

Sequence Diagrams





- A sequence diagram shows interactions between objects. Sequence diagrams are used during subsystem design, but they're equally applicable to dynamic modeling during analysis, system design and even requirements capture.
- A sequence diagram describes the sequence of operations during one scenario of a system use case and determines which object carries out each operation. The UML categorizes it as an interaction diagram—a diagram that highlights how objects interact with each other.

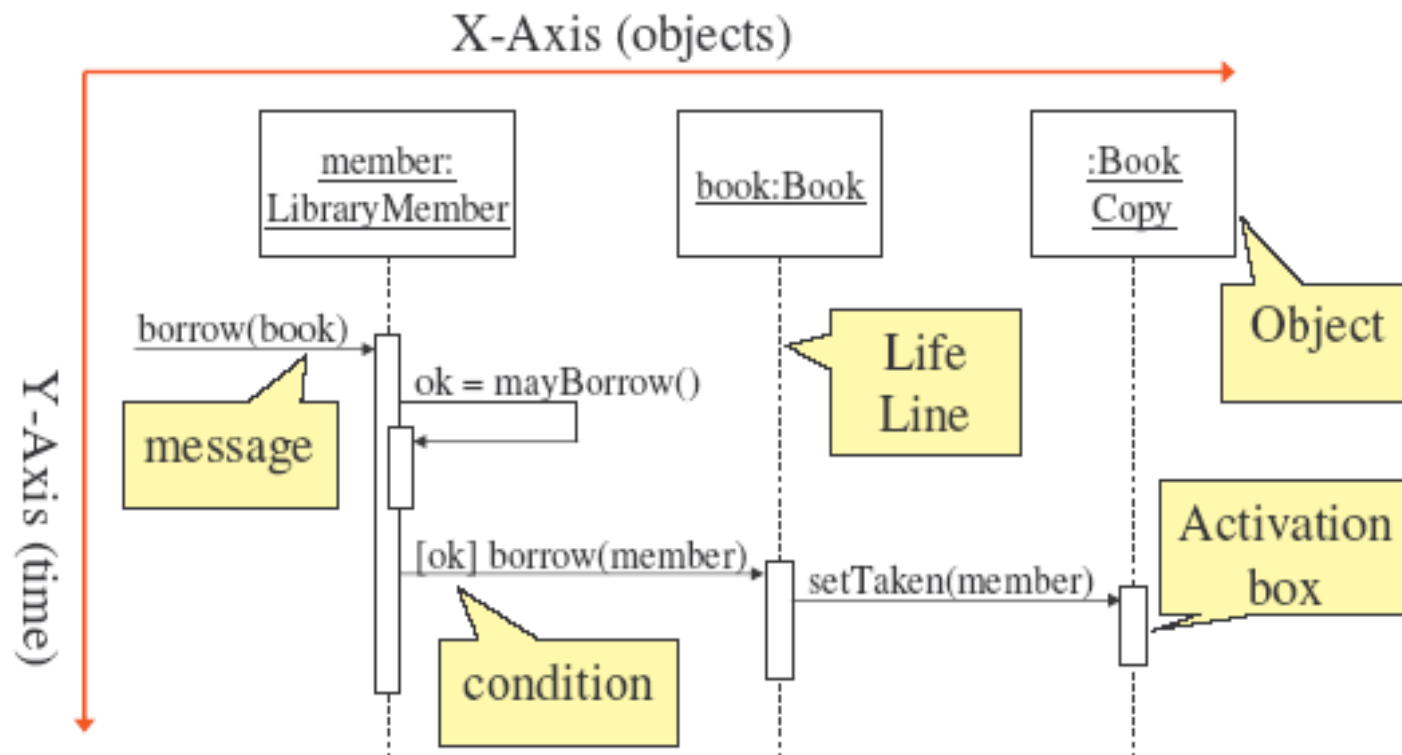


- Some business analysts use sequence diagrams as an alternative to activity diagrams with partitions (swimlanes). Instead of drawing one complex activity diagram to cover all scenarios, the BA draws one simple sequence diagram for each scenario.
- Each diagram is simple, since it describes only one scenario. The disadvantage of sequence diagrams for this purpose is that they require the BA to work out not only which object performs each action but also which object requests the action.
- On the other hand, sequence diagrams are an excellent way to design the distribution of operations among classes for programming purposes.

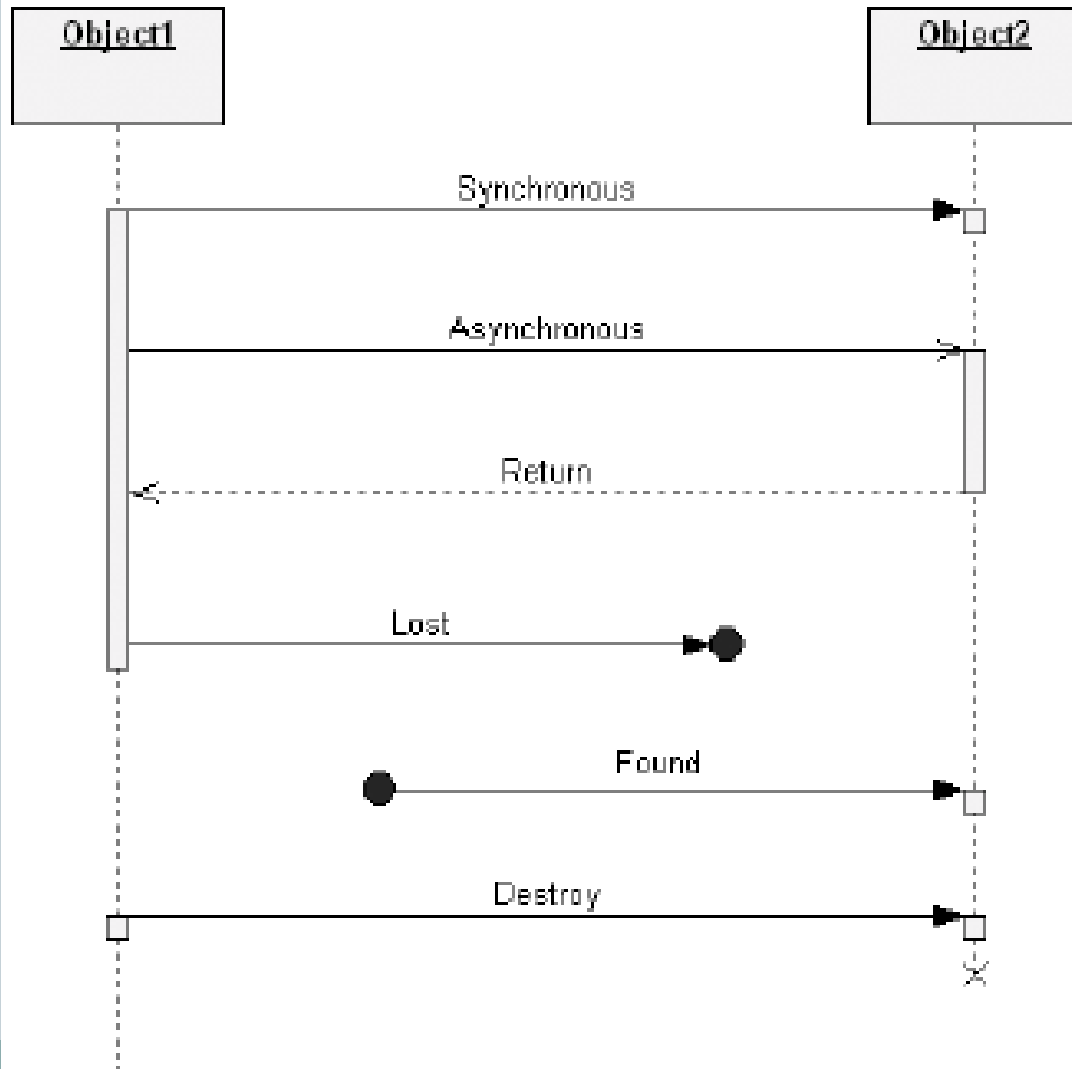
Essentials: Lifelines and Messages



- In sequence diagrams, the entities of interest (which are the same as for object diagrams) are written horizontally across the top of the diagram. A dashed vertical line, called the lifeline, is drawn below each object. These indicate the existence of the object.
- Messages (which may denote events or the invocation of operations) are shown horizontally. The endpoints of the message icons connect with the vertical lines that connect with the entities at the top of the diagram. Messages are drawn from the sender to the receiver. Ordering is indicated by vertical position, with the first message shown at the top of the diagram, and the last message shown at the bottom. As a result, sequence numbers aren't needed.



Types of Interaction



Advanced Concepts: Destruction Events

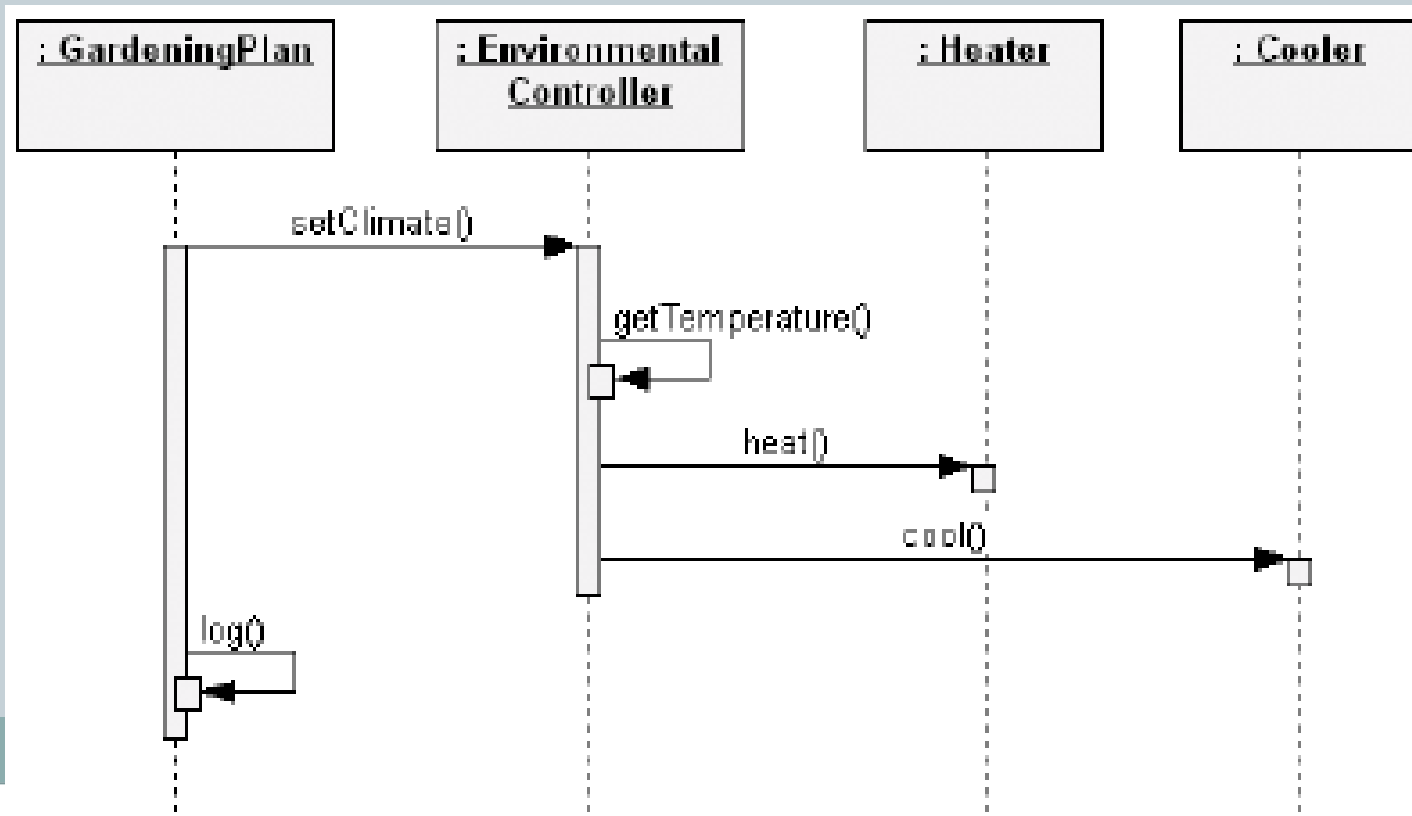


- A destruction event indicates when an object is destroyed. It is shown as an X at the end of a lifeline. If this object is involved in a composition, the other involved objects may also be destroyed.
- Sequence diagrams are conceptually very simple; however, you can add other elements to make them more expressive in the presence of certain complicated patterns of interaction.

Advanced Concepts: Execution Specification



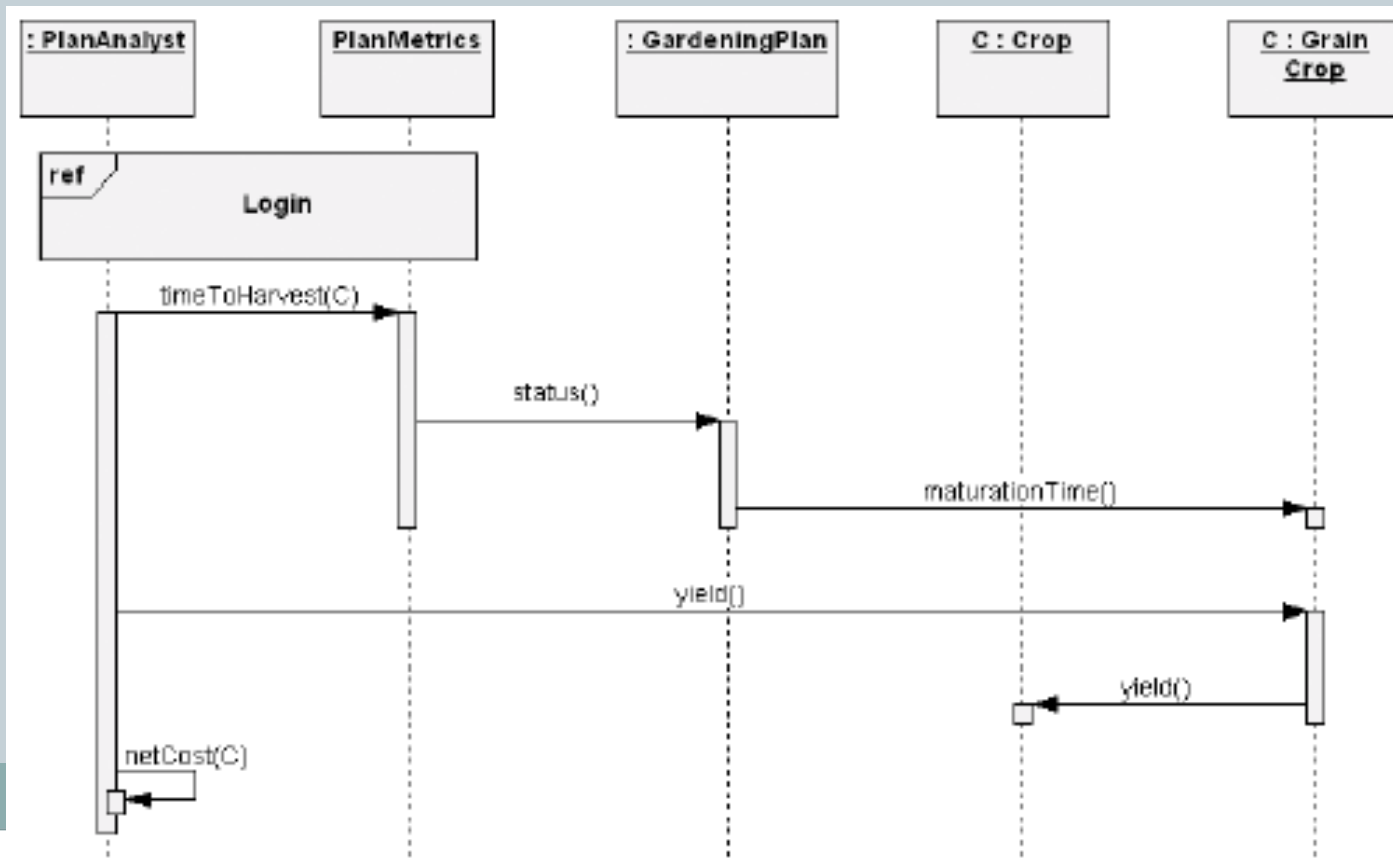
- Simple sequence diagrams may not indicate the focus of control as messages are passed. The vertical lines descending from each object in a sequence diagram with a box representing the relative time that the flow of control is focused in that object.



Advanced Concepts: Interaction Use

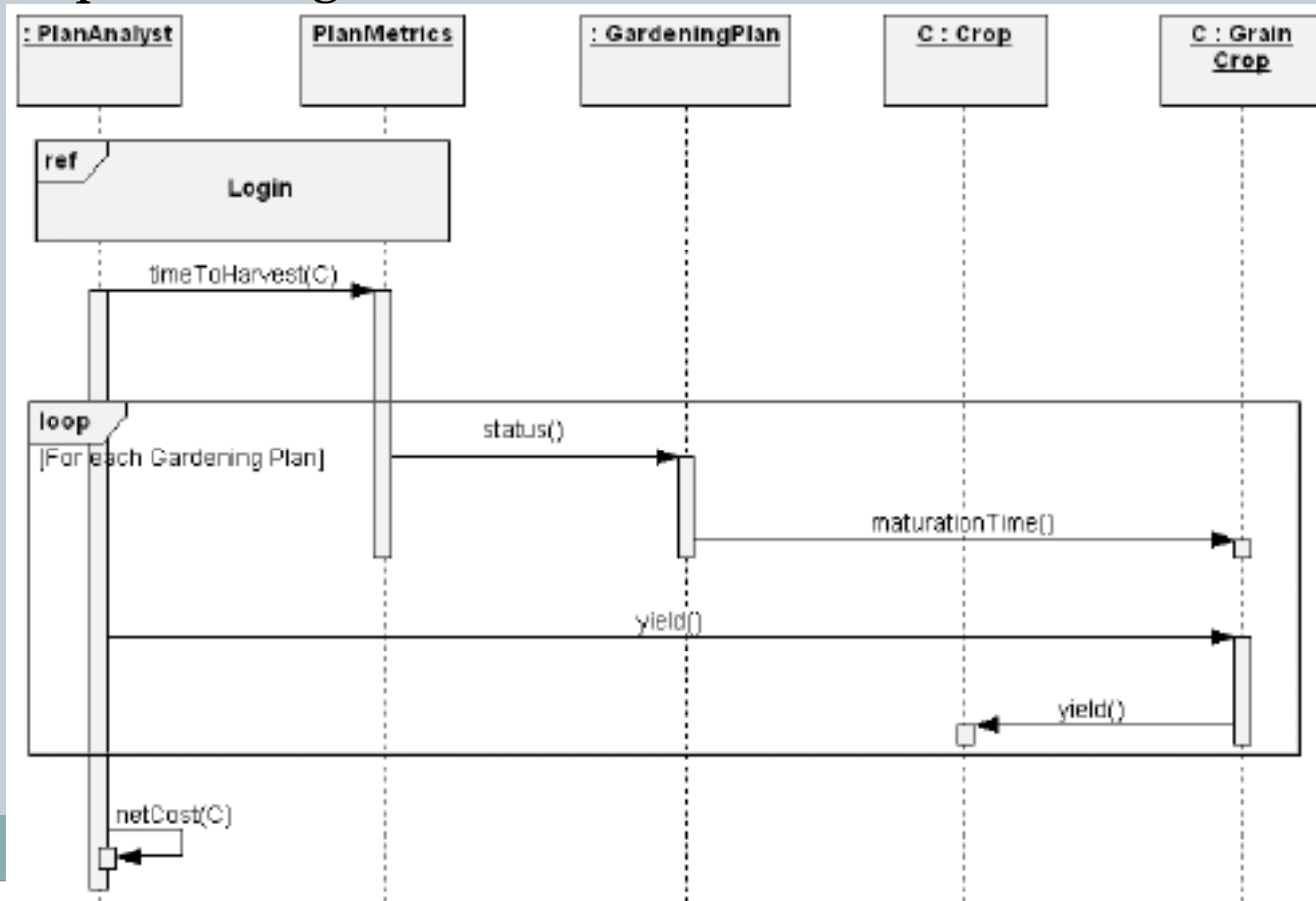


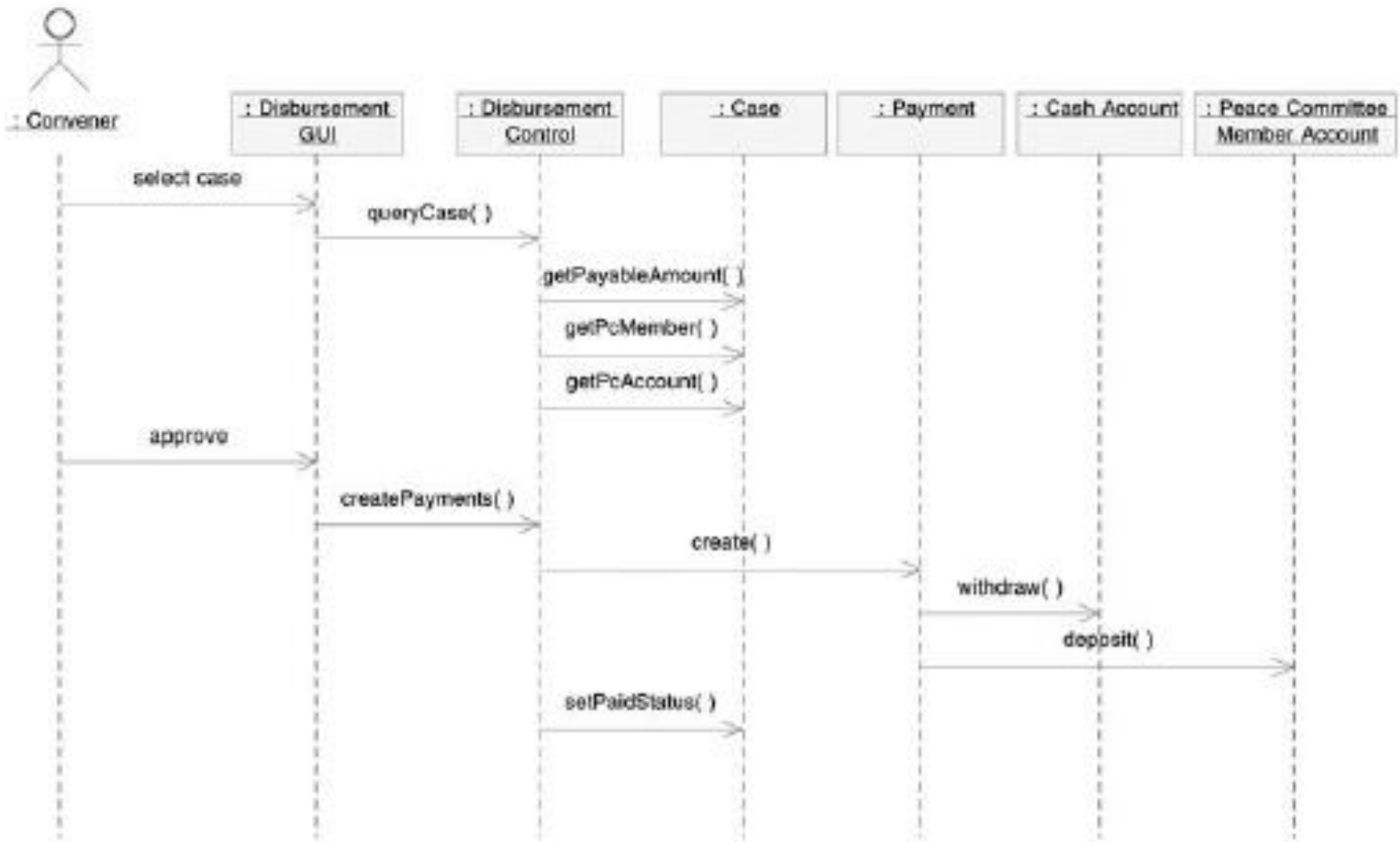
- UML 2.0 has various constructs available to simplify complex sequence diagrams. The first we will discuss is the interaction use. An interaction use is merely a way to indicate on a sequence diagram that we want to reuse an interaction that is defined elsewhere.



Advanced Concepts: Control Constructs

- Just as we saw fragments being used to simplify sequence diagrams, they can similarly be used to indicate flow control constructs on sequence diagrams.

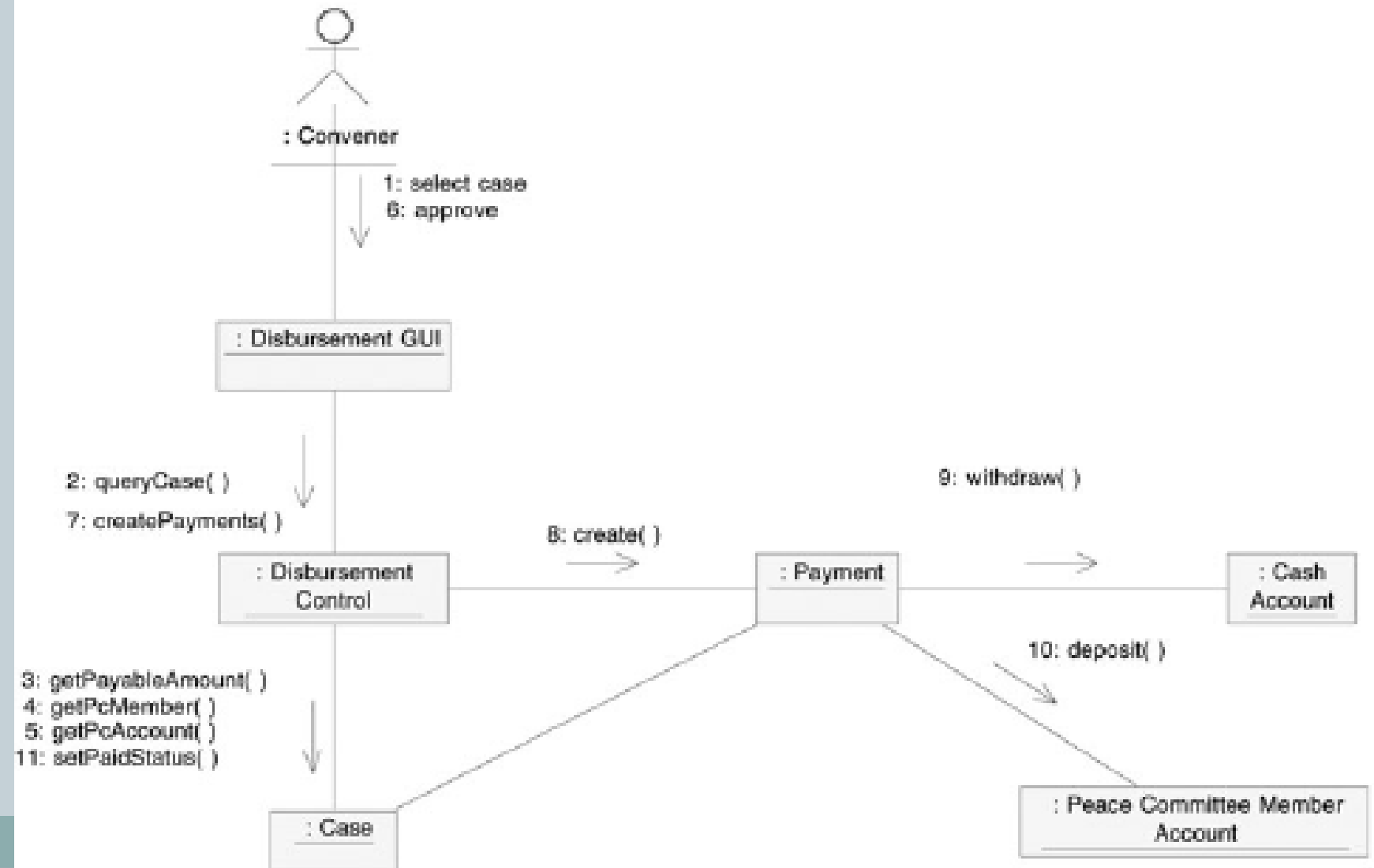




The Communications Diagrams



- Like the sequence diagram, the communication diagram is categorized in the UML as an interaction diagram. Both diagrams can show the sequencing of operations for a scenario and indicate which object does which operation. However, each highlights a different aspect of the collaboration: The communication diagram highlights structure—the ways in which objects are linked to each other—while the sequence diagram highlights timing—the order in which messages are sent between objects. In a communication diagram, objects are connected by solid lines (links). The messages are indicated as labeled arrows above the links. Each message is numbered to indicate sequencing.





- A sequence diagram that duplicates most of the semantics of the communication diagram shown. The advantage of using a sequence diagram is that it is easier to read the passing of messages in relative order. Sequence diagrams are often better than object diagrams for capturing the semantics of scenarios early in the development lifecycle, before the protocols of individual classes have been identified. Early sequence diagrams tend to focus on events as opposed to operations because events help to define the boundaries of a system under development. The advantage of using an object diagram is that it scales well to many objects with complex invocations. Each diagram has compelling benefits.