Unit 5. System Modeling

Introduction to System Modeling; Context Models; Interaction Models; Structural Models; Behavioral Models; Model-Driven Architecture

System modeling

- ✧ System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system.
- ✧ System modeling has now come to mean representing a system using some kind of graphical notation, which is now almost always based on notations in the Unified Modeling Language (UML).
- ✧ System modelling helps the analyst to understand the functionality of the system and models are used to communicate with customers.

Existing and planned system models

- ✧ Models of the existing system are used during requirements engineering.
- ✧ They help clarify what the existing system does and can be used as a basis for discussing its strengths and weaknesses.
- ✧ These then lead to requirements for the new system.
- ✧ Models of the new system are used during requirements engineering to help explain the proposed requirements to other system stakeholders.
- ✧ Engineers use these models to discuss design proposals and to document the system for implementation.
- ✧ In a model-driven engineering process, it is possible to generate a complete or partial system implementation from the system model.

System perspectives

- ✧ An external perspective, where you model the context or environment of the system. (Context Model)
- ✧ An interaction perspective, where you model the interactions between a system and its environment, or between the components of a system. (Interaction Model)
- ✧ A structural perspective, where you model the organization of a system or the structure of the data that is processed by the system. (Structural Model)
- ✧ A behavioral perspective, where you model the dynamic behavior of the system and how it responds to events. (Behavioral Model)

Use of graphical models

- ✧ As a means of facilitating discussion about an existing or proposed system
- ✧ As a way of documenting an existing system
- ✧ As a detailed system description that can be used to generate a system design finally leading to implementation.

UML: Unified Modeling Language

UML

- ✧ The Unified Modeling Language (UML) is a general-purpose modeling language
- ✧ Intended to provide a standard way to visualize the design of a system
- ✧ Developed by Grady Booch, Ivar Jacobson and James Rumbaugh at Rational Software in 1994–1995, with further development led by them through 1996
- ✧ In 1997, UML was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since.
- ✧ In 2005, UML was also published by the International Organization for Standardization (ISO) as an approved ISO standard.

UML

- ✧ Dynamic Models
  - Behavior diagrams emphasize what must happen in the system being modeled
  - Ex: Use Case Diagram, Activity Diagram,   State Transition Diagram, Sequence Diagram
- ✧ Static Models
  - Structure diagrams emphasize the things that must be present in the system being modeled
  - Ex: Class diagram, CRC Index, Object diagram, Component diagram, Deployment diagram

Context models

✧ C
o
n
t
e
x
t

m
o
d
e
l
s

a
r
e

System boundaries

- ✧ System boundaries are established to define what is inside and what is outside the system.
  - They show other systems that are used or depend on the system being developed.
- ✧ The position of the system boundary has a profound effect on the system requirements.
- ✧ Defining a system boundary is a political judgment
  - There may be pressures to develop system boundaries that increase / decrease the influence or workload of different parts of an organization.

Process perspective

- ✧ Context models simply show the other systems in the environment, not how the system being developed is used in that environment.
- ✧ Process models reveal how the system being developed is used in broader business processes.
- ✧ 0 Level data flow diagram may be used to represent the context model
- ✧ May also be represented as Block diagram of sub-systems

✧ Modeling user inter

Interaction Models

Structural models

- Structural models of software display the organization of a system in terms of the components that make up that system and their relationships.
- Structural models may be static models, which show the structure of the system design, or dynamic models, which show the organization of the system when it is executing.
- You create structural models of a system when you are discussing and designing the system architecture.

Class diagrams and CRC Index

- ✧ Class diagrams are used when developing an object-oriented system model to show the classes in a system and the associations between these classes.
- ✧ An object class can be thought of as a general definition of one kind of system object.
- ✧ An association is a link between classes that indicates that there is some relationship between these classes.
- ✧ When you are developing models during the early stages of the software engineering process, objects represent something in the real world, such as a patient, a prescription, doctor, etc.

Class diagrams

Class Identification

1.Hierarchical Object Oriented Design (HOOD) Approach

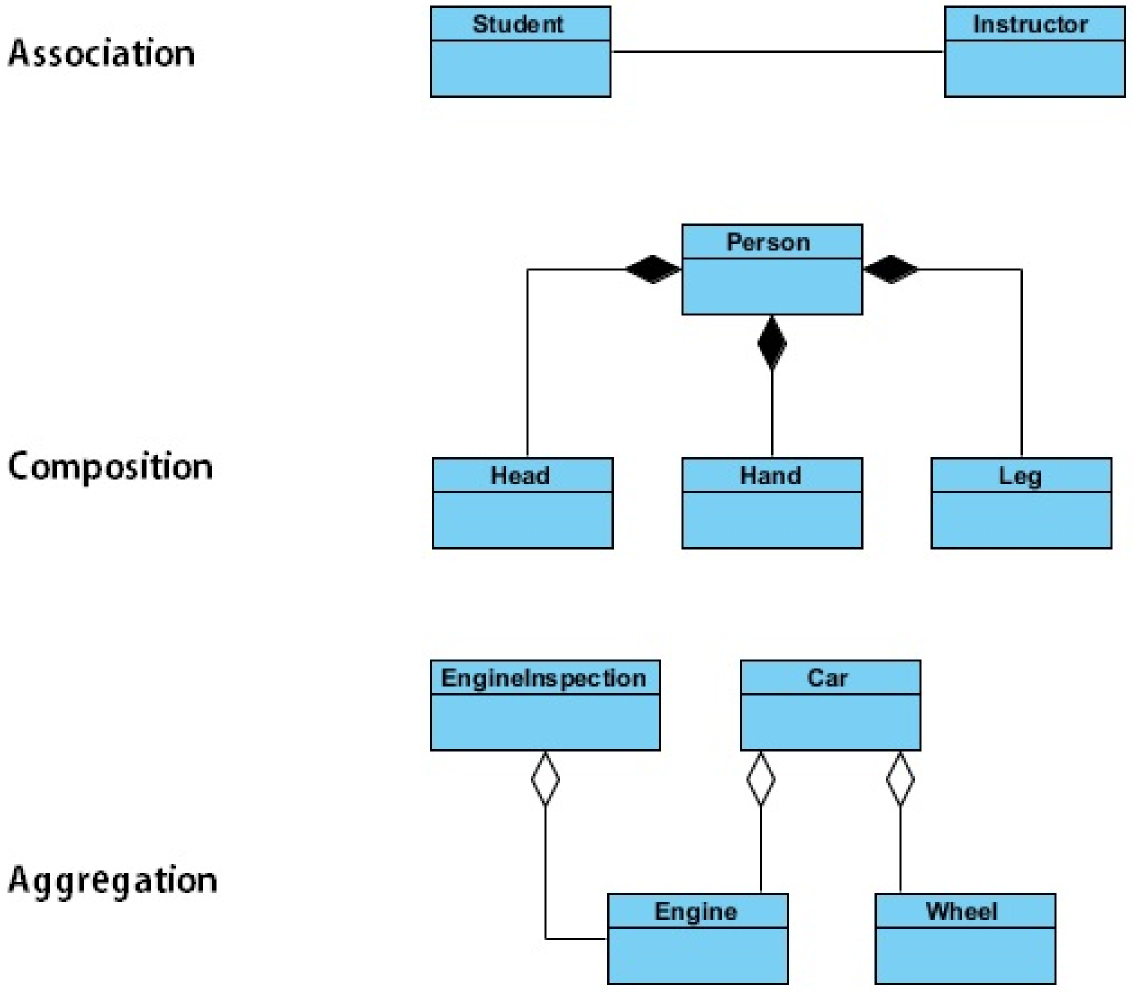
- English Grammar Based Approach
- Common Nouns: Class
- Verb: Methods
- Adjectives: Attributes

2. SOLID Principles

- Single Responsibility Principle
- Open-Closed Principle
- Liskov’s Substitution Principle
- Interface Segregation Principle
- Dependency Inversion Principle

Class diagrams

- Relationships



Class diagrams

- Visibility
  - Private (-) :
    - A private member is visible only from within the class
  - Protected (#) :
    - A protected member is visible from within the class and from the subclasses inherited from this class
  - Public (+) :
    - A public member is visible from anywhere in the system

Behavioural Model

- ✧ **Behavioral models** are **models** of the dynamic **behavior** of the system as it is executing
- ✧ They show what happens or what is supposed to happen when a system responds to a stimulus from its environment
- ✧ You can think of these stimuli as being of two types:
  - ▪ Data: Some data arrives that has to be processed by the system
  - ▪ *Events:* Some event happens that triggers system processing. Events may have associated data but this is not always the case
- ✧ Sequence diagram and State Transition Diagram may be used for Behavioural Modeling

Data modeling

- ✧ **Data modeling** is the analysis of **data** objects and their relationships to other **data** objects
- ✧ **Data modeling** is often the first step in **database** design
- ✧ Two Models
  - ▪ Flow Model:
    - Shows the movement of data from source to the sink
    - Data Flow Diagram is used for modeling the flow
  - ▪ Relationship Model:
    - Shows the relation between different objects of the system
    - Entity-Relationship (ER) diagram is used for modeling the flow

Data modeling

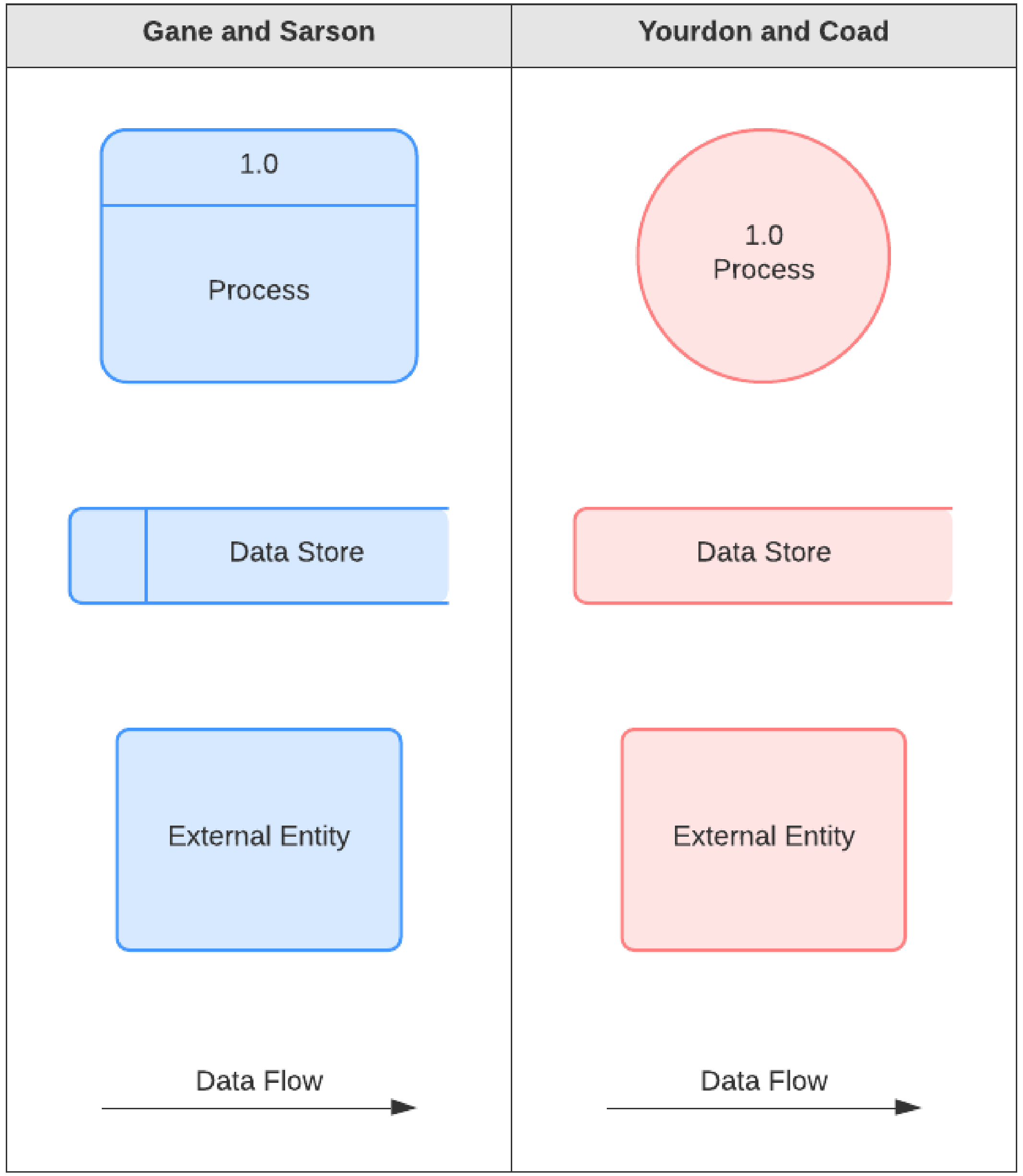
- ✧ **Data Flow Diagram**
- ✧ A **data-flow diagram** (DFD) is a way of representing a flow of a data of a process or a system
- ✧ The DFD also provides information about the outputs and inputs of each entity and the process itself
- ✧ A data-flow diagram has no control flow, there are no decision rules and no loops
- ✧ Levels: 0, 1, 2 … n Level DFD

Data modeling

- ✧ **Data Flow Diagram**
- ✧ Components of DFD
  - ▪ Entity
    - An external entity that communicates with the system and stands outside of the system.
    - An entity may be another system with which the modeled system communicates
  - ▪ Process
    - The process (function, transformation) is part of a system that transforms inputs to outputs
  - ▪ Database
    - database is used to represent storage of data for later use
  - ▪ Flow
    - flow shows the transfer of information (sometimes also material) from one part of the system to another

Data modeling

- ✧ **Data Flow Diagram**
- ✧ Notations:



Data modeling

- **ER Diagram**
  - The ER or (Entity Relational Model) is a high-level conceptual data model diagram
  - Entity-Relation model is based on the notion of real-world entities and the relationship between them
  - Components of the ER Diagram
    - Entities
    - Attributes
    - Relationships

Model-Driven Architecture

- ✧ Model-driven architecture is a software design approach that provides a set of guidelines for the structuring of specifications, which are expressed as models
- ✧ It was launched by the Object Management Group (OMG) in 2001
- ✧ Model-driven architecture supports model-driven engineering of software systems
- ✧ Model-driven engineering focuses on creating and using domain models, which are conceptual models of all the topics related to a specific problem.
- ✧ It highlights and aims at abstract representations of the knowledge and activities rather than the algorithmic concepts.

Model-Driven Architecture

- ✧ MDA focuses on developing a domain model which is a conceptual model of the problem domain
- ✧ The domain model incorporates both behavior and data
- ✧ A domain model is a formal representation of a knowledge domain with concepts, roles, data-types, individuals, and rules, typically grounded in a description logic
- ✧ In the Unified Modeling Language (UML), a class diagram is used to represent the domain model