/pub/pao/reports/1999/MCO_report.pdf)
- Fallo:
  - Implementación errónea de la traducción entre millas inglesas y metros
- Efecto
  - Perdida de la nave Mars Climate Orbiter en la aproximación a Marte
  - ¿ Perdida de Mars Polar Lander ?

Misiles Patriot

- Un misil Patriot de EEUU falla al interceptar un misil Scud Iraqui (1991)
  - (http://www.math.psu.edu/dan/disasters/patriot.html)
- Fallo
  - Calculo erróneo del tiempo desde que se arrancó el ordenador por errores aritméticos al aproximar el reloj
  - Los errores se deben al uso de sólo 24 bits para representar 1/10.
- Efecto
  - 28 soldados muertos y 100 heridos

Errores con Misiles

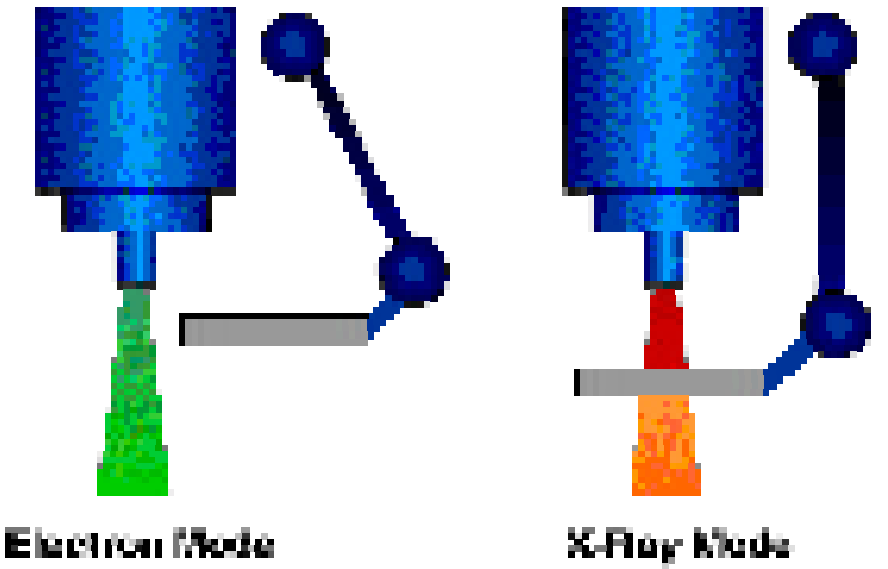
SEN References	Description	Date of Event
5/3	50 false alerts from NORAD defense system	1979
8/3	NORAD defense radar system mistook the Moon for a hostile incoming missile	
8/3	HMS Sheffield radar system identified incoming Argentinian Exocet missile as non-Soviet & thus friendly; no alarm was raised and the ship sank	
8/5	Computer error caused US naval vessel to open fire 180 degrees off target, in the direction of Mexican merchant ship	Jul 1983
10/2	180 degree heading error caused Soviet test missile to aim for Hamburg instead of the Arctic	Dec 1984

Fallos en AT&T

- Software de las centrales de conmutación de AT&T en EEUU (1990)
- Fallo
  - Sentencia Break mal empleada en una línea de código incluida como “patch”
- Efecto
  - 9 horas sin servicio telefónico en varias ciudades
  - otro error similar corta comunicaciones con Grecia (1979)

  - Desarrollo de herramientas propias

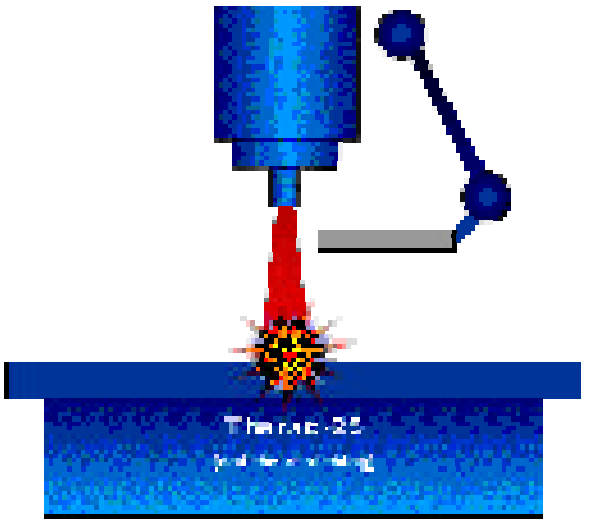
Therac-25



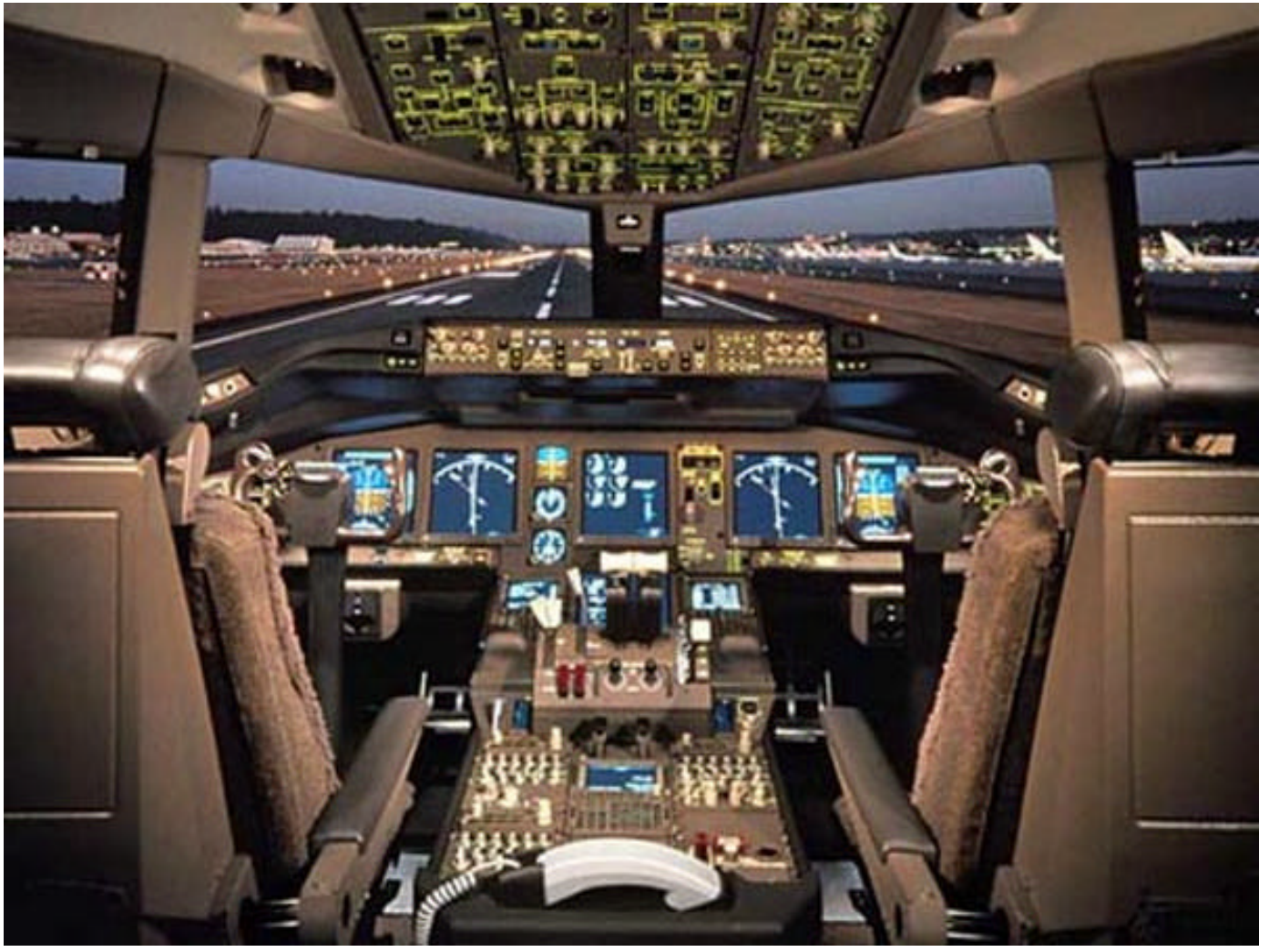
- Acelerador de electrones para tratamiento del cancer (1985-1987)

Therac-25

- Fallo
  - Sentencias muy seguidas de modo X y modo b (menos de 8 segundos). El sistema no lo haba hecho antes.
  - Él equipo emite radicación potente sin protector
- Efecto
  - 6 muertos



Aviación



Aviación Civil

- "Deadlock" en F16, confusión entre derecha e izquierda volando invertido
- Vuelo de Air New Zealand cae al detectarse un error software pero sin informar a la tripulación (1979)
- Piloto automático de China Airlines 747 hace caer el avión cerca de San francisco (1985)
- Errores en el nuevo software causan la caida de Korean Air Lines B747 en Guam (1997)
- Versión militar de Boeing B737-200 cae en Dubrovnik (Croacia) (1996)

Otros campos ..

SEN References	Description	Date of Event
10/2	"Compatible" teller machines of 2 British banks handled leap years differently, witholding cash and confiscating cards during New Year holiday	Jan 1985
10/3	Federal Reserve inter-bank transaction amounts multiplied by 1000 because data input procedures were inconsistent between client banks	
10/3	Robot killed Japanese auto worker attempting to repair another robot	Jul 1981
10/3	14000 Ford Lincolns recalled because computer in air suspension system had overheating problem, causing automobile to burst into flames	

Otros campos ..

- Hardware
- Sistemas operativos
- Industria del automóvil
- Procesos de negocios

Las grandes empresas crean grupos para fiabilidad del software
NASA, MICROSOFT, INTEL, LUCENT, VOLVO,

La fiabilidad se aborda ACTUÁLMENTE mediante técnicas de
TESTING/ANALISIS AUTOMÁTICO

¿ Qué puede obtenerse del análisis ?

- Optimización de código (secuencial)
- Ejecución paralela/distribuida
- Transformación
- Detección de errores
- Otras informaciones

Técnicas de Análisis Automático/Semi-automático

- Técnicas “tradicionales”
  - Compilación
  - Compilación especial (optimizada, “warnings”, ..)
  - Chequeo de “buen estilo” (p.e. lint)
  - Depuradores
  - .....

Técnicas de Análisis Automático/Semi-automático

- Análisis estático
  - Análisis de flujo de datos
  - Interpretación abstracta
  - Análisis basado en restricciones
  - Chequeo de tipos

"Análisis de de programas"

(Prog. Secuencial)

Técnicas de Análisis Automático/Semi-automático

- Análisis dinámico
  - Análisis de alcanzabilidad (Validación)
  - Alcanzabilidad + chequeo de propiedades (Lógica temporal) (Model-Checking)
  - Análisis en tiempo de ejecución (Testing)

"Análisis del espacio de estados"

(Prog. Concurrente)

Técnicas de Análisis Automático/Semi-automático

- Deducción automática
  - Representación del programa y las propiedades con Lógica de orden superior y uso de herramientas que asisten en la búsqueda de errores/demostraciones

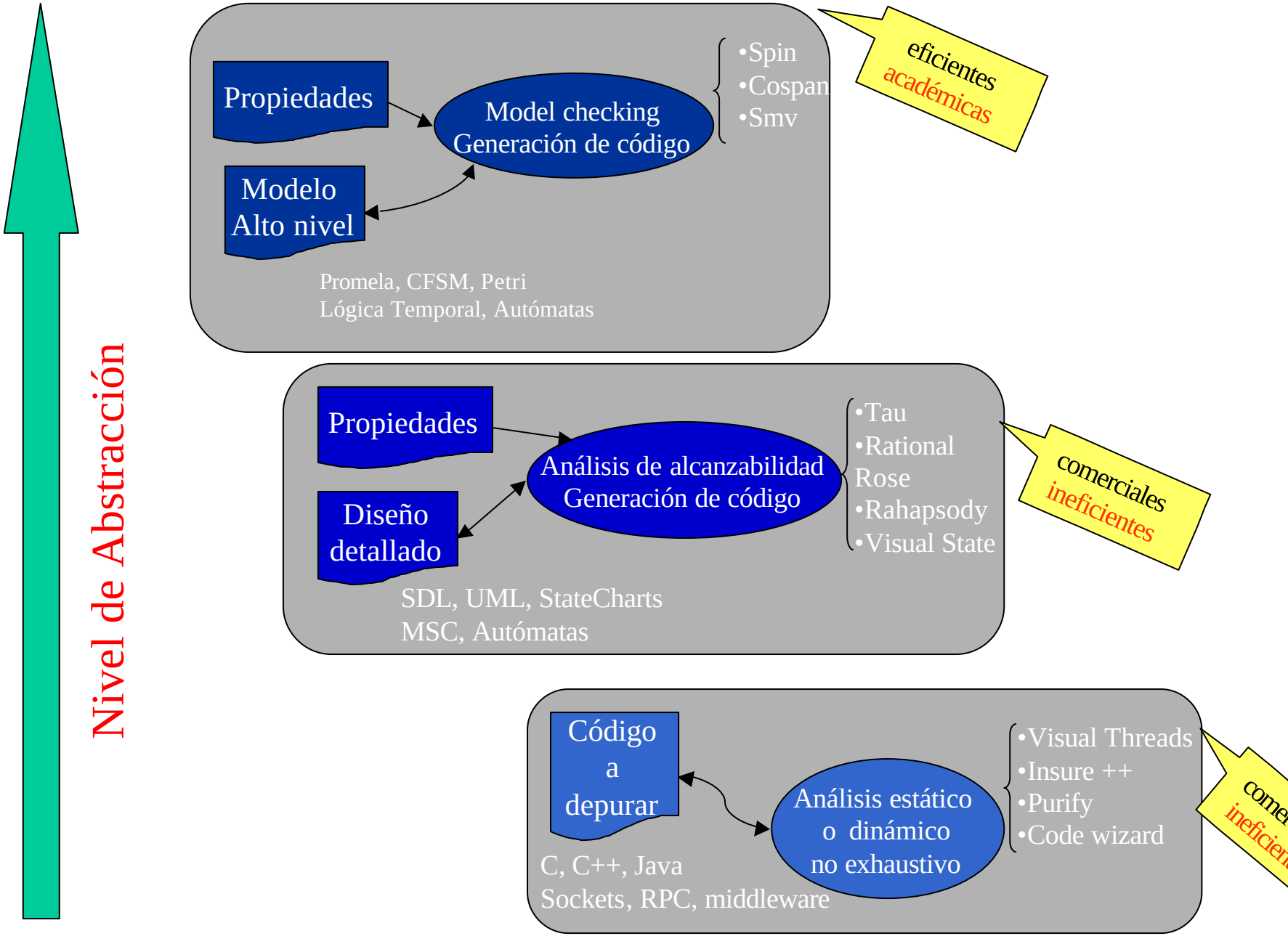
"Demostración de teoremas"

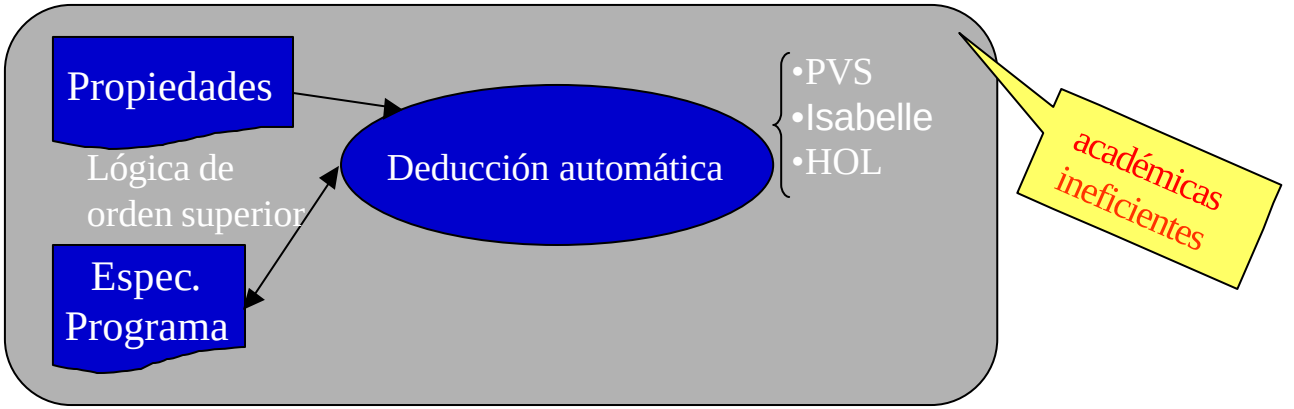
(Prog. Secuencial/Concurrente)

Otras Técnicas de Análisis Semi-Automático

- Técnicas de análisis a priori (sobre modelo)
  - Programación entera
  - Programación lógica
  - .....
- Técnicas de análisis posteriori (sobre código)
  - Pruebas de conformidad
  - Otros tipos de test

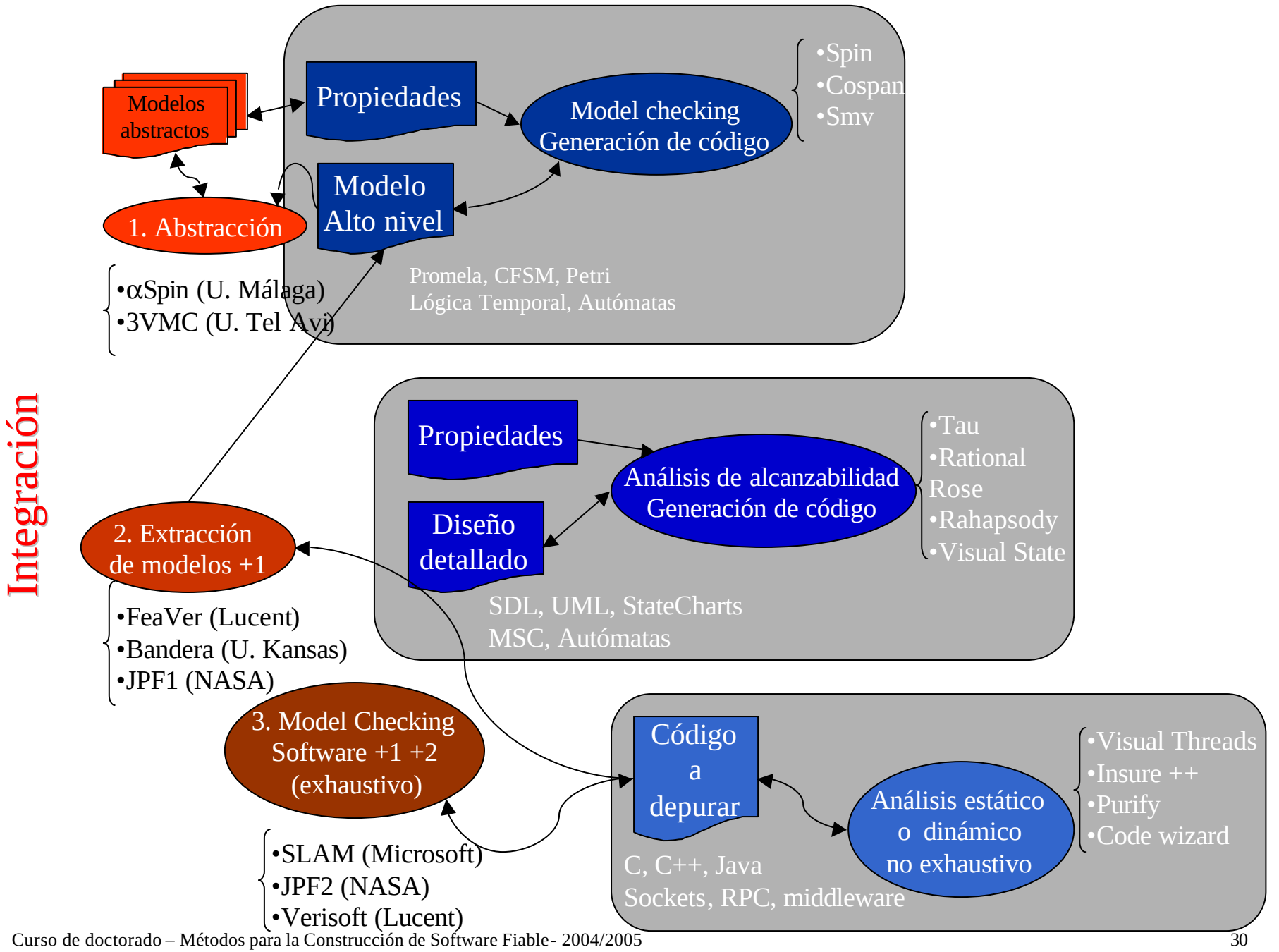
Revisión de Herramientas empleadas para Testing/Análisis Automático/Semi-automático


Nivel de Abstracción
Propiedades
Modelo
Alto nivel
Model checking
Generación de código
•Spin
•Cospan
•Smv
eficientes
académicas
Promela, CFSM, Petri
Lógica Temporal, Autómatas
Propiedades
Diseño
detallado
Análisis de alcanzabilidad
Generación de código
•Tau
•Rational
Rose
•Rahapsody
•Visual State
comerciales
ineficientes
SDL, UML, StateCharts
MSC, Autómatas
Código
a
depurar
Análisis estático
o dinámico
no exhaustivo
•Visual Threads
•Insure ++
•Purify
•Code wizard
C, C++, Java
Sockets, RPC, middleware


Propiedades
Lógica de
orden superior
Espec.
Programa
Deducción automática
•PVS
•Isabelle
•HOL
académicas
ineficientes

Nuevas Herramientas para Análisis Automático:

Análisis estático + Model checking + Demostración de Teoremas


Integración
Modelos abstractos
Propiedades
Model checking
Generación de código
•Spin
•Cospan
•Smv
1. Abstracción
Modelo Alto nivel
•αSpin (U. Málaga)
•3VMC (U. Tel Aviv)
Promela, CFSM, Petri
Lógica Temporal, Autómatas
Propiedades
Diseño detallado
Análisis de alcanzabilidad
Generación de código
•Tau
•Rational Rose
•Rahapsody
•Visual State
SDL, UML, StateCharts
MSC, Autómatas
2. Extracción de modelos +1
•FeaVer (Lucent)
•Bandera (U. Kansas)
•JPF1 (NASA)
3. Model Checking Software +1 +2 (exhaustivo)
•SLAM (Microsoft)
•JPF2 (NASA)
•Verisoft (Lucent)
Código a depurar
Análisis estático o dinámico no exhaustivo
•Visual Threads
•Insure ++
•Purify
•Code wizard
C, C++, Java
Sockets, RPC, middleware

Referencias

- Libros sobe análisis de software
  - F. Nielson, H.R. Nielson, C. Hankin "Principles of Program Analysis", 1998, Springer
  - G.H. Holzmann, "The SPIN Model Checker", 2004, Prentice-Hall
  - D. Peled "Software Reliability Methods", 2001,Springer
  - E. M. Clarke and O. Grumberg and D. A. Peled, "Model Checking", 2000, The MIT Press
  - B. Bérard et. al, *Systems and Software Verification*. *Model Checking Techniques and Tools*, 1999, Springer

- Artículos tipo "survey"
  - G. Clarke E. M., Wing J. M., Formal Methods: State of the Art and Future Directions, *ACM Workshop on Strategic Directions in Computing Research*, ACM Computing Surveys vol. 28(4): 626-643, 1996.
  - Gunter C., Mitchell J., Strategic Directions in Software Engineering and Programming Languages, *ACM Workshop on Strategic Directions in Computing Research*, ACM Computing Surveys, 28(4): 727-737, 1996.