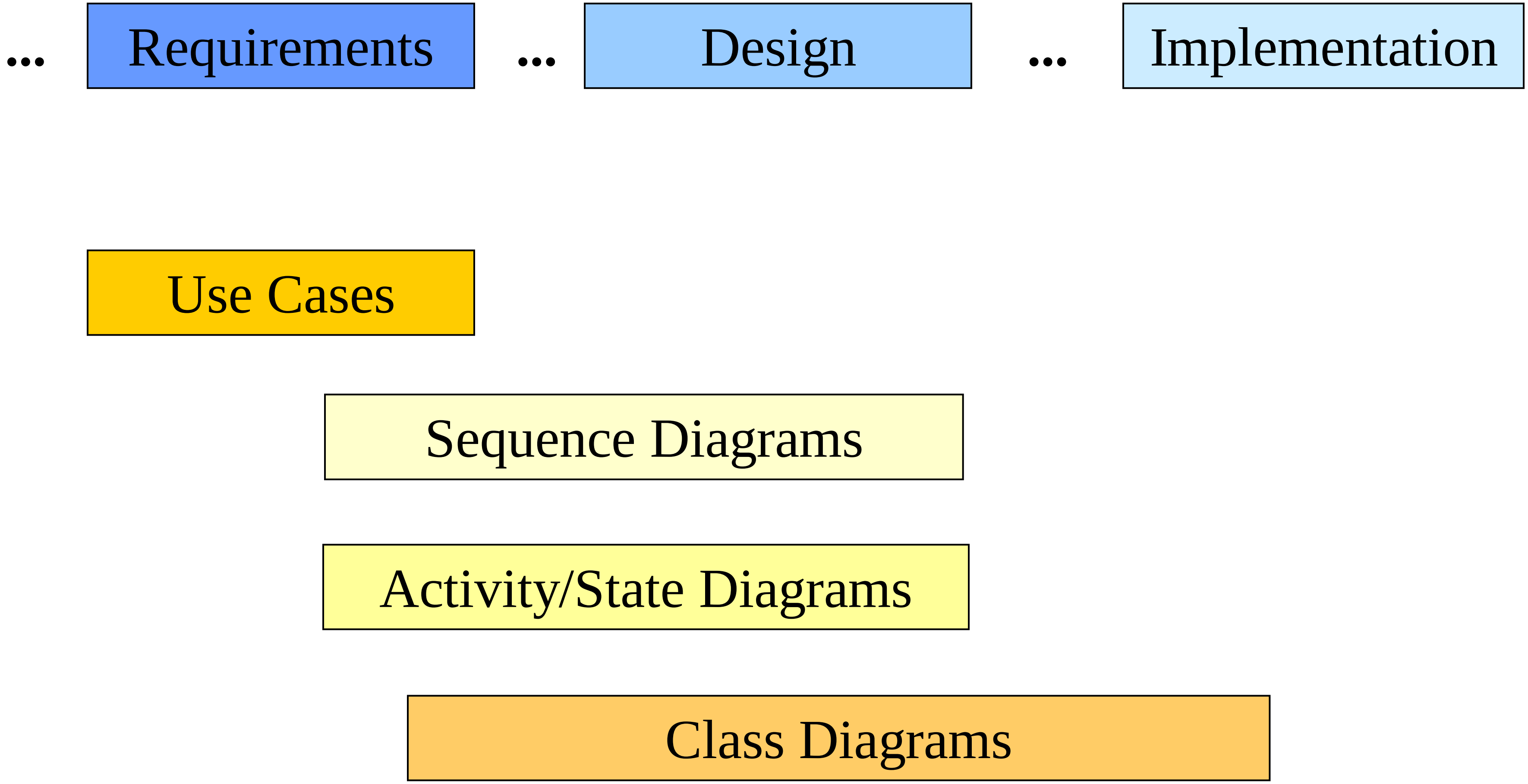
Software Engineering

UML Diagrams in the Process

UML Diagrams in the Process



… Requirements … Design … Implementation

Use Cases

Sequence Diagrams

Activity/State Diagrams

Class Diagrams

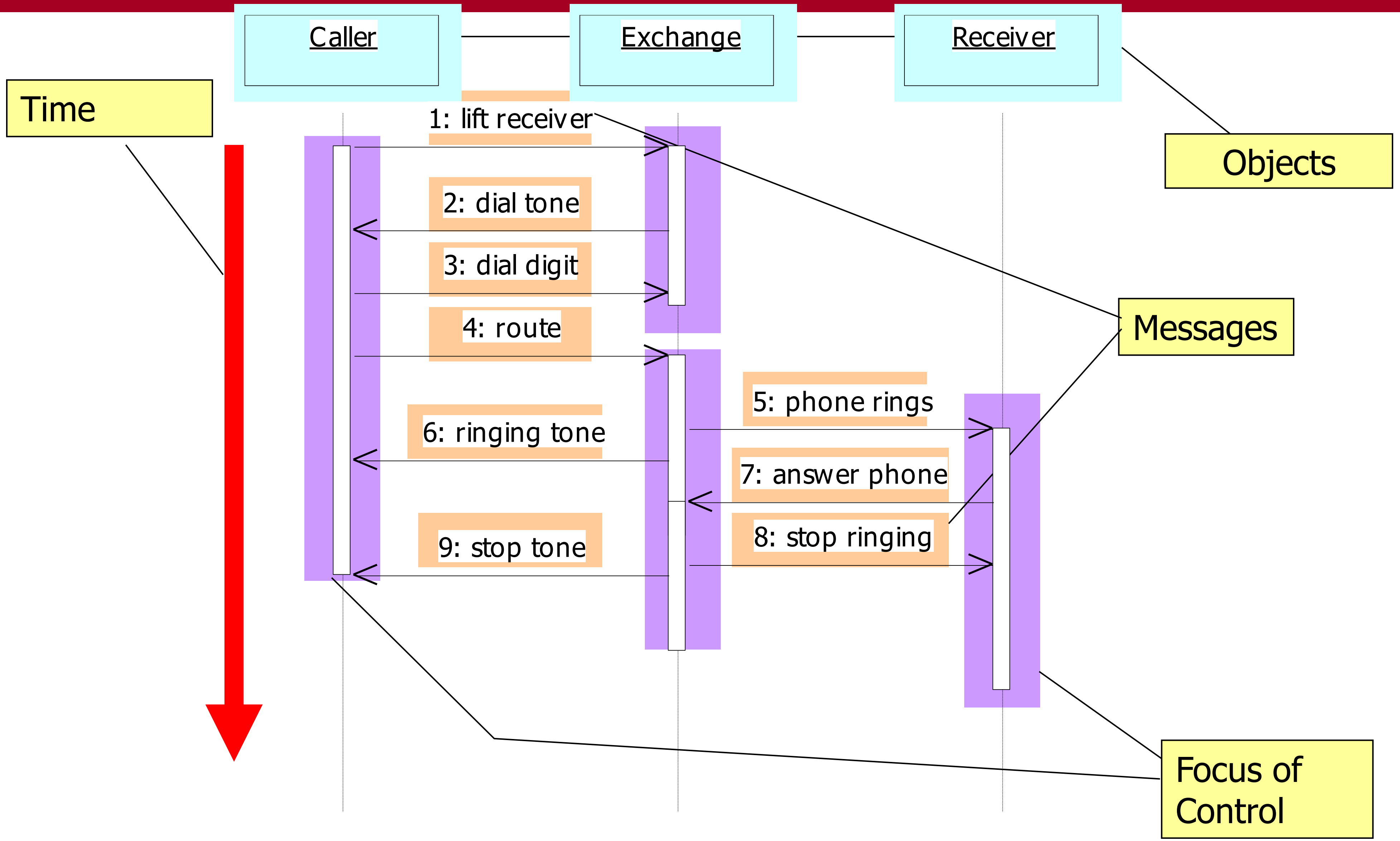
Software Engineering

Sequence Diagrams

Sequence Diagrams: Motivations

- Use Case Diagrams present an outside view of the system
- Sequence Diagrams are good at representing how objects collaborate.

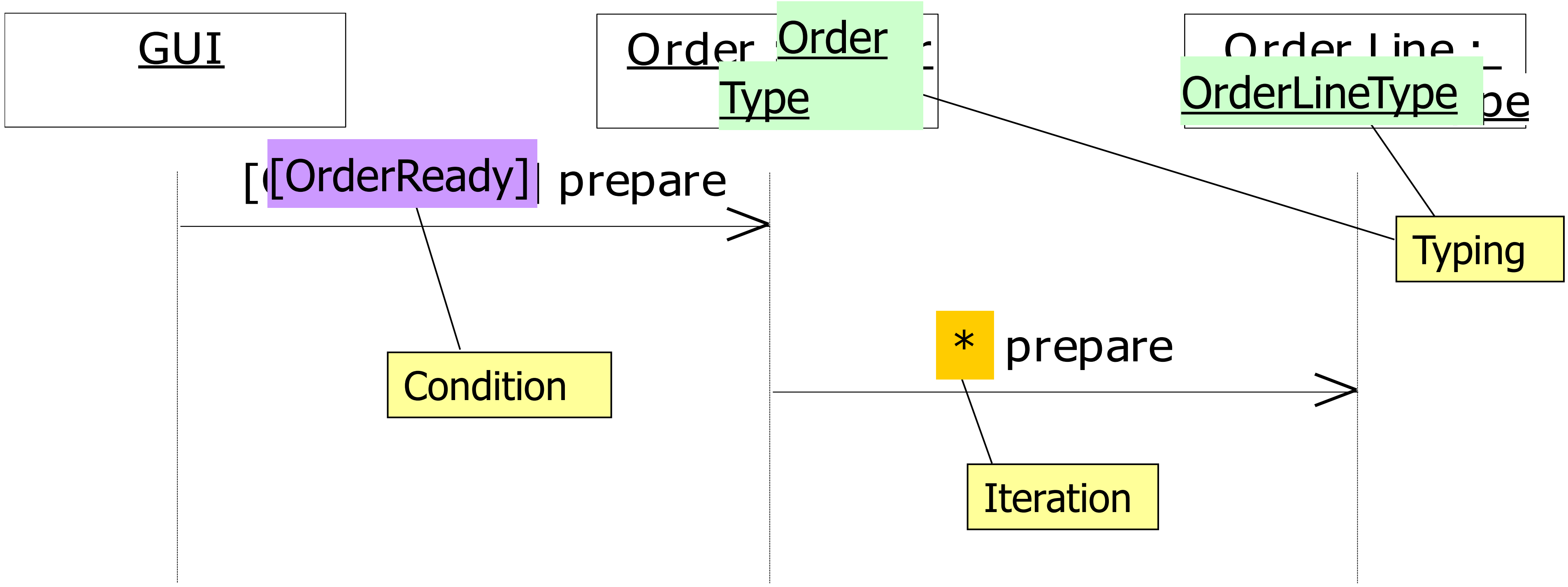
Sequence Diagram: Example



Sequence Diagrams: Ingredients

- Time: either unspecified or in fixed units
- Messages: a communication between two objects that conveys information with expectation that action will ensue
- Focus of Control: period during which an object is performing an action either directly or through a subordinate procedure.

Sequence Diagrams: More Concepts



GUI

Order Order Type

Order Line : OrderLineType

[OrderReady] prepare

Condition

Typing

* prepare

Iteration

Sequence Diagrams: More Concepts

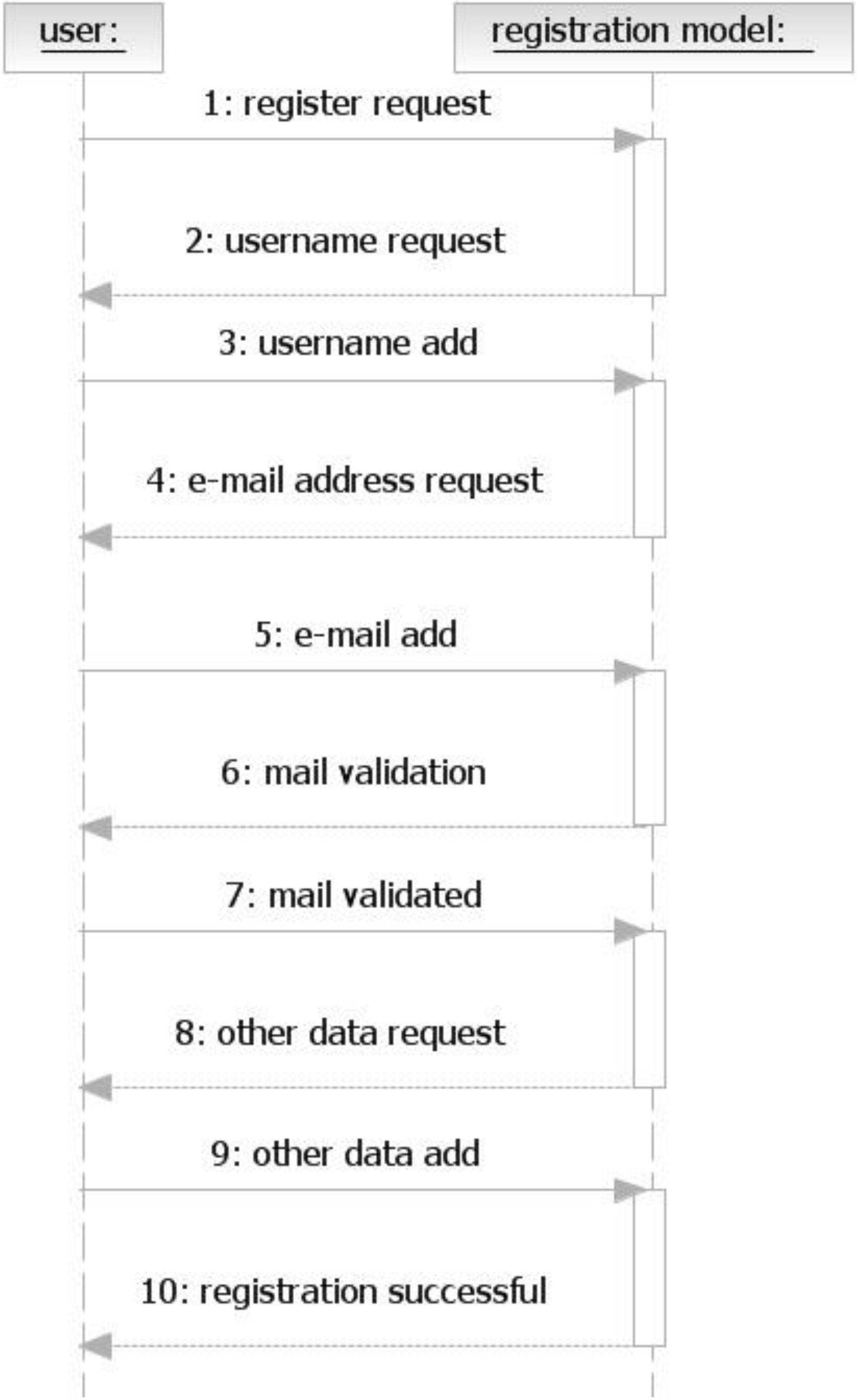
- Typing: define the types of objects
- Condition: define when a message is sent
- Iteration: message sent to all the instances of an object

- Focus Of Control: is optional!
- Message Numbering: is optional!

Sequence Diagrams vs. Use Case Diag.

Objects	Actors
Messages	Actions / Use Cases

Sequence Diagram of a registration



Software Engineering

State Machine and Activity Diagrams

Introduction

Quite often during both the specification and the development process you will be faced with the necessity of representing dynamic behaviors

Examples:

- Modeling workflows
- Modeling complex use cases
- Modeling lifecycle of objects

UML provides various diagrams for modeling dynamic behaviors, among which

- State Machine diagram
- Activity diagram

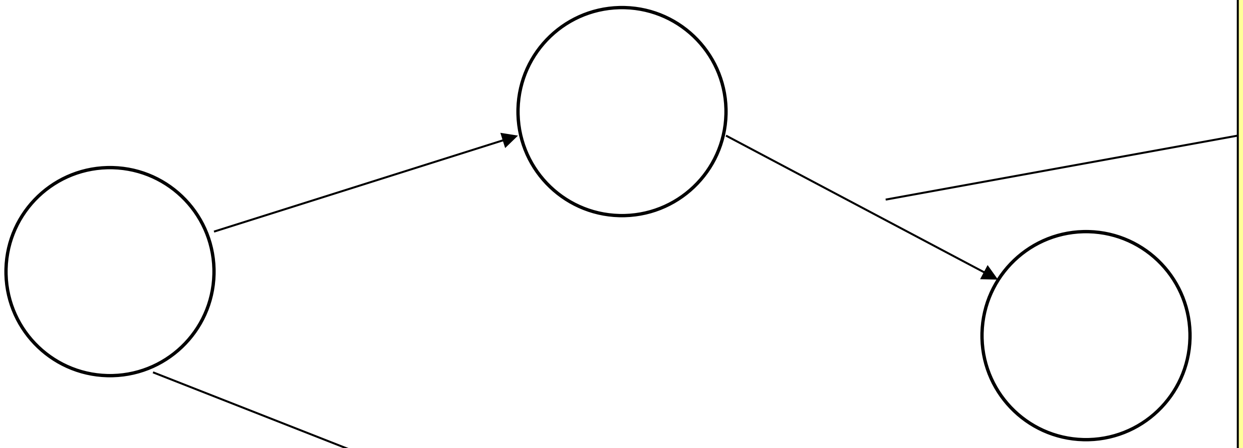
State Machine and Activity Diagram

State Diagram are closely related to:

- Finite State Automaton
- Petri Nets
- David Harel’s Statecharts

Activity diagrams can be seen a special types of State Diagram

Finite State Automata



Transitions

Describe how systems/algorithms/... change from state to state.

Examples.

- after(15sec) (Time based)
- on_red (Event Based)
- ...

States

Represent “conditions” in which a system/algorithm/... can be found.

Examples.

- “Ready”, “Busy”, ...

Initial States

Accepting States

Finite State Automata (ctd)

Application of FSA is pervasive in Computer Science

Examples:

- Lexical Checker
- Formal Verification
- ...

Software Engineering

State Machine Diagram

State Machine Diagram

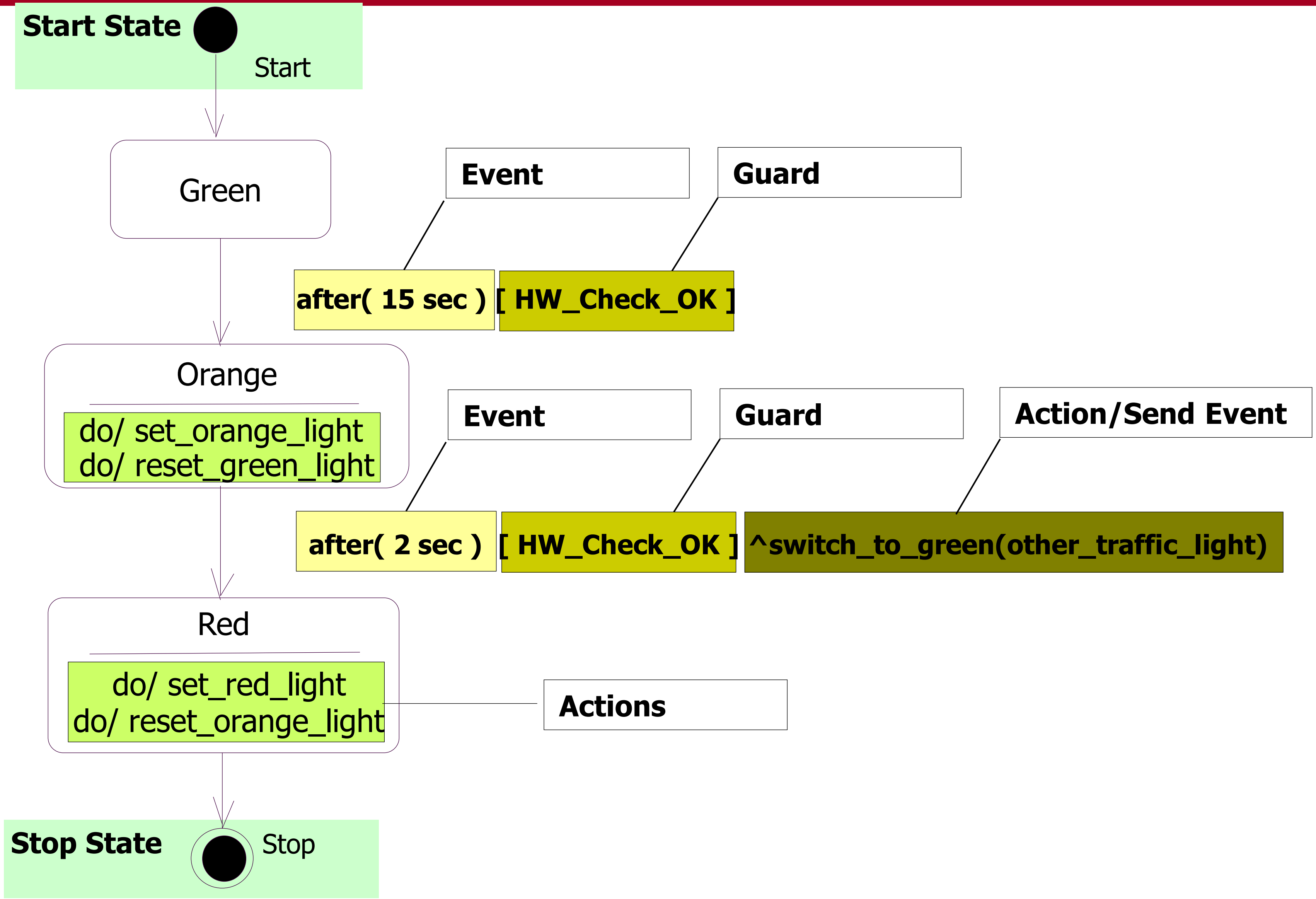
Definition:

A State Machine Diagram describes the sequences of states and actions through which an element (e.g. of the design) can proceed during its lifetime as a result to reacting to dicrete events.

Key Ingredients (… not surprisingly!):

- States (simple and composite)
- Transitions (event, guard, actions)

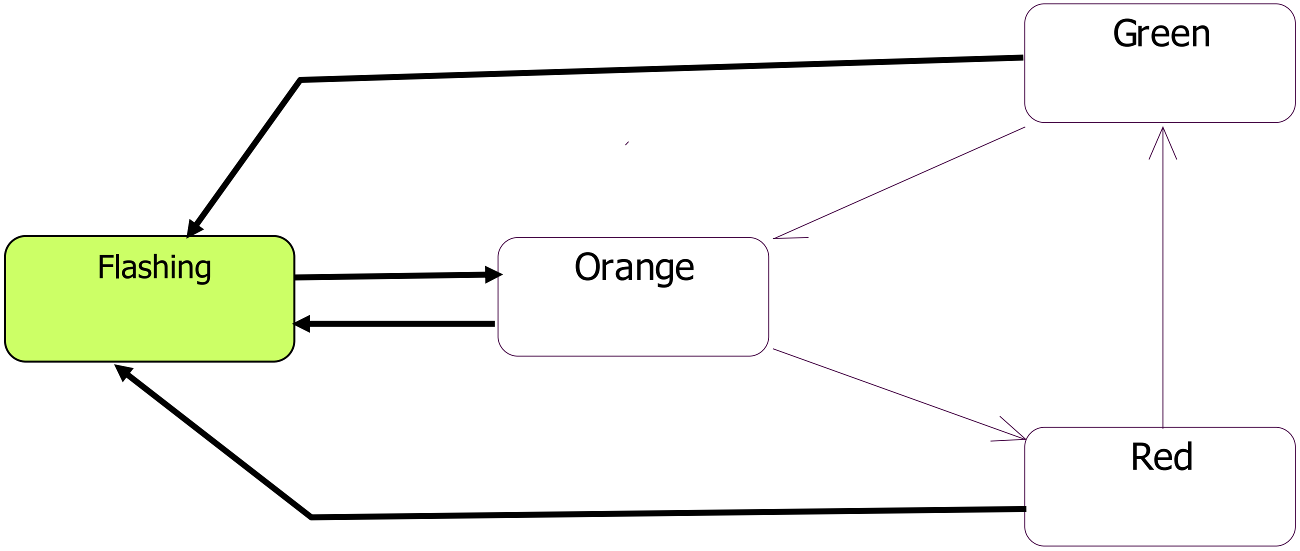
State Machine Diagram: Example



State diagram: Explanation

- Start State and Stop State: obvious
- Transition
  - **Event:** what may cause the transition
  - **Guard:** upon “event”, if true allows transition, if false prevents transition
  - **Action/Send Event:** if transition, then execute action/send event
- Actions within a state
  - **entry/** do action when entering the state
  - **do/** keep doing action while in state
  - **exit/** do action on leaving state

States: Hierarchical Decomposition



Issues:

- Cluttering of the diagram
- Structural information is lost

Software Engineering

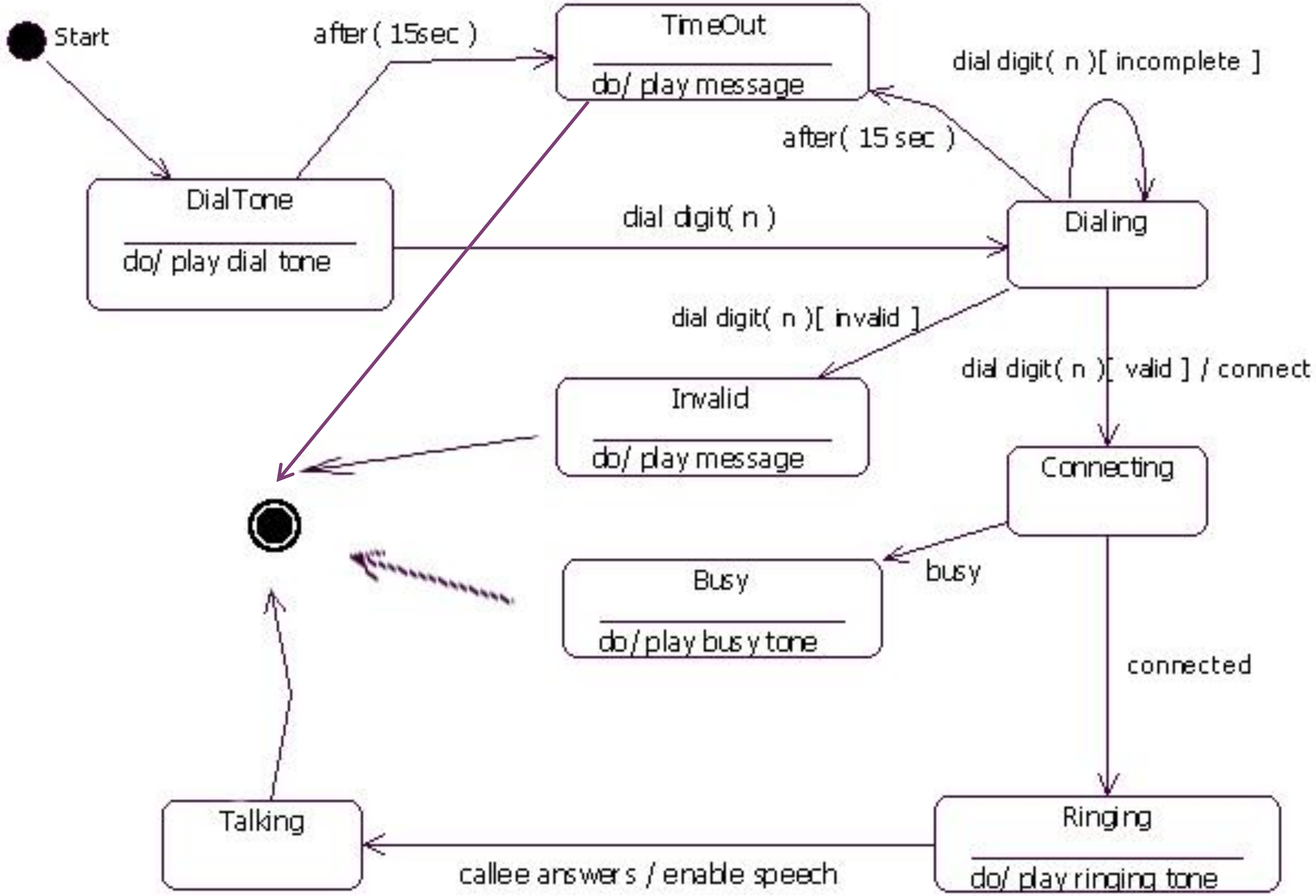
State Diagram: example

State Diagram: example

Phone Dialing

Specify, using statecharts, the process of placing a call with a telephone (e.g. when you lift the receiver, a dial tone is activated; then you can either …).

StateChart: example



Software Engineering

Activity Diagram

Activity Diagram

Definition

An activity diagram is a variation of a statechart in which the states represent the performance of actions and the transitions are triggered by the completion of the actions.

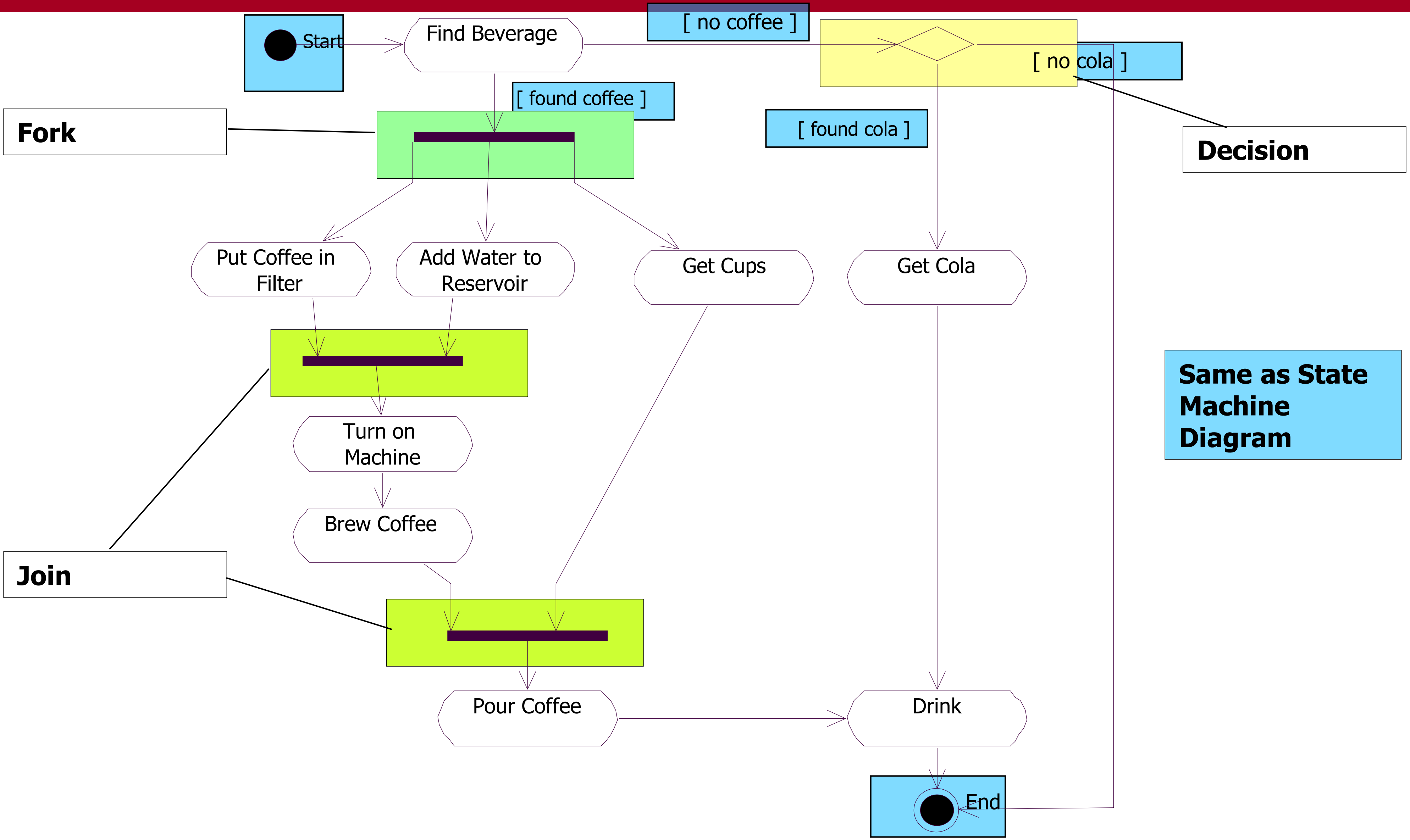
Key Ingredients

- Activities (States)
- Transitions (event, guard, actions)
- Merge/Fork/Join

Software Engineering

Activity Diagram: example

Activity Diagram: example



Same as State Machine Diagram

Activity Diagram: Explanation

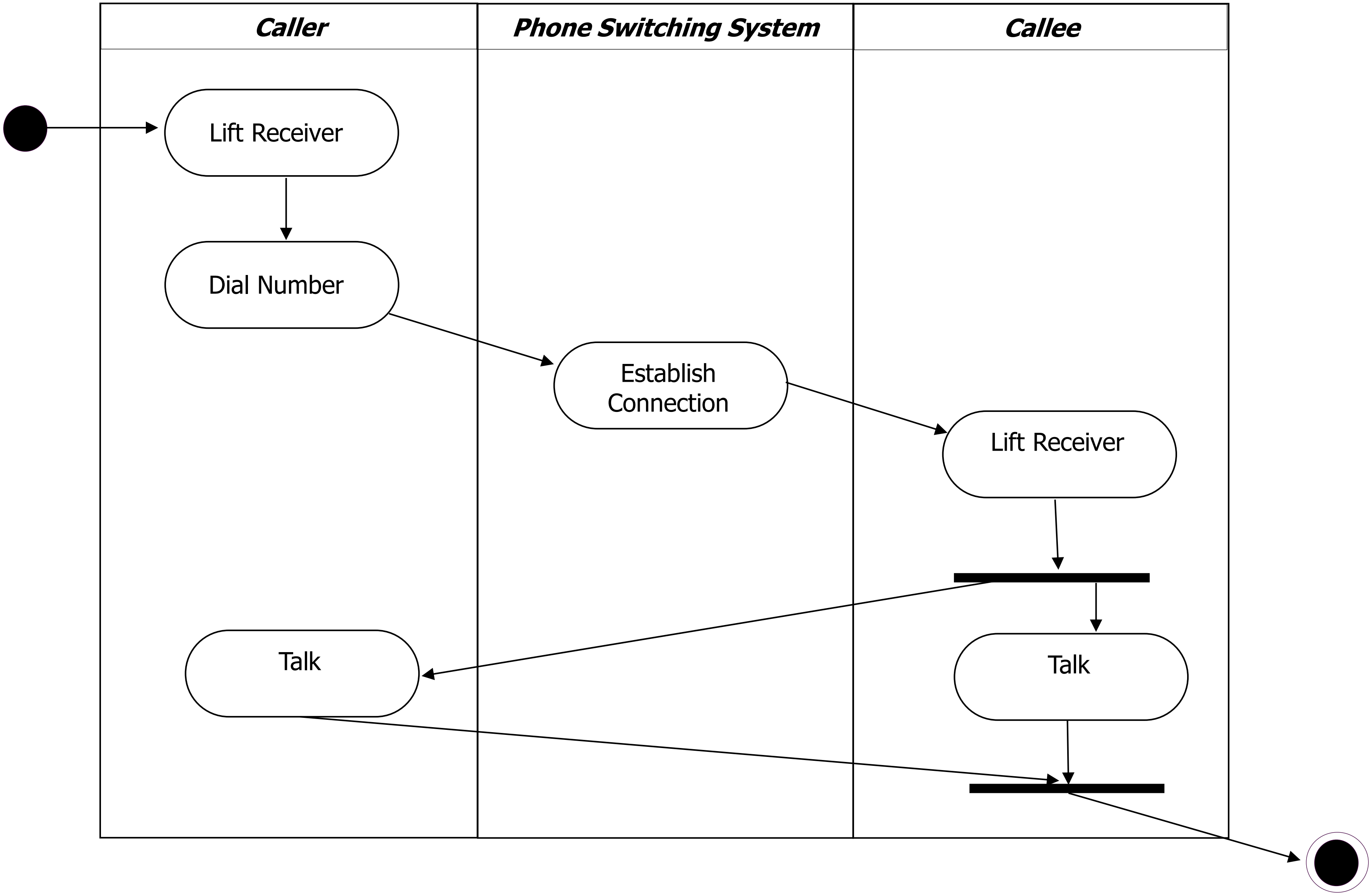
- Start/Stop/Activity/Guard: obvious
- Fork: allows for the parallel decomposition of activities.
- Join: synchronises parallel activities.
- Decision: conditional behavior.

Swimlanes

Sometime it is useful to highlight responsibilities in activity diagrams

Activities diagrams allow to do that with “swimlanes”

Swimlanes: Example



Sequence Diagrams vs Activity Diag.

- Sequence Diagram: description of **the whole system**

- Activity Diagram: description of **a part of the system** (usually an Use Case)

State and Activity Diagrams

Use them when really necessary!!!

(e.g. complex interactions, (in)formal verification,
business process definition and enhancement,
…)