

UML para Embedded Systems



Fábio Amado 33637
João Maio 33306

Índice

1. UML - o que é?
2. A Natureza dos Sistemas Embutidos
 1. Heterogeneidade
 2. Composição
 3. Complexidade, Reutilização e Síntese
 4. Contexto do Sistema
3. Requisitos para Metodologias de Projecto
4. UML: capacidades e falhas
 1. O que é que o UML tem ou vai ter?
 2. Falhas e Necessidades do UML
5. “Plataforma-UML”
6. Conclusão
7. Bibliografia

UML - o que é?

- A Unified Modeling Language (UML) é uma linguagem de modelação standard no campo da engenharia de software. O standard foi criado e é gerido pelo Object Management Group



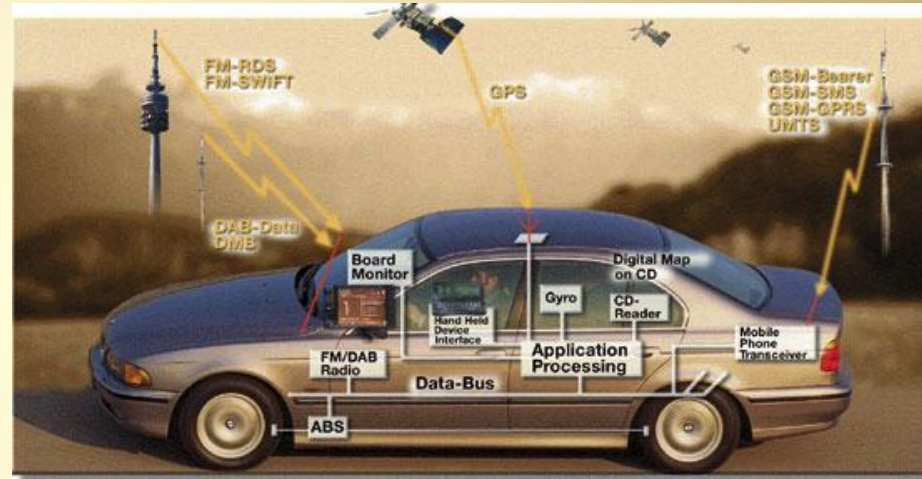
- A UML inclui um conjunto de técnicas de notação gráfica que permite criar modelos visuais de sistemas do tipo software-intensive

A Natureza dos Sistemas Embutidos (1)

- As características dos sistemas embutidos modernos, exigem novas abordagens para a sua:
 - Especificação
 - Projecto
 - Implementação
- As abordagens devem ser uma combinação de:
 - Métodos tradicionais de conjugar hardware e software
 - Novas formas de os interligar
- Isto possibilita o rápido desenvolvimento com baixo risco para os projectistas

A Natureza dos Sistemas Embutidos (2)

- Sistemas embutidos são compostos por múltiplos subsistemas ou unidades funcionais
- Usam um conjunto heterogéneo de modelos para levar a cabo a comunicação e a computação
- O mapeamento das funções para arquiteturas não é fixo. Isto permite aos developers encontrar formas óptimas de as implementar (quer no domínio do hardware, quer no do software)



Heterogeneidade (1)

- Sistemas embutidos passaram de funções simples, para produtos que desempenham diversas funcionalidades



- A convergência das várias áreas de computação, processamento de sinal e comunicação, levou a sistemas embutidos cada vez mais complexos

Heterogeneidade (2)

- Modelar estes sistemas requer um conjunto de noções de:
 - Tempo contínuo
 - Máquina de estados finitos
 - Dataflow
 - Evento discreto
 - Reactividade
- Para além destas implementações lógicas, existem outras de domínio físico como:
 - Microprocessadores
 - DSPs
 - Componentes analógicos
 - Blocos de HW digitais e de RF
 - Micro-Electro-Mechanical Systems (MEMS)

Composição

- Os sistemas embutidos modernos não são modelados nem implementados de origem.
- Quer a descrição do funcionamento como a da arquitectura, podem ser baseadas numa composição de subsistemas
- Com uma estratégia de projecto deste género, deve-se dar ênfase à validação da composição assim como à correcção das partes constituintes

Complexidade, Reutilização e Síntese

- Elevada complexidade
- Requisitos funcionais, exigem diversos conhecimentos de projecto
- Forte necessidade de maximizar o software através de plataformas que reutilizem e sintetizem métodos que derivem de “system-level models”

Contexto do Sistema

Deve ser capaz de modelar as:

- Funções do sistema e especificidades de execução
- Interacções quer com o ambiente quer com o utilizador



Requisitos para Metodologias de Projecto

É necessária uma metodologia que suporte as seguintes características:



- Modelação heterogénea
- Modelar o sistema no seu contexto de uso (*use-context*)
contexto de sistema (environment)
- *Mapeamento entre o desenho e a implementação*
- *Ferramentas robustas de verificação*
- *Ligação entre os modelos HW/SW*

O que é que o UML tem ou vai ter?

- A UML não é uma apenas uma linguagem

- Mecanismos de extensão

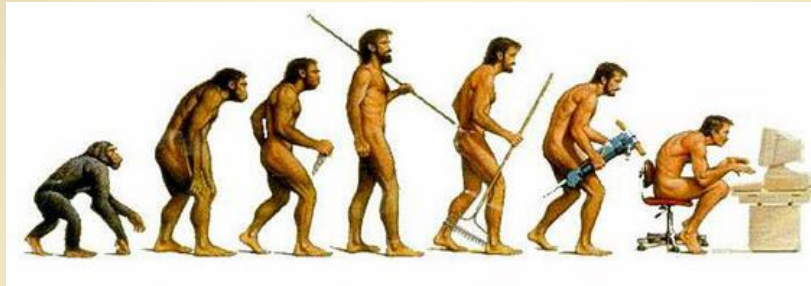


- *Modelação use-case*
- *Standard*

- *Orientada a objectos*
- *Interesse crescente*
- *Máquina de estados*
- *Decomposição estruturada*

O que é que o UML tem ou vai ter? (3)

No entanto, a UML está em constante evolução



- Melhorias no campo do tempo real (modelação de escalabilidade, performance e tempo)
- Planos para suportar SDL
- Semântica de acção
- Arquitectura direccionada ao modelo, e não à plataforma de implementação

Falhas e Necessidades do UML (1)



- Com a actual UML 1.4, e com a futura revisão 2.0, quais são as falhas principais para suportar o projecto de sistemas embutidos?

Falhas e Necessidades do UML (2)



- Modelar a plataforma
- A metodologia de ligação entre níveis da plataforma e seu melhoramento
- Definição de restrições e metodologias de orçamentação

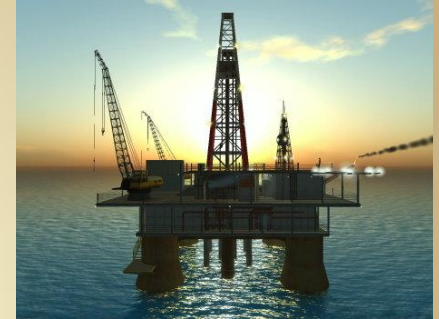
“Plataforma-UML” (1)



- Desenvolvimento de trabalho para implementar SoC em UML
- Propostas de extensões para a UML 2.0:
Conceito “plataforma-UML”, descreve o seguinte de conjuntos de estereótipos e *tags* como extensões:

“Plataforma-UML” (2)

- *Estereótipos uses e needs (relações entre aplicações/componentes)*
- *Estereótipo stack (camadas hierárquicas)*
- *Relações de coupling (ligações entre componentes)*
- *Tags para definir parâmetros (QoS, serviço e requisitos da aplicação)*



“Plataforma-UML” (3)



- *Camadas definidas na plataforma (ASP,API,ARC)*
- *Ponte entre target-independent e platform-based*

Aplicação futura do conceito em plataformas SoC comerciais.

Conclusão

- UML como base de projecto de sistemas embutidos
- Actualmente incompleta em ferramentas e metodologias
- Alterações propostas para a UML 2.0 não respondem a todas as necessidades
- O conceito “plataforma-UML”, deve permitir muitos dos atributos adicionais necessários para o desenho de sistemas embutidos e de tempo real

Bibliografia

- http://www.date-conference.com/archive/conference/proceedings/PAPERS/2002/DATE02/PDFFILES/08A_2.PDF
- http://en.wikipedia.org/wiki/Unified_Modeling_Language
- http://en.wikipedia.org/wiki/Object_Management_Group

Perguntas

