Projeto de Sistemas Ciberfísicos

Fernando Silvano Gonçalves e Leandro Buss Becker
Programa de PG em Automação e Sistemas (PGEAS)
Universidade Federal de Santa Catarina (UFSC)

Tópicos do Curso

- Caracterização dos Sistemas Ciberfísicos (CPS)
  - Problemas e Desafios Associados ao Desenvolvimento de CPSs

Caracterização dos Sistemas Ciberfísicos (CPS)

- CPS é um sistema que combina e coordena, sist. computacionais com elementos físicos
  - Utiliza computação e comunicação embarcados, interagindo com os processos físicos para adicionar novas propriedades aos mesmos
  - Varia desde sistemas miniaturizados até sistemas de larga escala
- Convergência entre computação, comunicação e controle

Nesse tipo de cyber inexiste atuação, controle

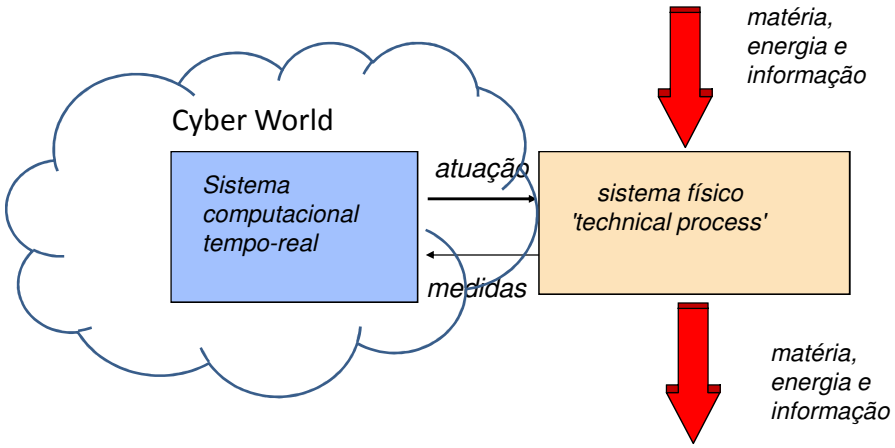
Caracterização dos Sistemas Ciberfísicos (CPS)

- Portanto, não confundir CPS com IoT (*Internet-of-Things*)
- CPS é a convergência entre computação, comunicação e **controle**
- IoT é um nível acima, e que não envolve necessariamente controle

Caracterização: CPS ou RTS?

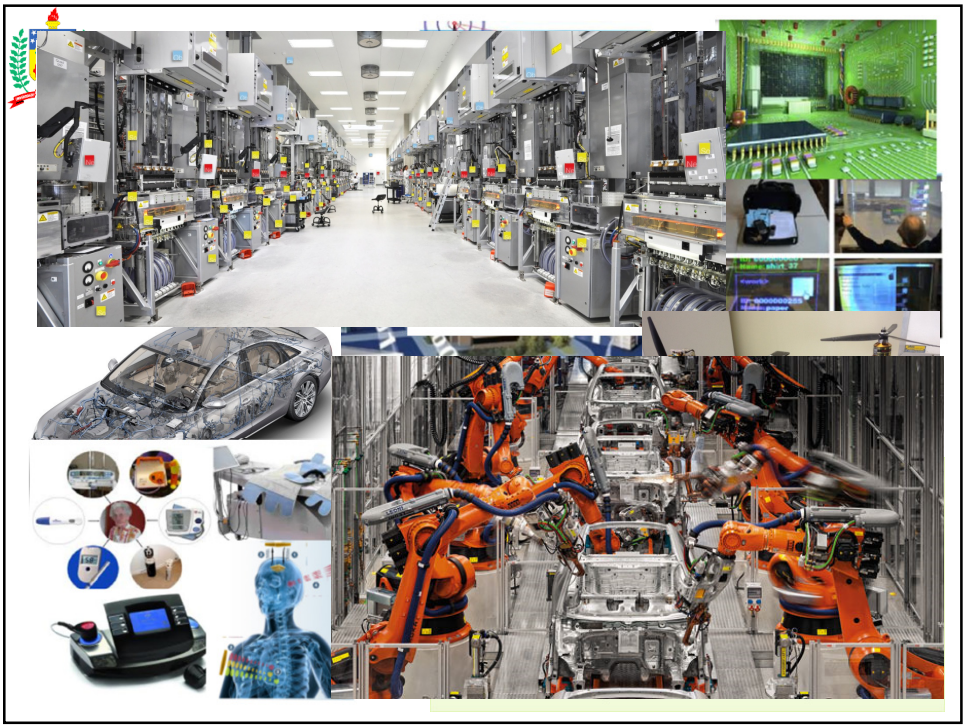
- Na década de 80 se fortaleceu a discussão sobre Sistemas de Tempo Real (RTS):
  - John A. Stankovic. **Misconceptions About Real-Time Computing: A Serious Problem for Next-Generation Systems**. Computer, v.21, pp.10-19, October 1988.

Sistemas Ciberfísicos (CPS)



Caracterização dos Sistemas Ciberfísicos (CPS)

- STR de ontem (em grande parte) são os CPS de hoje!
  - Nada se cria, tudo se copia...



Cyber-Physical Systems (CPS)

Projeto ProVant

http://provant.das.ufsc.br

Universidade Federal de Santa Catarina

UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS

Problemas e Desafios Associados ao Desenvolvimento de RTS

- Stankovic elencou os seguintes tópicos como pertinentes ao desenvolvimento de **RTS**:
  - Specification and verification
  - Scheduling theory
  - Operating systems
  - Programming languages and design methodologies
  - ~~Distributed databases~~
  - ~~Artificial intelligence~~
  - Fault tolerance
  - Architectures, and
  - Communication

} Perderam força no contexto dos STR, mas continuam em pauta nos CPS

Problemas e Desafios Associados ao Desenvolvimento de CPS

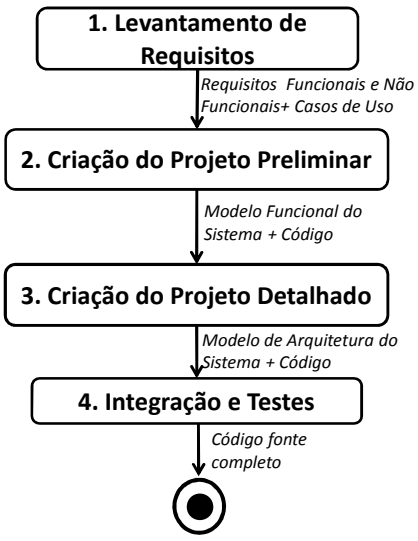
- Relendo os tópicos de Stankovic, podemos acrescentar o seguinte aos CPS:
  - Executable specification (simulation), and formal verification
  - Scheduling theory + (Worst case) execution time estimation
  - Operating systems
  - Programming languages + model-based design methodologies
  - Distributed databases
  - Artificial intelligence
  - Fault tolerance
  - Automatic code generation
  - Architectures, and
  - Communication

Tópicos do Curso

- Caracterização dos Sistemas Ciberfísicos (CPS)
  - Problemas e Desafios Associados ao Desenvolvimento de CPSs
- Processo de Desenvolvimento de Software para CPS
  - Ferramentas relacionadas às etapas de desenvolvimento

u1

Processo de Desenvolvimento de Software para CPS



Slide 14

u1 este é o processo antigo, mudar para o que aparece na figura
user; 28/10/2014

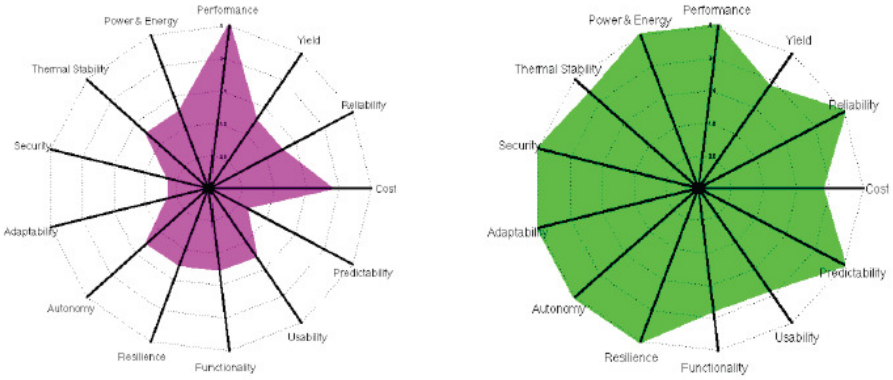
Processo de Desenvolvimento de Software para CPS

1. Levantamento de Requisitos

Requisitos

- *Requisitos funcionais* correspondem à listagem de todas as coisas que o sistema deve fazer
- *Requisitos não funcionais* são restrições que se coloca sobre como o sistema deve realizar seus requisitos funcionais

Características dos Requisitos Não Funcionais



Especificação de Requisitos

- Apenas uma linguagem de modelagem suporta a especificação de requisitos:
  - **System Modeling Language (SysML)**: *Requirements Diagram*
- Nas demais linguagens de modelagem, o suporte encontra-se nas ferramentas (tools)
  - E.g. DOORS

Requisitos Funcionais e Não Funcionais

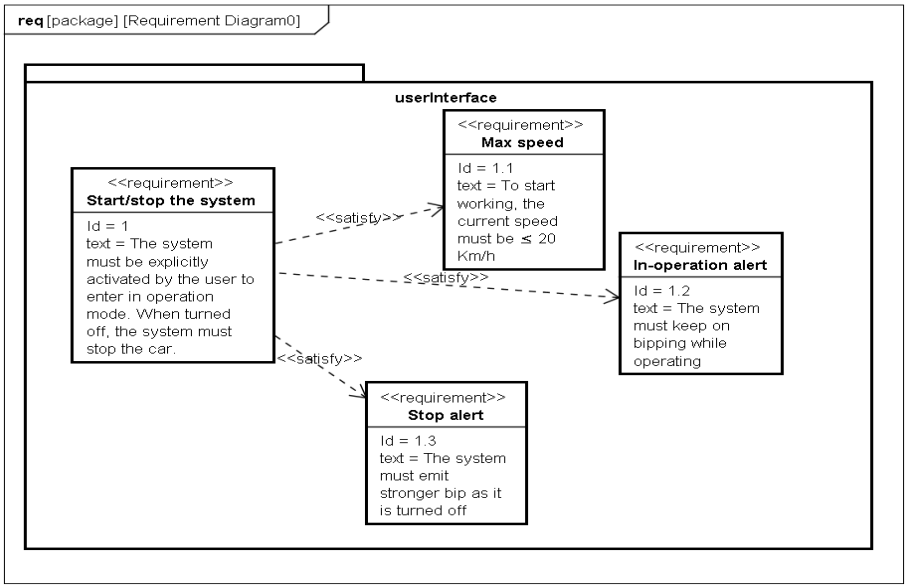
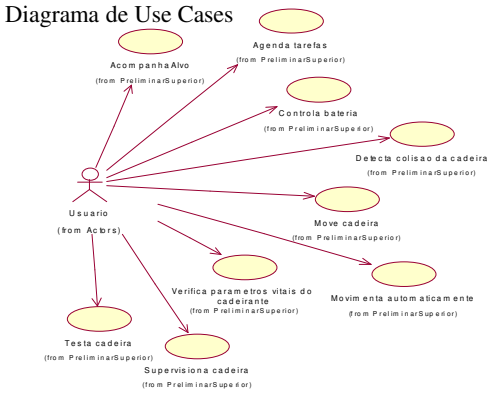


Diagrama de Casos de Uso

- Diagrama da UML, útil para representar as interações do sistema com o mundo externo



Processo de Desenvolvimento de Software para CPS

Modelagem Funcional

- Deve expressar as funcionalidades do sistema de maneira abstrata
  - Suporte à simulação é essencial em muitos casos
- Preferencialmente, deve ser independente de plataforma
  - Independência de plataforma não significa ignorância de plataforma

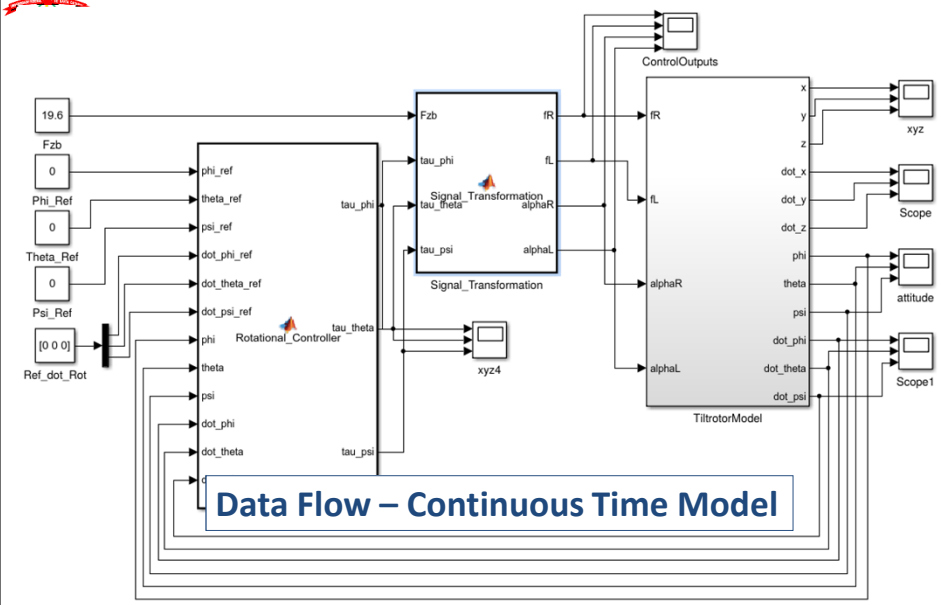
Modelagem Funcional

- Suporte à simulação vem muito mais do ambiente de desenvolvimento do que da linguagem de modelagem em si:
  - Simulink, LabView (Diagrama em Blocos - Dataflow e Continuous time)
  - Ptolemy (Diagrama em Blocos - Dataflow e Continuous time, etc)
  - Scade (Lustre)
  - Rational Software Architect RealTime (exec. UML)

Simulação de Modelos

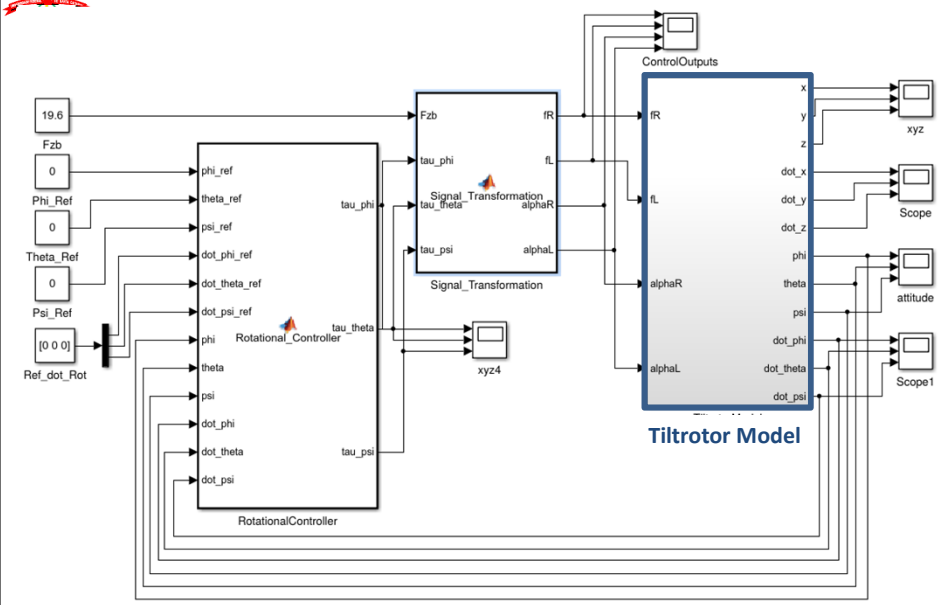
- Para simular um CPS se faz necessário modelar a dinâmica do processo físico sendo controlado
- Forma de se fazer isso varia de acordo com a ferramenta utilizada...

Diagrama em Blocos Dataflow (Simulink)



Exemplo de Simulação Simulink

Diagrama em Blocos Dataflow (Simulink)



Simulação Funcional

- Simulação requer o modelo da planta física do sistema

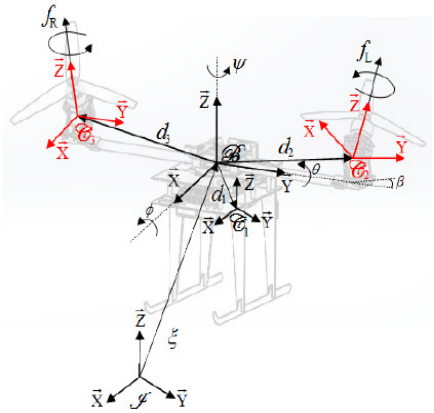
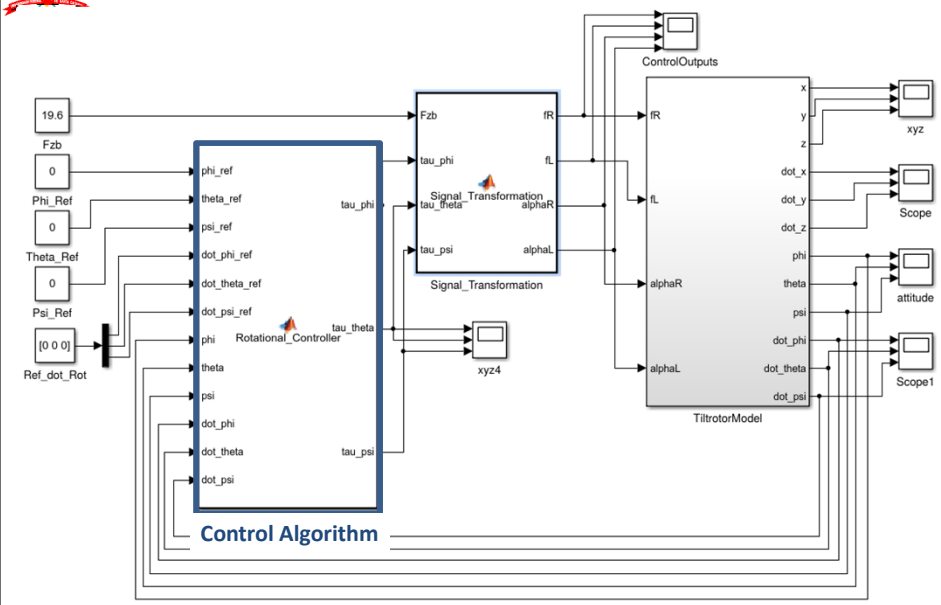


Diagrama em Blocos Dataflow (Simulink)



Algoritmo de Controle

- Equações do algoritmo de controle

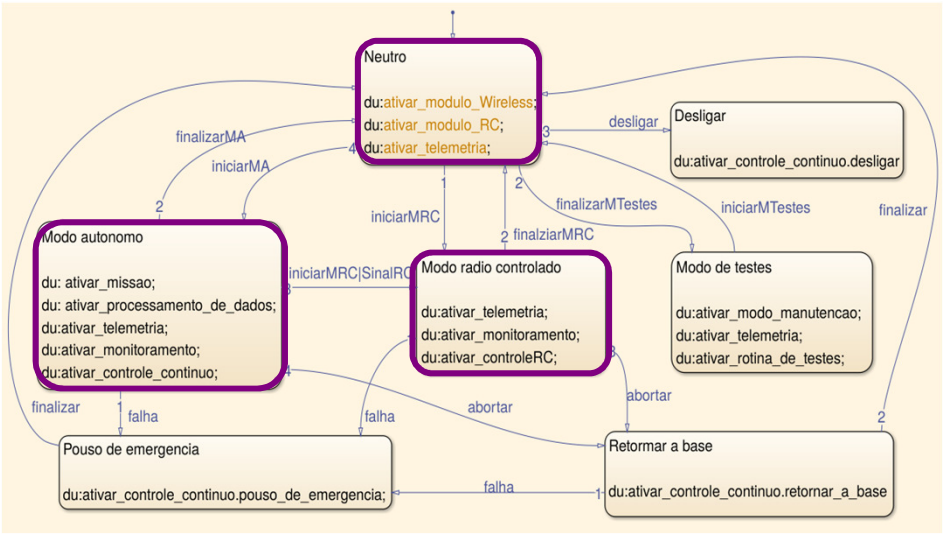
$$f_L = \frac{1}{2}\sqrt{\left(\frac{\tau_\theta}{h} + \frac{\tau_\psi}{l}\right)^2 + \left(F_{zb} - \frac{\tau_\phi}{l}\right)^2},$$

$$f_R = \frac{1}{2}\sqrt{\left(\frac{\tau_\theta}{h} - \frac{\tau_\psi}{l}\right)^2 + \left(F_{zb} + \frac{\tau_\phi}{l}\right)^2}.$$

$$\alpha_L = \arctan\left(\frac{\frac{\tau_\theta}{h} + \frac{\tau_\psi}{l}}{F_{zb} - \frac{\tau_\theta}{l}}\right),$$

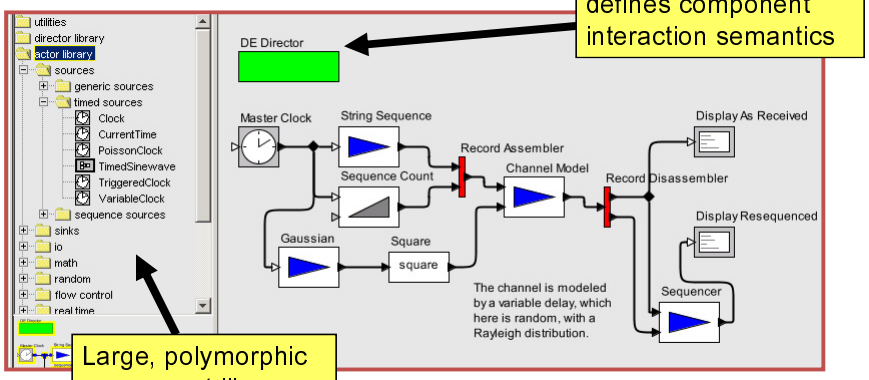
$$\alpha_R = \arctan\left(\frac{\frac{\tau_\theta}{h} - \frac{\tau_\psi}{l}}{F_{zb} + \frac{\tau_\theta}{l}}\right).$$

Diagrama de Estados



Ptolemy Project Principles

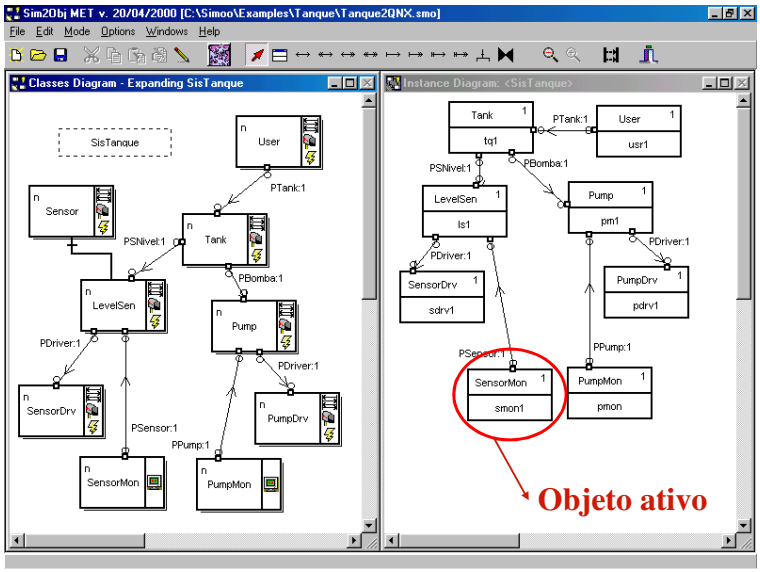
Basic Ptolemy II infrastructure:



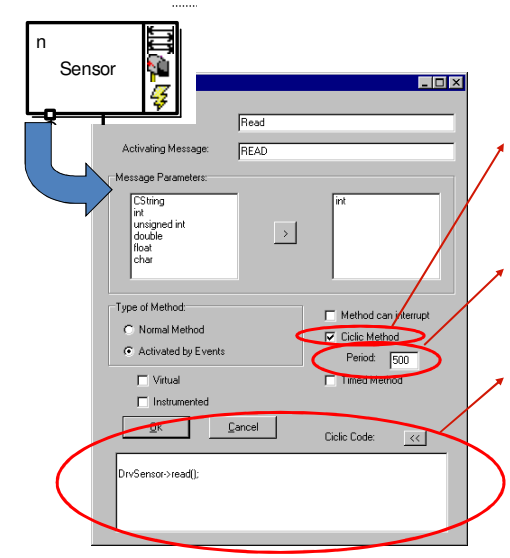
Ferramentas OO (UML like)

- OO se tornou popular para modelagem de RTS a partir da década de 90
- Mix entre modelo funcional e modelo de arquitetura
- Exemplo de ferramentas:
  - ObjectTime (hoje IBM Rational Real-Time Suite)
  - (Artisan) Real-Time Studio
  - Papyrus UML
  - SIMOO-RT

Modelagem Arquitetura: Diagramas de Classes e Instâncias

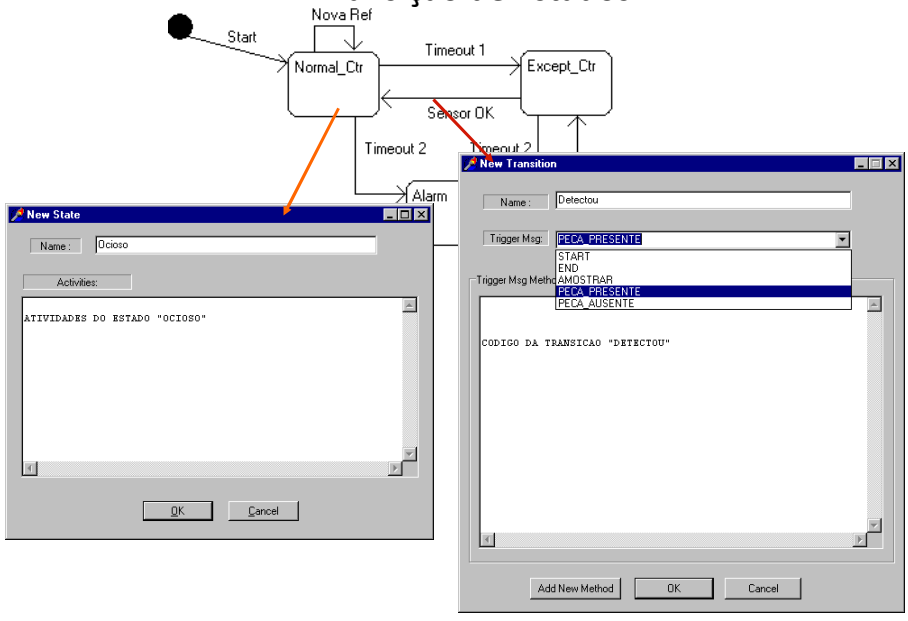


Especificações temporais: atributos especiais

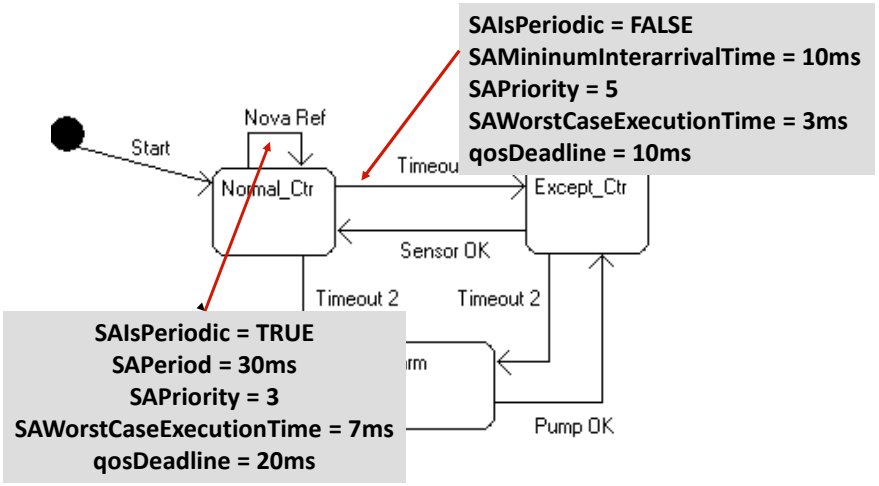


- **Operações cíclicas**
  «SAIsPeriodic»:= TRUE
- **Valor default para o ciclo**
  «SAPeriod»:= 500ms
- **Código cíclico**

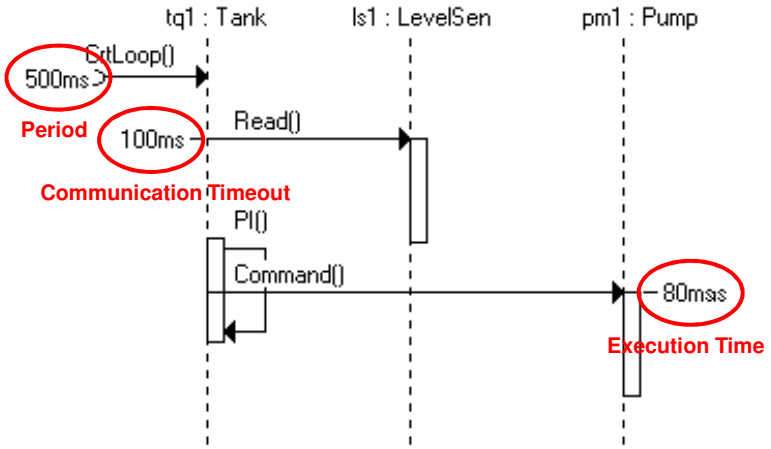
Especificação Comportamento: Diagrama de Transição de Estados



Especificação Comportamento:
Diagrama de Transição de Estados



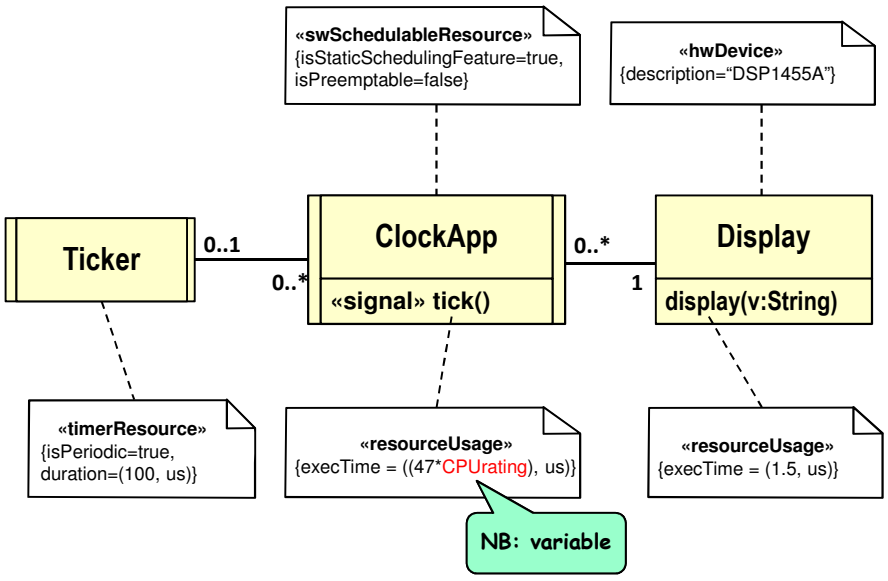
Modelagem Funcional: Diagrama de Sequência



Linguagem UML

- Atualmente é o padrão incontestável para a modelagem de sistemas OO
- Perfil MARTE (Modeling and Analysis of Real-Time and Embedded Systems) amadurecido
  - Disponível na maioria das ferramentas comerciais

Modelo UML usando MARTE



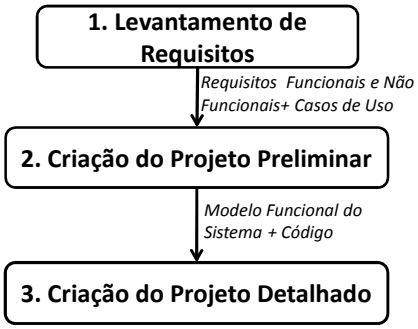
Geração de Código para Modelo Funcional

- Gerar o código(s) do sistema (controle), de preferência automaticamente
- Optar por código independente de plataforma
- Possíveis variações:
  - Linguagem de programação + bibliotecas matemáticas
  - Tipo de processador (com ou sem ponto flutuante)

Modelagem Funcional: Resumo

- No contexto de CPS, simulação é mandatório
- HH da empresa desenvolvedora da ferramenta CASE faz a diferença para facilitar simulações
  - Ferramentas originalmente criadas para simulação levam vantagem
  - Acurácia das simulações é diretamente proporcional à qualidade do modelo da planta
- Resulta em modelo + código (independente de plataforma)

Processo de Desenvolvimento de Software para CPS



Diferenças entre Projeto Preliminar e Projeto Detalhado

Projeto Preliminar	Projeto Detalhado
1. Modelo conceitual, abstração, não contém detalhes de implem.	1. Modelo físico, "planta" da implementação
2. Genérico – pode derivar em vários projetos	2. Concreto, específico para uma implement.
3. Menos formal	3. Mais formal
4. Pode ser descartado ao longo do ciclo de vida	4. Deve ser mantido ao logo do ciclo de vida

Atividades Projeto Detalhado $^{1}/_{3}$

1. Definir a estrutura ideal de hw/sw
2. Projetar as interfaces para componentes externos (hardware, software e usuário)
3. Decidir entre processamento centralizado ou distribuído
4. Determinar concorrências entre tarefas

Atividades Projeto Detalhado $^{2}/_{3}$

5. Determinar estratégias de armazenamento de dados, manutenção e alocação de memória
6. Projetar banco de dados e estruturas de manutenção
7. Projetar mecanismos de inicialização e desligamento do sistema
8. Projetar algoritmos e funções de processamento de dados

Atividades Projeto Detalhado $^{3}/_{3}$

9. Projetar tratamento de erros
10. Conduzir analise de desempenho
11. Projetar software a ser usado nos testes
12. Escrever a documentação

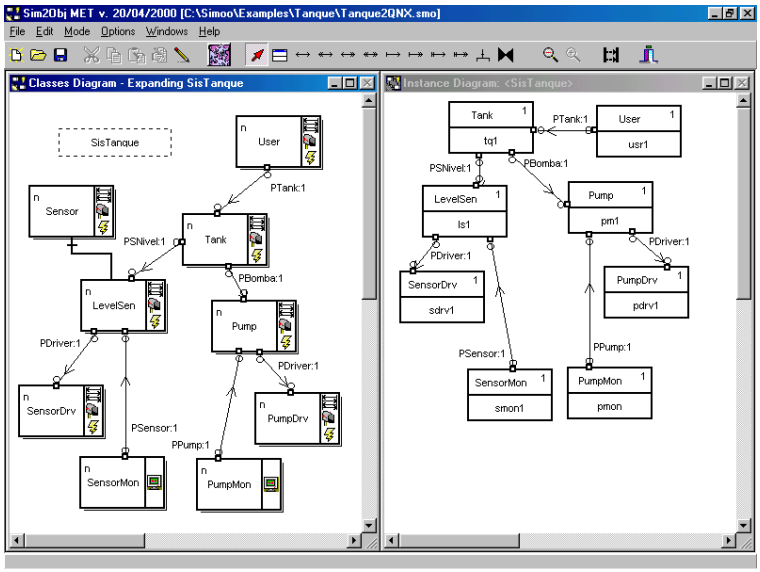
–Manual do Usuário

–Manual do Programador

Linguagem UML

- Devido ao seu caráter genérico UML + MARTE é adequada
- SysML também aparece como alternativa
- Pros: Transição “suave” entre projeto preliminar e projeto detalhado
- Contras: Limitada em termos de formalismos e análises

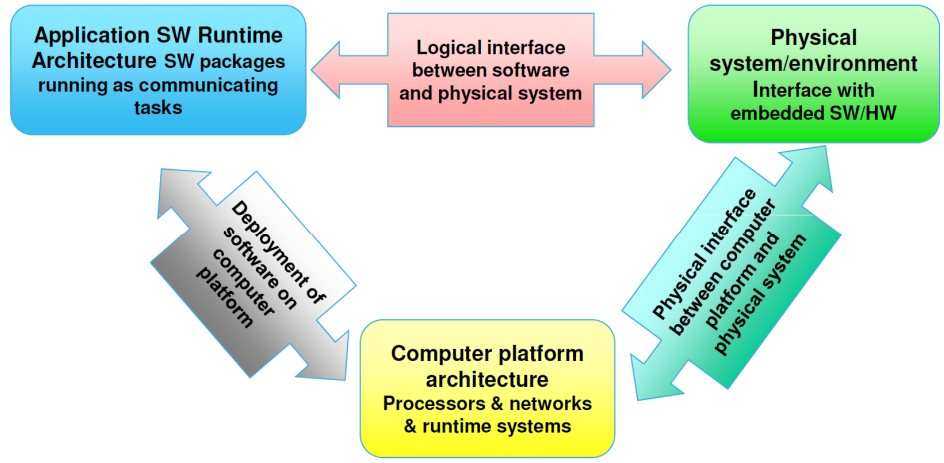
Diagramas de Classes e Instâncias



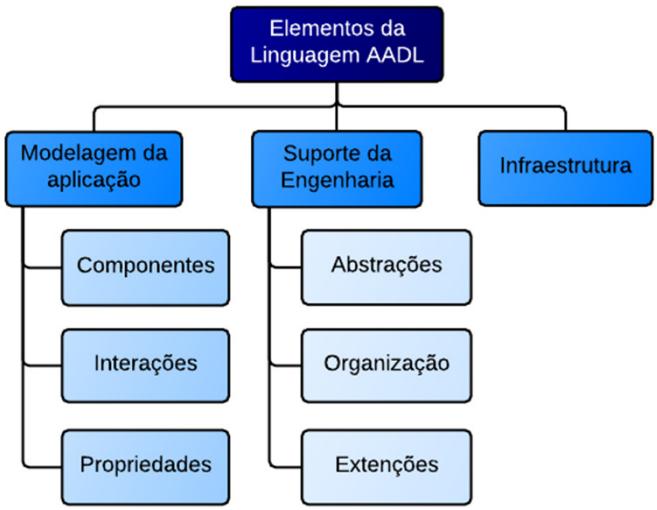
Linguagem AADL

- Alternativa para a criação do projeto detalhado
- Desenvolvido pela SAE em 2004 para auxiliar o projeto da arquitetura de sistemas embarcados
- Padrão de modelagem e desenvolvimento voltado a diferentes comunidades
- Linguagem textual com representação gráfica
- Notação baseada em componentes

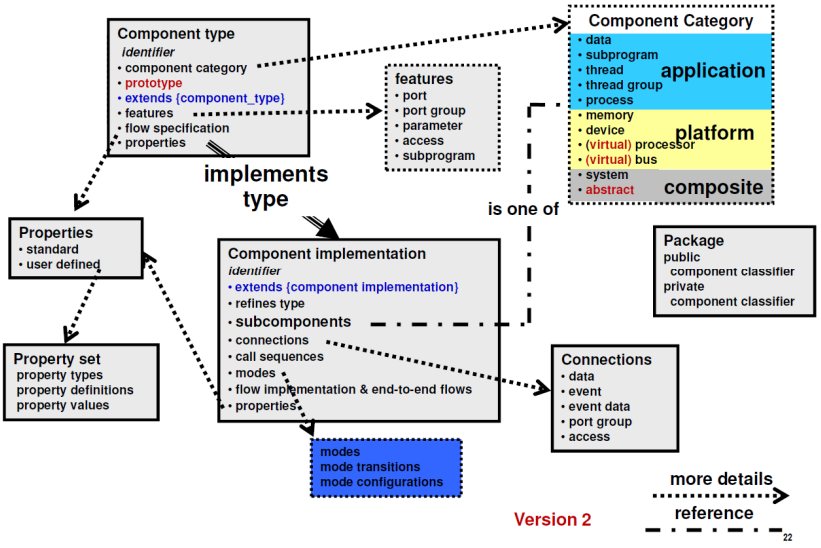
Elementos da Linguagem AADL



Elementos da Linguagem AADL

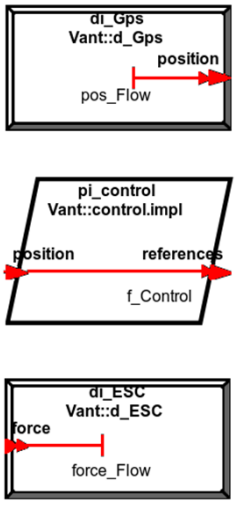


AADL – Componentes e Conexões



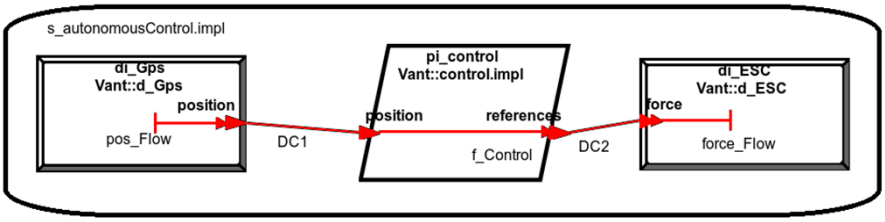
Representação AADL

```
device d_Gps
    features
        position : out data port ;
    flows
        pos_Flow : flow source position;
end d_Gps;

process implementation control.impl
    connections
        DC1 : port position -> references;
    flows
        f_Control : flow path position -> DC1 -> references;
end control.impl;

device d_ESC
    features
        force : in data port ;
    flows
        force_Flow : flow sink force;
end d_ESC;
```



Representação AADL

```
system implementation s_autonomousControl.impl
    ...
    flows
        forceControl : end to end flow di_Gps.pos_Flow -> DC1
        -> pi_control.f_Control -> DC2 -> di_ESC.force_Flow;
end s_autonomousControl.impl;
```



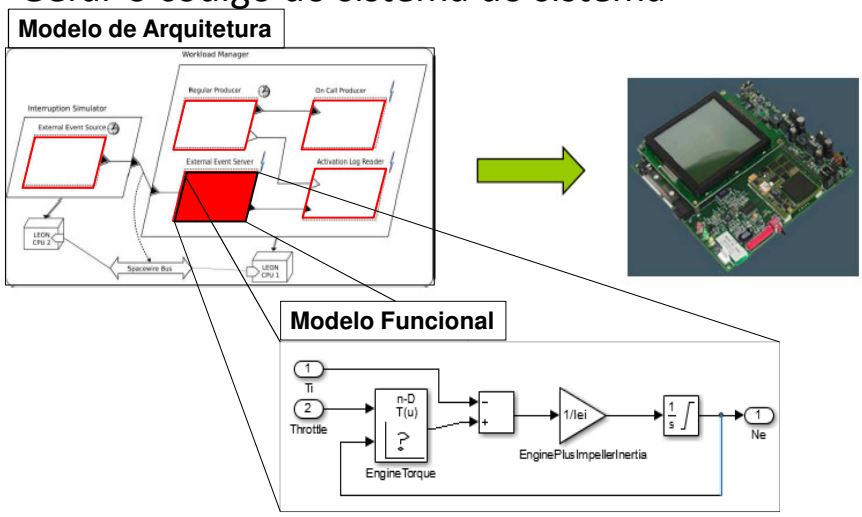
Representação AADL

```
thread implementation t_control.impl
    properties
        Dispatch_Protocol => periodic;
        Period => 5 ms;
        Deadline => 5ms;
end t_control.impl;

device d_Gps
    features
        position : out data port;
    properties
        Dispatch_Protocol => periodic;
        Period => 100 ms;
        Deadline => 100 ms;
end d_Gps;
```

```
processor p_Arm7
        features
            channel: requires bus access rs232;
        properties
            SEI::MIPSCapacity => 2000.0 MIPS;
end p_Arm7;
```

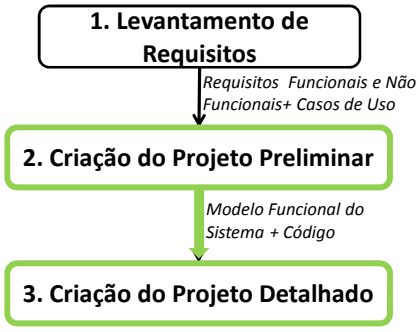
Geração de Código para Modelo Arquitetural

- Gerar o código do sistema do sistema



u2

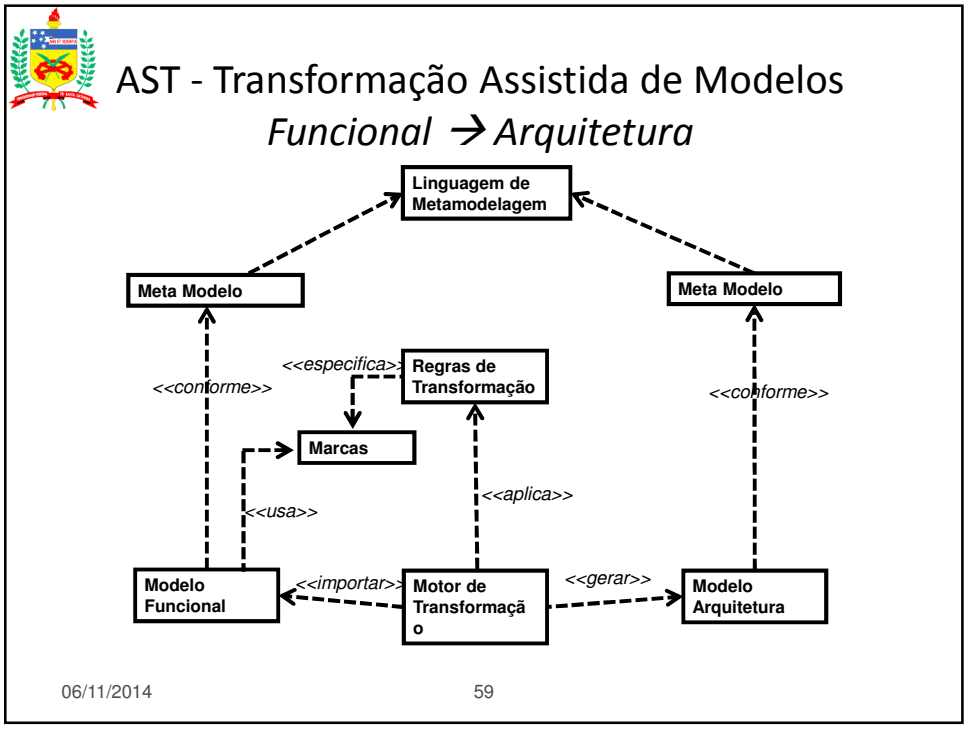
Processo de Desenvolvimento de Software para CPS

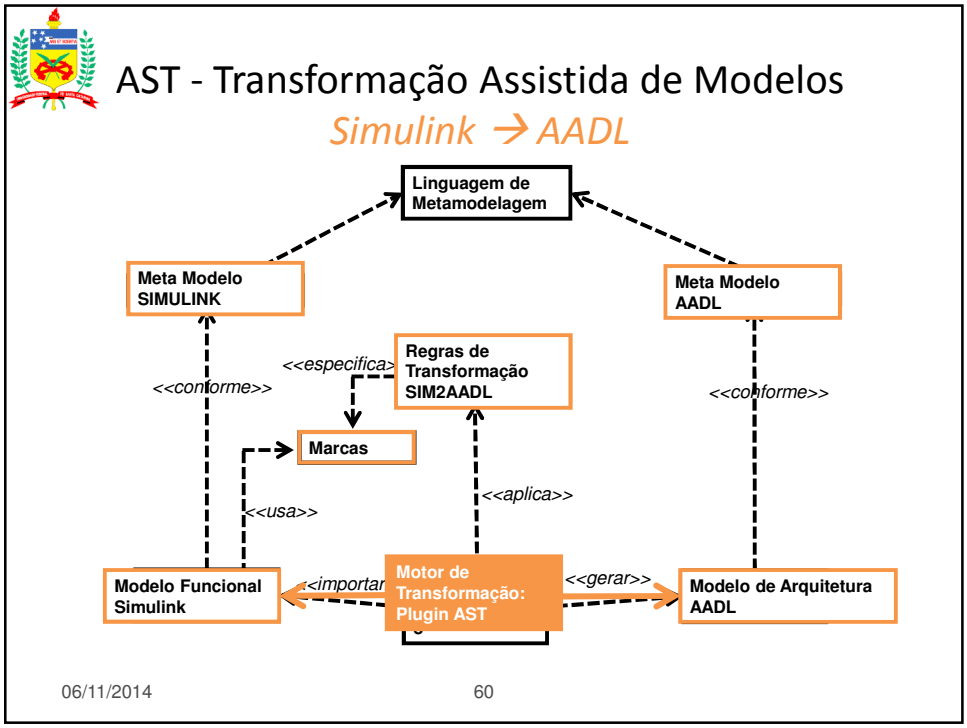


Proposta para transição automática entre projeto preliminar e projeto detalhado

Slide 58

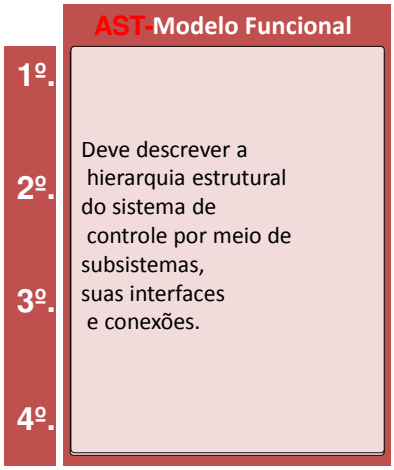
u2 este é o processo antigo, mudar para o que aparece na figura
user; 28/10/2014


AST - Transformação Assistida de Modelos
Funcional → Arquitetura
Linguagem de Metamodelagem
Meta Modelo
Meta Modelo
<<especifica>>
Regras de Transformação
<<conforme>>
<<conforme>>
Marcas
<<aplica>>
<<usa>>
Modelo Funcional
<<importar>>
Motor de Transformaçã o
<<gerar>>
Modelo Arquitetura
06/11/2014
59


AST - Transformação Assistida de Modelos
Simulink → AADL
Linguagem de Metamodelagem
Meta Modelo SIMULINK
Meta Modelo AADL
<<especifica>>
Regras de Transformação SIM2AADL
<<conforme>>
<<conforme>>
Marcas
<<aplica>>
<<usa>>
Modelo Funcional Simulink
<<importar
Motor de Transformação: Plugin AST
<<gerar>>
Modelo de Arquitetura AADL
06/11/2014
60

AST – Regras de Transformação

Correlação entre os Modelos Funcional e de Arquitetura



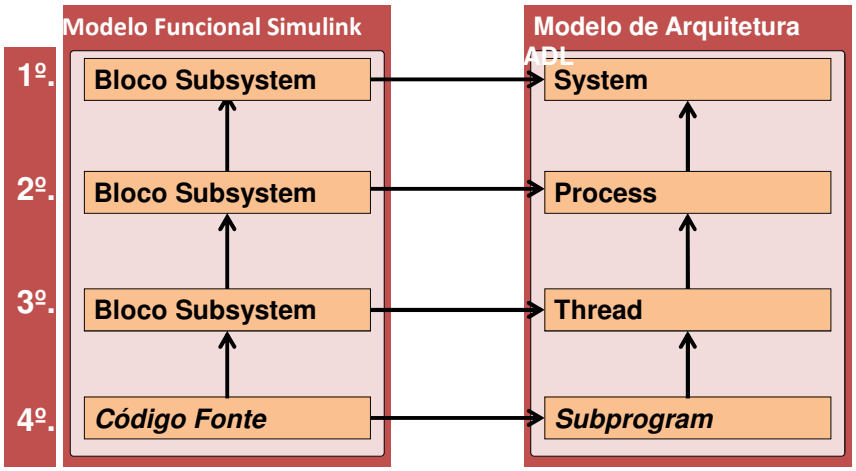
Ferramentas Modelagem Funcional → Diagrama de Blocos:

- Simulink
- SCADE
- LabView
- Scilab

06/11/2014 61

AST – Regras de Transformação

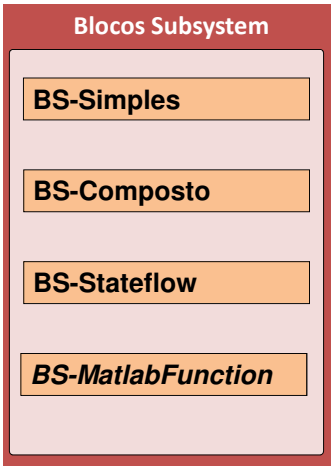
Correlação entre os Modelos Simulink e AADL



06/11/2014 62

AST – Regras de Transformação

Modelo Simulink → Tipos de Bloco Subsystem



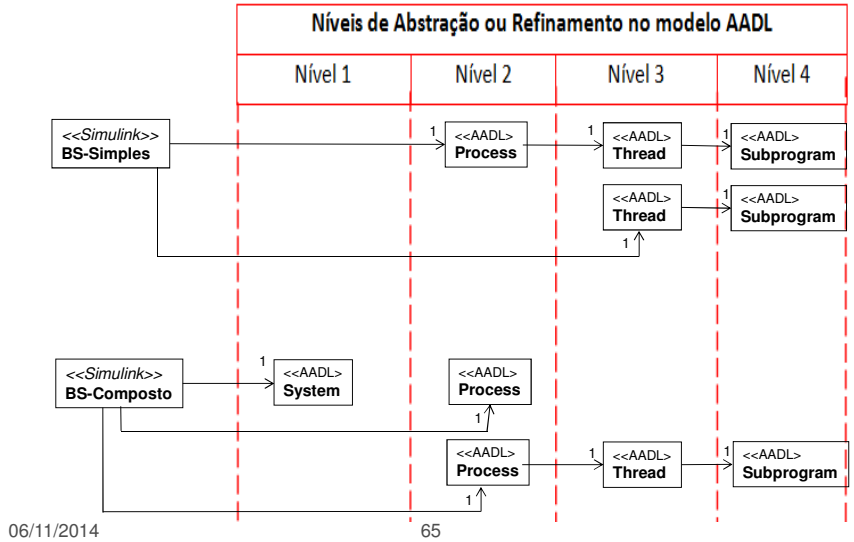
AST – Regras de Transformação

Tipos de Mapeamento

- **Mapeamento Estrutural**
  - Tipos de componentes: *system, process, thread, suprogram*.
  - Interfaces dos componentes (portas)
  - Conexões entre os componentes
- **Mapeamento Operacional**
  - Gera a seção **modes** (system ou process)
  - Gera um componente do tipo **thread** que possui a seção ***annex behavior_specification***

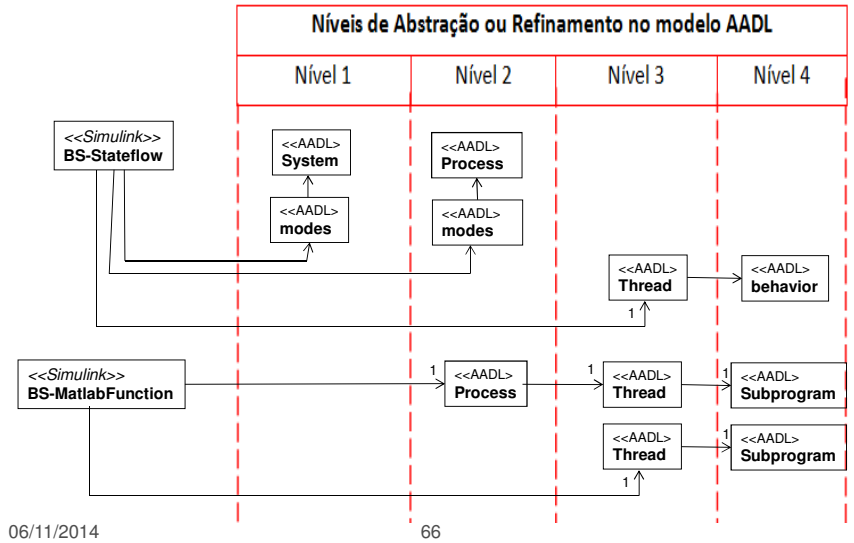
AST – Regras de Transformação

Variações de Mapeamento



AST – Regras de Transformação

Variações de Mapeamento



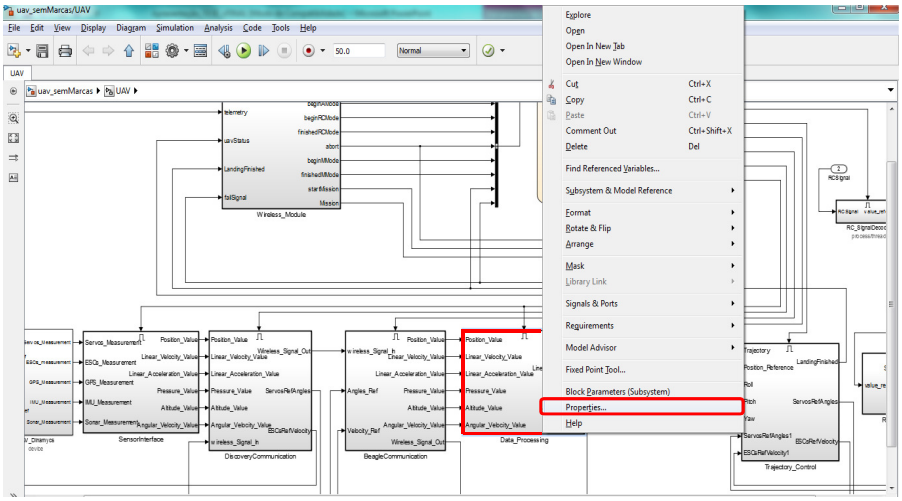
Conjunto de Marcas

Composite	<<system>>	Primeiro nível hierárquico
	<<process>>	Segundo nível hierárquico
	<<thread>>	Terceiro nível hierárquico
	<<device>>	Qualquer nível hierárquico
	<<process/thread>>	Qualquer nível hierárquico
Stateflow	<<modes>>	Segundo nível hierárquico
	<<thread/behavior>>	Terceiro nível hierárquico
Simple	<<process/thread>>	Primeiro ou Segundo nível hierárquico
	<<thread>>	Terceiro nível hierárquico
	<<device>>	Qualquer nível hierárquico
MatlabFunction	<<process/thread>>	Primeiro ou Segundo nível hierárquico
	<<thread>>	Terceiro nível hierárquico
	<<device>>	Qualquer nível hierárquico

06/11/2014

Marcação de um Modelo Funcional Simulink

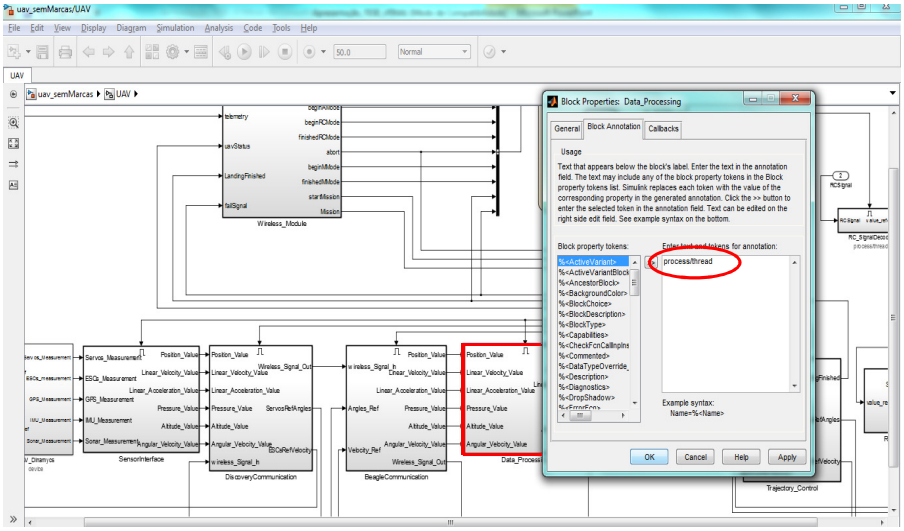
- manual -



06/11/2014 68

Marcação de um Modelo Funcional Simulink

- manual -

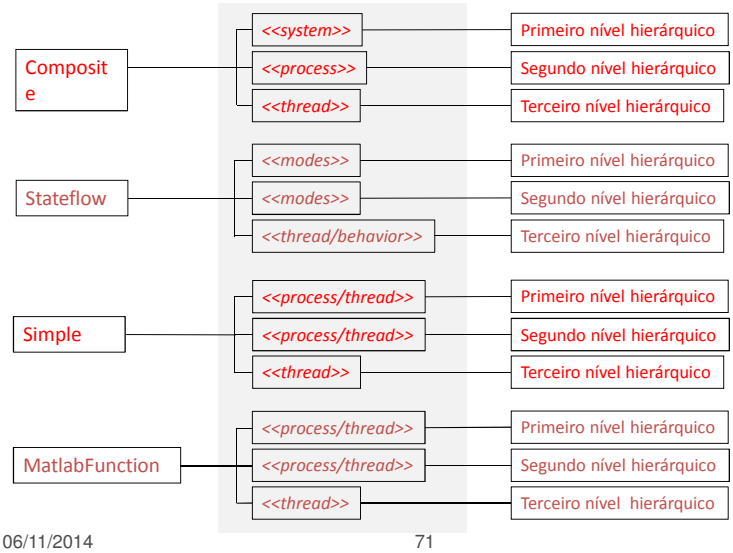


Marcação de um Modelo Funcional Simulink

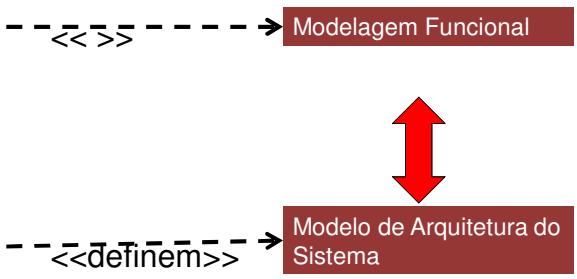
- automática-

- É de responsabilidade do motor de transformação de modelos → Plugin AS2T.
  - Não fornece variações de mapeamento.

Marcação de um Modelo Funcional Simulink
- automática-



Resumo



u4

Processo de Desenvolvimento de Software para CPS

Integração e Testes

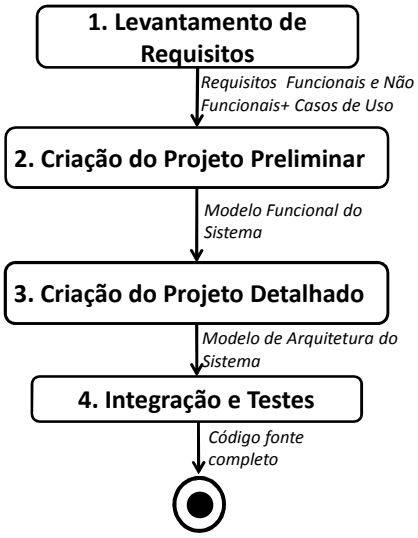
- Consiste em integrar o código gerado na etapa do projeto detalhado com os códigos produzidos na etapa de projeto preliminar
- Etapa ainda não detalhada no método proposto

Slide 73

u4 este é o processo antigo, mudar para o que aparece na figura
user; 28/10/2014

u3

Processo de Desenvolvimento de Software para CPS



Considerações Finais

- CPS de hoje são os RTS de ontem
- Discutiu-se uma proposta de método de desenvolvimento de software para CPS

Slide 75

u3 este é o processo antigo, mudar para o que aparece na figura
user; 28/10/2014