

Especificação, Modelação e Projecto de Sistemas Embutidos

Linguagens de especificação: SDL

Paulo Pedreiras
pbrp@ua.pt



Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

SDL

Linguagem desenhada para permitir a **especificação de sistemas distribuídos**

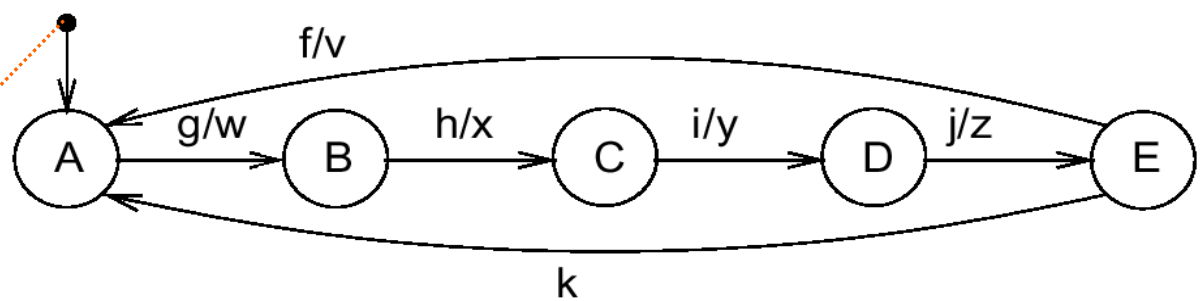
- Remonta ao início da década de 70
- Semântica formalmente definida no final da década de 80
- Standardizado pelo ITU (*International Telecommunication Union*), recomendação Z.100, em 1980
 - Updates em 1984, 1988, 1992, 1996 e 1999
- Baseado na **passagem assíncrona** de mensagens
 - **Adequado a sistemas distribuídos**

SDL

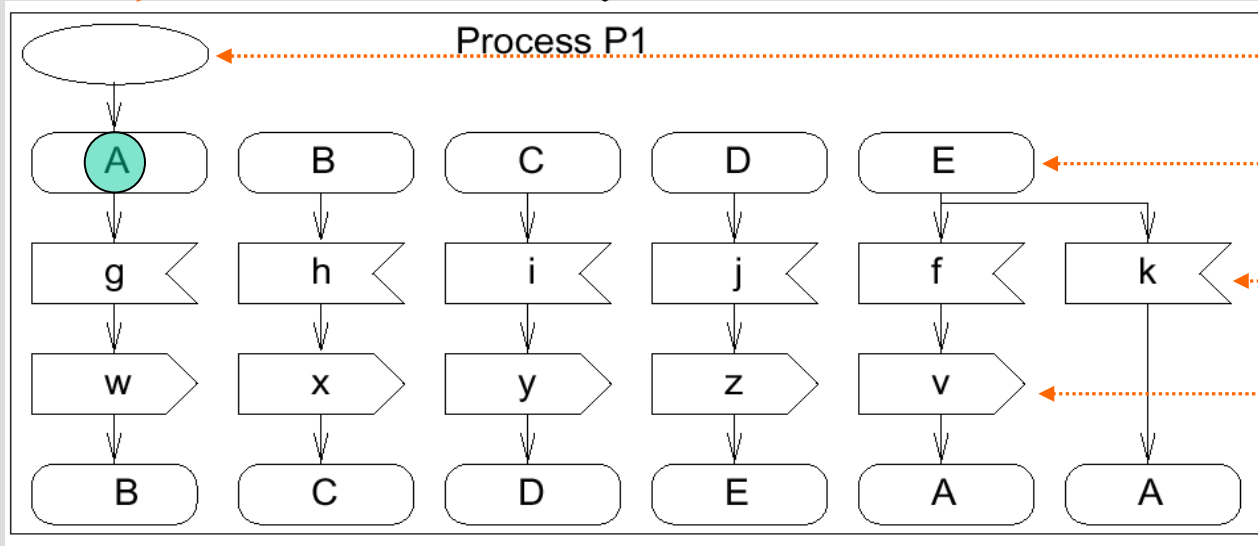
- Disponibiliza formatos **gráficos e texto**
 - (propósito de) agrada(r) a todos os tipos de utilizadores
- Tal como os StateCharts, o seu **modelo de computação** é baseado em máquinas de estados finitos concorrentes (**CFSM**). Cada FSM é designada por **processo**
- Todavia usa **passagem de mensagens** em vez de memória partilhada para comunicação
- Adicionalmente o SDL suporta **explicitamente operações** sobre **dados**
 - O tipo de dados pode ser pré-definido ou definido na própria descrição SDL

SDL: representação de FSMs/processos

- Representação em **diagrama de estados:**



- Representação em **SDL:**



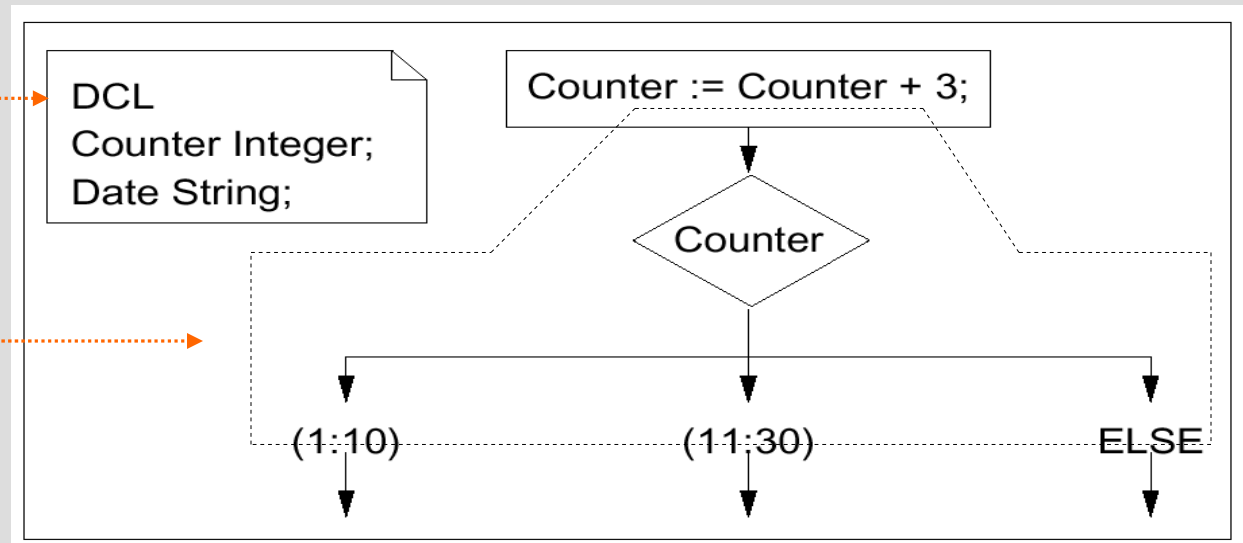
estado
inicial
estado
input
output

Operações sobre dados

- As **variáveis** podem ser **declaradas** localmente para cada processo
- Os respectivos **tipos** podem ser **pré-definidos** ou **definidos** no próprio SDL
- O SDL suporta **tipos de dados abstractos** (*abstract data types/ADTs*).
- É também possível **especificar decisões**

Declaração de variáveis

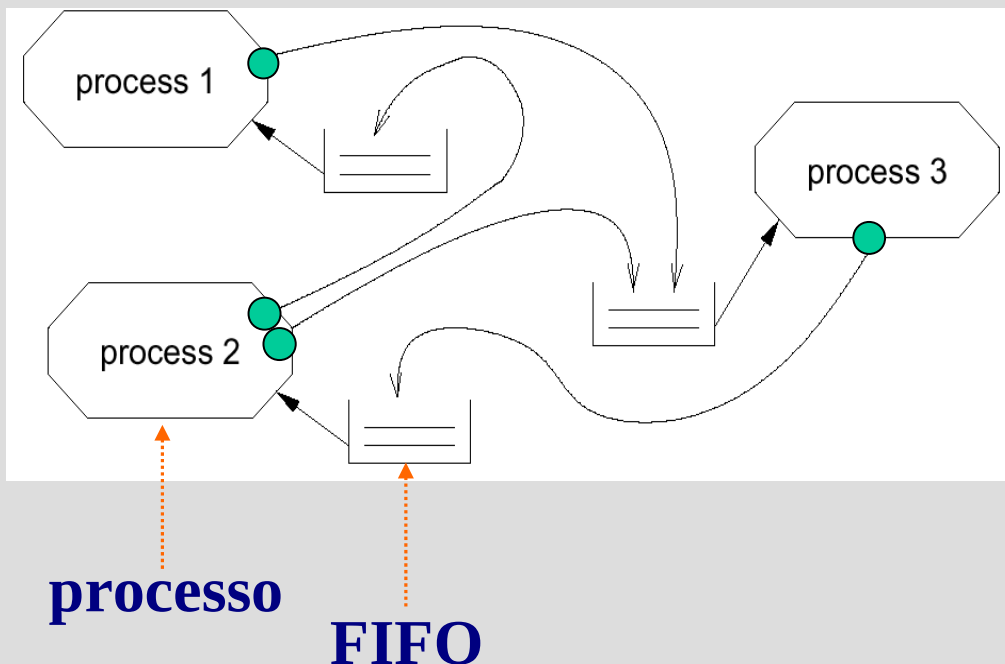
Decisão



Comunicação entre SDL-FSMs/processos

A **comunicação** entre FSMs (ou “processos”) é baseada em **passagem de mensagens**

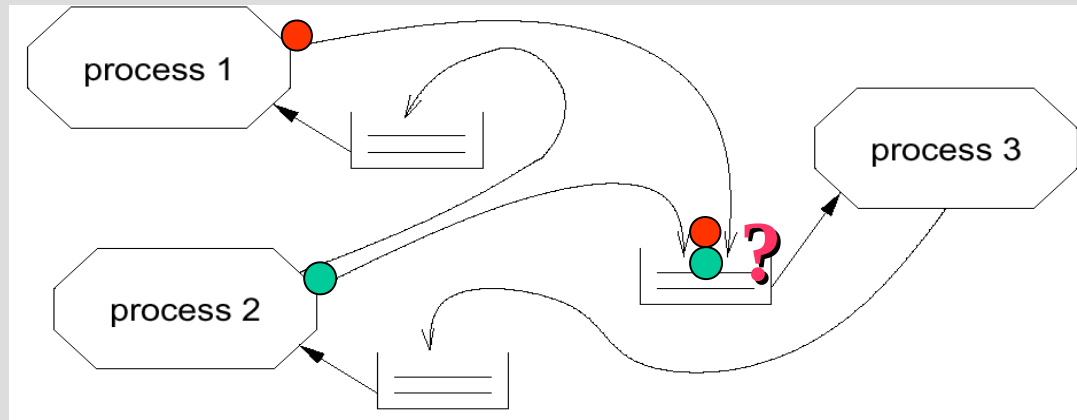
- Mensagens armazenadas numa **fila** tipo **FIFO**
- É pressuposto que a **dimensão** do FIFO seja **infinita**.



- Cada processo obtém a **próxima entrada** do FIFO
- Verifica se esse input causa alguma **transição**
- Em caso **afirmativo** a **transição** é **efectuada**
- Caso contrário o input é **ignorado**
 - excepção: mecanismo SAVE.

É determinístico?

A **ordem** de chegada das mensagens é (no caso geral) **indefinida**. Assim, a ordem pela qual são armazenadas é desconhecida!

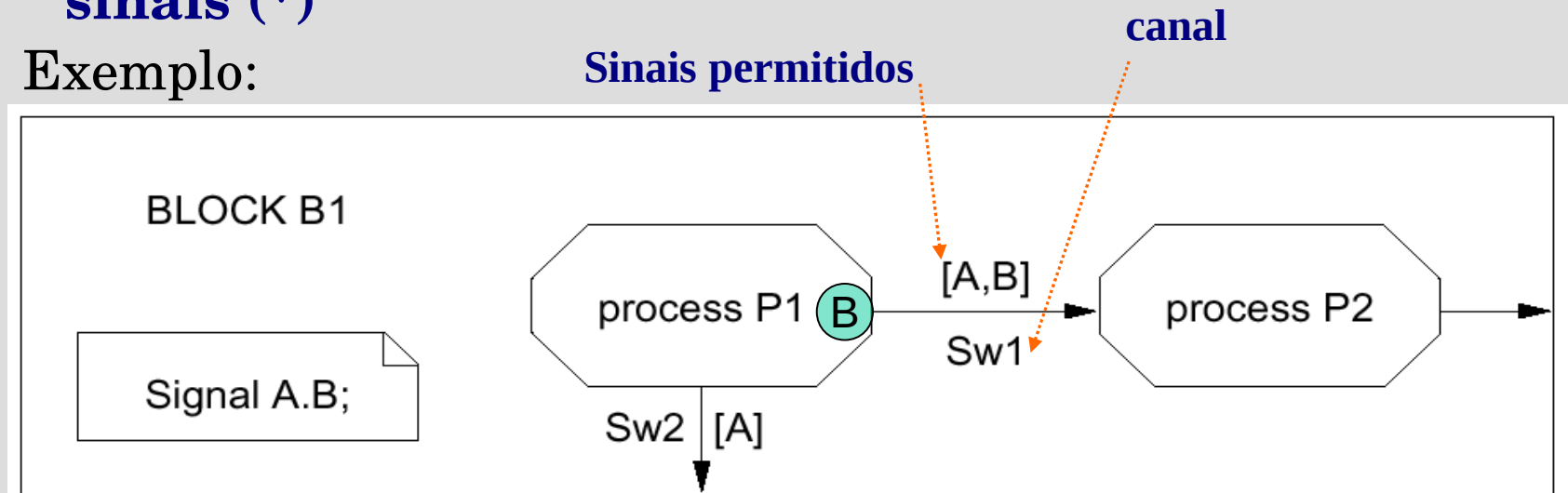


Exemplo: os processos P1 e P2 enviam duas mensagens para P3 “**em simultâneo**”

- **Ambas** as ordens de armazenamento são “**legais**”
- Os simuladores podem mostrar **diferentes comportamentos** para os **mesmos inputs**, sendo todos **correctos** (i.e., a variável tempo não faz parte do modelo)

Especificação da interacção entre processos

- A **interacção** entre processos pode ser descrita por meio dos denominados “**Diagramas de Interação entre Processos**” (*Process Interaction Diagrams*), os quais são um caso particular dos diagramas de blocos.
- Para além dos **processos**, estes diagramas especificam **canais de comunicação** usados para envio e recepção de **sinais (*)**
- Exemplo:



(*) Em SDL “sinal” designa entradas e saídas dos automata modelados

Designação dos destinatários dos sinais

1. Por identificação directa do processo:

Exemplo: OFFSPRING -
representa todos os processos
gerados dinamicamente.



2. Explicitamente:

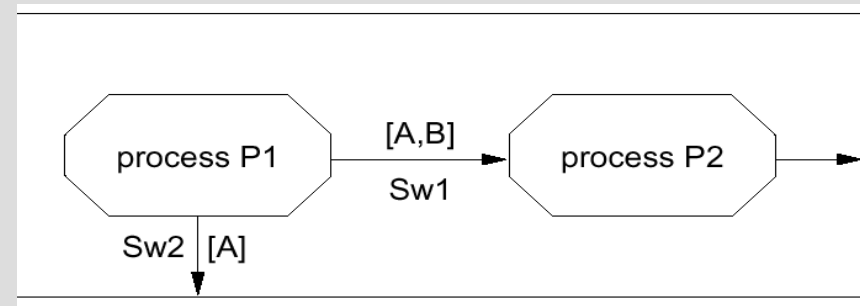
Incluindo a designação do nome
do canal



3. Implicitamente:

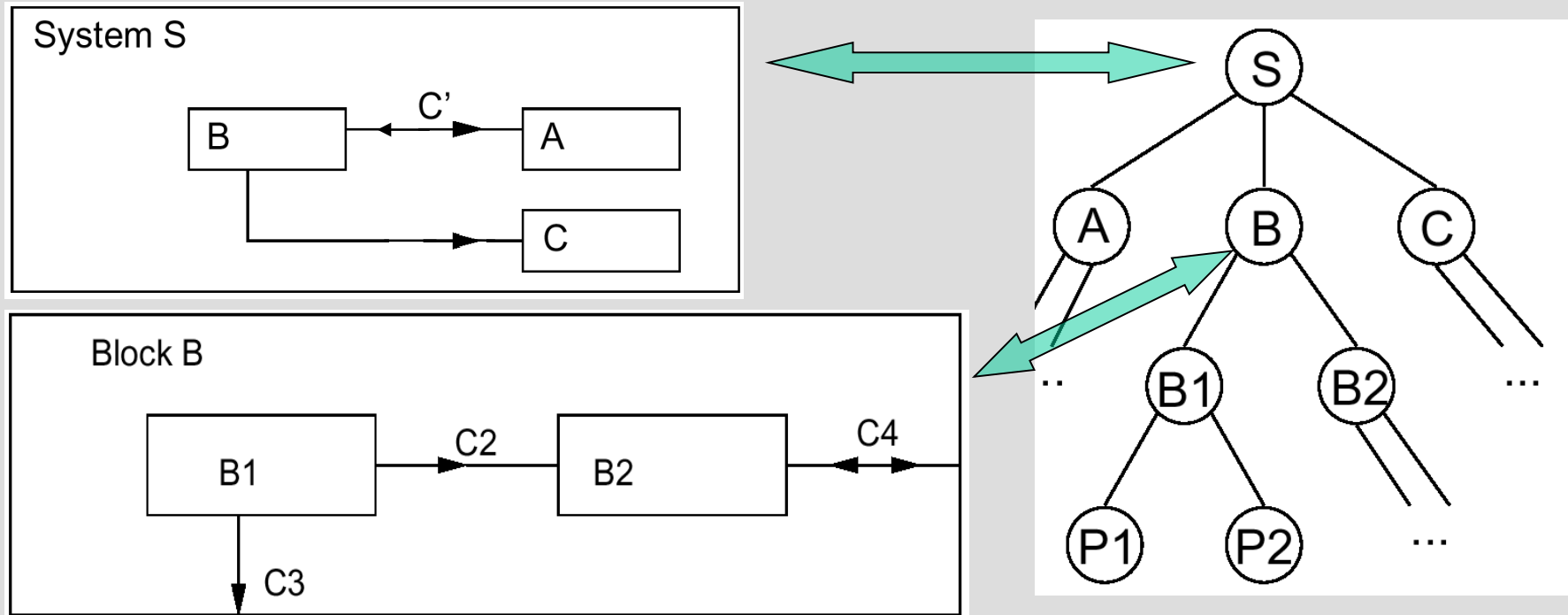
Se o nome do sinal identifica
univocamente o canal

- e.g. B \rightarrow Sw1



Hierarquia em SDL

Os diagramas de interacção de processos podem ser incluídos em **blocos**. O **bloco raiz** (*root block*) é denominado **sistema**.

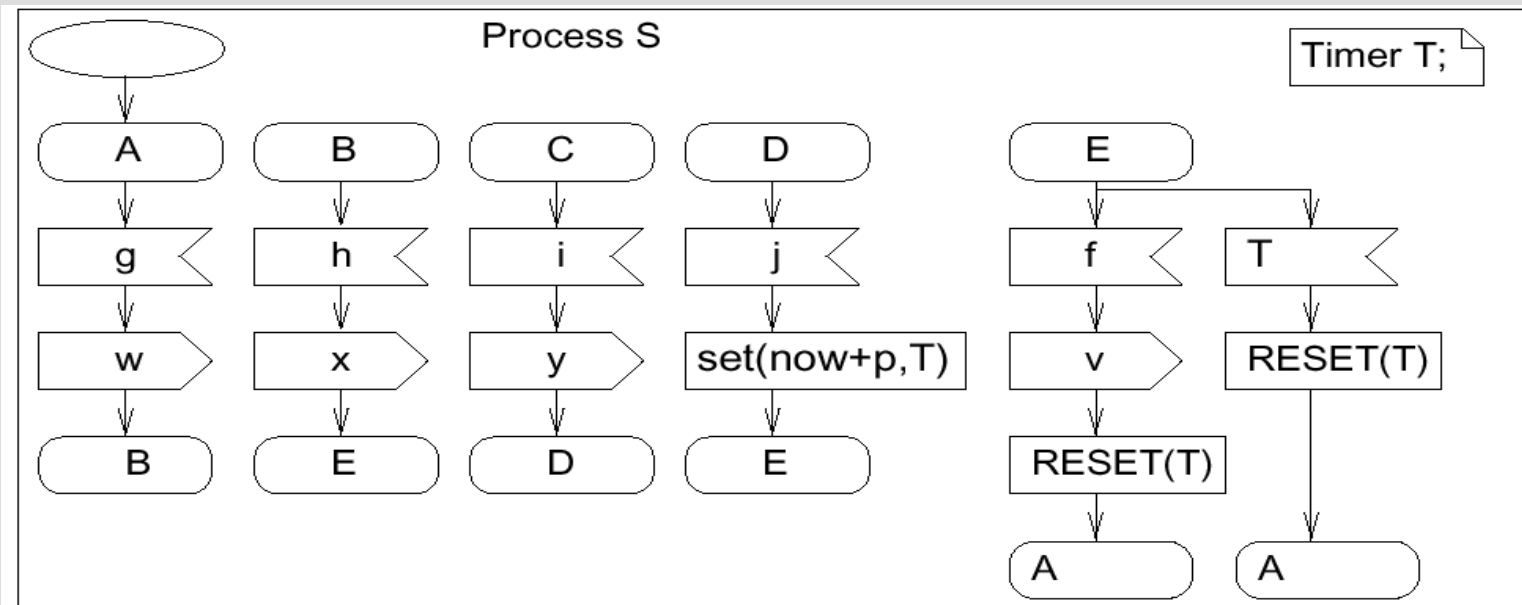


!!!! Os processos **não podem conter outros processos**, contrariamente ao que acontece em StateCharts

Suporte a Timers

Em SDL os timers podem ser declarados localmente.

- Quando o **tempo** especificado **expira** é gerado um **sinal** que é colocado na respectiva fila FIFO
- Um **RESET** permite **remover** um timer (também da fila FIFO)



Saída do estado “E” dá-se por **input** do evento “f” **ou** pela passagem de um **tempo** “p”

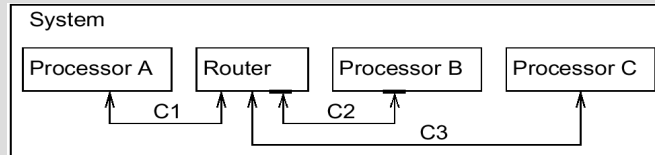
SDL: outros elementos da linguagem

O SDL inclui um grande número de outros **elementos**, como:

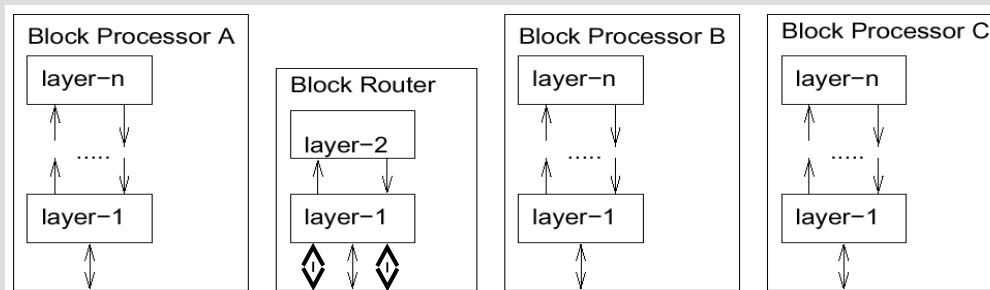
- **Procedimentos** (*procedures*)
- **Criação** e **terminação** dinâmica de **processos**
- **Descrição** de **dados** avançada
 - e.g. definição de estruturas, ...

Áreas de aplicação

E.g. Protocolos de comunicação



!!! Especificação do protocolo ISDN foi efectuada em SDL



- Implementação da stack em camadas (*layered*)
- O comportamento de cada camada é típica/ modelado por uma FSM
 - Pode atingir níveis de **complexidade muito elevados**
 - **Tratamento** de condições de **erro**, ordenar/encaminhar **pacotes de dados, ...**

O SDL é também usado em muitas outras áreas, nomeadamente em **sistemas embutidos distribuídos**

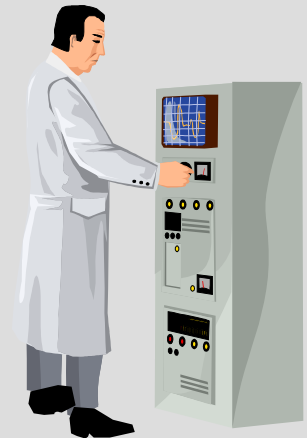
- E.g. a ESA (Agência Espacial Europeia) usa SDL em diversos projectos

Exemplo: máquina de vending

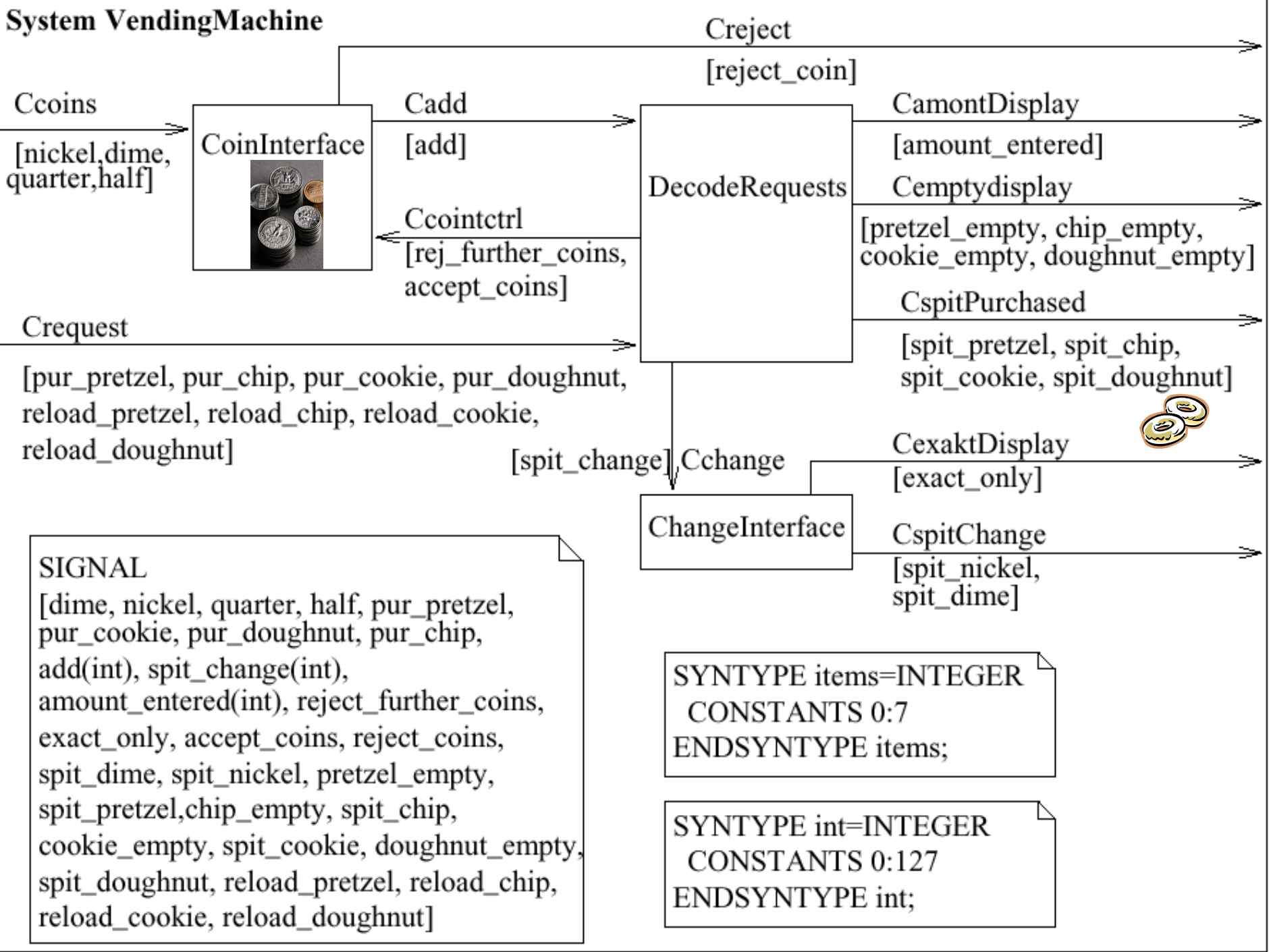
Máquina que vende *pretzels*, chips (batatas fritas), bolos e donuts:

Aceita as moedas de: *nickel*, *dime*, *quarter* e meio *dollar*.

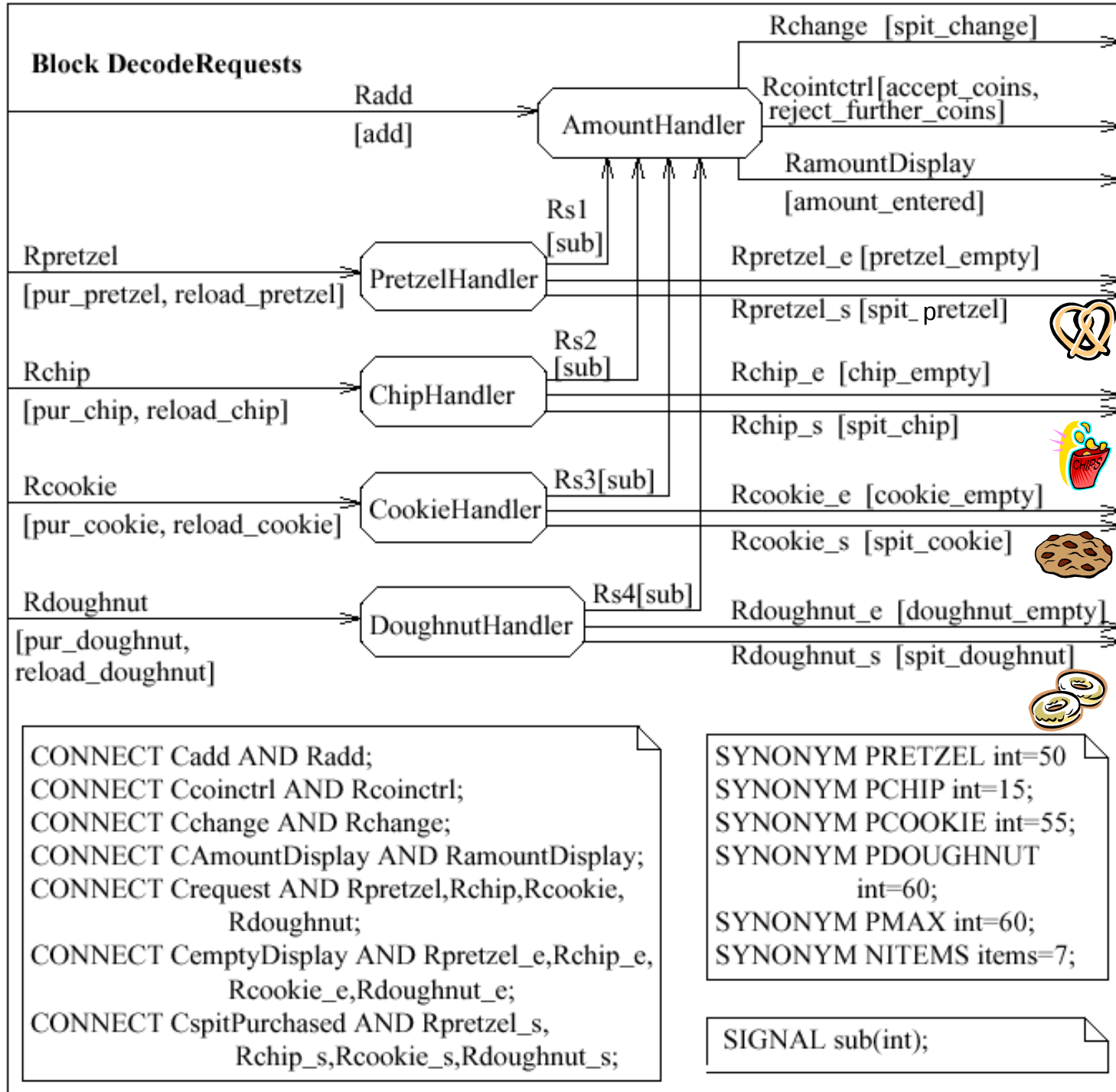
Aplicação centralizada.



° [J.M. Bergé, O. Levia, J. Roullard: High-Level System Modeling, Kluwer Academic Publishers, 1995]



Decode Requests

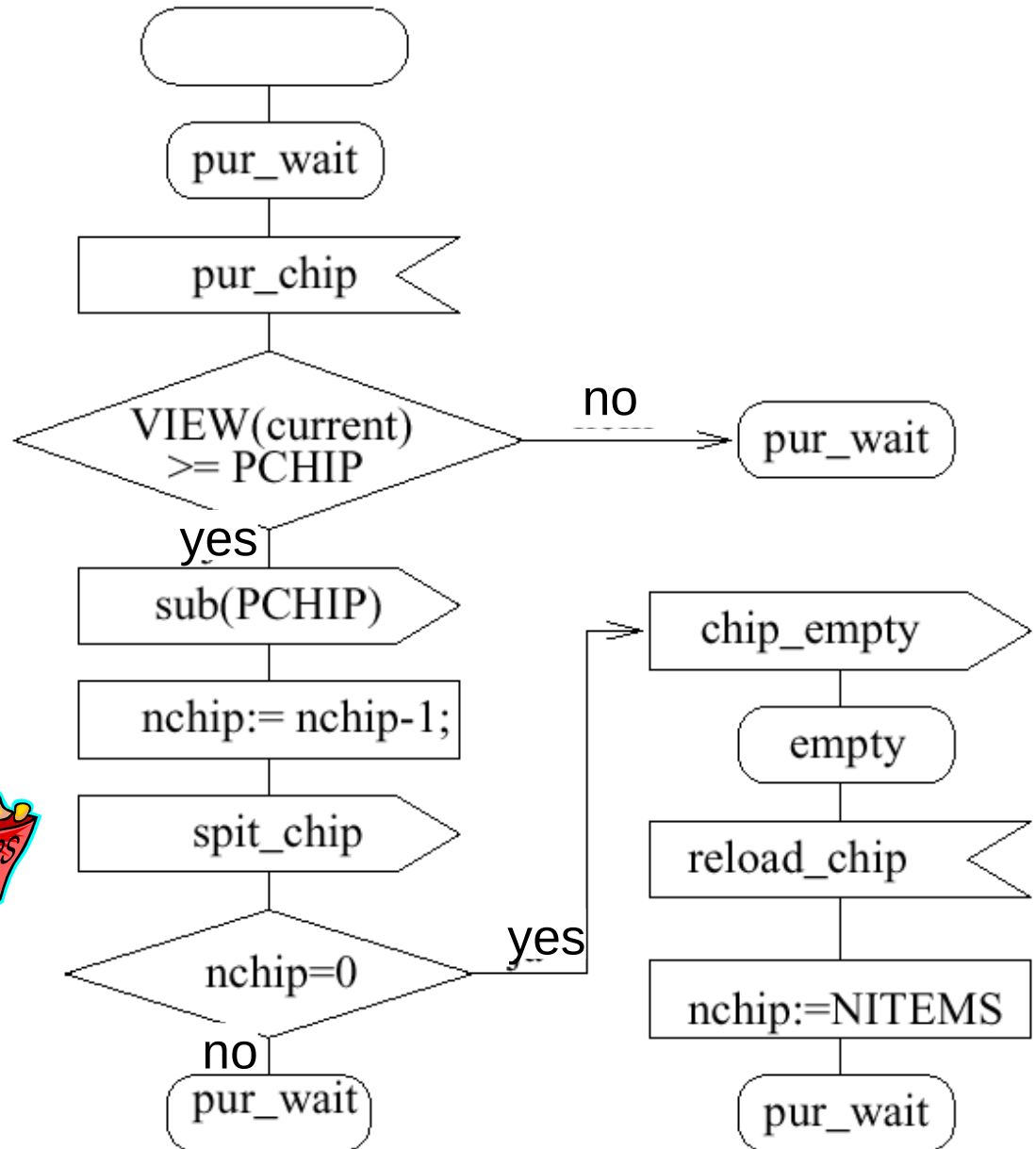


Processo „ChipHandler”

Process ChipHandler

DCL nchip items:=NITEMS;

VIEWED current int;



Versões e ferramentas

- SDL-88
- SDL-92: adicionada orientação a objectos
- SDL-96
- SDL-2000:
 - Suporte gráfico extendido;
 - Processos e blocos substituídos por “agentes”; aceitação limitada.
- Ferramentas para ligação a UML
- Mais informação: www.sdl-forum.org

Avaliação

- Excelente para **aplicações distribuídas** (foi usado para especificar o ISDN)
- **Ferramentas** comerciais **disponíveis** de várias fontes - SINTEF, Telelogic, Cinderella(www.cinderella.dk).
- **Não** necessariamente **determinístico**
 - ordem de chegada dos inputs é desconhecida – não é linguagem síncrona
- Implementação requer **dimensionamento** dos FIFOs
 - Pode ser **difícil de calcular**
- Gestão de **tempo** limitada (**timeouts, apenas**)
- Uso de **hierarquia limitado**
- **Não** suporta descrição de **propriedades não funcionais**.

Sumário

SDL

- Modelo de computação: máquina de estados finitos + comunicação baseada em passagem de mensagens não bloqueante
- Representação de processos
- Comunicação e diagramas de blocos
- Timers
- Exemplo: máquina de *vending*
- Versões e avaliação