

Week 6 System Modeling

▼ Definition

- System modeling is the process of developing abstract models of a system, with each model presenting a difference view or perspective of that system

▼ Usages

▼ For the existing system

- Help clarify what are the existing system does
- A basis for discussing its strengths and weaknesses
- Lead to requirements for the new system

▼ For new system

- Help explain the proposed requirements to other system stakeholders
- Engineers use these models to discuss design proposals and to document the system for implementation

▼ Unified Modeling Language (UML)

▼ Use case diagram

▼ Show the interactions between a system and its environment

▼ Drawbacks

- Inappropriate for non-functional requirements
- For situations in which the requirements model must have significant detail and precision (e.g. safety critical systems), a use case may not be sufficient

▼ Activity diagram

▼ Shows the activities involved in a process or in data processing

▼ With swimlanes

- Show who is responsible for each activity in a process
- Each swimlane shows the name of a role at the top

▼ Sequence diagram

- Show the interactions between actors and the system and between system components

▼ Class diagram

- Show the object classes in the system and the associations between these classes

▼ State machine diagram

- Show how the system reacts to internal and external events

▼ Models

▼ Interaction Models

- Helps to identify user requirements
- System-to-system interaction
- Component interaction

▪ Class-based Models

▼ Behavioral Models

- Models of dynamic behavior of a system as it is executing

▼ Context Models

- Context models are used to illustrate the operational context of a system -- they show what lies outside the system boundaries (show other systems in the environment)

▼ System boundaries

- System boundaries are established to define what is inside and what is outside the system.

- The position of the system boundary is a political judgement

▼ **Modeling**

▼ Data-driven Modeling

- Controlled by the data input to the system
- Show the sequence of actions involved in processing input data and generating an associated output

▼ Event-driven Modelling

- Shows how a system responds to external and internal events (state machine diagram)

▼ **Model-driven Engineering**

▼ Definition

- Model-driven engineering (MDE) is an approach to software development where models rather than programs are the principal outputs of the development process.
- The programs that execute on a hardware/software platform are then generated automatically from the models

▼ Usage

▼ Pros

- Allows systems to be considered at higher levels of abstraction
- Generating code automatically means that it is cheaper to adapt systems to new platforms

▼ Cons

- Models for abstraction and not necessarily right for implementation
- Savings from generating code may be out weighted by the costs of developing translators for new platforms