

Chapter 8

Dynamic Modeling

Statechart and Activity

(Textbook Chapter 5)



Course instructor : TA.Nada Alamoudi

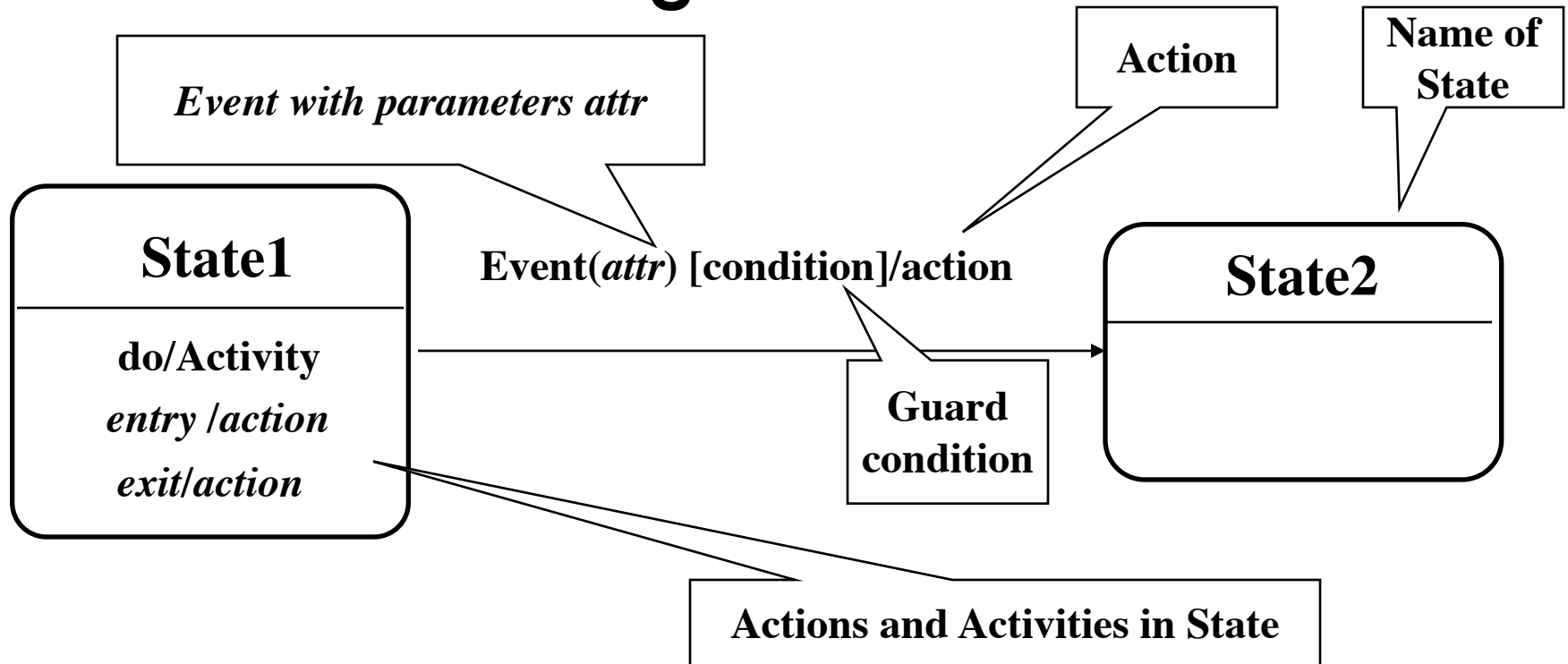
Dynamic Modeling with UML

- Two UML diagrams types for dynamic modeling:
 - **Interaction diagrams** describe the dynamic behavior *between* objects
 - **State chart diagrams** describe the dynamic behavior *of a single object*:
 - **State Chart Diagram**: A state machine that describes the response of an object of a given class to the receipt of outside stimuli (Events).
 - **Activity Diagram**: A special type of state chart diagram, where all states are action states (Moore Automaton).

Dynamic Modeling

- We distinguish between two types of operations:
 - **Activity**: Operation that takes time to complete
 - associated with states
 - **Action**: Instantaneous operation
 - associated with events
- A state chart diagram relates events and states for one class
- An object model with several classes with interesting behavior has *a set* of state diagrams

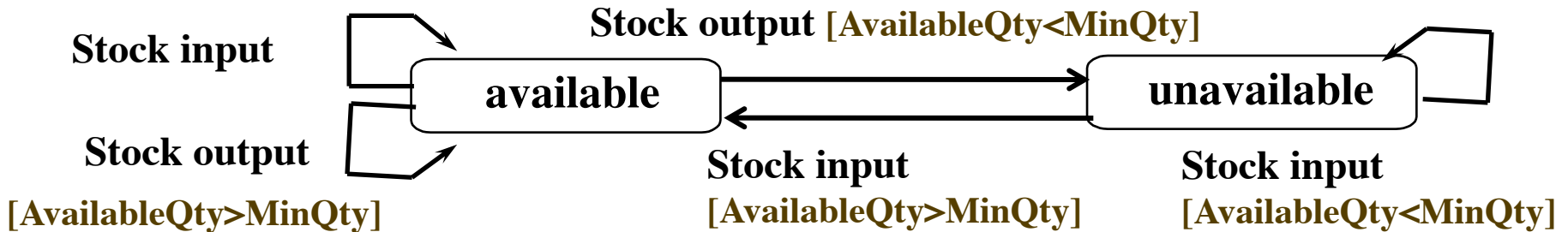
UML Statechart Diagram Notation



- Note:

- Conditions are enclosed with brackets: []
- Actions and activities are prefixed with a slash /

Example of a StateChart Diagram




State Chart Diagram vs Sequence Diagram

- State chart diagrams help to identify:
 - Changes to an individual object over time
- Sequence diagrams help to identify:
 - The temporal relationship between objects over time
 - Sequence of operations as a response to one or more events.

Dynamic Modeling of User Interfaces

- Statechart diagrams can be used for the design of user interfaces
- States: Name of screens
- Actions or activities are shown as bullets under the screen name

Requirements Analysis Document Template

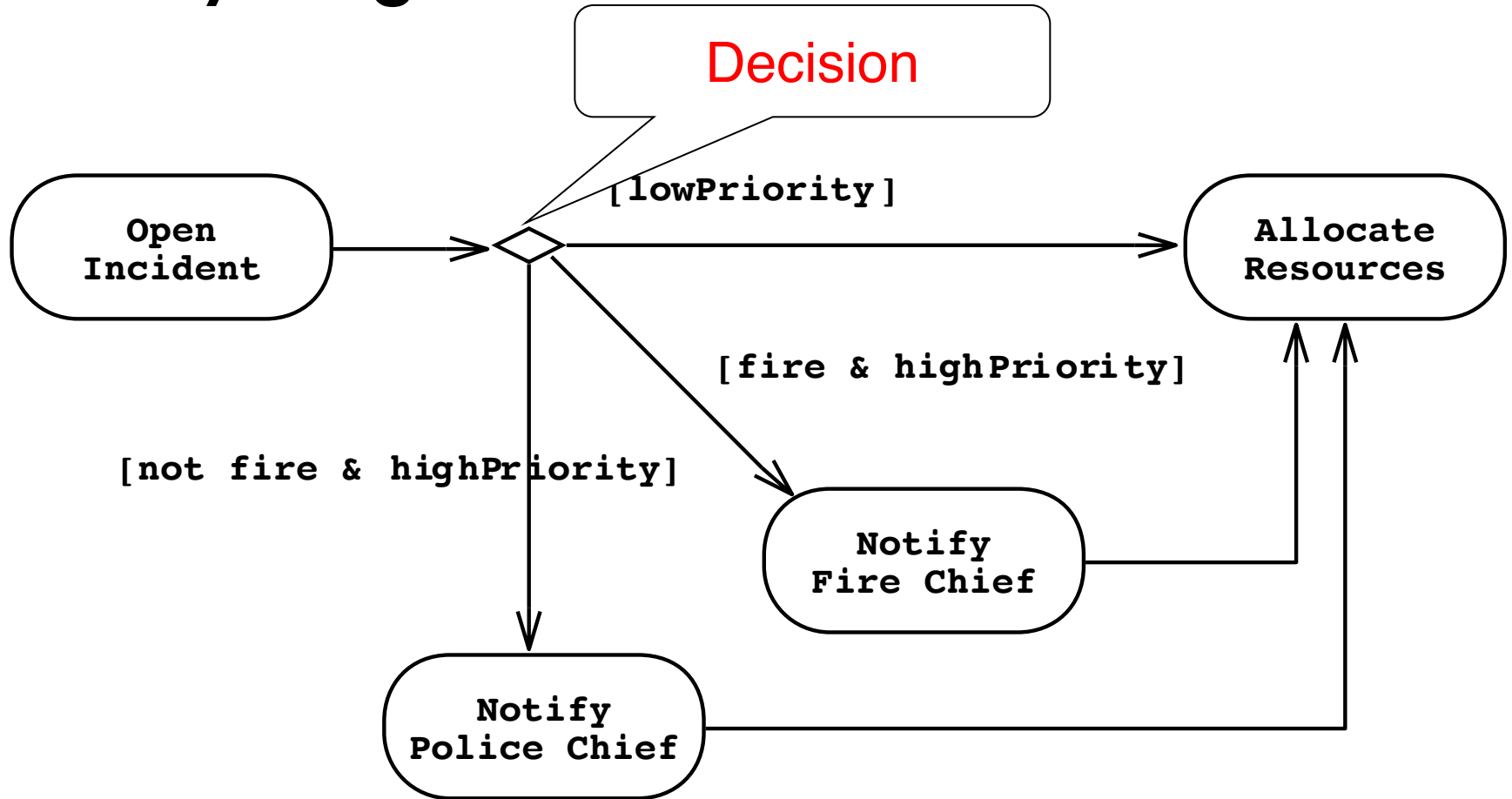
1. Introduction
2. Current system
3. Proposed system
 - 3.1 Overview
 - 3.2 Functional requirements
 - 3.3 Nonfunctional requirements
 - 3.4 Constraints ("Pseudo requirements")
 -  3.5 System models
 - 3.5.1 Scenarios
 - 3.5.2 Use case model
 - 3.5.3 Object model
 - 3.5.3.1 Data dictionary
 - 3.5.3.2 Class diagrams
 - 3.5.4 Dynamic models
 - 3.5.5 User interface
4. Glossary

Activity Diagrams

- An activity diagram is a special case of a statechart diagram
- The states are activities (“functions”)
- An activity diagram is useful to depict the workflow in a system

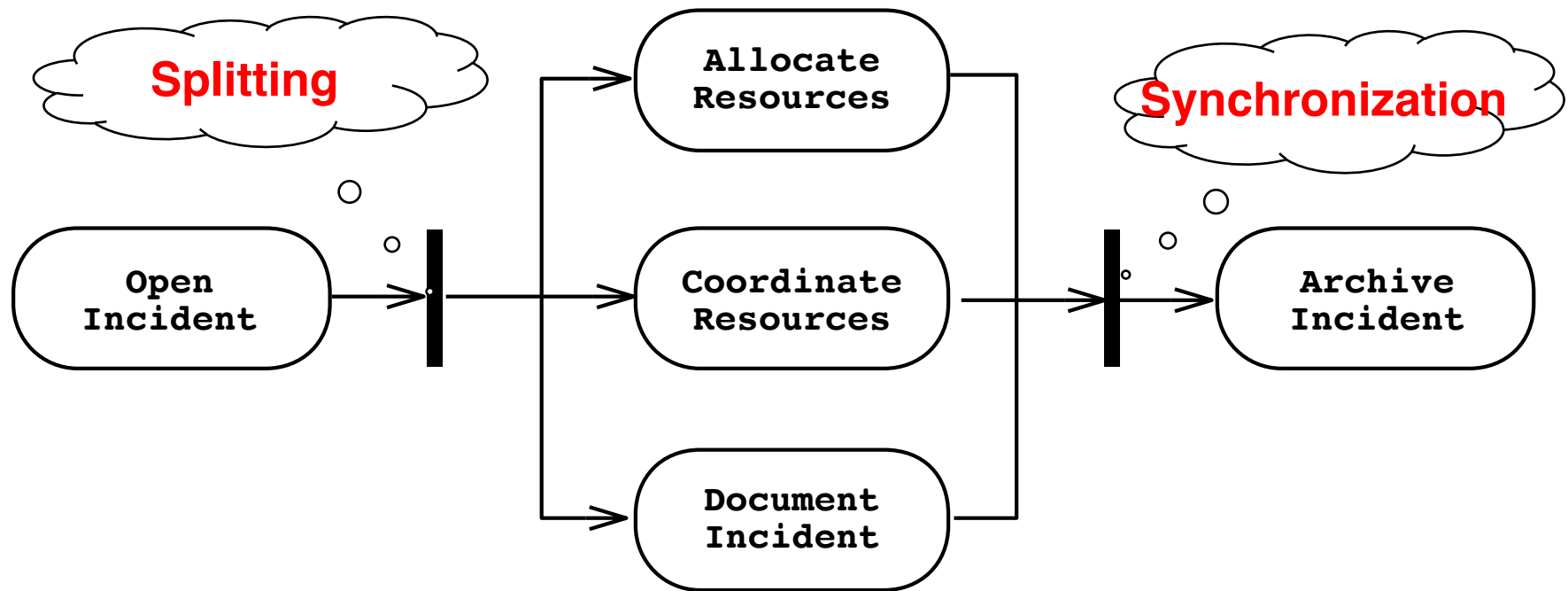


Activity Diagrams allow to model Decisions



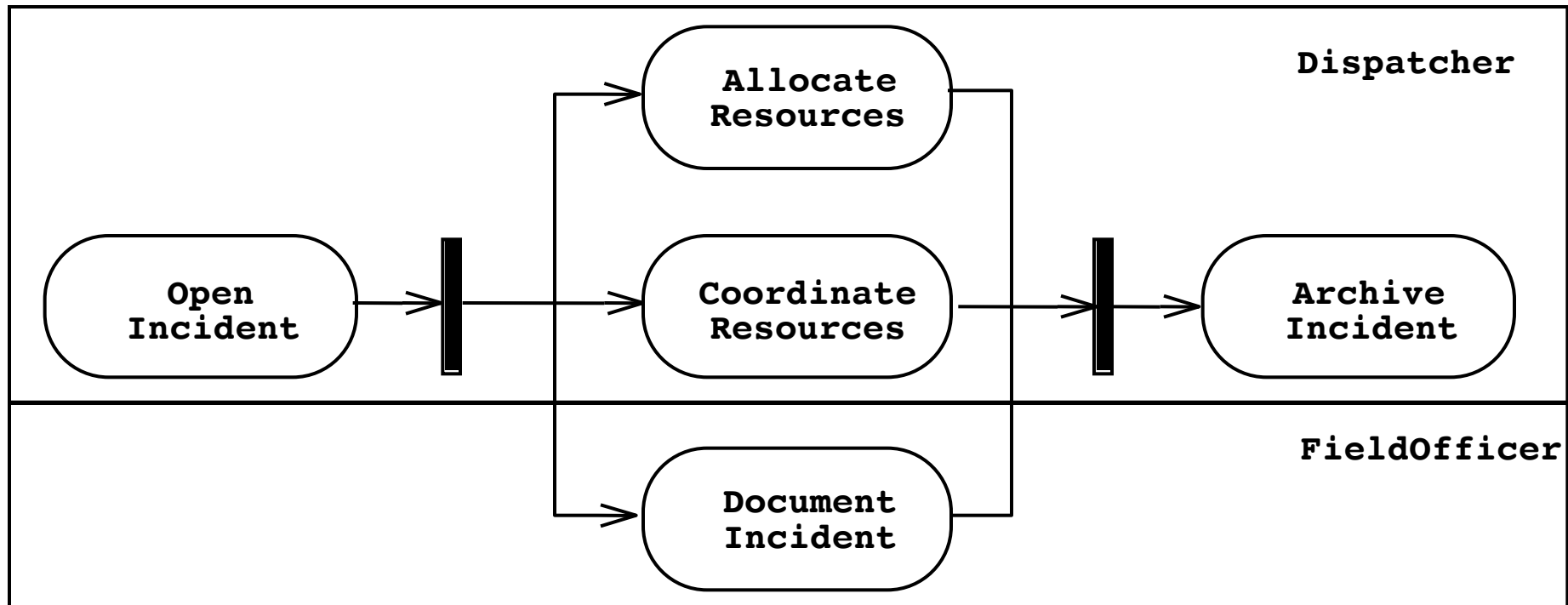
Activity Diagrams can model Concurrency

- Synchronization of multiple activities
- Splitting the flow of control into multiple threads



Activity Diagrams: Grouping of Activities

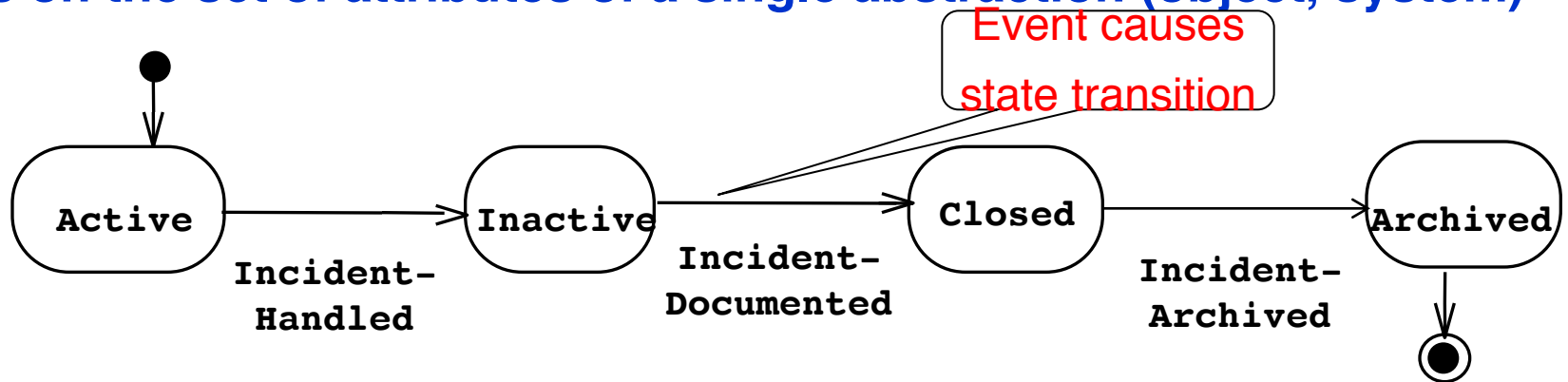
- Activities may be grouped into **swimlanes** to denote the object or subsystem that implements the activities.



Activity Diagram vs. Statechart Diagram

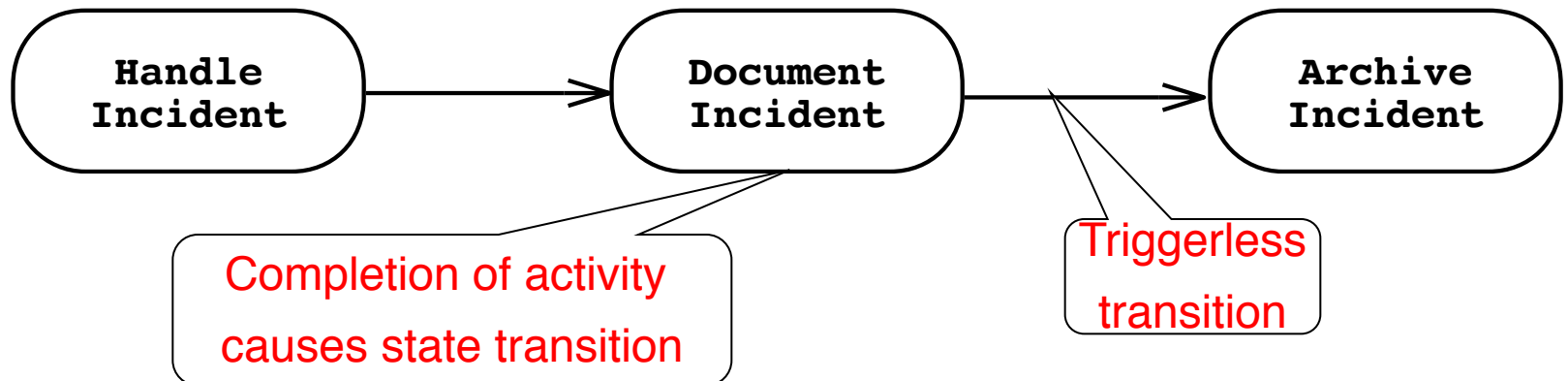
Statechart Diagram for Incident

Focus on the set of attributes of a single abstraction (object, system)



Activity Diagram for Incident

(Focus on dataflow in a system)



Model Validation and Verification

- **Verification** is an equivalence check between the transformation of two models
- **Validation** is the comparison of the model with reality
 - Validation is a critical step in the development process: Requirements should be validated with the client and the user.
 - Techniques: Formal and informal reviews (Meetings, requirements review)
- **Requirements validation** involves several checks
 - Correctness, Completeness, Ambiguity, Realism

Checklist for a Requirements Review

- Is the model correct?
 - A model is correct if it represents the client's view of the the system
- Is the model complete?
 - Every scenario is described
- Is the model consistent?
 - The model does not have components that contradict each other
- Is the model unambiguous?
 - The model describes one system, not many
- Is the model realistic?
 - The model can be implemented