

# Lecture # 28

# Software Modeling

# Need for Modeling - 1

- ◉ Modeling is a central part of all activities that lead up to the development good software
- ◉ We build models to communicate the desired structure and behavior of our system

# Need for Modeling - 2

- ◉ We build models to visualize and control the system's architecture
- ◉ We build models to better understand the system we are building, often exposing opportunities for simplification and reuse
- ◉ We build models to minimize risk

# Need for Modeling - 3

- ◉ We build models so that we can better understand the system we are developing
- ◉ We build models of complex systems because we cannot comprehend such a system in its entirety

A model is a simplification  
of reality

# Four Aims of Modeling

- ◉ Models help us to visualize a system as it is or as we want it to be
- ◉ Models permit us to specify the structure or behavior of a system
- ◉ Models give us a template that guides us in constructing a system
- ◉ Models document the decisions we have made

# Do We Model Everything?

- ◉ Modeling is not just for big systems
- ◉ Small pieces of software can also benefit from modeling
- ◉ Larger and more complex systems benefit more from modeling

# Principles of Modeling - 1

- ◉ The choice of what models to create has profound influence on how a problem is attacked and how a solution is shaped
- ◉ Every model may be expressed at different levels of precision



# Principles of Modeling - 2

- The best models are connected to reality
- No single model is sufficient. Every nontrivial system is best approached through a small set of nearly independent models

# Problem Analysis

- Activity that encompasses learning about the problem to be solved, understanding the needs of potential users, trying to find out who the user really is, and understanding all constraints on the solution
- Defining the product space – the range of all possible software solutions that meets all known constraints

# Points to Note

- ◉ Understanding the needs of potential users
- ◉ Trying to find out who the user really is
- ◉ Understanding all constraints on the solution
- ◉ All three are very difficult

# Need for a Software Solution

- ◉ There is recognition that a problem exists and requires a solution
- ◉ A new software idea arises
- ◉ Either there is no automated system or there is a need for some improvements in the existing automated system

# Subjects to Study for Modeling Manual System

- People and/or machines currently playing some role
- Items produced, processed, or consumed by these people and machines
- Functions performed by these people and machines
- Basic modes of operation that determine what functions are performed and when

# Subjects to Study for Modeling to Improve a System - 1

- **People/machines** who have a need for some service to be performed or some item to be produced
- **Items** that need to be produced to satisfy the need
- **Items that are necessary in order to produce the required new service or item**

# Subjects to Study for Modeling Improve a System - 2

- ◉ **Functions** that need to be performed in order to generate the required new service or item
- ◉ **Basic modes of operations** that determine what functions are performed and when

# Steps in Problem Analysis

- ◉ Gain agreement on the problem definition
- ◉ Understand the root causes – the problem behind the problem
- ◉ Identify the stakeholders and the users
- ◉ Define the solution system boundary
- ◉ Identify the constraints to be imposed on the solution



# Principles of Modeling

- ◉ Partitioning
- ◉ Abstraction
- ◉ Projection

# Partitioning

- Captures aggregation/part of relations among elements in the problem domain

# Abstraction

- Captures generalization/specialization relations among elements in the problem domain

# Projection

- Captures different views of elements in the problem domain

# Avoid the Urge to Design

- One can easily get into the trap of completely designing the proposed system
- The focus of modeling during requirements engineering is **understanding** and that of preliminary design is **optimization**

# Software Modeling

- A number of modeling techniques have been developed over the years

# Features of Modeling Techniques

- Facilitate communication
- Provide a means of defining the system boundary
- Provide a means of defining partitions, abstractions, and projections

# Features of Modeling Techniques

- Encourage the analyst to think and document in terms of the problem as opposed to the solution
- Allow for opposing alternatives but alert the analyst to their presence
- Make it easy to modify the knowledge structure



# Modeling Techniques

- ◉ Object-oriented modeling
  - > Static and dynamic modeling
- ◉ Functional modeling
- ◉ Dynamic modeling

# Focus of this lecture

- ◉ We'll introduce object-oriented modeling here and discuss it in detail in the next lecture
- ◉ We'll talk about functional modeling and dynamic modeling details in later lectures

# Object-Oriented Modeling - 1

- ◉ The main building block of all software systems is the object or class
- ◉ An object is a thing, generally drawn from the vocabulary of the problem space or the solution space
- ◉ A class is a description of a set of common objects

# Object-Oriented Modeling - 2

- Object-oriented paradigm helps in software modeling of real-life objects in a direct and explicit fashion
- It also provides a mechanism so that the object can inherit properties from their ancestors, just like real-life objects

# Object-Oriented Modeling - 3

- Every object has identity, state, and behavior

# Object-Oriented Modeling - 4

- ◉ The object-oriented approach has proven itself to be useful, because it has proven to be valuable in building systems in all sorts of problem domains and encompassing all degrees of size and complexity

# Object-Oriented Modeling - 5

- Object-oriented development provides the conceptual foundation for assembling systems out of components using technology such as Java Beans or COM+

# Object-Oriented Modeling - 6

- ◉ A number of consequences flow from the choice of viewing the world in an object-oriented fashion
  - > What is the structure of good OO architecture?
  - > What artifacts the project should create?
  - > Who should create them?
  - > How should they be measured?



# Summary

- ◉ We discussed the need for modeling
- ◉ The primary aim of modeling is to better understand the system to be built
- ◉ There are many modeling techniques available to model a system
- ◉ Object-oriented paradigm helps in software modeling of real-life objects in a direct and explicit fashion