

System Modeling

Lecture # 24



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

- Class diagrams are used when developing an object-oriented system model to show the classes in a system and the associations between these classes.
- It is static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.
- A UML class diagram is made up of: A set of classes and A set of relationships between classes
- A class is a description of a group of objects all with similar roles in the system, which consists of:
 - **Structural features** (attributes) define what objects of the class "know"
 - **Behavioral features** (operations) define what objects of the class "can do"

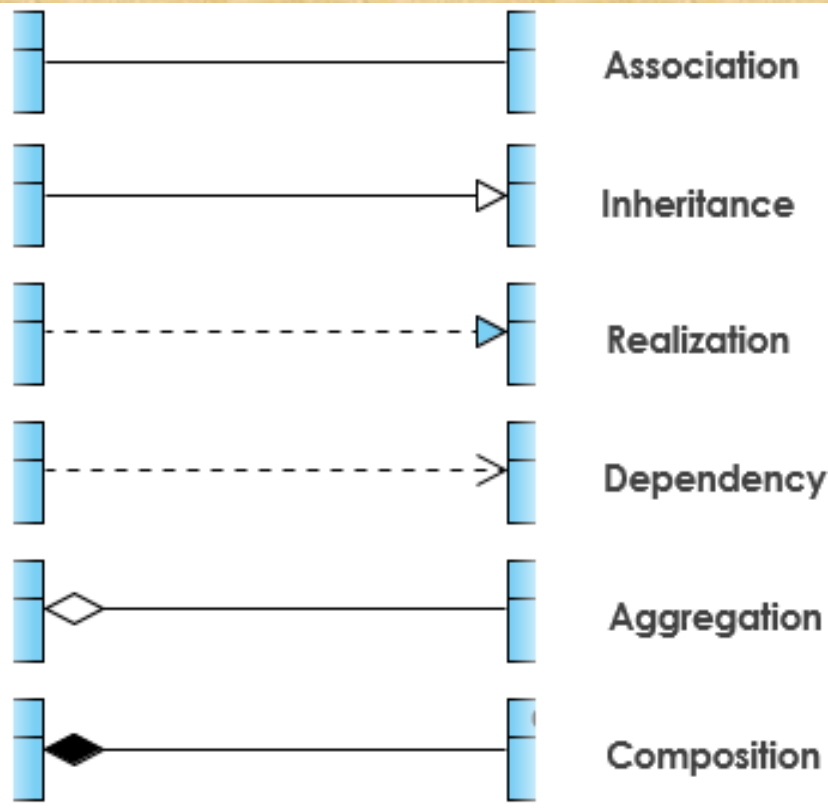


Class Diagram

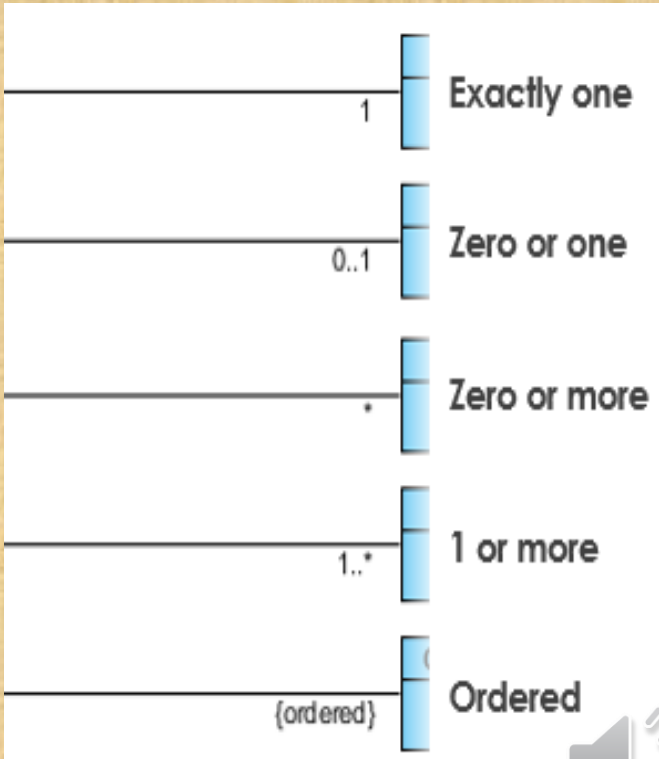
Class

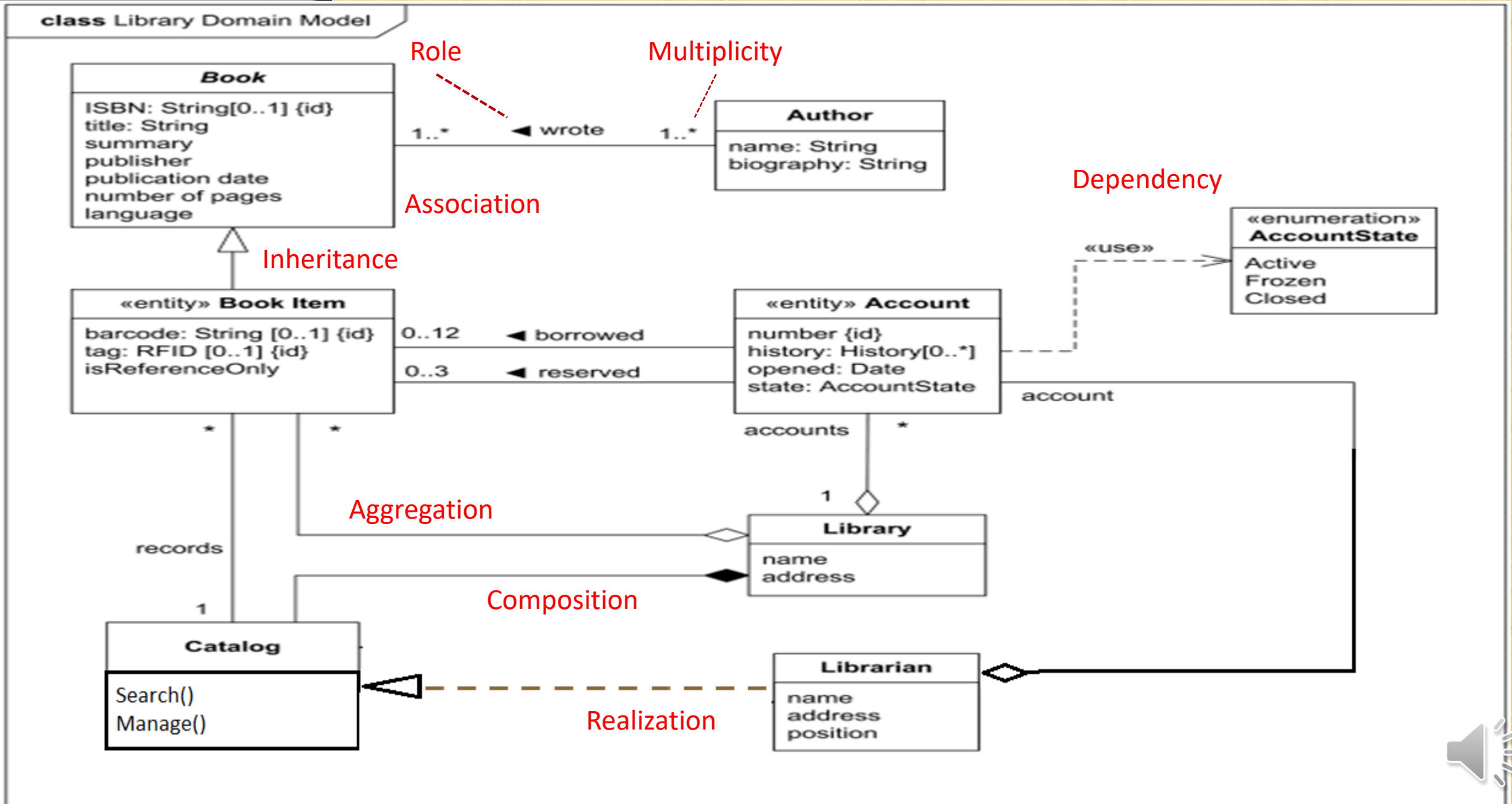
Name	Order
Attributes	<ul style="list-style-type: none"> - deliveryDate: Date - orderNumber: int - placementDate: Date - taxes: Currency - total: Currency
Operations	<ul style="list-style-type: none"> # calculateTaxes(Country, State): Currency # calculateTotal(): Currency getTaxEngine() {visibility=implementation}

Relationship between Classes



Cardinality





Behavioral Models

-
- Behavioral models are models of the dynamic behavior of a 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, although this is not always the case

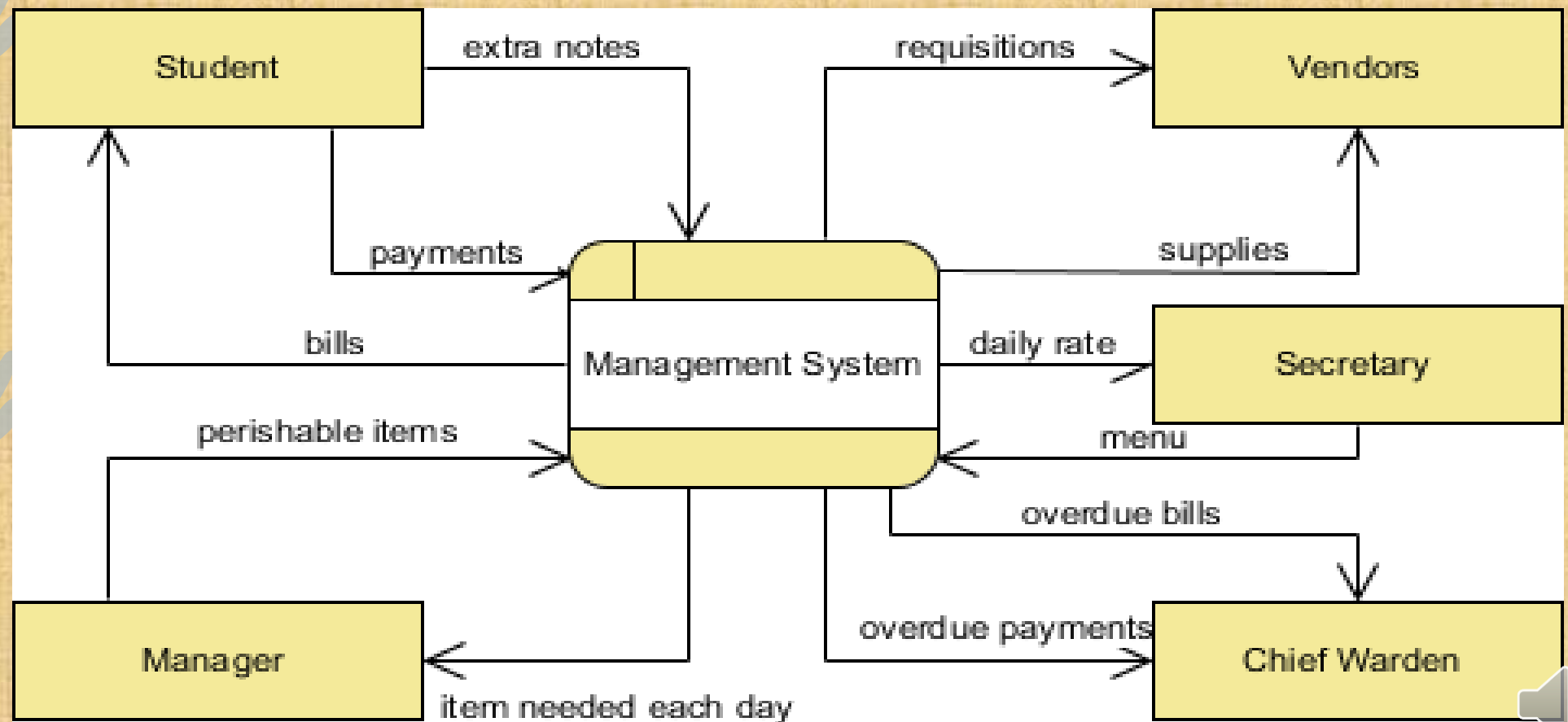


Data-Driven Modeling

- Many business systems are data-processing systems that are primarily driven by data. They are controlled by the data input to the system, with relatively little external event processing.
- Data-driven models show the sequence of actions involved in processing input data and generating an associated output.
- They are particularly useful during the analysis of requirements as they can be used to show end-to-end processing in a system.

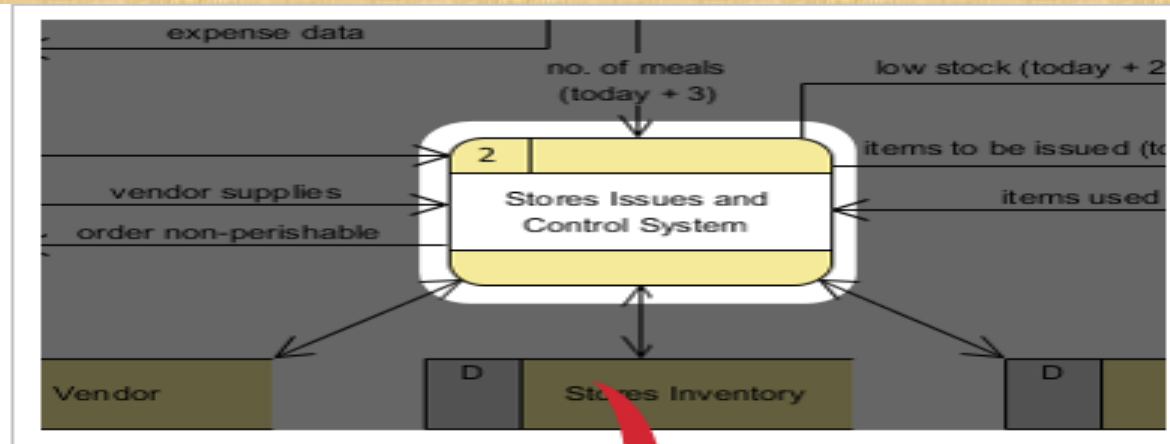


Data Flow Diagram (Level 0 or Context Level)

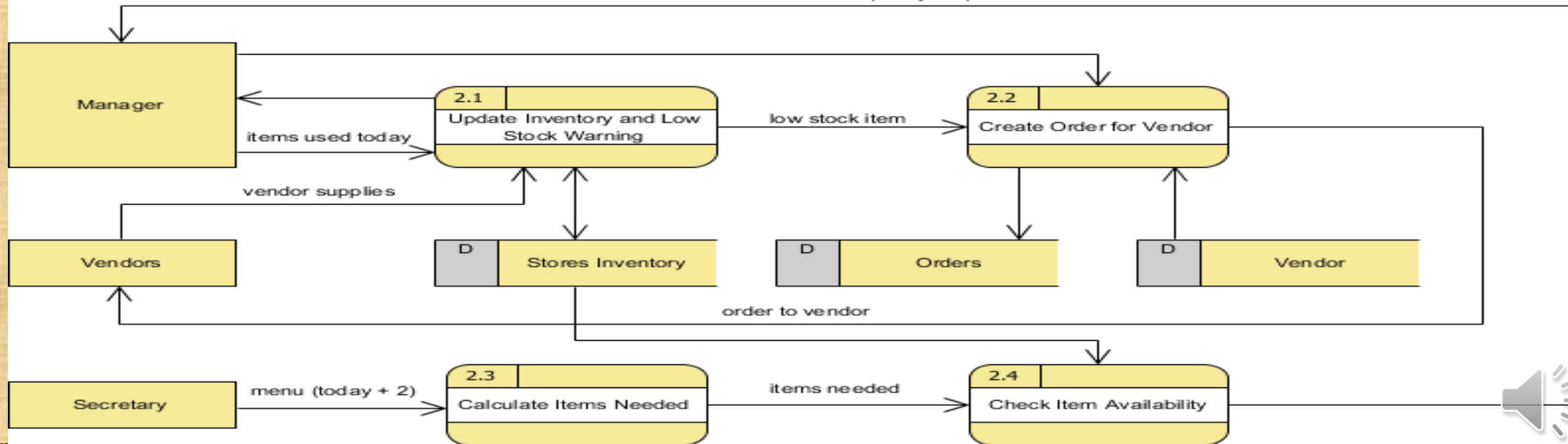




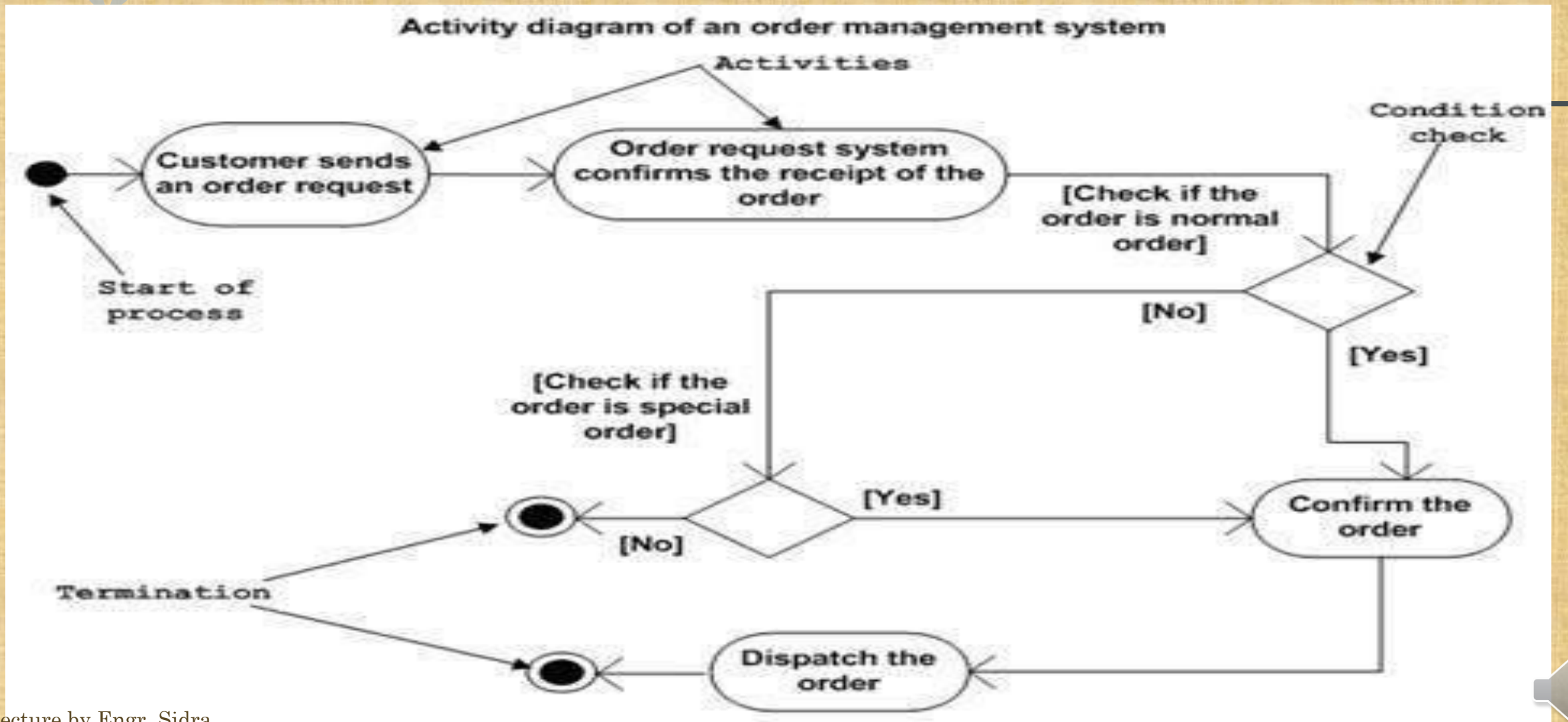
Data Flow Diagram (Level 2)



low stock items (today + 2)



Activity Diagram

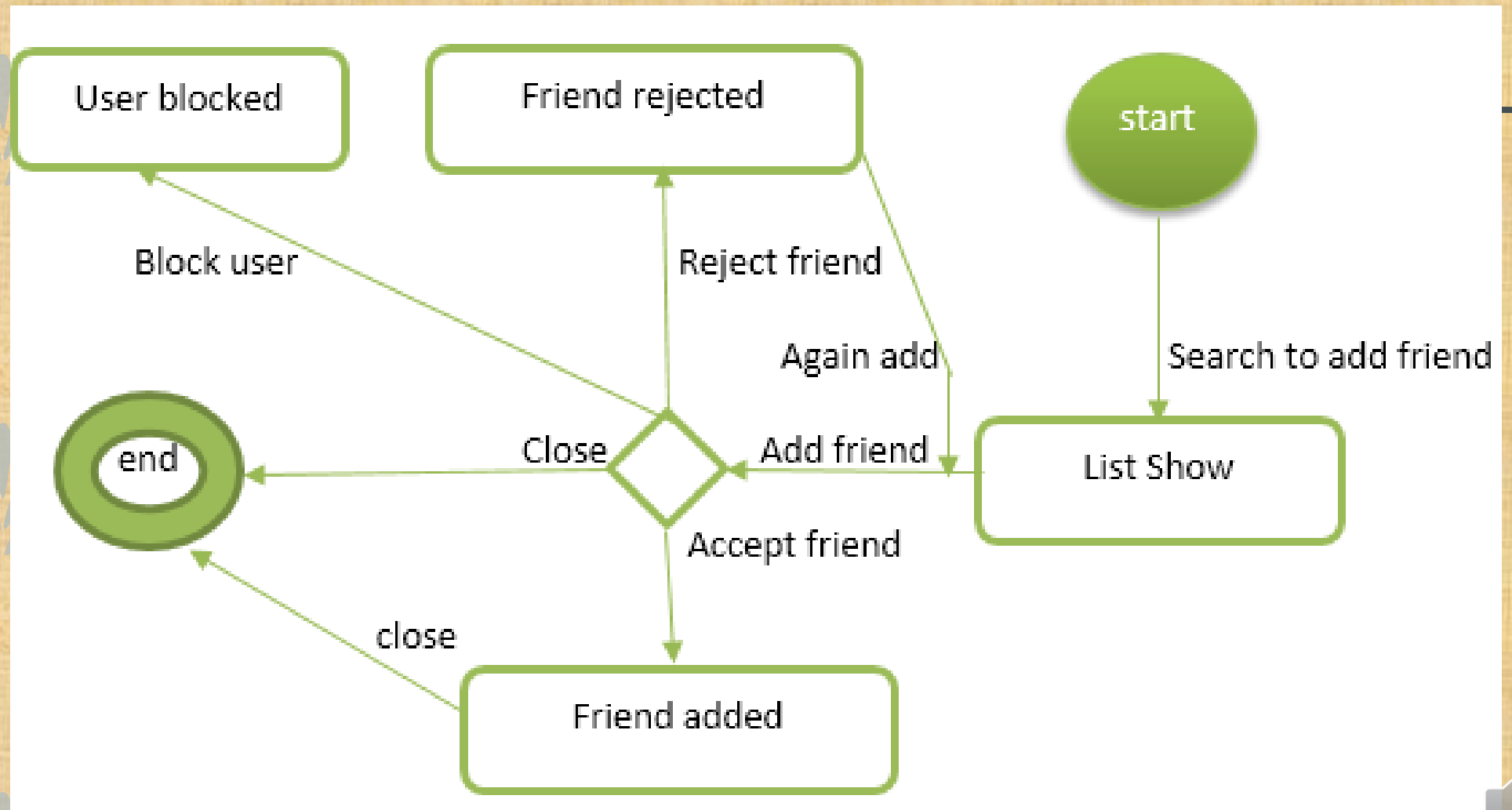


Event-Driven Modeling

- Real-time systems are often event-driven, with minimal data processing. For example, a landline phone switching system responds to events such as ‘receiver off hook’ by generating a dial tone.
- Event-driven modeling shows how a system responds to external and internal events.
- It is based on the assumption that a system has a finite number of states and that events (stimuli) may cause a transition from one state to another.



State Transition Diagram



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 modelling real-time systems.
- State machine models show system states as nodes and events as arcs between these nodes. When an event occurs, the system moves from one state to another.
- Statecharts are an integral part of the UML and are used to represent state machine models



State Machine Models

