National University
Of Computer and Emerging Sciences

Software Design & Architecture

System Modelling
Behavioural Models

Sajid Anwer

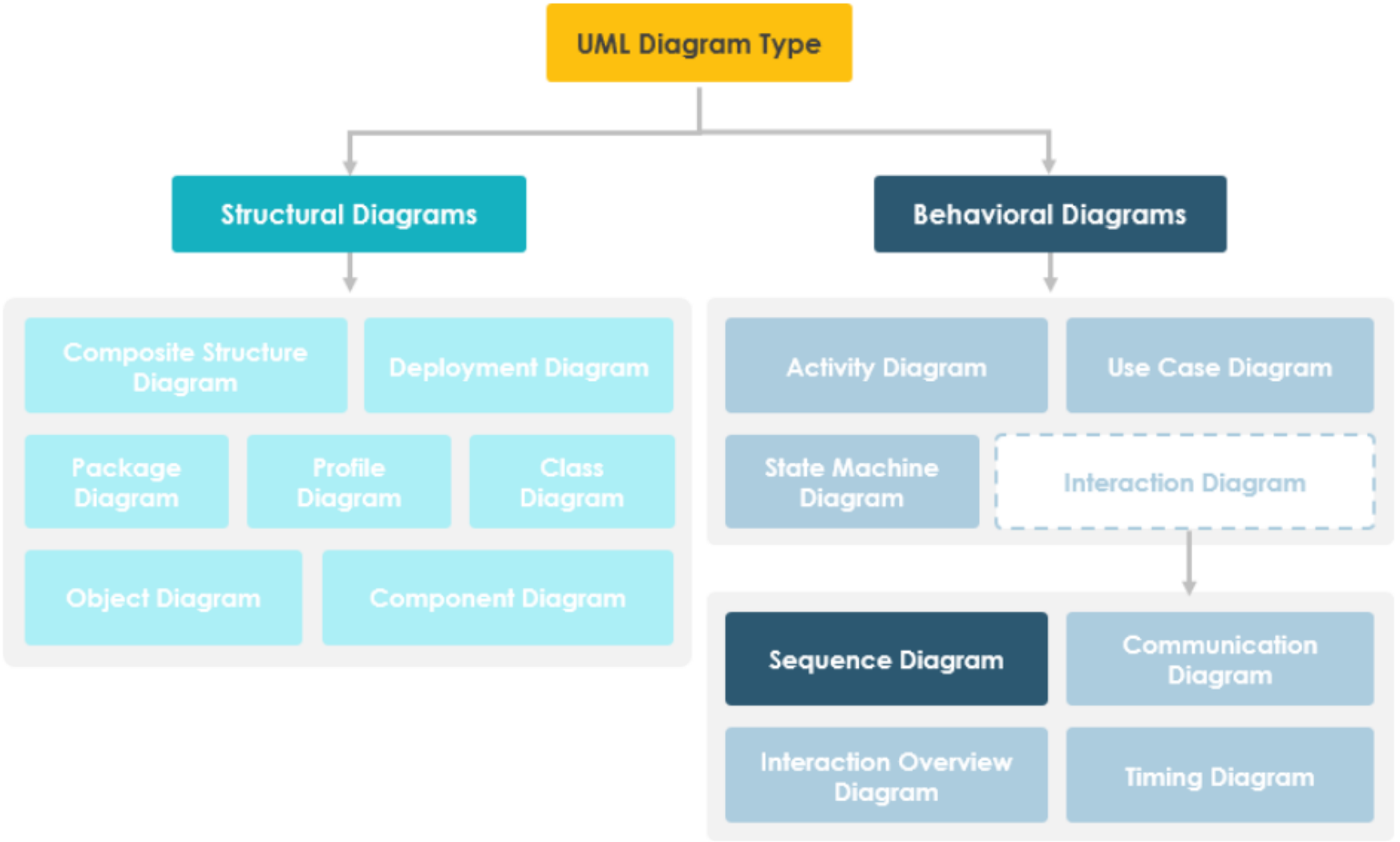
Department of Software Engineering,
FAST-NUCES, CFD Campus

Lecture Outline

- Behavioral Models
  - » Activity Diagram
  - » State Machine Diagram
  - » Interaction Models
    - Use case Diagram
    - Sequence Diagram

Already Covered

Behavioral Models



Behavioral Models

- Behavioral models are models of the *dynamic behavior* of a system as it is executing. They show *what happens* or what is supposed to happen when a *system responds* to a *stimulus from its environment*.

- You can think of these stimuli as being of two types:
  - *Data* Some data arrives that has to be processed by the system.
    - Business systems – Billing system
  - *Events* Some event happens that triggers system processing.
    - Events may have associated data, although this is not always the case.
    - Realtime systems – Microwave Oven

Behavior Models – Activity Diagram

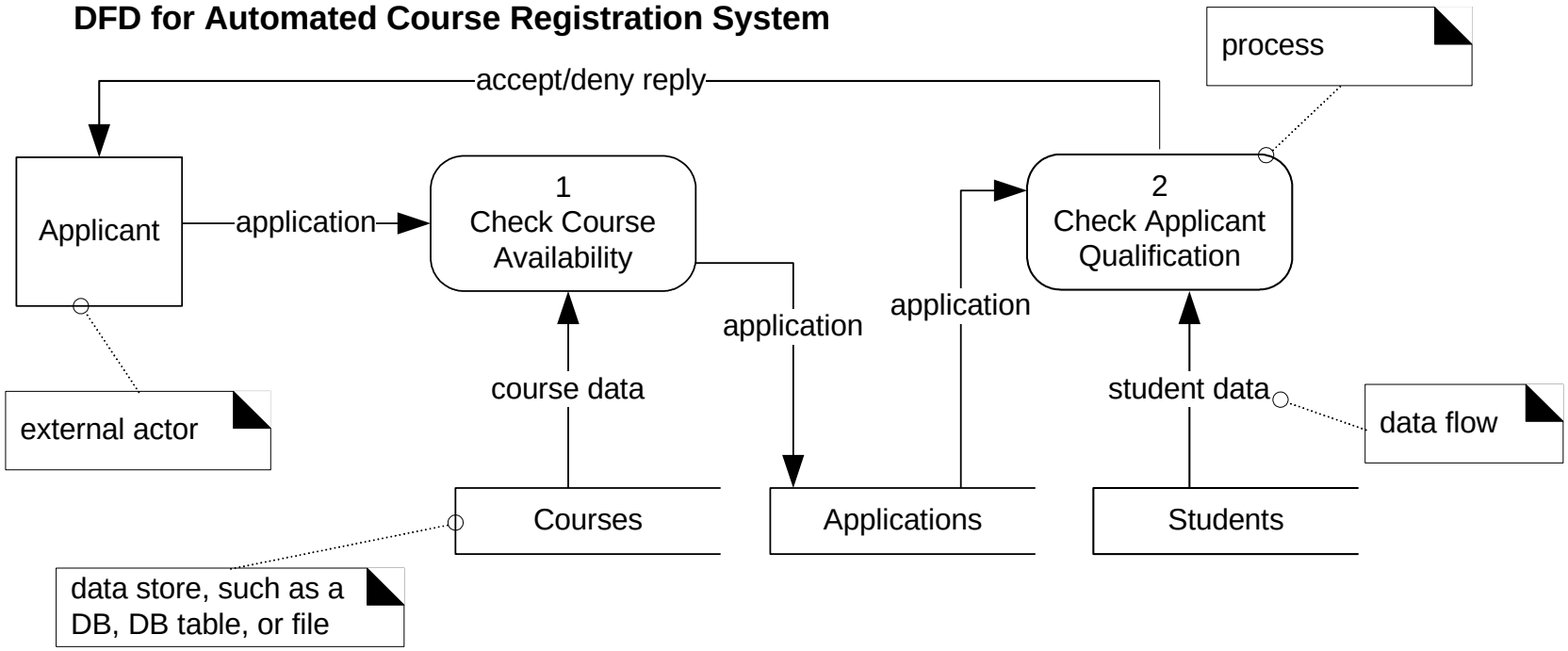
- It is used to model system behavior in the context of data processing.
  - » Data flow Diagram (DFD) can also be used to fulfill this purpose but DFDs has different rational.

- It is a conceptual diagram, used to:
  - » Model workflows between/within use cases
  - » Model complex workflows in operations on objects
  - » Model in detail complex activities in a high level activity Diagram
  - » Analysis of system requirements, *how?*

- It uses the constructs of flow chart with some other advanced constructs.

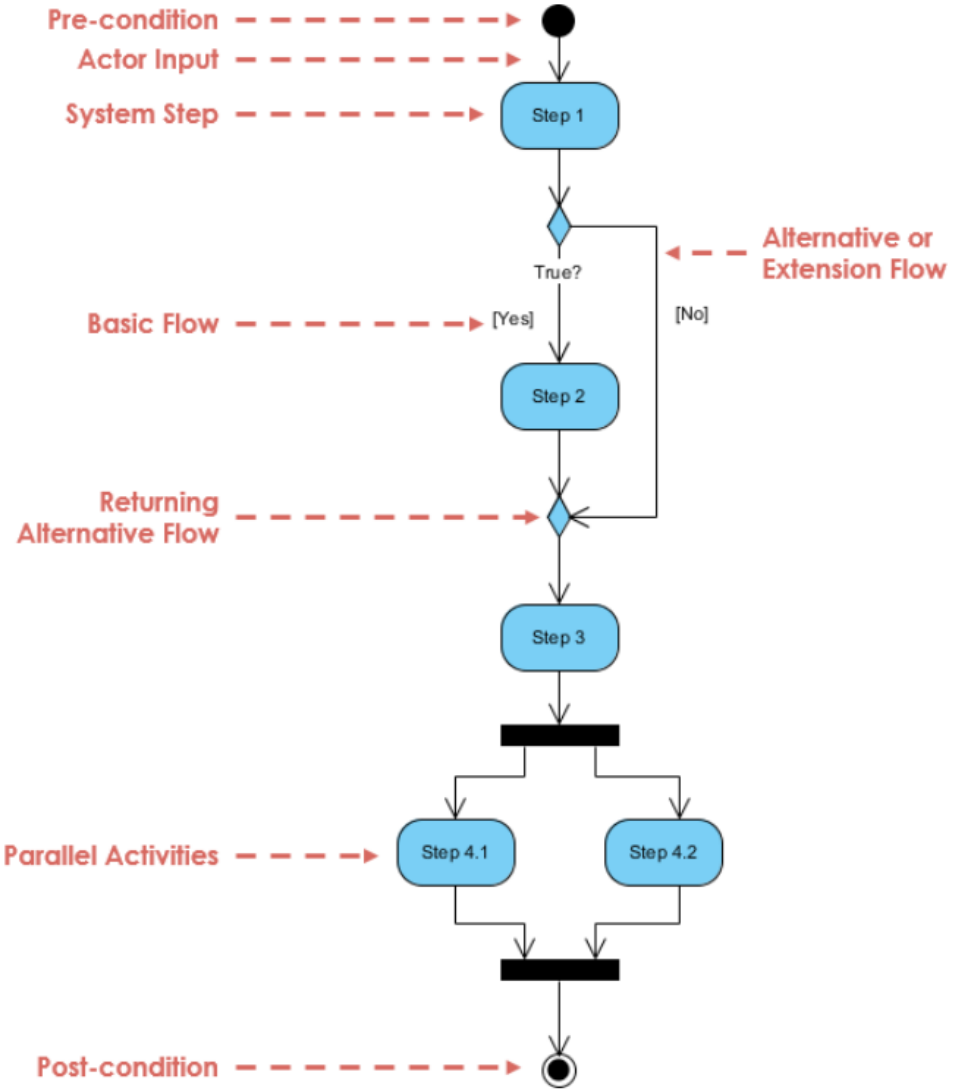
- It is not directly related to system implementation but helps to improve system understanding.

Behavior Models – Data Flow Diagrams



- DFDs were useful to document the *major data flows* or to explore a new high-level design in terms of data flow, however, not consider the *control flow.*
- UML activity diagrams can be used for data flow modeling, replacing traditional DFD notation.
- The activity diagram can show both *control flow* and *data flow*.

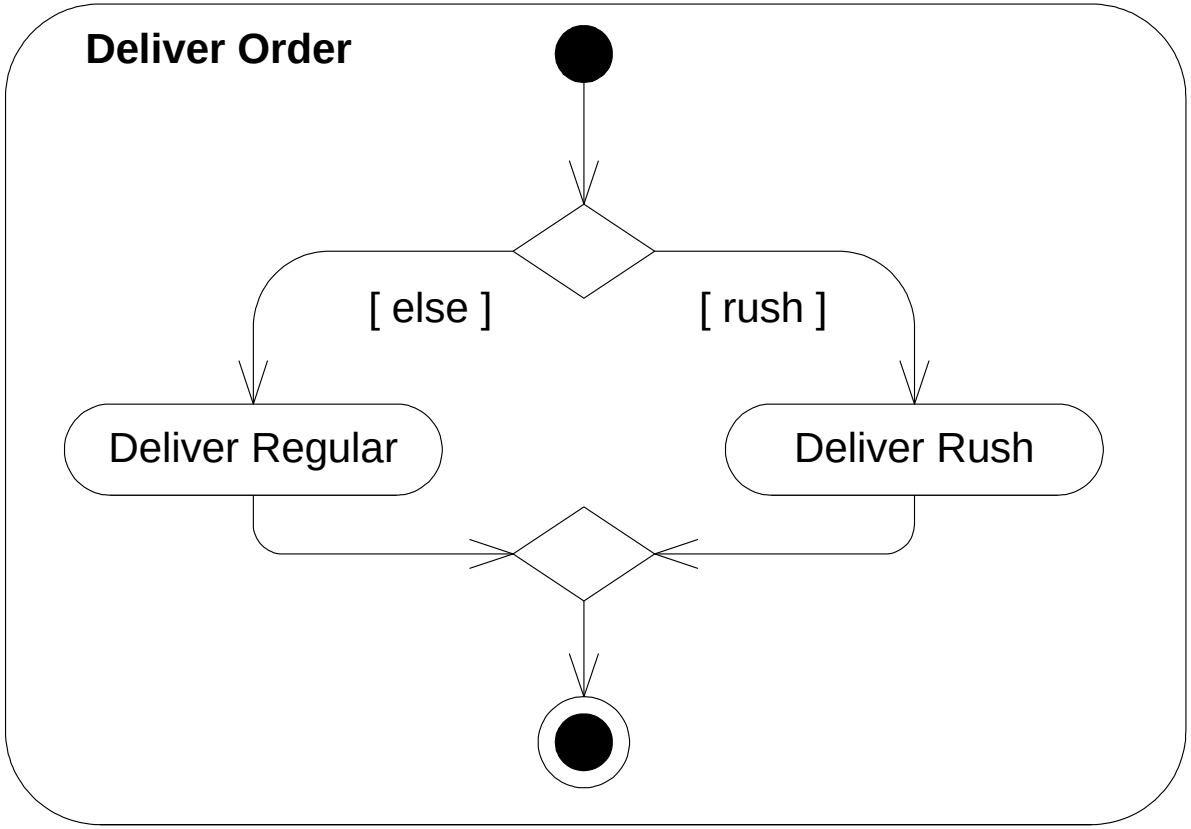
Behavior Models – Activity Diagram



Behavior Models – Activity Diagram

Decision: Any branch happens. Mutual exclusion

Merge: *Any* input leads to continuation. This is in contrast to a *join*, in which case *all* the inputs have to arrive before it continues.



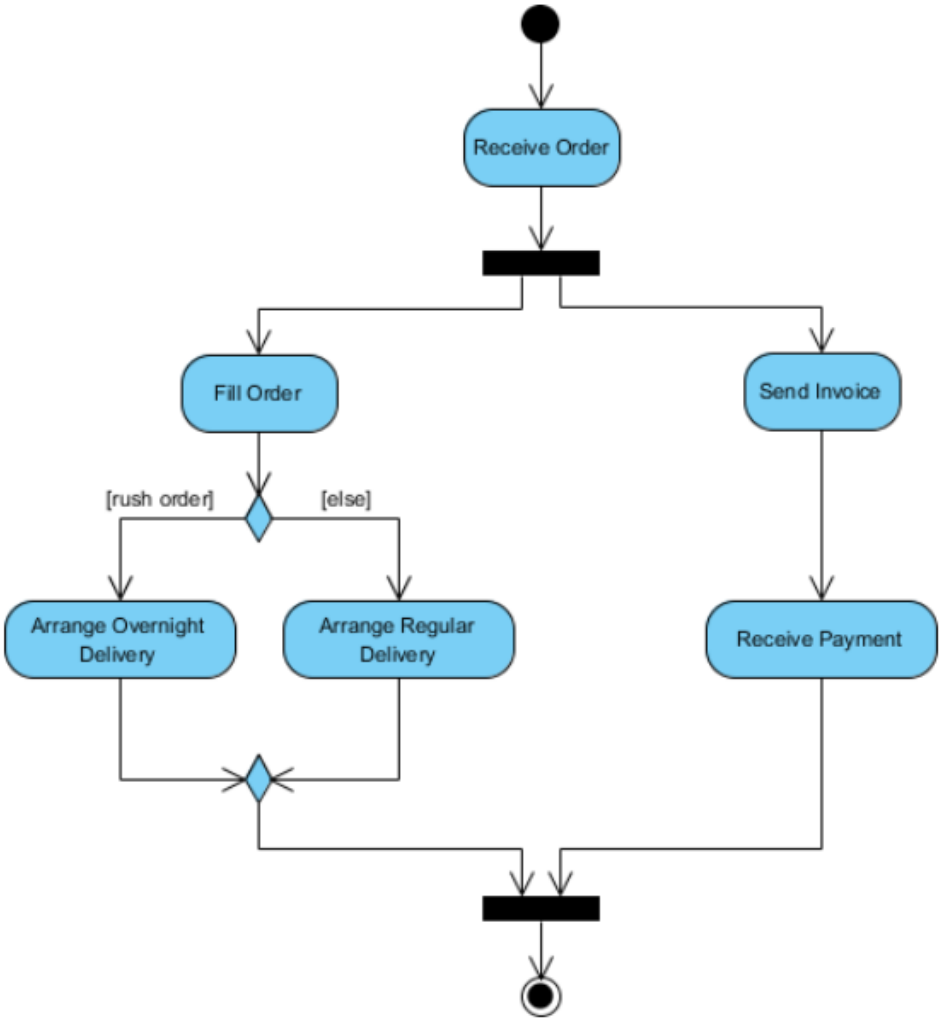
Behavior Models – Activity Diagram

- Once the order is received, the activities split into two parallel sets of activities. One side fills and sends the order while the other handles the billing.

- On the Fill Order side, the method of delivery is decided conditionally. Depending on the condition either the Overnight Delivery activity or the Regular Delivery activity is performed.

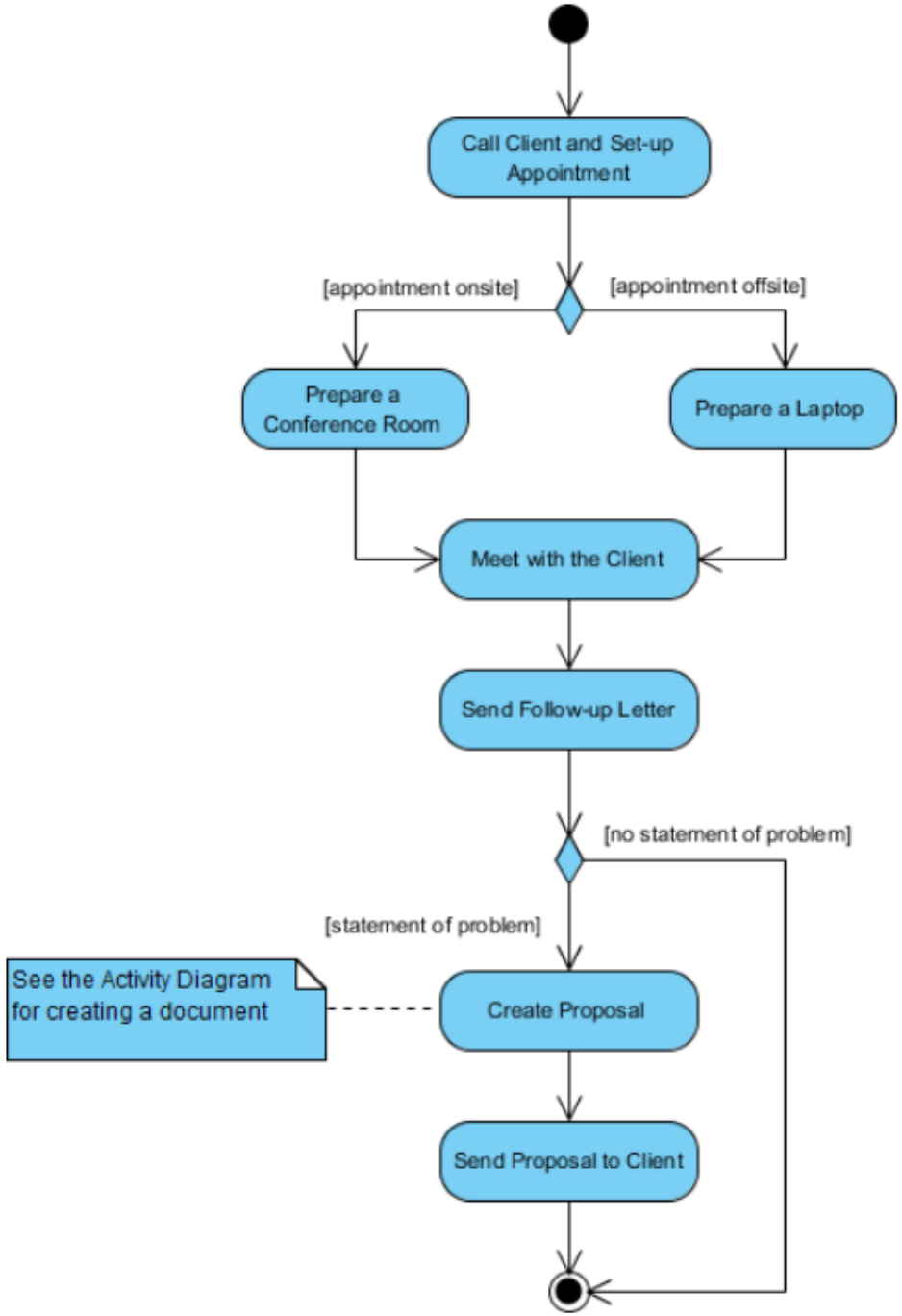
- Finally the parallel activities combine to close the order.

Behavior Models – Activity Diagram



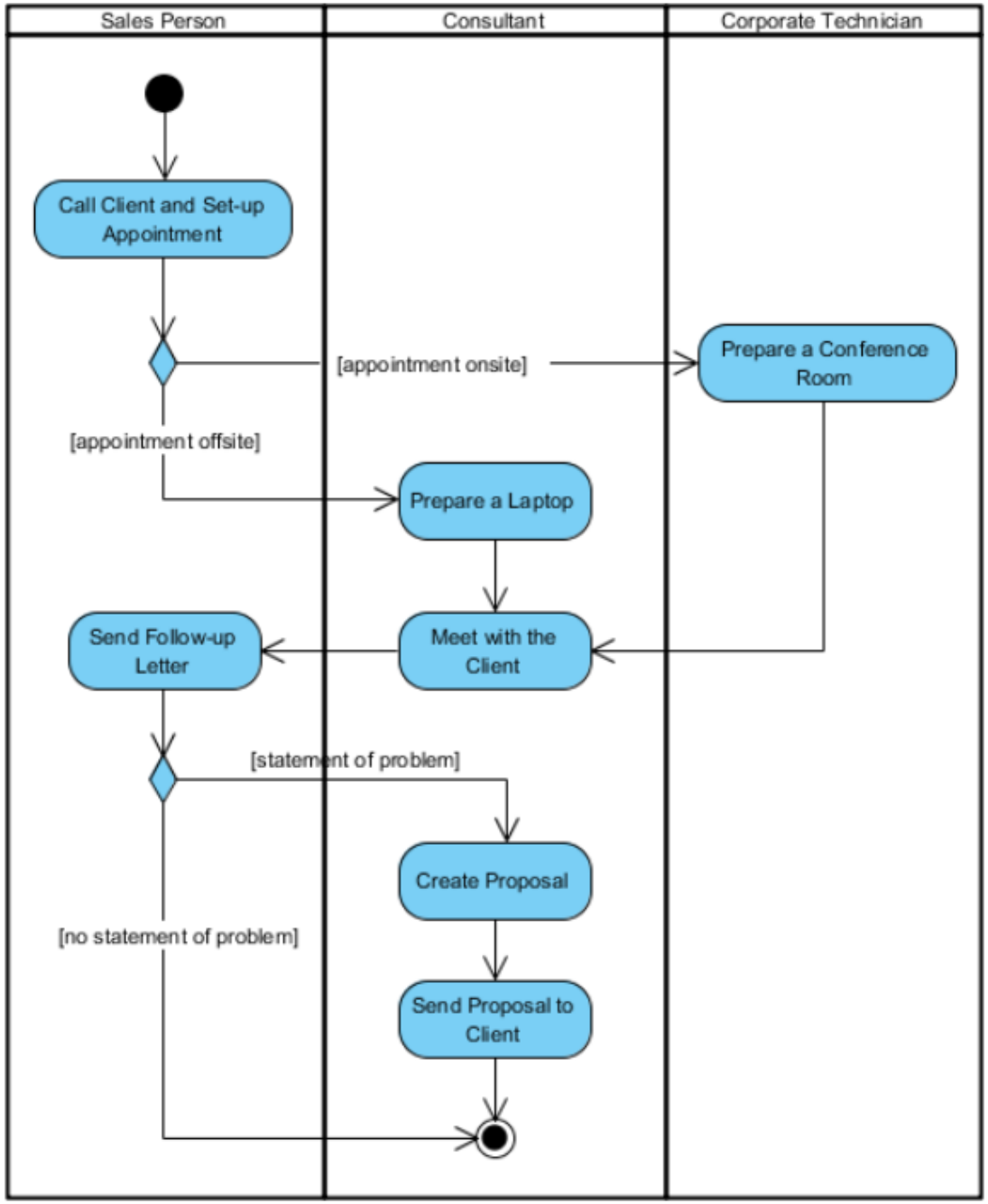
Behavior Models – Activity Diagram

- Activity diagram for meeting a new client



Behavior Models – Activity Diagram

- Swimlane
- A way to group activities performed by the same actor.



Behavior Models – Event Driven Modelling

- Real-time systems are often event-driven, with *minimal data processing*. For example, a landline phone switching system responds to events such as ‘receiver off hook’ by generating a dial tone.

- Event-driven modeling shows how a system responds to *external and internal events*.

- It is based on the assumption that a system has a *finite number of states* and that events (stimuli) may cause a *transition from one state to another*.

- *State machines* is the most widely used model for event driven scenarios.

Behavior Models – State Machines

- A *state machine diagram* shows the lifecycle of an object: what *events* it experiences, its *transitions*, and the *states* it is in between these events.
  - » Events may be internal or external.
  - » System states are 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.

Behavior Models – State Machines – Types of Events

- *External Event* – Events caused by an outside actor, outside of the system's boundary.
- *Internal Event* – Events caused by an internal action, inside the system's boundary.
- *Temporal Event* – Events caused by a determined date or time clock.

Behavior Models – State Machines – States

- If an object always responds the same way to an event, then it is considered *state-independent* (or *modeless*) wrt that event.

- By contrast, *state-dependent objects* react differently to events depending on their state or mode.

  » Telephone states

- **Guideline:** Consider state machines for state-dependent objects with complex behavior, not for state-independent objects, *why?*

Behavior Models – State Machines – States

- Physical Devices controlled by Software
  - » Microwave oven, Radio
- Transactions - The way transaction respond to events
  - » Cancel a sale
- Role Mutators, i.e., Objects that change their role.
  - » Full-time vs. half-time worker
- Communication Protocols
- UI Page/Window Flow or Navigation
- UI Flow Controllers or Stateful Sessions
  - » Server-side software objects
- Use Case System Operations
  - » For Process Sale: makeNewSale, enterItem, etc.
- Indivitual UI Window Event Handling
  - » Window's action "Edit-Copy"

Behavior Models – State Machines

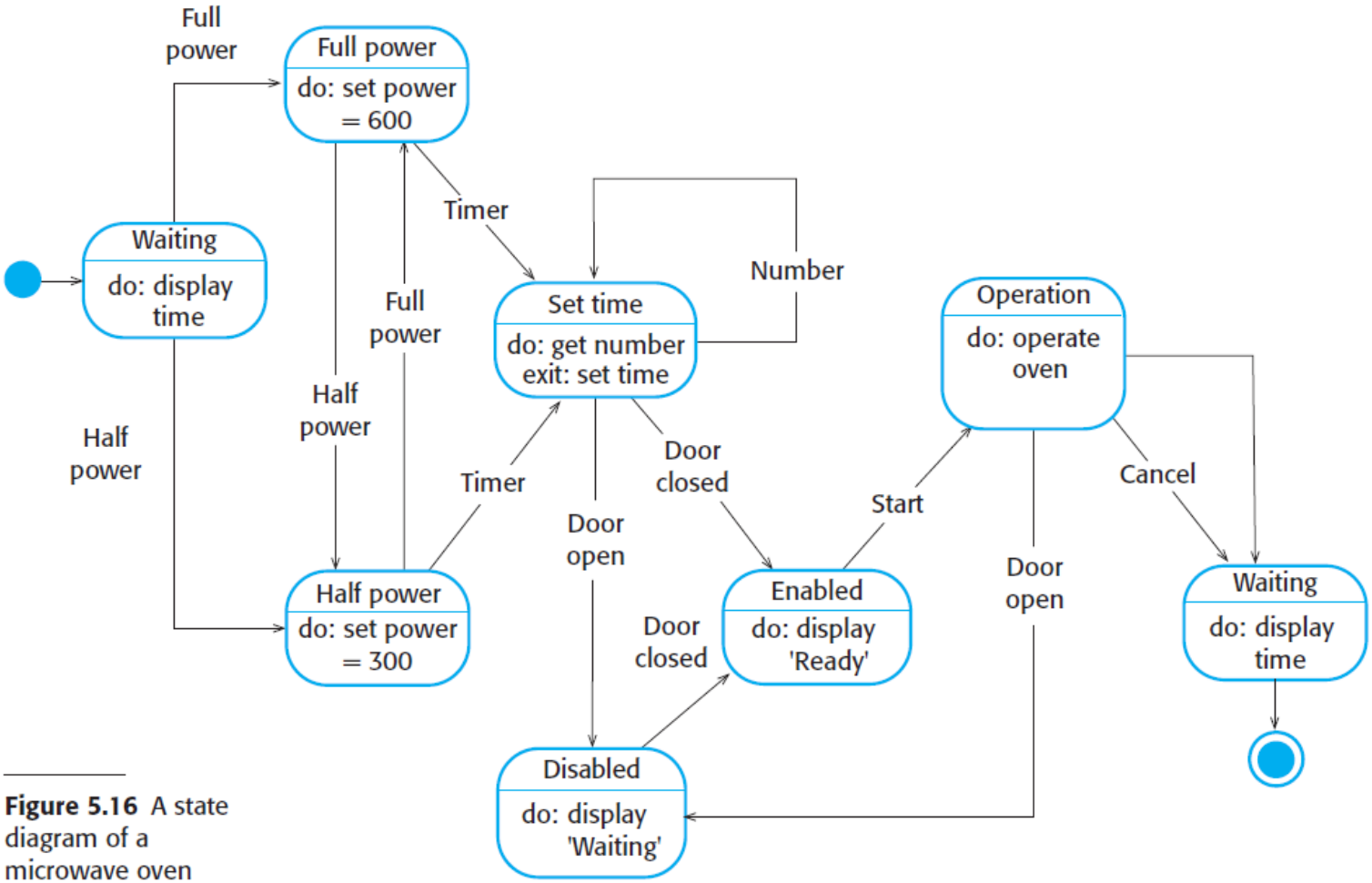


Figure 5.16 A state diagram of a microwave oven

Behavior Models – State Machines

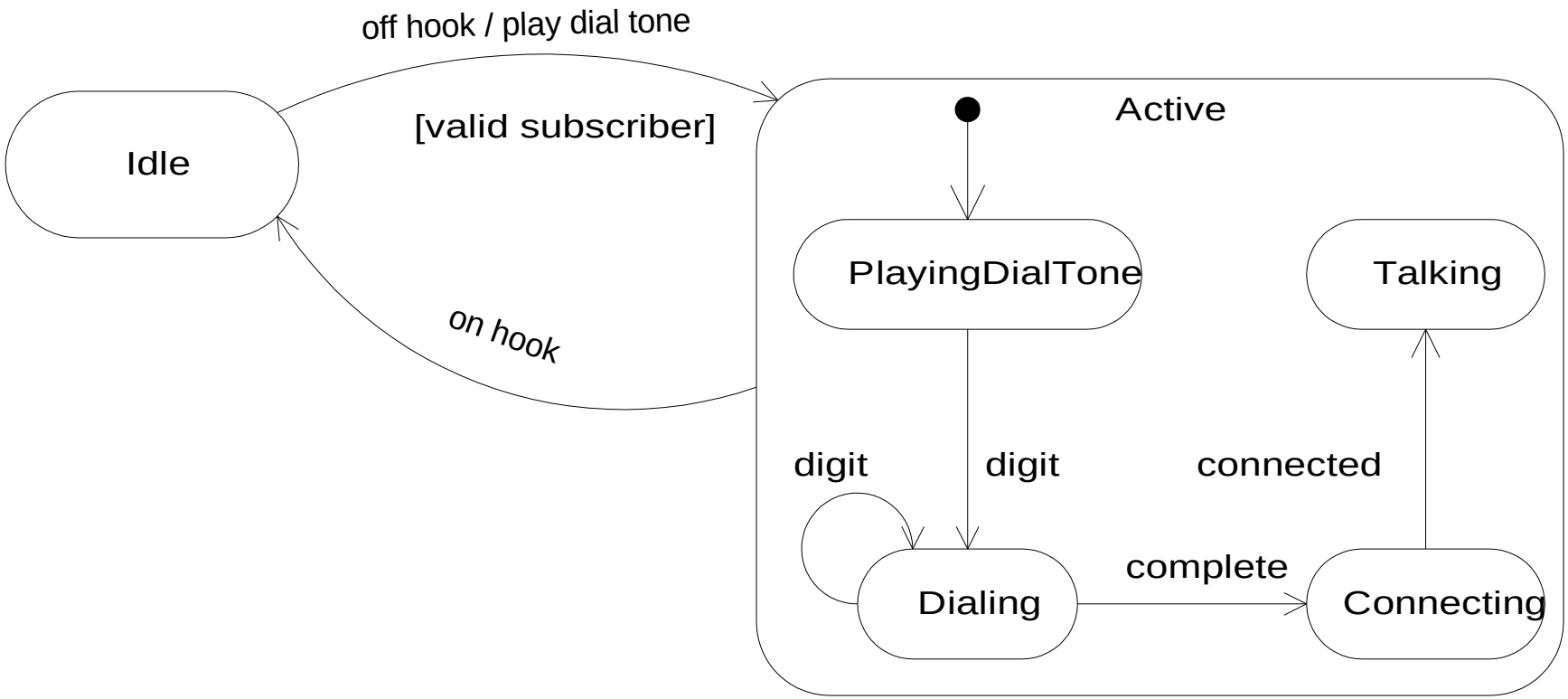
State	Description
Waiting	The oven is waiting for input. The display shows the current time.
Half power	The oven power is set to 300 watts. The display shows "Half power."
Full power	The oven power is set to 600 watts. The display shows "Full power."
Set time	The cooking time is set to the user's input 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 "Not ready."
Enabled	Oven operation is enabled. Interior oven light is off. Display shows "Ready to cook."
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 "Cooking complete" while buzzer is sounding.
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.

Behavior Models – State Machines – States

- A state may be represented as *nested* substates.
  - » In UML, substates are shown by nesting them in a superstate box.

- A substate inherits the transitions of its superstate.

Behavior Models – State Machines – Nested States



Behavior Models – State Machines – Nested States