LMF – Grupo ForMULa

Desenvolvimento Rigoroso com UML-RT

Mestrando: *Rodrigo Teixeira Ramos*
Orientador: *Augusto Sampaio*
Co-Orientador: *Alexandre Mota*

Agenda

- Introdução
  - Motivação, Problema, Contribuição e Resultados
- UML-RT
- Leis de transformação de modelos UML-RT
- Desenvolvimento de um Estudo de Caso
- Completude relativa das leis
- Atribuição de uma semântica à UML-RT
- Considerações Finais
  - Conclusões, trabalhos relacionados e futuros

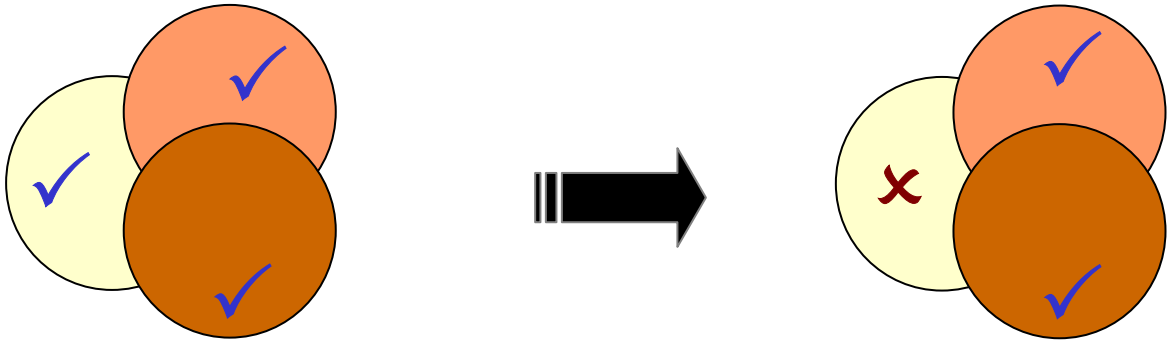
Motivação

- Interesse crescente por Metodologias de desenvolvimento baseadas em modelos
  - Modelos e Transformações

- Linguagem padrão (OMG) de modelagem:
  - UML
  - Extensão de UML: UML-RT

Problema

- A preservação de comportamento durante a evolução do modelo
- Considerar as diversas visões do modelo

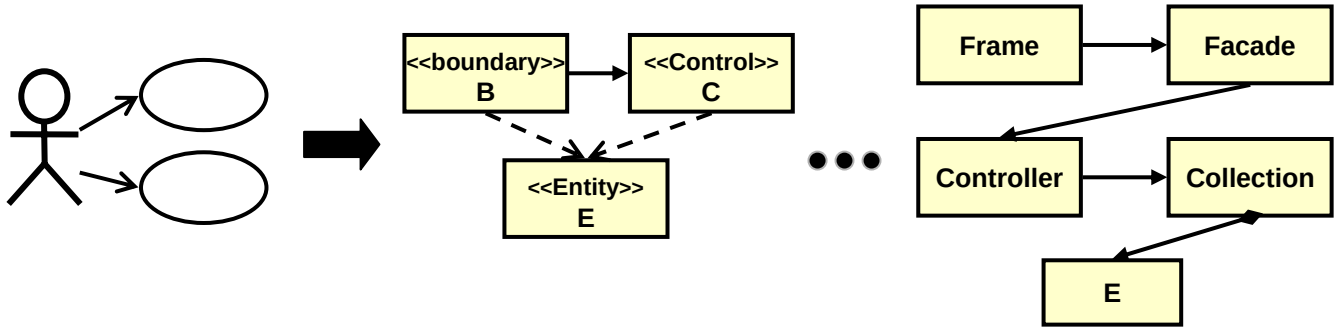


Contribuição

- Escopo
  - Leis básicas de transformação de modelos UML-RT
    - Visões Estruturais e Comportamentais
  - Atribuição de uma semântica à UML-RT
    - Via mapeamento para uma notação formal (*OhCircus)*
    - Possibilita a prova de leis
  - Estudo da abrangência das leis propostas
  - Validação através de um estudo de caso

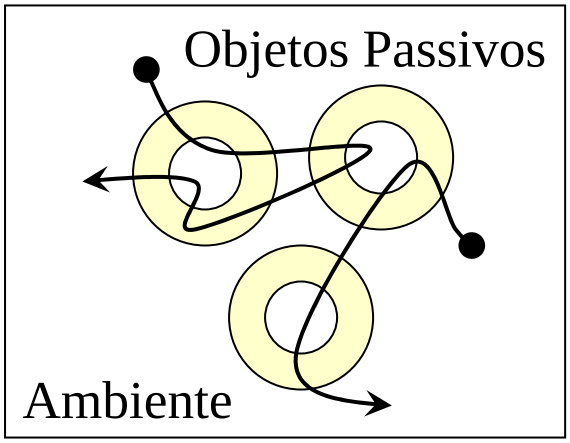
Resultados práticos

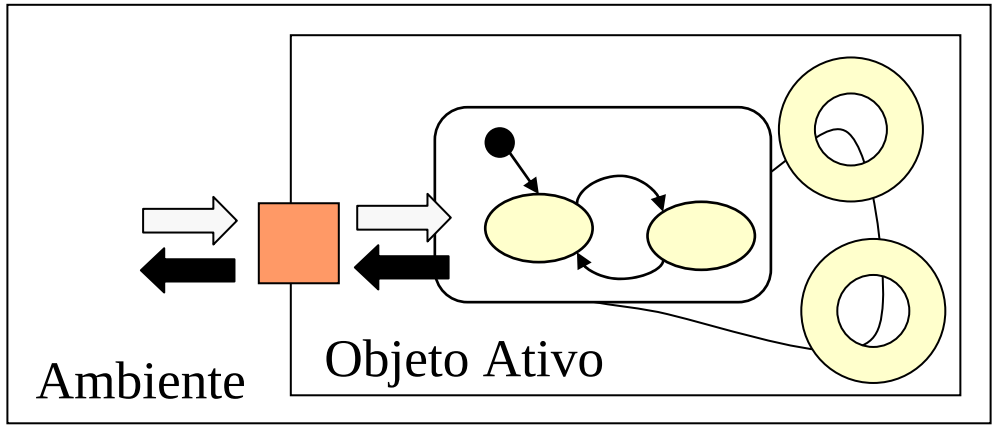
- Resultados práticos
  - Desenvolvimento sistematizado apoiado em métodos formais
  - Utilização de transformações de modelo:
    - Concretização do modelo
    - Aumento de qualidade do modelo



UML-RT
Um linguagem com Objetos Ativos

- Modelagem de sistemas concorrentes e distribuídos
- Profile de UML que adiciona novos elementos:
  - Cápsulas (Objetos ativos), Protocolos e Portas

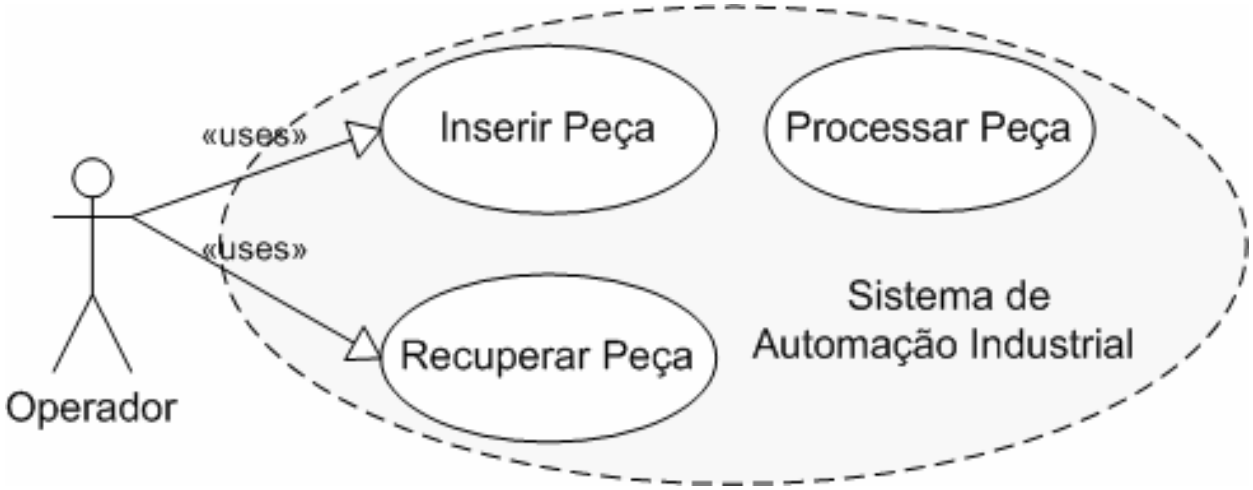




- Necessita de um conjunto reduzido de diagramas:
  - Diagramas de classe, estado e estrutura

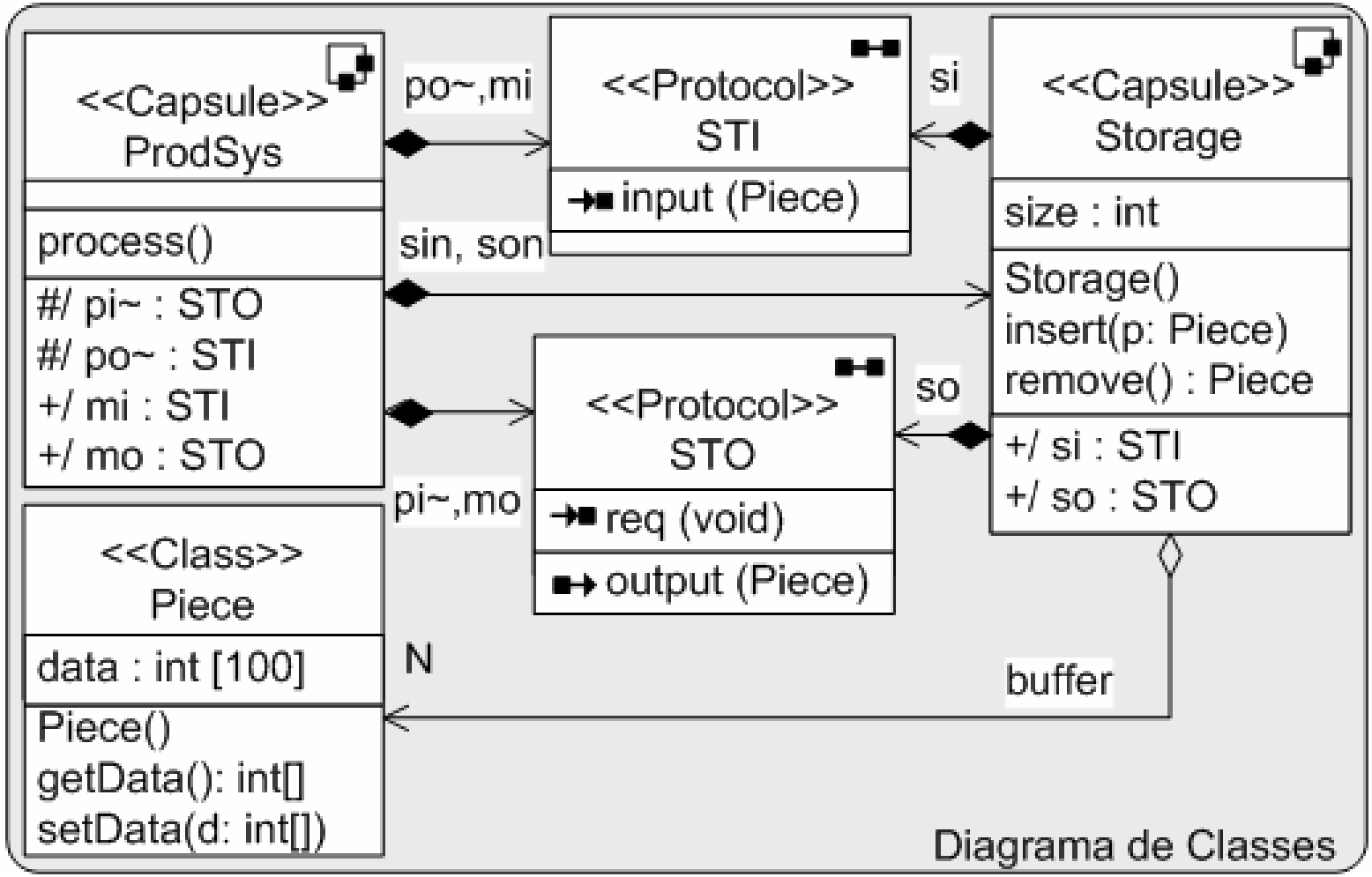
Exemplo - Diagrama de Caso de Uso

- Sistema de Automação industrial
- Diagrama de Caso de Uso:

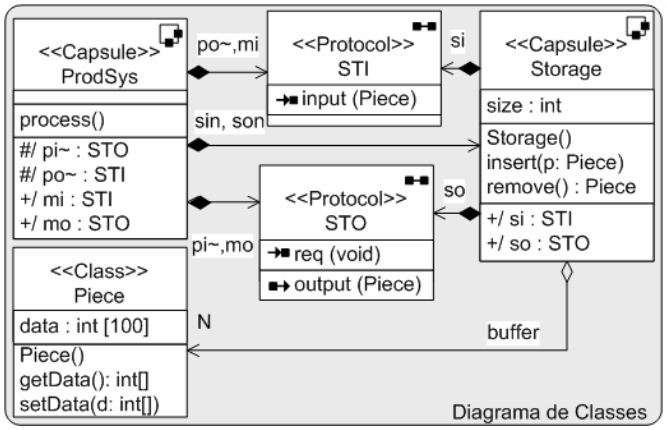


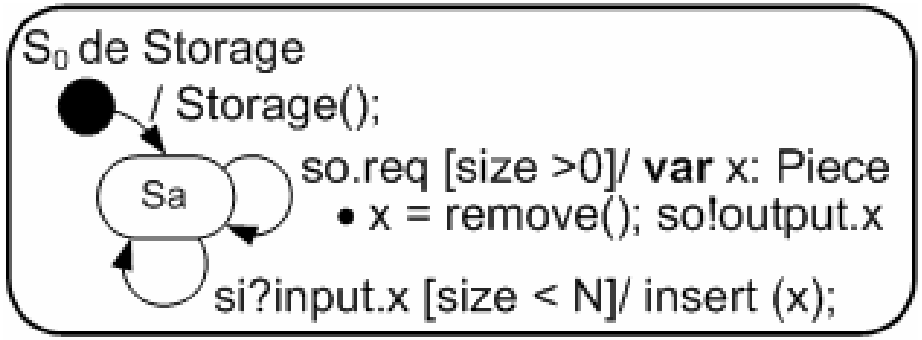
UML-RT

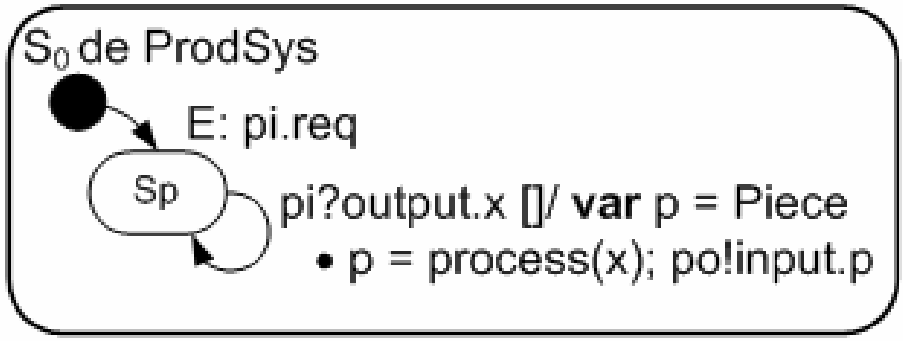
Exemplo – Diagrama de Classe

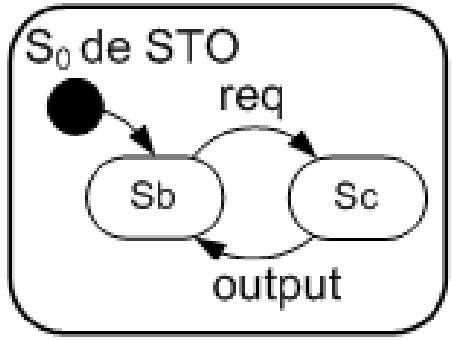


UML-RT
Exemplo – Diagrama de Estado



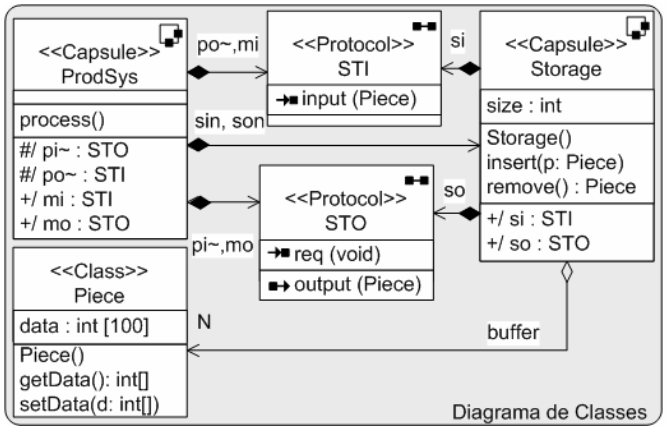


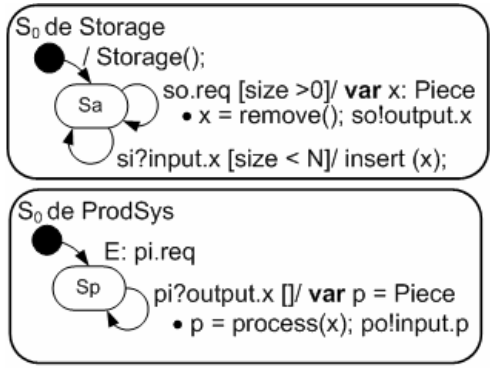


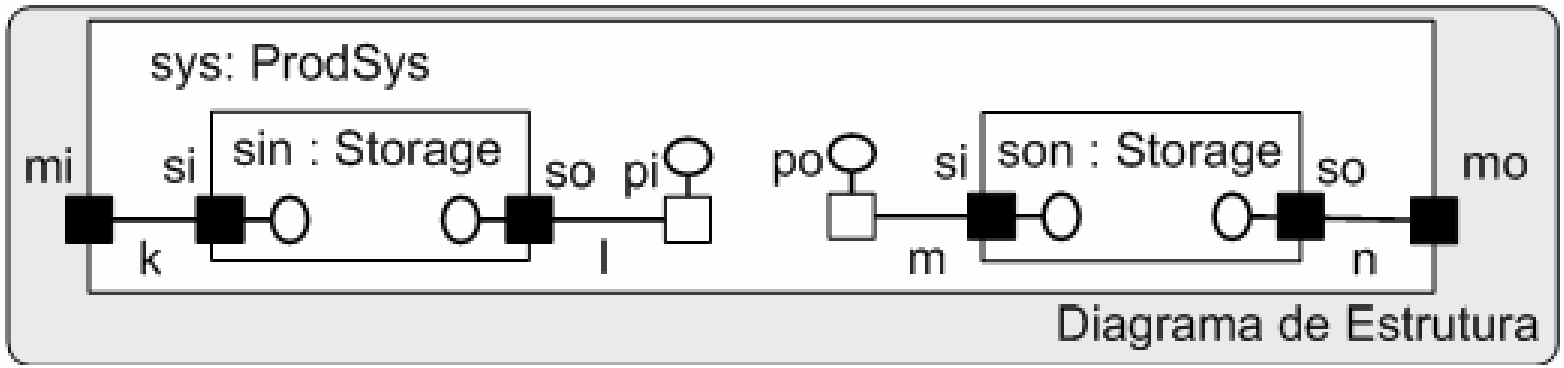


UML-RT

Exemplo – Diagrama de Estrutura

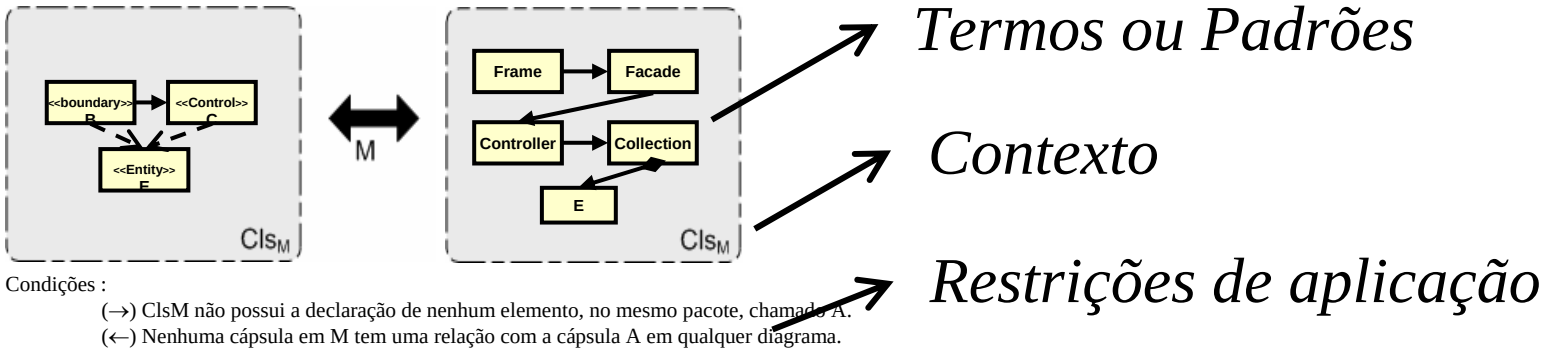






Leis de Transformação

- Cada Lei define duas regras de transformação, que formam uma equações condicional

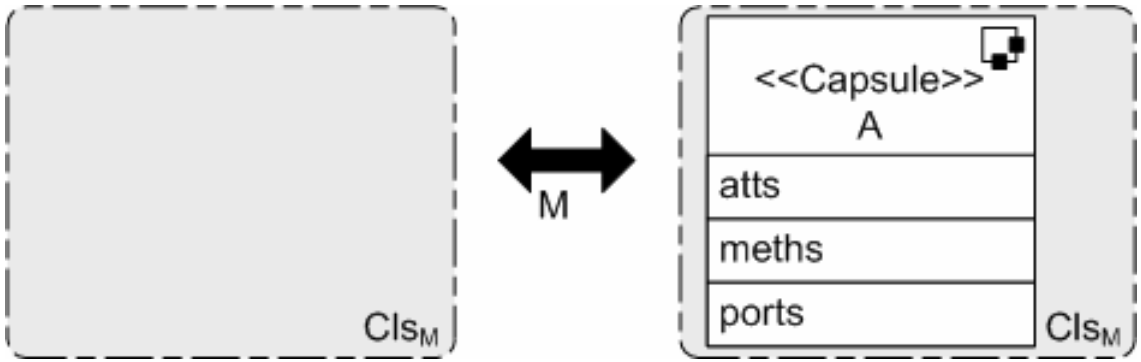


- Consideram conjuntamente os diagramas:
  - Classe, Estrutura e Estado

Leis de Transformação

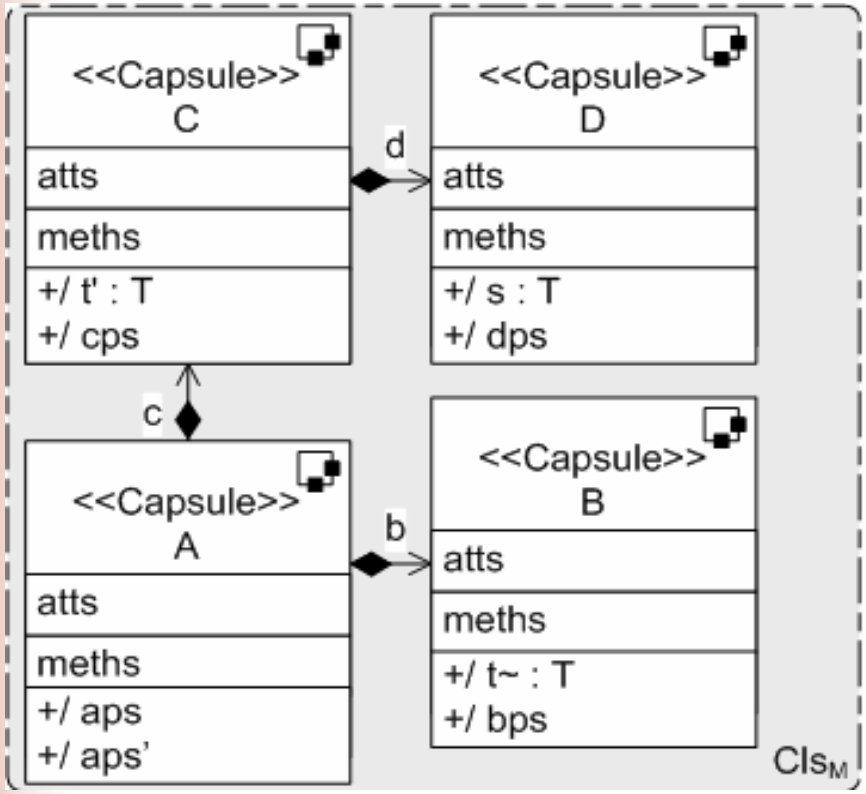
- Leis para :
  - Declaração de Cápsulas, classes, protocolos, associações
  - Conexão e Substituição de cápsulas
  - Alterar estrutura hieráquica dos diagramas
  - Mover o comportamento de protocolos
  - Transformar classes em cápsulas
  - Decompor Cápsulas

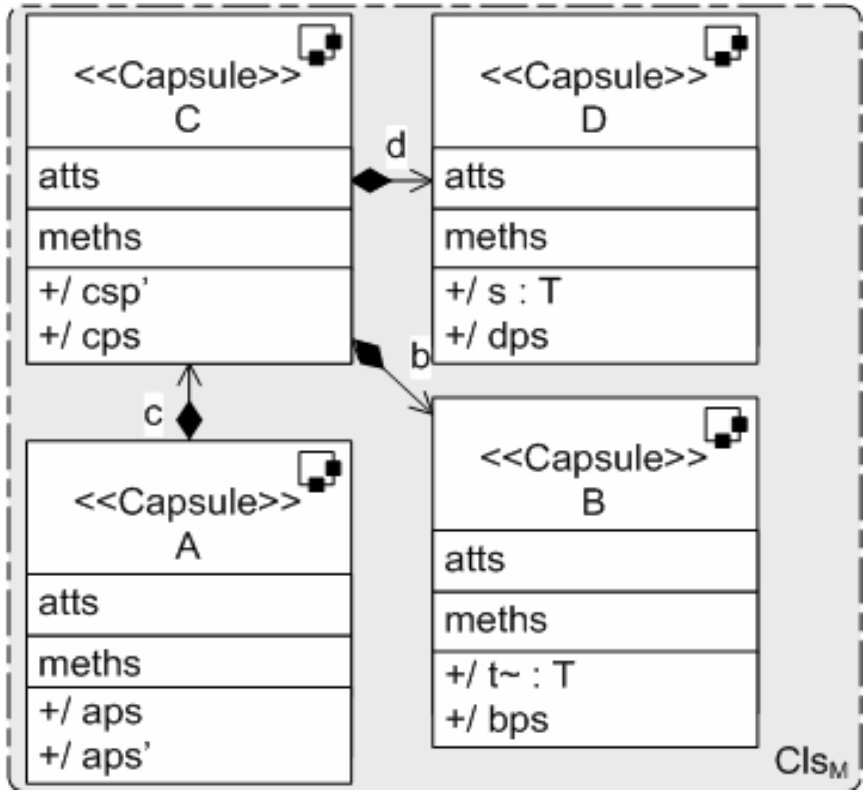
Lei 1- Declarar Cápsula



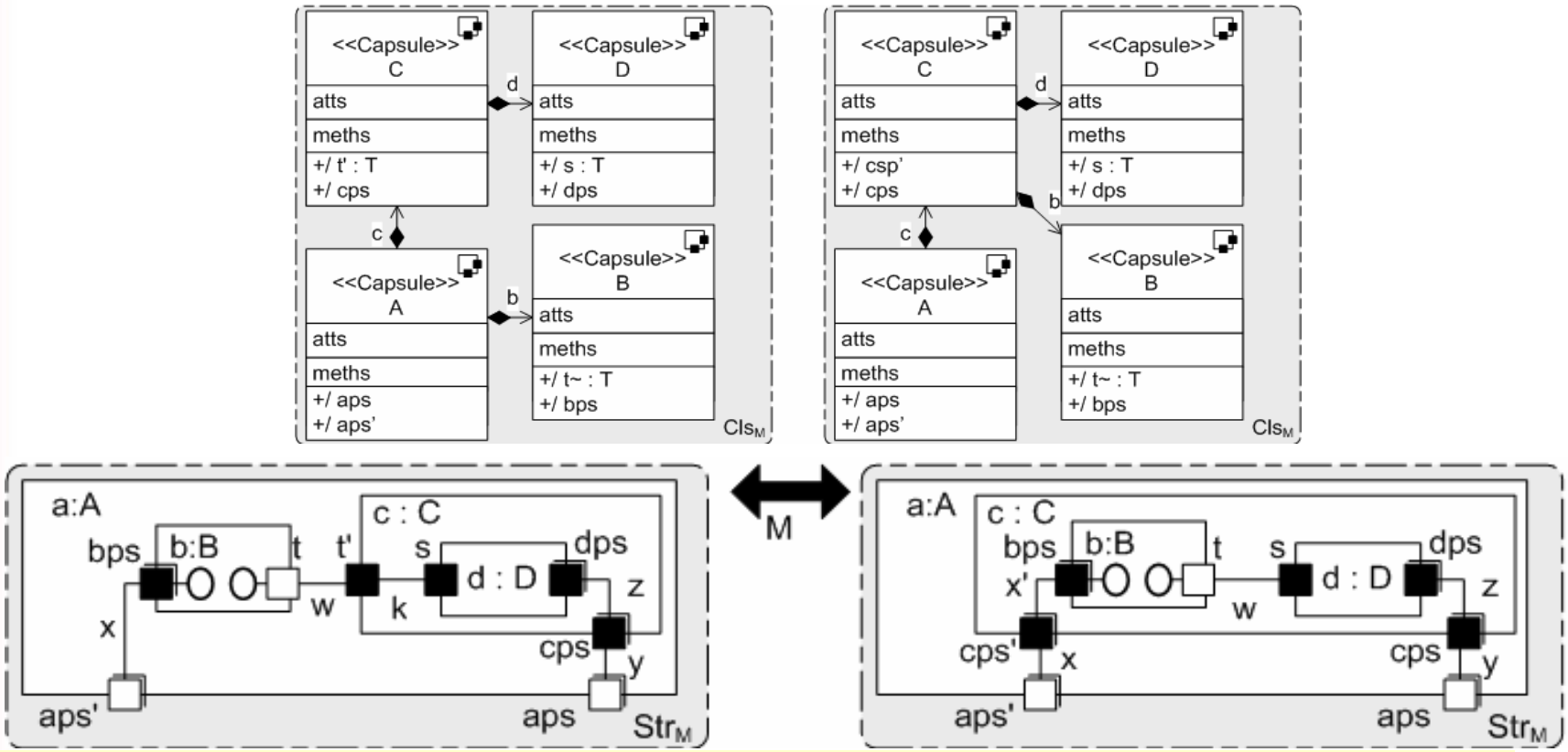
- Condições :
  - (→) Cls_M não possui a declaração de nenhum elemento, no mesmo pacote, chamado *A*.
  - (←) Nenhuma cápsula em *M* tem uma relação com a cápsula *A* em qualquer diagrama.

Lei 11 - Encapsular Cápsula





Lei 11 - Encapsular Cápsula

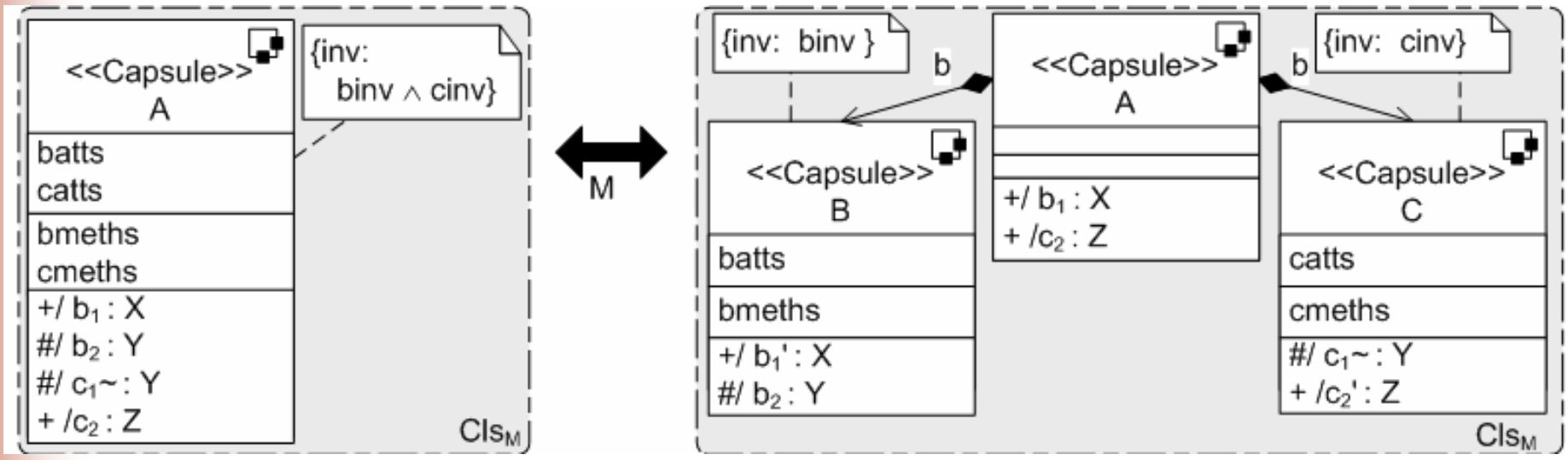


- Condições:

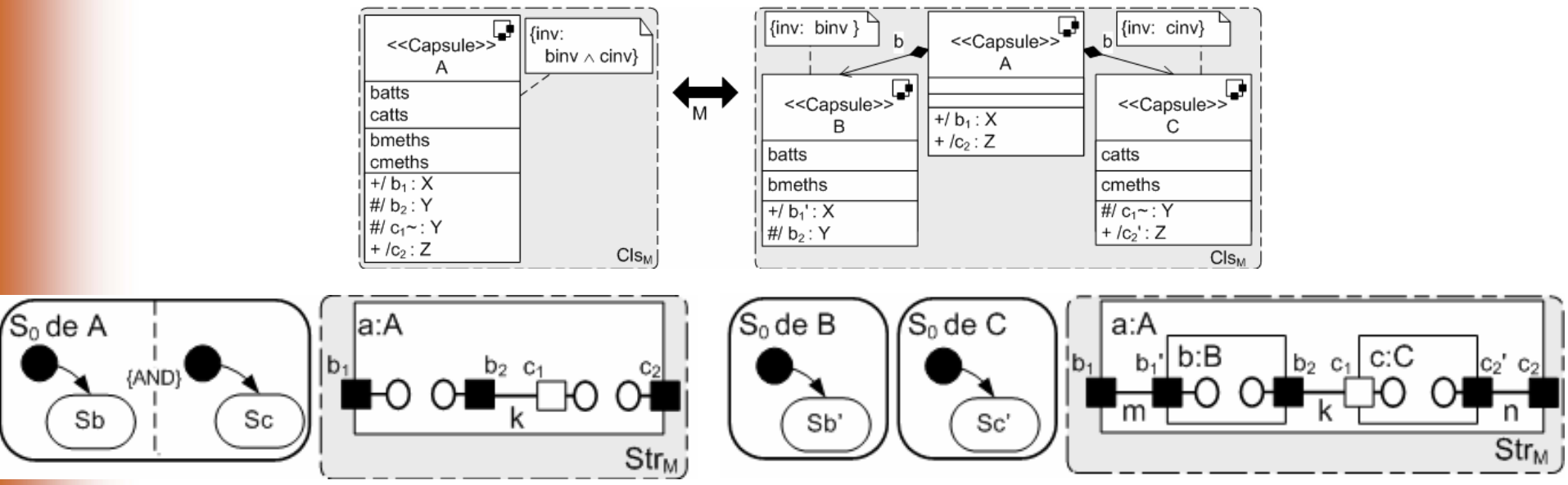
  (↔) O protocolo T e todos protocolos associados às portas em bps e aps' possuem uma máquina de estados determinística.

  ...

Lei 20 - Decomposição Paralela de uma Cápsula



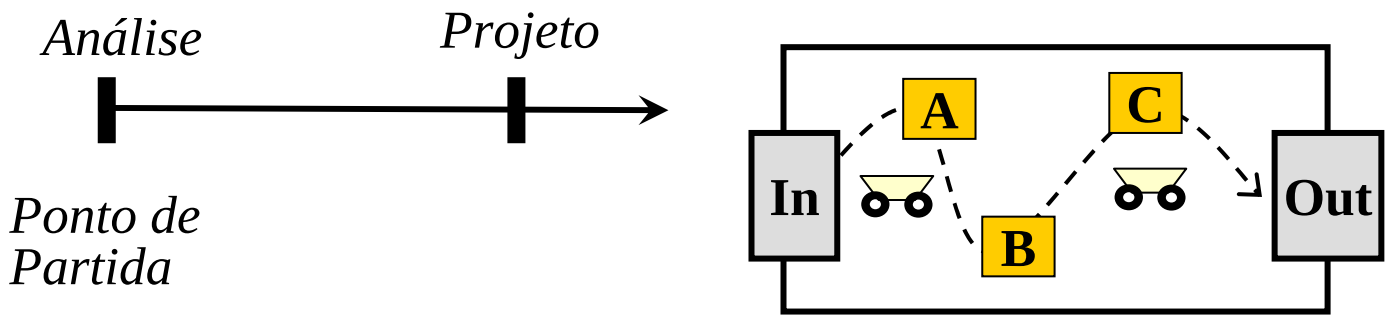
Lei 20 - Decomposição Paralela de uma Cápsula



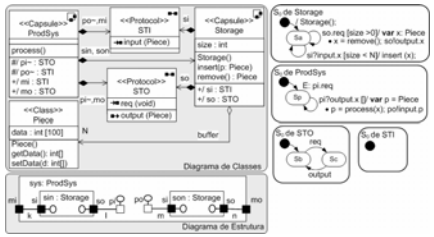
- Condições:
  - (→) <*hbatts, binv, bmeths, (b1, b2), Sbi*> e <*hcatts, cinv, cmeths, (c1, c2), Sci*> particionam A;
  - (↔) Os protocolos X e Z possuem uma m´aquina de estados determinística ...

Estudo de Caso
Desenvolvimento

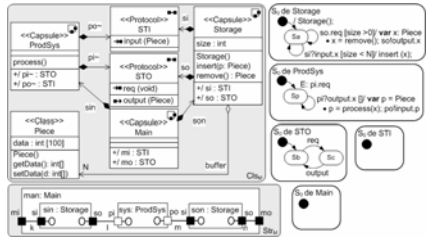
- Sistematização de atividades informais do RUP
  - Definição de uma arquitetura candidata
  - Identificação de elementos de projeto
- Enfoque no aumento de modulariadade do modelo
  - Leis para diagramas de classe e estrutura
- Concretização do Modelo



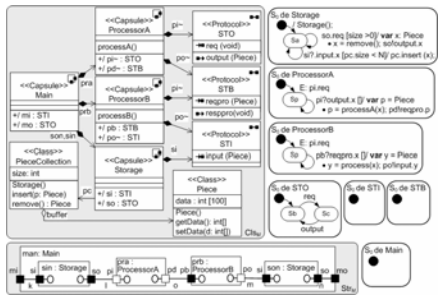
Estudo de Caso
Seqüência de passos



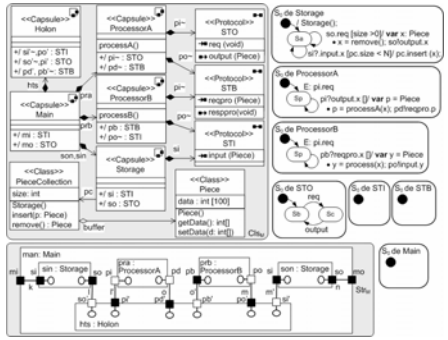
Ponto de Partida



Arquitetura Candidata



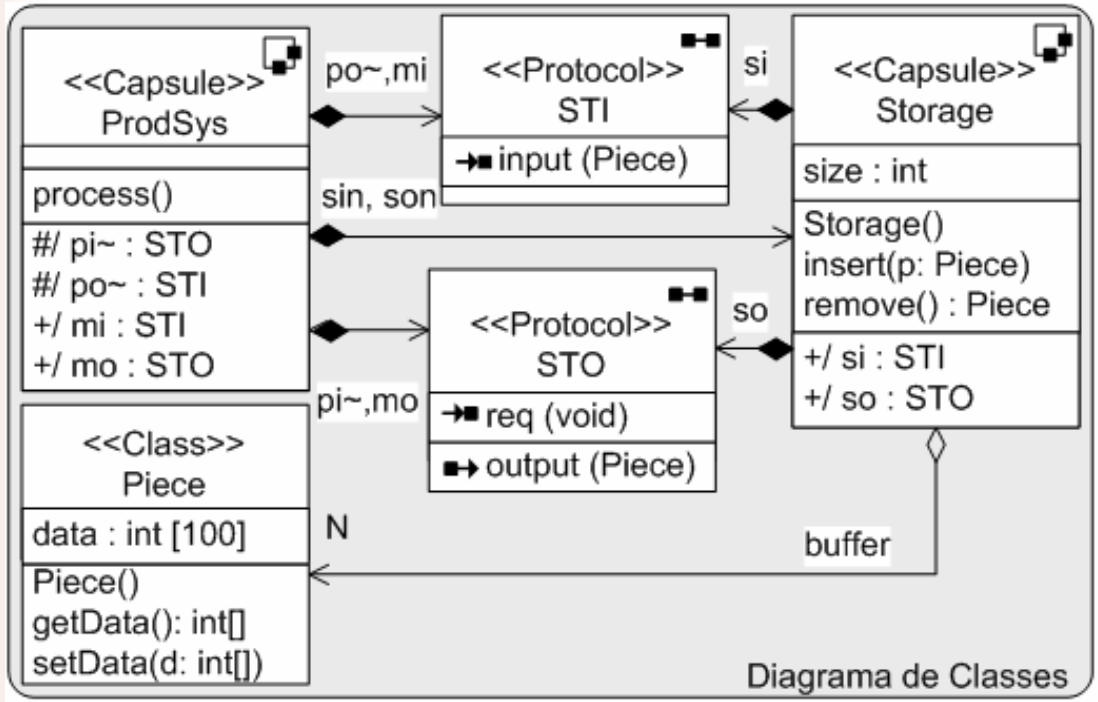
Identificação de Processadores

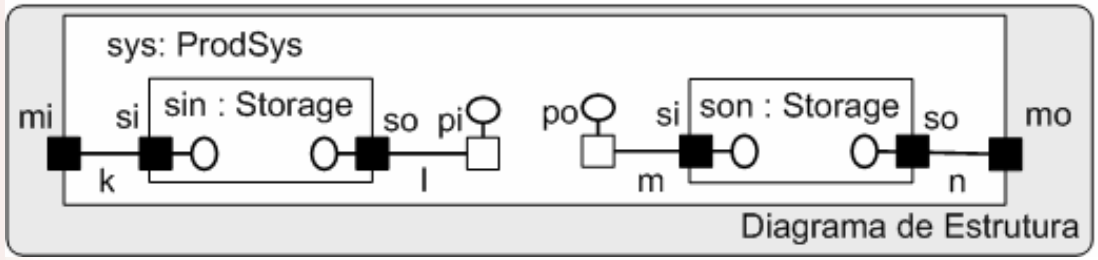


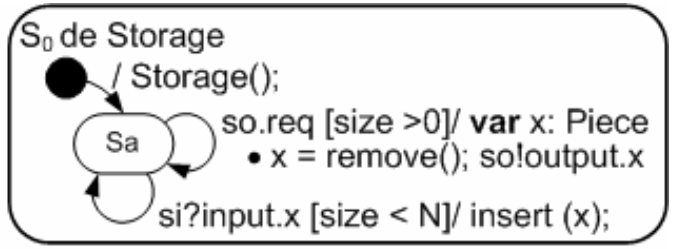
Identificação de Transportadores

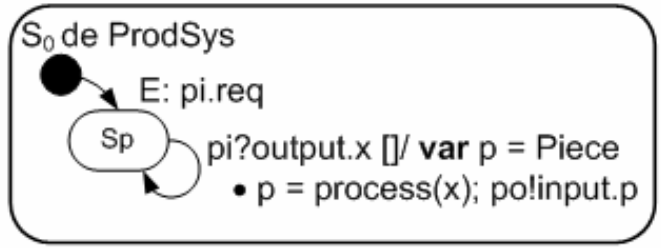
Estudo de Caso

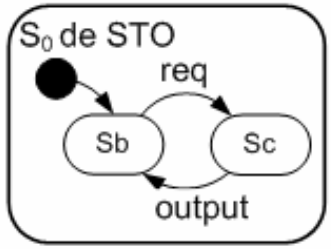
Ponto de Partida





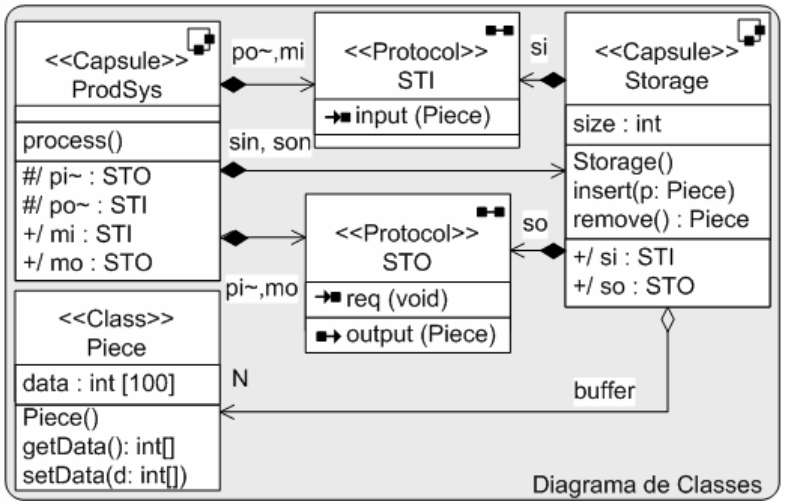


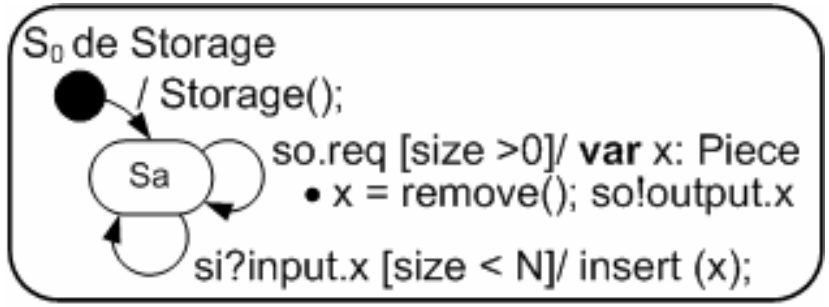


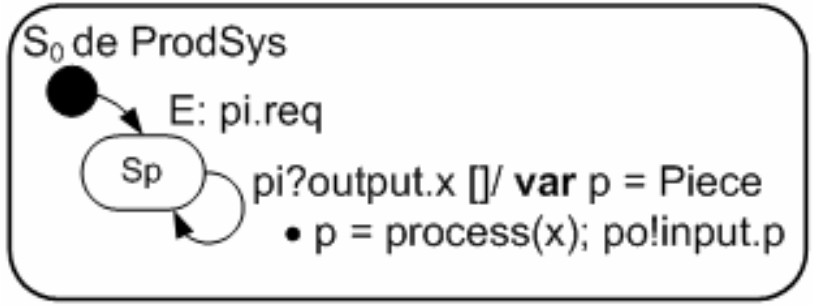


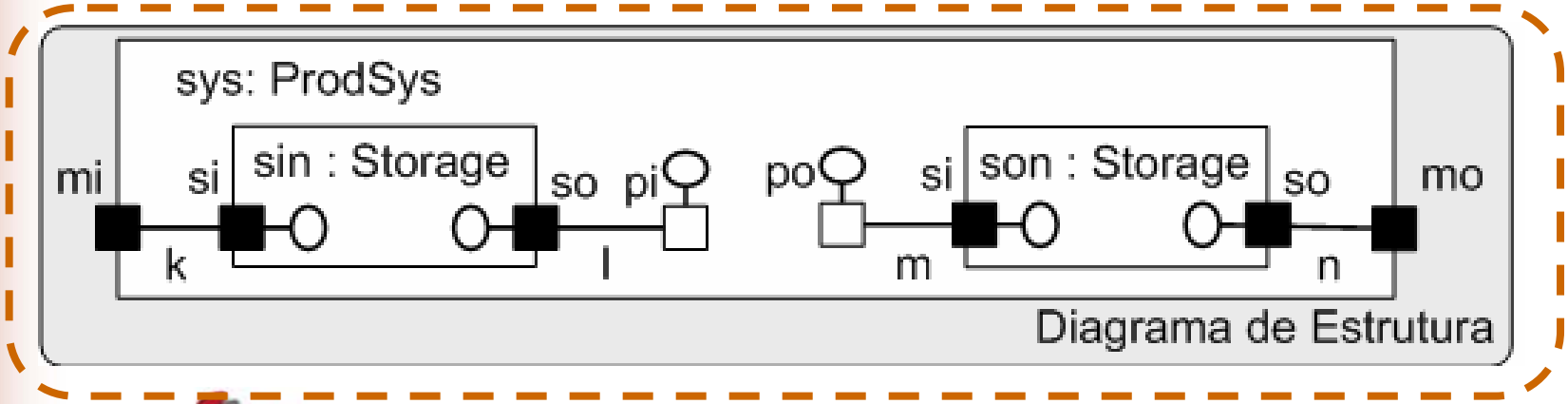
Estudo de Caso

Explicitar a cápsula Main



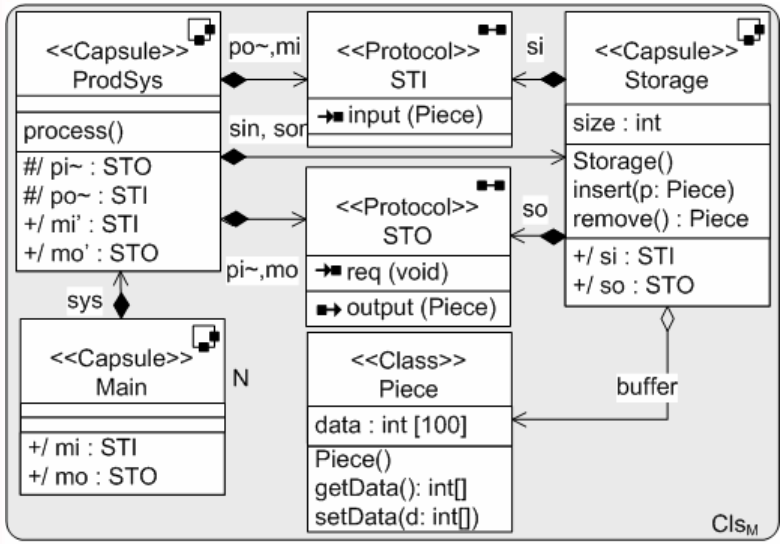


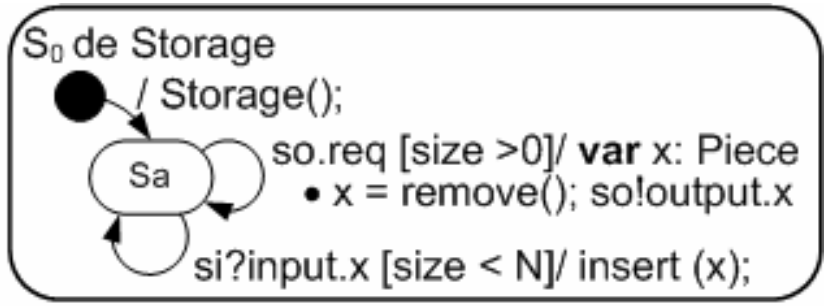


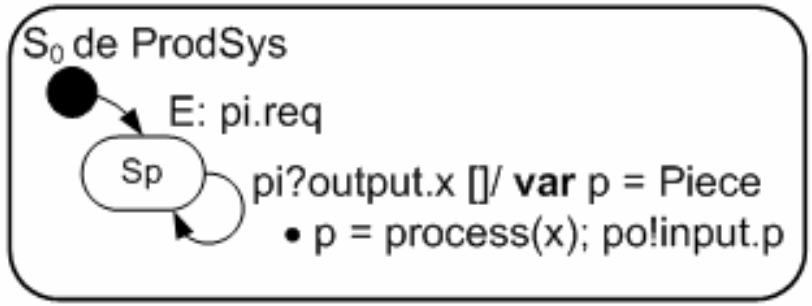


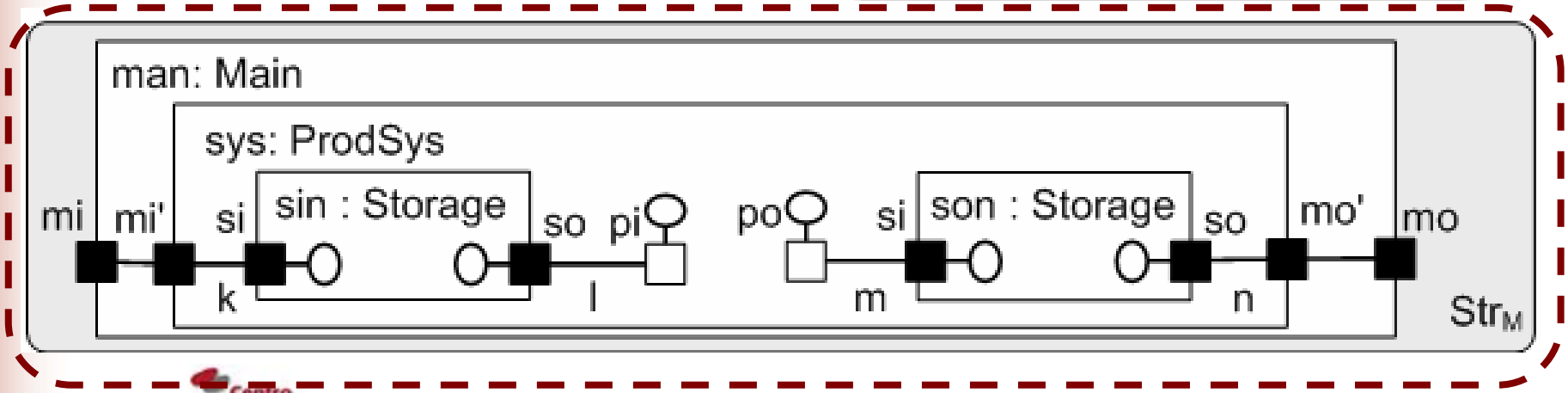
Estudo de Caso

Explicitar a cápsula Main



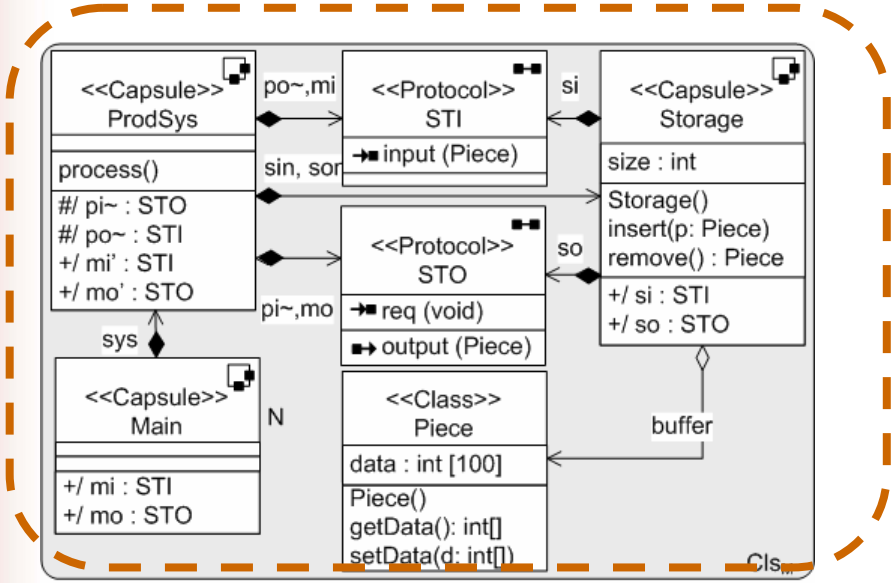


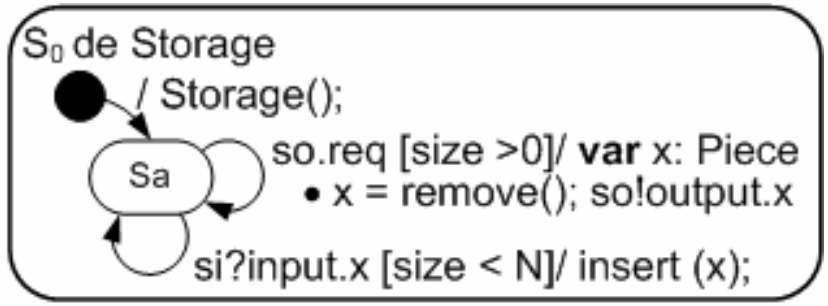


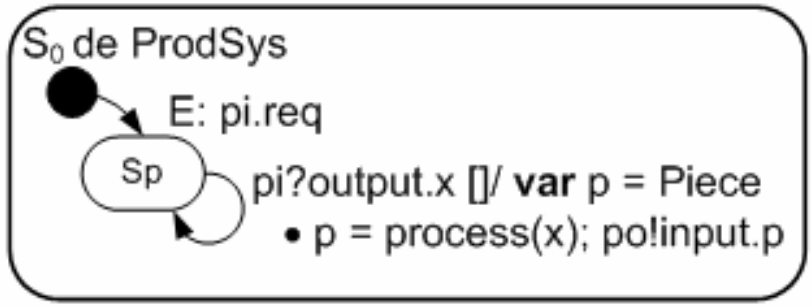


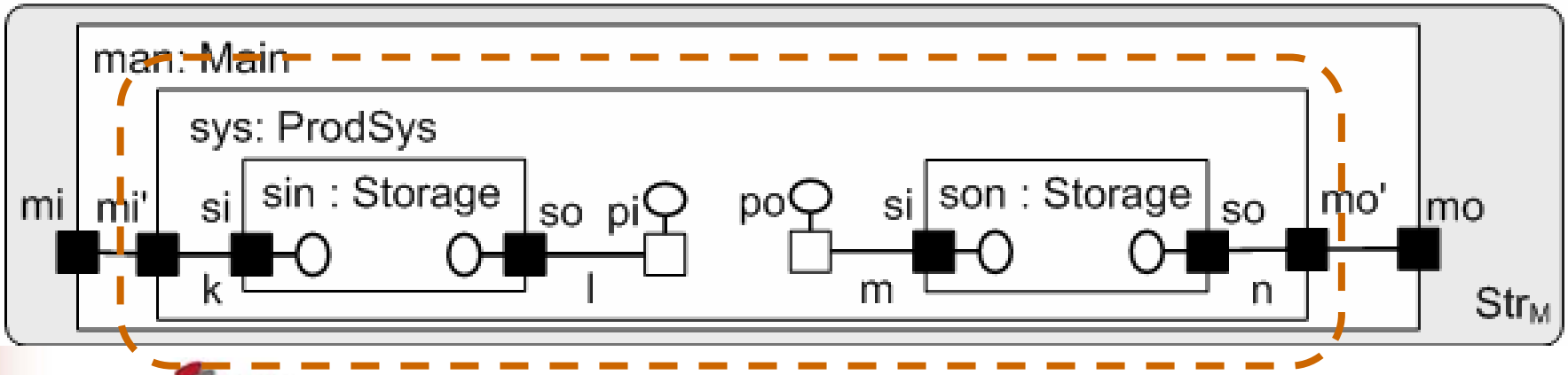
Estudo de Caso

Desencapsular Repositórios



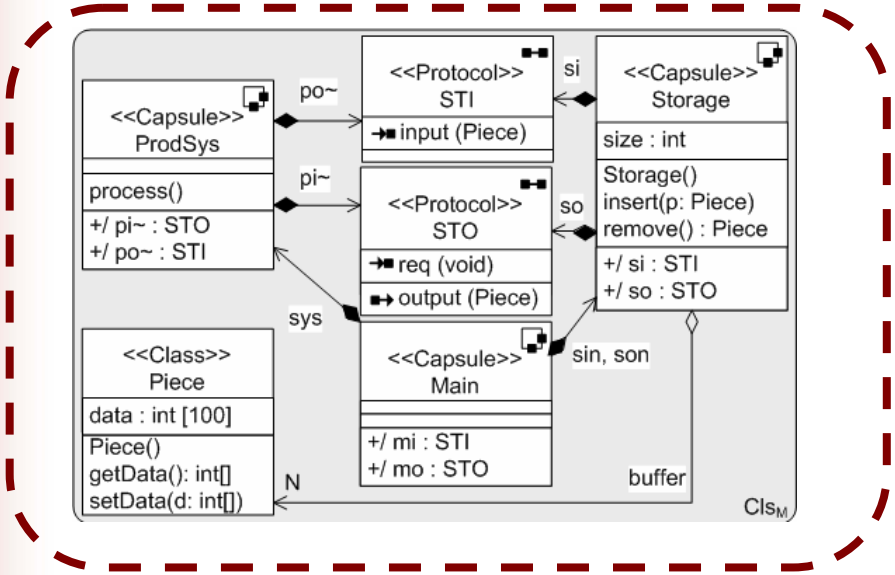


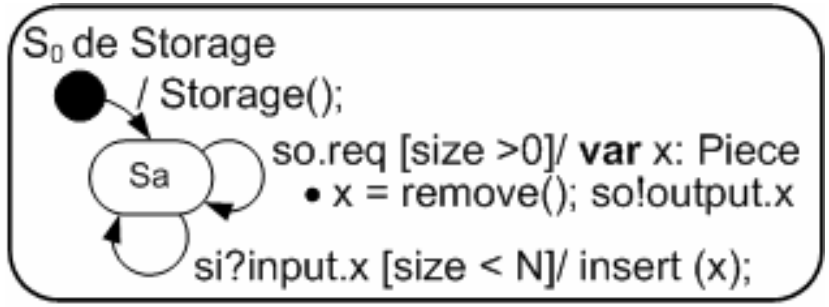


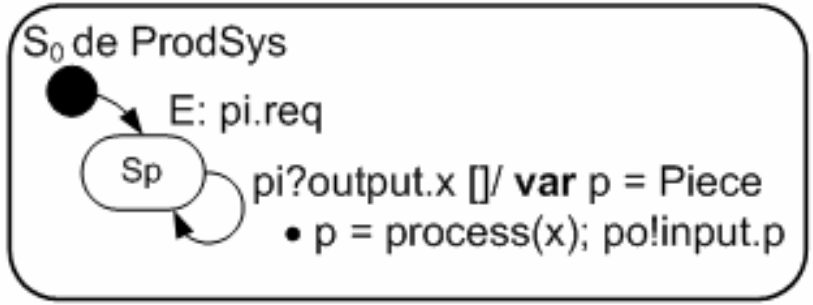


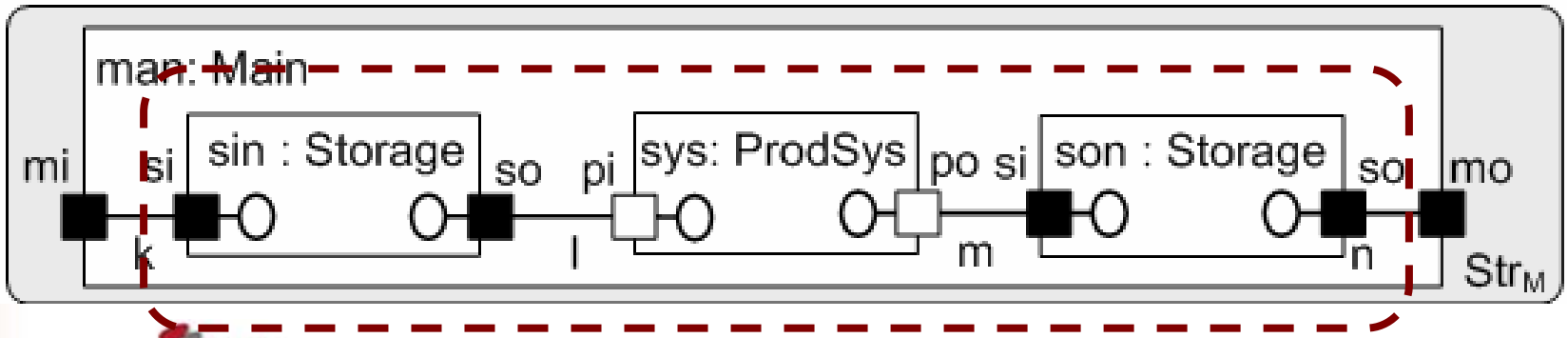
Estudo de Caso

Desencapsular Repositórios

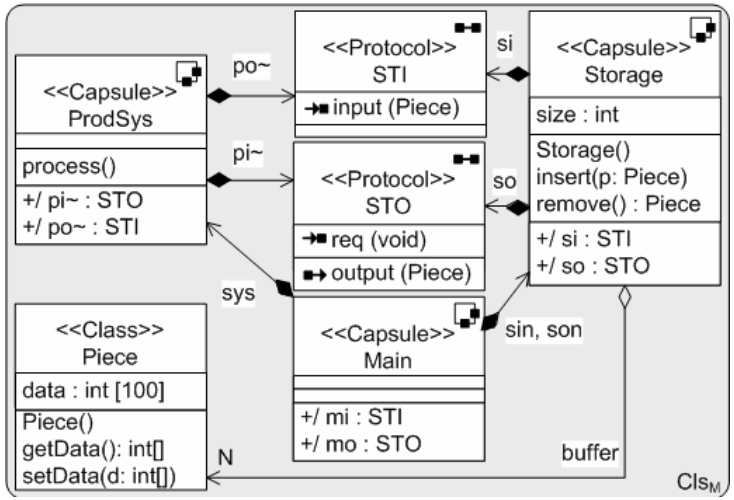


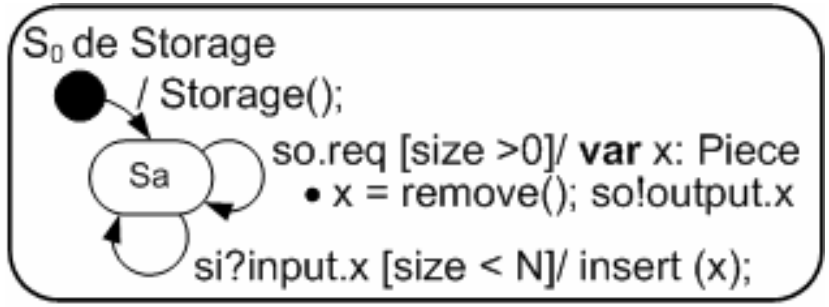


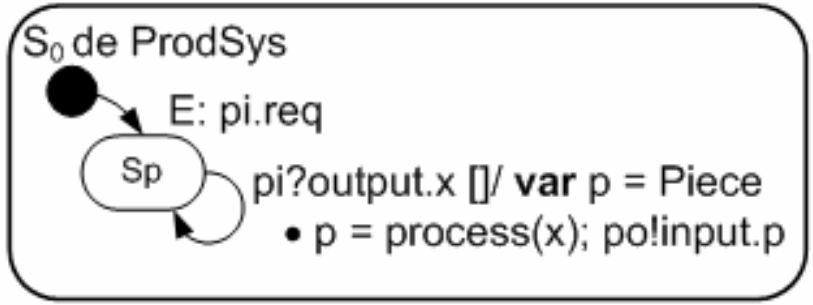


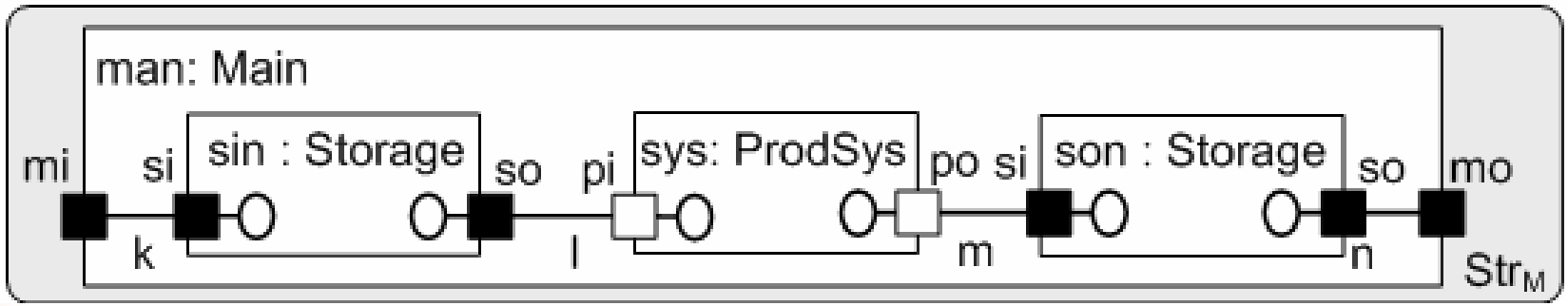


Estudo de Caso
Arquitetura Candidata



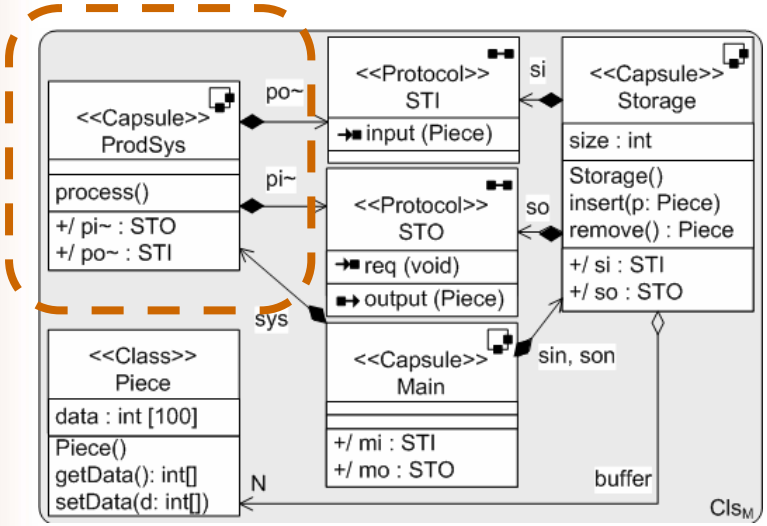


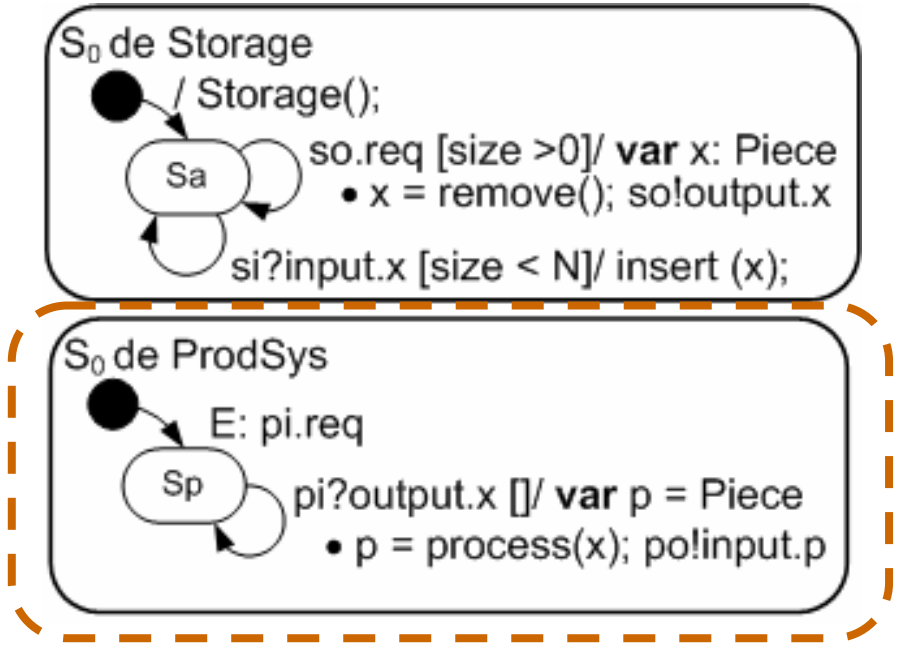


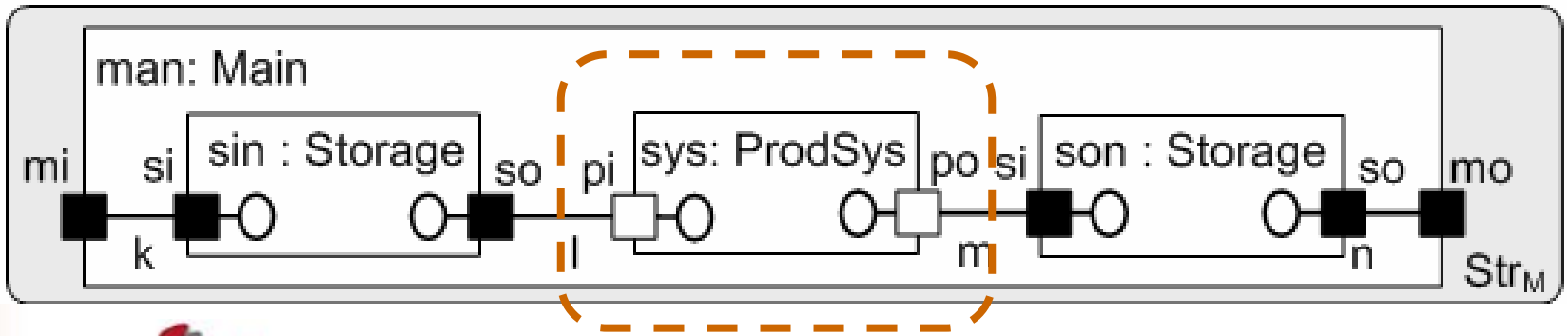


Estudo de Caso

Particionar ProdSys

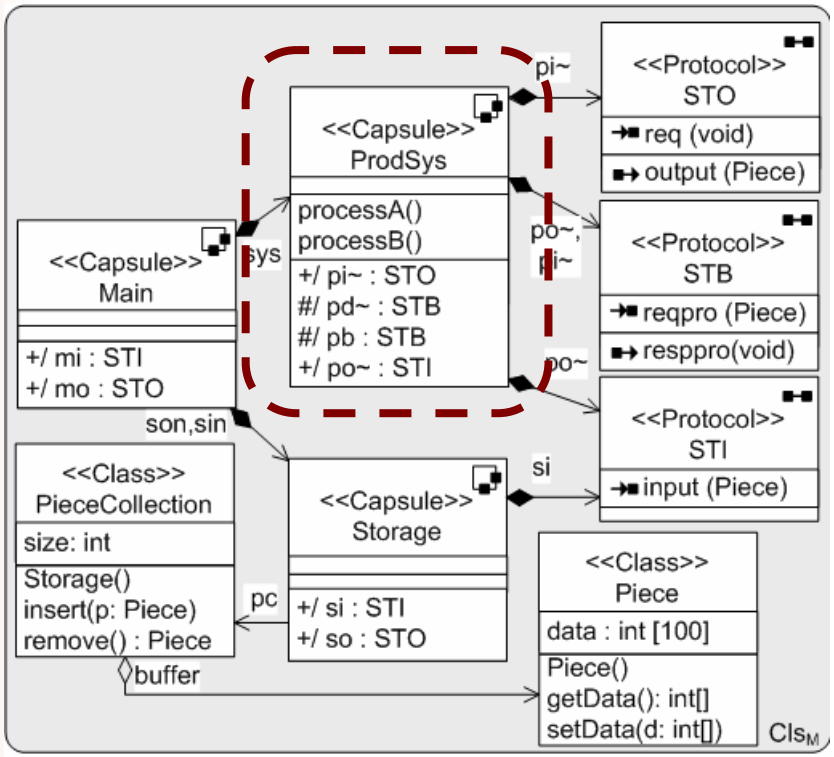


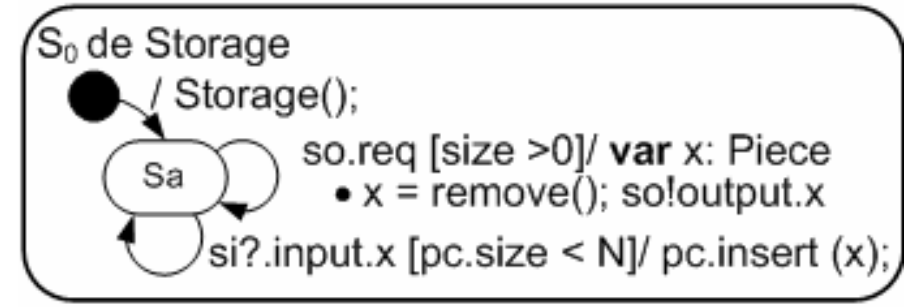


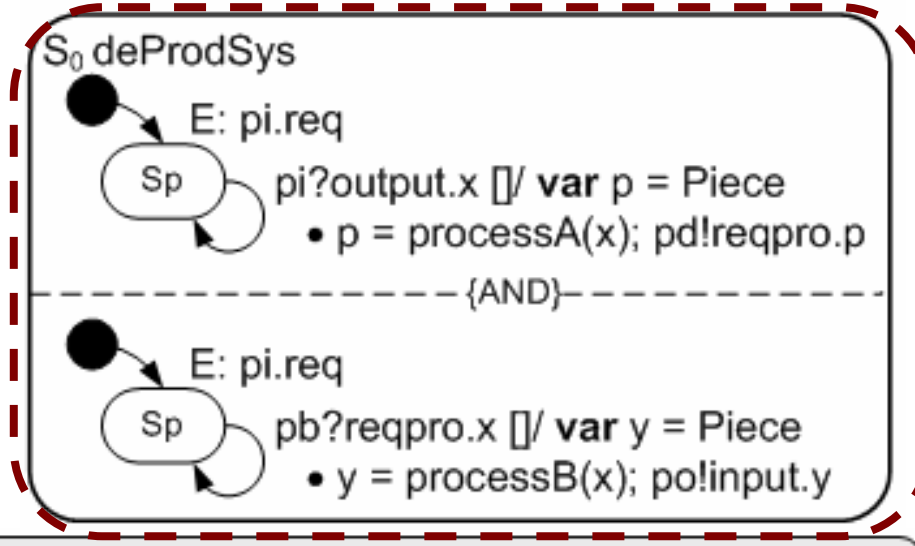


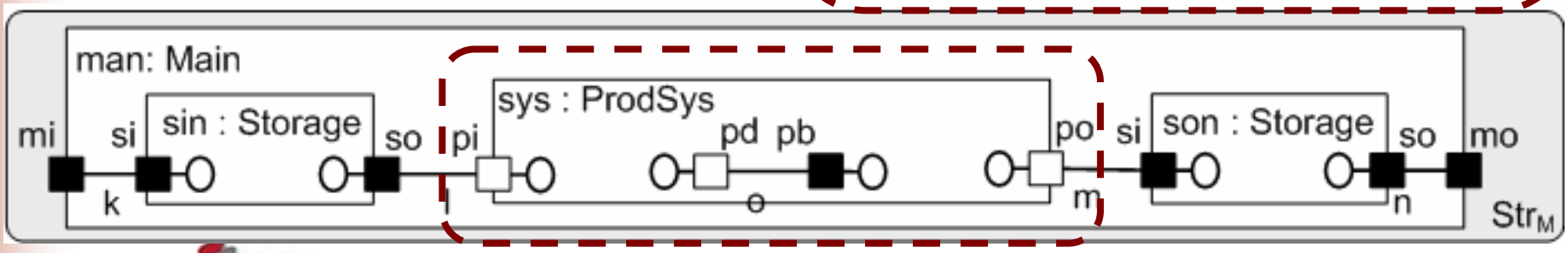
Estudo de Caso

Particionar ProdSys



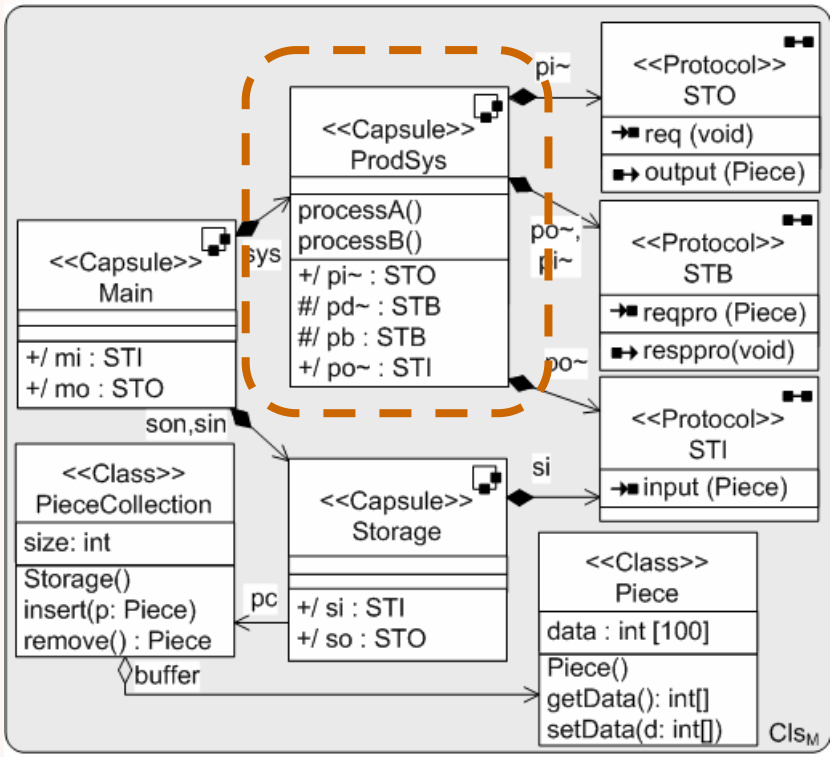


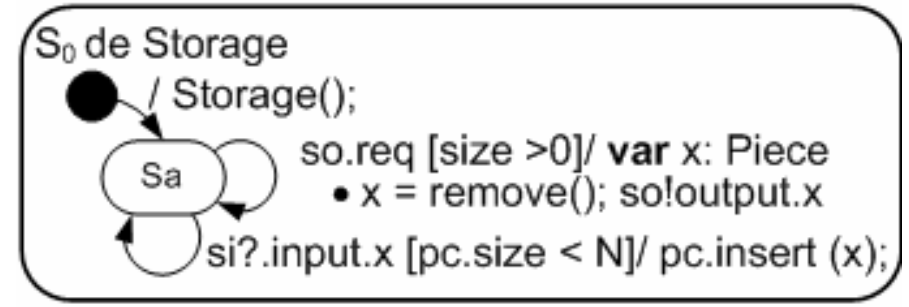


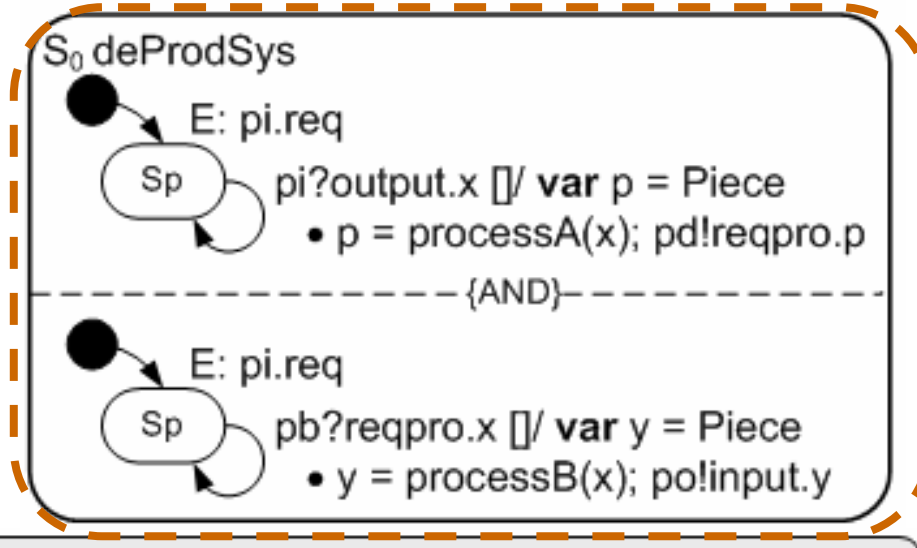


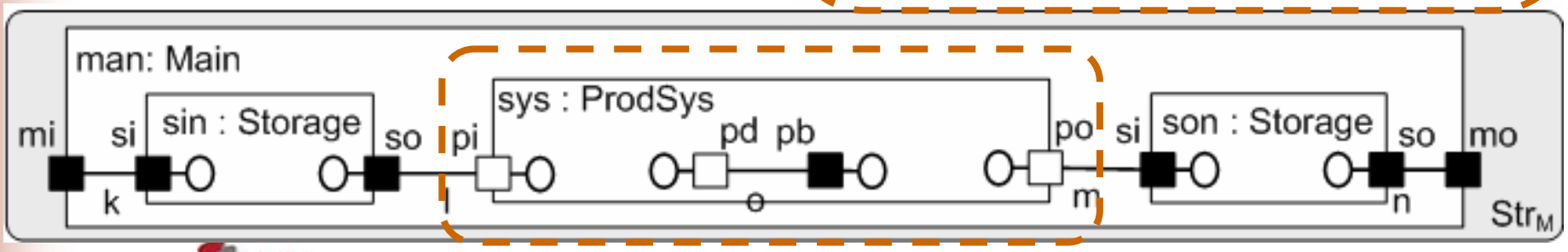
Estudo de Caso

Decompor ProdSys



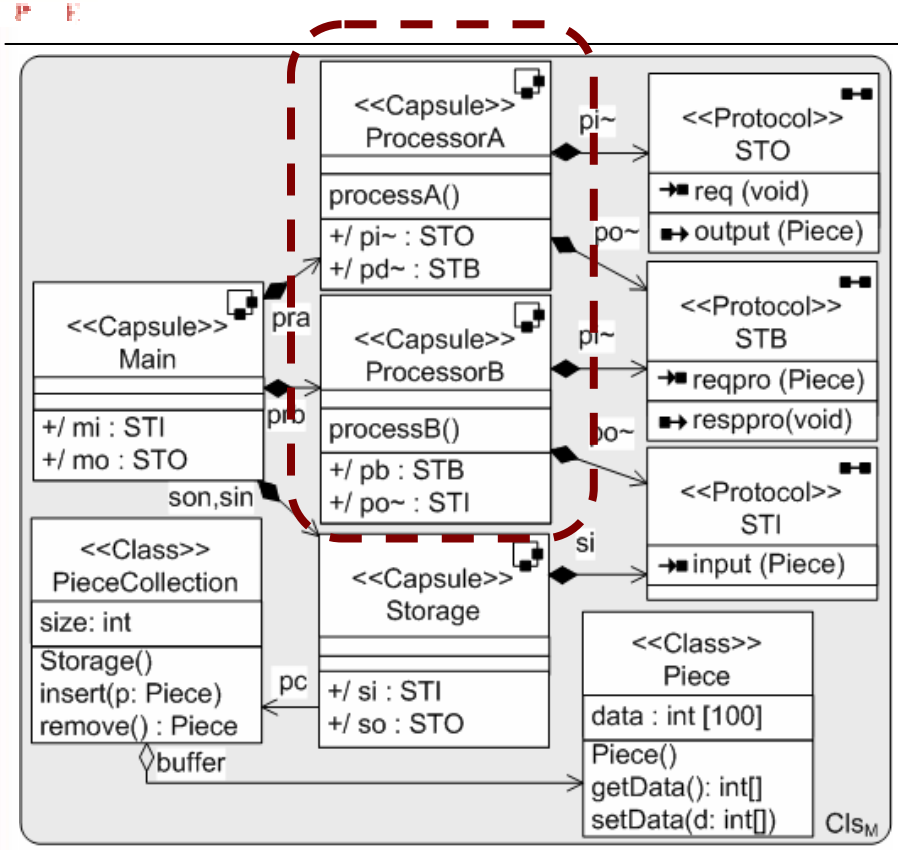


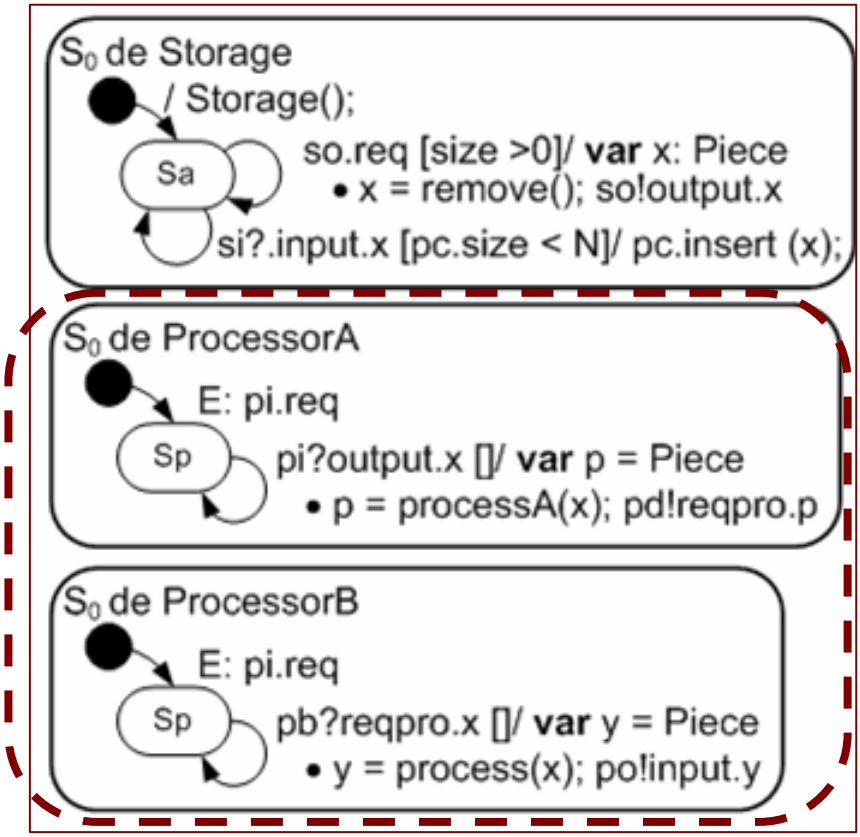


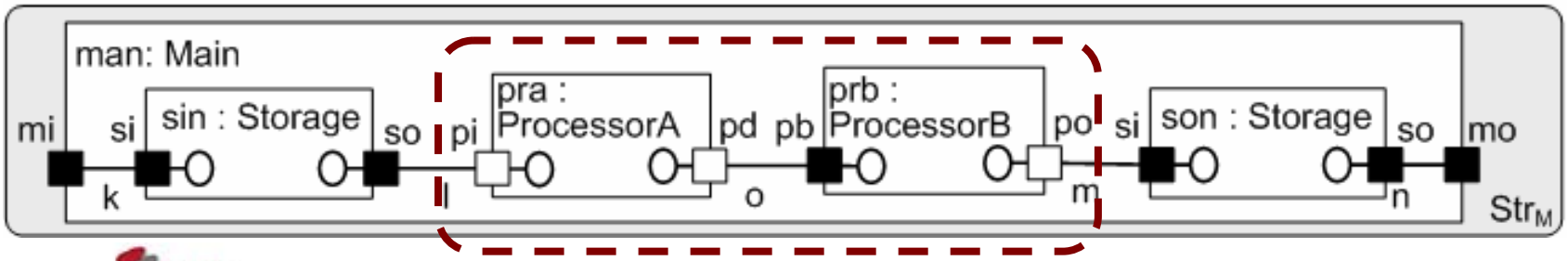


Estudo de Caso

Decompor ProdSys

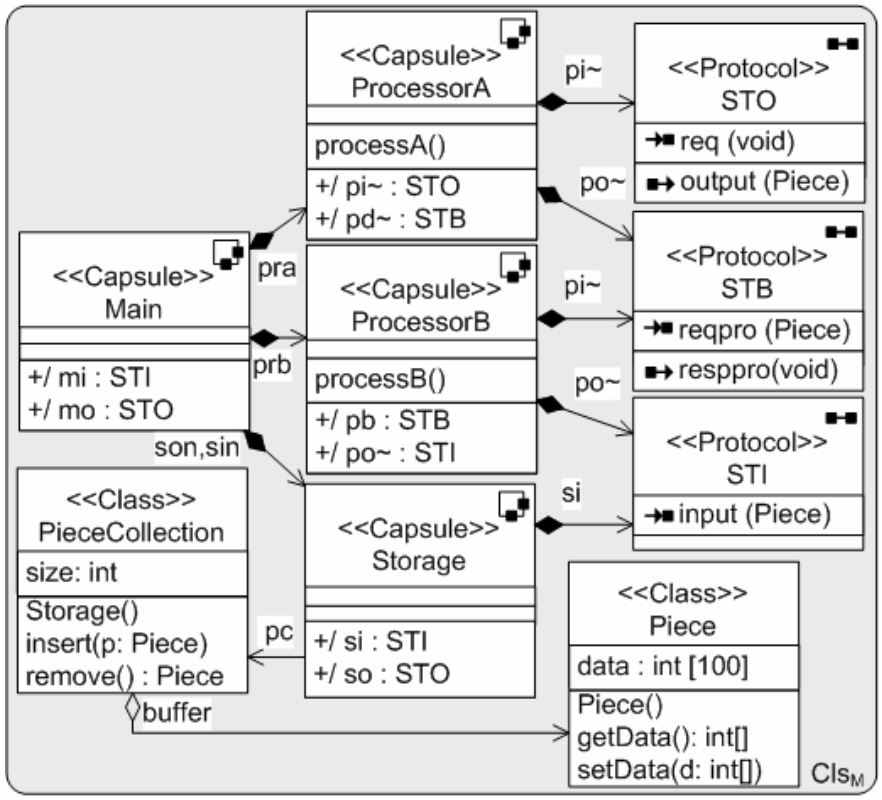


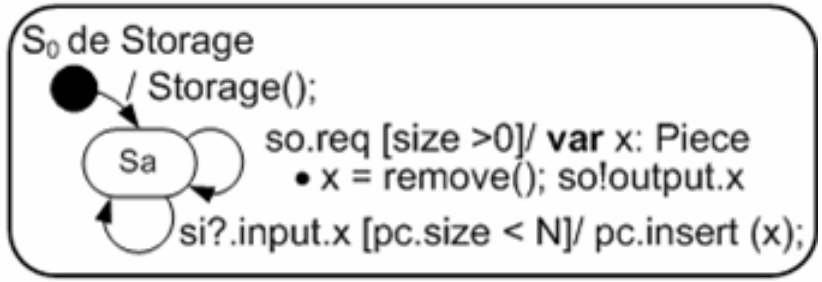


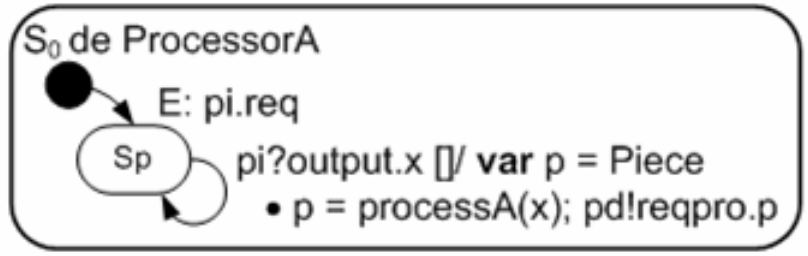


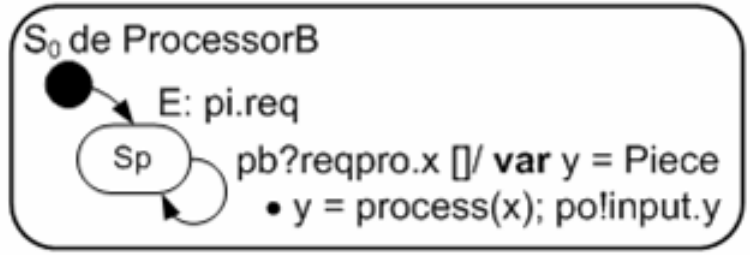
Estudo de Caso

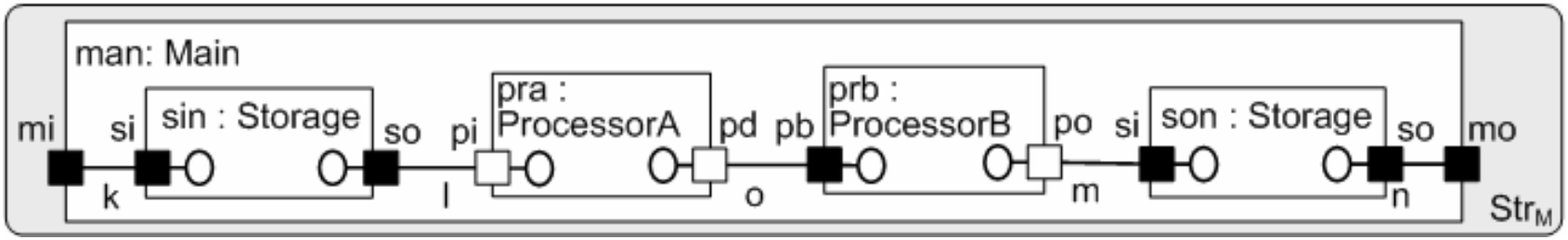
Identificação de Processadores





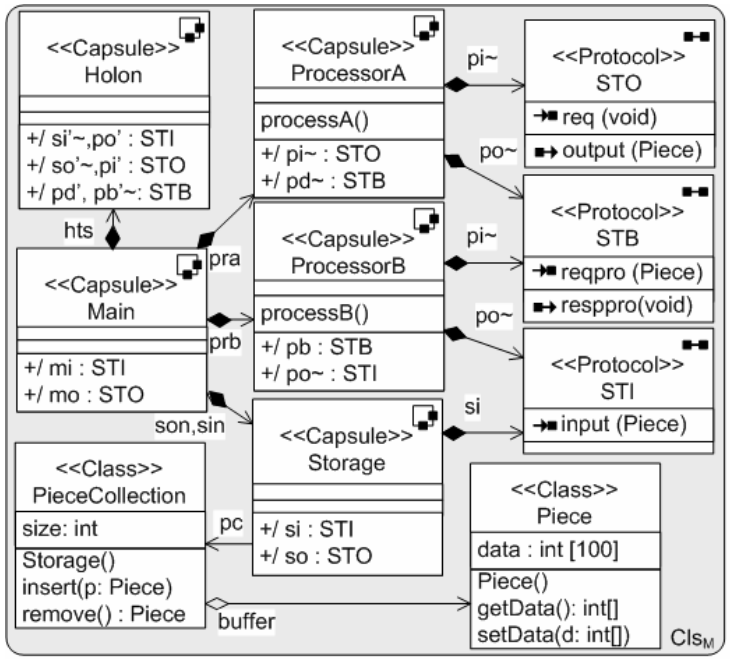


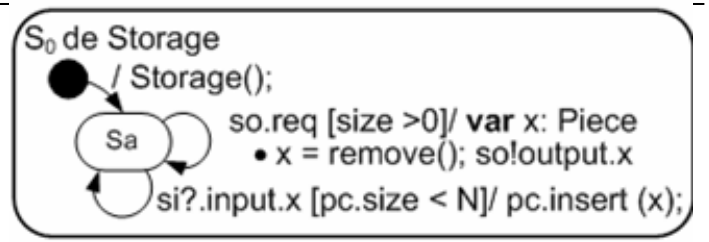


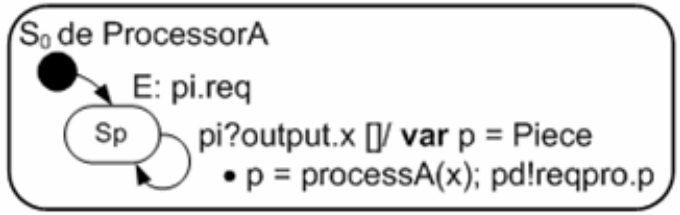


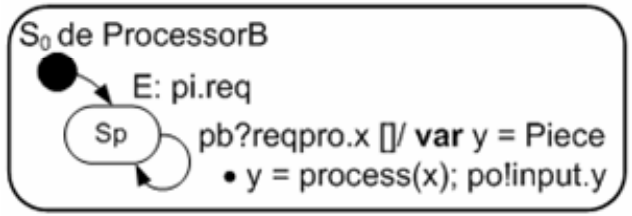
Estudo de Caso

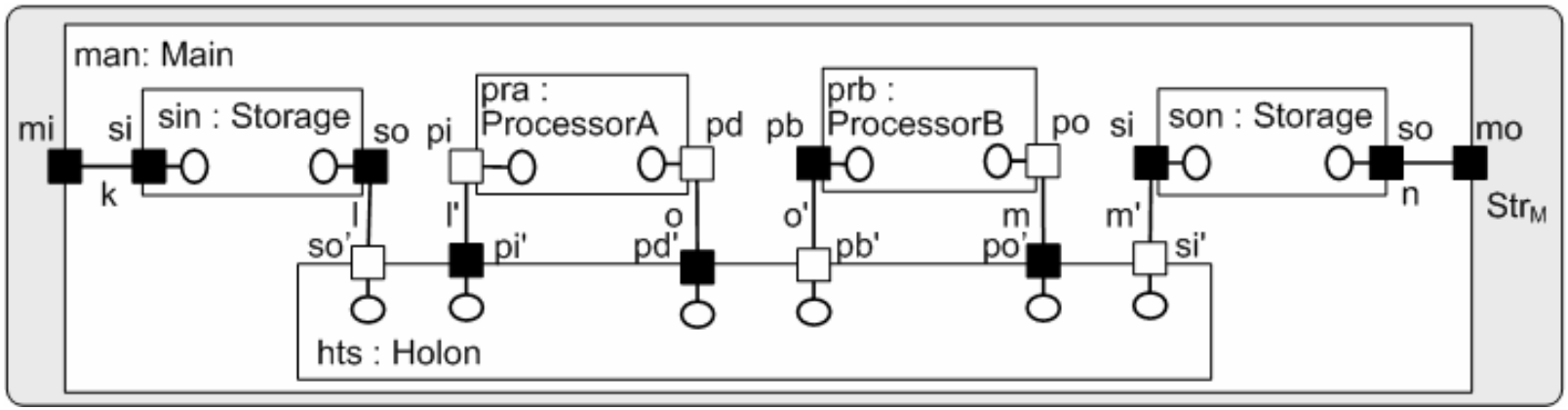
Modelo Final





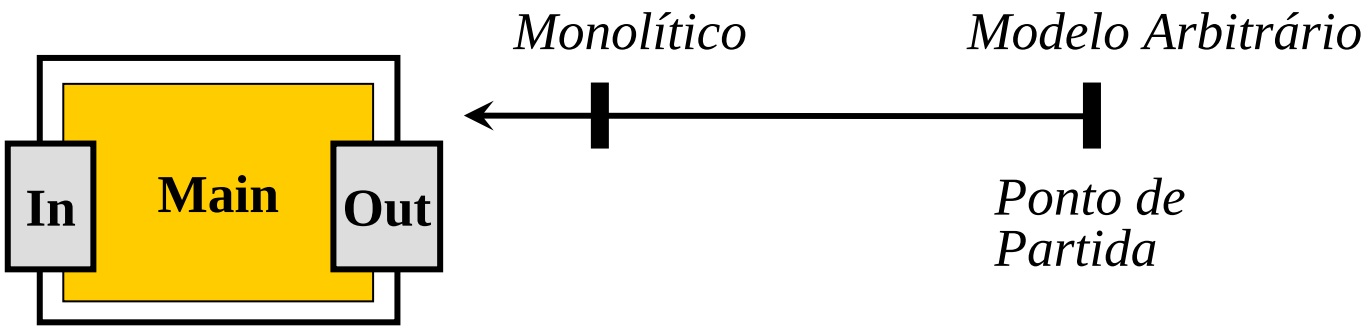




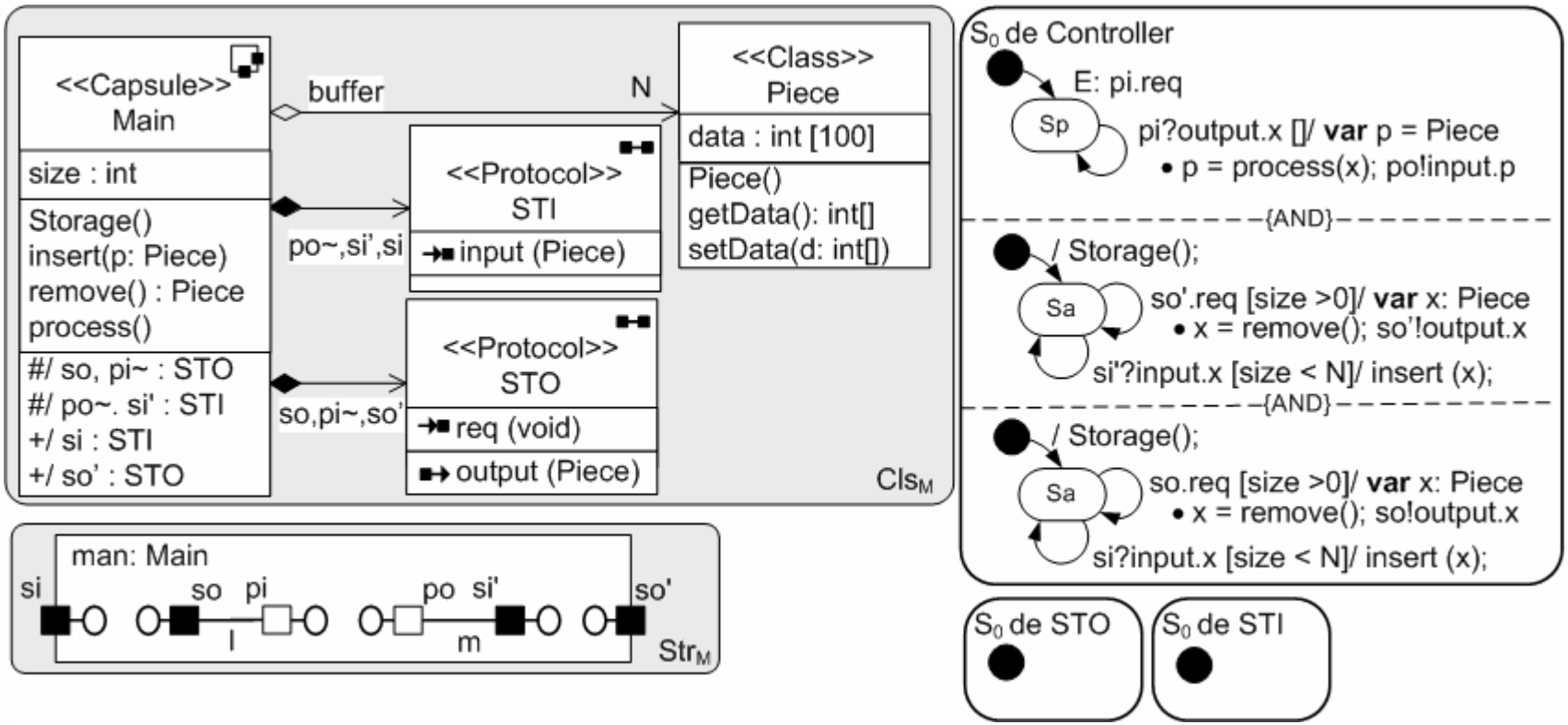


Estratégia de Normalização

- Estudo da abrangência do conjunto de leis
- Reduzir o modelo a uma única cápsula *Main*
  - *Composição de Portas*
  - *Remoção de hierarquias*
  - *Mover comportamento de protocolos para cápsulas*
  - *Composição de Cápsulas*
  - *Remoção de declarações não utilizadas*



Estudo de Caso
Forma Normal



Mapeamento de UML-RT para *OhCircus*

- Atribuição de semântica aos elementos de UML-RT
- *OhCircus* estende *Circus* com conceitos de OO
  - *Circus* combina Z , CSP
  - Definido utilizando a UTP (*Unifying Theories of Programming*)

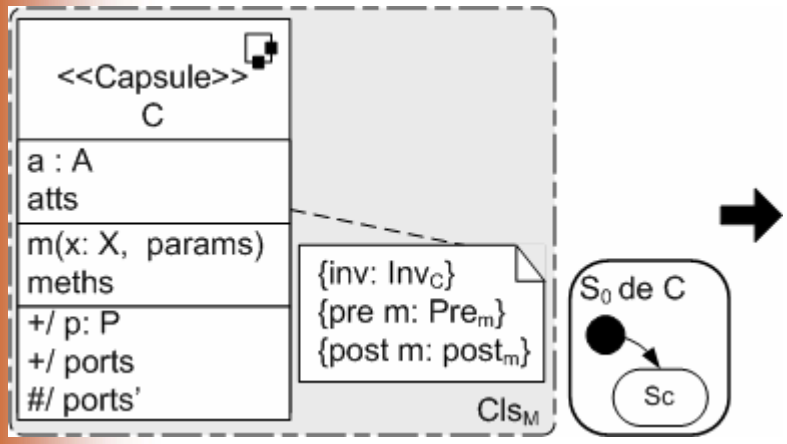
Mapeamento de UML-RT para *OhCircus*

- Mapeamentos da estrutura e do comportamento (*statechart*) dos elementos

- Ênfase nos novos elementos encontrados em UML-RT
  - Classes de UML ⇒ classes de *OhCircus*
  - Cápsulas e Protocolos ⇒ processos
  - Portas ⇒ canais
  - Conexões ⇒ Renomeação das portas para canais comuns
- Suposições
  - Predicados são descritos como invariante de classe e cápsula
  - Associações entre classes são vistas como atributos
  - Modelo síncrono de comunicação
  - Semântica de cópia

Mapeamento das Cápsulas

- Cápsulas = 2 processos
  - Estática: métodos, atributos e máquina de estados
  - Configuração dinâmica : paralelismo de seu comportamento, o de suas sub-cápsulas, e de suas portas
- Mapeamento da visão estática de uma cápsula:



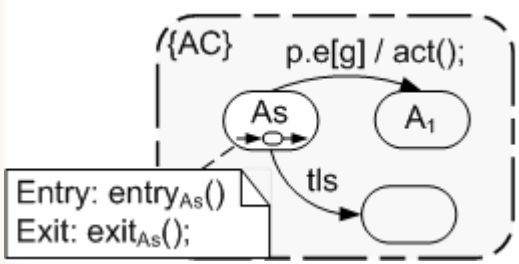
$$
\begin{array}{l}
\textbf{channel}\ \mathsf{p} : \mathcal{T}_{\mathsf{P}};\ \mathcal{TL}(\mathsf{ports});\ \mathcal{TL}(\mathsf{ports}') \\
\textbf{process}\ \mathit{Chart}_{\mathsf{C}} \mathrel{\widehat{=}} \textbf{begin} \\
\qquad \textbf{state}\ \mathsf{C}_{state} \mathrel{\widehat{=}} [\mathsf{a} : \mathcal{T}(\mathsf{A});\ \mathcal{TL}(\mathsf{atts}) \mid \mathsf{Inv}_{\mathsf{C}}] \\
\qquad m \mathrel{\widehat{=}} [\Delta \mathsf{C}_{state};\ \mathsf{x} : \mathcal{T}(\mathsf{X}); \\
\qquad\qquad\qquad\qquad \mathcal{TL}(\mathsf{params}) \mid \mathsf{Pre}_{\mathsf{m}} \wedge \mathsf{Post}_{\mathsf{m}}] \\
\qquad \mathcal{TL}(\mathsf{meths}) \\
\qquad \bullet\ \mathcal{H}(\mathsf{S}_{\mathsf{C}}) \\
\textbf{end}
\end{array}
$$

Mapeamento de Máquinas de Estados

- *Statecharts* das Cápsulas
  - Estados são mapeados em processos de ações

$$\mathcal{H} : S_M \rightarrow CSPAction$$

- Ex.: Mapeamento de um estado simples:



$$\begin{aligned}\mathcal{H}(\mathsf{As}) = \mathsf{entry}_{\mathsf{As}}();\ ((\mathsf{g}\ \&\ \mathsf{p.e} \rightarrow \mathsf{exit}_{\mathsf{As}}();\ \mathsf{act}();\ \mathcal{H}(\mathsf{A}_1)) \\ \Box\ ((\mathit{notevent}_{\mathsf{AC}} \rightarrow \mathsf{exit}_{\mathsf{As}}();\ \mathit{exitevent}_{\mathsf{AC}} \rightarrow \mathit{SKIP}) \\ \Box\ \mathcal{TL}(\mathsf{tls})))\end{aligned}$$

OhCircus

- Mapeamento da cápsula *Storage no processo* $Chart_{Storage}$

$| \; N : \mathbb{N}$

[redacted]

process $Chart_{Storage} \mathrel{\widehat{=}}$ **begin**

state $StorageState \mathrel{\widehat{=}} [\mathsf{buff} : \mathsf{seq}\,\mathsf{Piece};\ \mathsf{size} : 0..N \mid \mathsf{size} = \#\mathsf{buff} < N]$

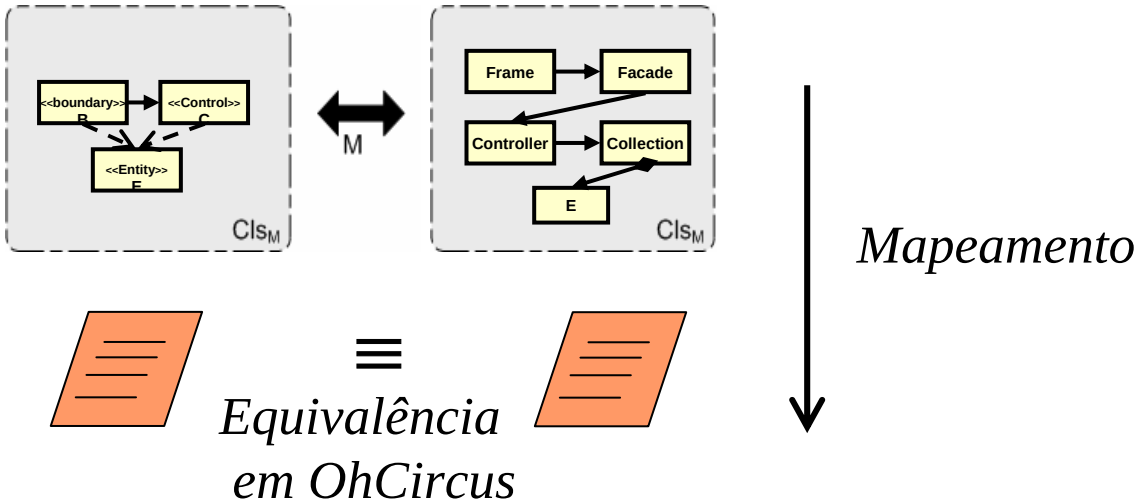
[redacted]

- [redacted]

end

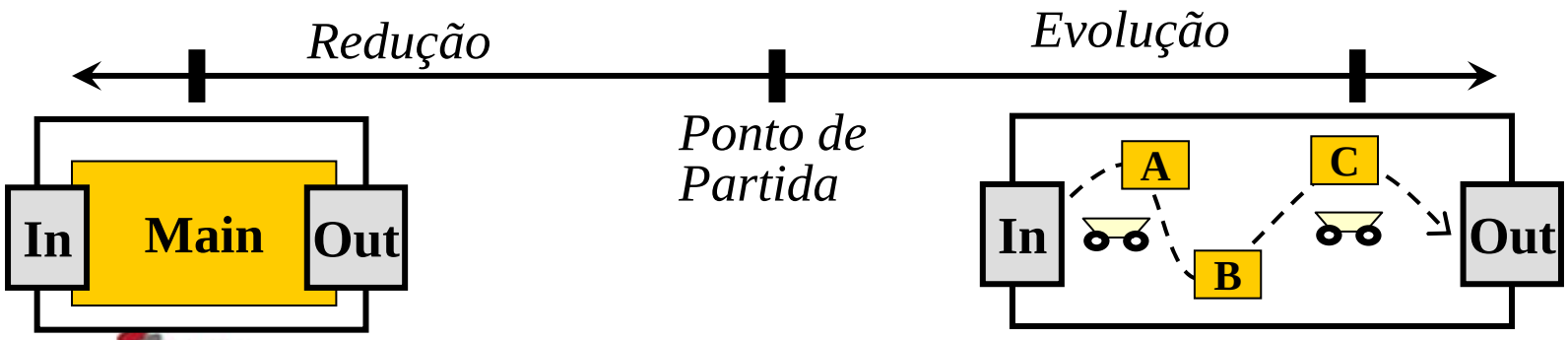
Prova das Leis

- Utilização do <u>cálculo de refinamentos de *OhCircus*</u> *e da semântica definida na UTP*
- Prova de duas leis utilizando o mapeamento
  - Lei 11 – Encapsular Cápsula
  - Lei 20 – Decompor Cápsula



Conclusões

- Atribuição de uma semântica formal a UML -RT
- Proposição de um conjunto abrangente de leis básicas de transformação para UML-RT
- Derivação de leis de projeto para UML-RT
- Prova de Leis
- Aplicação das leis durante os principais passos de uma estratégia de normalização e durante o desenvolvimento de um estudo de caso



Trabalhos Relacionados

- Mapeamento de UML
  - diagramas de classe → Z [Bruel97, Evans98, Amalio03]
  - diagramas de estado → CSP [Ng03, Rasch02]
- Mapeamento de UML-RT
  - Diagramas de estrutura → CSP [Fischer01, Engels01]
  - Estudo de caso em CSP-OZ [Möller04]
- Transformações para UML [Evans99, Lano99, Sunyé01]
  - Leis para Alloy [Gheyi04]
- Transformações informais para UML-RT
  - Pequeno conjunto de leis [Engels02, McClean03]
  - Aplicação [Sandner00, Engels02]

Trabalhos Futuros

- Prova formal de:
  - todo o conjunto de leis básicas
  - estratégia de normalização
- Aplicação na:
  - realização de novos estudos de caso
  - derivação de padrões de projeto para UML-RT
- Ambiente de apoio à aplicação de leis
- Incorporação das contribuições deste trabalho
  - outras linguagens de modelagem (UML 2.0 e ADLs)

LMF – Grupo ForMULa

Desenvolvimento Rigoroso com UML-RT

Rodrigo Teixeira Ramos
UFPE – Cin