

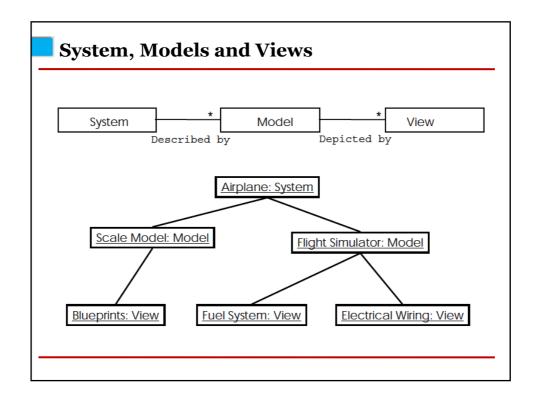
What is Modeling?

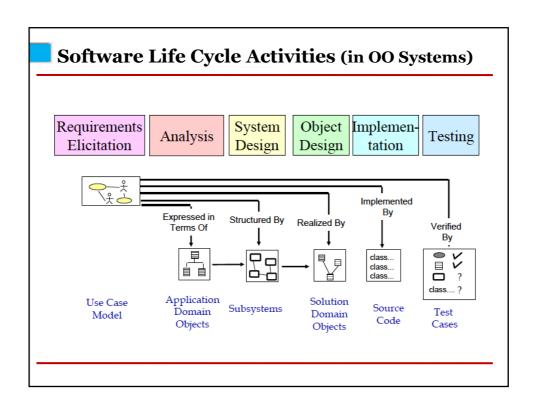
### System, Models and Views

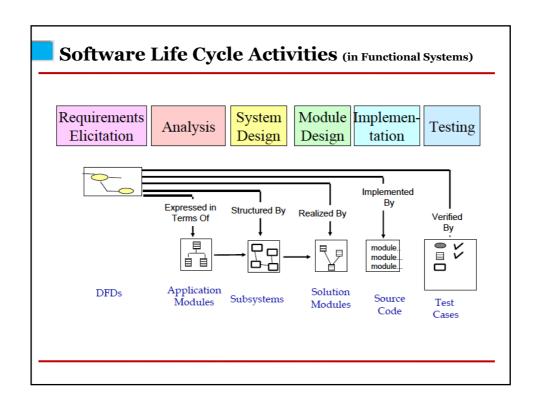
- A model is an abstraction describing a subset of a system/component
- A view depicts selected aspects of a system/component
- A notation is a set of graphical or textual rules for depicting a model
- Views and models of a single system typically overlap each other

#### Examples:

- · System: Aircraft
- Models: All blueprints, electrical wiring, fuel system, Flight Controller







### What constitutes a good model?

- A model should
  - use a standard notation
  - be understandable by clients and users
  - lead software engineers to have insights about the system
  - provide abstraction
- · Models are used:
  - to help create designs
  - to permit analysis and review of those designs.
  - as the core documentation describing the system.

# **Unified Modeling Language**



- A visual Modeling language
- A standardized general-purpose modelling language in the field of software engineering.
- used to specify, visualize, construct and document the artefacts of an object-oriented software-intensive system under development
- A set of 9 diagrams (UML 1.4) and 13 diagrams (UML 2.0)
- · Not an industry standard but used widely

#### UML

- Is not a S/W development methodologies
  - Descriptive not prescriptive
- Combines best practices from
  - Data Modeling (ex: ER diagram)
  - Business Modeling (ex: Workflows)
  - Object Modeling, and
  - Component Modeling
- · Combines
  - Booch method (G Booch)
  - OMT (J Rumbaugh)
  - OOSE (E Jacobson)
- An extensible language
  - Stereotypes, profiles

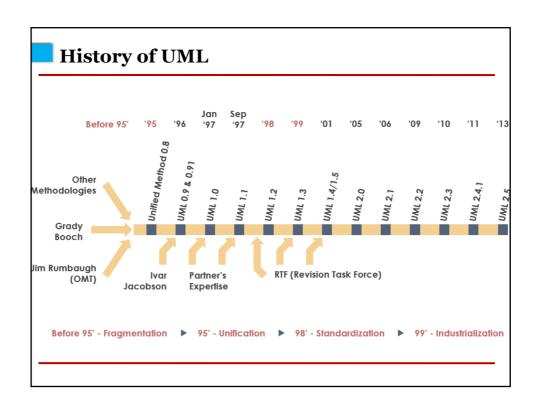
# **UML Users**

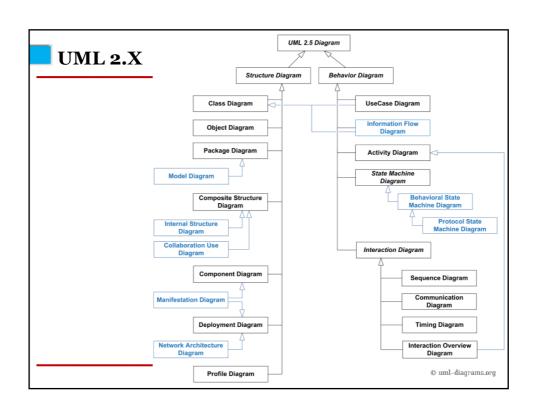


- Architect
- · Domain expert
- Designer
- Programmer/Developer
- Instructor







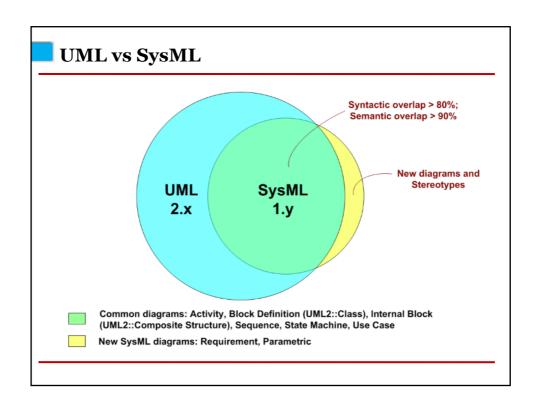


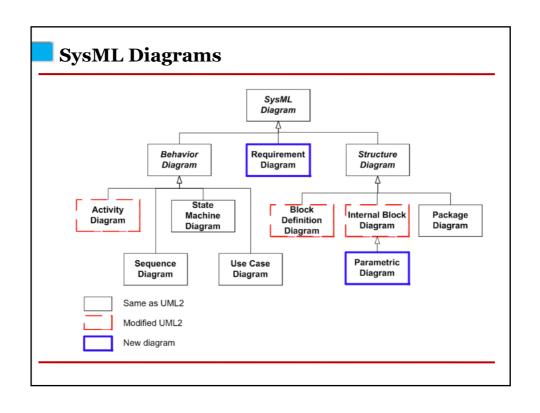
#### **Important UML Diagrams UML Class Diagrams Use Cases** information structure user's view relationships between Lists functions data items visual overview of the modular structure for main requirements the system UML Package Diagrams (UML) Statecharts Overall architecture responses to events Dependencies dynamic behavior between components event ordering, reachability, deadlock, etc **UML Sequence Diagrams Activity diagrams** individual scenario business processes; concurrency and interactions between users and system synchronization; Sequence of dependencies messages between tasks;

# **System Modeling Language**



- · A visual Modeling language
- SysML dialect of the UML for systems engineering applications in 2003
- used to specify, visualize, construct and document the artefacts for Model-Based Systems Engineering (MBSE) applications.
- SysML supports the specification, analysis, design, verification and validation of a broad range of systems and systems-ofsystems.
- · An industry standard and widely used.





#### We will discuss

- Use Case Model (Use Case Diagram and Descriptions)
- · Sequence diagram
- State chart/State Machine/State Diagram
- · Activity Diagram
- · Class diagram
- Component/Package Diagram

#### **Use Cases**

- Define system functional requirements in terms of Actors and Use cases
  - Each use case specify a piece of functionality
  - A use case can be elaborated in terms of sequence of interactions between Actor and the domain objects
  - Simple use cases may involve only one interaction
  - More complicated use cases may involve several interactions

## **Problem Specification**

The UTD wants to computerize its registration system

- The Registrar sets up the curriculum for a semester
- Students select 3 core courses and 2 electives
- Once a student registers for a semester, the billing system is notified so the student may be billed for the semester
- Students may use the system to add/drop courses for a period of time after registration
- Professors use the system to set their preferred course offerings and receive their course offering rosters after students register
- Users of the registration system are assigned passwords which are used at logon validation

What's most important?

#### **UML - Actors in Use case**

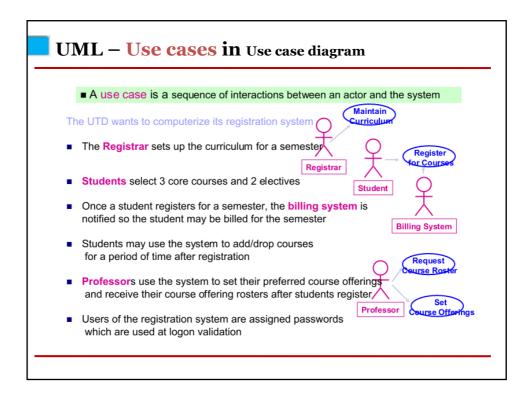
An actor is someone or some thing that must interact with the system under development

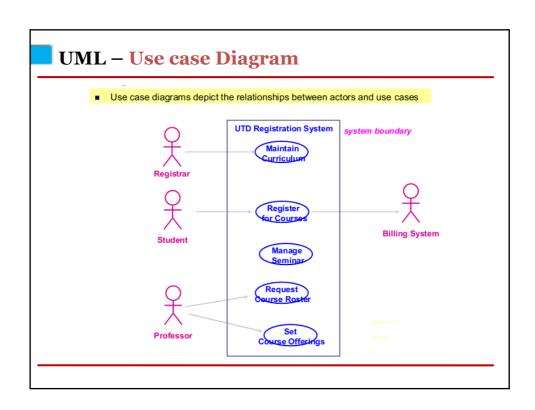
The UTD wants to computerize its registration system

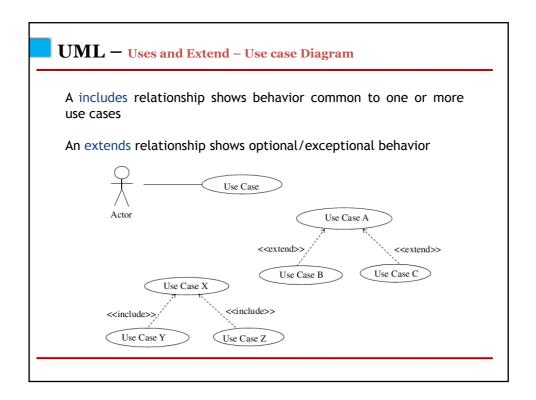
- The Registrar sets up the curriculum for a semester
- Students select 3 core courses and 2 electives
- Once a student registers for a semester, the billing system is notified so the student may be billed for the semester
- Students may use the system to add/drop courses for a period of time after registration
- Professors use the system to set their preferred course offerings and receive their course offering rosters after students register
- Users of the registration system are assigned passwords which are used at logon validation

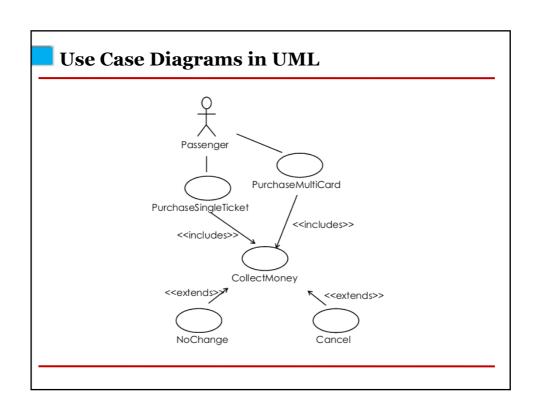


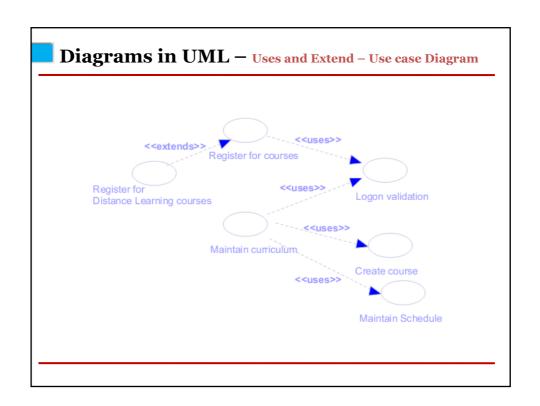


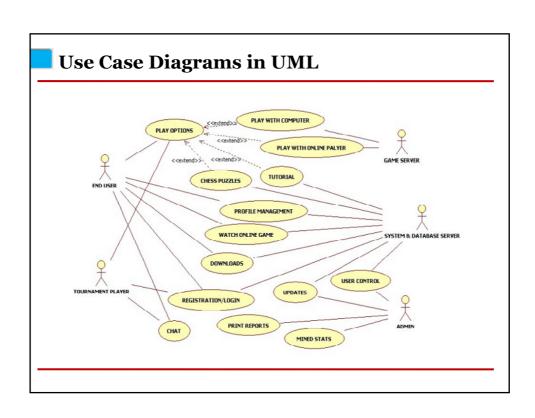












### Question? - Design a use case diagram

A POS (Point-Of-Sale) system is a computer system typically used to manage the sales in retail stores. It includes hardware components such as a computer, a bar code scanner, a printer and also software to manage the operation of the store.

The most basic function of a POS system is to handle sales. When a customer arrives at a POS counter with goods to purchase, the cashier will start a new sale transaction. When the barcode of a good is read by the POS system, it will retrieve the name and price of this good from the backend catalog system and interact with inventory system to deduce the stock amount of this good.

When the sale transaction is over, the customer can pay in cash, credit card or even check. After the payment is successful, a receipt will be printed. Note that for promotion, the store frequently issue gift

coupons. The customer can use the coupons for a better price when purchasing goods.

Another function of a POS system is to handle returns.... [The details of which are not given here]