Engenharia de Software

Modelos de Dinâmica – Parte 2

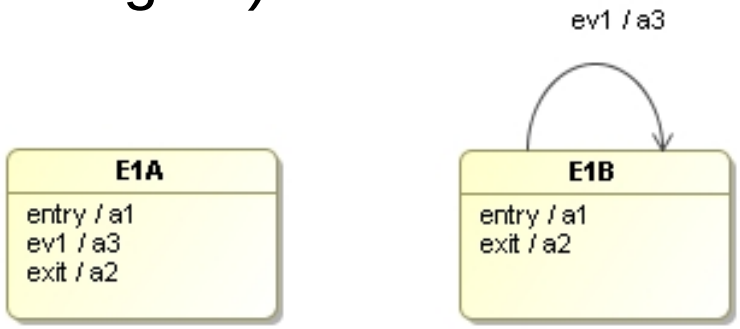
Luís Morgado

Instituto Superior de Engenharia de Lisboa

Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

Transições Internas

- Eventos podem produzir execução de acções sem causar transição de estado
  - **Transição interna**
  - Análogo a ***entry*** e ***exit*** mas associadas a eventos específicos
  - Caso não existam acções *entry* e *exit*, comportamento idêntico a auto-transições (estado destino = estado origem)

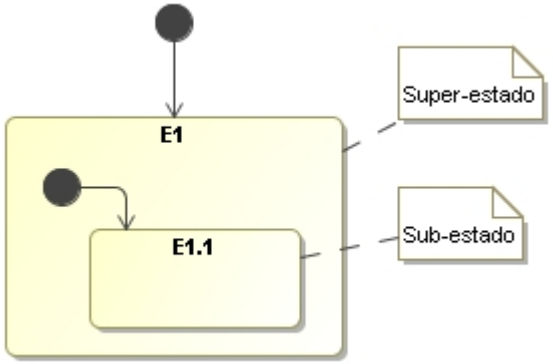


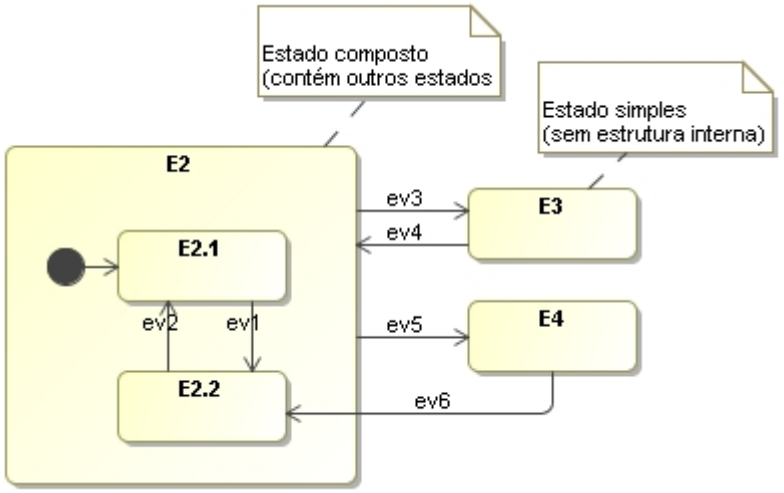
Máquinas de Estados Hierárquicas

- Partilha de comportamento
  - Subestados apenas necessitam de especificar as diferenças de comportamento em relação aos respectivos superestados
  - Reutilização de comportamento
    - Permite **abstrair o que é comum**
      - Automaticamente processado nos níveis superiores
    - Subestados **partilham comportamento** com o superestado
- Abstracção
  - **Controlo de complexidade**
    (redução selectiva de complexidade)
    - *Zoom IN/OUT*
  - Sem abstracção mesmo sistemas moderadamente complexos tornam-se difíceis de modelar e implementar

Máquinas de Estados Hierárquicas

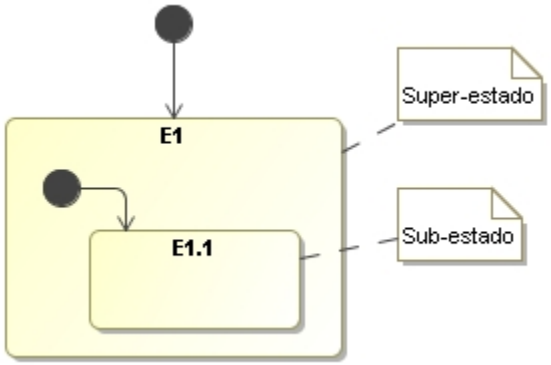
- Estruturação do modelo a diferentes níveis de abstracção





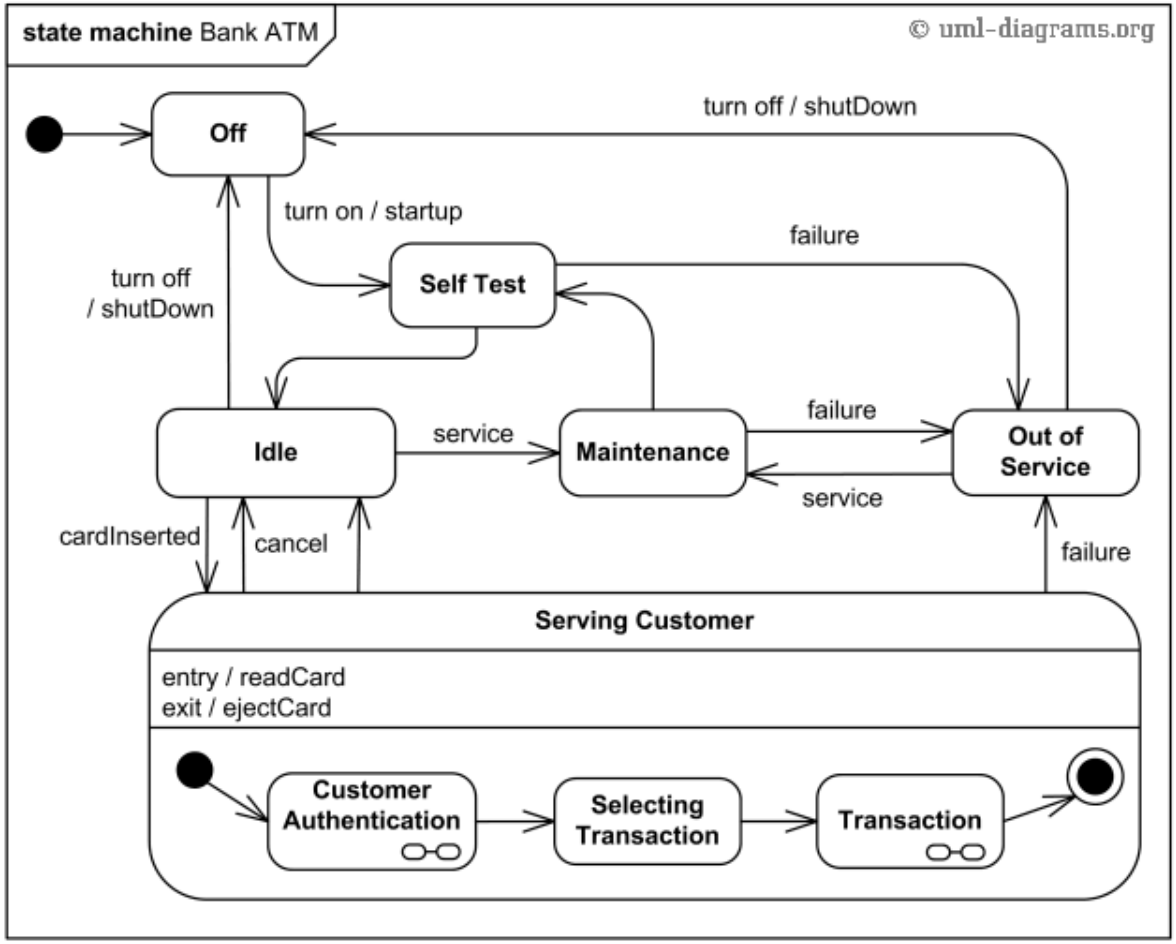
Máquinas de Estados Hierárquicas

- **Semântica**
  - Se um sistema está num subestado E1.1 também está (implicitamente) no respectivo superestado E1
    - **Estado composto**
  - Se o sistema está no subestado E1.1, o processamento dos eventos será feita nesse contexto.
    - Se não for definido um evento para E1.1 ?
      - Será processado no contexto do estado mais geral E1



Máquinas de Estados Hierárquicas

Exemplo

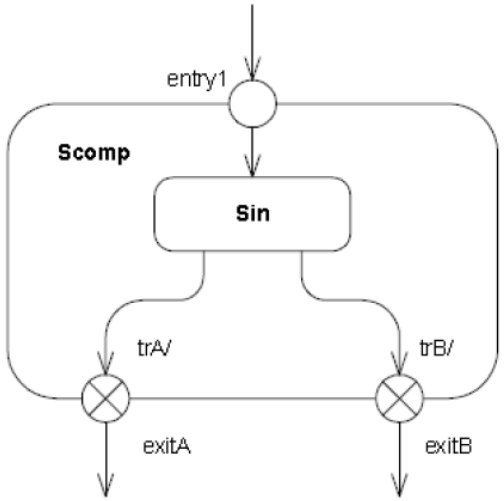


Máquinas de Estados Hierárquicas

Organização de sub-máquinas de estado

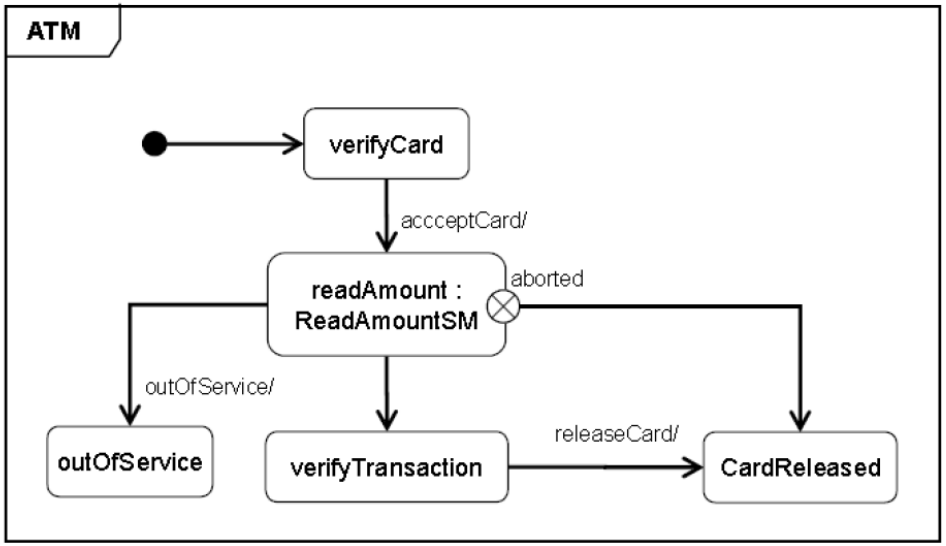
Pseudo-estados

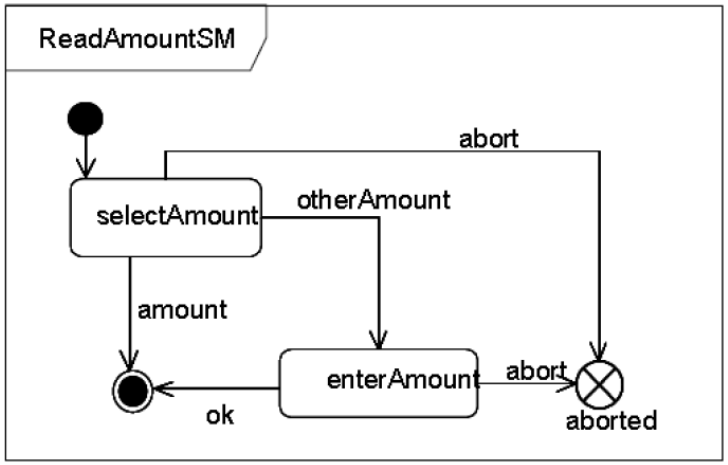
- **Entrada (*entry*)***:* Representa entrada num estado composto
- **Saída (*exit*)***:* Representa saída de um estado composto



Máquinas de Estados Hierárquicas

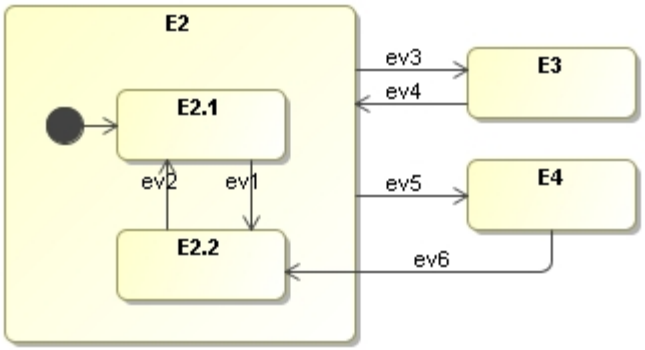
Exemplo

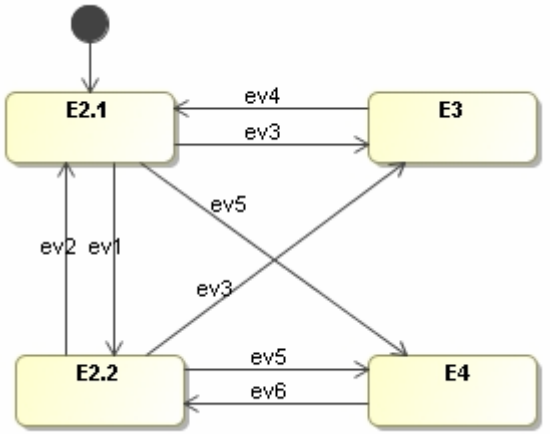




Máquinas de Estados Hierárquicas

- Planificação



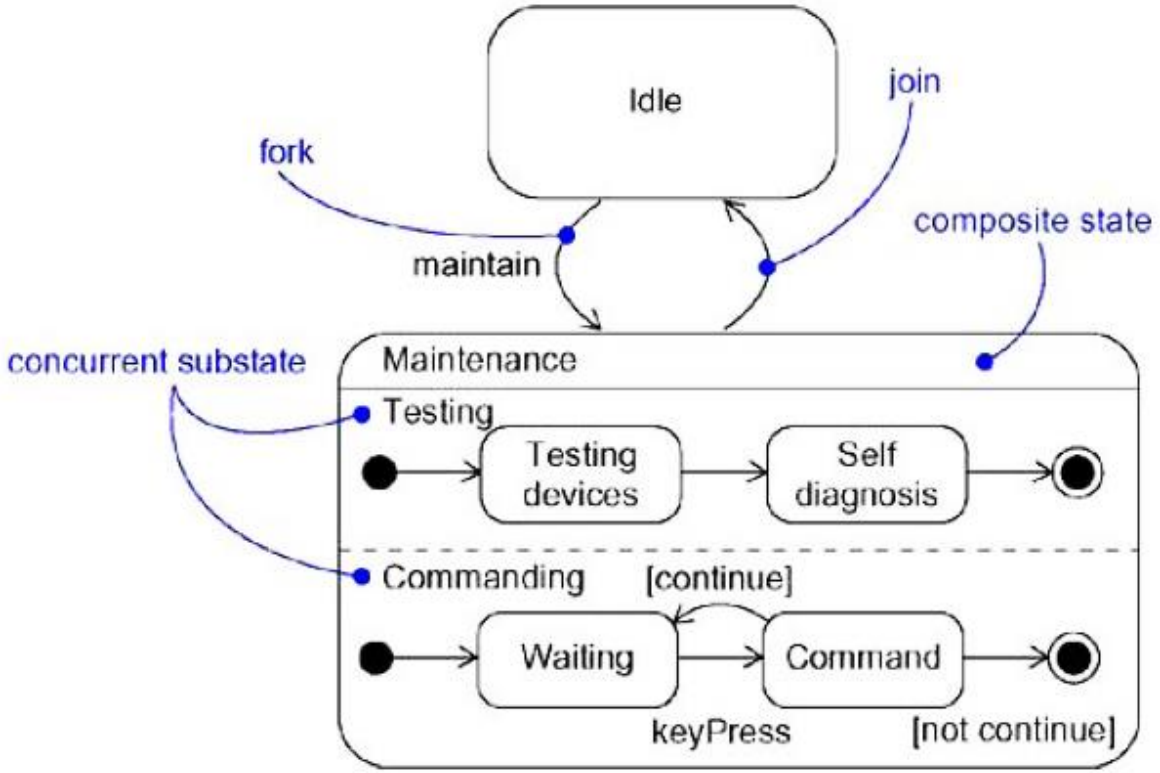


- Máquinas de Estados Hierárquicas (MEH) não são apenas uma forma de representação visual
- São um método de reutilização comportamental
  - **Factorização comportamental**
  - Ao **evitar repetições** permite o **crescimento da complexidade** do sistema a descrever **sem que isso provoque um aumento exponencial da complexidade do modelo**

Regiões Ortogonais

- Decomposição hierárquica
  - Decomposição disjuntiva (**OR**)
    - Exemplo: O sistema está no estado E2.1 **ou** no estado E2.2
  - Decomposição conjuntiva (**AND**)
    - Duas ou mais regiões ortogonais (independentes)
    - Um sistema está em todos os estados ortogonais em simultâneo
  - Permite lidar com o problema do aumento combinatório de estados em sistemas compostos por partes independentes que operam em concorrência

Regiões Ortogonais (Sub-estados concorrentes)



[UML User Guide, Booch *et al.* 1998]

Regiões Ortogonais

- As regiões (sub-máquinas) podem não ser totalmente ortogonais
  - Interacção (coordenação de comportamento através da troca de eventos)
  - Sincronização
- UML
  - Não requer:
    - Execução independente (*thread*) para cada região ortogonal (apesar de poder assim ser implementado)
  - Requer:
    - Não assumir uma ordem específica de processamento de eventos entre regiões ortogonais

Sequência de Transição de Estado

- MEH com regiões ortogonais
  - Sistema pode estar em diferentes estados
    - Todos os estados da **hierarquia**
    - Estados das **regiões ortogonais**
  - Árvore de estados
    - **Estado global** do sistema
    - **Configuração de estado**

- Executar
  - Acções de saída da **configuração de estado** origem (*exit*)
  - Acções associadas à transição
  - Acções de entrada da **configuração de estado** destino (*entry*)

Acções de Entrada e de Saída

- Sequência de activação
  - A activação de acções de entrada (***entry***) deve acontecer do estado mais **exterior para o mais interior**
  - A activação de acções de saída (***exit***) deve acontecer na ordem inversa, do estado mais **interior para o mais exterior**

Modelo de Execução

- Na prática as acções não são instantâneas
- Duas situações de operação de uma ME
  - Inactiva (*IDLE*)
  - Activa (BUSY)
- O que acontece se ocorrer um evento enquanto está a decorrer o processamento de outro evento?
- Gestão de eventos
  - *Preemptiva*
    - Potenciais problemas de concorrência
  - Não *preemptiva*
    - *"Run to completion"*

Processamento de Eventos

- **Tipos adicionais (UML)**
  - **Sinal**
    - Representa a recepção de um sinal assíncrono
    - **<nome-sinal> (<lista-parâmetros>)**
  - **Evento temporal**
    - Representa o expirar de um limite temporal
    - **AFTER <duração>**
  - **Evento condicional**
    - Representa a satisfação de uma condição booleana específica
    - **WHEN <condição>**
  - **Evento de evocação**
    - Representa a evocação síncrona de uma operação
    - **<nome-operação> (<lista-parâmetros>)**

Caso Prático

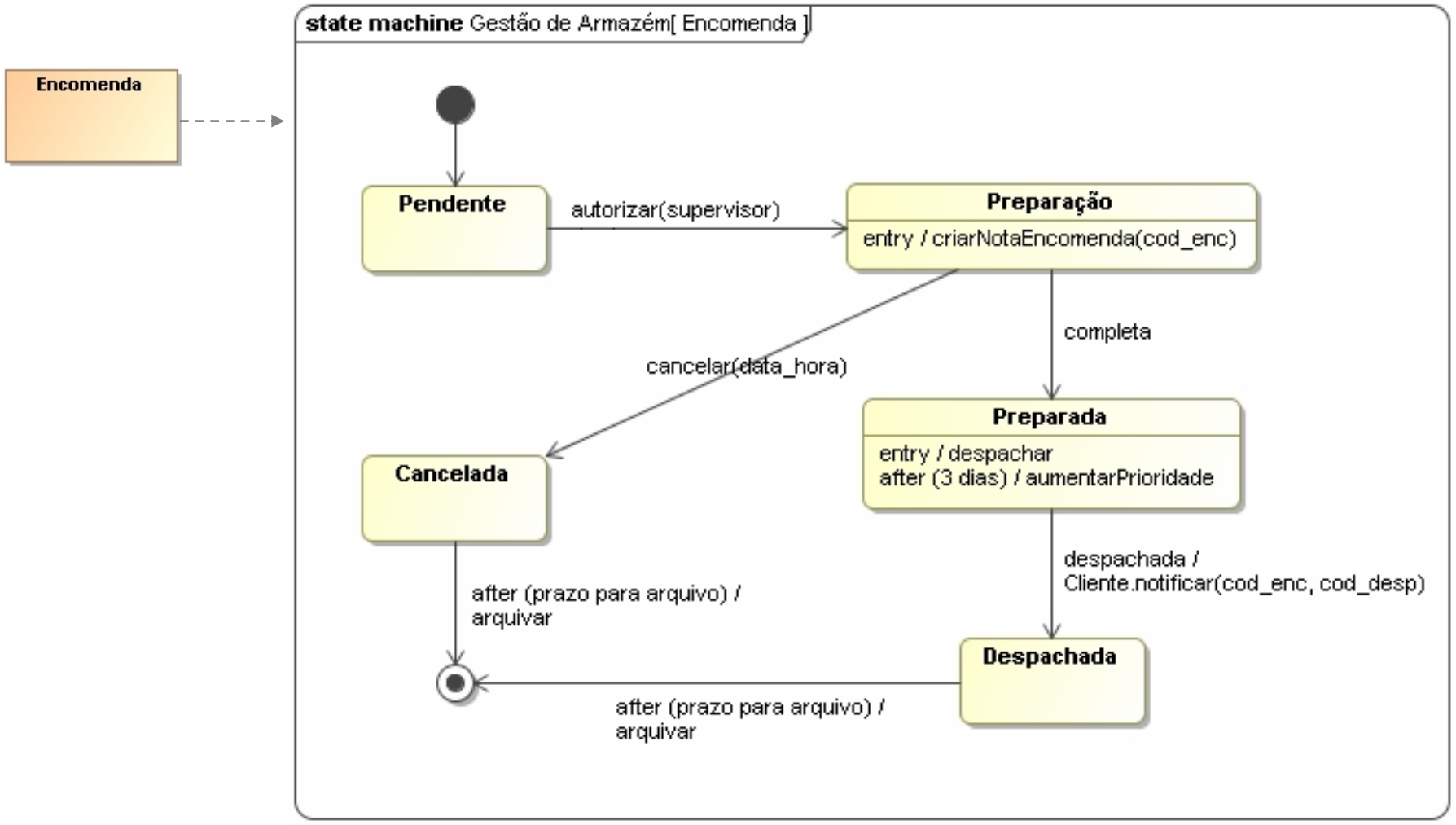
A empresa XYZ, proprietária de uma rede de lojas alimentares, pretende implementar um sistema que permita gerir a circulação de produtos nas suas várias lojas [...]

Os clientes podem fazer encomendas. Quando uma encomenda de um cliente é recebida fica pendente, até que um supervisor dê autorização para a sua realização. Deve ser mantido o registo do supervisor que autorizou cada encomenda. Após a autorização de um supervisor é preparada a encomenda. No início da preparação é sempre criada uma nota de encomenda com o código da encomenda.

Após a encomenda estar completa esta fica preparada para ser despachada para o cliente. O despacho pode no entanto não acontecer de imediato, pelo que, após 3 dias de espera por despacho, a prioridade da encomenda deve ser aumentada. Após o despacho o cliente deve ser notificado.

Uma encomenda pode ser cancelada enquanto está em preparação. Após o despacho ou o cancelamento, o registo da encomenda mantém-se durante um prazo predefinido. Após esse prazo a encomenda é arquivada e o seu registo eliminado do armazém.

Caso Prático



Bibliografia

[Booch et al., 1998]
G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley, 1998.

[Eriksson et al., 2004]
H. Eriksson, M. Penker, B. Lyons, D. Fado, *UML 2 Toolkit*, Wiley, 2004.

[OMG, 2020]
Unified Modeling Language (Specification), OMG, 2020.