Chapter 2 System Modeling

-sushant chalise

System modeling is the process of developing abstract model of a system, with each model presenting a different view or perspective of the system. System modeling has generally cone to mean representing the system using some kind of graphical notation, which is now almost based on notations in Unified Modeling Language (UML).

Models are used during the requirement engineering process to help to derive the requirements for a system, during the design process to describe the system to engineers implementing the system and after implementation to document the system's structure and structure and operation. The most important aspect of system model is that it leaves out detail .A model is an abstraction of system being studied rather than an alternative representation of that system. You may develop different models to represent the system from different perspective. For example:

1. An external perspective, where you model the context or environment of the system.
2. A structural perspective, where you model the organization of a system or the structure of the data that is processed by the system.
3. A behavioral perspective, where you model the dynamic behavior of the system and how it responds to events.

The UML has many diagrams types and so supports the creation of many different types of system model. However, a survey in 2007 showed that users of the UML thought that five diagram types could represent the essentials of a system:

1. Activity diagrams, which show the activities involved in process or in data processing.
2. Use case diagrams, which show the interactions between a system and its environment.
3. Sequence diagrams, which show interactions between actors and the system and between system components.
4. Class diagrams, which show the object classes in the system and the associations between these classes.
5. State diagrams, which show how the system reacts to internal and external events.

2.1 CONTEXT MODEL

- Context models are used to illustrate the boundaries of a system. Social and organizational concerns may affect the decision on where to position system boundaries.
- At the early stage in the requirement elicitation and analysis process one should decide on the boundaries of the system.
- It involves working with the stake holders to distinguish what the system.
- One makes this decision early in the process to limit the system cost and time needed for analysis.
- In some cases the boundary between a system and environment is relatively clear
  eg: automated system repacking existing manual or computerized system, the error of the new system is usually the same as the existing system

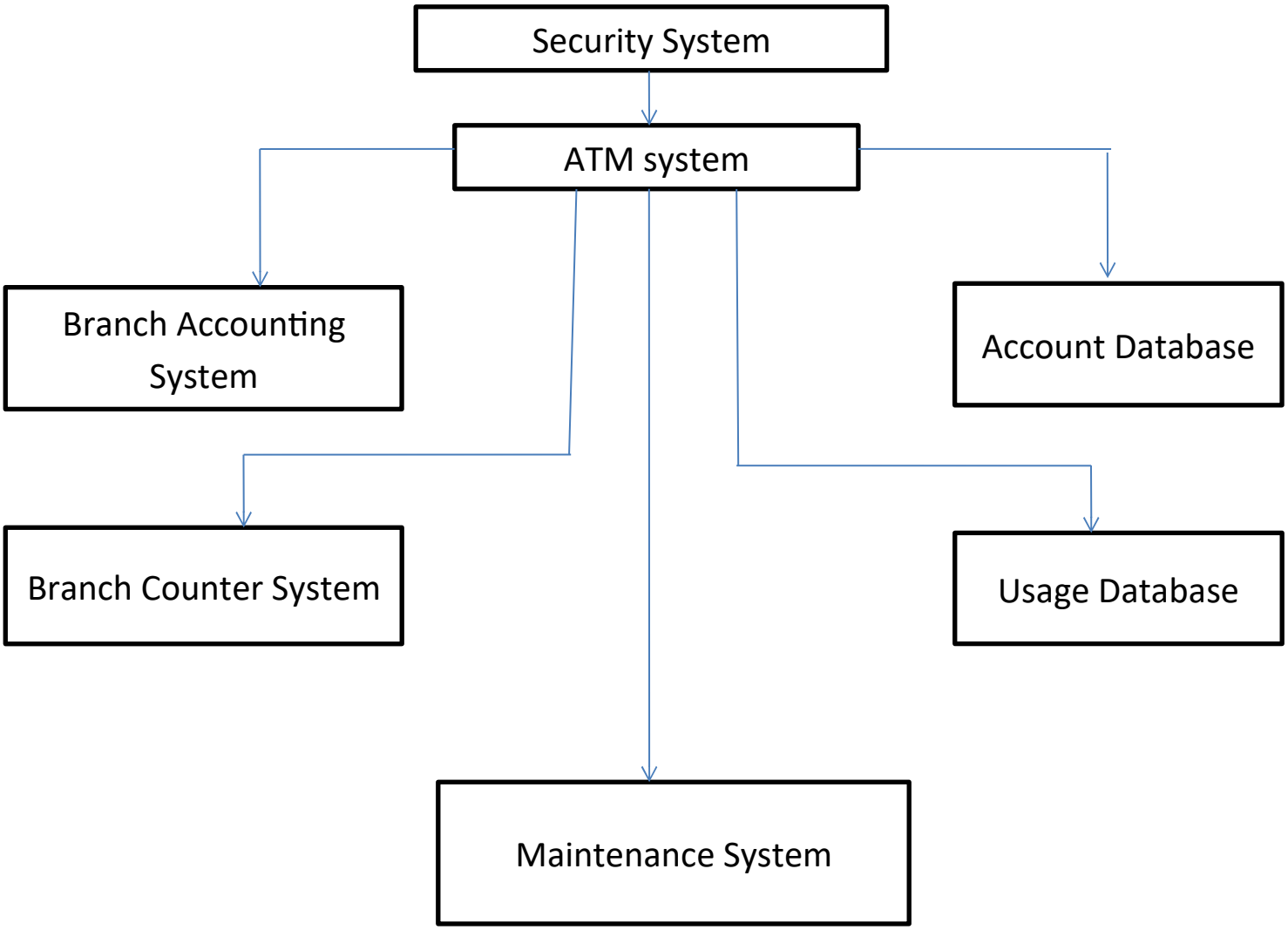

Fig: The Context of ATM system

2.2 Behavioral Model

- Behavioral models are used to describe the overall behavior system
- Here two types of behavioral models are discussed
    - data flow models— It models the data processing in the system
    - state machine system—It models how the system reacts to model

- These models may be used separately or together or together, depending on the type of system that is being developed.
- Most business system are primarily driven by data.
- They are controlled by the data inputs to the system with relatively little external event processing.

Data-flow model

- It is the system model that show a functional perspective where each transformation represents a single function or process.
- DFM are used to show how data flow through a sequence of processing steps.
  Eg: processing could be to filter duplicate records in a customer database
- The data is transferred at each step before moving on to the next step.
- The processing steps or transformation represents software process or functions when data flow diagrams are used to document a software design.

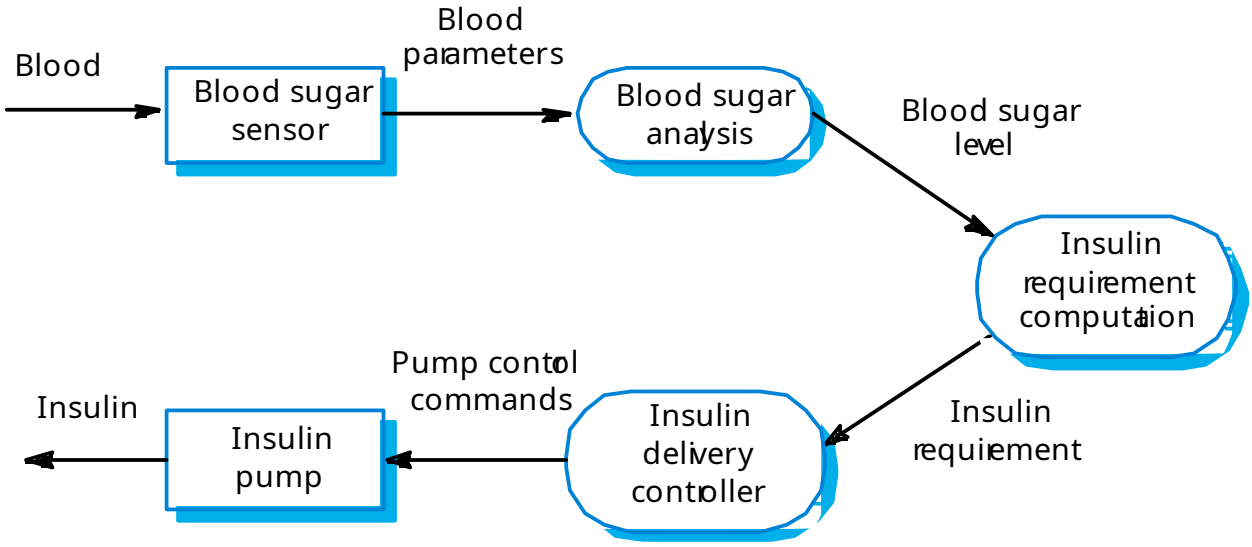

Fig: Insulin Pump DFD

State machine models

- These model the behavior of the system in response to external and internal events.
- They show the system's responses to stimuli so are often used for modeling real-time systems.
- State machine models show system states as

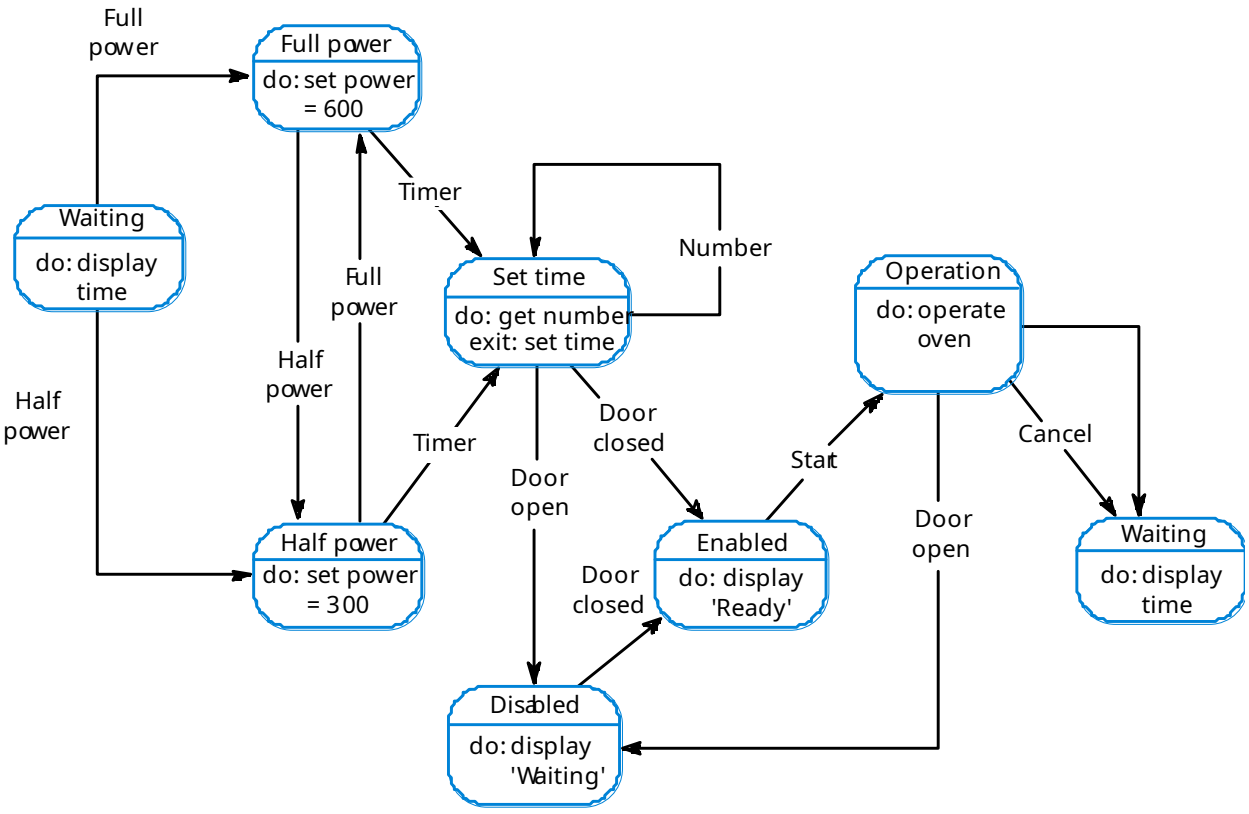

Fig: Microwave oven model

Microwave oven stimuli

Stimulus	description
Half Power	The user has pressed the half power button
Full power	The user has pressed the full power button
Timer	The user has pressed one of the timer button
Number	The user has pressed a numeric key
Door Open	The oven door switch is not closed
Door Closed	The oven door switch is closed
Start	The user has pressed the start button
Cancel	The user has pressed the cancel button

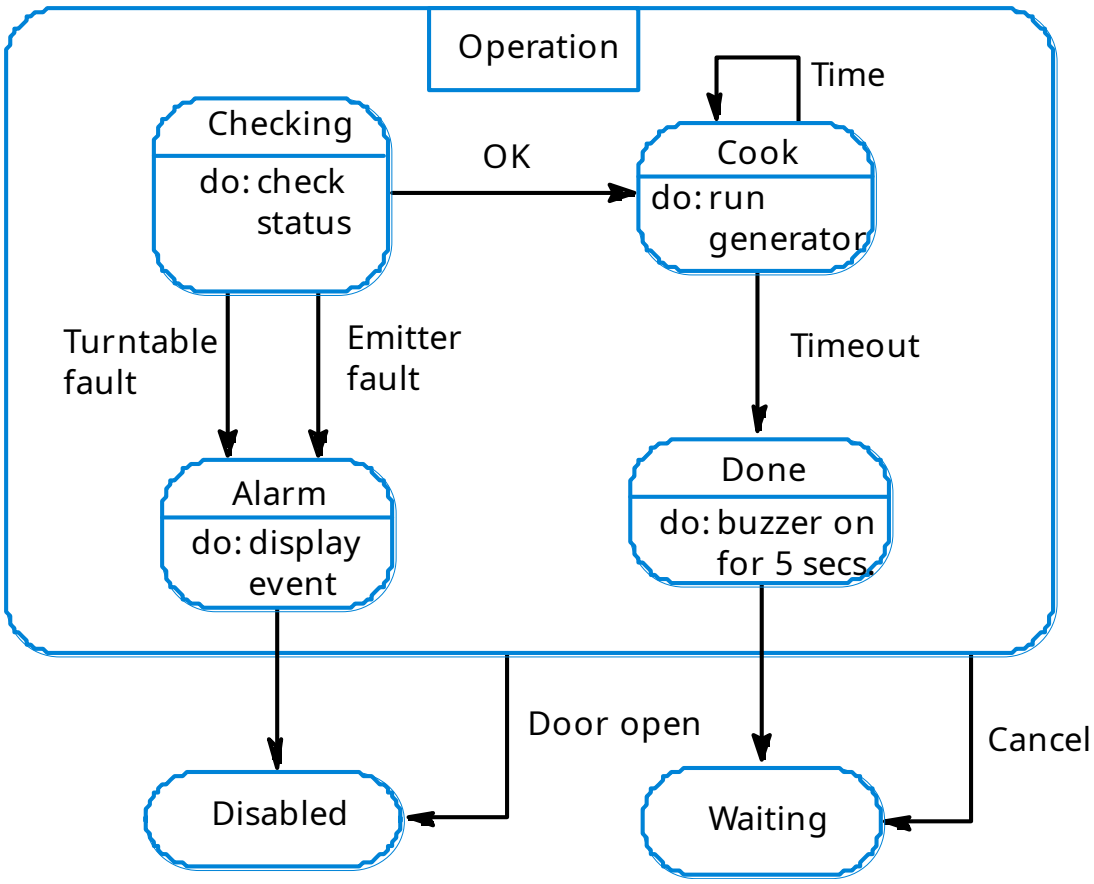

Fig: Microwave oven operation

2.3 Data Flow and Object Model

2.3.1 Data model

- It used to describe the logical structure of data processed by the system.
- It is an entity-relation-attribute model which sets out the entities in the system, the relationships between these entities and the entity attributes
- This model is widely used in database design. Can readily be implemented using relational databases.
- It has no specific notation provided in the UML but objects and associations can be used.

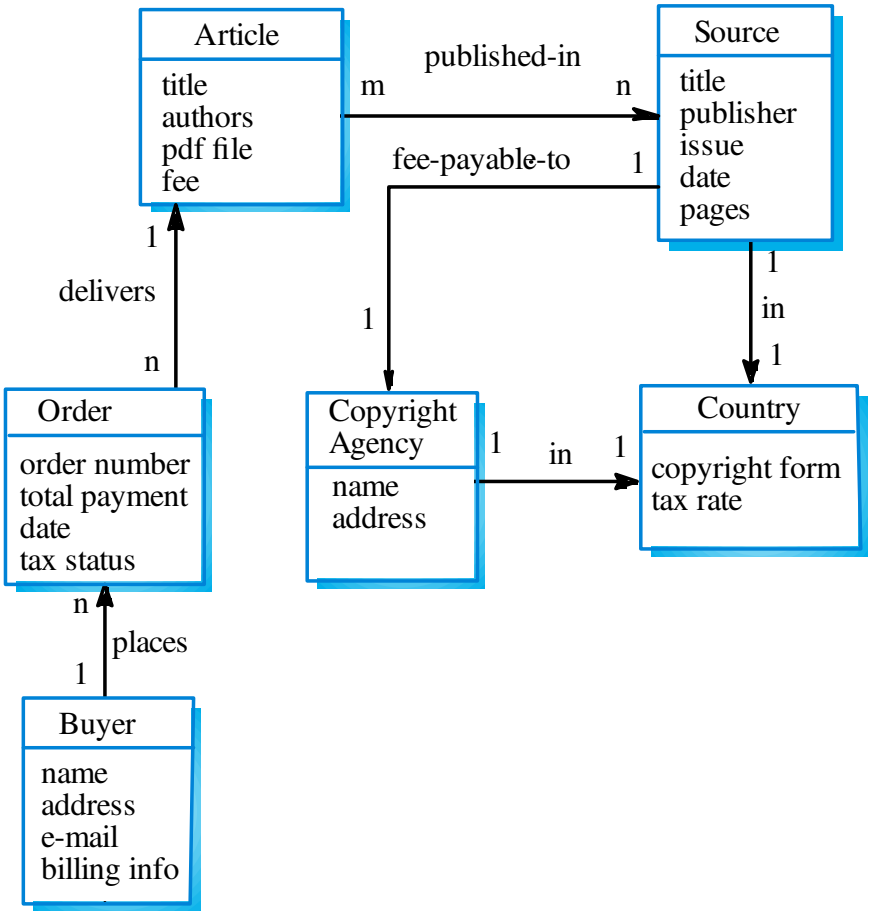

Fig: Library Symantec model

2.3.2 Object Model

- Object models describe the system in terms of object classes and their associations.
- An object class is an abstraction over a set of objects with common attributes and the services (operations) provided by each object.
- It is a natural ways of reflecting the real-world entities manipulated by the system
- More abstract entities are more difficult to model using this approach
- Object class identification is recognised as a difficult process requiring a deep understanding of the application domain
- Object classes reflecting domain entities are reusable across systems
- Various object models may be produced
    - Inheritance models;
    - Aggregation models;
    - Interaction models.

Inheritance model

- Organise the domain object classes into a hierarchy.
- Classes at the top of the hierarchy reflect the common features of all classes.
- Object classes inherit their attributes and services from one or more super-classes. These may then be specialised as necessary.
- Class hierarchy design can be a difficult process if duplication in different branches is to be avoided.

Object Model and the UML

- The UML is a standard representation devised by the developers of widely used object-oriented analysis and design methods.
- It has become an effective standard for object-oriented modelling.
- Notation
    - Object classes are rectangles with the name at the top, attributes in the middle section and operations in the bottom section;
    - Relationships between object classes (known as associations) are shown as lines linking objects;
- Inheritance is referred to as generalisation and is shown 'upwards' rather than 'downwards' in a hierarchy

Object Aggregation

- An aggregation model shows how classes that are collections are composed of other classes.
- Aggregation models are similar to the part-of relationship in semantic data models.

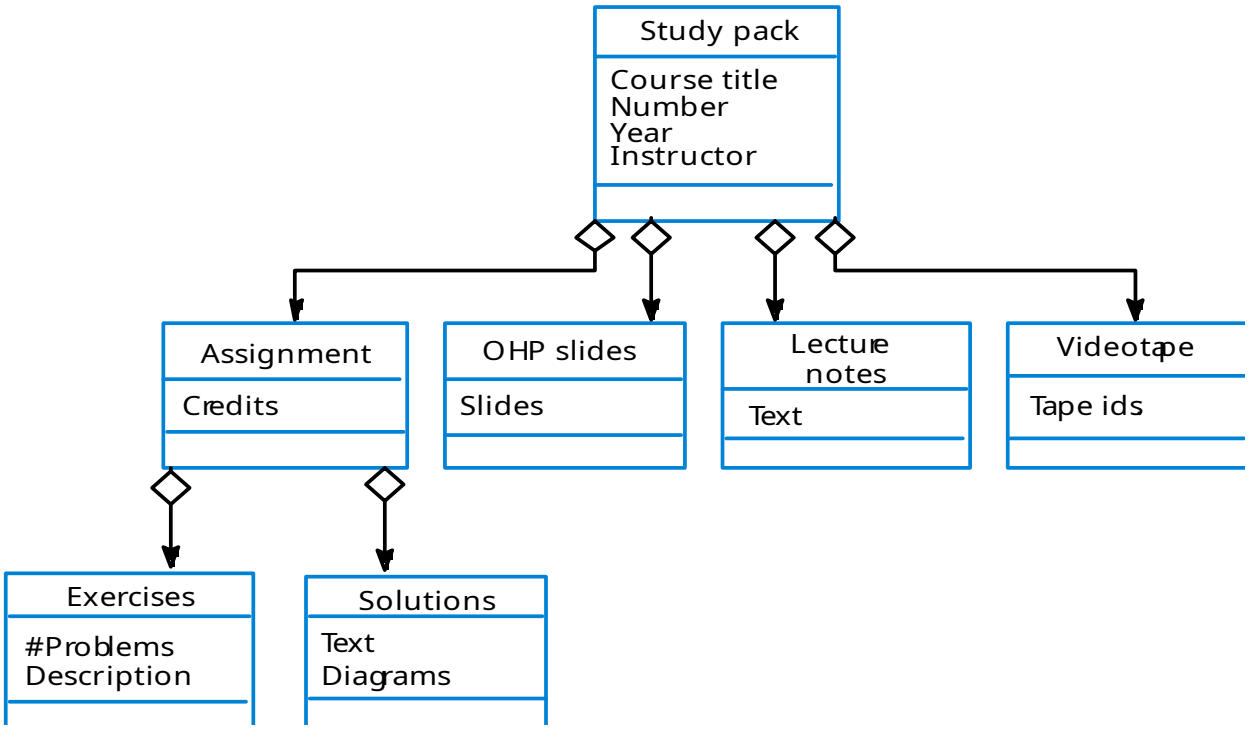

Fig:Object Aggregation

2.4 Structured Methods

- Structured methods incorporate system modelling as an inherent part of the method.
- Methods define a set of models, a process for deriving these models and rules and guidelines that should apply to the models.
- CASE tools support system modelling as part of a structured method.
- This model of software display the organization of a system in terms of components that make up that system and their relationship.
- They 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.

Weakness of the method:

- They do not model non-functional system requirements.
- They do not usually include information about whether a method is appropriate for a given problem.
- They may produce too much documentation
- The system models are sometimes too detailed and difficult for users to understand.