UML DIAGRAM NEW

Tutorial Content

- UML basic

1. Use case diagram
2. Class diagram
3. Activity diagram
4. Sequence diagram
5. State Machine diagram

What is UML?

- Standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, business modeling and other non-software

•The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

- The UML is a very important part of developing object oriented software and the software development process.
- The UML uses mostly graphical notations to express the design of software projects.
- Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

UML Diagram – What is UML?

The Unified Modeling Language (UML) is a standard language for

Specifying

Visualizing

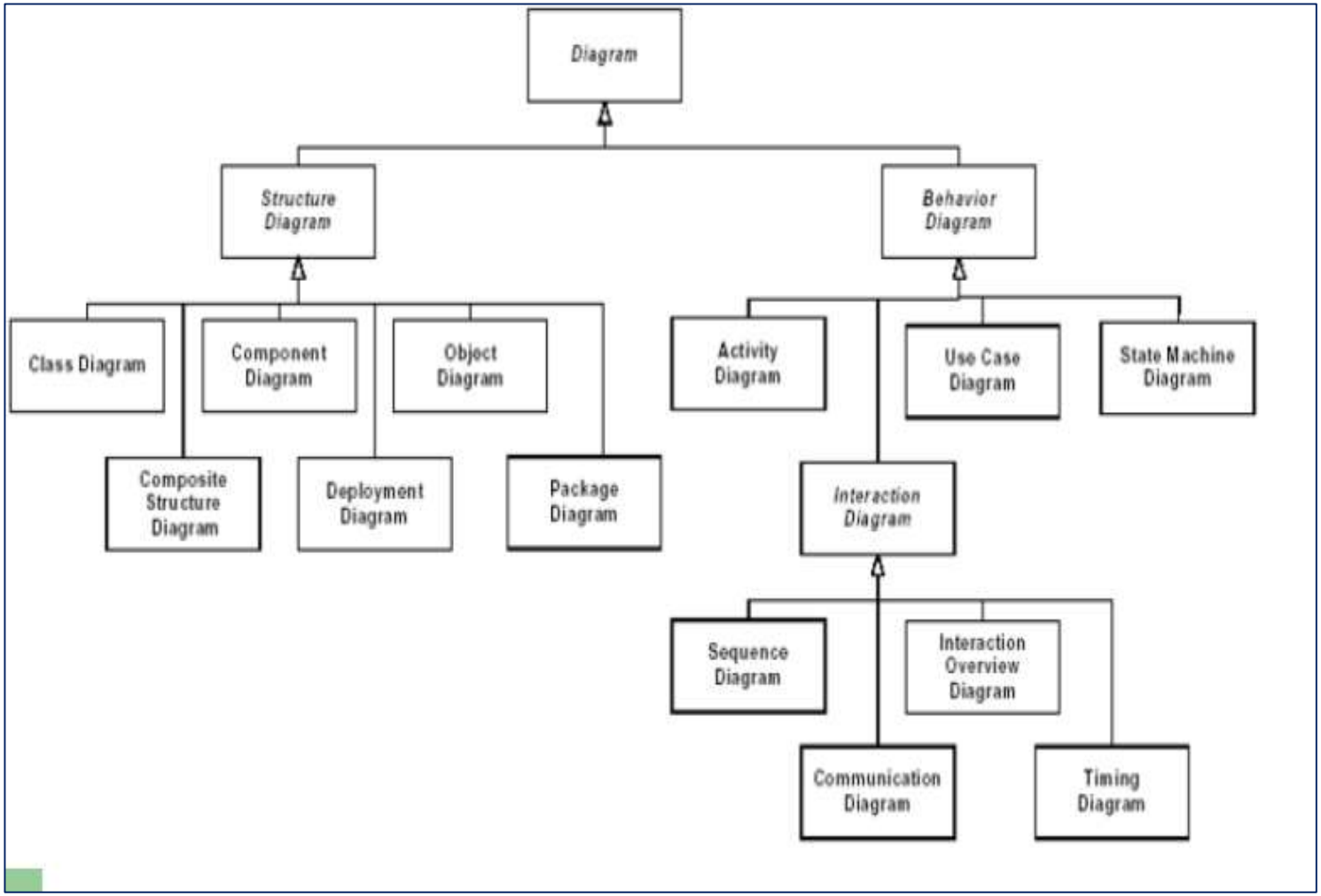
Constructing

Documenting

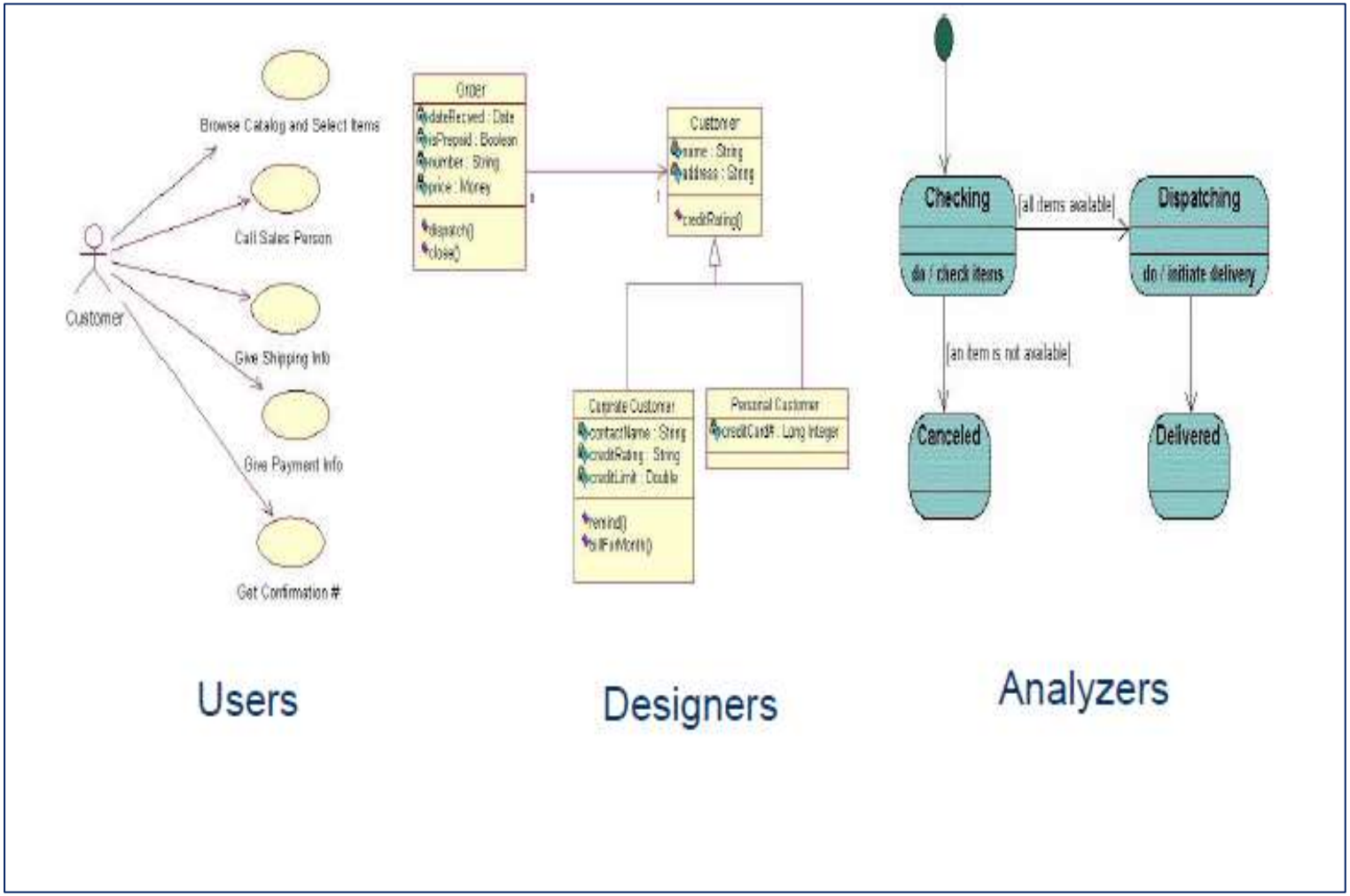
Business Modeling

Communications

UML Diagram Hierarchy



Different Views



Unified Modeling LANGUAGE

Language for:

Visualizing: Graphical models with precise semantics

Specifying: Models are precise, unambiguous and complete to capture all important Analysis, Design, and Implementation decisions.

Constructing: Models can be directly connected to programming languages, allowing forward and reverse engineering

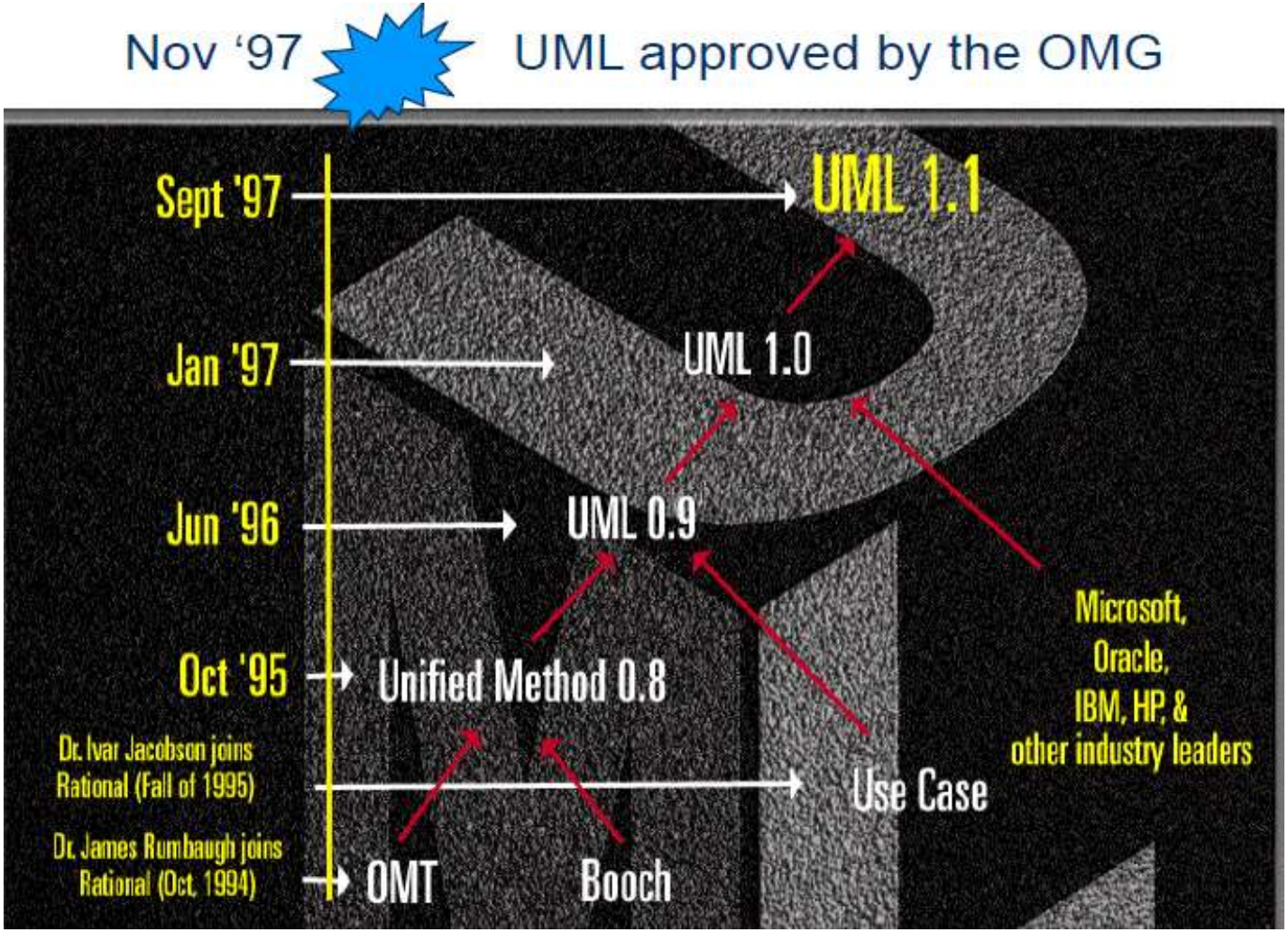
Documenting

Notions for UML Modeling

- Booch …….Grady Booch
- OMT (Object Modeling Technology). Dr.James Rumbaugh
- Unified Modeling Language(UML)

- Booch, OMT ,Jacobson
- Note: Standards Governing Boards such as ANSI & Object Management Group(OMG)

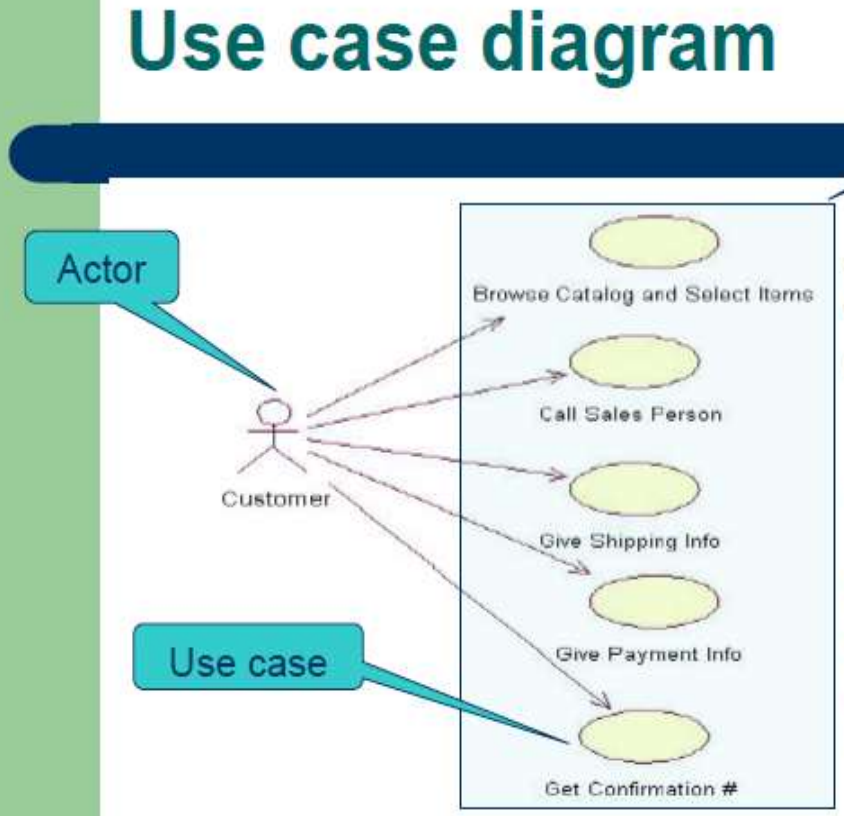
Brief History of UML



Diagrams in the UML

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Activity diagram
- Component diagram
- Deployment diagram

Use case diagram



- overview the usage requirements
- presentations project stakeholders
- "the meat" of the actual requirements

Actor:

An actor is a person, organization, or external system that plays a role in one or more interactions with your system

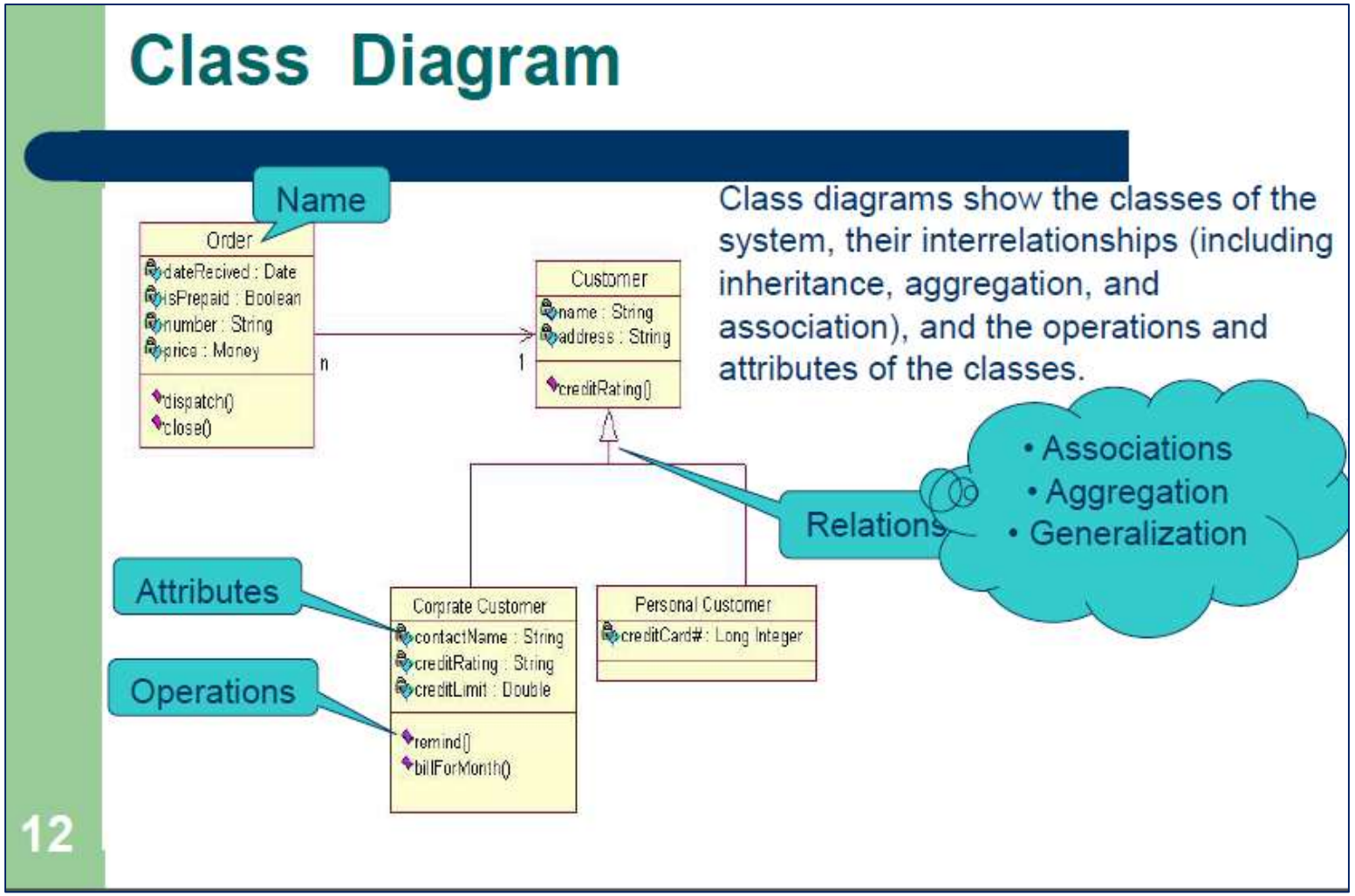
System boundary:

indicates the scope of your system. Anything within the box

Online C2C shopping

11

Class Diagram



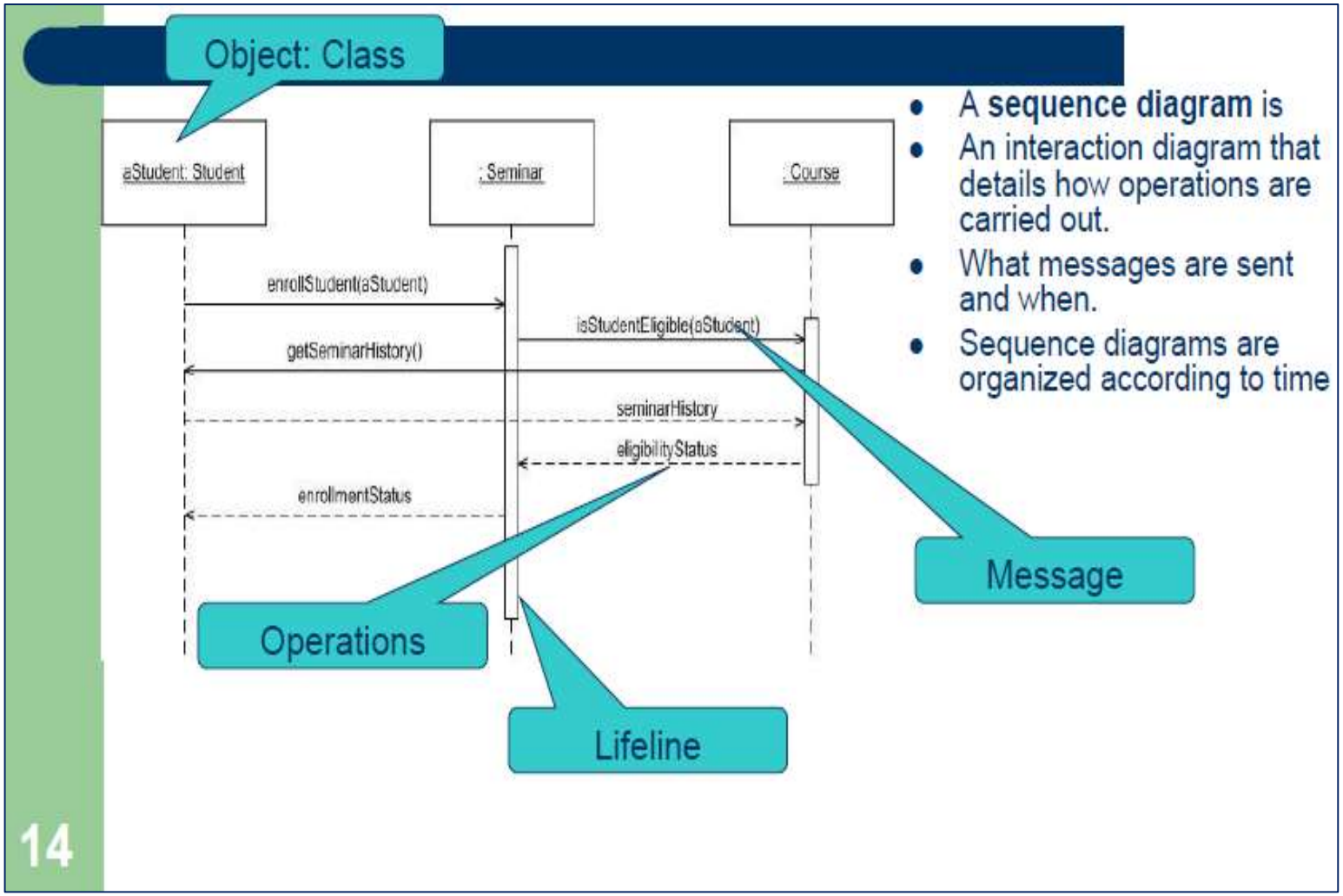
Relationships between Class Diagrams

- **Association**--a relationship between instances of the two classes. There is an association between two classes if an instance of one class must know about the other in order to perform its work. In a diagram, an association is a link connecting two classes.

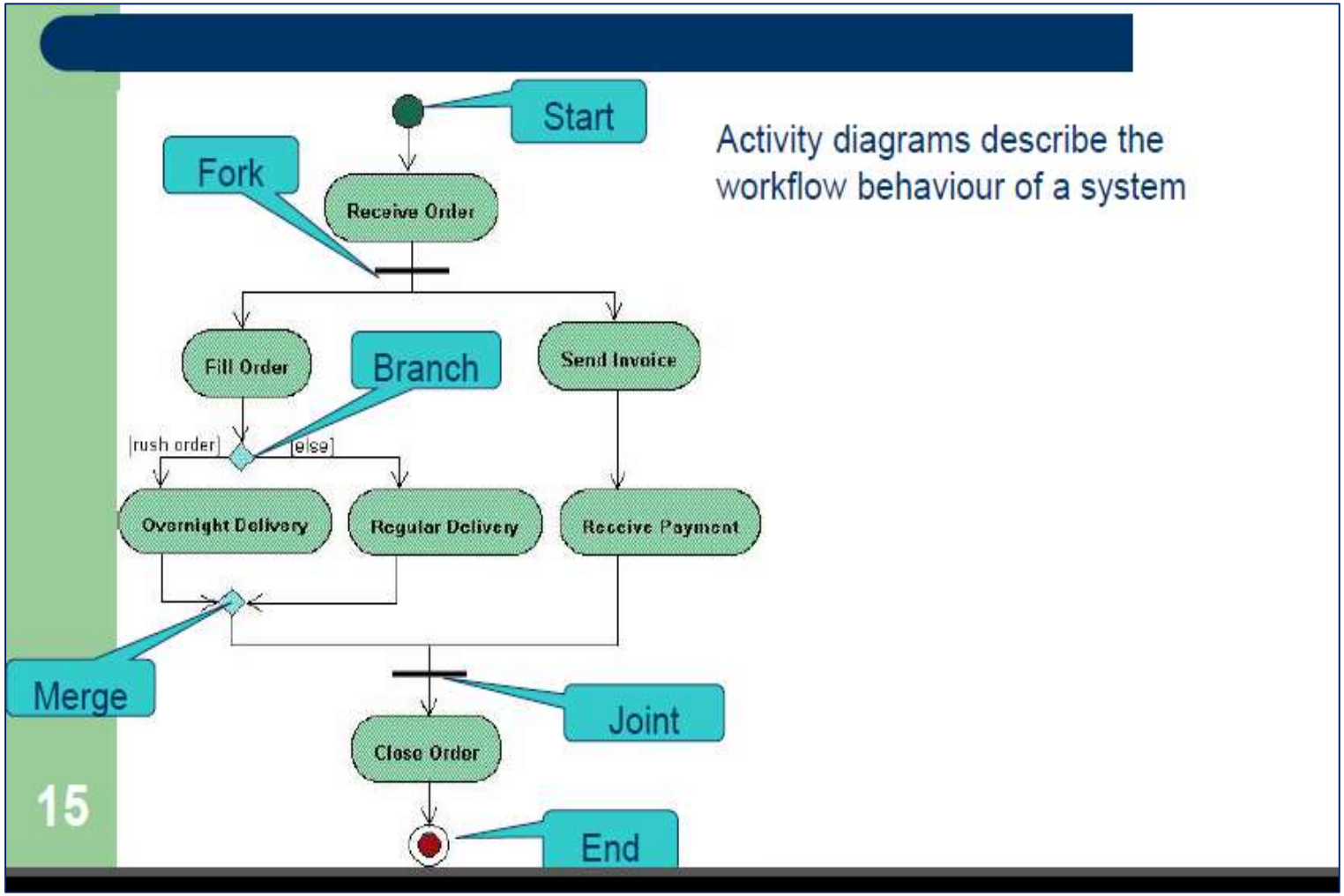
- **Aggregation**--an association in which one class belongs to a collection. An aggregation has a diamond end pointing to the part containing the whole.

- **Generalization**--an inheritance link indicating one class is a superclass of the other. A generalization has a triangle pointing to the superclass.

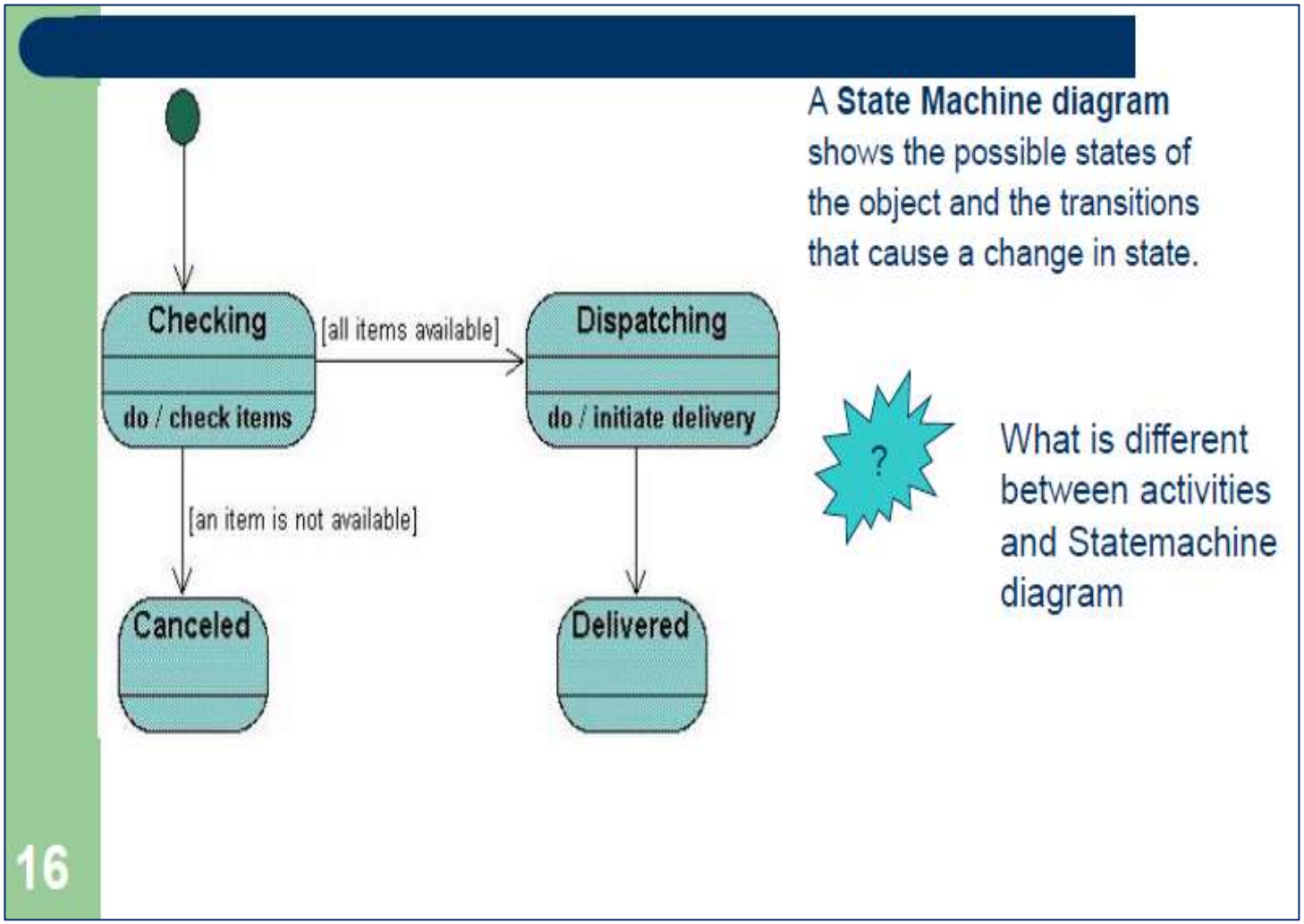
Sequence Diagrams



Activities Diagram



State Machine Diagram

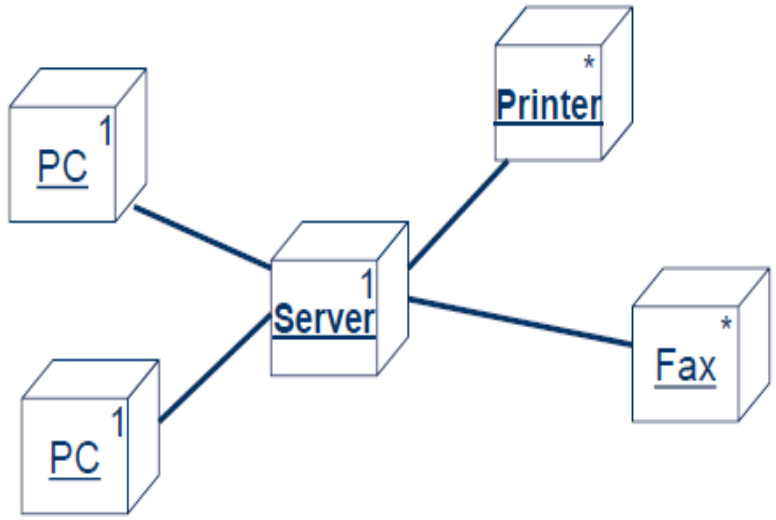


Component Diagrams

- Component is any form of software
- Component diagrams show software components and their relationships (dotted arrows)
- They show physical components from a high level view to show how code-modules are distributed.
- They are more often used together with deployment diagrams and not separate.

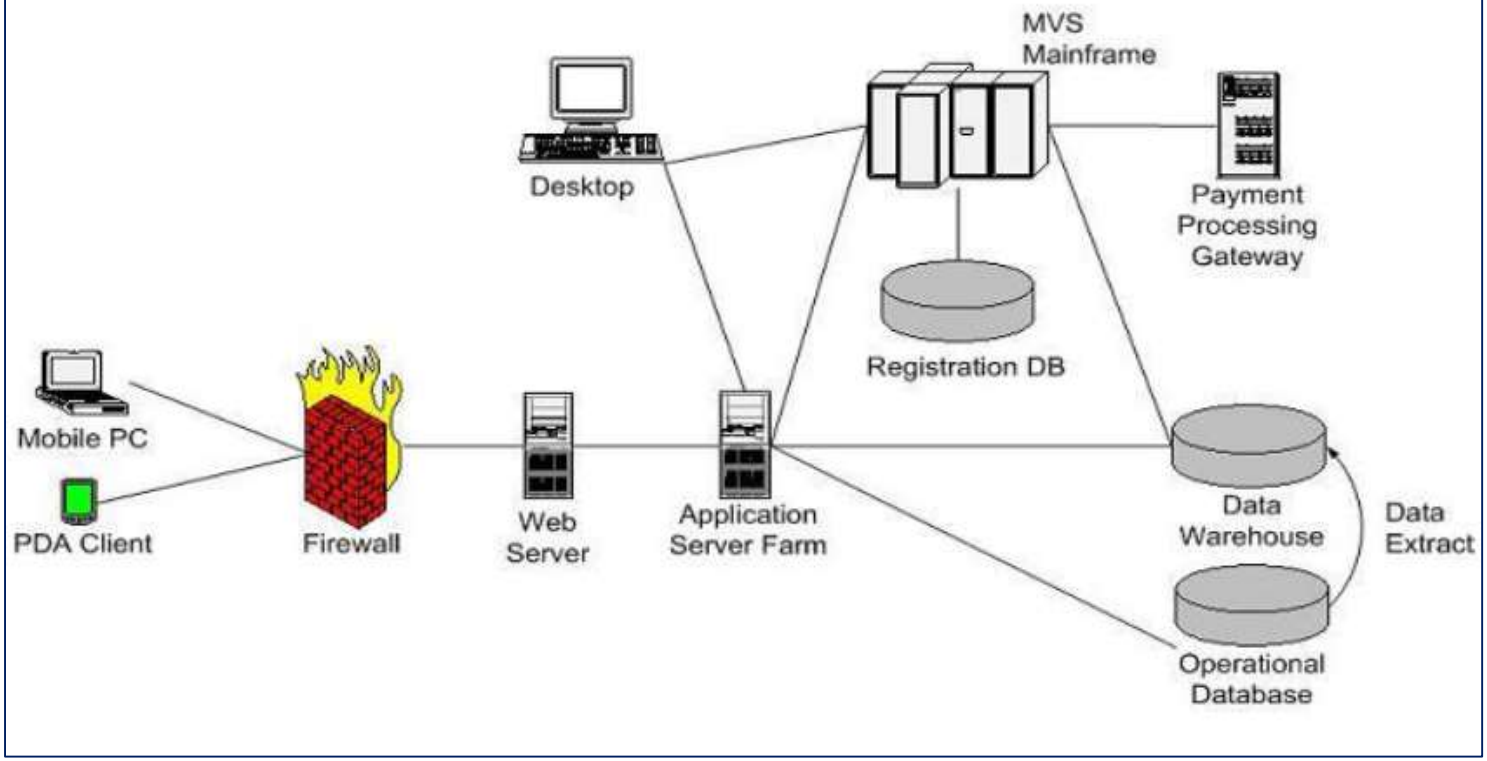
Deployment Diagrams

- Each node or processing element is represented by a 3D-box.
- The communication/relationships are represented with solid lines



Network Diagrams

- Network diagrams are often drawn using software-based drawing tools (figure below was drawn using Microsoft Visio)



Object flow

- Shows output objects and (optionally) their state