

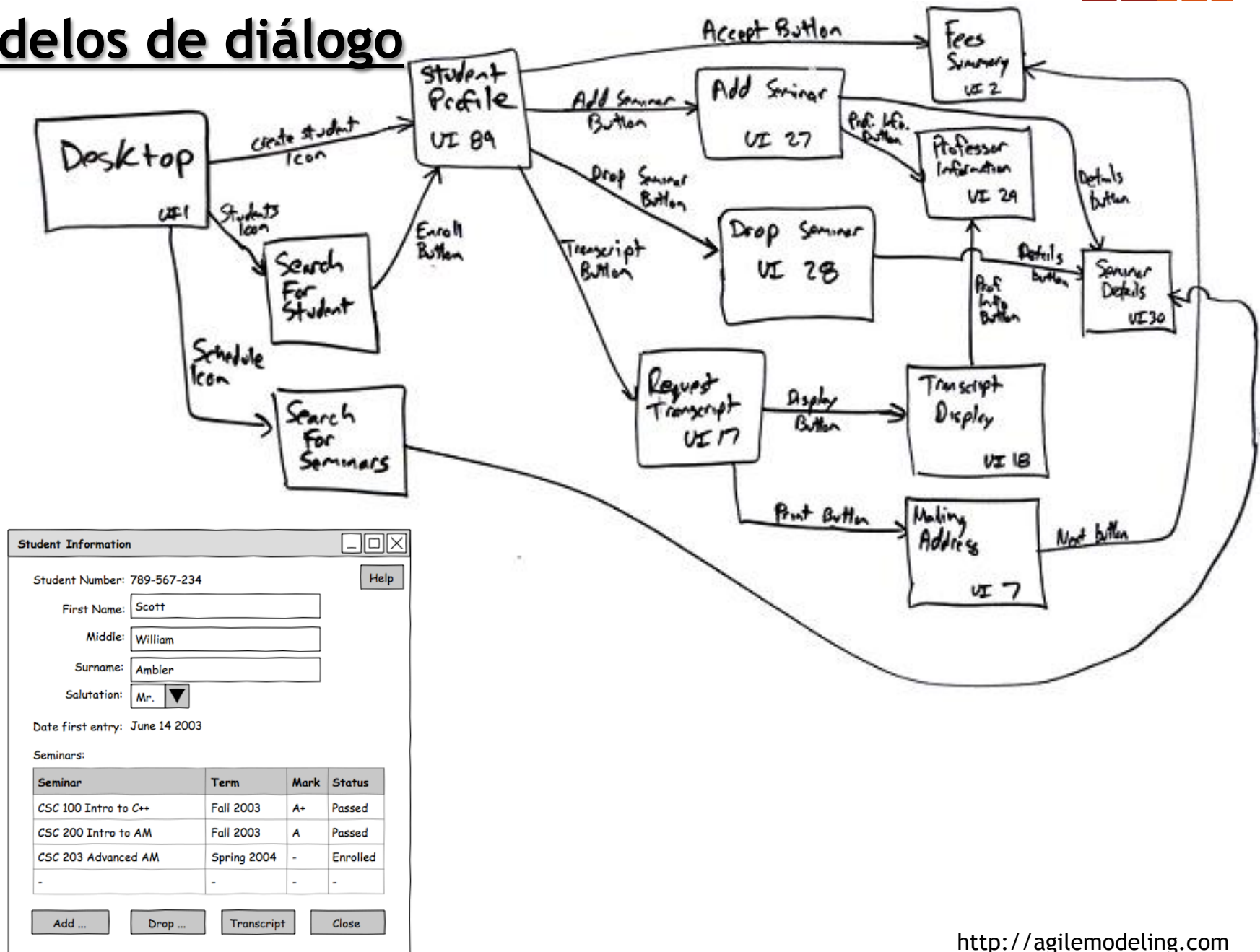


# Desenvolvimento de Sistemas Software

## Aula Teórica 11: Modelação de comportamento / Máquinas de Estado



# Modelos de diálogo



**Student Information**

Student Number: 789-567-234 Help

First Name:

Middle:

Surname:

Salutation:  ▼

Date first entry: June 14 2003

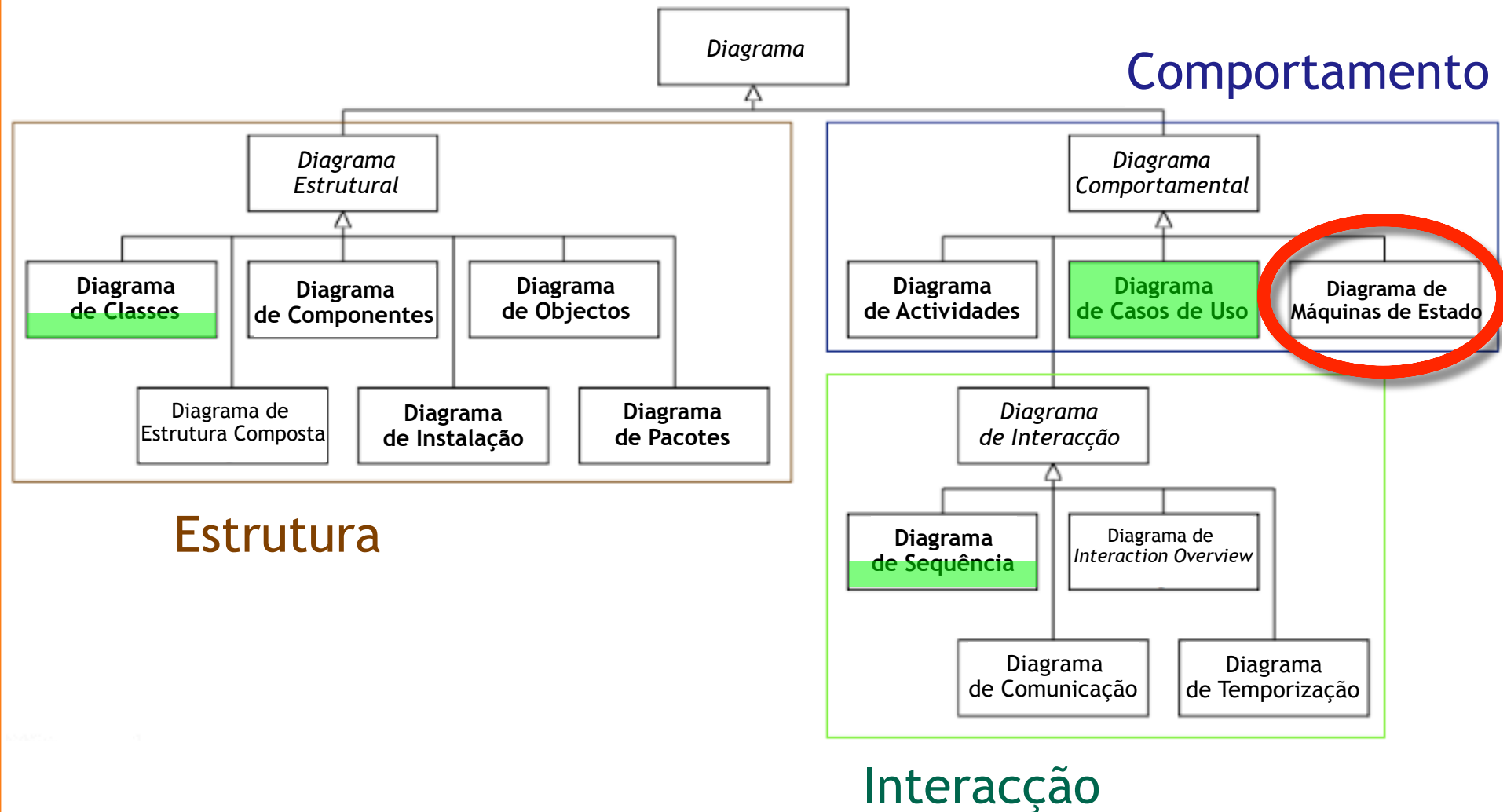
Seminars:

Seminar	Term	Mark	Status
CSC 100 Intro to C++	Fall 2003	A+	Passed
CSC 200 Intro to AM	Fall 2003	A	Passed
CSC 203 Advanced AM	Spring 2004	-	Enrolled
-	-	-	-

Add ... Drop ... Transcript Close



# Diagramas da UML 2.x





## Introdução aos Diagramas de Estado – Aplicação

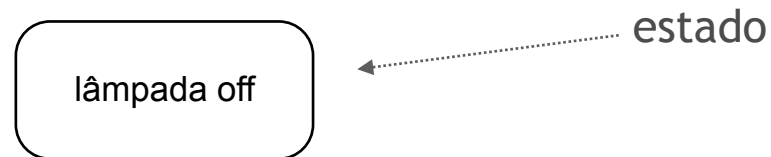
- Os Diagramas de Estado permitem modelar o comportamento de um dado objecto/sistema de forma global.
- A ênfase é colocada no estado do objecto/sistema – modelam-se todos os estados possíveis que o objecto/sistema atravessa em resposta aos eventos que podem ocorrer.
- Úteis para modelar o comportamento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



# Diagramas de Estado

## Notação base

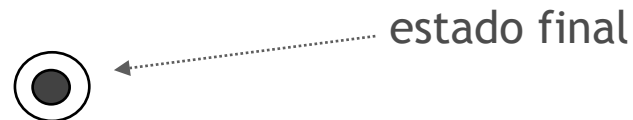
- Estado — define uma possível estado do objecto (normalmente traduz em valores específicos dos seus atributos)



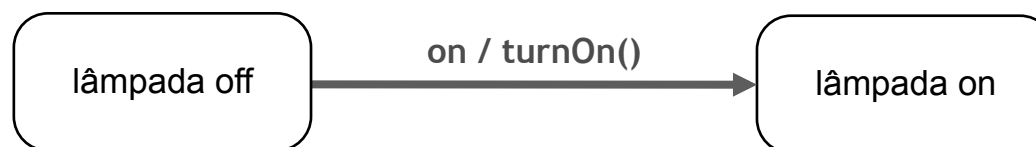
- Estado inicial — estado do objecto quando é criado



- Estado final — destruição do objecto

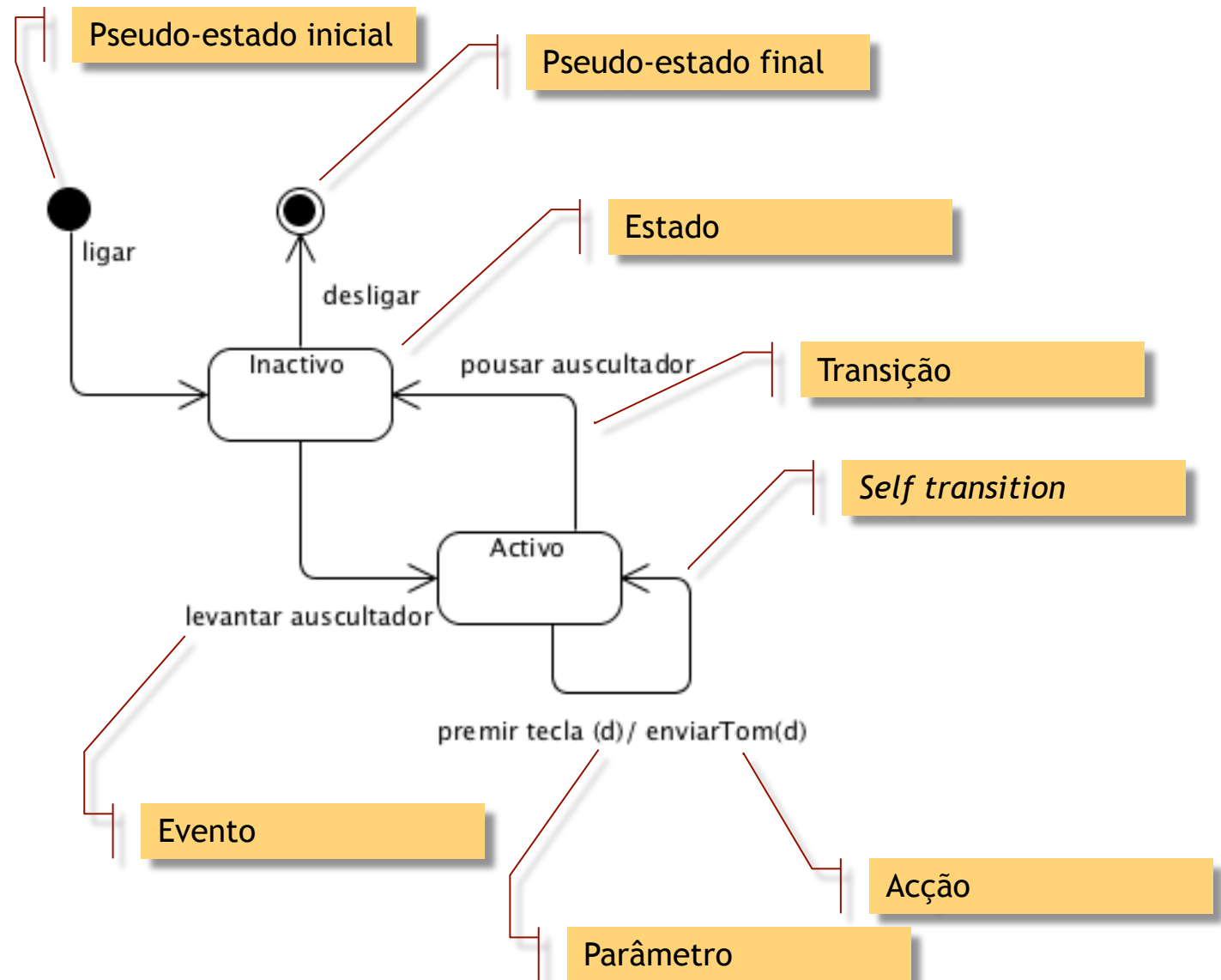


- Transições — evento[guarda]/acção (todos são opcionais!)



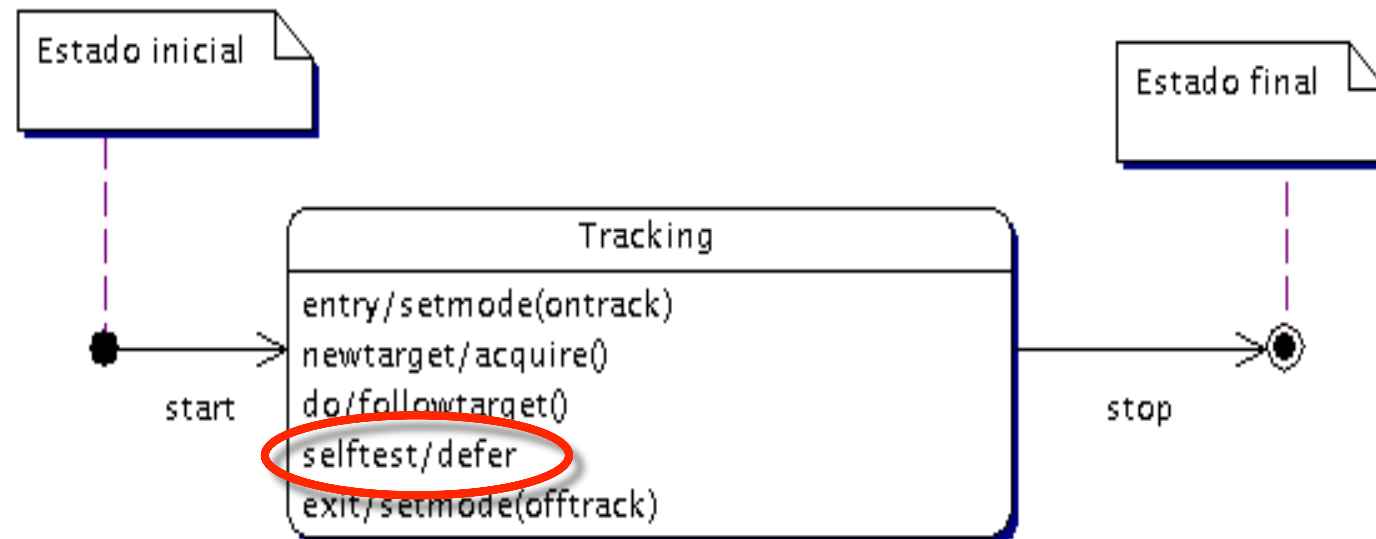


# Maquina de Estados básica





## Actividades internas





## Actividades internas

- Actividades que não provocam transições de estado...

*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*

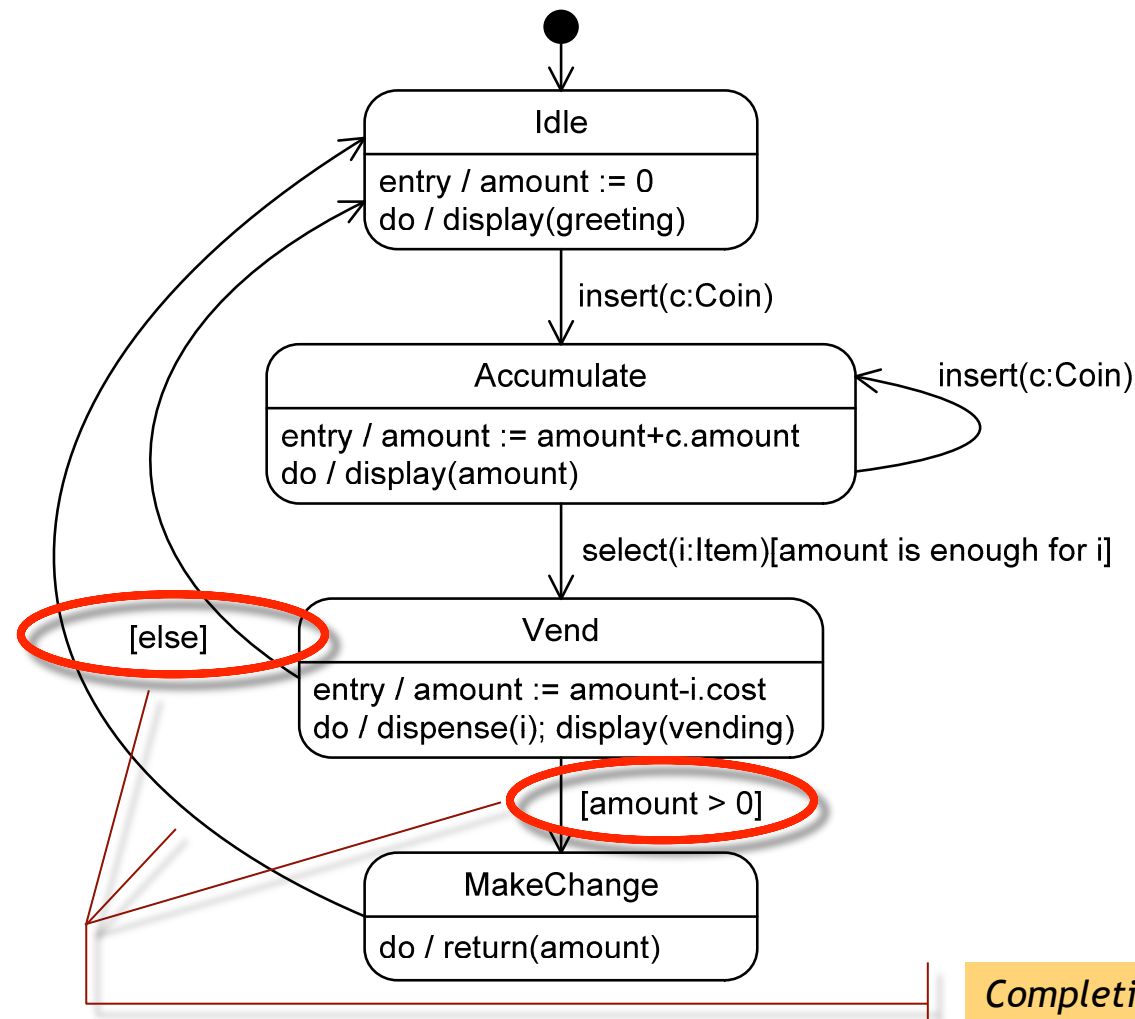
- “*acção*” é automaticamente executada quando o objecto sai do estado.





## Completion transitions

- Acontecem de forma automática logo que possíveis.

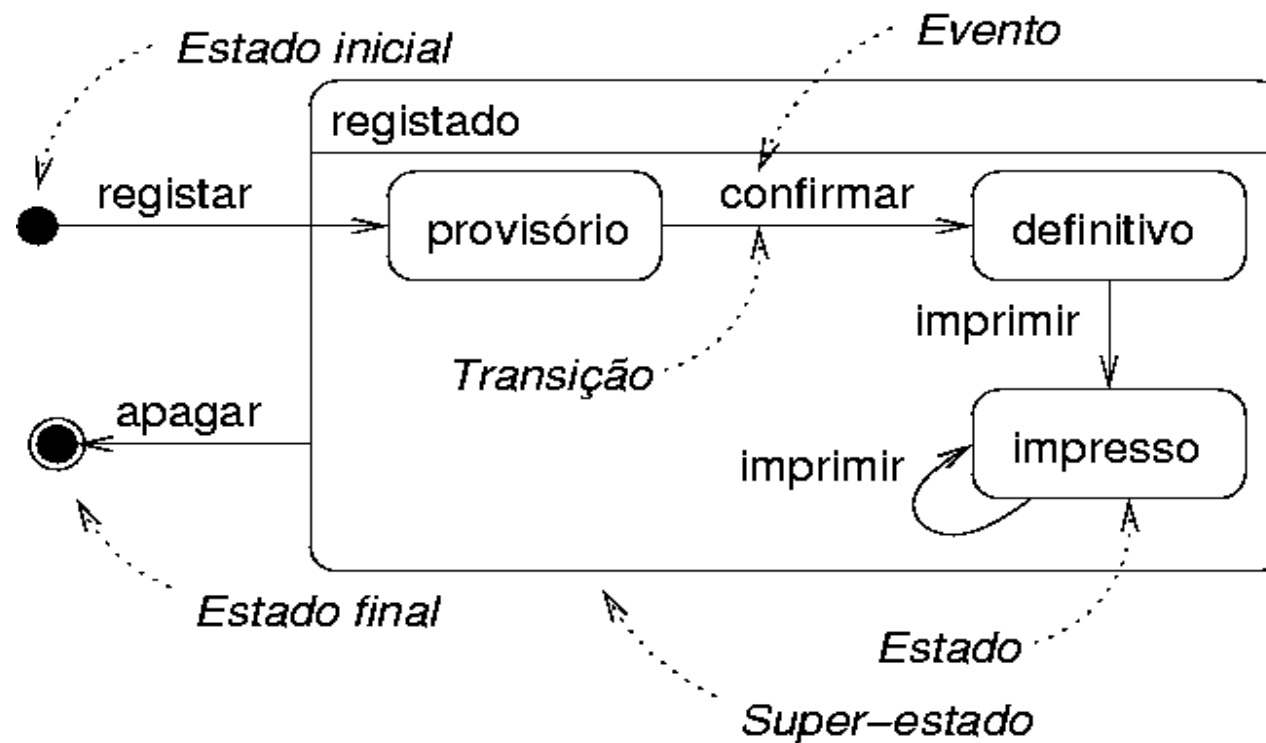


Completion transitions



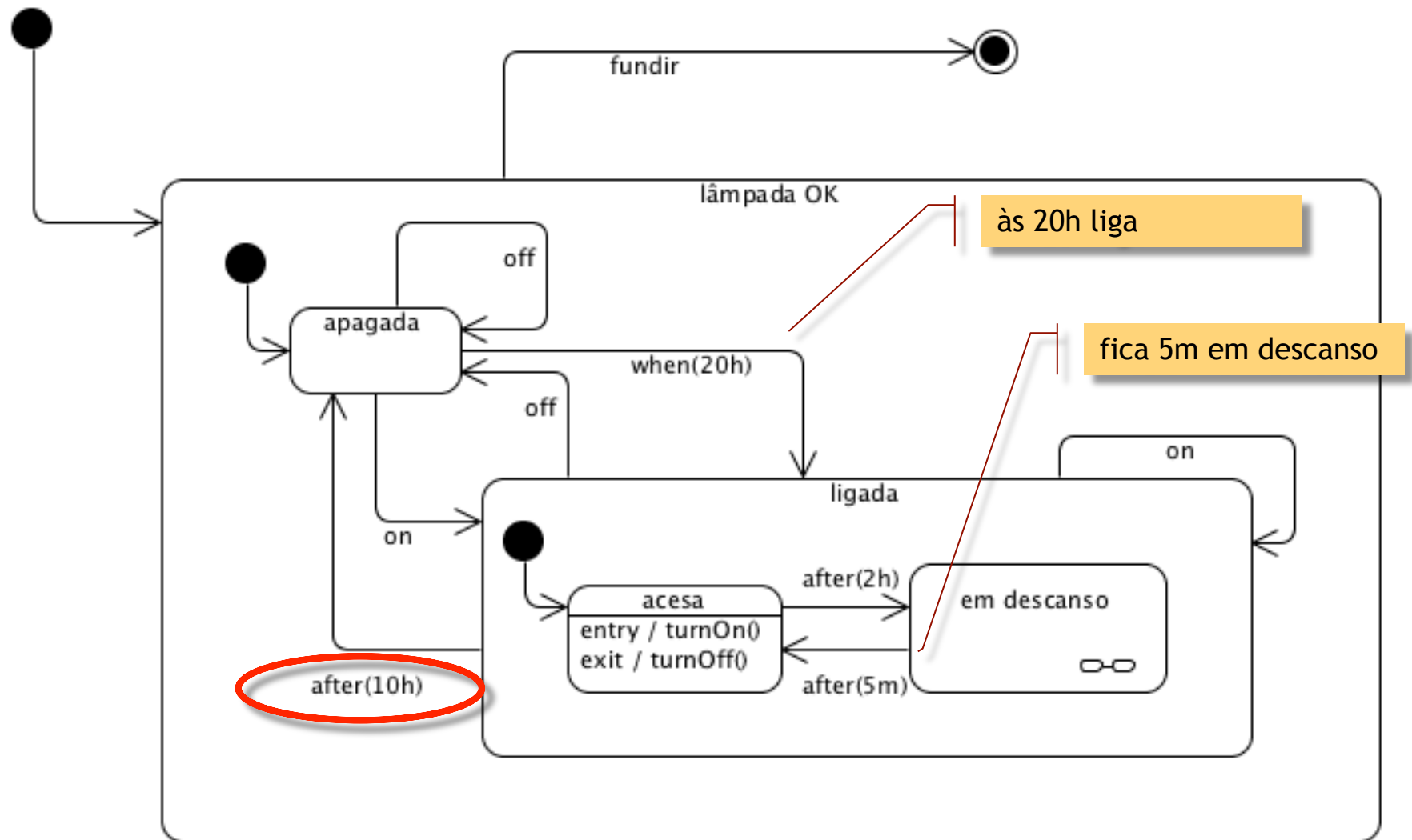
## Estados e Estados Compostos (super-estados)

- Super-estado – permitem estruturar os modelos



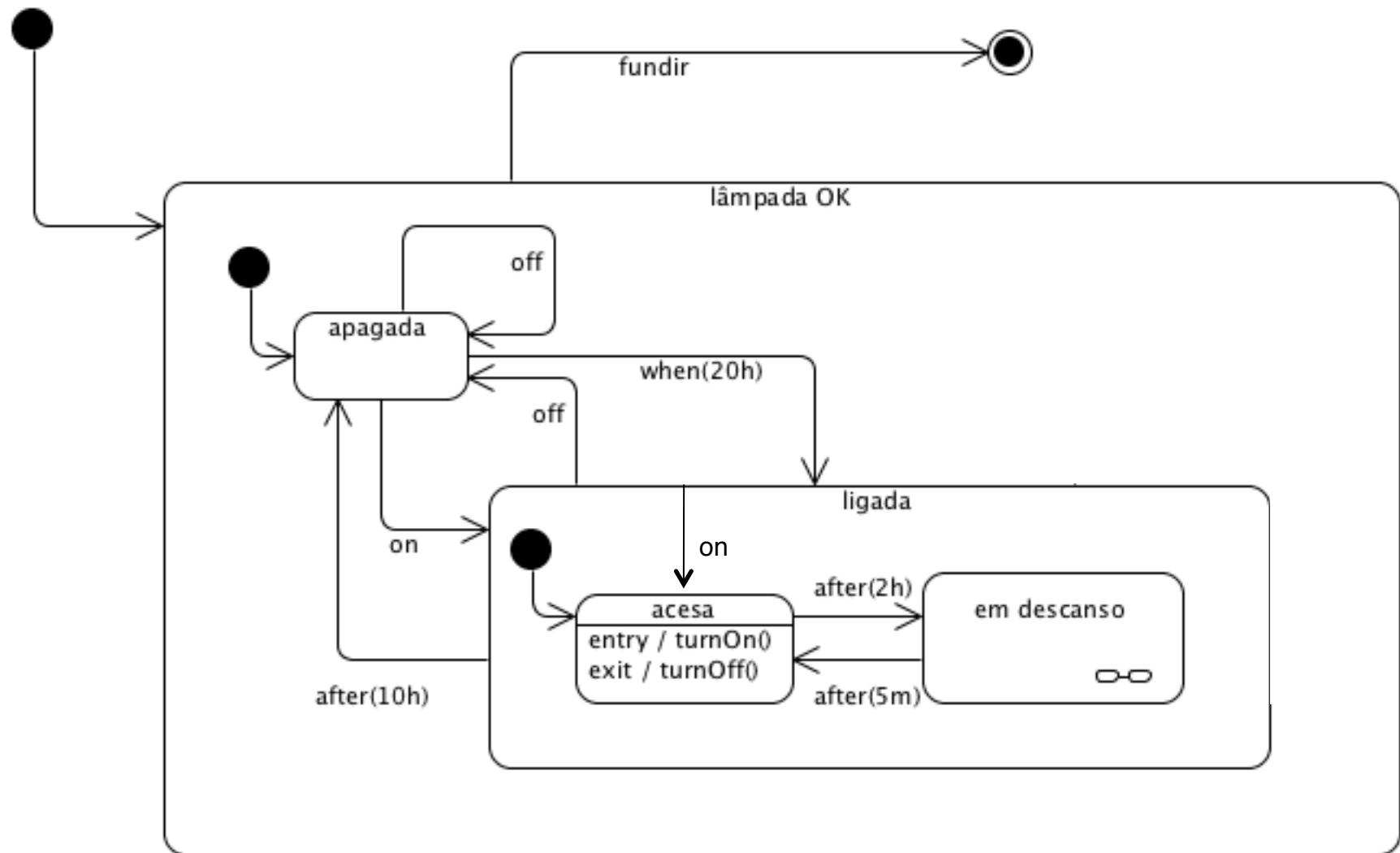


## Eventos *when/after*





## Eventos *when/after*





# 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
  - eventos *when/after*
  - Transições
- Actividades internas
- Estados simples vs estados Compostos