

## Chapter 1: Introduction to UML

- What is UML?
- Concept of Modeling
- Goal of UML
- UML Extension and Mechanism
- UML Diagram

01/11/2009

Chapter 1

1

## What is UML?

- UML is a visual language for modeling and communicating by specify, visualize, construct, and document about software systems (OO styles) through the use of diagrams and supporting text.
- UML stand for:
  - Unified: Unification from modeling languages
  - Modeling: Making a semantically – formal specification of the meaning and behavior of something to complete abstraction of a system
  - Language: graphical language

01/11/2009

Chapter 1

2

## What is a Model?

- Model is a simplified representation of a reality system at some particular point in time or space intended to promote understanding of the real system. Model is similar but simpler than the system its represent. Modeling is the process of producing model.
- Example: *Model of Accounting System, Model of Hotel System, Model of Inventory Control System*

01/11/2009

Chapter 1

3

## Why should to model?

- Models give us a template that guides us in constructing a system.
- Models help us visualize a system at different levels of abstraction; this makes it easier to manage complexity and to understand the system.
- It is not expensive to experiment with multiple solutions when you operate on a high level of abstraction.
- Models help us to find the extensions of system to make it efficiency.
- Manage the risk of mistakes.
- Models document the decisions we have made.
- Models help for communication between different stakeholders.

01/11/2009

Chapter 1

4

## Some systems do not need to model

- The problem domain is well known.
- The solution is relatively easy to construct.
- Very few people need to collaborate to build or use the solution (often only one).
- The solution requires minimal ongoing maintenance.
- The scope of future needs is unlikely to grow substantially

01/11/2009

Chapter 1

5

## Goal of UML

- Define an easy-to-learn but semantically rich visual modeling language.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.
- Include ideas from other modeling language.
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices

01/11/2009

Chapter 1

6

## UML Extension Mechanisms

- **Stereotype:** appears inside of << >> and characterizes a type of element like a class or relationship without specifying its implementation.

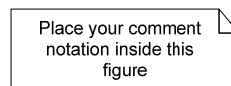


01/11/2009

Chapter 1

7

- **Comment:** use to give comments in the icon below and place it anywhere on any UML diagram.



- **Constraint:** use throughout the UML diagrams to limit the use of model elements. Constraint place inside the { }.

{age>20 and <121}

01/11/2009

Chapter 1

8

## UML Diagram

UML Diagrams categorized into 3 views:

- **Functional View**
  - Use Case Diagram
  - Activity Diagram
- **Static View**
  - Class and Object Diagram
  - Component and Deployment Diagram
- **Dynamic View**
  - Sequence and Collaboration Diagram
  - Statechart Diagram

01/11/2009

Chapter 1

9

## UML History

- Modeling languages began to appear between mid-**1970s** and the late 1980s
- Modeling language increase from less than 10 to 50 during 1989 to 1994
- The development of UML began in late 1994 when Grady Booch and James Rumbaugh of Rational Software Corporation
- In the Fall of 1995, Ivar Jacobson joined with Booch and Rumbaugh
- The efforts of Booch, Rumbaugh, and Jacobson resulted in the release of the UML 0.9 and 0.91 documents in June and October of 1996.
- Later on: UML 1.1, UML 1.3, 1.4, 1.5 and UML 2.0

01/11/2009

Chapter 1

10