INTRODUCTION TO UML

Lotfi ben Othmane

Iowa State University

Fall 2017

Some slides are taken from a lecture of Majid Ali Khan

Spring 2005

Several diagrams are taken from UML Distilled
By Martin Fowler

GOAL

Learn what is UML and

how to use the diagrams that describe the architecture

MODELING

- Describing a system at a high level of abstraction
- Is it necessary to model software systems?

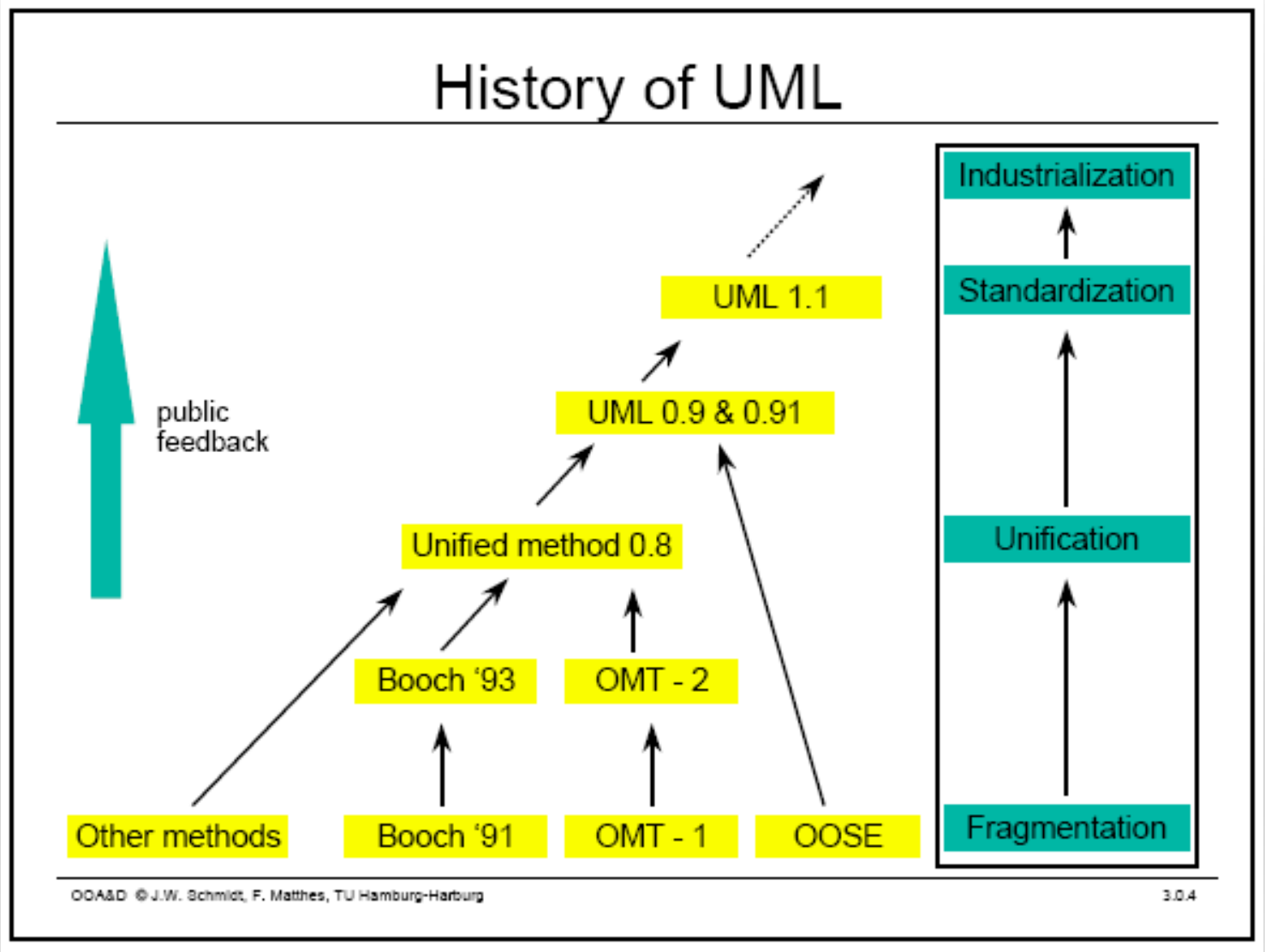
WHAT IS UML?

- UML stands for "Unified Modeling Language"
- It is a industry-standard graphical language for specifying, visualizing, constructing, and documenting the artifacts of software systems
- The UML uses mostly graphical notations to express the OO analysis and design of software projects
- Simplifies the complex process of software design

WHY UML FOR MODELING?

- Use graphical notation to communicate more clearly than natural language (imprecise) and code(too detailed)
- Help acquire an overall view of a system
- UML is *not* dependent on any one language or technology
- UML moves us from fragmentation to standardization

HISTORY OF UML



TYPES OF UML DIAGRAMS

Most used diagram types:

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Components diagram
- Collaboration Diagram
- State Diagram
- Activity diagram
- Deployment diagram

USE CASE DIAGRAM

- Mainly used to capture user requirements
- Provides an external view of the system
- Used to describe user scenarios
- Capture a generalized description of how a system will be used.
- Provides an overview of the intended functionality of the system

LEVELS OF USE CASES

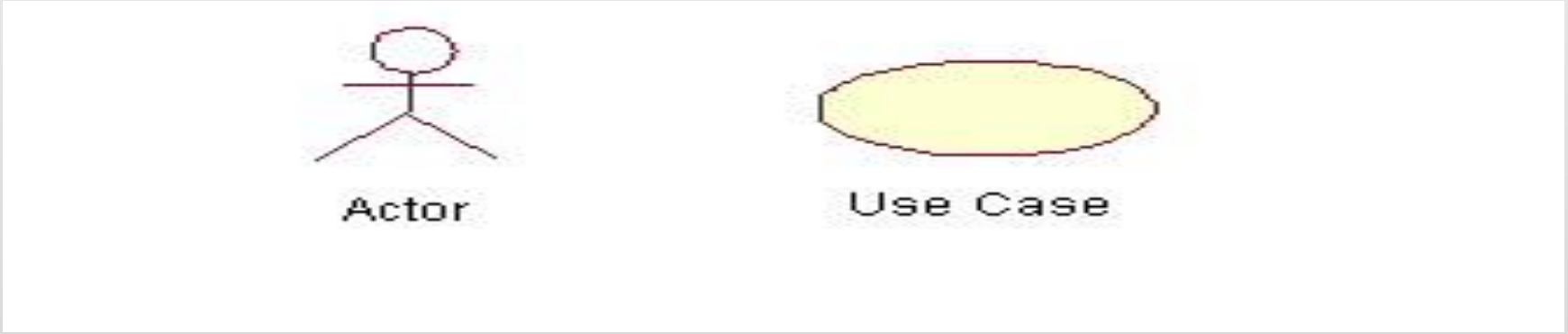
- Goal of use case: How the actors use the system (from customer perspective)

- System use case –interactions with the system
- Business use cases – how business responds to events

- Feature set shall not be mixed with use cases
  - A feature could be a use case, a set in a use case, or a variant behaviors

USE CASE DIAGRAM

- Actors: A role that a user plays with respect to the system.
- Use case: A set of scenarios that describe the interactions of the actors with the system.

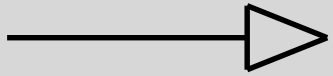


- System boundary: Rectangle shape representing the boundary of the system.

USE CASE DIAGRAM

- Association: communication between an actor and a use case.

- Generalization: relationship between a general use case and a special use case

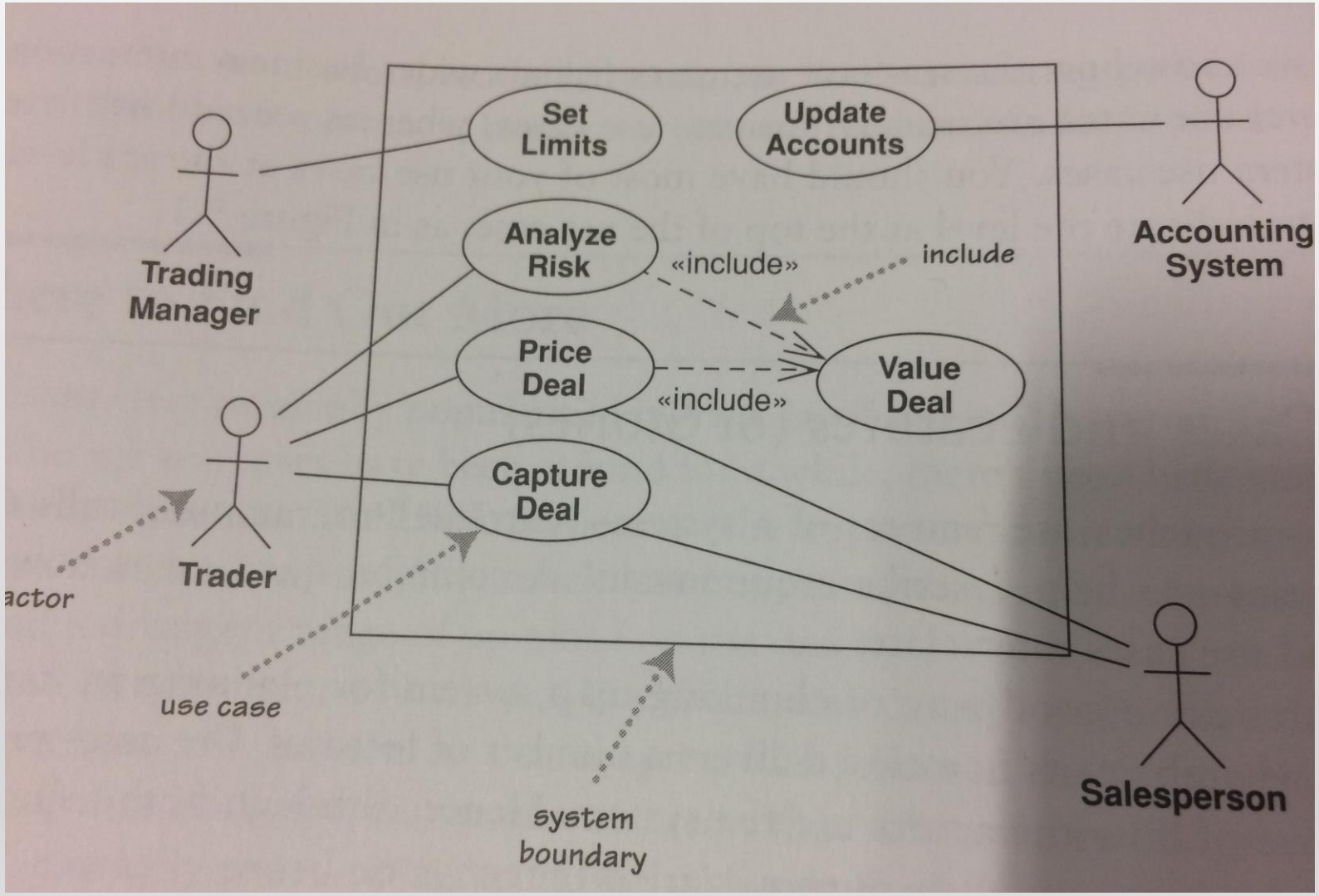


USE CASE DIAGRAM

- Include: a chunk of behavior is similar across more than one use case.

- Extend: use case add behavior to the base use case.

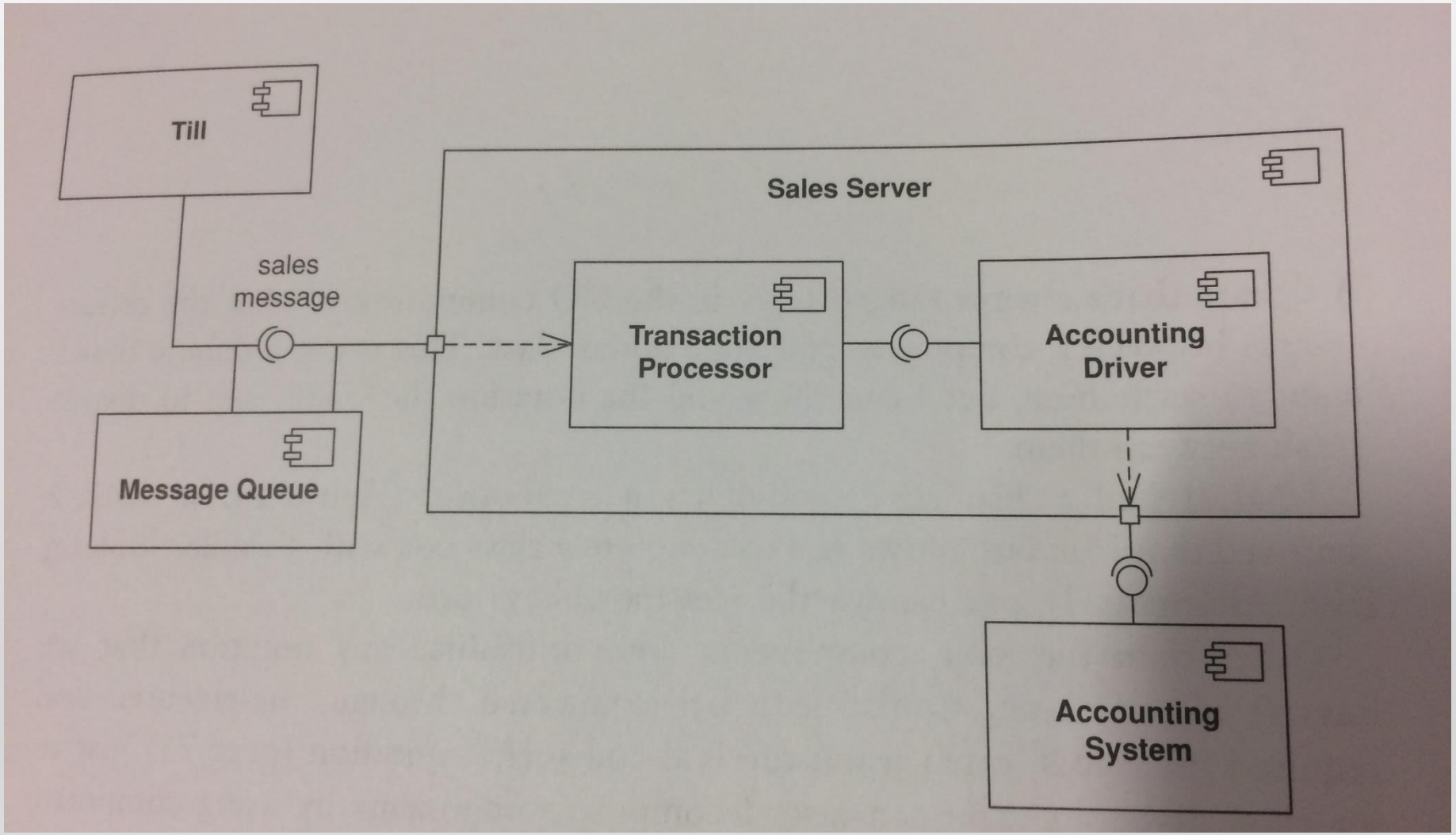
EXAMPLE OF USE CASE



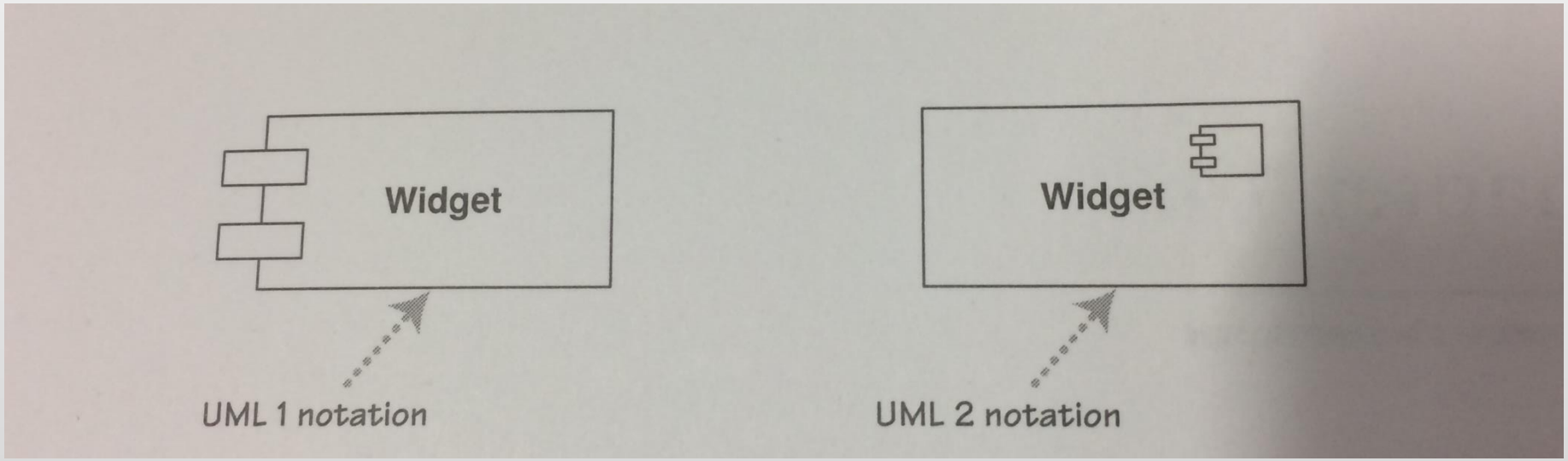
COMPONENT DIAGRAM

- Components are independent pieces that compose the software from customers perspectives
  - Customers can upgrade each component separately
  - Old components can work with new components seamlessly
  - Support mix and match components of different providers
- A component could be a class or a set of classes
- Your goal is to identify the interfaces of and data exchanges among the independent components of software

COMPONENT DIAGRAM



COMPONENT DIAGRAM

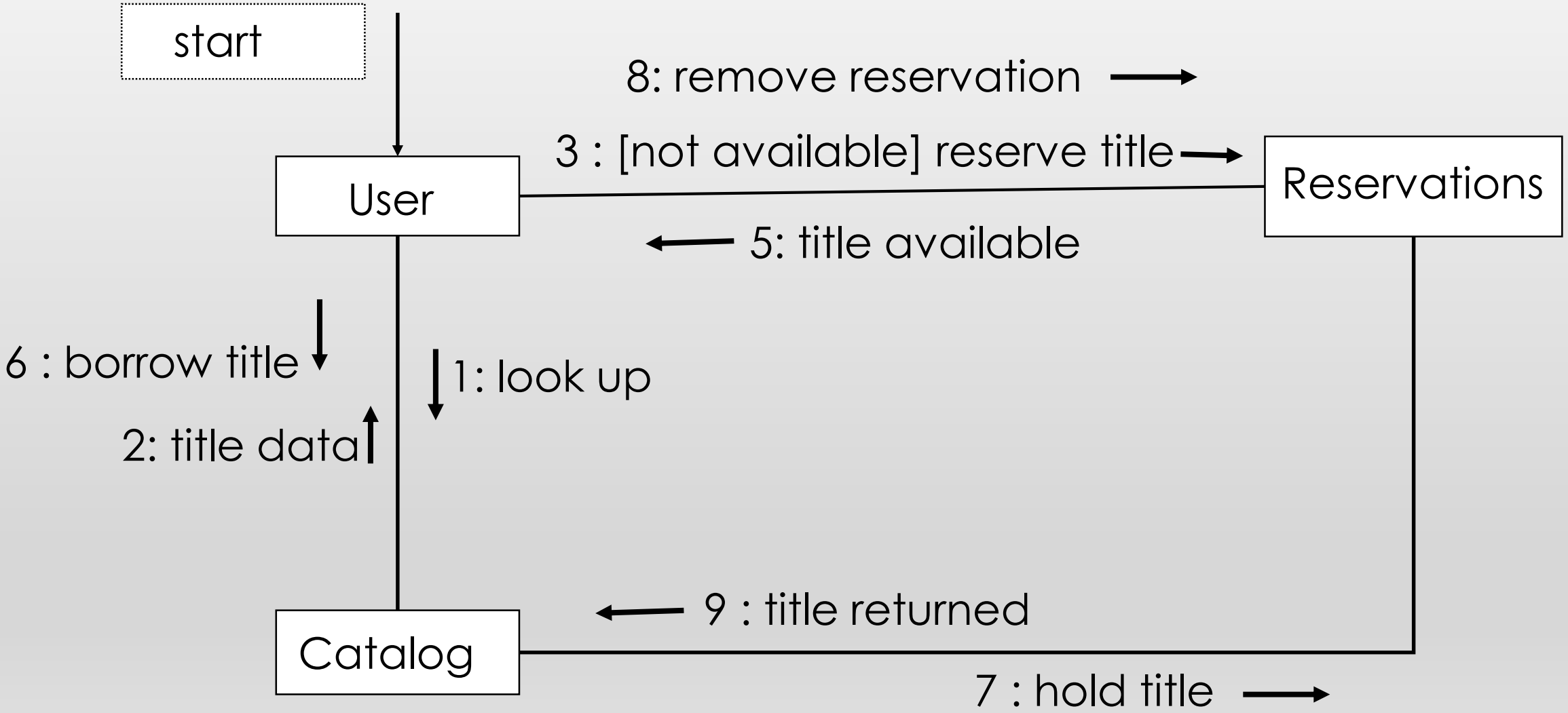


COMMUNICATION DIAGRAMS

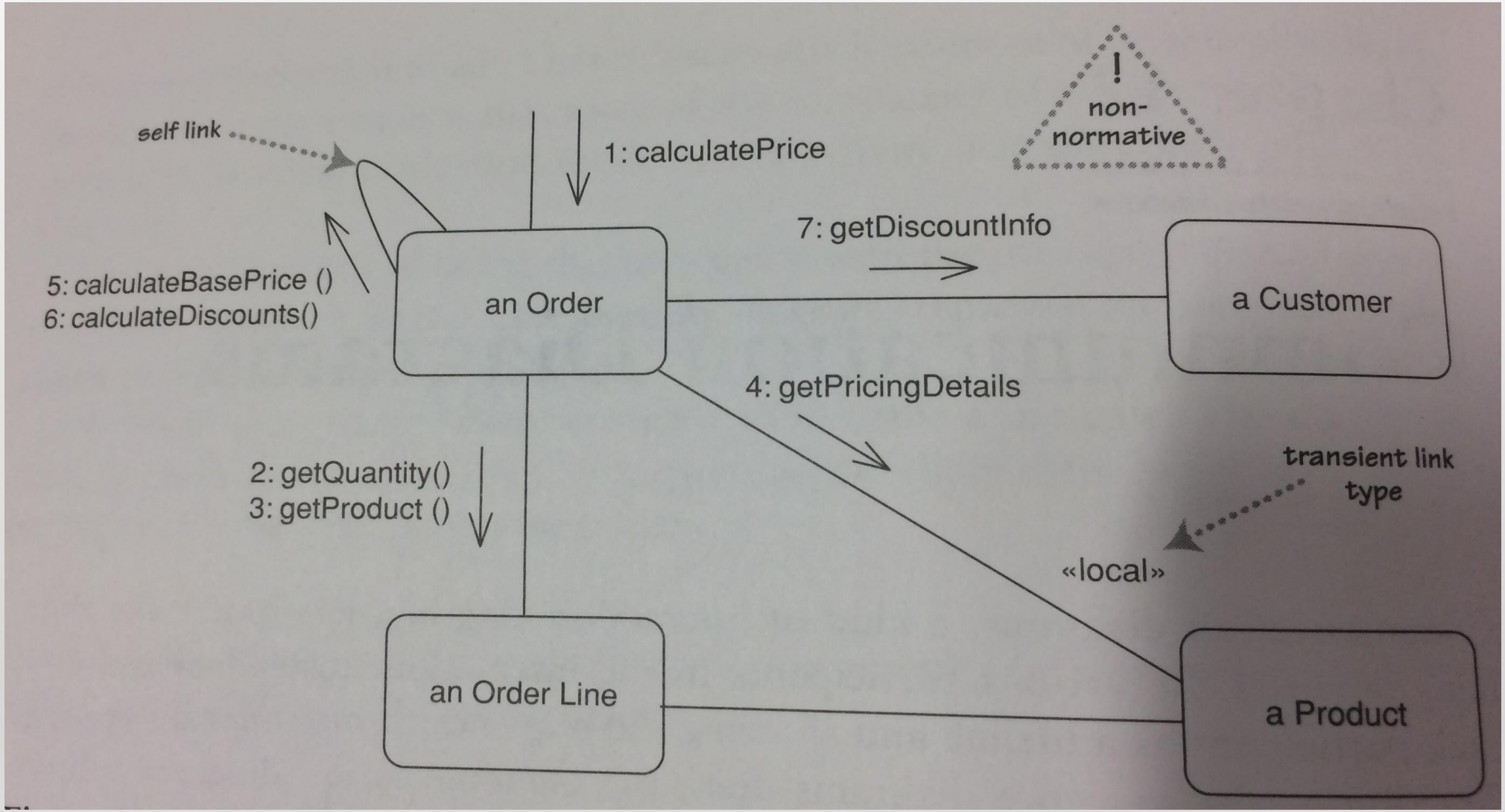
- Shows the data links between the various participants in the interaction
- The objects are listed as rectangles and the arrows indicate the messages being passed
- The numbers next to the messages show the sequence of the messages as they are passed between the objects

- The diagram is used to show how the components cooperate for a given use case –> It could be used to validate the component diagram

EXAMPLE OF COMMUNICATION DIAGRAM



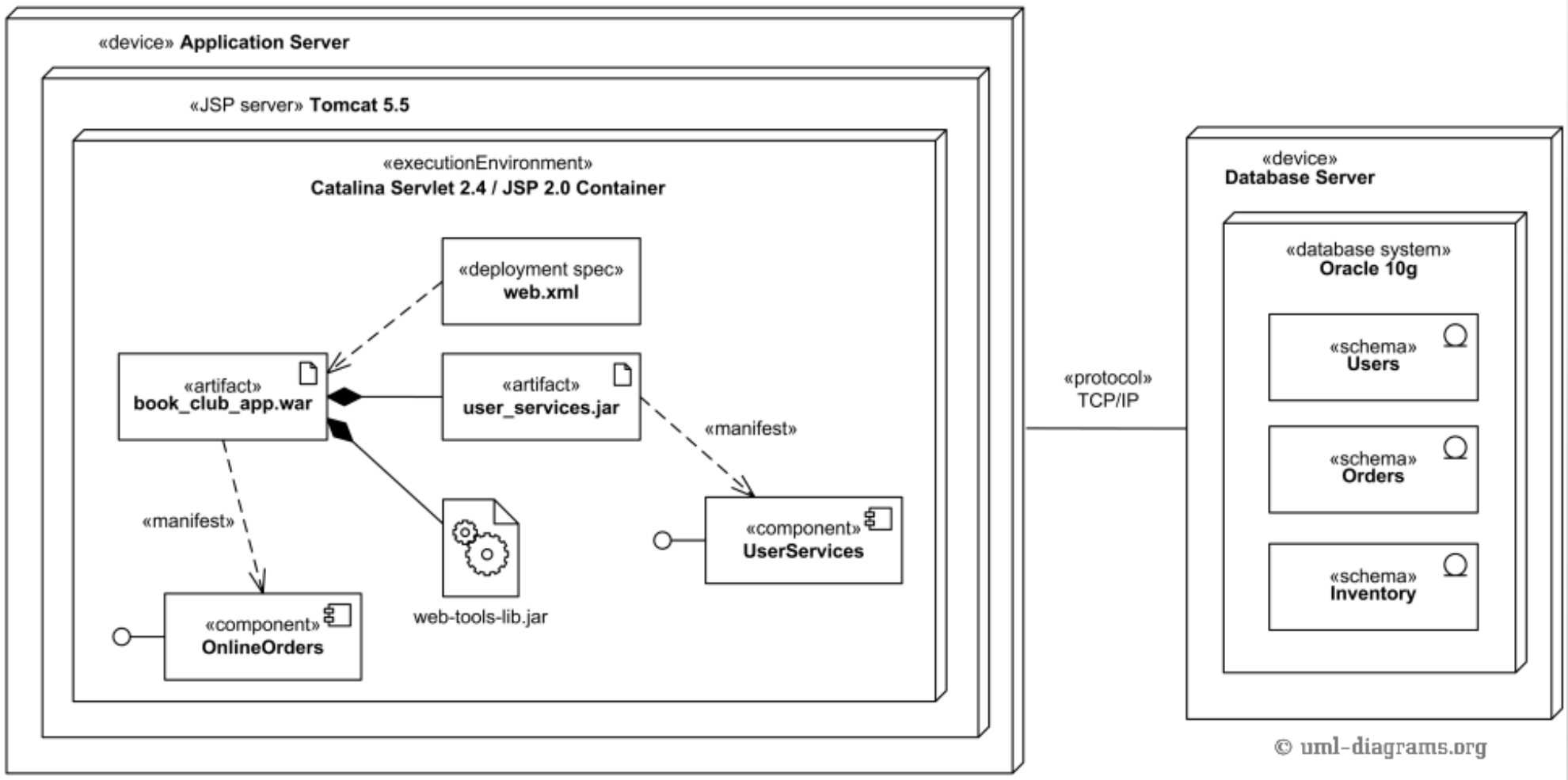
EXAMPLE OF COMMUNICATION DIAGRAM



DEPLOYMENT DIAGRAM

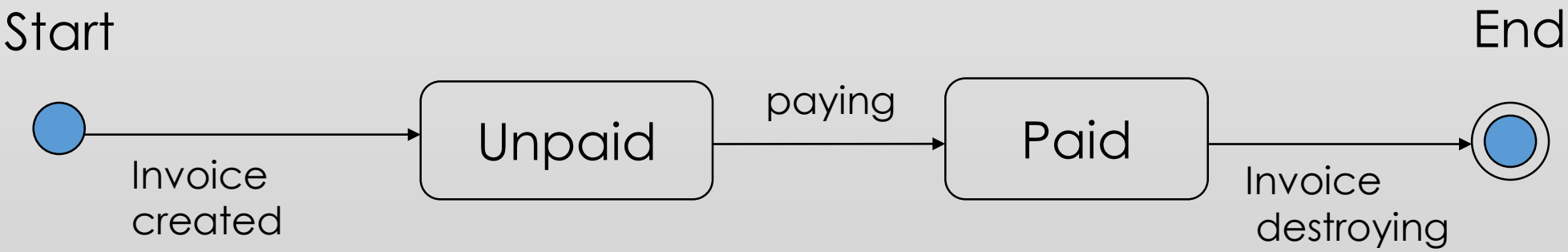
- Deployment diagram models the allocation of artifacts such as software and files to nodes such as devices. It also models the communication method such as RMI, REST, SOAP, HTTP.

DEPLOYMENT DIAGRAM

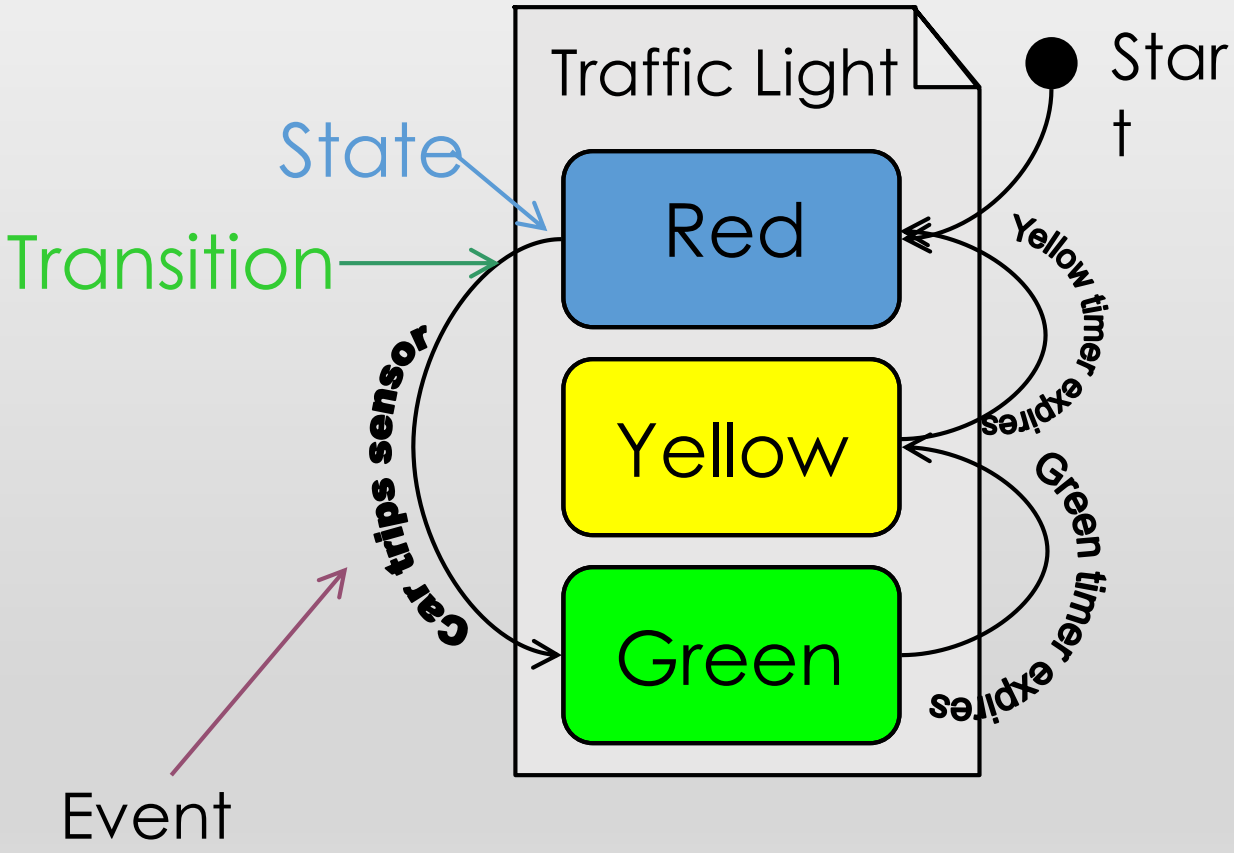


STATE DIAGRAMS (BILLING EXAMPLE)

State Diagrams show the sequence of states an object goes through during its life cycle in response to stimuli.



STATE DIAGRAMS (TRAFFIC LIGHT EXAMPLE)



SUMMARY

- UML offers a graphical notation language to model system structure and behavior
- By using standard notations architecture could be communicated easier.
- The language has multiple diagrams, including the use case, class, state, activity, sequence, interaction, and deployment diagrams.
- Architect selects the diagram based on the aspects that they want to model: structure, interaction, message sequence, object-state change….

EXERCISE

Description: The customer browses the catalog and adds desired items to the shopping basket. When the customer wishes to pay, they describe the shipping and credit card information and confirm the sale. The system checks the authorization on the credit card and confirms the sale both immediately and with a follow-up e-mail.

Work: Develop

1- use case diagram

2- component diagram

3- communication diagram