



Data Dictionary

Data Dictionary

- ▶ A tool for recording and processing information (metadata) about the data that an organisation uses.
- ▶ A central catalogue for metadata.
- ▶ Can be integrated within the DBMS or be separate.
- ▶ May be referenced during system design, programming, and by actively-executing programs.
- ▶ Can be used as a repository for common code (e.g. library routines).

Benefits of a DDS

3

Benefits of a DDS are mainly due to the fact that it is a central store of information about the database.

Benefits include

- ▶ improved documentation and control
- ▶ consistency in data use
- ▶ easier data analysis
- ▶ reduced data redundancy
- ▶ simpler programming
- ▶ the enforcement of standards
- ▶ better means of estimating the effect of change.

DDS Facilities

A DDS should provide two sets of facilities:

- To record and analyse data requirements independently of how they are going to be met conceptual data models (entities, attributes, relationships).
- To record and design decisions in terms of database or file structures implemented and the programs which access them internal schema.

One of the main functions of a DDS is to show the relationship between the conceptual and implementation views. The mapping should be consistent - inconsistencies are an error and can be detected here.

DD Information

- ▶ The names associated with that element (aliases)
- ▶ A description of the data element in natural language.
- ▶ Details of ownership.
- ▶ Details of users that refer to the element.
- ▶ Details of the systems and programs which refer to or update the element.
- ▶ Details on any privacy constraints that should be associated with the item.
- ▶ Details about the data element in data processing systems, such as the length of the data item in characters, whether it is numeric alphabetic or another data type, and what logical files include the data item.
- ▶ The security level attached to the element in order to control access.
- ▶ The total storage requirement.
- ▶ The validation rules for each element (e.g. acceptable values).
- ▶ Details of the relationship of the data items to others.

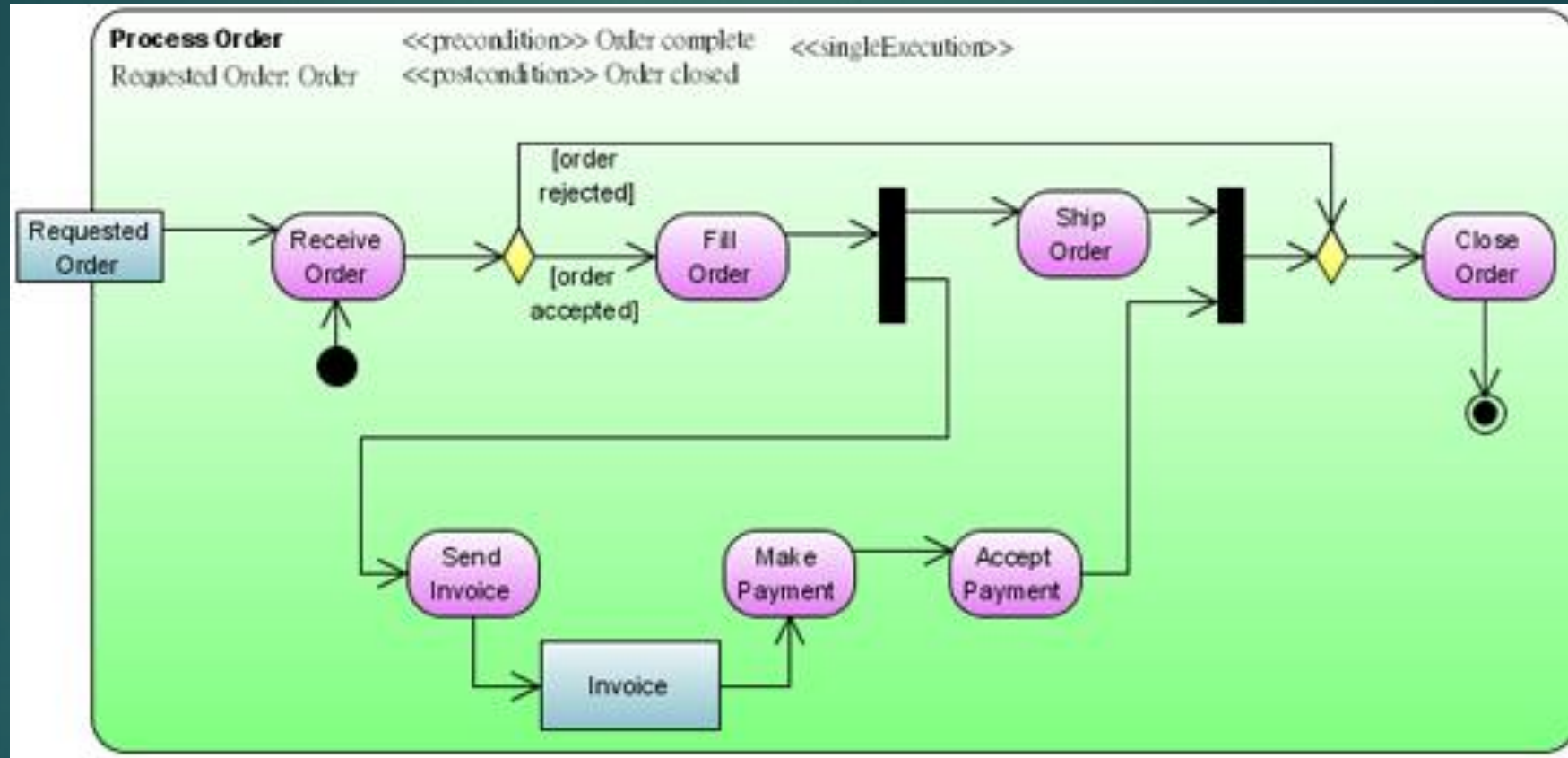
Behavioural modeling

- ▶ behavioral diagrams visualize, specify, construct, and document the dynamic aspects of a system.
- ▶ The behavioral diagrams are categorized as follows: use case diagrams, interaction diagrams, state-chart diagrams, and activity diagrams

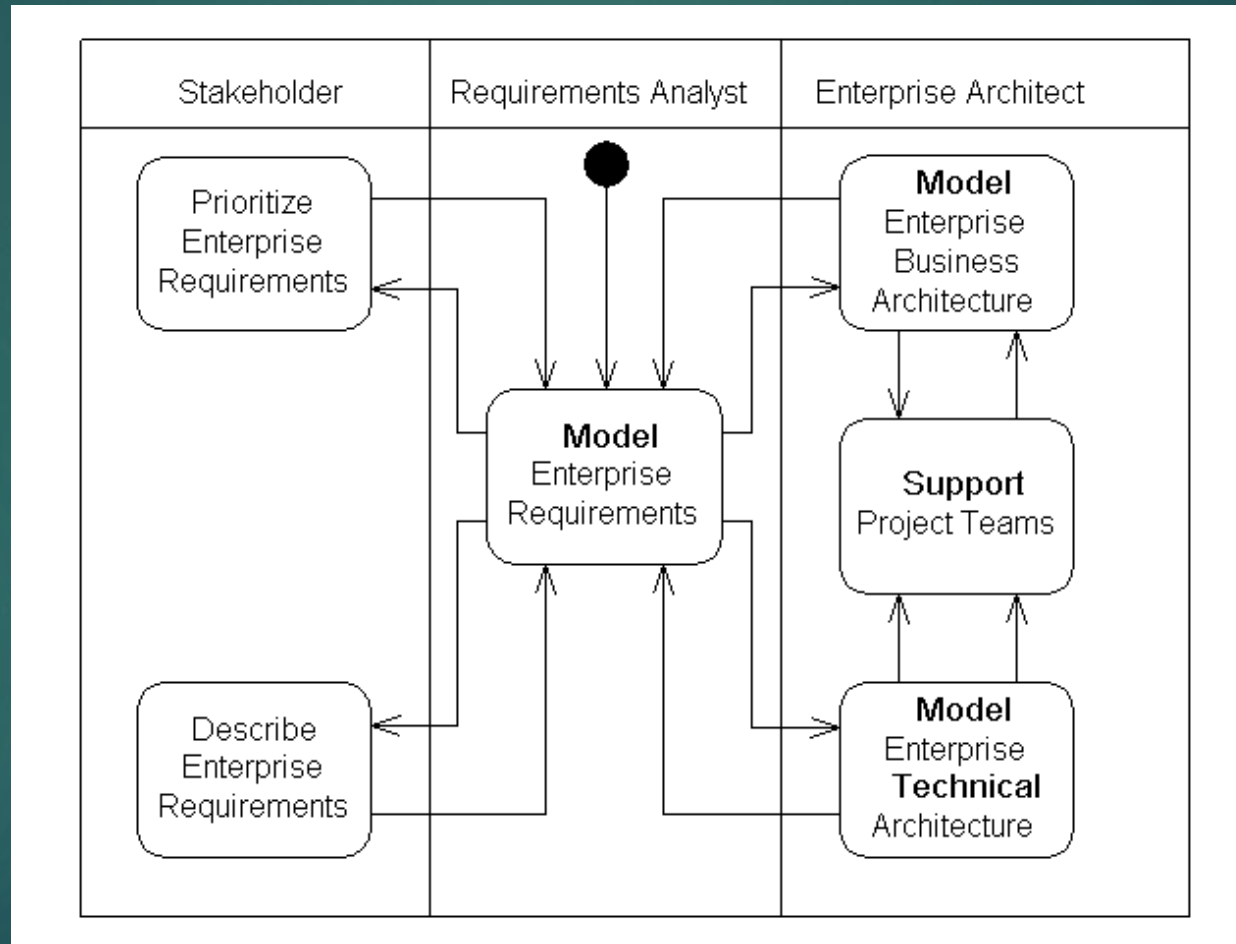
Activity diagram

- ▶ [UML 2 Activity diagrams](#) helps to describe the flow of control of the target system, such as the exploring complex business rules and operations, describing the use case also the business process. It is object-oriented equivalent of flow charts and data-flow diagrams (DFDs).

Activity diagram



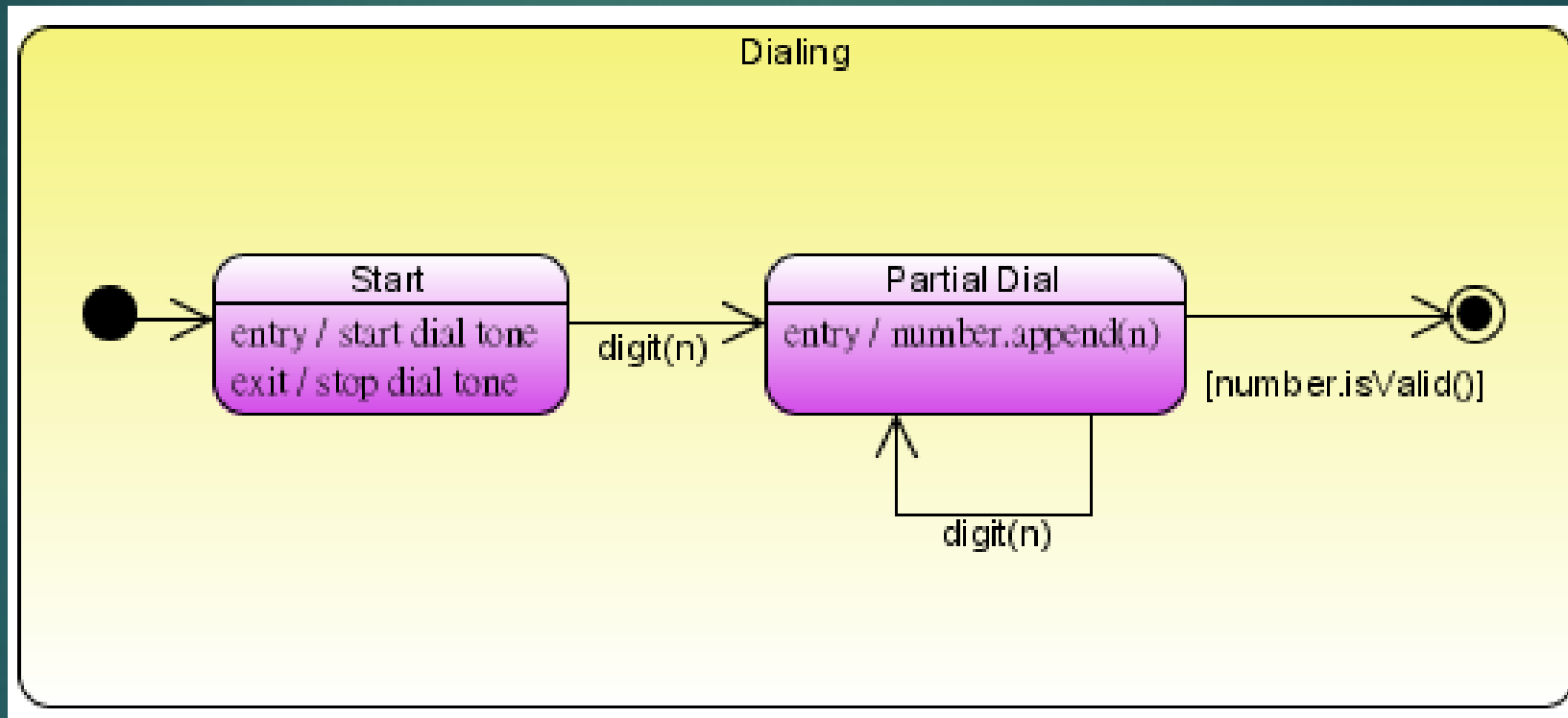
Activity diagram



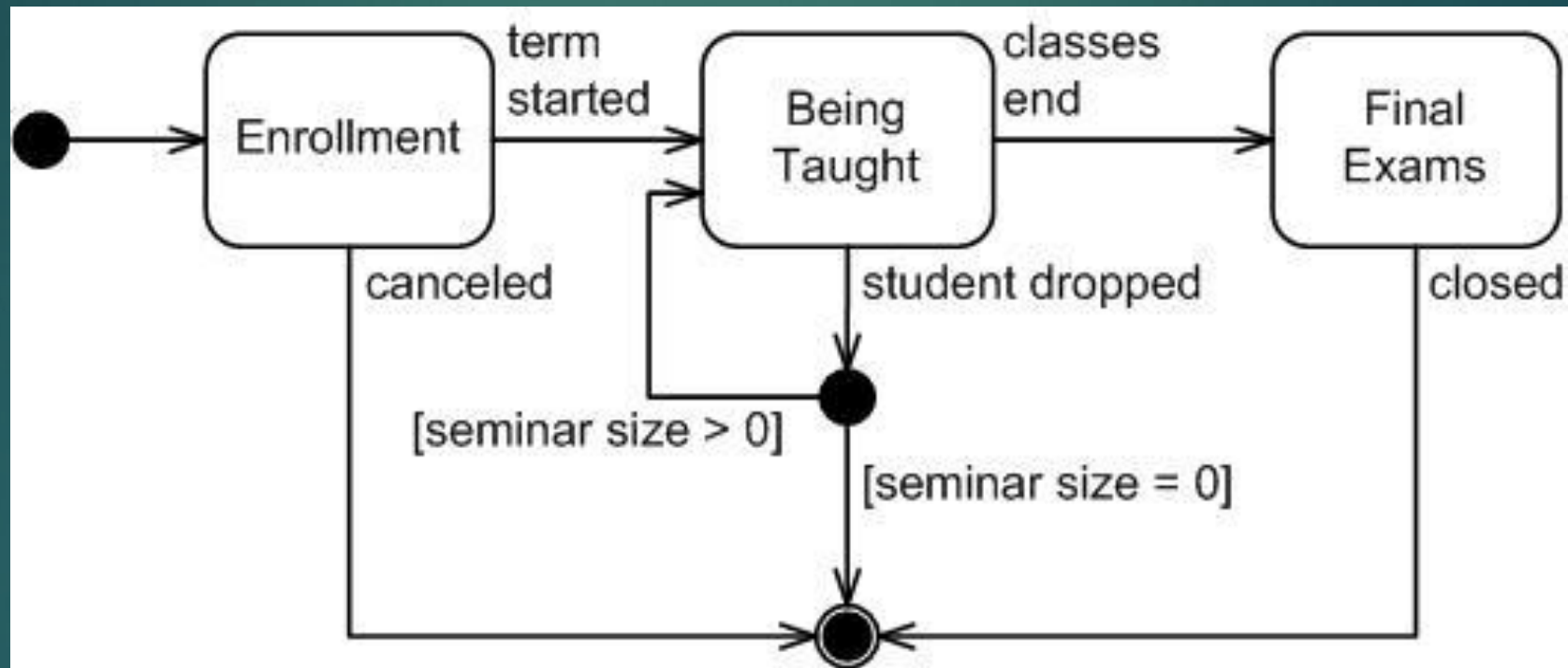
State machine diagram

- ▶ [UML 2 State machine diagrams](#) can show the different states of an entity also how an entity responds to various events by changing from one state to another. The history of an entity can best be modeled by a finite state diagram.

State machine diagram



State machine diagram



Use cases diagram

UML 2 Use cases diagrams describes the behavior of the target system from an external point of view. Use cases describe "the meat" of the actual requirements.

- ▶ **Use cases.** A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.
- ▶ **Actors.** An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.
- ▶ **Associations.** Associations between actors and use cases are indicated by solid lines. An association exists whenever an actor is involved with an interaction described by a use case.

Use case

- ▶ A use case describes the sequence of actions a system performs yielding visible results. It shows the interaction of things outside the system with the system itself. Use cases may be applied to the whole system as well as a part of the system.

Actor

- ▶ An actor represents the roles that the users of the use cases play. An actor may be a person (e.g. student, customer), a device (e.g. workstation), or another system (e.g. bank, institution).
- ▶ The following figure shows the notations of an actor named Student and a use case called Generate Performance Report.



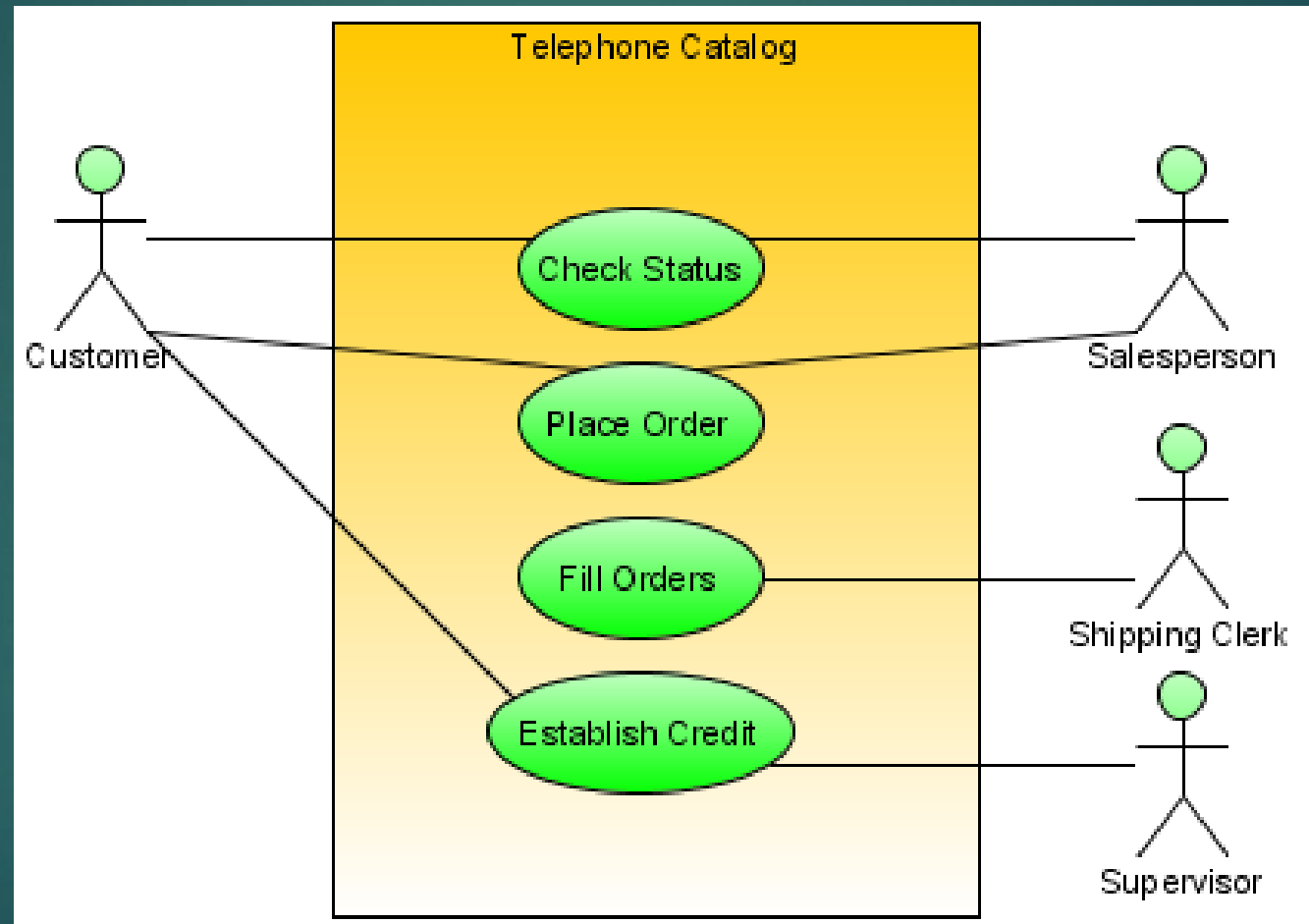
Use case diagrams are used –

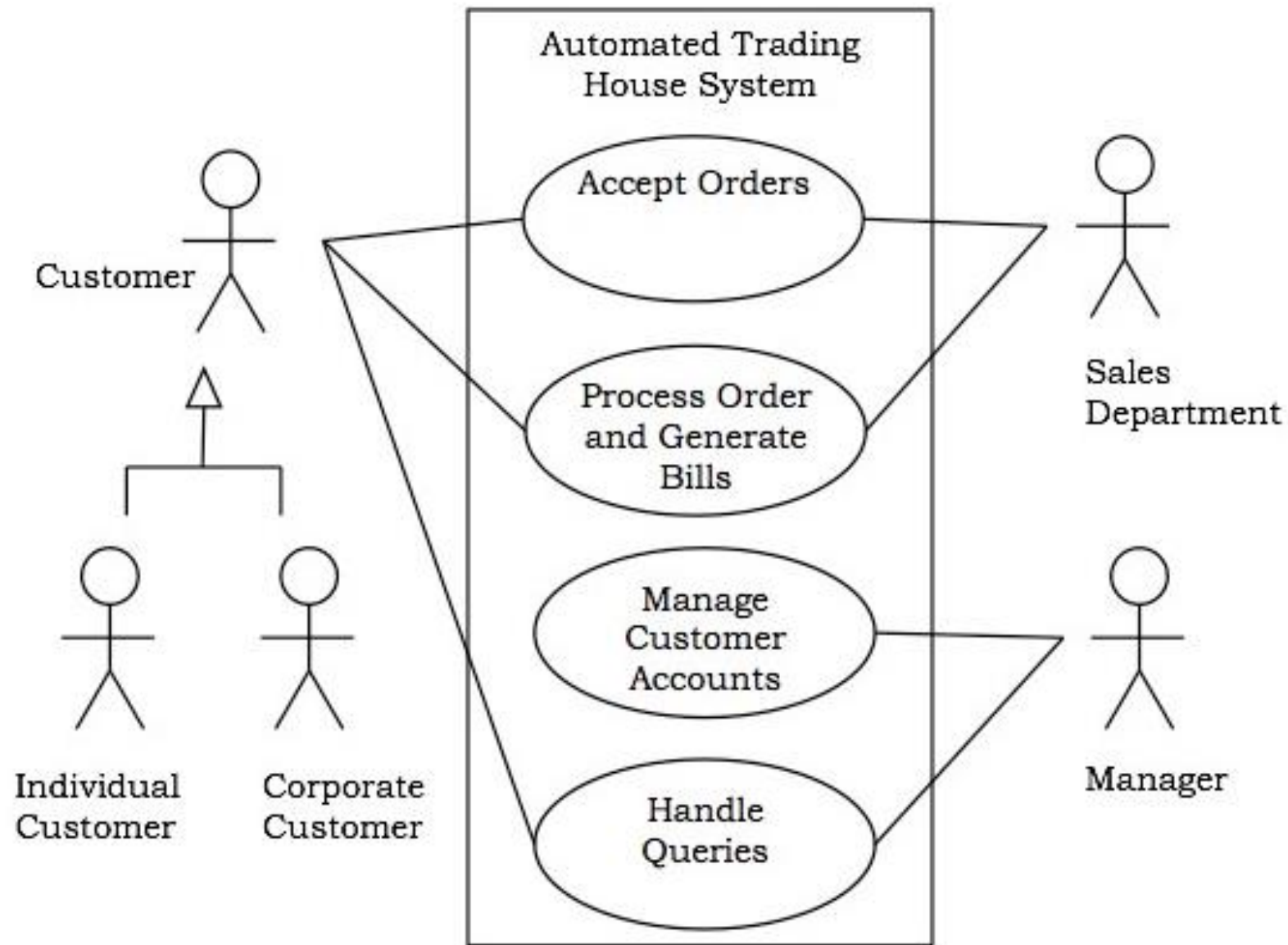
- ▶ To model the context of a system by enclosing all the activities of a system within a rectangle and focusing on the actors outside the system by interacting with it.
- ▶ To model the requirements of a system from the outside point of view.

Use cases diagram



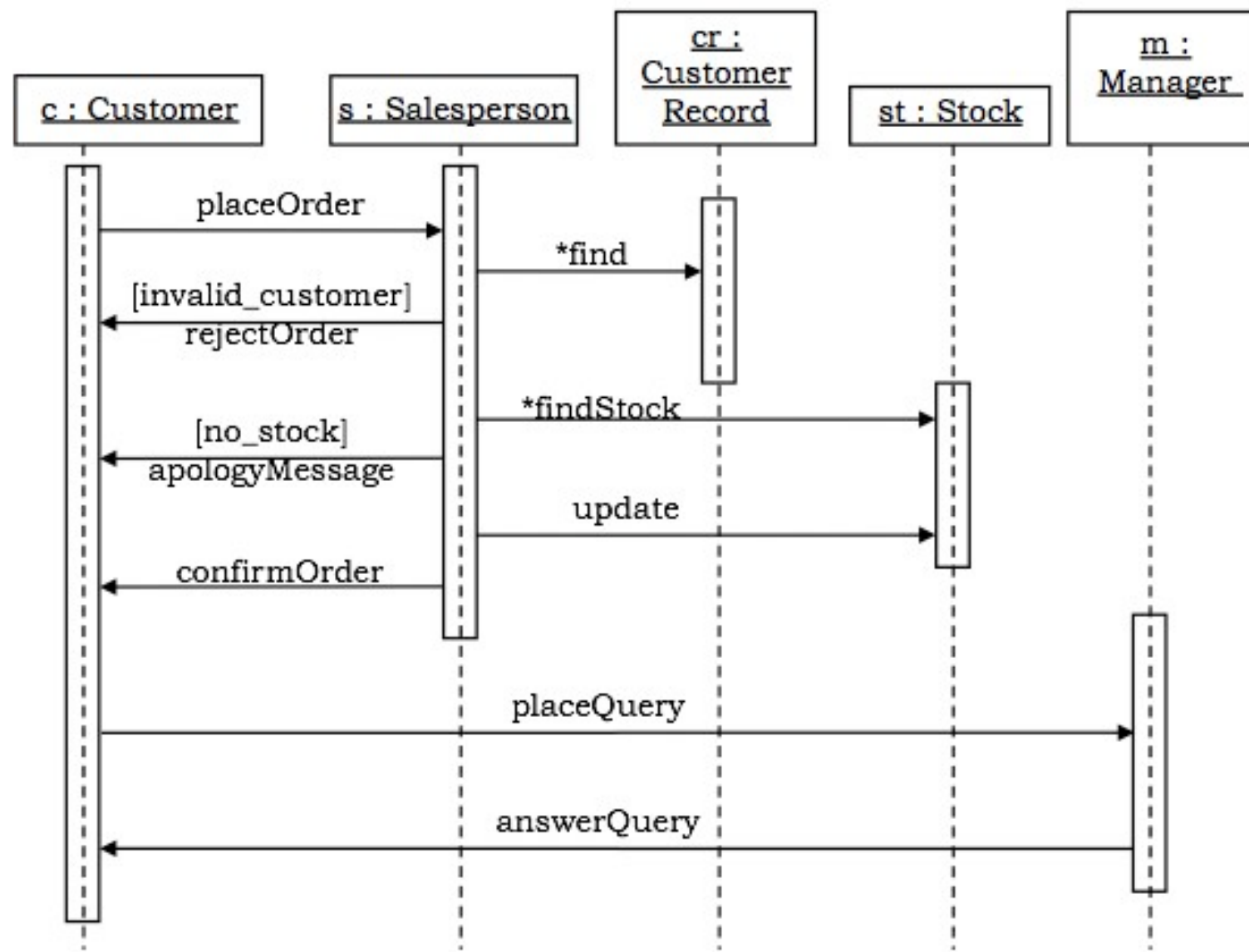
Use cases diagram



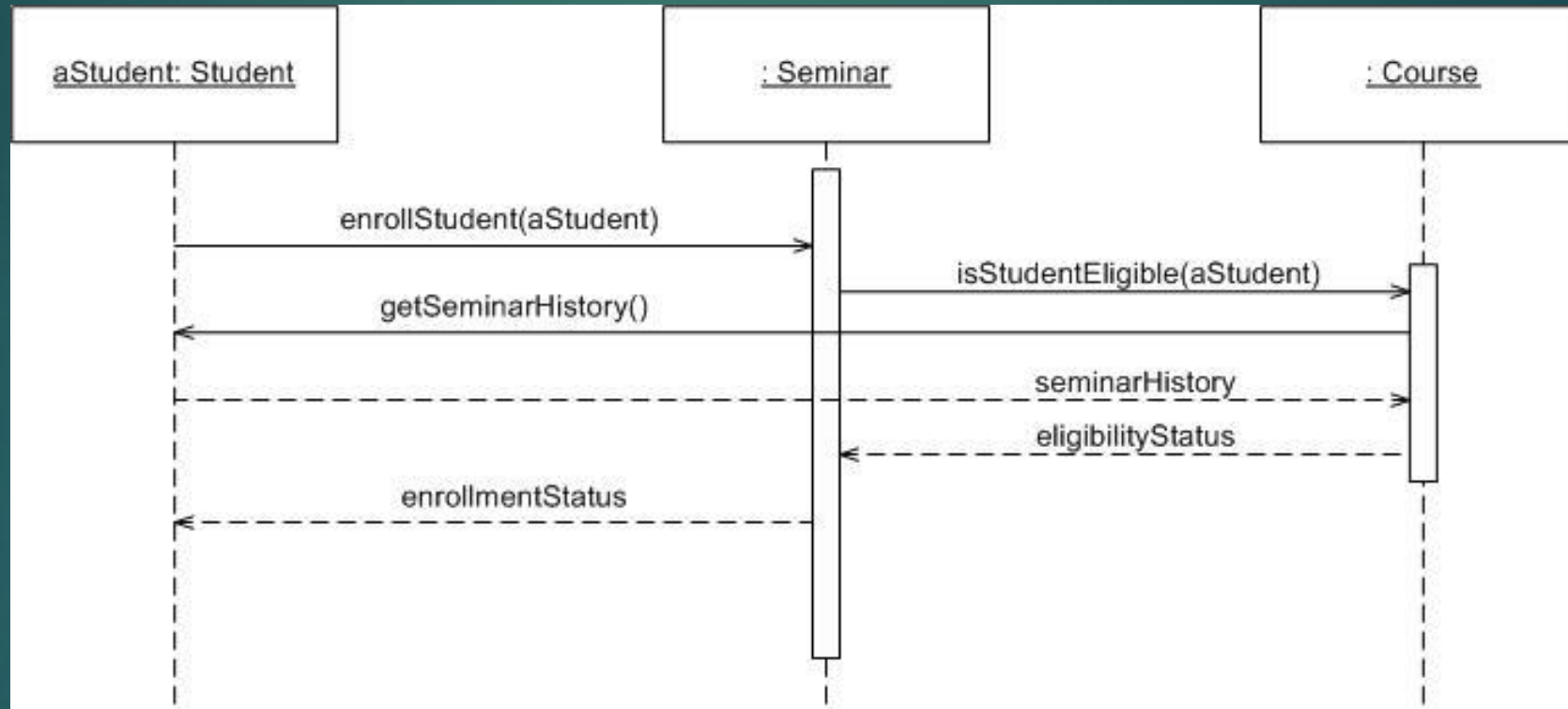


Sequence diagram

- ▶ [UML 2 Sequence diagrams](#) models the collaboration of objects based on a time sequence. It shows how the objects interact with others in a particular scenario of a use case.
- ▶ Sequence diagrams are interaction diagrams that illustrate the ordering of messages according to time.
- ▶ **Notations** – These diagrams are in the form of two-dimensional charts. The objects that initiate the interaction are placed on the x-axis. The messages that these objects send and receive are placed along the y-axis, in the order of increasing time from top to bottom.



Sequence diagram



Collaboration Diagrams

- ▶ Collaboration diagrams are interaction diagrams that illustrate the structure of the objects that send and receive messages.
- ▶ **Notations** – In these diagrams, the objects that participate in the interaction are shown using vertices. The links that connect the objects are used to send and receive messages. The message is shown as a labeled arrow.
- ▶ **Example** – Collaboration diagram for the Automated Trading House System is illustrated in the figure below

