UML Interactions

Objectives

- Describe dynamic behavior and show how to capture it in a model.
- Demonstrate how to read and interpret:
  - A sequence diagram
  - A communication diagram
- Explain the similarities and differences between communication and sequence diagrams.

Objects needs to collaborate

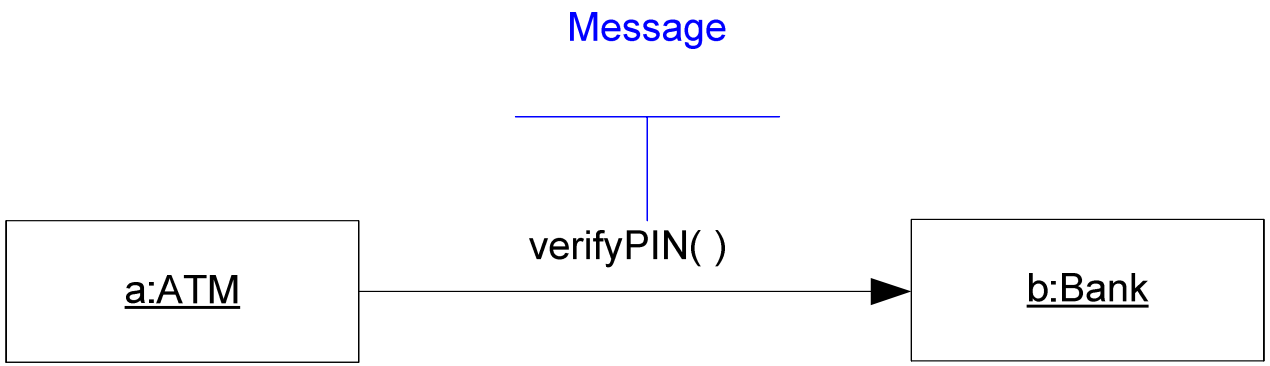
- Objects are useless unless they can collaborate to solve a problem.
  - Each object is responsible for its own behavior and status.
  - No one object can carry out every responsibility on its own.
- How do objects interact with each other?
  - They interact through messages.

Interactions (1)

- Interactions are used in a number of different situations.
- They are used to get a better grip of an interaction situation for an individual designer or for a group that needs to achieve a common understanding of the situation.
- Interactions are also used during the more detailed design phase where the precise inter-process communication must be set up according to formal protocols. When testing is performed, the traces of the system can be described as interactions and compared with those of the earlier phases.
- An interaction can be displayed in several types of diagrams:
  - Sequence Diagrams,
  - Interaction Overview Diagrams,
  - Communication Diagrams,
  - Timing Diagrams.

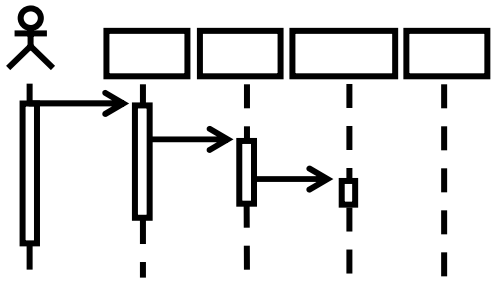
Interactions (2)

- Interaction diagrams emphasize the communication between objects, not the data manipulation associated with that communication.
- Focus on specific messages between objects and how these messages come together to realize functionality.
- A message shows how one object asks another object to perform some activity.



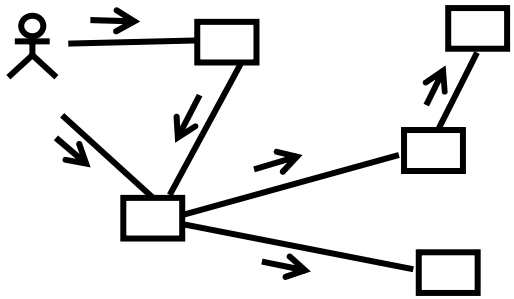
Interaction diagrams (1)

- Sequence Diagram
  - Time oriented view of object interaction



Sequence Diagrams

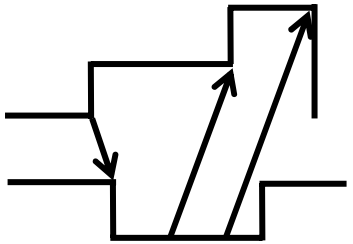
- Communication Diagram
  - Structural view of messaging objects



Communication Diagrams

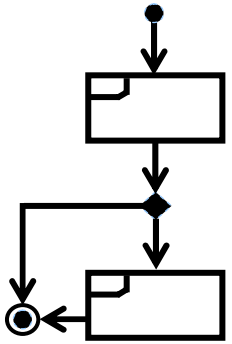
Interaction diagrams (2)

- Timing Diagram
  - Time constraint view of messages involved in an interaction



Timing Diagrams

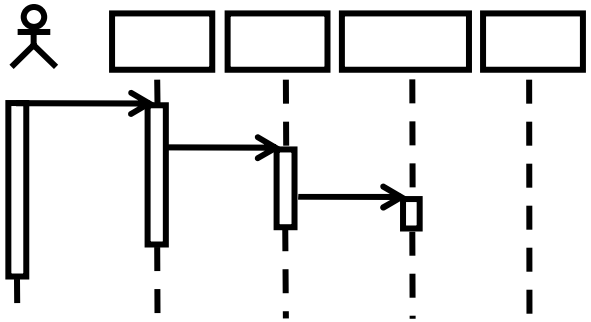
- Interaction Overview Diagram
  - High level view of interaction sets combined into logic sequence



Interaction Overview Diagrams

Sequence diagram (1)

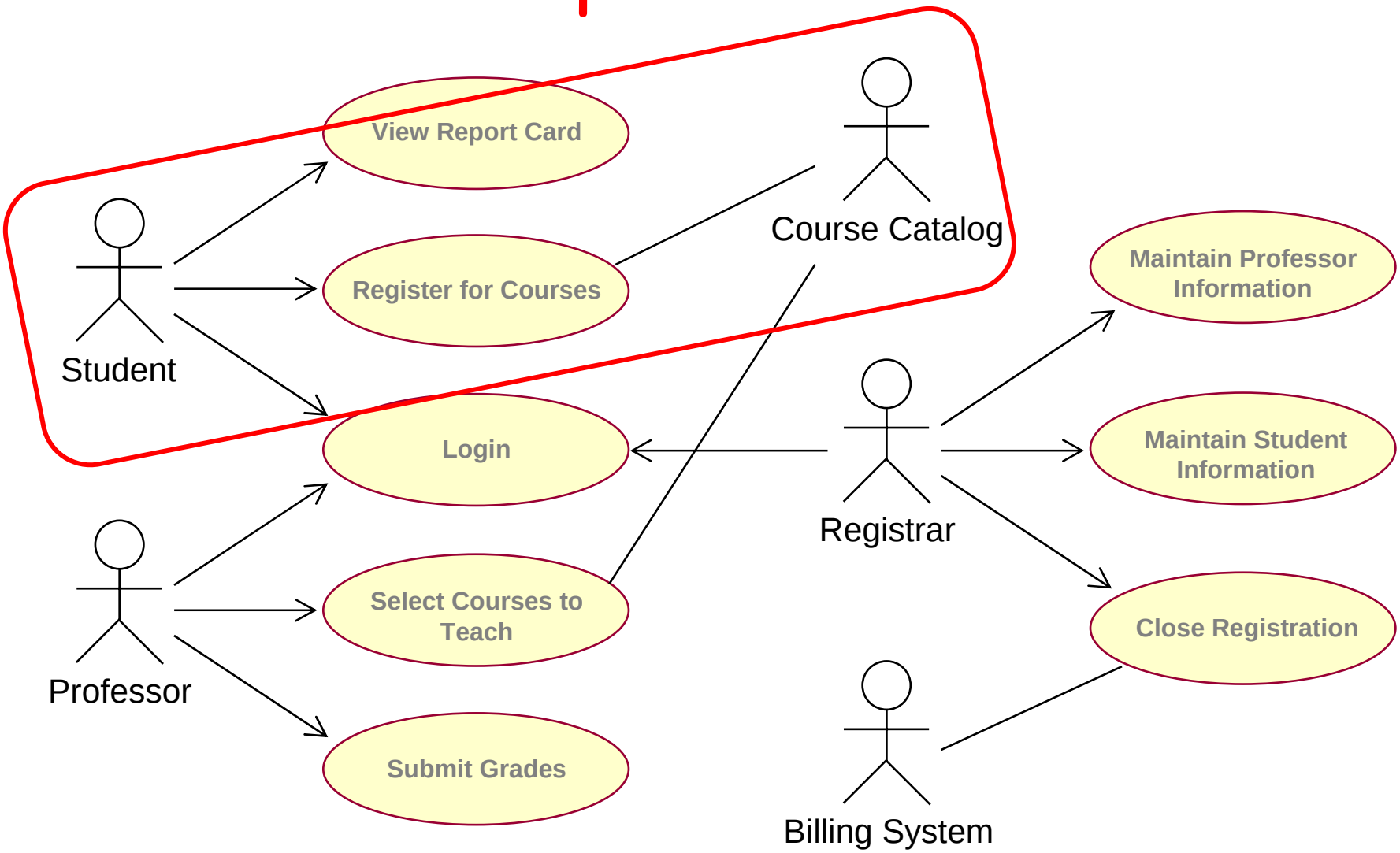
- A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.
- The diagram shows:
  - The objects participating in the interaction.
  - The sequence of messages exchanged.



Sequence Diagram

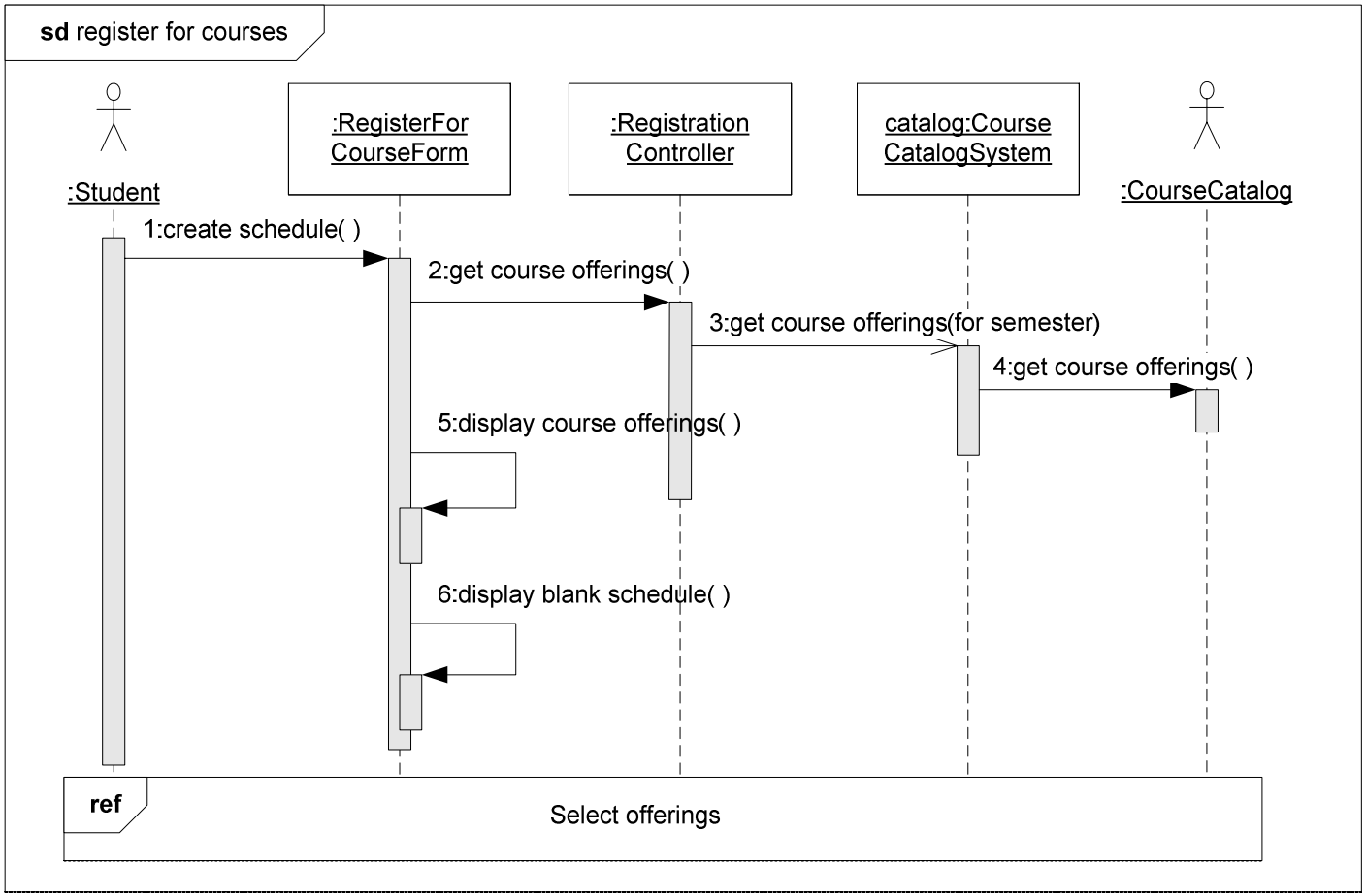
Sequence diagrams are particularly important to designers because they clarify the roles of objects in a flow and provide basic information for determining class responsibilities and interfaces.

Example: use Case



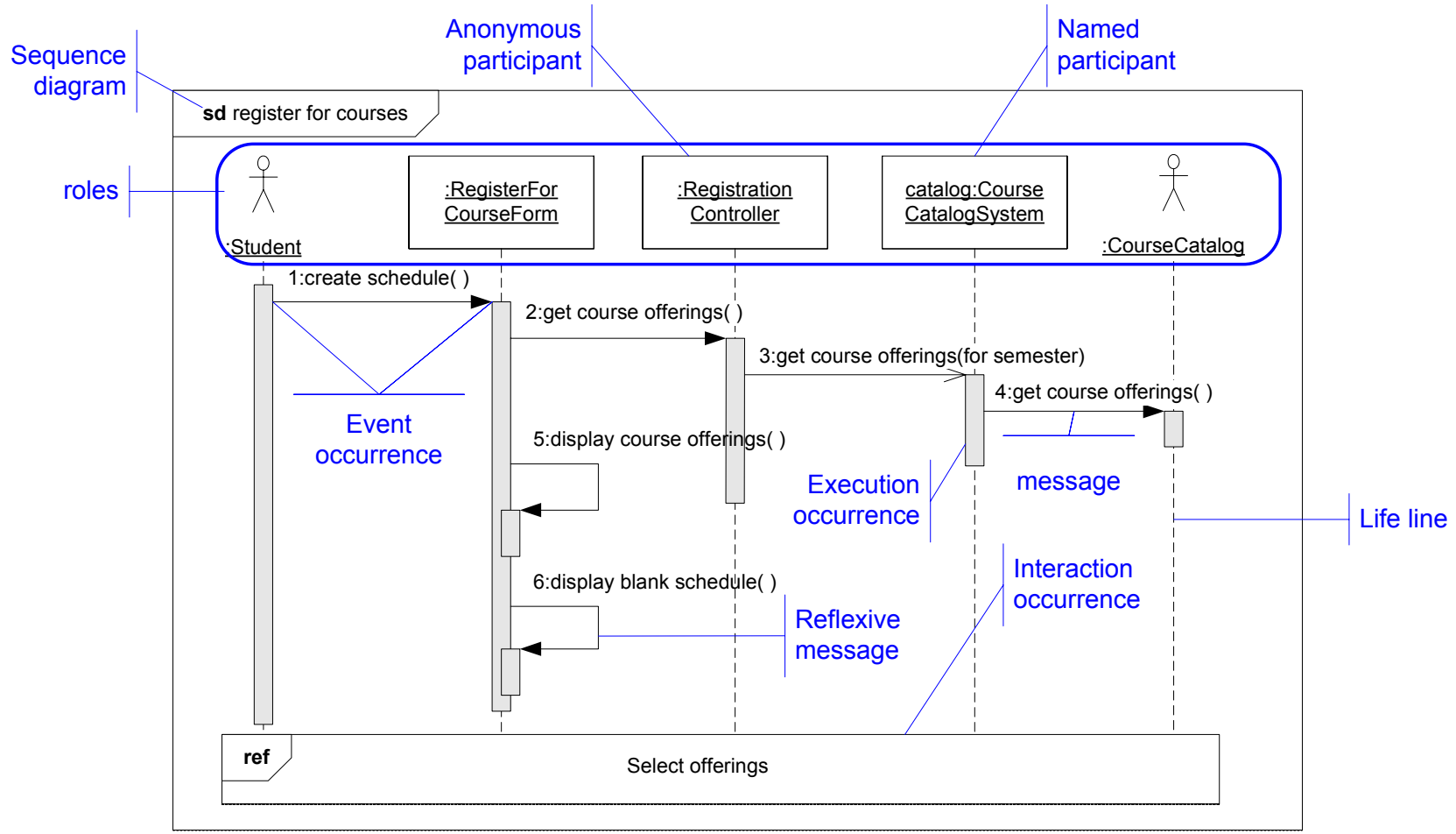
Sequence diagram (2)

- The SD below shows the object interactions to support the Register for Courses' use case, Create a Schedule sub-flow



Sequence diagram (3)

- Elements of a sequence diagram



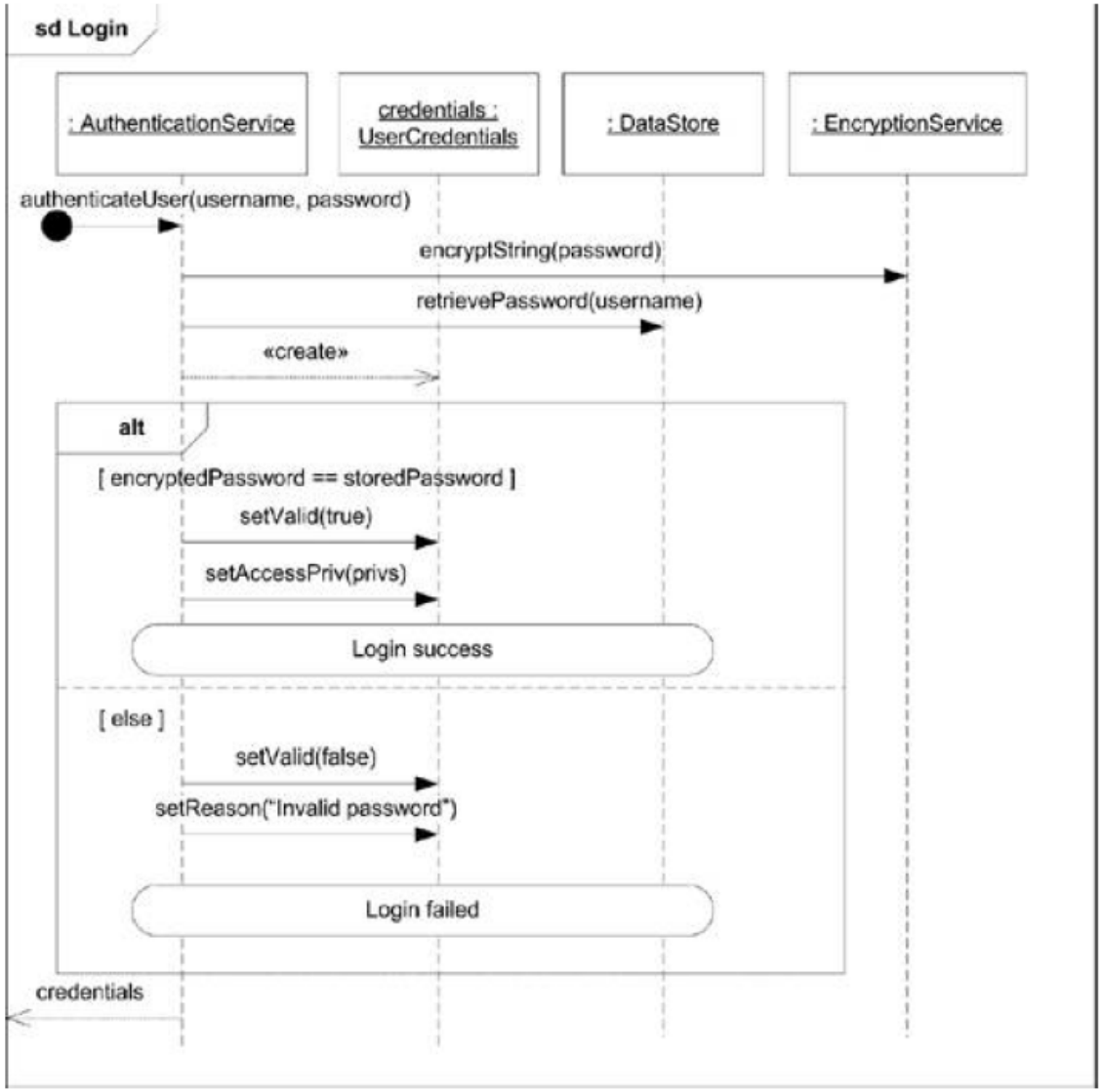
Combined fragments

- Often there are times when a particular sequence of event occurrences has special constraints or properties. For example, you may have a critical region within your interaction where a set of method calls must execute atomically, or a loop that iterates over a collection. UML calls these smaller pieces *interaction fragments*.
- Interaction fragment can be placed in a container called a combined fragment. Additional details can be added for each fragment, and you can specify how several fragments relate to each other.
- Each combined fragment is made up of an *interaction operator* and one or more interaction fragments, which are the *interaction operands* . An interaction operator specifies how the interaction operands should be interpreted.

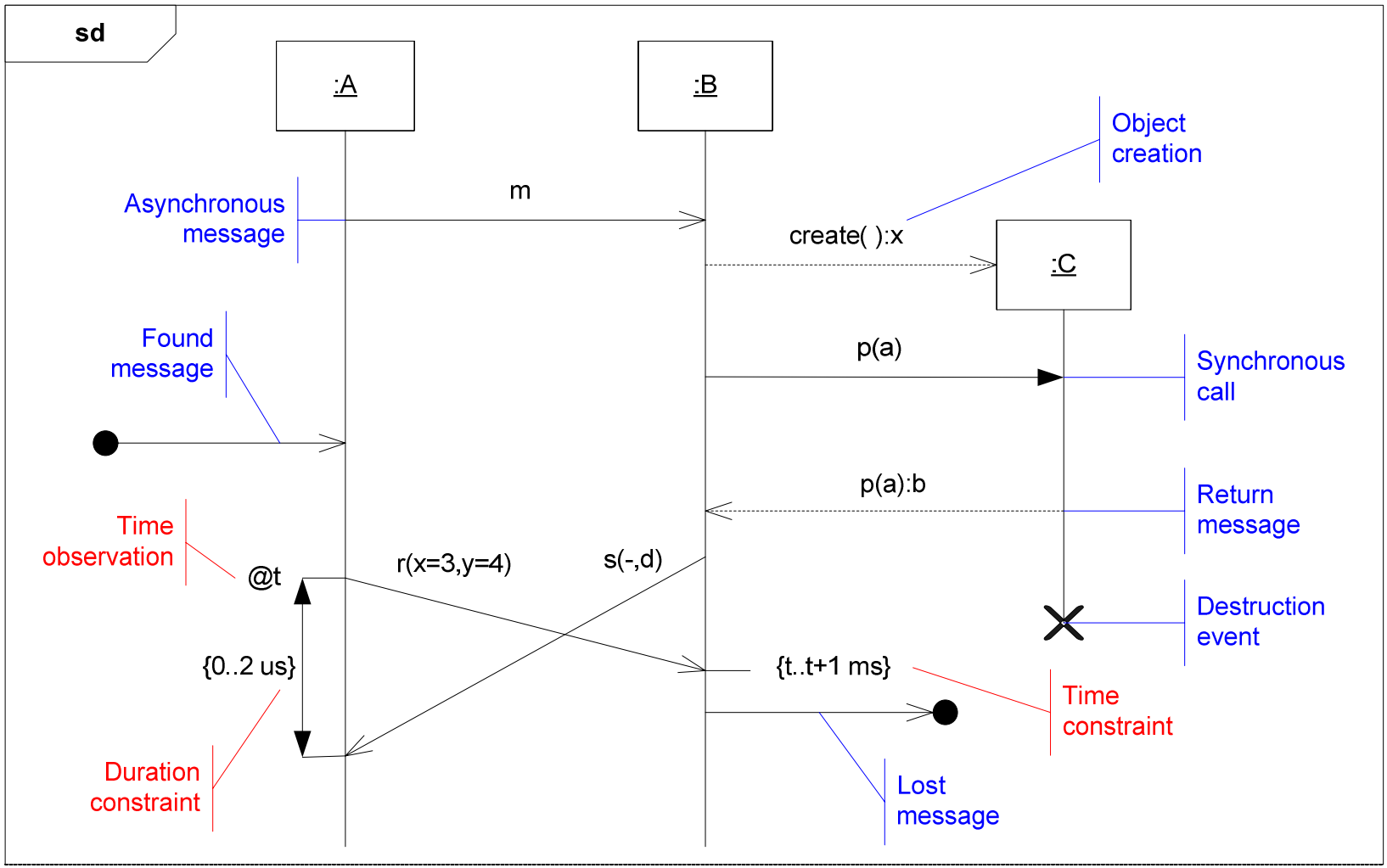
Interaction operators

- alt: (alternatives) n operands with guards
- opt: (option) executes only if the guard is true
- break: terminates the enclosing interaction
- par: (parallel) n operands, interleaving
- seq: (weak sequencing) operands can be interleaved
- strict: strict sequencing of the operands
- neg: (negative) represents traces that are defined to be invalid
- critical: (critical region) traces of the region cannot be interleaved by other OccurrenceSpecifications
- assertion: The sequences of the operand of the assertion are the only valid continuations.
- loop: The loop operand will be repeated a number of times. The Guard may include a lower and an upper number of iterations of the loop as well as a Boolean expression.
- (see UML superstructure, chapter Interactions for details)

Example of combined fragment usage

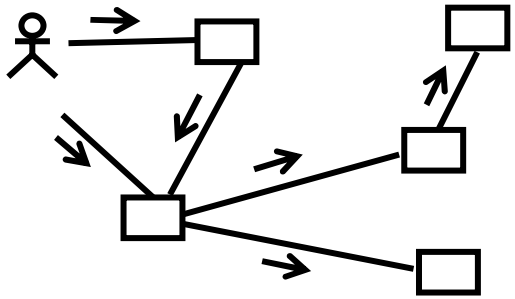


Message kinds



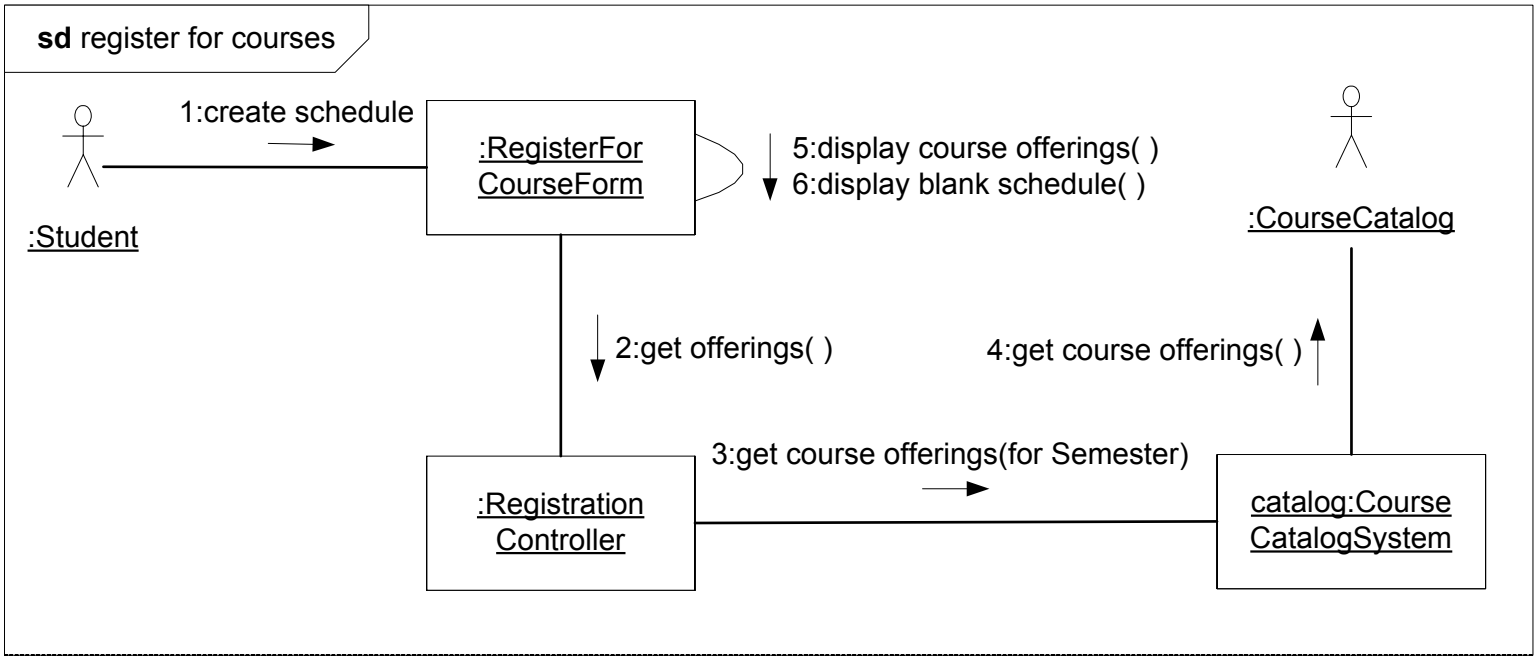
Communication diagram (1)

- A communication diagram emphasizes the organization of the objects that participate in an interaction.
- The communication diagram shows:
  - The objects participating in the interaction.
  - Links between the objects.
  - Messages passed between the objects.

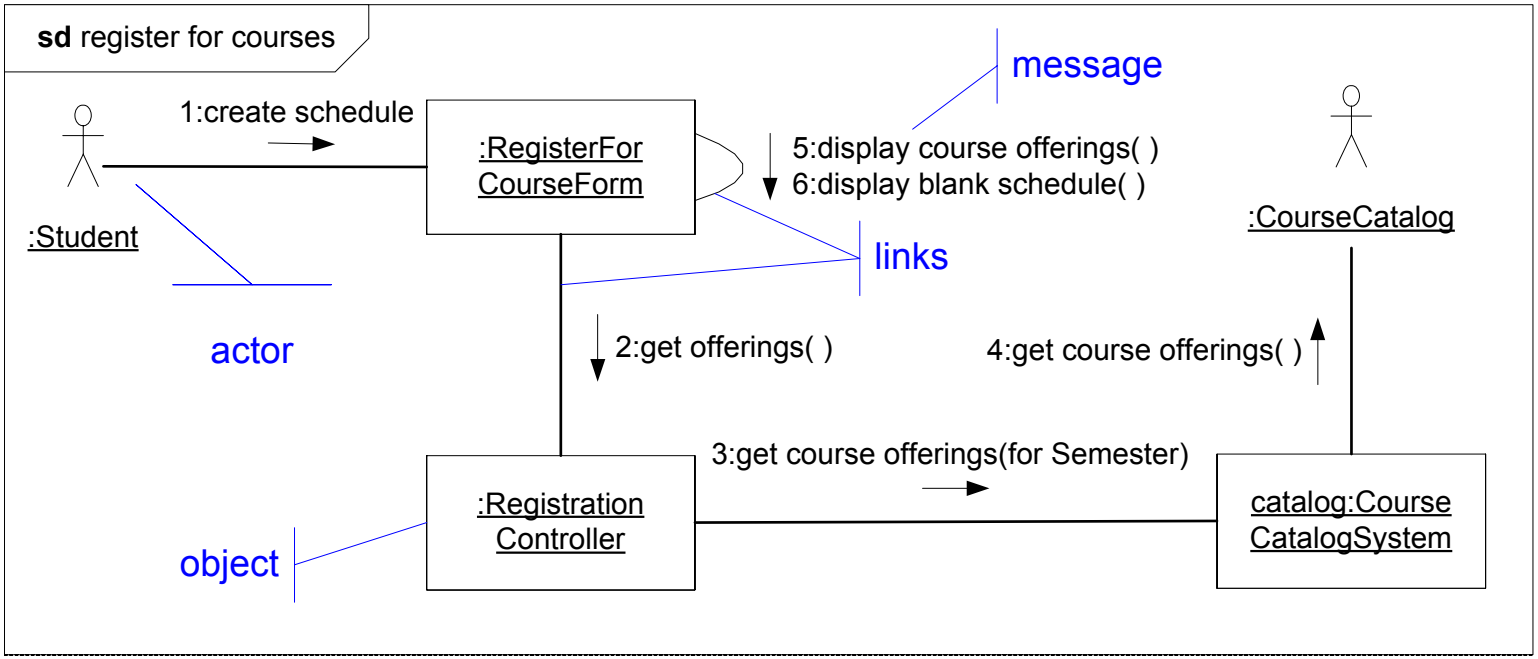


Communication Diagrams

Communication diagram (2)

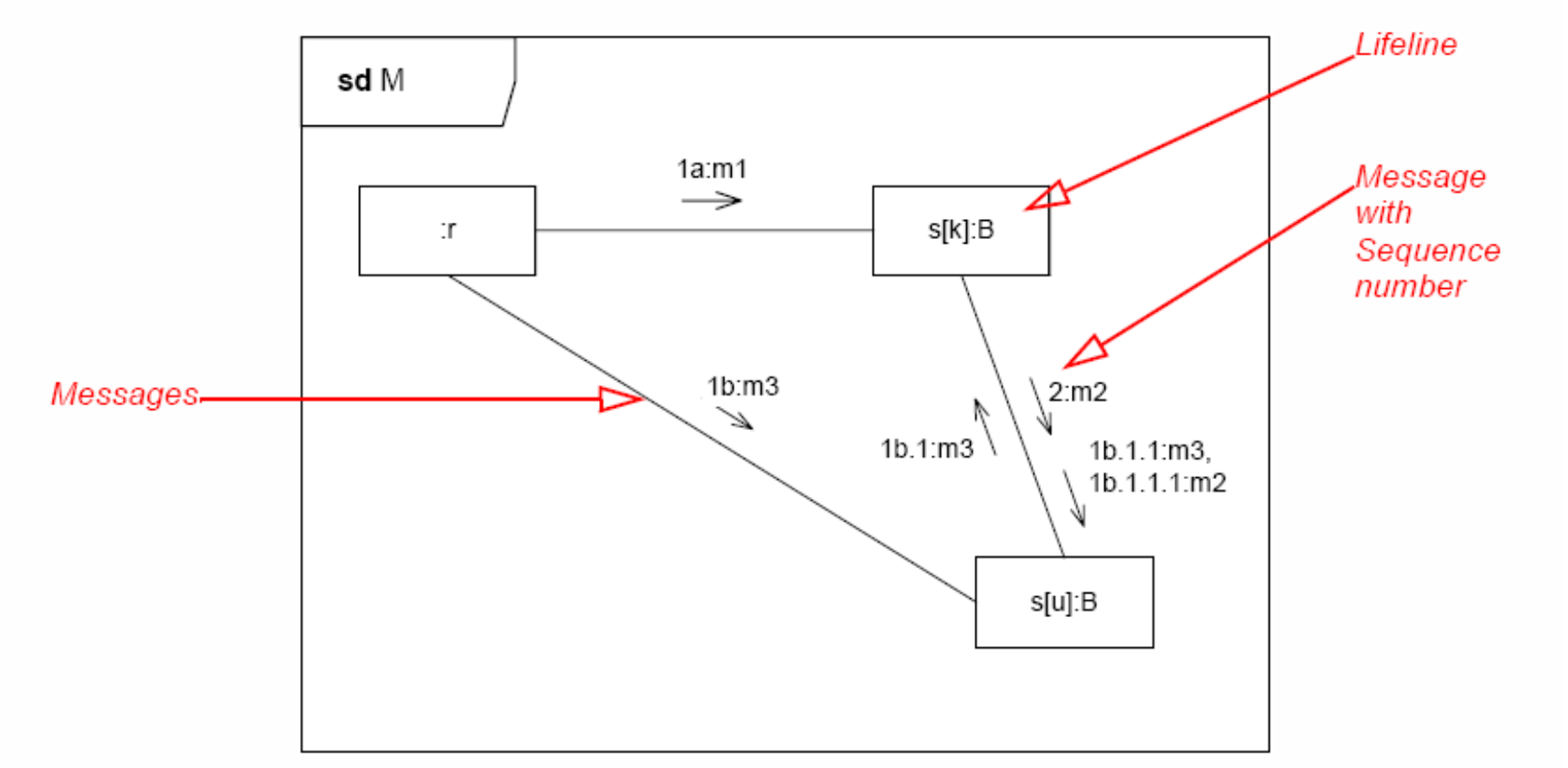


Communication diagram contents



Communication diagram

- Messages with concurrency



Sequence and communication diagrams: Similarities

- Semantically equivalent
  - Can convert one diagram to the other without losing any information
- Model the dynamic aspects of a system
- Model a use-case scenario

Sequence and communication diagrams: Differences

Sequence diagrams	Communication diagrams
– Show the explicit sequence of messages – Show execution occurrence – Better for visualizing overall flow – Better for real-time specifications and for complex scenarios	– Show relationships in addition to interactions – Better for visualizing patterns of communication – Better for visualizing all of the effects on a given object – Easier to use for brainstorming sessions

Timing diagram