

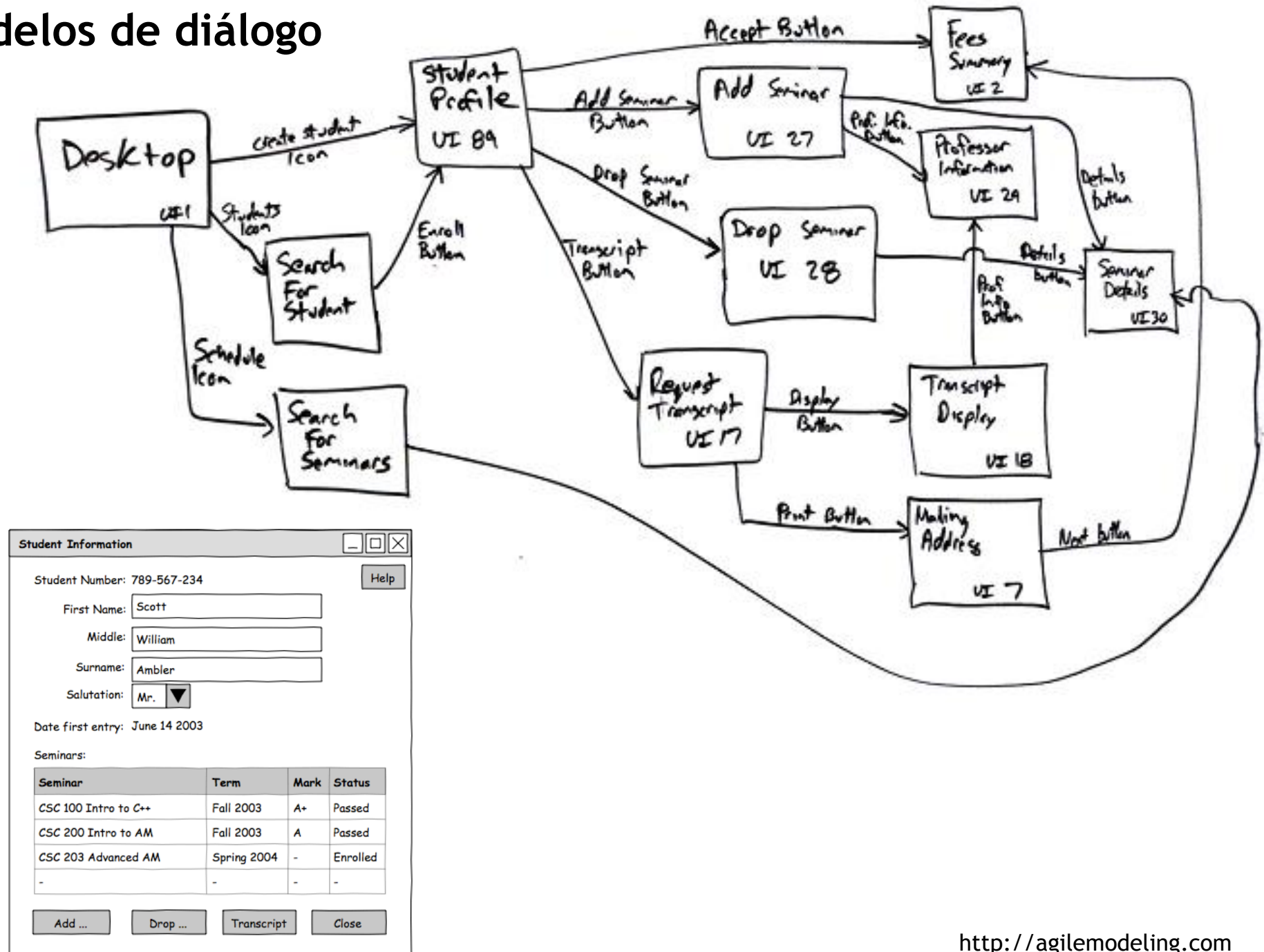


Desenvolvimento de Sistemas Software

Aula Teórica 9: Modelação de comportamento / Máquinas de Estado

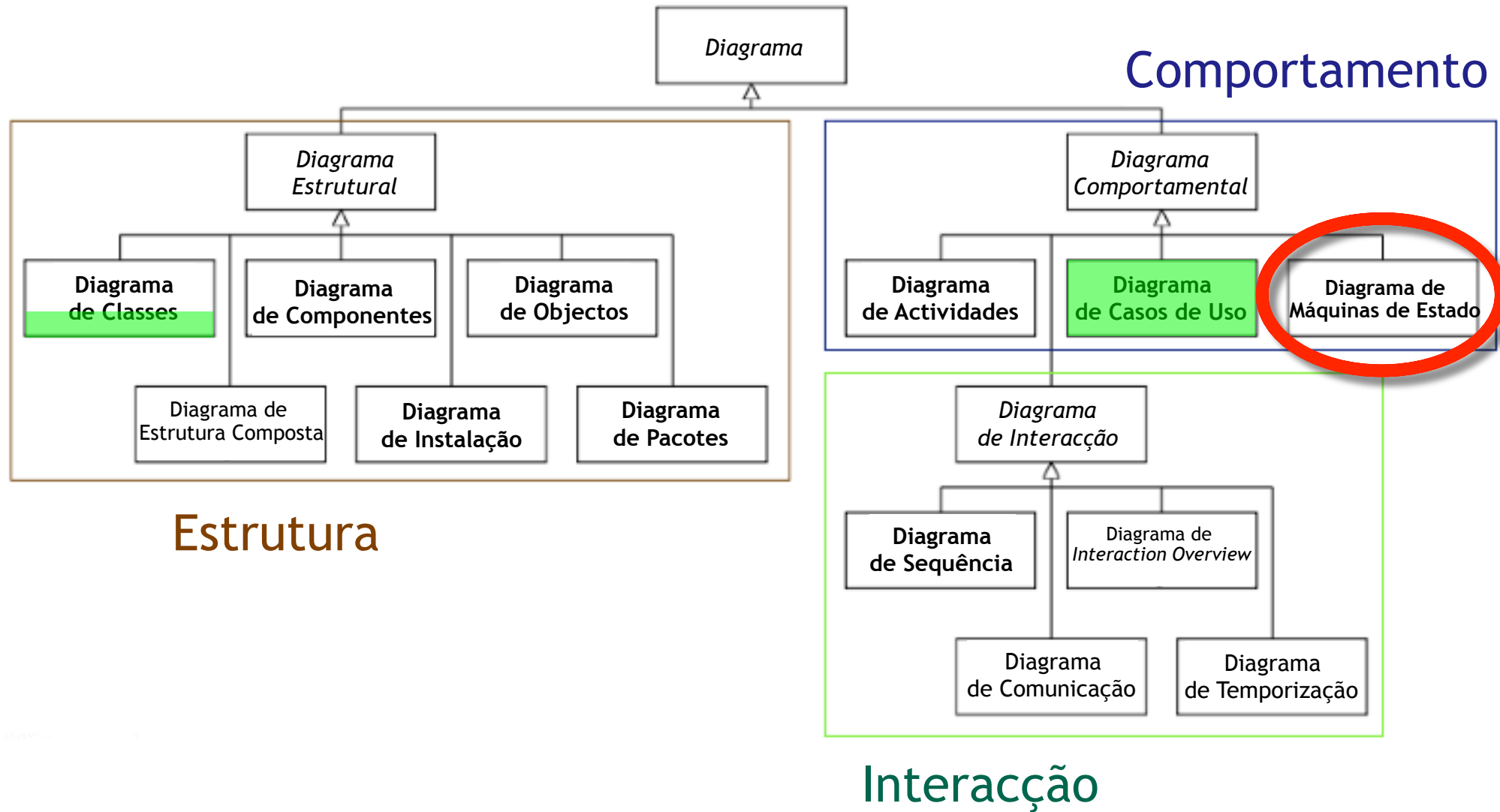


Modelos de diálogo





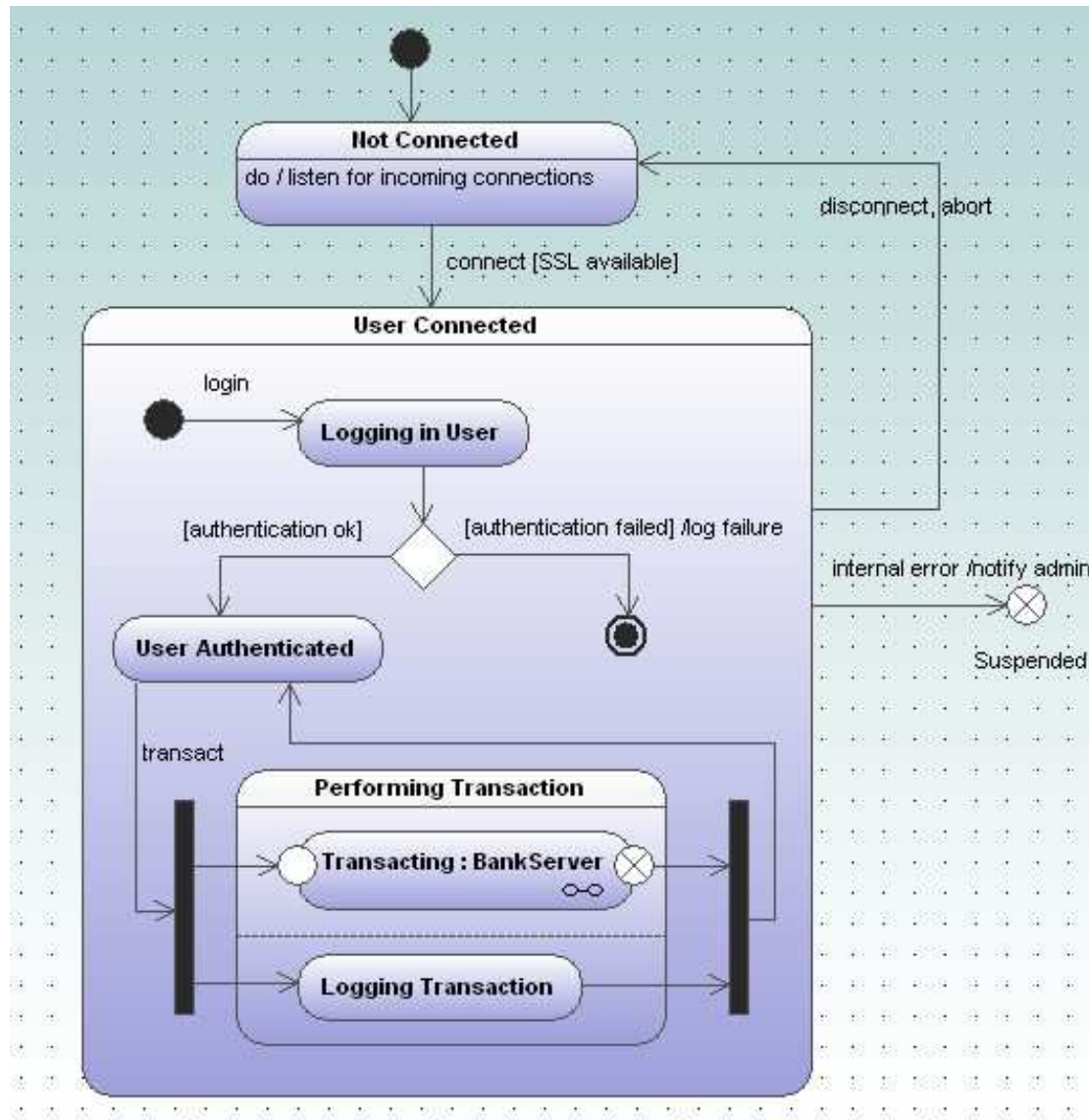
Diagramas da UML 2.x



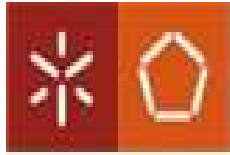


Introdução aos Diagramas de Estado – Aplicação

- Os Diagramas de Estado permitem modelar o comportamento de um dado objecto/sistema de forma global.
- A ênfase é colocada no estado do objecto/sistema – modelam-se todos os estados possíveis que o objecto/sistema atravessa em resposta aos eventos que podem ocorrer.
- Úteis para modelar o comportamento de um objecto de forma transversal aos *use case* do sistema.
- Devem utilizar-se para classes em que se torne necessário compreender o comportamento do objecto de forma global ao sistema.
- Nem todas as classes vão necessitar de diagramas de estado.
- Úteis para modelar o comportamento do sistema (em particular a interface).



Especificação dos passos do estabelecimento de uma ligação segura (SSL) a um servidor bancário para realizar transacções



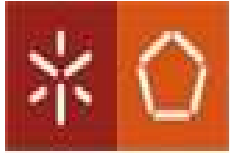
DIAGRAMAS DE MÁQUINAS DE (TRANSIÇÃO DE) ESTADOS

▣ Para que se compreenda a verdadeira importância dos DME em UML, é, em rigor, necessário compreender a sua **génese** e **objectivos**, em especial apresentando as **teorias/modelos** nos quais se baseiam.

▣ **Génese:** São adaptações UML de notações há muito existentes para a descrição do comportamento de sistemas sob a forma de Autómatos (máquinas que funcionam sozinhas em resposta a entradas de um dado tipo), que em cada momento se encontram num estado interno que representa a sua memória do passado e o seu conhecimento do presente para que possa transitar correctamente para outro estado quando surge a **entrada ie. evento seguinte**;

▣ As adaptações visam permitir que UML possa especificar o comportamento de dispositivos físicos (**máquinas hardware**) e de entidades lógicas ou “**máquinas software**” (numa perspectiva OO, **certos objectos**);

▣ **Vamos definir alguns conceitos fundamentais sem os quais os DMEs não podem ser compreendidos nem correctamente usados.**

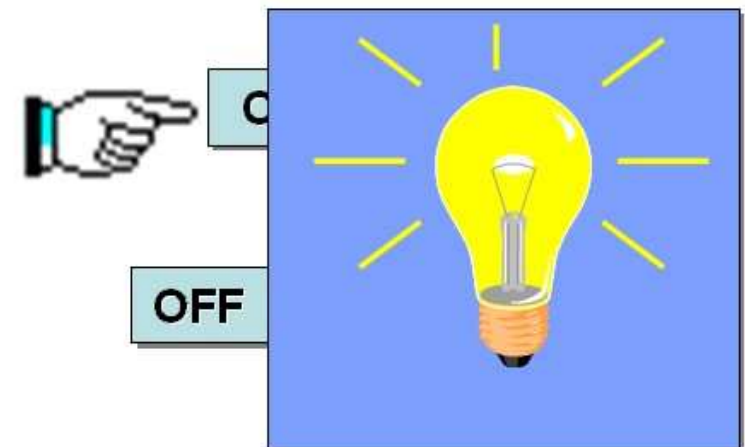
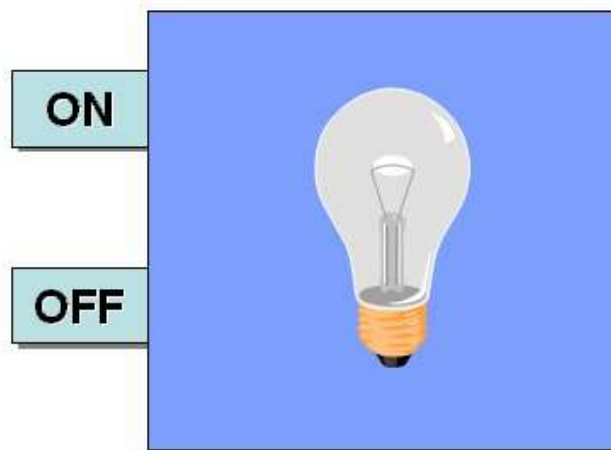


AUTÓMATOS (Máquinas de Estados Finitos)

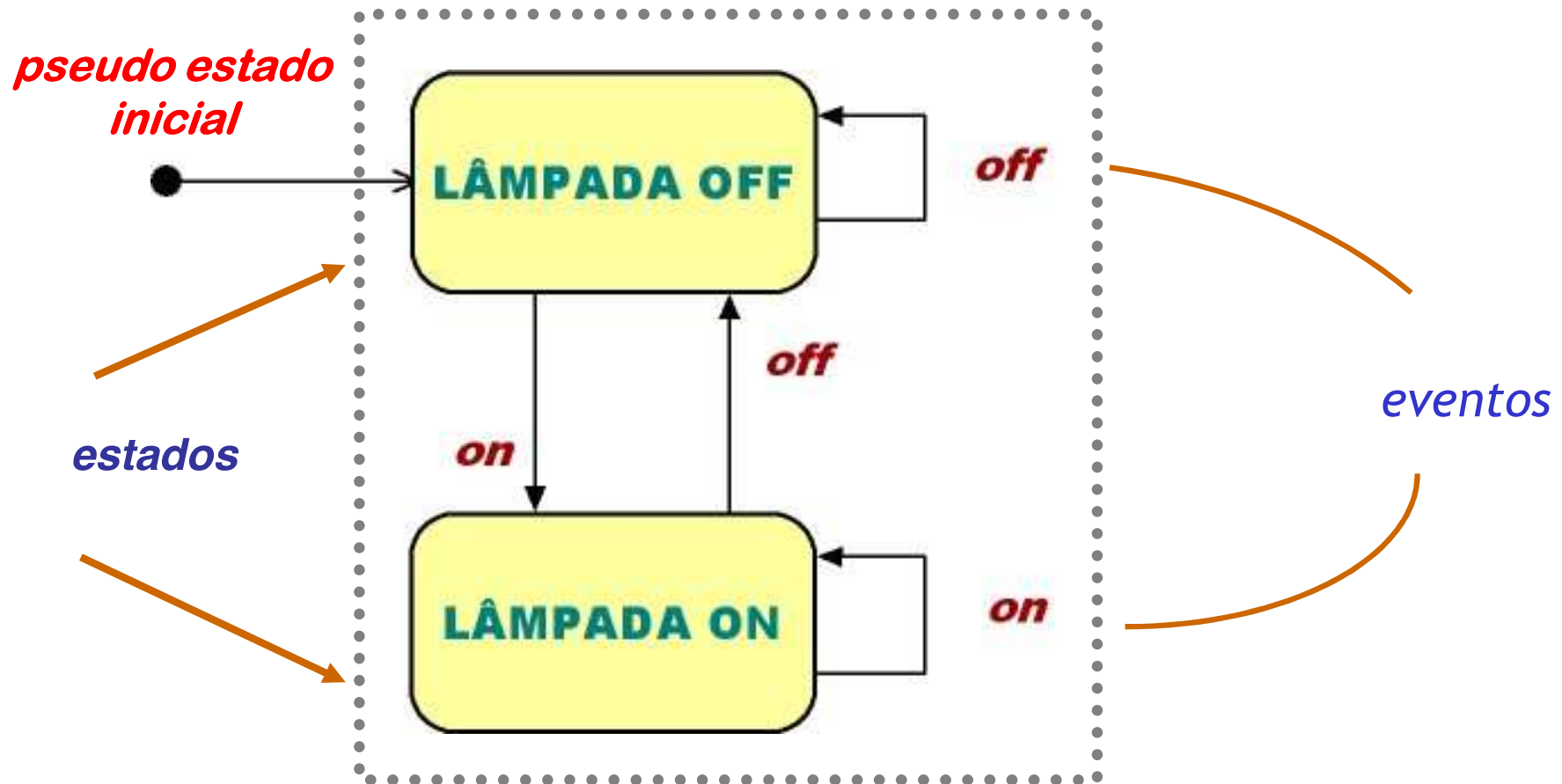
- ▶ São máquinas cujo comportamento é uma consequência não apenas da última entrada mas também de todo o passado de entradas (visto como a sequência de entradas realizadas, por exemplo máquina de café);
- ▶ Caracterizam-se por, a cada momento, se encontrarem num estado interno que “representa” toda a sua experiência passada (ou seja, o resultado da sequência de todas as entradas registadas). Comportamento consiste em transitar de estado em estado (estados são em número finito).
- ▶ Interactuamos com estes autómatos diariamente, porque diariamente usamos as máquinas de venda de qualquer coisa, cf. as máquinas de venda de bebidas, de chocolates, etc.; usamos também as ATM, usamos as bombas de gasolina e as estações de lavagem automática de carros.
- ▶ Há, porém, muitas “máquinas software” (classes) que produzem “objectos” que possuem características de comportamento, passivo ou activo, muito semelhantes a estas máquinas baseadas em electrónica e mecânica.

AUTÓMATOS

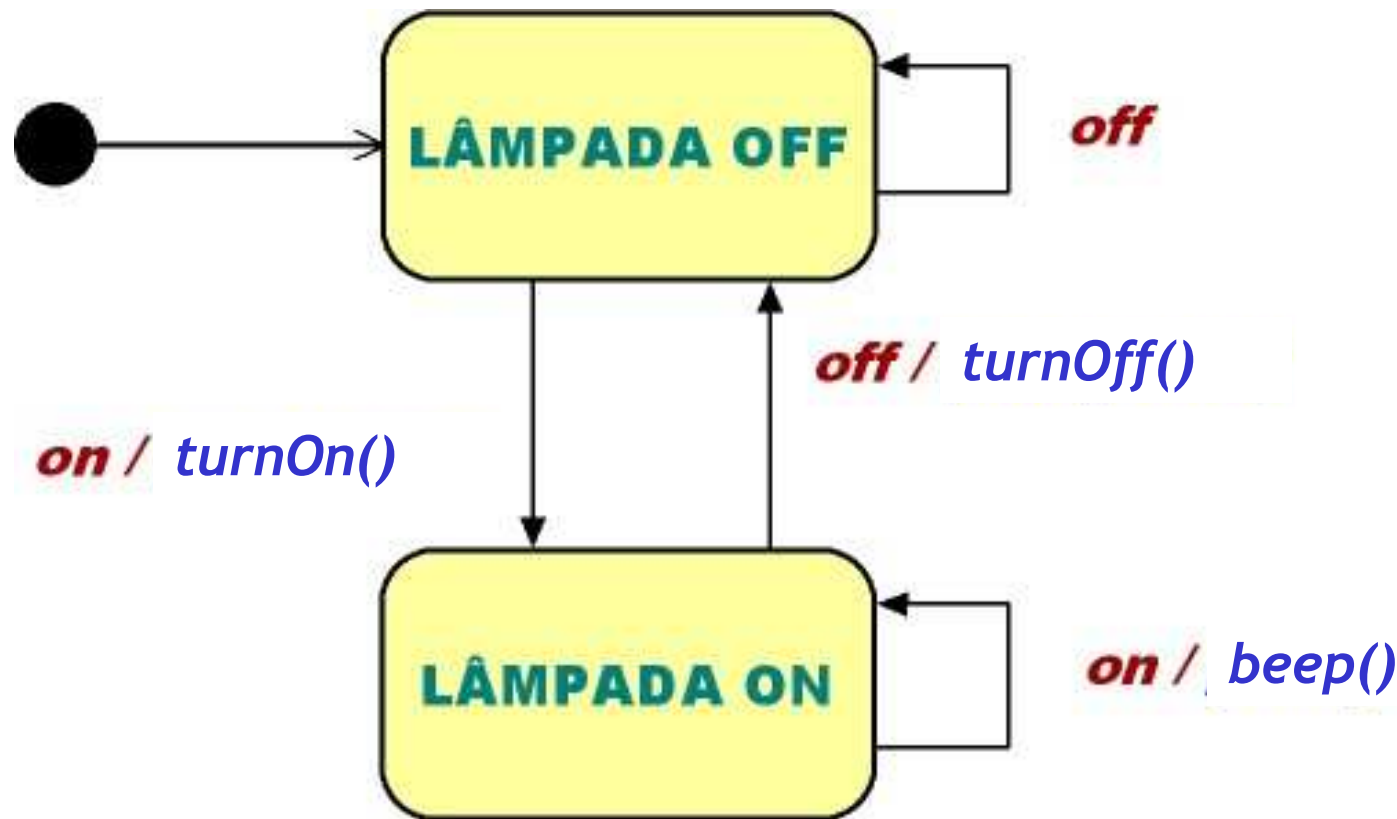
- ▶ São máquinas cujo **comportamento** é uma consequência não apenas da **última entrada** mas também de **todo o passado** (visto como a sequência de entradas realizadas).
- ▶ Caracterizam-se por possuírem um **estado interno** que “representa” toda a sua experiência passada (**comportamento** => **transição de estado**).



▣ Representação gráfica do **comportamento do autómato** em termos de estados, entradas e transições entre estados.



▣ Quando muda de estado, um autómato pode gerar “**outputs**” (resultados de acções) que se descrevem associados às **entradas-eventos**, tal como em **entrada / acção**





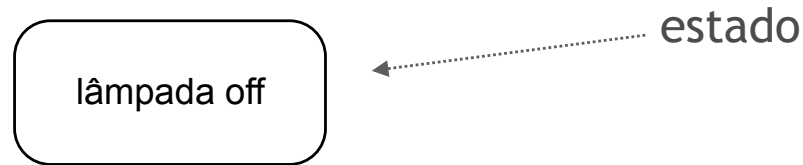
☐ Máquinas de Estados apenas podem modelar **comportamento discreto**, ou seja, **não contínuo e baseado na reacção a eventos**, **comportamento reactivo discreto**

☐ Ao contrário do estado de um objecto OO, que corresponde aos valores dos seus atributos num dado momento, os estados de um autómato são mais abstractos



Notação base

- Estado – define uma possível estado do objecto (normalmente traduz em valores específicos dos seus atributos)



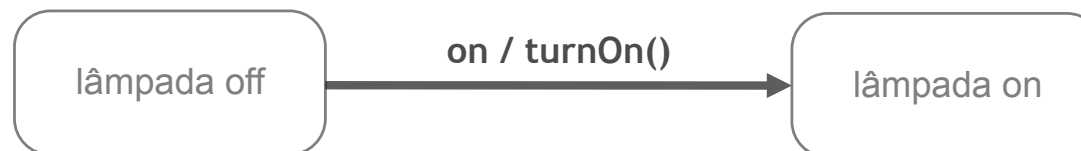
- Estado inicial – estado do objecto quando é criado



- Estado final – destruição do objecto

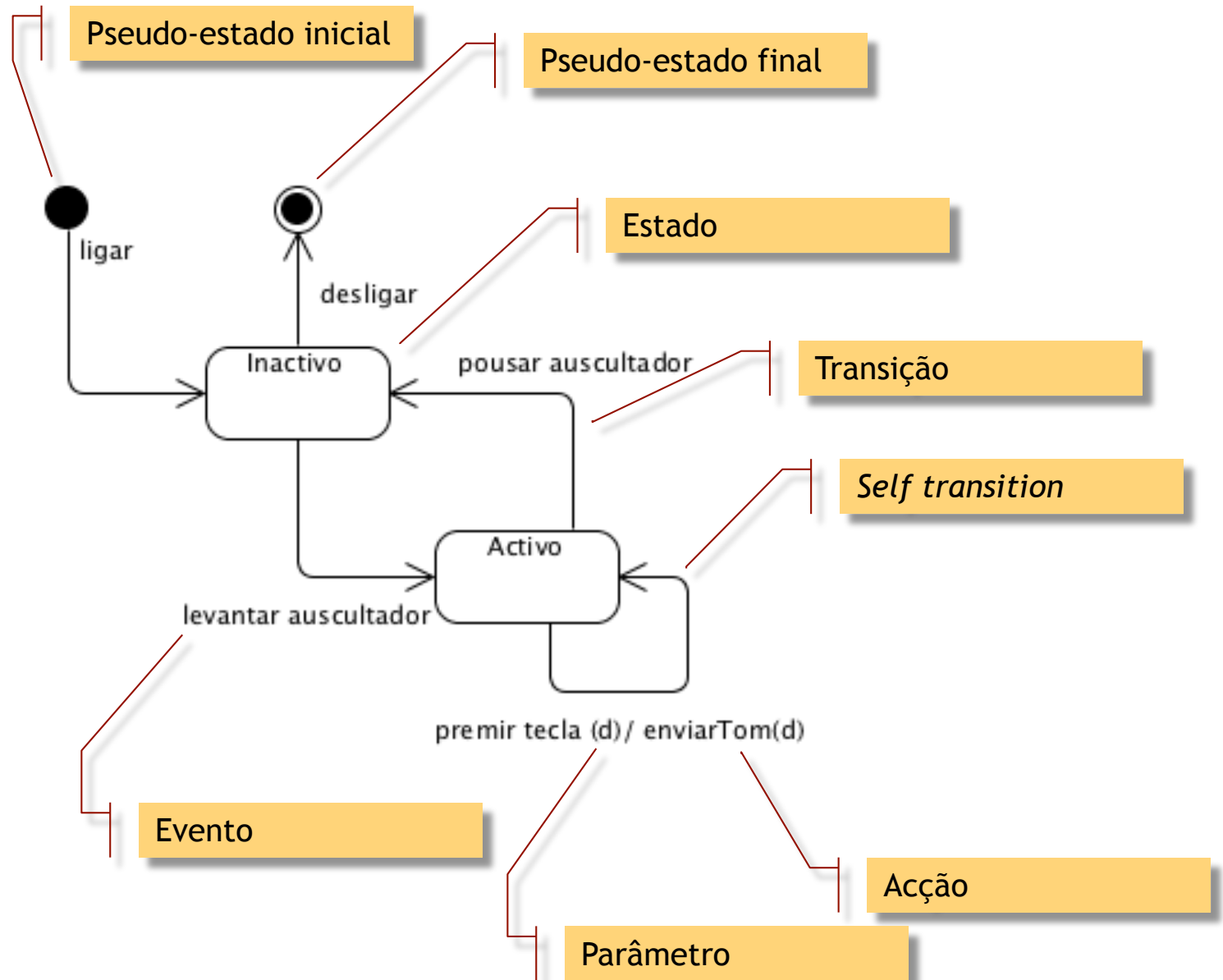


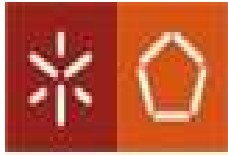
- Transições – evento[guarda]/acção (todos são opcionais!)





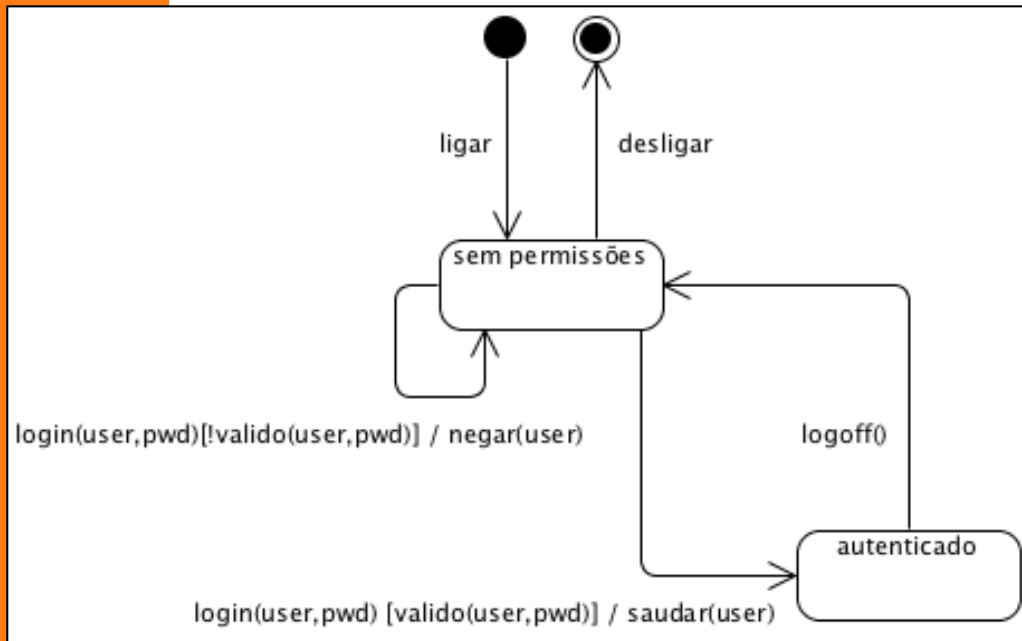
Maquina de Estados básica





▣ Um **estado** é uma **situação particular** dentro das várias situações possíveis no ciclo de vida de um autómato, durante a qual o autómato possui dadas propriedades (cf. **aberto**, **fechado**, **feliz**, **infeliz**), satisfaz alguma condição (cf. **transitável** ou **não**, **válido** ou **inválido**), realiza alguma actividade (cf. faz soar o alarme, ou faz pisca-pisca), ou então apenas espera a ocorrência de algum evento válido.

▣ Tal como foi dito antes, **os estados relevantes de um DME são aqueles que respondem de forma diferente aos eventos que podem ocorrer** (ver exemplo da lâmpada).



Transições representam passagens de um estado a outro pela ocorrência de um **evento**.

Associado ao **evento** de uma transição podemos ter **parâmetros** e **acções**.

Há transições que são **condicionadas**, ou seja, **guardadas** por uma condição entre **[...]**. Se **true** há transição senão ...

Quando uma máquina está num dado **estado** e ocorre um **evento**, **apenas uma transição de saída pode ser tomada**.

Eventos, transições e actividades são realizados de forma instantânea.

Forma geral:

evento(pars) [guarda] / acção

☐ Porém, os **estados** podem ter **actividades internas**, ou seja, quer quando são alcançados, ou atingidos, podem **executar de imediato as acções associadas ao evento interno entry**, quer quando são abandonados podem executar as acções associadas ao evento **exit**. Podem ainda **definir eventos que têm tratamento dentro do mesmo estado**, ou seja, que não provocam transições de estado mas apenas acções realizadas internamente e apenas com efeito interno (designadas **“self-transitions”**).

entry/acção

- “acção” é automaticamente executada quando o objecto entra no estado;

evento/acção

- “acção” é automaticamente executada se “evento” ocorrer (transição interna);

do/acção

- “acção” é continuamente executada enquanto o objecto estiver no estado;

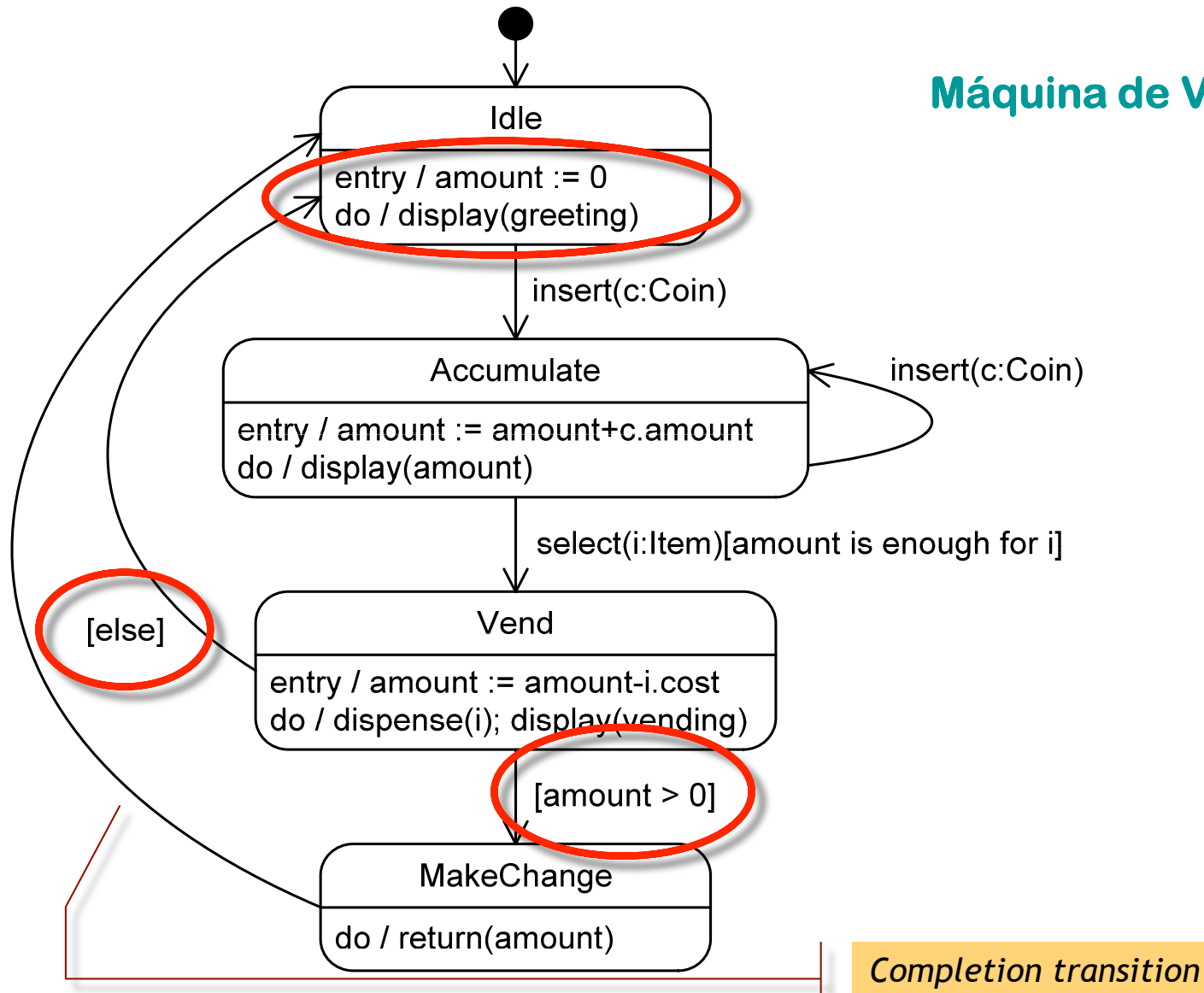
evento/defer

- “evento” é deferido até o estado actual ser abandonado;

exit/acção

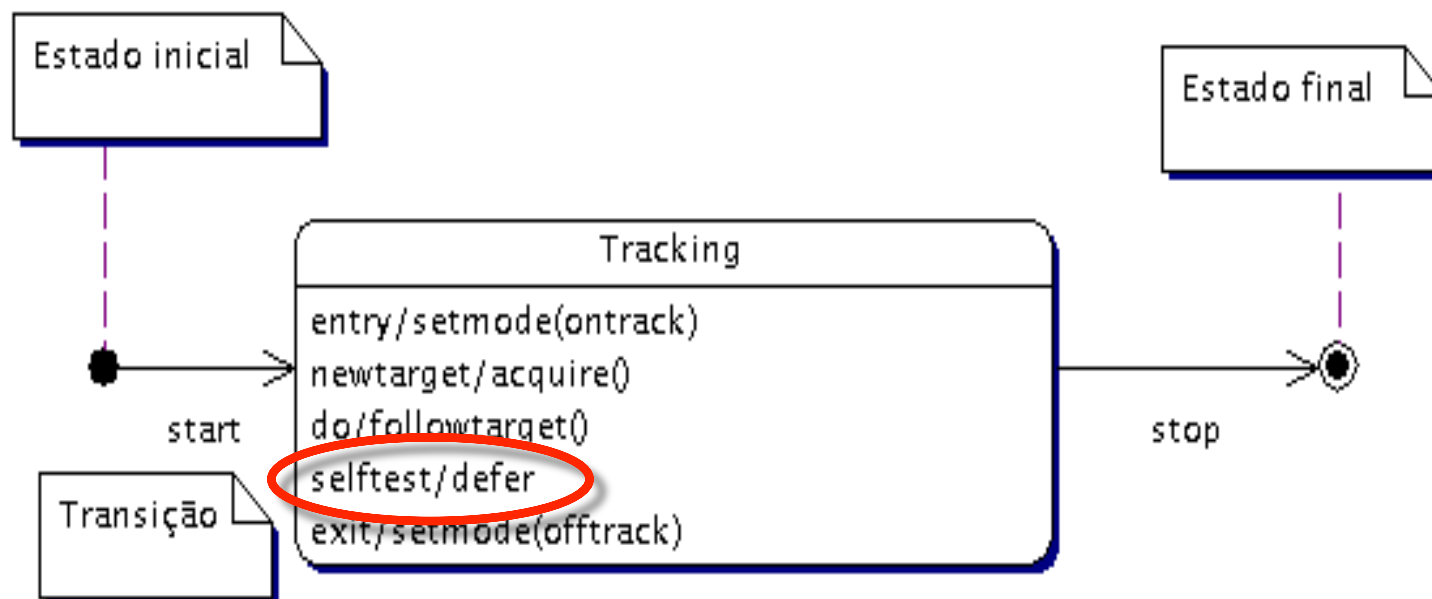
- “acção” é automaticamente executada quando o objecto sai do estado.

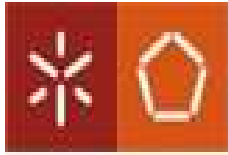
Máquina de Venda II





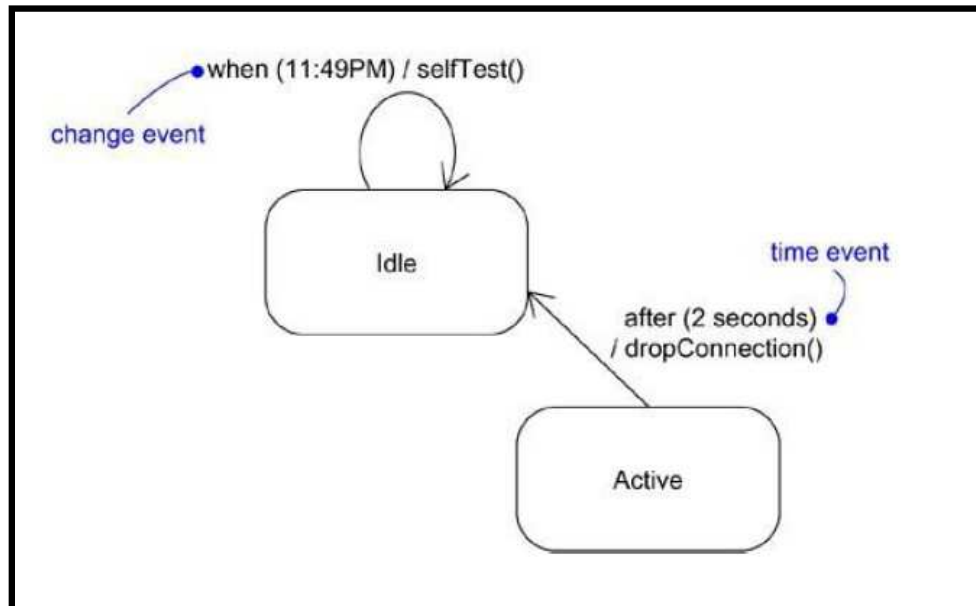
Actividades internas



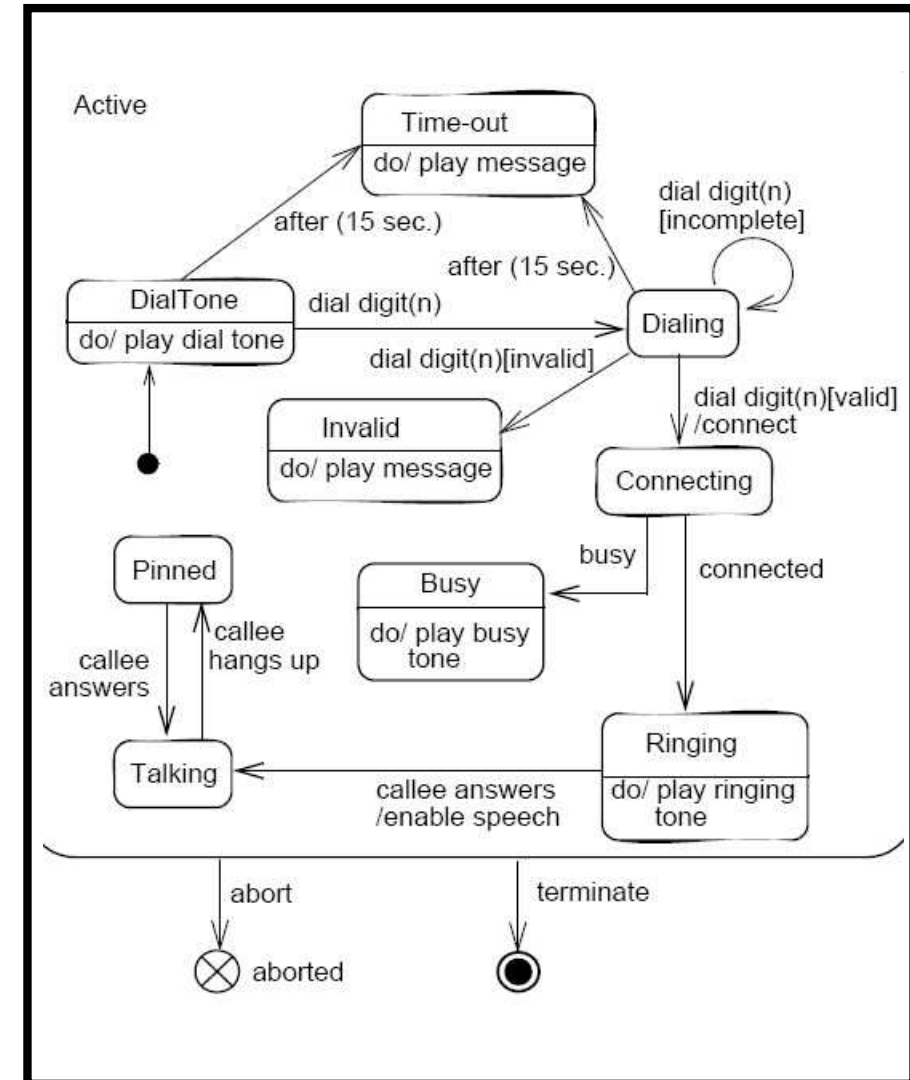


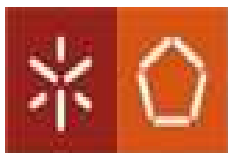
OUTROS EVENTOS

Tipo de Evento	Descrição	Sintaxe
<i>de tempo relativo ou de tempo absoluto</i>	Tempo relativo à entrada no estado origem ou tempo absoluto	<i>after(time)</i> after(10s) after(21:00)
<i>de tempo absoluto</i>	Chegada a um dado tempo absoluto; Verificação de condição	<i>when(time)</i> <i>when(expr_bool)</i> when(1:00 AM) when(qt >= 50 ct)

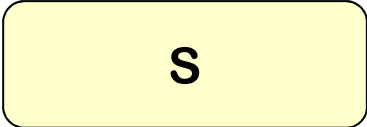
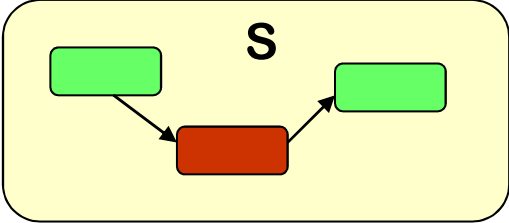
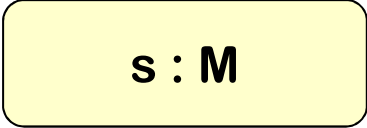


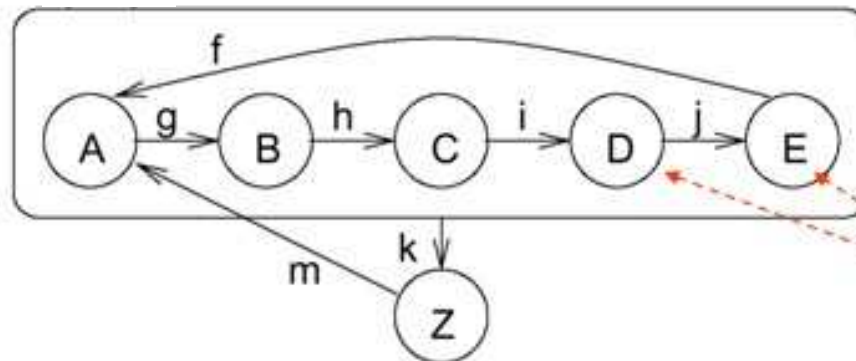
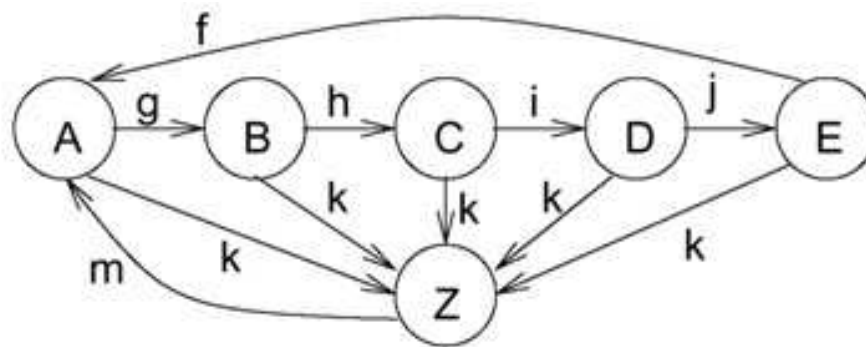
Exemplos de utilização dos eventos especiais **when()** e **after()**





OUTROS ESTADOS I

Tipo de Estado	Descrição	Notação
<i>simples</i>	Estado sem subestrutura	
<i>composto sequencial</i>	Estado composto formado por subestados, dos quais apenas um está activo quando o estado composto está activo	
<i>estado de submáquina</i>	Estado que referencia uma máquina de estados	



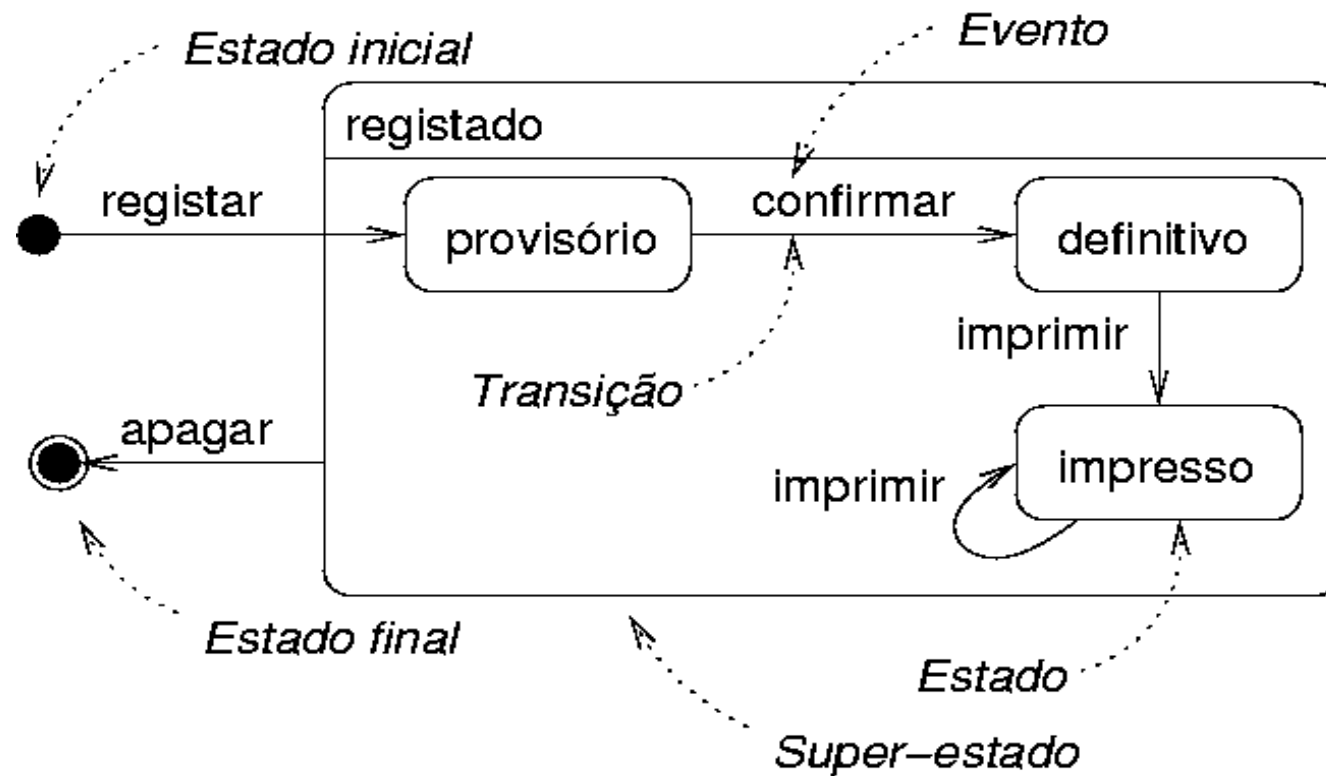
Estados compostos são muito úteis na estruturação dos diagramas de estado

super-estado S

sub-estados

Estados e Estados Compostos (super-estados)

- Super-estado – permitem estruturar os modelos





Diagramas de Estado (*Statecharts*)

Sumário

- Introdução aos Diagramas de Máquinas de Estado – Aplicação
- Noções base: estados e transições
- Notação base
 - Estados
 - Pseudo-estado inicial
 - Pseudo-estado final
 - Transições
- Actividades internas
- Eventos temporizados: when/after
- Estados simples vs estados Compostos