SWEN 223

Software Engineering Analysis  
UML State Diagrams

Thomas Kuhne

Victoria University of Wellington  
[Thomas.Kuehne@ecs.vuw.ac.nz](mailto:Thomas.Kuehne@ecs.vuw.ac.nz), Ext. 5443, Room Cotton 233



State Diagrams

Applicability

* can be used to specify the states exhibited by an object or the system
* determine the responses (state transitions) to outside stimuli (events)
* are concerned with when operations execute, rather than what operations do, or how they are implemented

Applications

Reactive Behaviour

* Events

» make object/system transition between states » are enabled/disabled, depending on state » yield different actions/transition, depending on state

Areas

* Broad Range of Applications

» computer programs, business processes, protocols, web page navigation, ...

—i

**JIMS**

14/04/2016 \ f y

Foundation

Finite State Machines

* States and Events

» limited expressiveness

» can “recognise” regular languages

» can also be represented as state transition tables

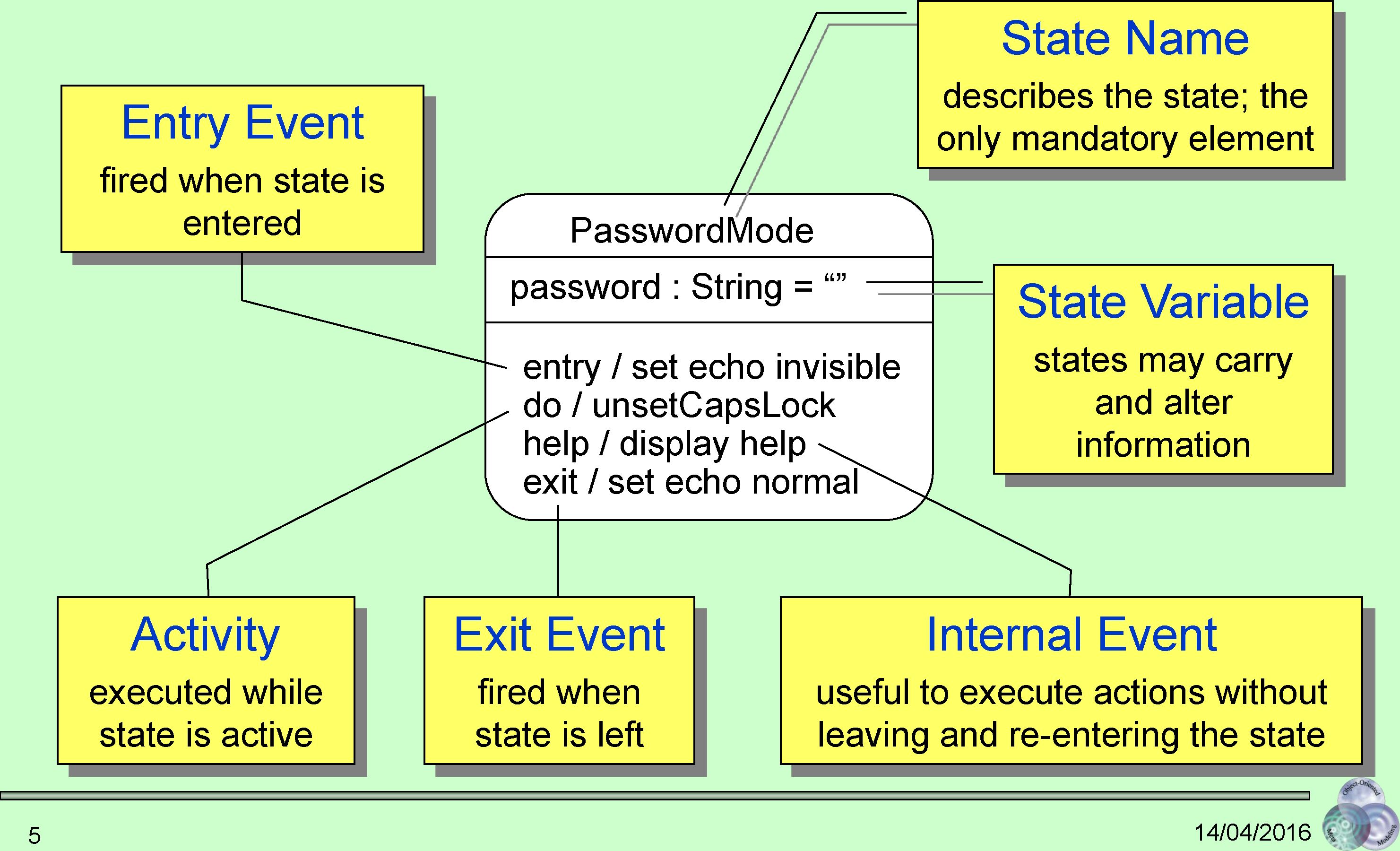
Harel Statecharts

* Multiple States

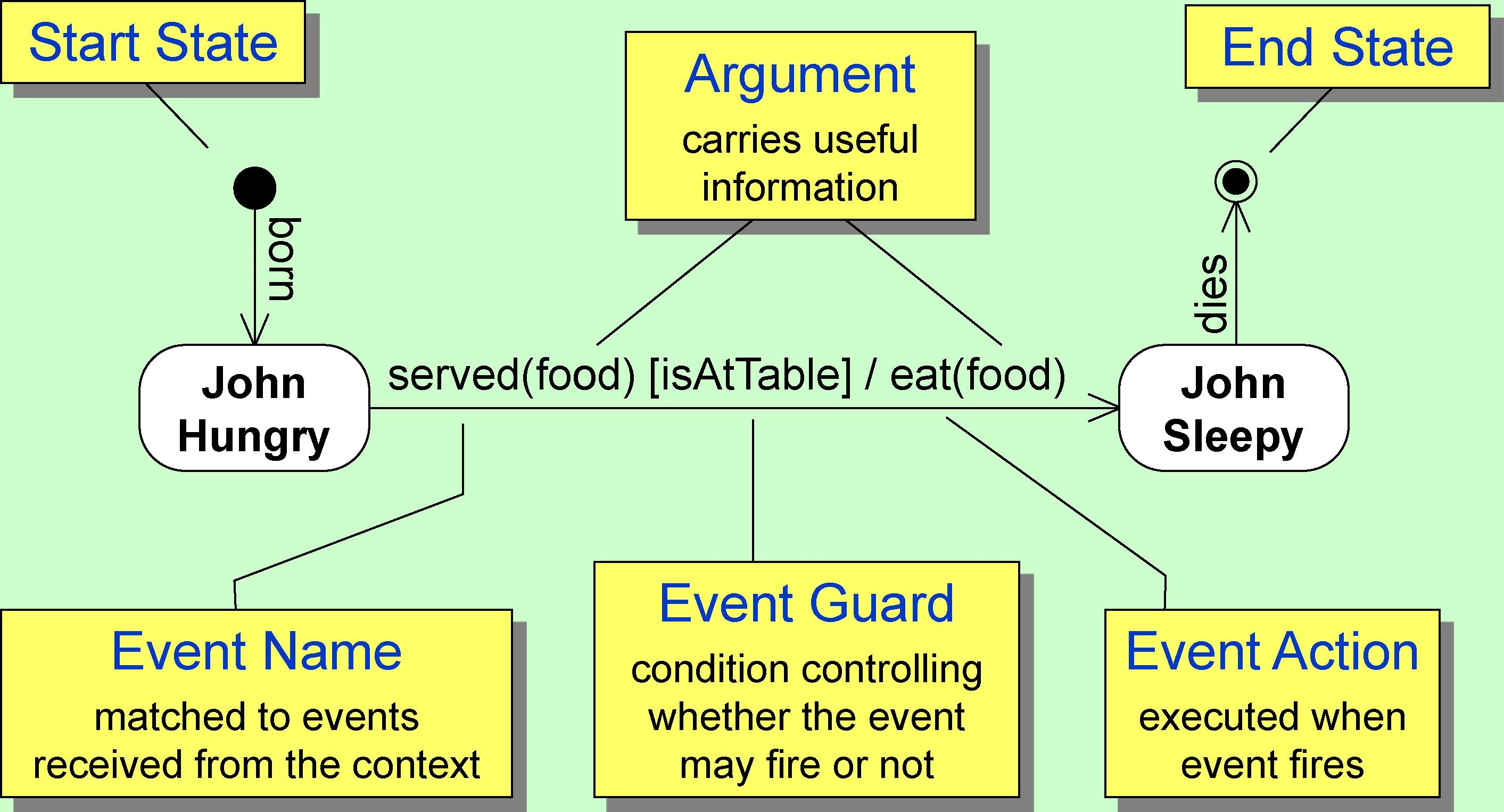
» extremely useful to structure state diagrams

—i

States



Transitions



Events

(Some) Kinds of Events

* SignalEvent

» represents the receipt of an asynchronous event and is queued by the receiver until it's ready to handle it

* CallEvent

» models the synchronous receipt of a message by an object, invoking a call of an operation

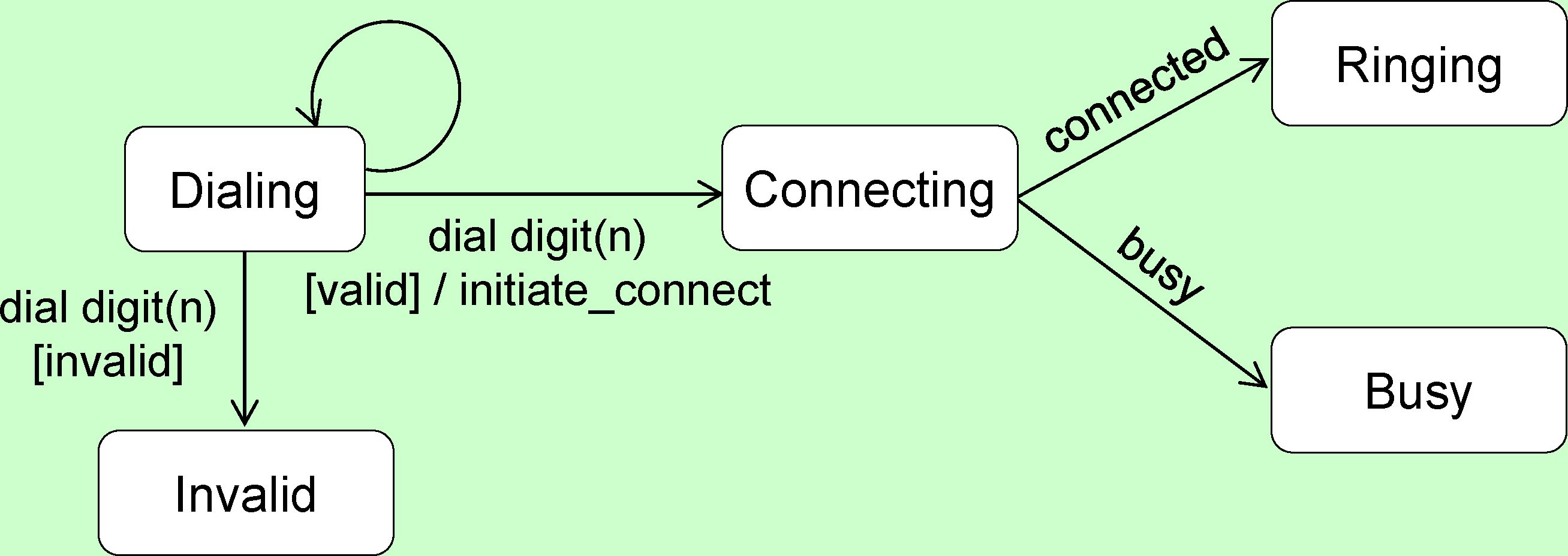
* TimeEvent

» after the specified time, the event occurs. The keyword after is often used in conjunction with time events

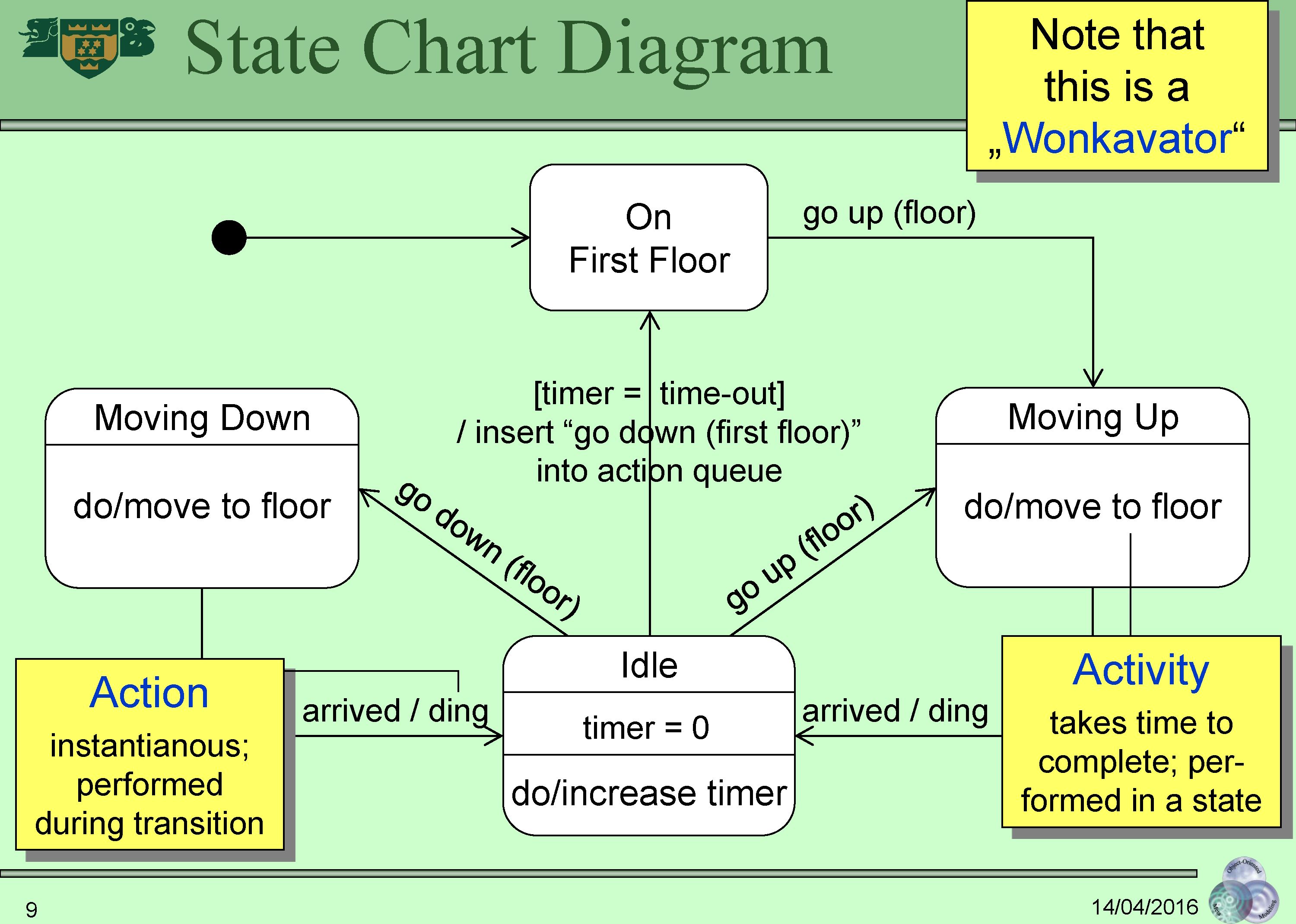
1 *<r°\*

State Chart Diagram

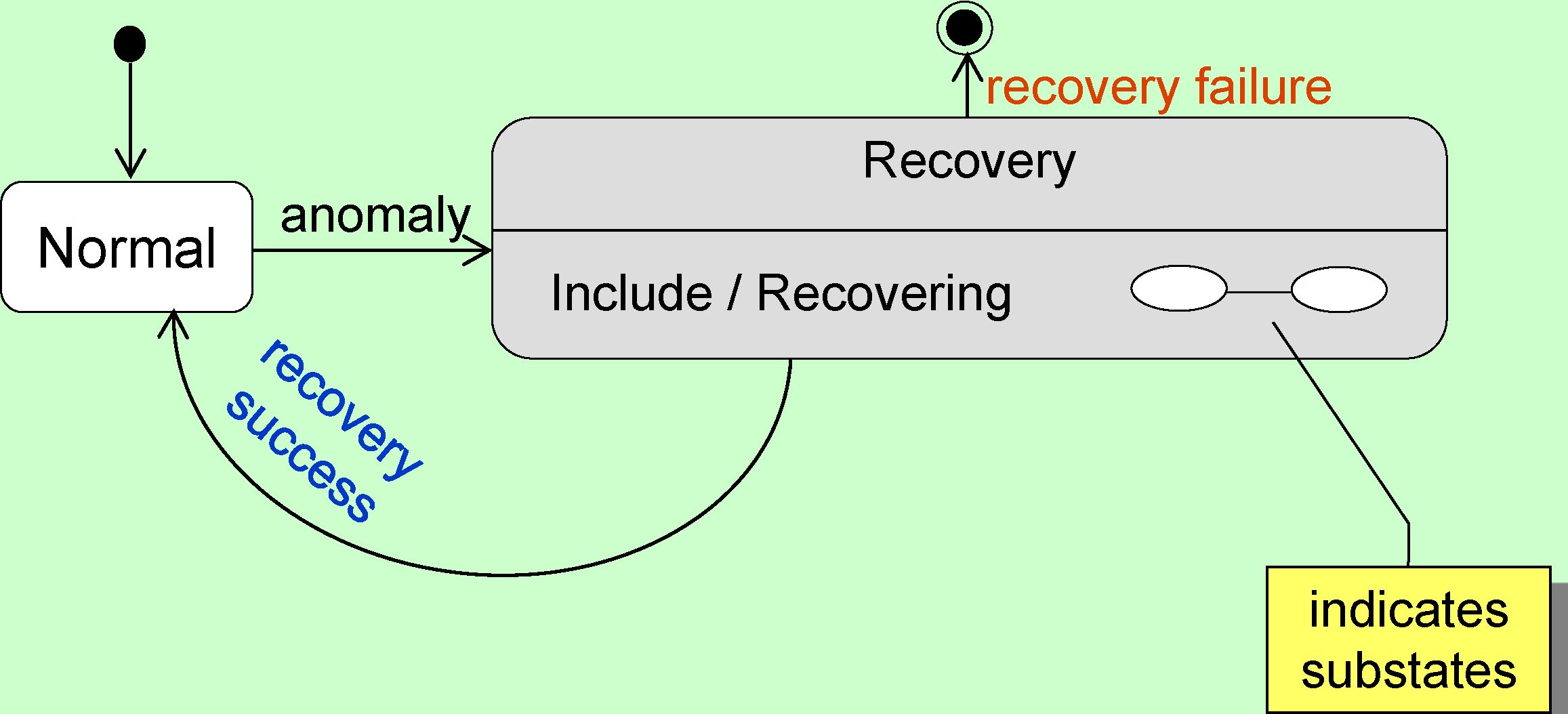
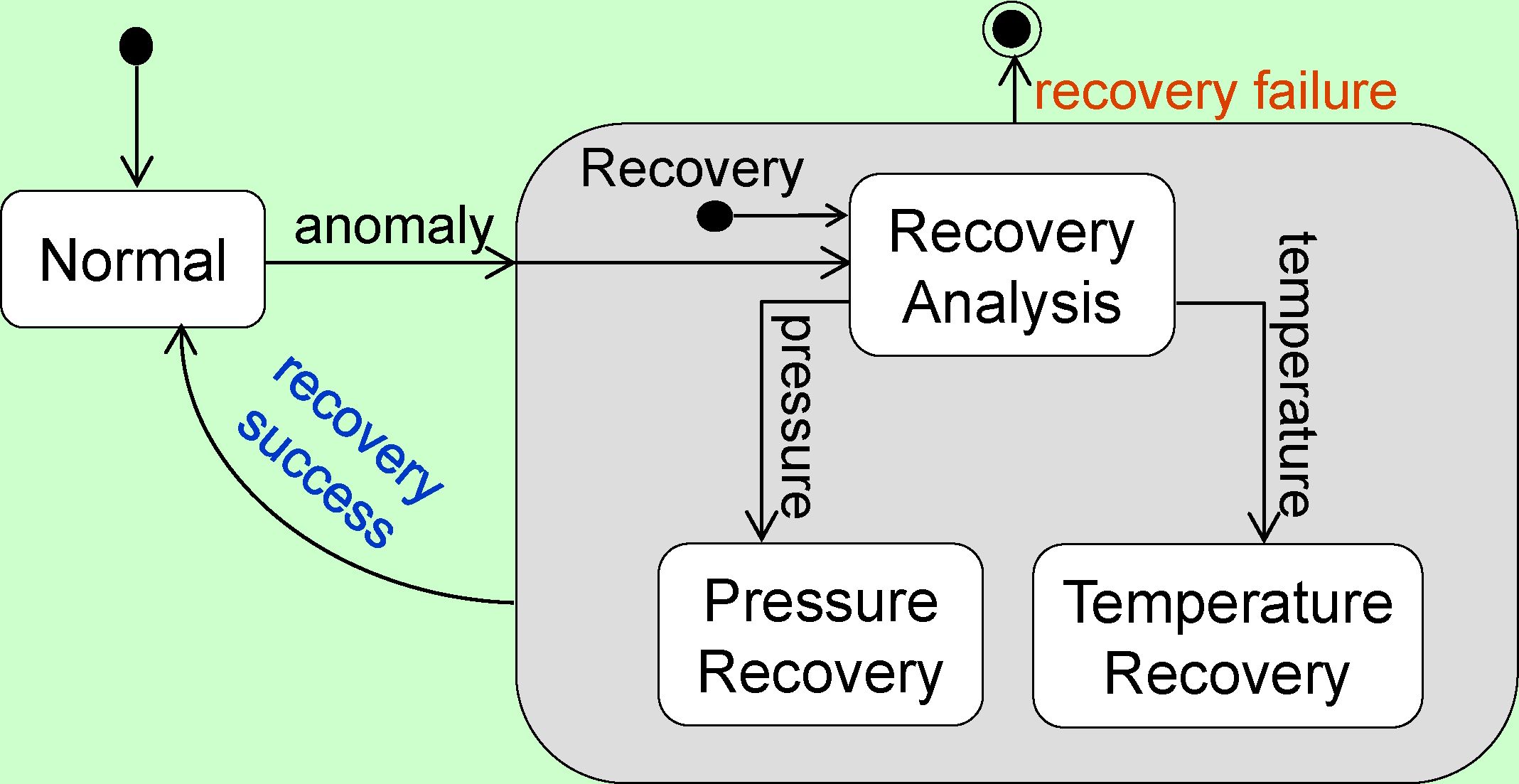
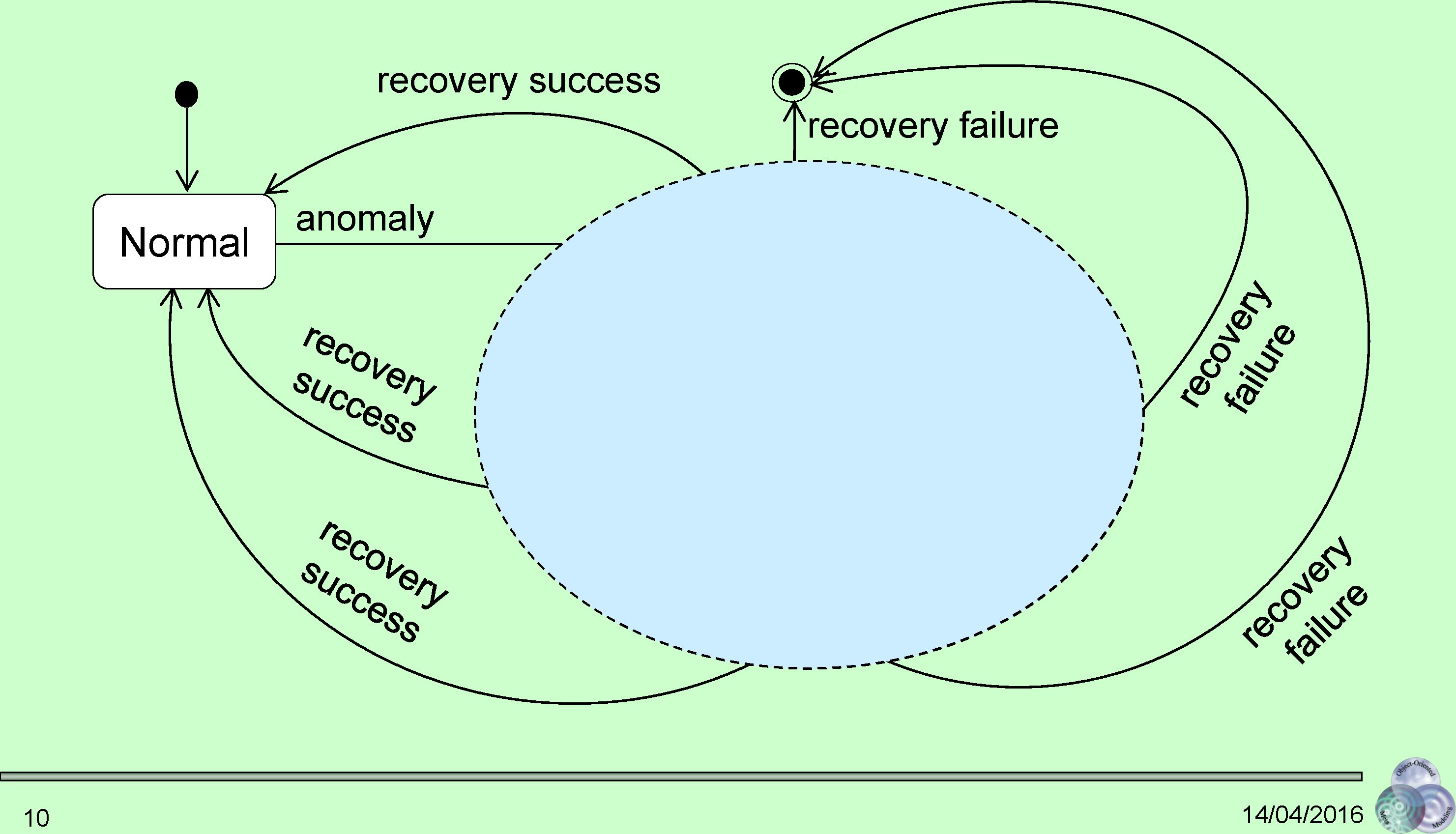
dial digit(n) [incomplete]



\*



State Chart Diagram



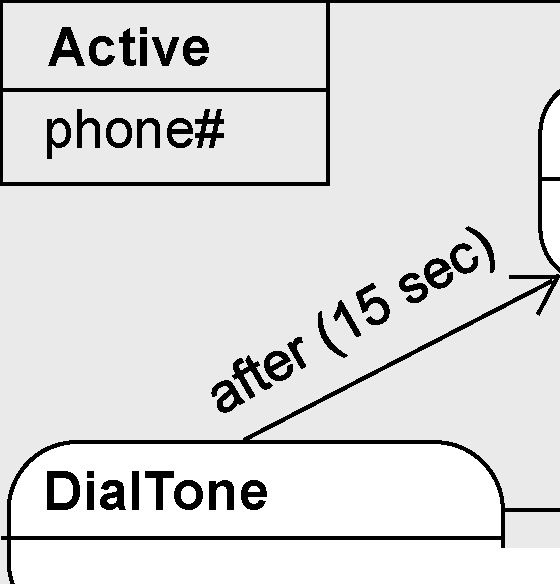
lift receiver /get dialedtone

**Timeout**

**do / play message**

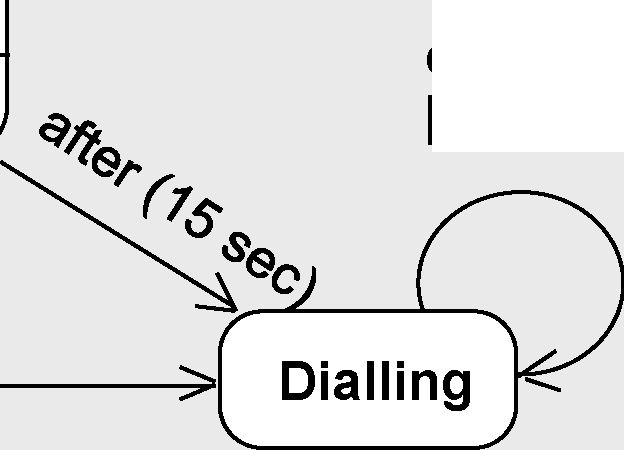
**\do / play dial tone**J

*7*K



dial digit (n)

**dial digit(n) [incomplete]**



Idle

7V

Invalid

dial digit (n) [invalid]

dial digit (n)

[valid]

/connect

Pinned

Connecting

A

callee

answers

|  |  |
| --- | --- |
| r Busy N | ^ busy |
| ^do / play busy toney |
|  |

callee hangs up

do / play message

caller hangs up /disconnect

V

Talking

callee answers /enable speech

o o

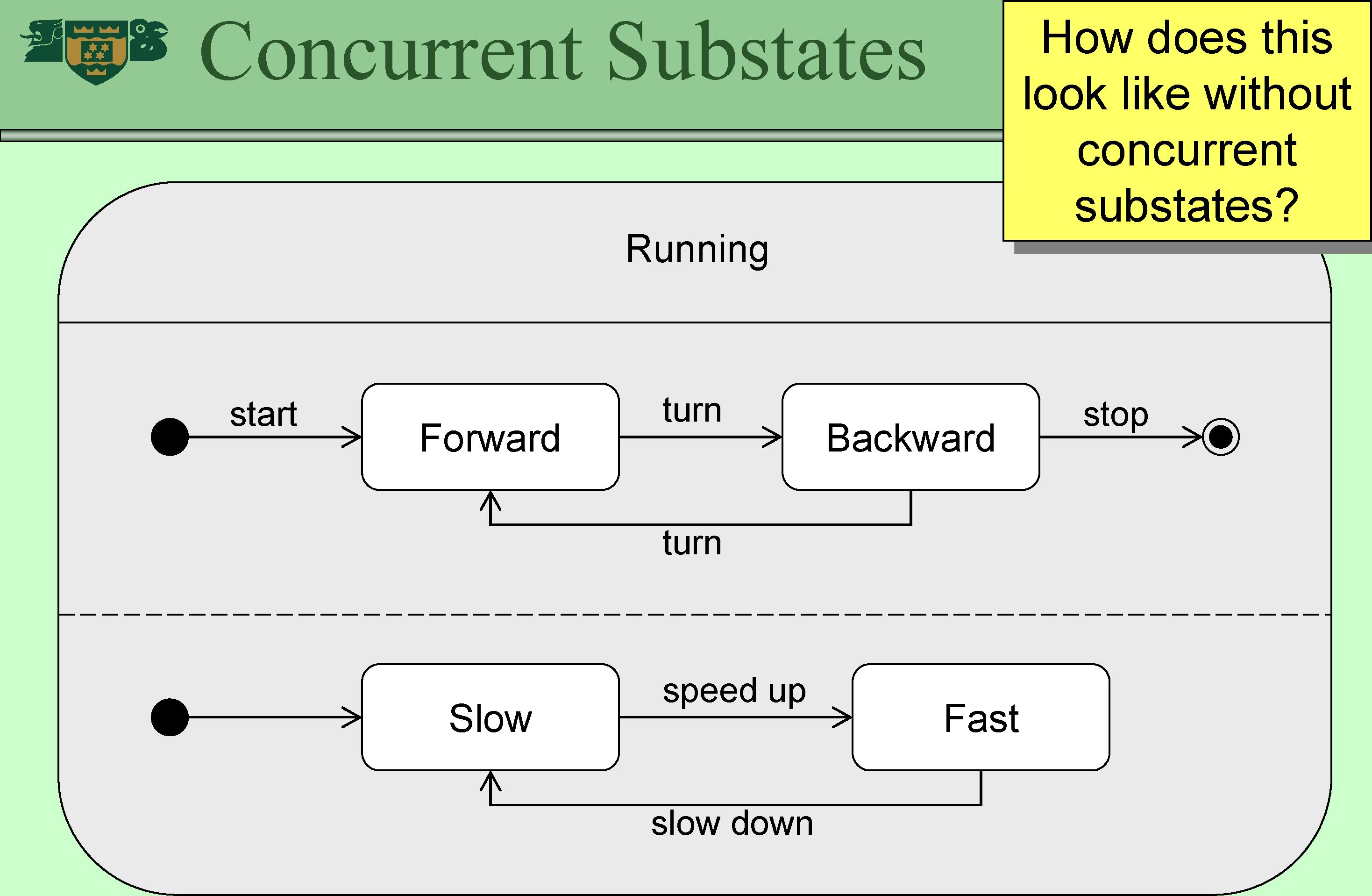
3 3 CD

a

V CL

Ringing

do / play ringing tone



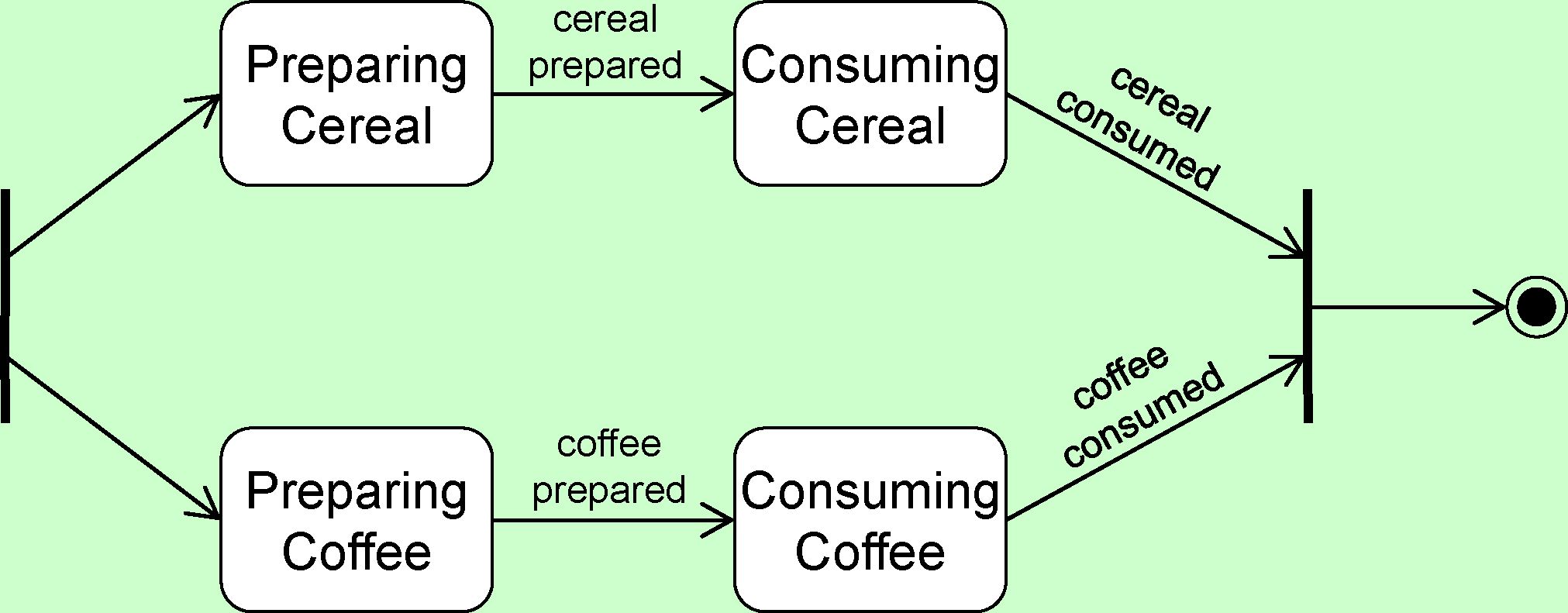
-I **,ru\.**

Fork and Join

Parallel Breakfast



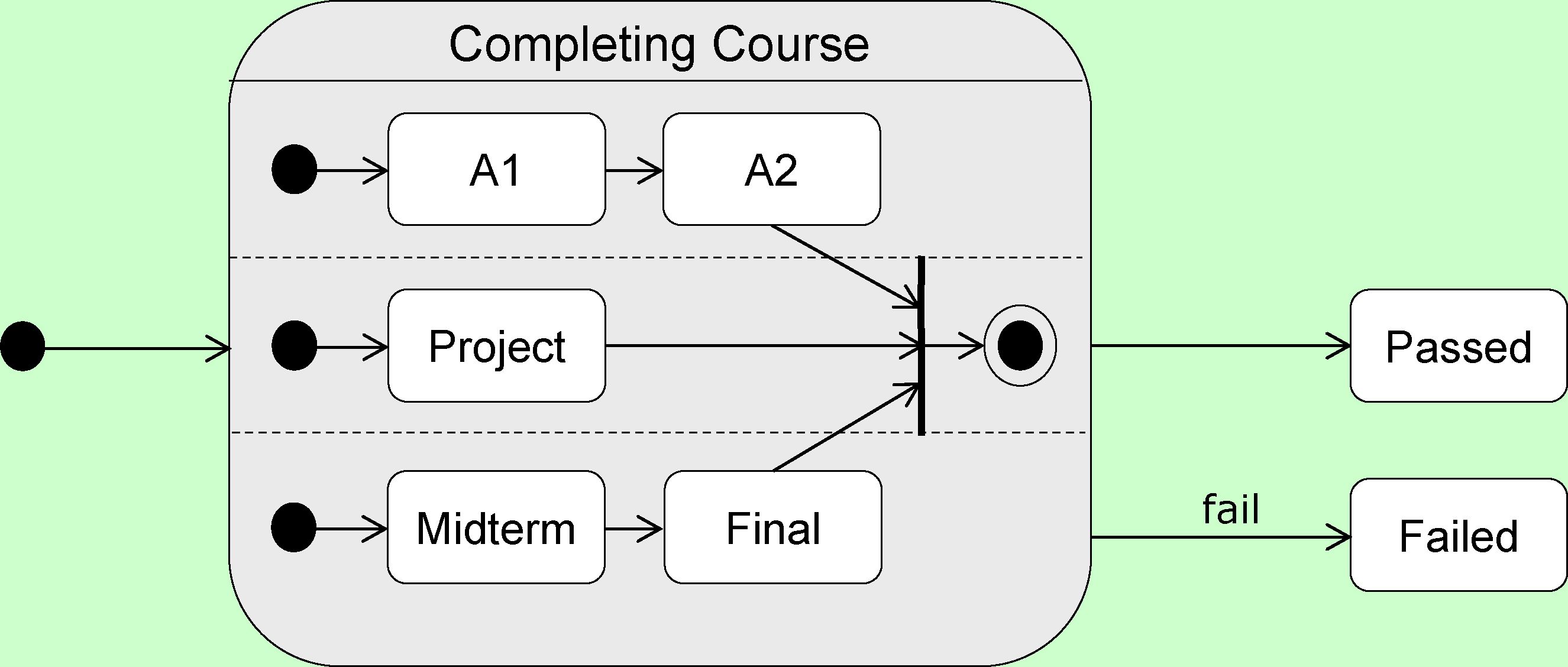
>



\*



Concurrent Substates



Multiple States

* Classic State Diagrams are XOR diagrams

» machine can only be in one of several states

* Statecharts are OR diagrams

» machine can be in multiple states

» extreme form are Petri Nets

» the possible combinations constitute new global states themselves

* different interpretation of a single substate
* way to concisely describe large finite state machines

—i

Statecharts

Complexity Reduction Through...

* Superstates

» combination of several substates into one superstate

» reduces complexity by hiding

* aggregated states and multiplied transitions
* Concurrent Machines

» parallel execution » reduces complexity by factorising

* multiplied states and corresponding transitions

—i

**JIMS**

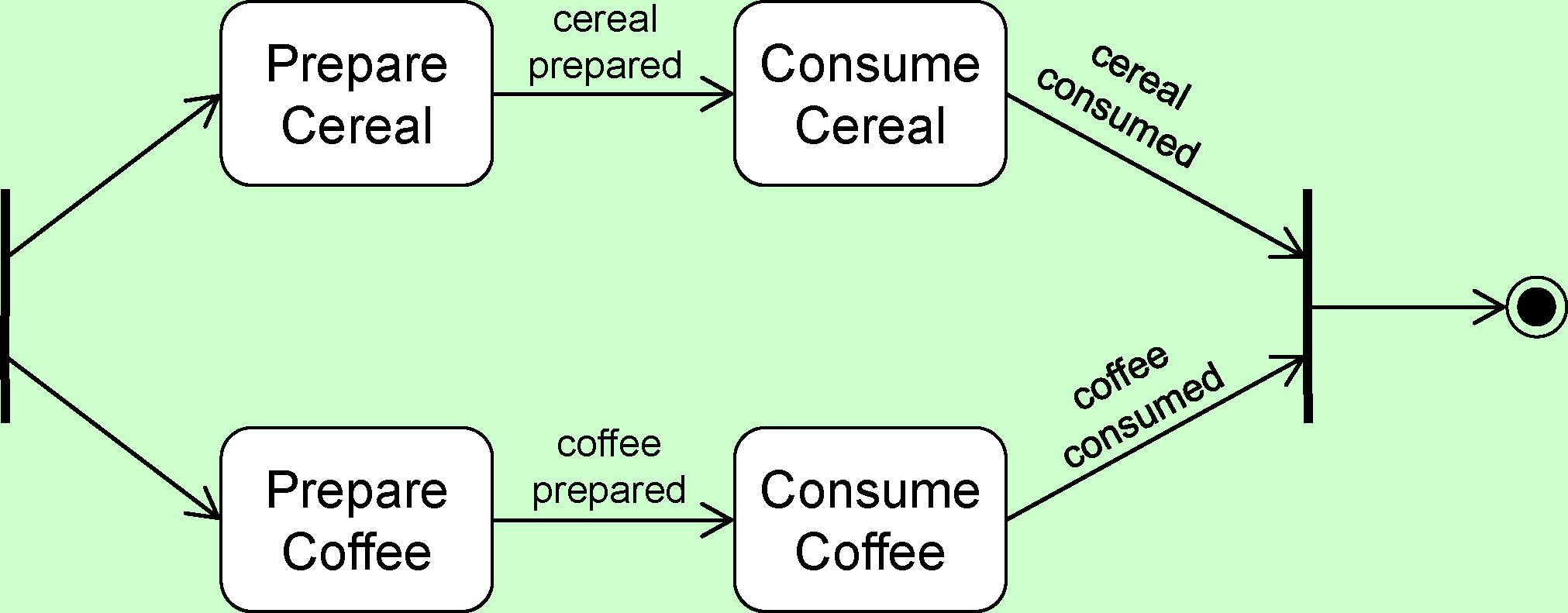
14/04/2016 % ’fy

Activity Diagrams Revisited

Parallel Breakfast as Parallel Activities



>



\*



Activity Diagrams Revisited

Activity Diagrams & State Diagrams

* A Correspondence

» Activities correspond to States » Control flow lines correspond to (termination) Events

* Same Semantic Foundation

» token flow -+ Petri Nets

* Different interpretation and applications

» context with events vs automatic/implicit “next” events » reactive behaviour vs control flow

—i

**dMS**

14/04/2016 % f y

Imprecise Natural Language

Informal "Specifications"

* *“I'm going to teach you how to solve the Rubik's cube in about 30min.”*

» does not appear to be a fast method, right?

* *“No eating or drinking from cups without lids.”*

» so I can eat from a plate?

