ASE: Modeling Languages
(UML & SysML)

Saurabh Tiwari
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
System, Models and Views

![Diagram of system, models, and views with examples: Airplane as a system described by a scale model and depicted by blueprints, fuel system, and electrical wiring views. Flight Simulator as another model depicted by its own views.]
Software Life Cycle Activities (in OO Systems)
Software Life Cycle Activities (in Functional Systems)

- Requirements Elicitation
- Analysis
- System Design
- Module Design
- Implementation
- Testing

Expressed in Terms Of
Structured By
Realized By
Implemented By
Verified By

DFDs
Application Modules
Subsystems
Solution Modules
Source Code
Test Cases
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
History of UML

Before 95' - Fragmentation  ➤  95' - Unification  ➤  98' - Standardization  ➤  99' - Industrialization
# Important UML Diagrams

<table>
<thead>
<tr>
<th>UML Class Diagrams</th>
<th>Use Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>information structure</td>
<td>user’s view</td>
</tr>
<tr>
<td>relationships between data items</td>
<td>Lists functions</td>
</tr>
<tr>
<td>modular structure for the system</td>
<td>visual overview of the main requirements</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UML Package Diagrams</th>
<th>(UML) Statecharts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall architecture</td>
<td>responses to events</td>
</tr>
<tr>
<td>Dependencies between components</td>
<td>dynamic behavior</td>
</tr>
<tr>
<td></td>
<td>event ordering, reachability, deadlock, etc</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UML Sequence Diagrams</th>
<th>Activity diagrams</th>
</tr>
</thead>
<tbody>
<tr>
<td>individual scenario interactions between users and system</td>
<td>business processes; concurrency and synchronization;</td>
</tr>
<tr>
<td>Sequence of messages</td>
<td>dependencies between tasks;</td>
</tr>
</tbody>
</table>
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-of-systems.

- An industry standard and widely used.
UML vs SysML

- **Syntactic overlap > 80%; Semantic overlap > 90%**

**Common diagrams:** Activity, Block Definition (UML2::Class), Internal Block (UML2::Composite Structure), Sequence, State Machine, Use Case

**New SysML diagrams:** Requirement, Parametric
SysML Diagrams

- **SysML Diagram**
  - **Behavior Diagram**
    - Activity Diagram
    - State Machine Diagram
    - Sequence Diagram
  - **Requirement Diagram**
  - **Structure Diagram**
    - Block Definition Diagram
    - Internal Block Diagram
    - Parametric Diagram

- Same as UML2
- Modified UML2
- New diagram
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?
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.
A use case is a sequence of interactions between an actor and the system.

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.
Use case diagrams depict the relationships between actors and use cases.

- Registrar
- Student
- Professor
- Billing System

- Maintain Curriculum
- Register for Courses
- Manage Seminar
- Request Course Roster
- Set Course Offerings

UTD Registration System (system boundary)
A **includes** relationship shows behavior common to one or more use cases.

An **extends** relationship shows optional/exceptional behavior.
Use Case Diagrams in UML
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 barcode 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]