Métodos formales en la ingeniería de software: en busca de la bala de plata

Carlos Gustavo Lopez Pombo[1]

[1]Departamento de computación
Facultad de ciencias exactas y naturales
Universidad de Buenos Aires

Ciclo charla de borrachos, 2004

Outline

Desastres de la ingeniería de software:

1. Explosión de la misión espacial Arine 5:
2. Algoritmo de división del procesador Pentium:
3. Orbitador climático de Marte:

Desastres de la ingeniería de software:

1. Explosión de la misión espacial Arine 5:
   - Perdidas: U$S 500 millones
2. Algoritmo de división del procesador Pentium:
3. Orbitador climático de Marte:

Desastres de la ingeniería de software:

1. Explosión de la misión espacial Arine 5:
   - Perdidas: U$S 500 millones
2. Algoritmo de división del procesador Pentium:
3. Orbitador climático de Marte:

Desastres de la ingeniería de software:

1. Explosión de la misión espacial Arine 5:
2. Algoritmo de división del procesador Pentium:
   - Perdidas: $>$ U$S 400 millones
3. Orbitador climático de Marte:

Desastres de la ingeniería de software:

1. Explosión de la misión espacial Arine 5:
2. Algoritmo de división del procesador Pentium:
   - Perdidas: $>$ U$S 400 millones
3. Orbitador climático de Marte:

Desastres de la ingeniería de software:

1. Explosión de la misión espacial Arine 5:
2. Algoritmo de división del procesador Pentium:
3. Orbitador climático de Marte:
   - Perdidas: U$S 240 millones

Desastres de la ingeniería de software:

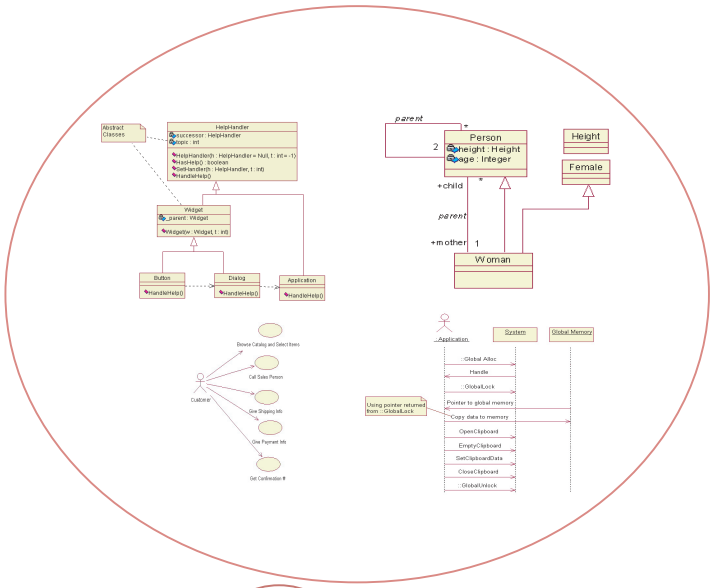
1. Explosión de la misión espacial Arine 5:
2. Algoritmo de división del procesador Pentium:
3. Orbitador climático de Marte:
   - Perdidas: U$S 240 millones

Modelado de sistemas

Sólo una idea...

Modelado de sistemas

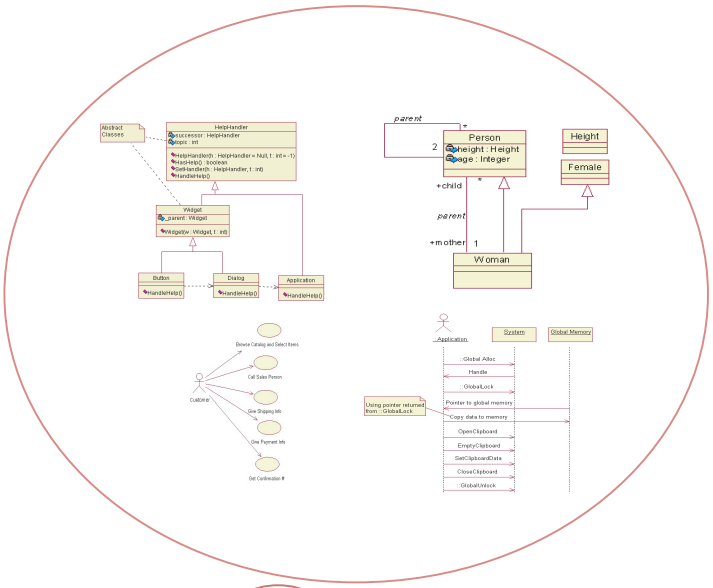
Ejemplo: diseño con *UML*



DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

Modelado de sistemas

Ejemplo: diseño con *Z*



$$
\begin{array}{|l}
\textit{ASchedulerInit} \\
\hline
\textit{AScheduler}' \\
\hline
current' = nullPid \\
ready' = \emptyset \\
blocked' = \emptyset \\
free' = Pid
\end{array}
$$

$$
\begin{array}{|l}
\textit{ATimeOut} \\
\hline
\Delta ASchedaler \\
p! : Pid \\
\hline
current \neq nullPid \\
current' = nullPid \\
ready' = ready \cup \{current\} \\
blocked' = blocked \\
free' = free \\
p! = current
\end{array}
$$

$$
\begin{array}{|l}
\textit{ADispatch} \\
\hline
\Delta ASchedaler \\
p! : Pid \\
\hline
current = nullPid \\
ready \neq \emptyset \\
current' \in ready \\
ready' = ready \setminus \{current'\} \\
blocked' = blocked \\
free' = free \\
p! = current'
\end{array}
$$

DEPARTAMENTO DE COMPUTACIÓN
Facultad de Ciencias Exactas y Naturales - UBA

Logros de la aplicación de métodos formales en la ingeniería de software:

1. Météor, Matra Transport, France:
2. SPOT 4, CS-CI, France:
3. Procesador CISC AAMP5, SRI International, EEUU:

Logros de la aplicación de métodos formales en la ingeniería de software:

1. Météor, Matra Transport, France:
   - Herramienta: Atelier-B
   - Resultado: Se redujo el costo para el software crítico
     - No se encontró ningún error durante el testing de 80000 lineas de código
     - El costo estimado para las reformas estructurales de seguridad supera ampliamente el costo que significó la verificación formal
2. SPOT 4, CS-CI, France:
3. Procesador CISC AAMP5, SRI International, EE

Logros de la aplicación de métodos formales en la ingeniería de software:

1. Météor, Matra Transport, France:
   - Herramienta: Atelier-B
   - Resultado: Se redujo el costo para el software crítico
     - No se encontró ningún error durante el testing de 80000 lineas de código
     - El costo estimado para las reformas estructurales de seguridad supera ampliamente el costo que significó la verificación formal
2. SPOT 4, CS-CI, France:
3. Procesador CISC AAMP5, SRI International, EE

Logros de la aplicación de métodos formales en la ingeniería de software:

1. Météor, Matra Transport, France:
   - Herramienta: Atelier-B
   - Resultado: Se redujo el costo para el software crítico
     - No se encontró ningún error durante el testing de 80000 lineas de código
     - El costo estimado para las refomas estructurales de seguridad supera ampliamente el costo que significó la verificación formal
2. SPOT 4, CS-CI, France:
3. Procesador CISC AAMP5, SRI International, EE

Logros de la aplicación de métodos formales en la ingeniería de software:

1. Météor, Matra Transport, France:
2. SPOT 4, CS-CI, France:
   - Herramienta: IFAD VDM-SL Toolbox
   - Resultado: Se redujo el costo significativamente:
     - 38 % menos de código fuente
     - 36 % menos de esfuerzo total
     - Generación automática de código C++
3. Procesador CISC AAMP5, SRI International, EEUU:

Logros de la aplicación de métodos formales en la ingeniería de software:

1. Météor, Matra Transport, France:
2. SPOT 4, CS-CI, France:
   - Herramienta: IFAD VDM-SL Toolbox
   - Resultado: Se redujo el costo significativamente:
     - 38 % menos de código fuente
     - 36 % menos de esfuerzo total
     - Generación automática de código C++
3. Procesador CISC AAMP5, SRI International, EEUU:

Logros de la aplicación de métodos formales en la ingeniería de software:

1. Météor, Matra Transport, France:
2. SPOT 4, CS-CI, France:
   - Herramienta: IFAD VDM-SL Toolbox
   - Resultado: Se redujo el costo significativamente:
     - 38 % menos de código fuente
     - 36 % menos de esfuerzo total
     - Generación automática de código $C++$
3. Procesador CISC AAMP5, SRI International, EEUU:

Logros de la aplicación de métodos formales en la ingeniería de software:

1. Météor, Matra Transport, France:
2. SPOT 4, CS-CI, France:
3. Procesador CISC AAMP5, SRI International, EEUU:
   - Herramienta: *PVS*
   - Resultado: Fue posible probar la correctitud (respecto de la especificación) de una arquitectura dividida en cuatro unidades independientes, consistente en 500000 transistores, y un conjunto de 108 instrucciones.

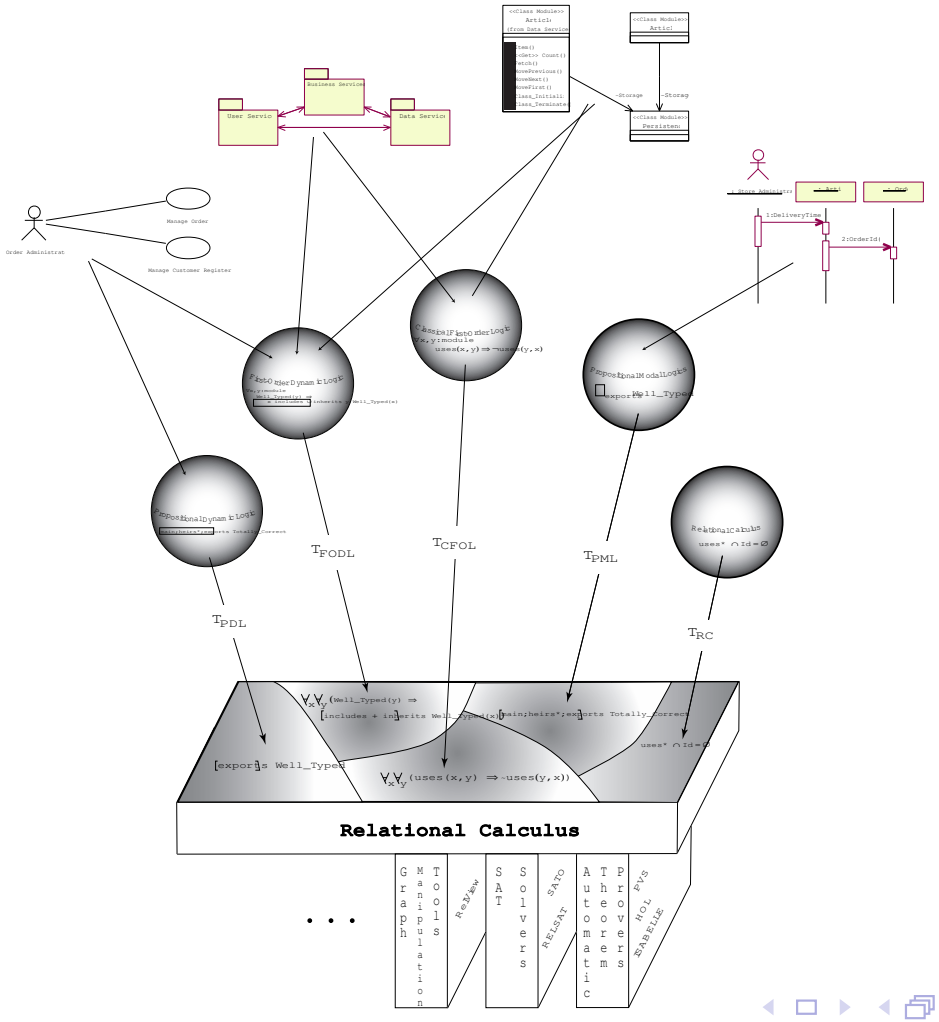
Logros de la aplicación de métodos formales en la ingeniería de software:

1. Météor, Matra Transport, France:
2. SPOT 4, CS-CI, France:
3. Procesador CISC AAMP5, SRI International, EEUU:
   - Herramienta: *PVS*
   - Resultado: Fue posible probar la correctitud (respecto de la especificación) de una arquitectura dividida en cuatro unidades independientes, consistente en 500000 transistores, y un conjunto de 108 instrucciones.

Logros de la aplicación de métodos formales en la ingeniería de software:

1. Météor, Matra Transport, France:
2. SPOT 4, CS-CI, France:
3. Procesador CISC AAMP5, SRI International, EEUU:
   - Herramienta: *PVS*
   - Resultado: Fue posible probar la correctitud (respecto de la especificación) de una arquitectura dividida en cuatro unidades independientes, consistente en 500000 transistores, y un conjunto de 108 instrucciones.

El proyecto **Ar$_g$entum**




Marcelo F. Frias and Carlos G. Lopez Pombo.
Interpretability of first-order linear temporal logics in fork algebras.
Journal of Logic and Algebraic Programming, 2004.
in press.


Juan P. Galeotti and Mario Roman.
Reasoning across dynamic logic and linear temporal logic using fork algebras.
Master's thesis, October 2004.
Advisors: Marcelo F. Frias and Carlos G. Lopez Pombo.


Carlos G. Lopez Pombo, Sam Owre, and Natarajan Shankar.
A semantic embedding of the $\mathbf{A_g}$ dynamic logic in PVS.
Technical Report SRI-CSL-02-04, Computer Science Laboratory, SRI International, July 2002.