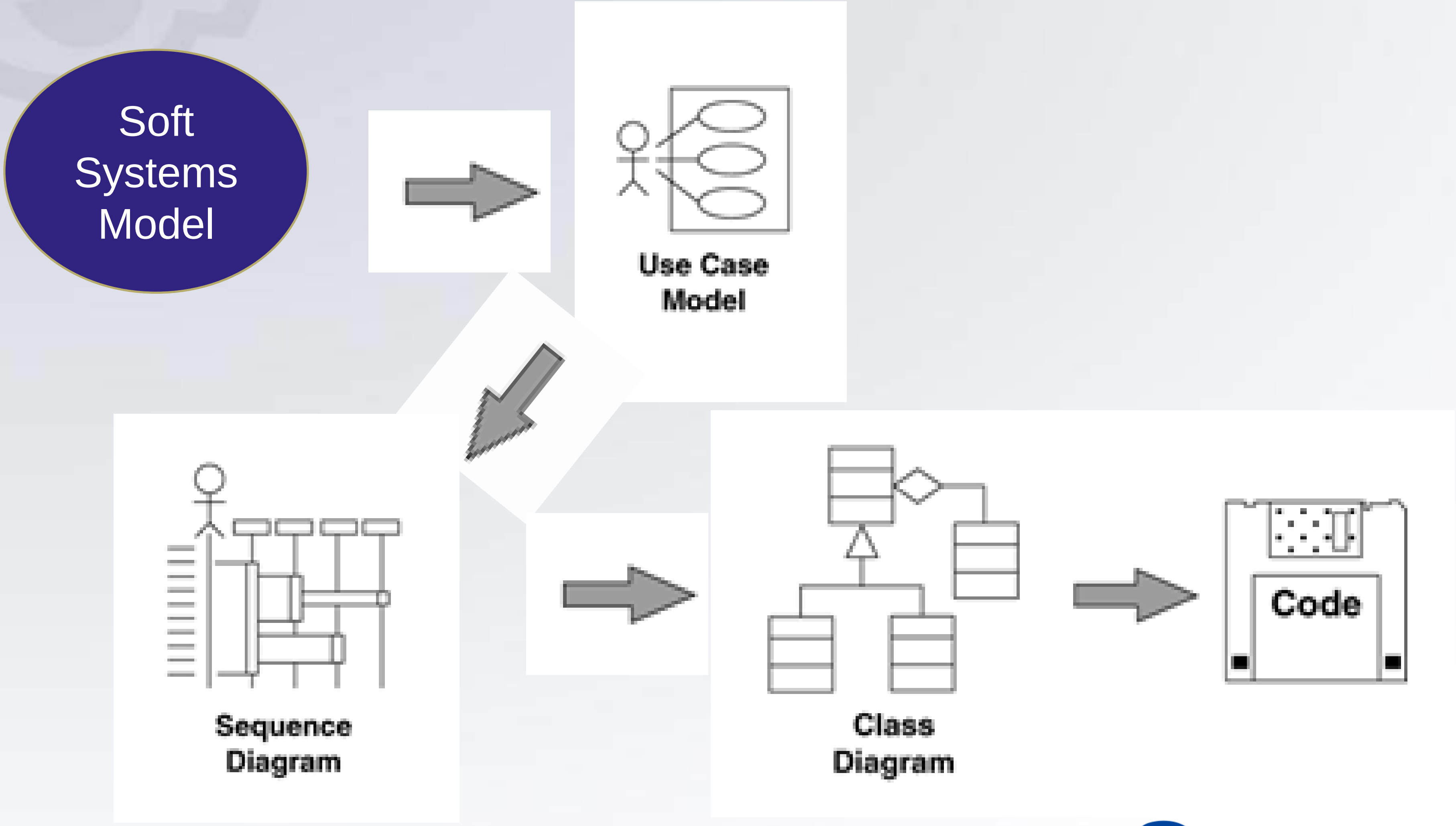
ENTERPRISE
ARCHITECT

sequence diagram

Department of Software

Beginnings of a Method



Soft Systems Model

Use Case Model

Sequence Diagram

Class Diagram

Code

Types of Diagrams

- **Structural Diagrams –** focus on static aspects of the software system.
  - Class, Object, Component, Deployment.
- **Behavioral Diagrams –** focus on dynamic aspects of the software system.
  - Use-case, Interaction, State Chart, Activity.

Types of Diagrams

Sequence diagram

A ***Sequence* diagram** is one of four types of *Interaction* diagram. (The other three are Timing Diagram, Interaction Overview Diagram and Communication Diagram.)

A Sequence diagram is a structured representation of behavior as a series of sequential steps over time. It is used to depict work flow, message passing and how elements in general cooperate over time to achieve a result.

It mainly emphases on time ordering and messages. It is used to illustrate the dynamic view of system.

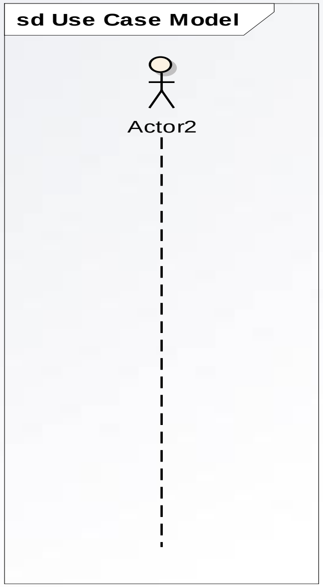
In simpler words, a sequence diagram shows different parts of a system work in a ‘sequence’ to get something done.

Sequence Diagram Elements

The sequence diagram is made up of collection of participants or objects. Participants are system parts that interact each other during sequence diagram. The participants interact with each other by sending and receiving message

1. Actor:

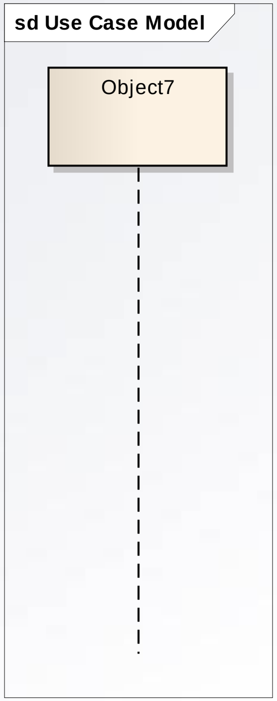
An ***Actor*** is a user of the system. user can mean a human user, a machine, or even another system or subsystem in the model. Anything that interacts with the system from the outside or system boundary is termed an Actor.



Sequence Diagram Elements

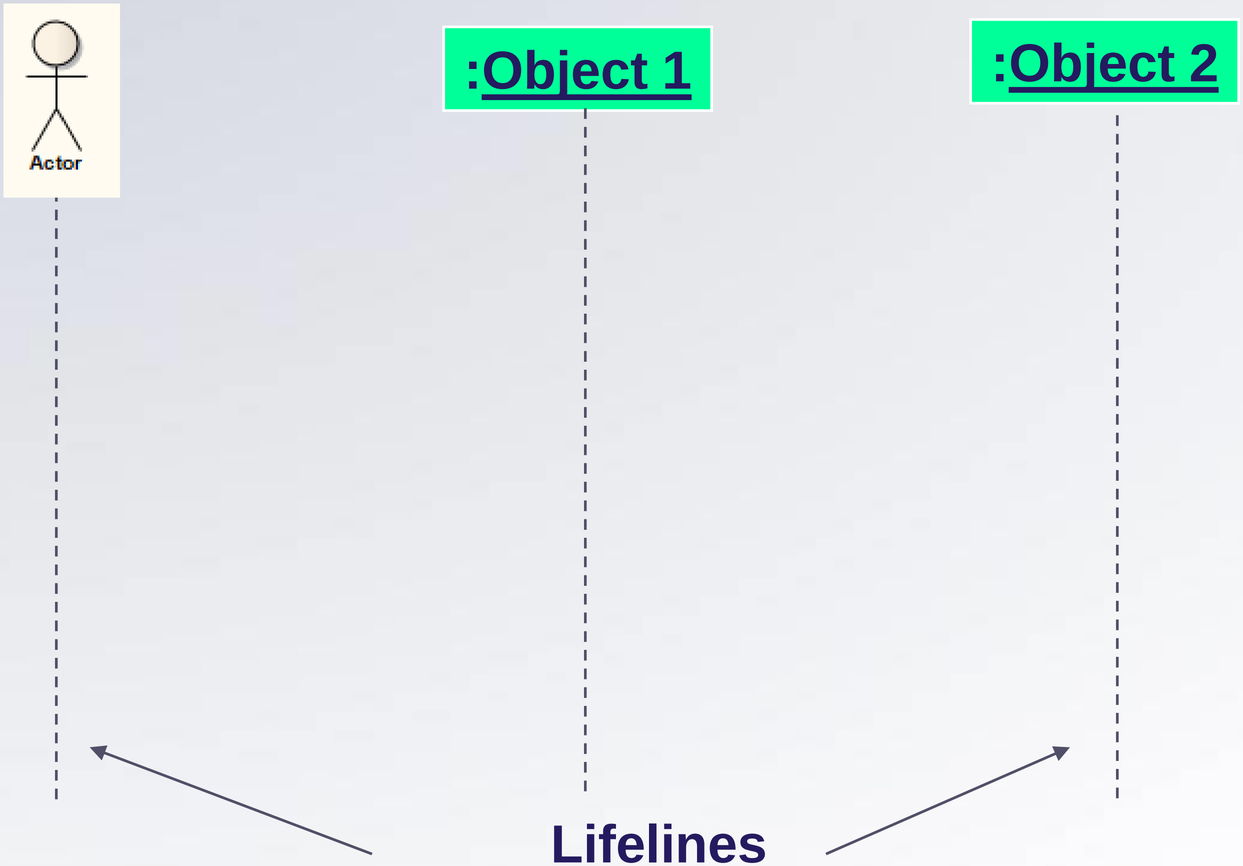
2. Lifeline:

Lifeline represents the existence of an object over a period of time. It is represented by vertical dashed line under object or actor.



Most objects that appeared in ‘Interaction diagram’ will be in existence for the duration of an interaction. So, these objects are aligned at top at top of diagram with their lifeline from top to bottom of diagram.

Sequence Diagram Elements



Lifelines

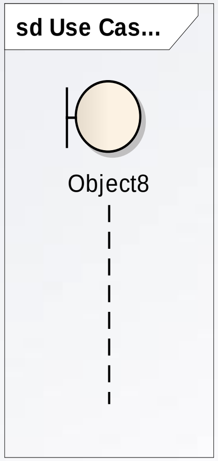
Identify the existence of the object over time.

Sequence Diagram Elements

3. Boundary:

A ***boundary*** is a user interface screen, and is used in the conceptual phase to capture users interacting with the system at a screen level.

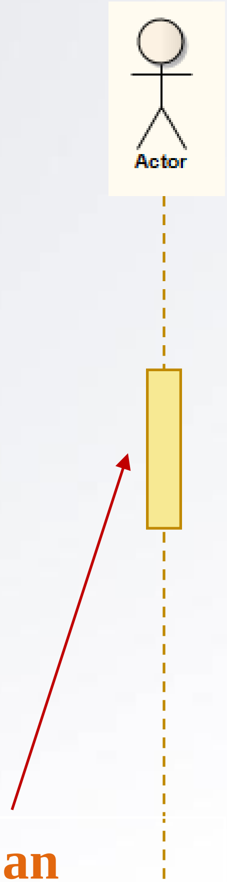
A lifeline with a boundary element indicates a system boundary/ software element in a system; for example, user interface screens, database gateways or menus that users interact with, are boundaries.



Sequence Diagram Elements

Activation bar: It shows the period of time during which an object is performing an action.

The top of rectangle is aligned with start of the action. The bottom is aligned with its completion and can be marked by a written message. It is represented by tall thin rectangle:



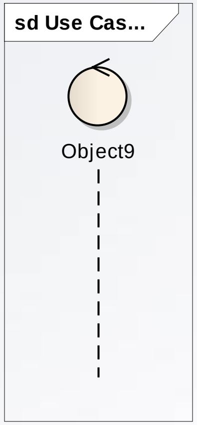
Activations Indicate when an object is performing an action

Sequence Diagram Elements

4. Control:

A ***Control*** is a stereotyped Object that models a controlling entity or manager.

Controls are objects that mediate between boundaries and entities. They orchestrate the execution of commands coming from the boundary by interacting with entity and boundary objects. Controls often correspond to use cases and map to use case controllers in the design model.

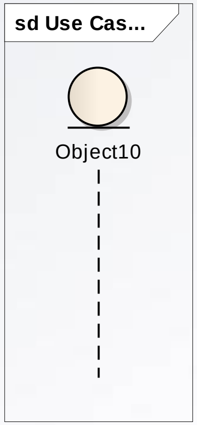


And a lifeline with a control element indicates a controlling entity or manager. It organizes and schedules the interactions between the boundaries and entities and serves as the mediator between them.

Sequence Diagram Elements

5. Entity:

An ***Entity*** is a stereotyped Object that models a store or persistence mechanism that captures the information or knowledge in a system.

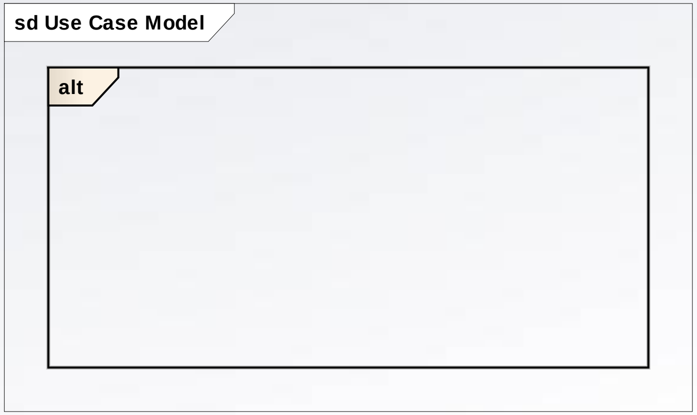


A lifeline with an entity element represents system data. For example, in a customer service application, the Customer entity would manage all data related to a customer.

Sequence Diagram Elements

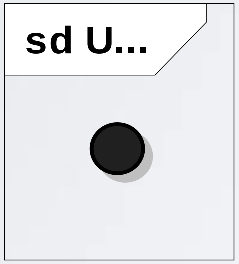
6. Fragment:

A ***Fragment*** reflects a piece or pieces of interaction (called *interaction operands*) controlled by an *interaction operator*, whose corresponding boolean conditions are known as *interaction constraints*. It displays as a transparent window, divided by horizontal dashed lines for each operand.



Sequence Diagram Elements

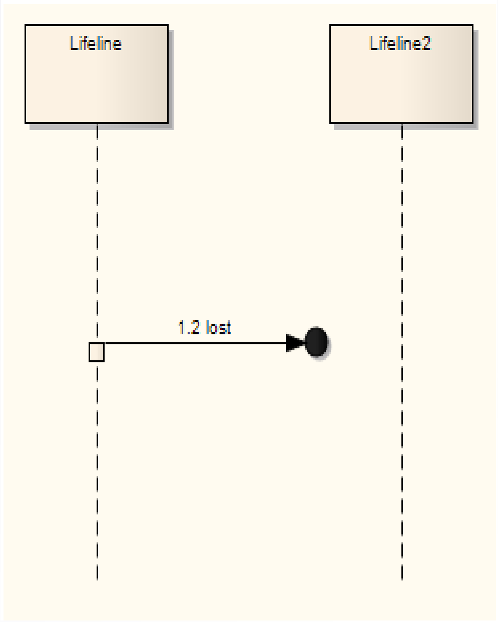
7. Endpoint:



An ***Endpoint*** is used in Interaction diagrams (Sequence, Timing, Communication or Interaction Overview) to reflect a lost or found message in sequence.

A lost message is a message where the sending event occurrence is known, but there is no receiving event occurrence.

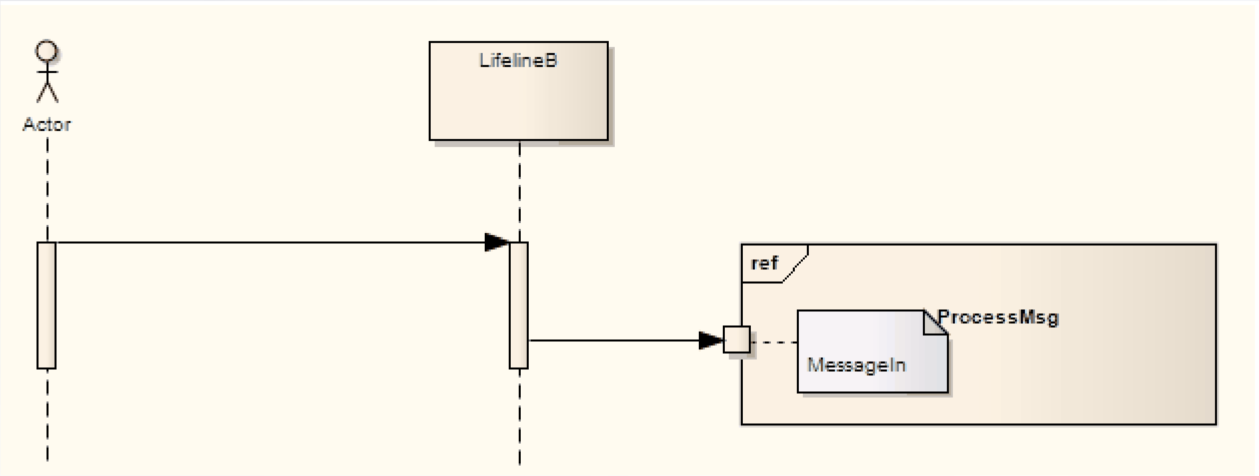
A found message is a message where the receiving event occurrence is known, but there is no (known) sending event occurrence.



Sequence Diagram Elements

8. Diagram Gate:

A ***Diagram Gate*** is a simple graphical way to indicate the point at which messages can be transmitted into and out of interaction fragments. A fragment might be required to receive or deliver a message; **internally**, an ordered message reflects this requirement,



SPARX SYSTEMS

Thank You