Object Oriented Analysis & Design

Engr. Abdul-Rahman Mahmood

DPM, MCP, QMR(ISO9001:2000)

armahmood786@yahoo.com	alphasecure@gmail.com
alphapeeler.sf.net/pubkeys/pkey.htm	http://alphapeeler.sourceforge.net
pk.linkedin.com/in/armahmood	http://alphapeeler.tumblr.com
www.twitter.com/alphapeeler	armahmood786@jabber.org
www.facebook.com/alphapeeler	alphapeeler@aim.com
abdulmahmood-sss alphasecure	mahmood_cubix 48660186
armahmood786@hotmail.com	alphapeeler@icloud.com
http://alphapeeler.sf.net/me	http://alphapeeler.sf.net/acms/

UML Components

Part-2

Objectives

- The runtime components
- The executable components
- Class assignment and interfaces of components

components types

- Component:
  - Physical set of object based or functional construct that provides functionality through well defined communication mechanism.
- Types:
  - **Compile time / Link Time**: Object code libraries.
  - **Run-time**: in-memory instantiation of these build time constructs.
  - **Executable:** Example is building an application to perform a count of statements in the source code files:

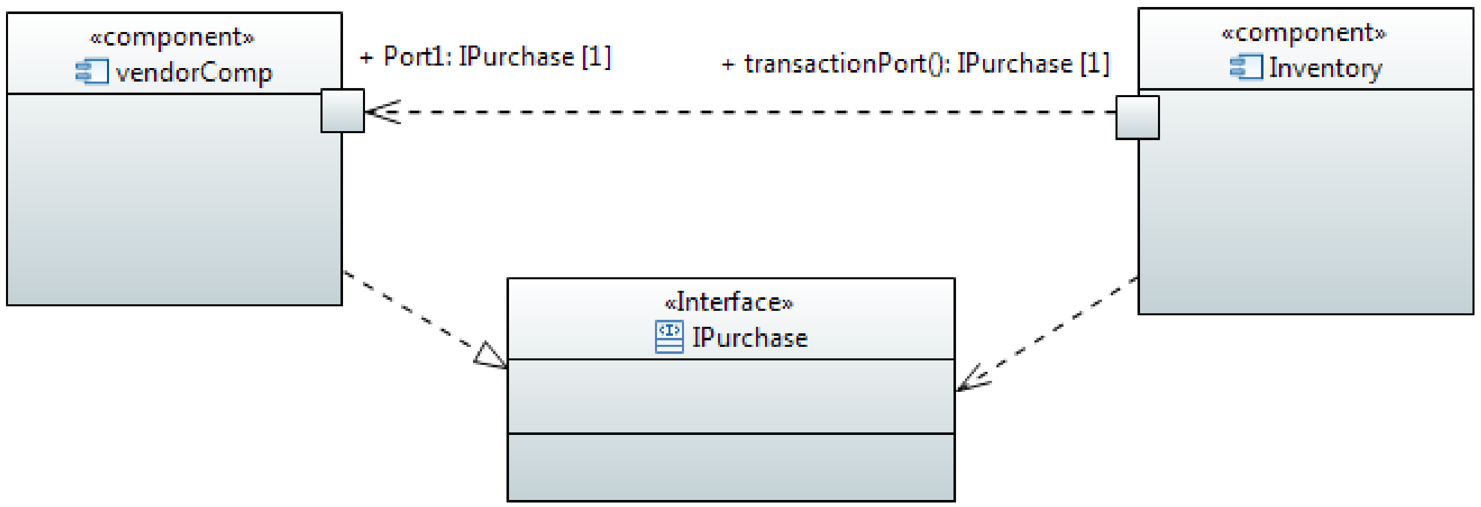
    ```
    cat *.cpp |grep “;”| wc –l
    ```

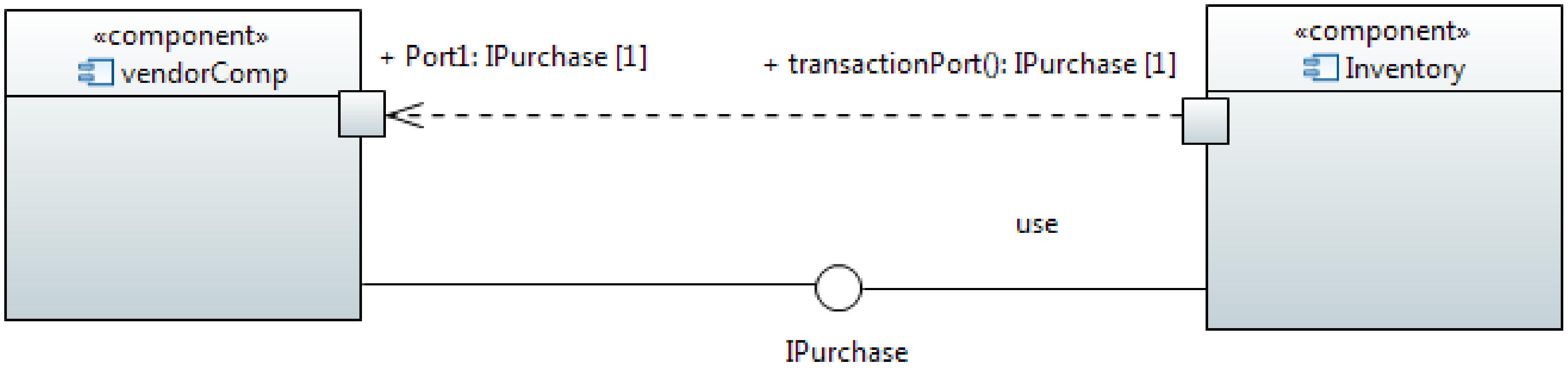
    Here pipe provides a data port to the components for communication.

Papyrus class tutorial

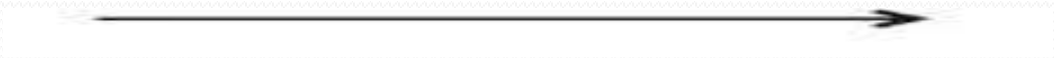
- 1. Define components
- 2. Define interface
- 3. Format interface
- 4. Using dependencies
- 5. Define ports
- 6. Use port types
- 7. Using ports

Papyrus components demo



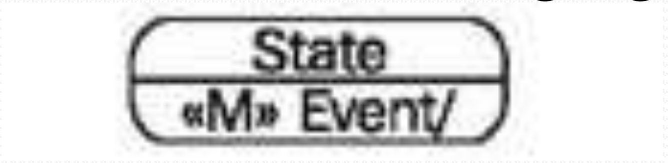


State-chart diagrams

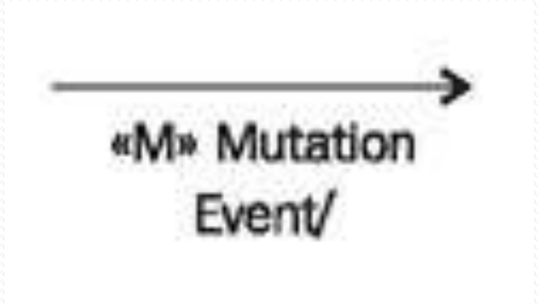
- **State:** The state of an object is always determined by its attributes and associations. States in statechart diagrams represent a *set* of those value combinations, in which an object *behaves the same* in response to events.
- Therefore, not every modification of an attribute leads to a new state.
- **Transition:** A transition represents the change from one state to another. 
-

State-chart diagrams

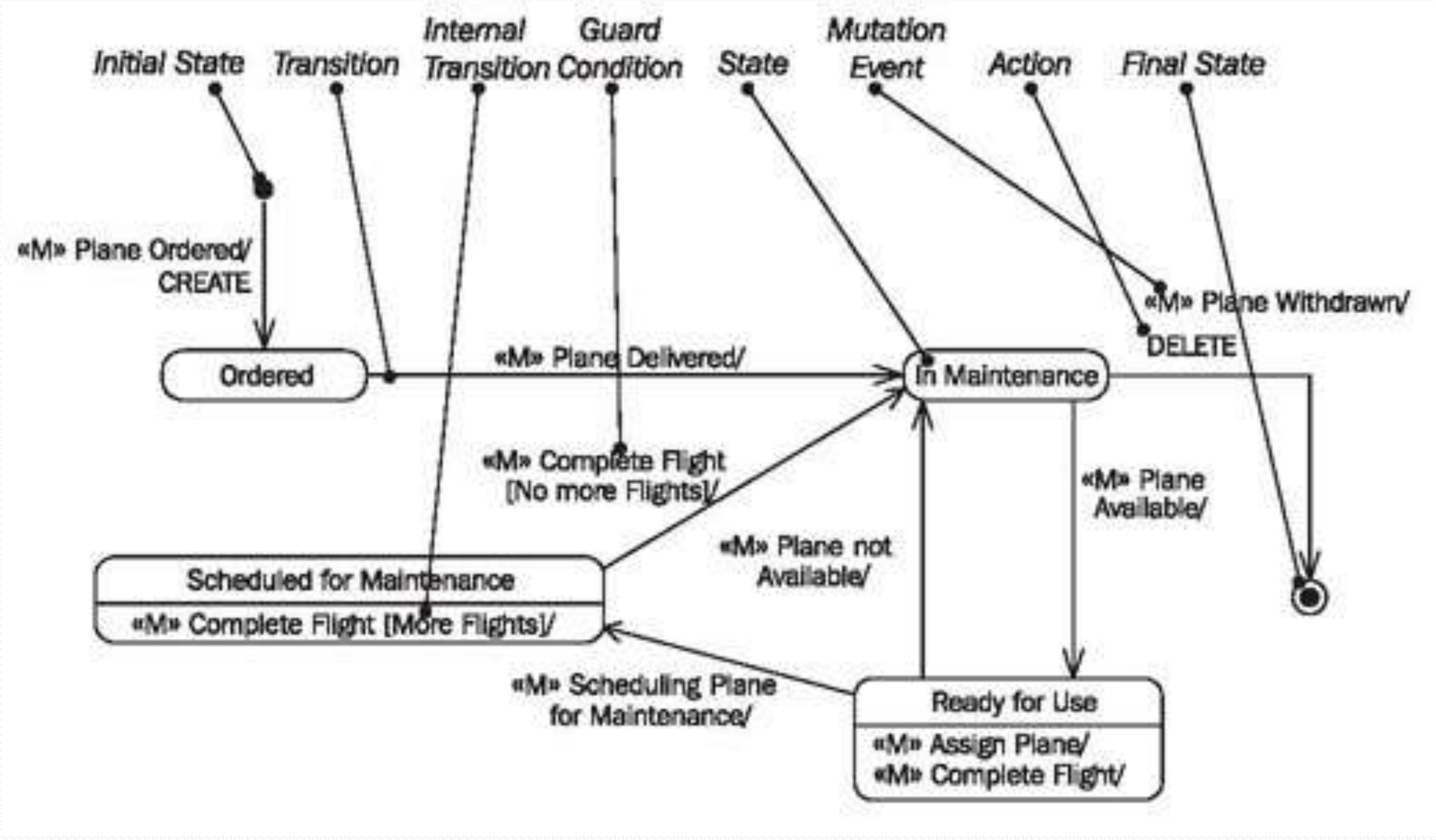
- **Internal Transition:** transition from one state to itself. Object handles event without changing its state.



- **Mutation Event:** The initiator of a transition from one state to another, or for an internal transition, where the state remains the same.
-

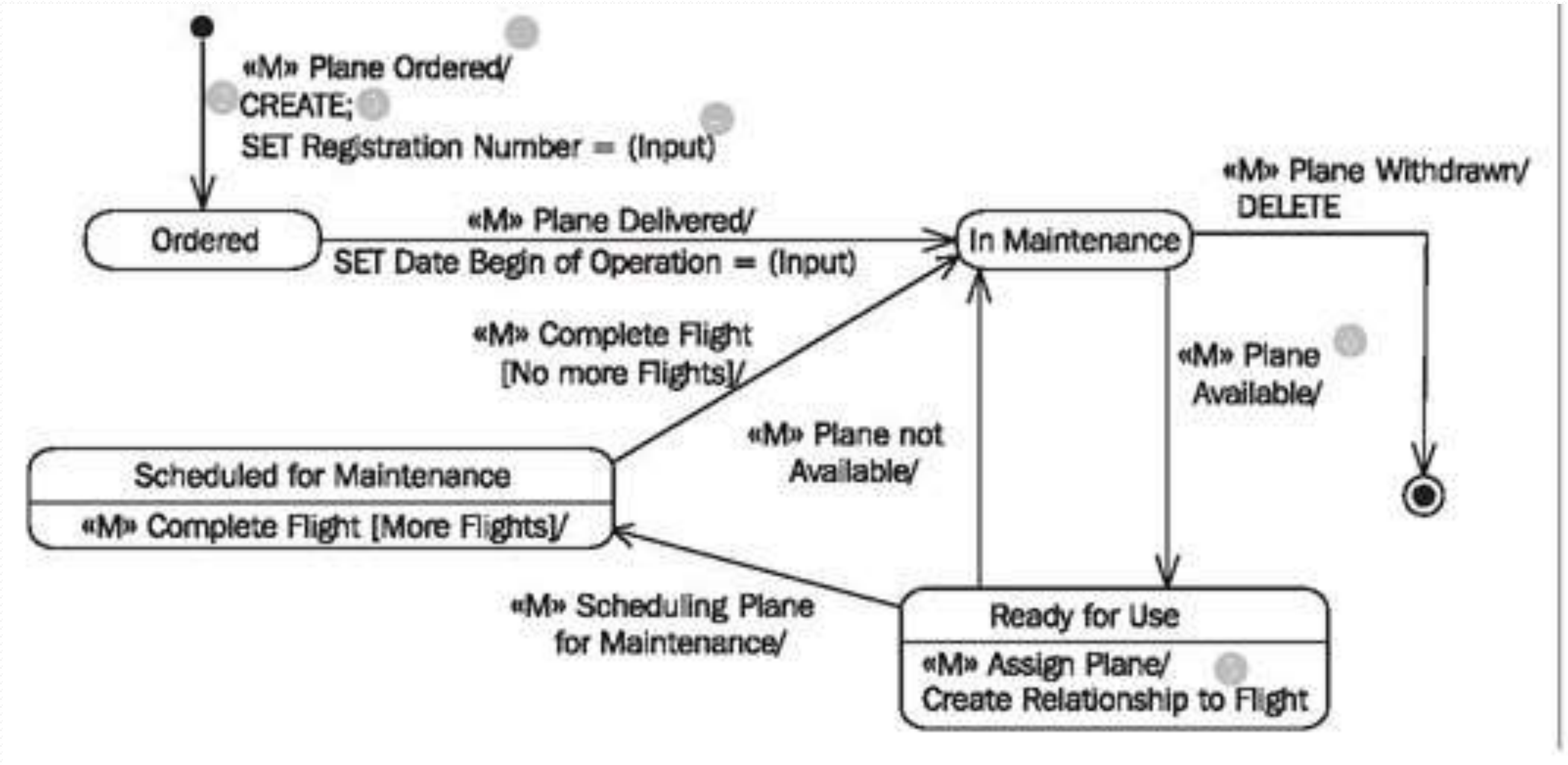


State-chart diagrams



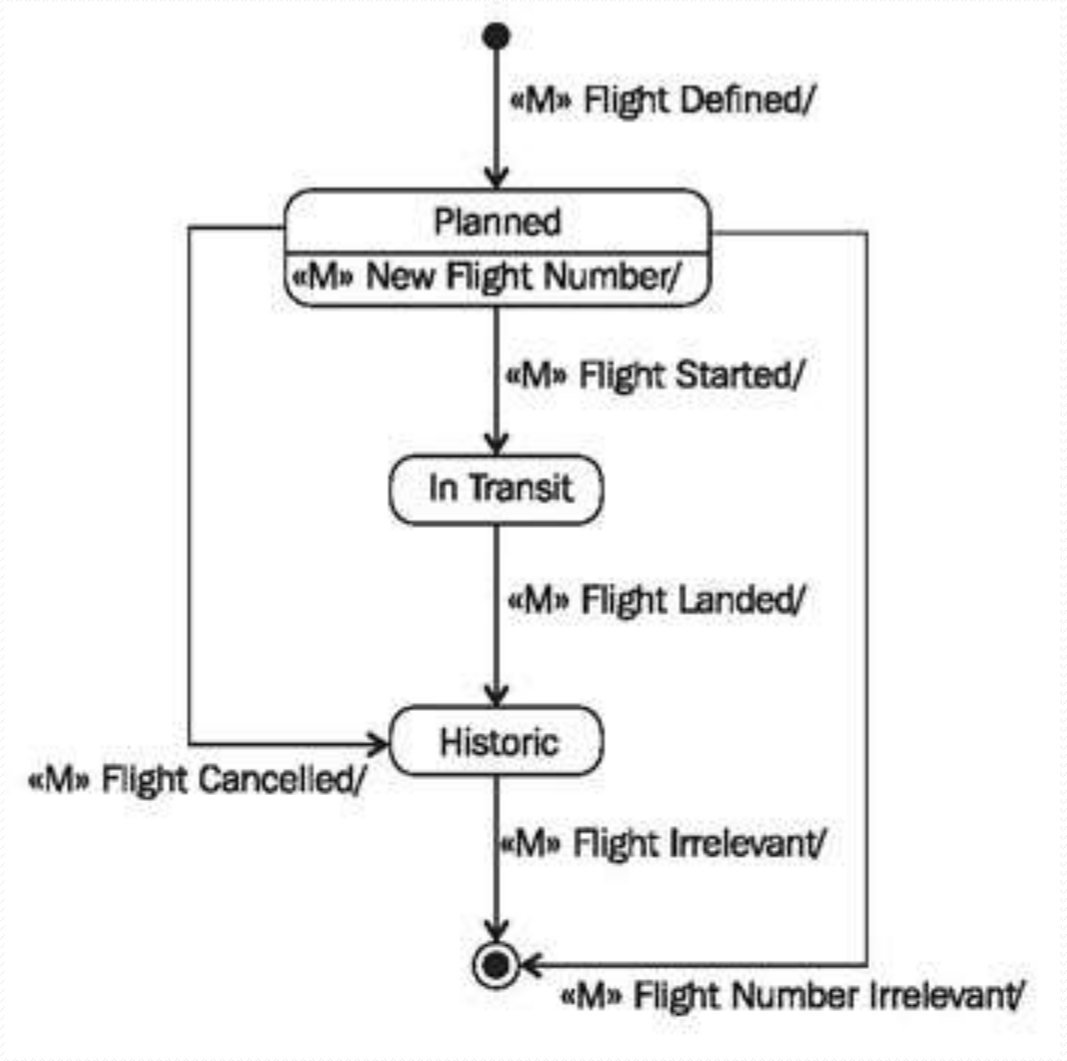
Elements of the statechart diagram

State-chart diagrams



Statechart diagram with internal transitions and guard conditions.

State-chart diagrams



Statechart diagram of the class "Flight"

State-chart diagrams

- **Checklist : Statechart Diagrams**
- Identify mutation events relevant for the object—What affects the object?
- Group relevant events chronologically—How does a normal life look?
- Model states and transitions—Which states are there?
- Add actions to the statechart diagram—What do objects do?
- Verify the statechart diagram—Is everything correct?

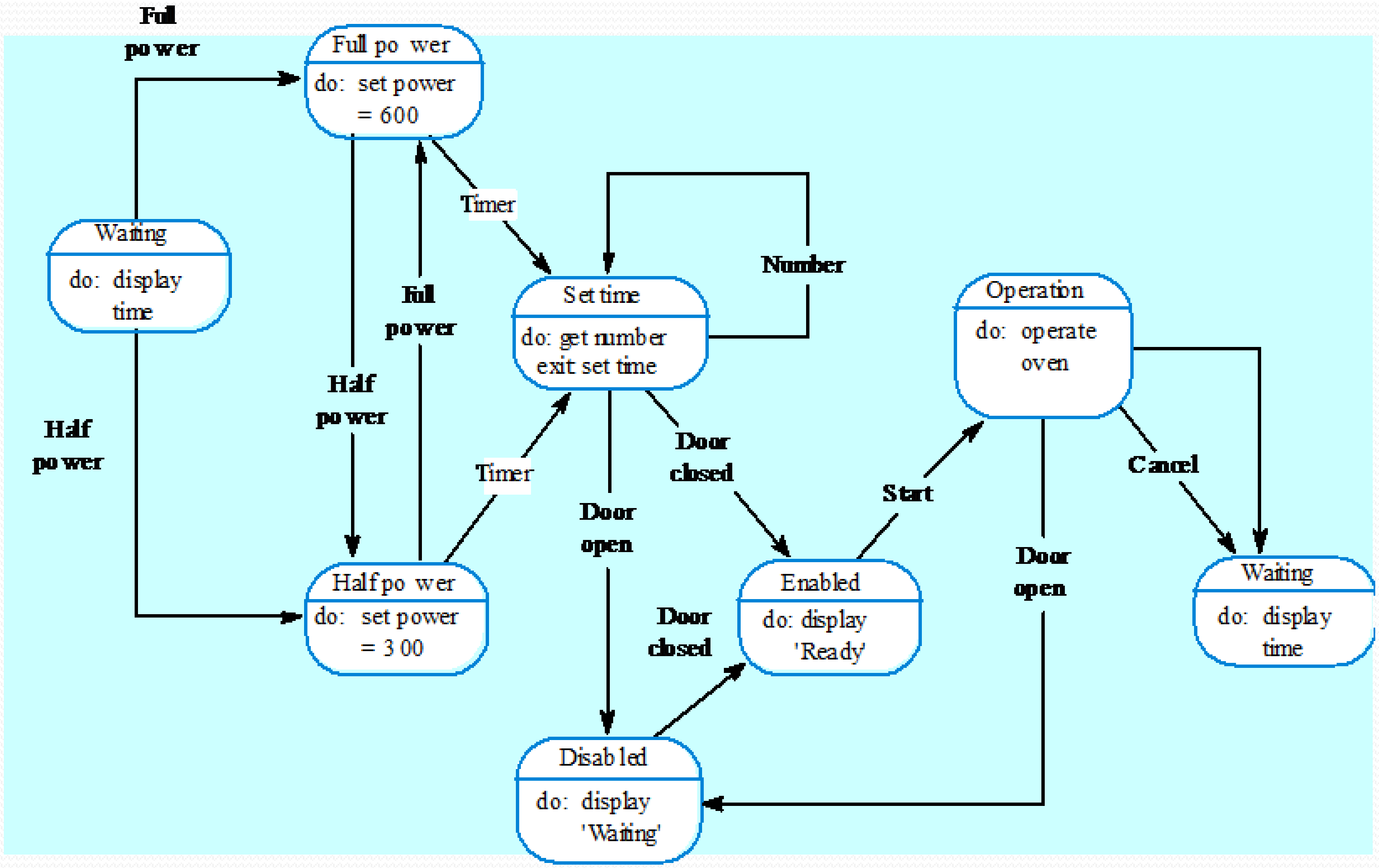
State machine models

- **These model the behaviour of the system in response to external and internal events.**
- They show the system's responses to stimuli so are often used for modelling **real-time systems**.
- State machine models **show system states as nodes** and **events as arcs** between these nodes. When an event occurs, the system moves from one state to another.
- **Statecharts** are an **integral part of the UML** and are used to represent state machine models.

Statecharts

- Allow the **decomposition of a model into sub-models** (see following slide).
- A brief description of the actions is included following the ‘do’ in each state.
- Can be complemented by tables describing the states and the stimuli.

Microwave oven model



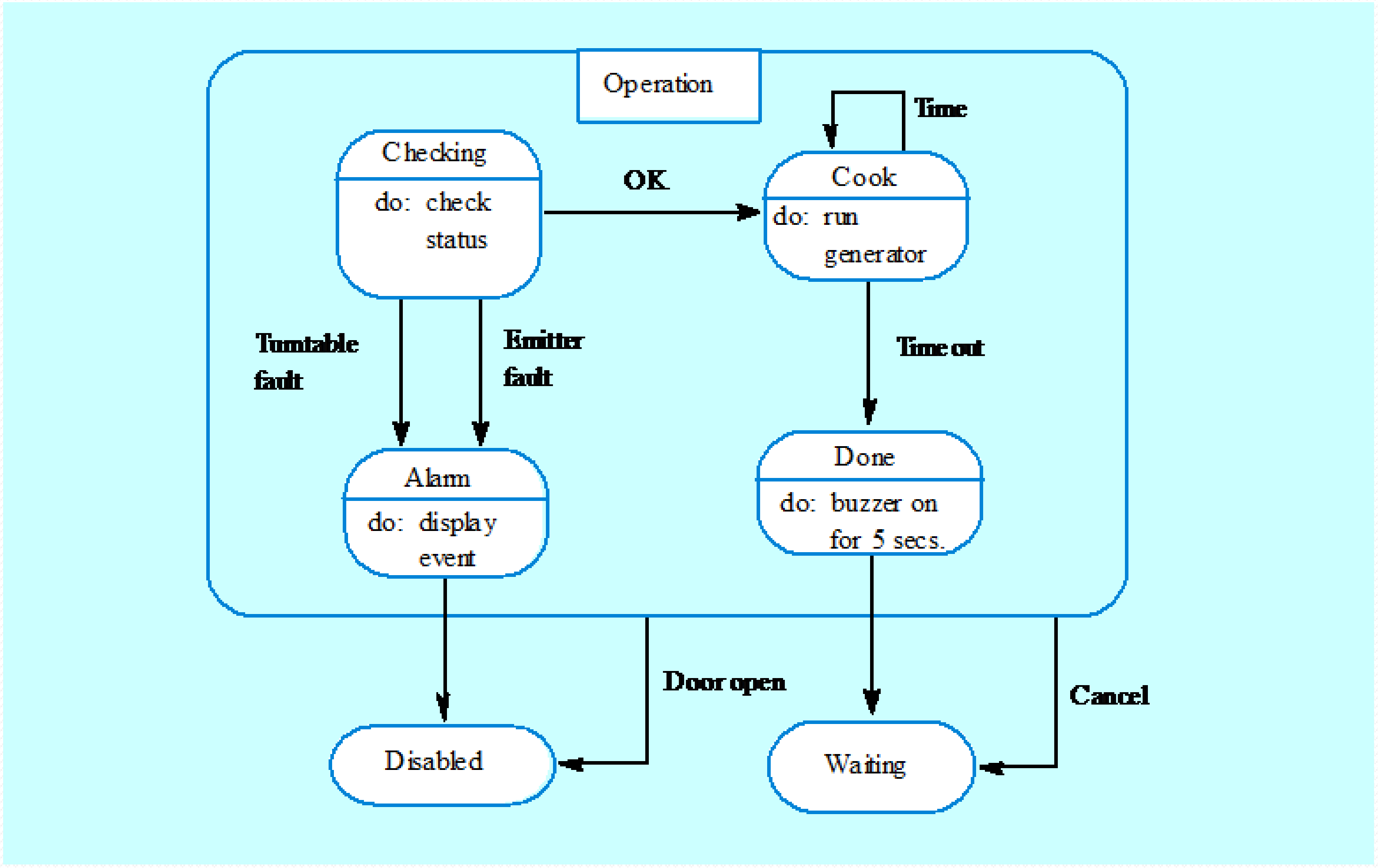
Microwave oven state description

State	Description
Waiting	The oven is waiting for input. The display shows the current time.
Half power	The oven power is set to 300 watts. The display shows ŌHalf powerÕ.
Full power	The oven power is set to 600 watts. The display shows ŌFull powerÕ.
Set time	The cooking time is set to the userÕs input value. The display shows the cooking time selected and is updated as the time is set.
Disabled	Oven operation is disabled for safety. Interior oven light is on. Display shows ŌNot readyÕ.
Enabled	Oven operation is enabled. Interior oven light is off. Display shows ŌReady to cookÕ.
Operation	Oven in operation. Interior oven light is on. Display shows the timer countdown. On completion of cooking, the buzzer is sounded for 5 seconds. Oven light is on. Display shows ŌCooking completeÕ while buzzer is sounding.

Microwave oven stimuli

Stimulus	Description
Half power	The user has pressed the half power button
Full power	The user has pressed the full power button
Timer	The user has pressed one of the timer buttons
Number	The user has pressed a numeric key
Door open	The oven door switch is not closed
Door closed	The oven door switch is closed
Start	The user has pressed the start button
Cancel	The user has pressed the cancel button

Microwave oven operation



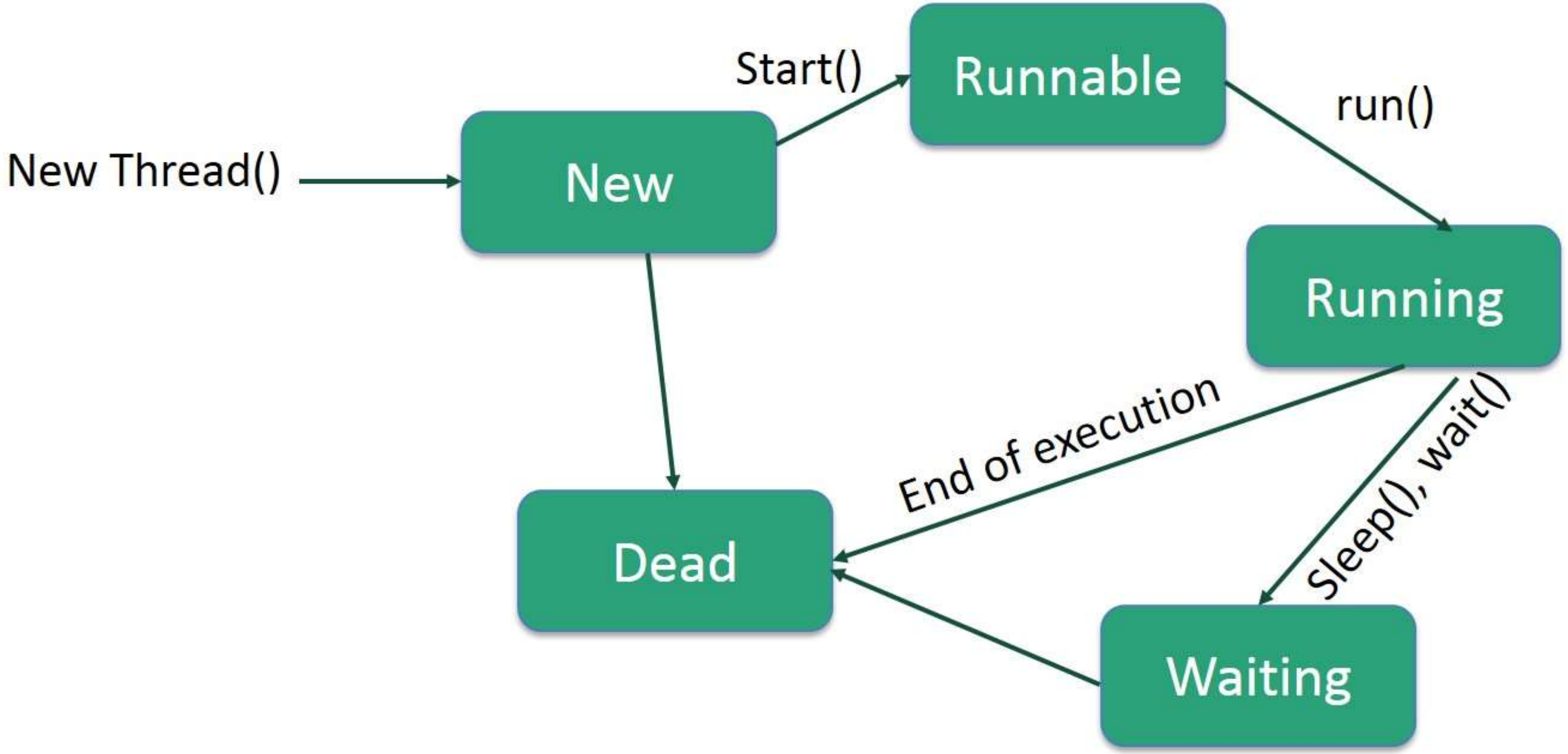
Life Cycle of a Thread

- A thread goes through various stages in its life cycle. E.g., a thread is born, started, runs, and then dies.
- **New** – A new thread begins its life cycle in the new state. It remains in this state until the program starts the thread. It is also referred to as a **born thread**.
- **Runnable** – After a newly born thread is started, the thread becomes runnable. A thread in this state is considered to be ready for executing its task.
- **Running** – executing task.
- **Waiting** – Sometimes, a thread transitions to the waiting state while the thread <u>waits for another thread </u>to perform a task. A thread transitions back to the runnable state only when another thread signals the waiting thread to continue executing.

Life Cycle of a Thread

- **Timed Waiting** – A runnable thread can enter the timed waiting state for a specified interval of time. A thread in this state transitions back to the runnable state when that <u>time interval expires </u>or when the event it is waiting for occurs. E.g., sleep() call.
- **Terminated (Dead)** – A runnable thread enters the terminated state when it completes its task or otherwise terminates.

- The following diagram shows the complete life cycle of a thread.



Class Exercise :

- Draw a state chart diagram for water tanker problem.

Solution

- States:

- 1. Out of port
- 2. Docked
- 3. Docked & Connected
- 4. Loading
- 5. Docked & disconnected
- 6. Un-Docked
- 7. Travelling to destination
- 8. Unloading