



Projeto de Sistemas de Tempo Real

Walter Fetter Lages

w.fetter@ieee.org

Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Engenharia Elétrica

ENG04008 Sistemas de Tempo Real



Introdução

- Especificação
- Definição da Arquitetura
 - *Hardware*
 - *Software*
- Detalhamento do projeto
- Implementação
- Teste
- Prototipação
- Interface com o usuário
- Escolha das ferramentas
- Documentação



Notações

- Desenvolvimento de um projeto é transformar notações de um estágio nas notações do estágio seguinte
- Nos níveis mais altos, estas transformações não são bem definidas
- Notação vaga e imprecisa, que não captura a semântica da transformação



Níveis de Notação

- Informal
 - Linguagem corrente
- Estruturada
 - Representação gráfica
 - Componentes bem definidos
 - Interconexões bem definidas
- Formal
 - Base matemática
 - Descrições precisas



Especificação

- Descrição informal
- Análise da especificação
 - Definição da funcionalidade
 - Comportamento temporal deve ser definido de forma explícita
 - Definição de testes
- Modelo do ambiente
 - Taxas de interrupções
 - Quantidade de objetos externos a serem tratados
 - Modos de falha



Análise de Especificações

- Análise estruturada
- Análise orientada a objetos
- Cálculo de predicados
 - Extensões para incorporar tempo
 - Problemas de modularidade
- Redes de Petri
- Problemas:
 - Especificações não explícitas
 - Sintaxe \times semântica



Detalhamento do Projeto

- *Top-down*
 - Decomposição
 - Abstração
- *Botton-up*
 - Facilita a reutilização de módulos
 - Adapta-se melhor a pequenos projetos



Encapsulamento

- Abstração exige subcomponentes
 - Funções bem definidas
 - Interconexões não ambíguas
 - Interfaces claras
 - *Information hiding*
 - Decomposição composicional
 - Coesão
 - Coincidental, Lógica , Temporal, Procedural, Comunicacional, Funcional
 - Acoplamento
 - Forte, fraco



Implementação

- *Gap* entre metodologia de projeto e linguagem de programação
- Linguagem de baixo nível
 - Assembly
- Linguagens alto nível sequenciais
 - Fortran, Pascal, Modula-2, C, C++
- Linguagens de alto nível concorrentes
 - Ada, Occam, Java
- Linguagens que suportam a noção de tempo
 - VHDL, Verilog

Características de Linguagens



- Segurança
 - Detecção de erros
 - Tempo de compilação
 - Tempo de execução
- Legibilidade
- Flexibilidade
- Simplicidade
- Portabilidade
- Eficiência
 - Propriedade da Linguagem ou do Compilador?



Teste

- Dificuldade de teste devido as características temporais
- Interações entre processos são difíceis de reproduzir
- Respostas a condições imprevistas do ambiente
- Simulação
 - Teste de arquiteturas
 - Teste com módulos simulados



Prototipação

- Verificação das especificações
 - Correção
 - Complitude
- Custo deve ser bem menor do que do sistema real
 - Uso de linguagens de mais alto nível
 - Matlab



Interface com o Usuário

- Informações para o usuário
 - Quantidade
 - Qualidade
- Deve ser isolada do resto do sistema
- Validação dos comandos/dados fornecidos pelo usuário
- Fadiga do operador
- Costume do operador
- Sistemas de suporte à decisão