▸ ▸ ▸ **Module 5**
Interaction Diagrams

IBM

IBM Software Group

Essentials of Visual Modeling with UML 2.0
Module 5: Interaction Diagrams

Rational. software

Topics

Objectives

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.

2

IBM

Objects Need 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?
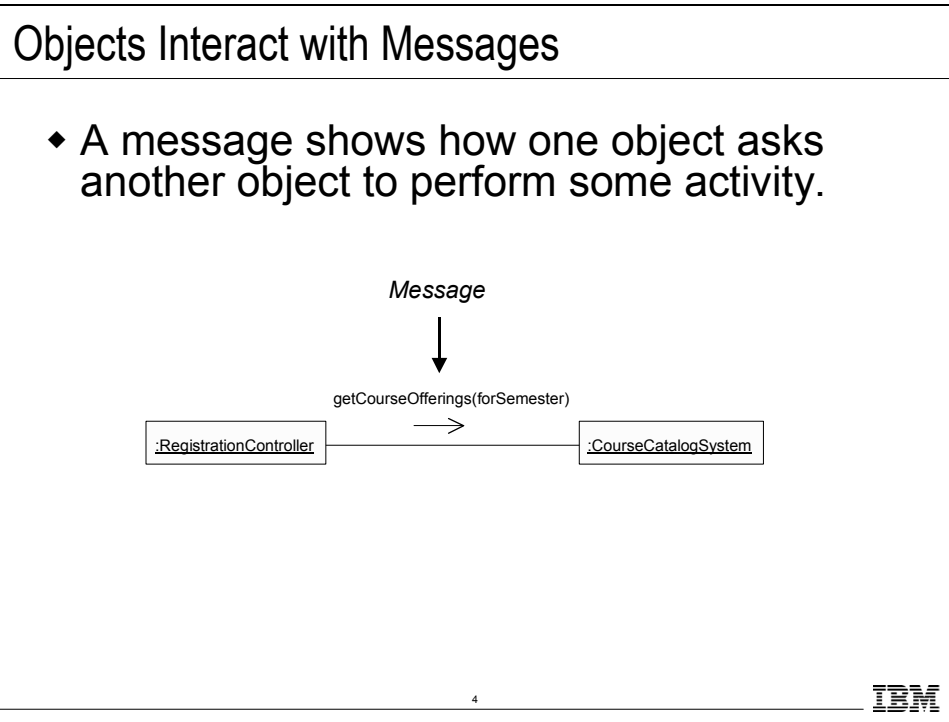   - ▪ They interact through messages.

3

IBM

Objects need to realize the behavior specified in each use-case scenario. How is this done? The objects must collaborate together to bring about the desired behavior in the system.

Is there a mechanism that allows these objects to work together? There is, and that mechanism is called a **message**.

Objects Interact with Messages

A **message** can be defined as:

> The specification of a communication among objects that conveys information with the expectation that activity will ensue. (*The Unified Modeling Language User Guide*, Booch, 1999.)

- When you pass a message, the action that results is an executable statement that forms an abstraction of a computational procedure. An action may result in a change of state.
- Messages are the mechanism that permits objects to interact with each other. A message is often implemented by a simple activity. For example, one object calls an operation in another. When the activity has been executed, the control is returned to the caller along with a return value.

© Copyright IBM Corp. 2004

What is an Interaction Diagram?
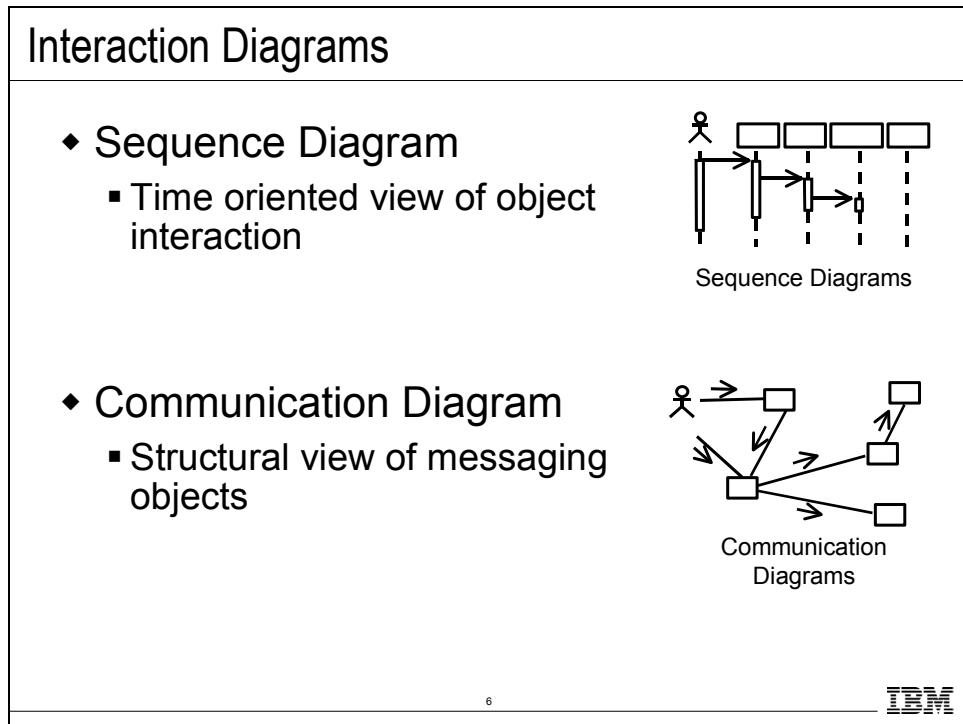
What is an Interaction Diagram?

- ◆ Generic term that applies to several diagrams that emphasize object interactions
  - Sequence Diagram
  - Communication Diagram

- ◆ Specialized Variants
  - Timing Diagram
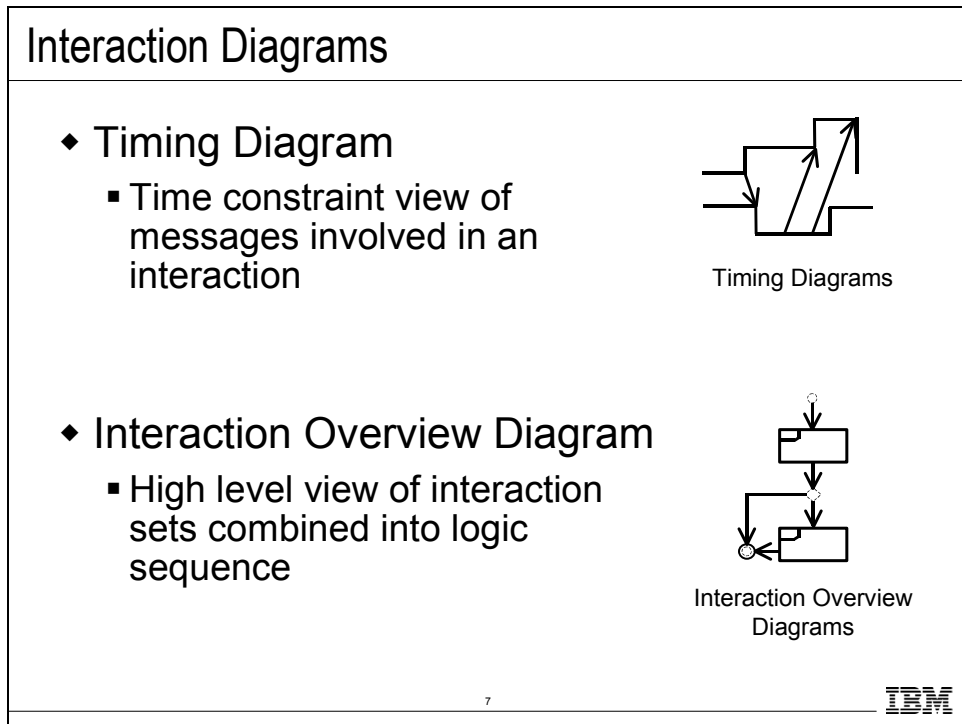 - Interaction Overview Diagram

5

IBM

Interaction Diagrams

Interaction Diagrams

- ◆ Sequence Diagram
  - ▪ Time oriented view of object interaction

  Sequence Diagrams

- ◆ Communication Diagram
  - ▪ Structural view of messaging objects

  Communication Diagrams

6   IBM

The **sequence diagram** is a time-oriented view of the interaction between objects to accomplish a behavioral goal of the system. An interaction may be modeled at any level of abstraction within the system design, from subsystem interactions to instance-level interaction for a single operation or activity.

The **communication diagram** is a structural view of the messaging between objects, taken from the Collaboration diagram concept of UML1.

Interaction Diagrams

Interaction Diagrams

- ◆ 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

7

IBM

The **timing diagram** is an optional diagram designed to specify the time constraints on messages sent and received in the course of an interaction. This diagram probably has more usefulness in real-time applications where timing is critical.

The **interaction overview diagram** is a high-level view of the sets of interactions combined into logic sequence, including flow-control logic to navigate between the interactions. Think of this as a cross between a Sequence Diagram, for the interactions sets, and an Activity Diagram, for the logic sequence.

Where Are We?

Where Are We?

☆ ◆ Sequence diagrams
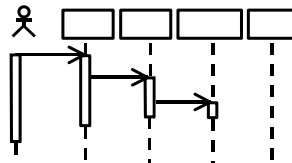- ◆ Communication diagrams
- ◆ Interaction diagram comparison
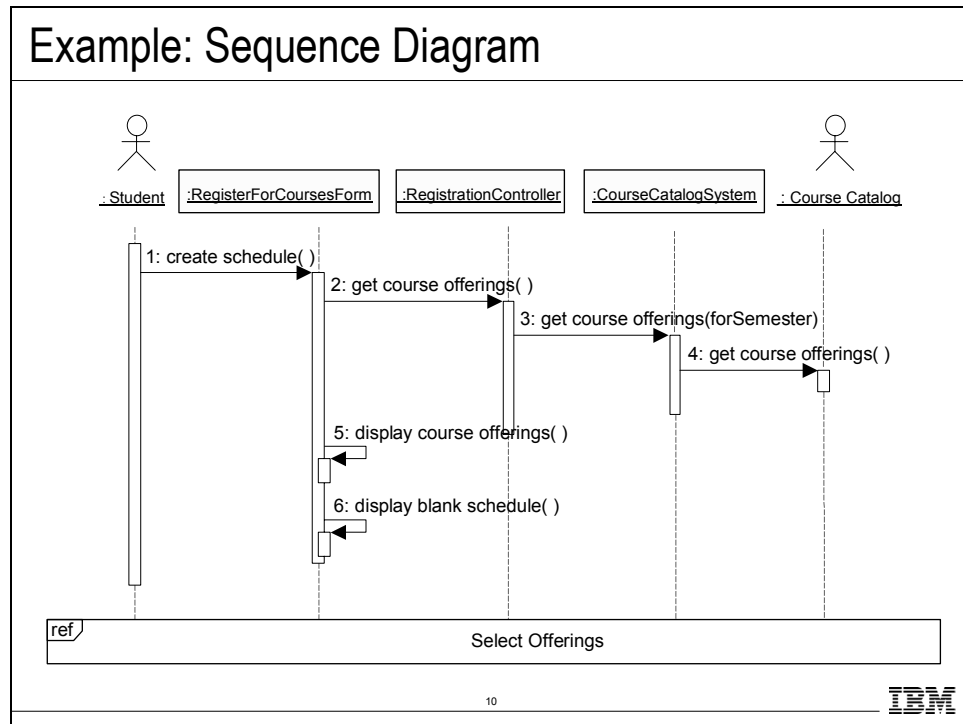
8

IBM

What Is a Sequence Diagram?

A **sequence diagram** describes a pattern of interaction among objects, arranged in a chronological order. It shows the objects participating in the interaction by their "lifelines" and the messages that they send to each other.

In most cases, we use a sequence diagram to illustrate use-case realizations. That is, realizations show how objects interact to perform the behavior of all or part of a use case. One or more sequence diagrams may illustrate the object interactions that enact a use case. A typical organization is to have one sequence diagram for the main flow of events and one sequence diagram for each independent sub-flow of the use case.

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: Sequence Diagram

You can have objects and actor instances in sequence diagrams, together with messages describing how they interact. The diagram describes what takes place in the participating objects, in terms of activations, and how the objects communicate by sending messages to one another.
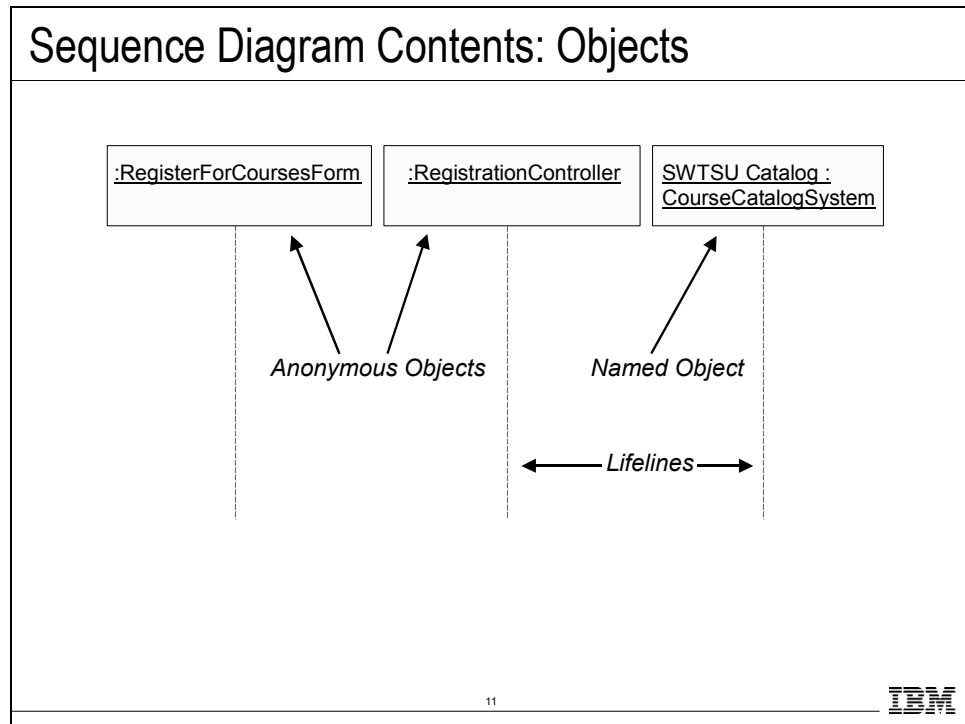
You can make a sequence diagram for each variant of a use case's flow of events.

The above example shows the object interactions to support the Register for Courses' use case, Create a Schedule sub-flow. Note the following responsibility allocation rationale.

- The RegisterForCoursesForm knows what data it needs to display and how to display it. It does not know where to go to get it. That is one of the RegistrationController's responsibilities.
- Only the RegisterForCoursesForm interacts with the Student actor.
- The RegistrationController understands how Students and Schedules are related.
- Only the CourseCatalogSystem class interacts with the external legacy Course Catalog System.

Note the inclusion of the actors. This is important as it explicitly models what elements communicate with the "outside world."
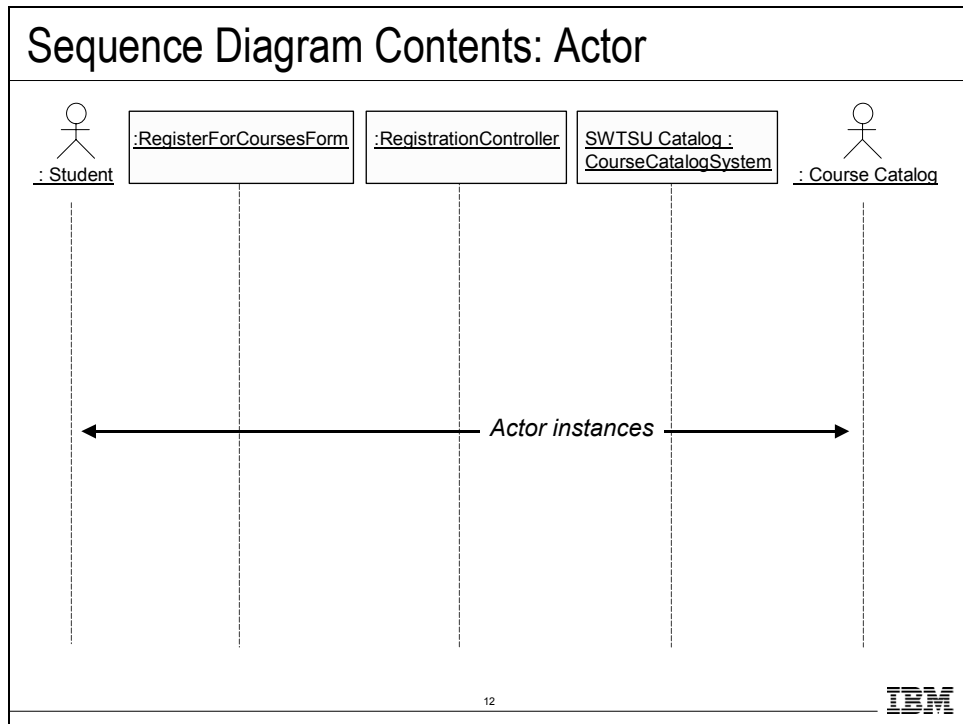
Sequence Diagram Contents: Objects

An **object** is shown as a vertical dashed line called the "lifeline." The lifeline represents the existence of the object at a particular time. An object symbol is drawn at the head of the lifeline, and shows the name of the object and its class underlined and separated by a colon:

> objectname : classname

You can use objects in sequence diagrams in the following ways:

- A lifeline can represent an object. Thus, you can use a lifeline to model both class and object behavior. Usually, a lifeline represents all objects of a certain class.
- An object's class can be unspecified. Normally you create a sequence diagram with objects first and specify their classes later.
- The objects can be unnamed. However, name them if you want to discriminate different objects of the same class.
- Several lifelines in the same diagram can represent different objects of the same class. As stated previously, the objects should be named so that you can discriminate between the two objects.
- A lifeline that represents a class can exist in parallel with lifelines that represent objects of that class. The object name of the lifeline that represents the class can be set to the name of the class.
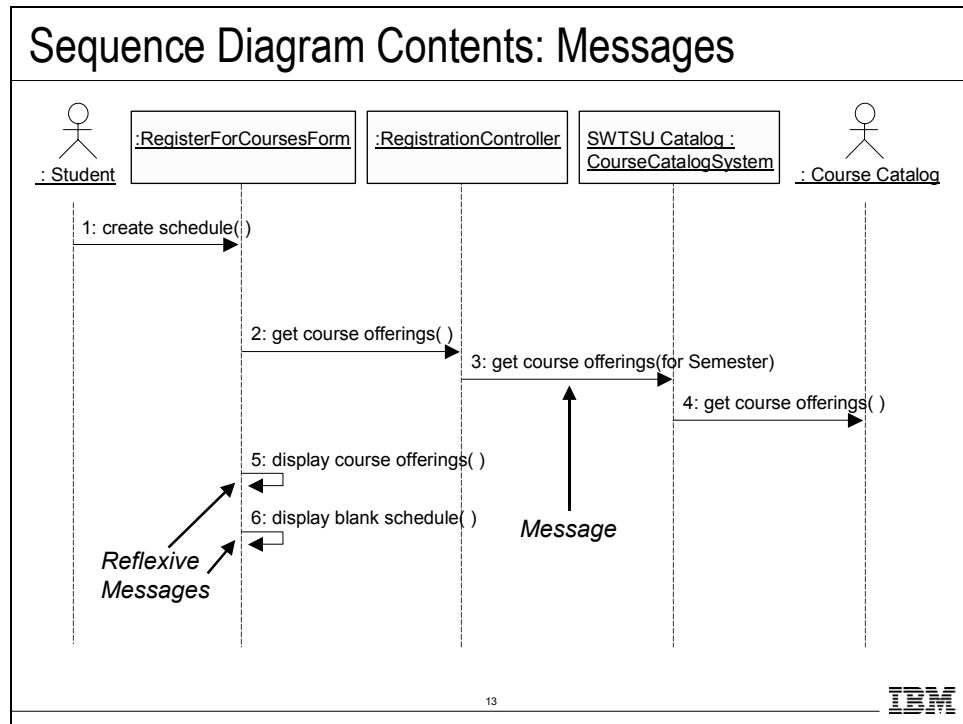
Sequence Diagram Contents: Actor

Normally an **actor** instance is represented by the first (leftmost) lifeline in the sequence diagram, as the invoker of the interaction. If you have several actor instances in the same diagram, try keeping them either to the leftmost or to the rightmost lifelines.

Don't show the interaction between actors in a sequence diagram because actors are, by definition, external to the system.

© Copyright IBM Corp. 2004

Sequence Diagram Contents: Messages

```
Sequence Diagram Contents: Messages
```


A **message** is a communication between objects that conveys information with the expectation that activity will ensue.
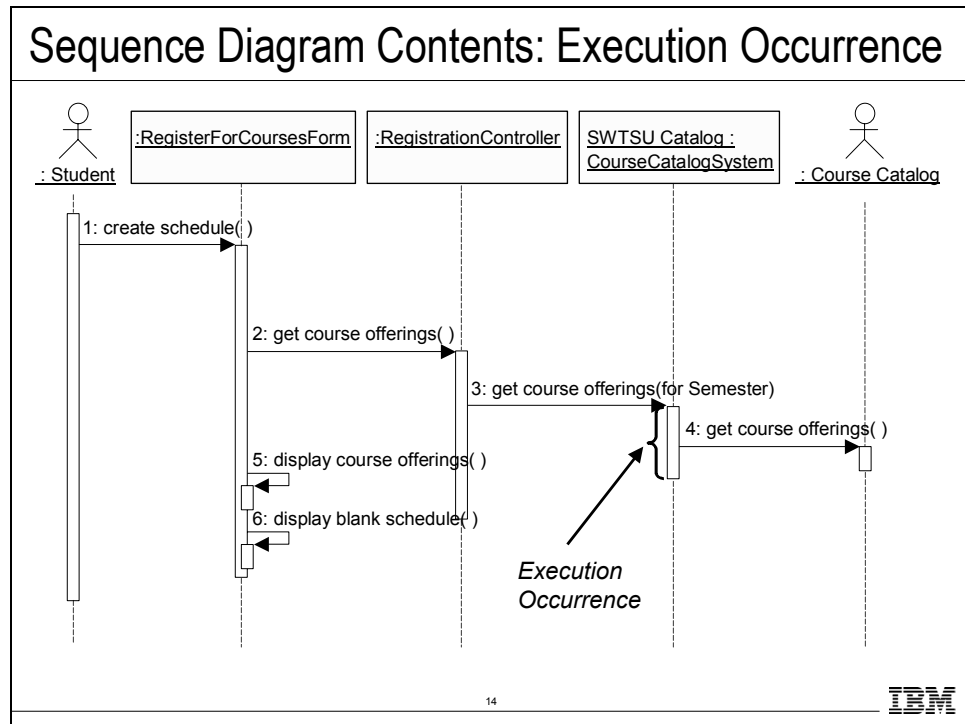
In sequence diagrams, a message is shown as a horizontal solid arrow from the lifeline of one object to the lifeline of another object.

For a message from an object to itself, the arrow may start and finish on the same lifeline. The arrow is labeled with the name of the message and its parameters. The arrow may also be labeled with a sequence number to show the sequence of the message in the overall interaction.

Sequence numbers are often omitted in sequence diagrams, where the physical location of the arrow shows the relative sequence.

A message can be unassigned, meaning that its name is a temporary string that describes the overall meaning of the message. (// is a way to represent responsibilities and is discussed further in the OOAD course.) It is not the name of an operation of the receiving object. You can later assign the message by specifying the operation of the message's destination object. The specified operation then replaces the name of the message.

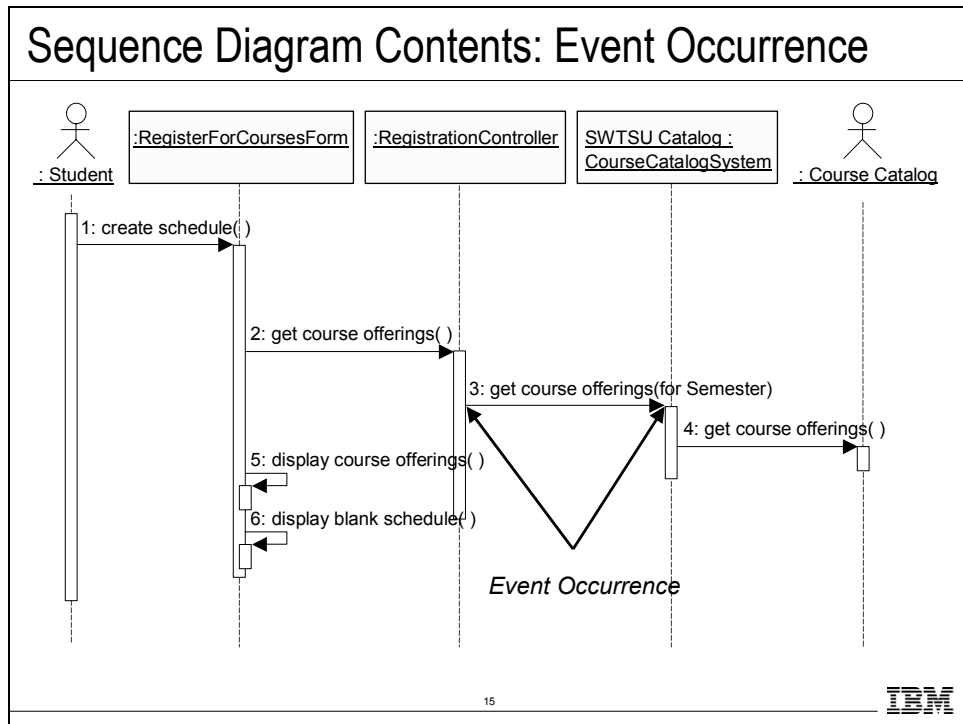## Sequence Diagram Contents: Execution Occurrence

The **execution occurrence** is a tall, thin rectangle that shows the period of time during which an object is performing an action, either directly or through a subordinate procedure.

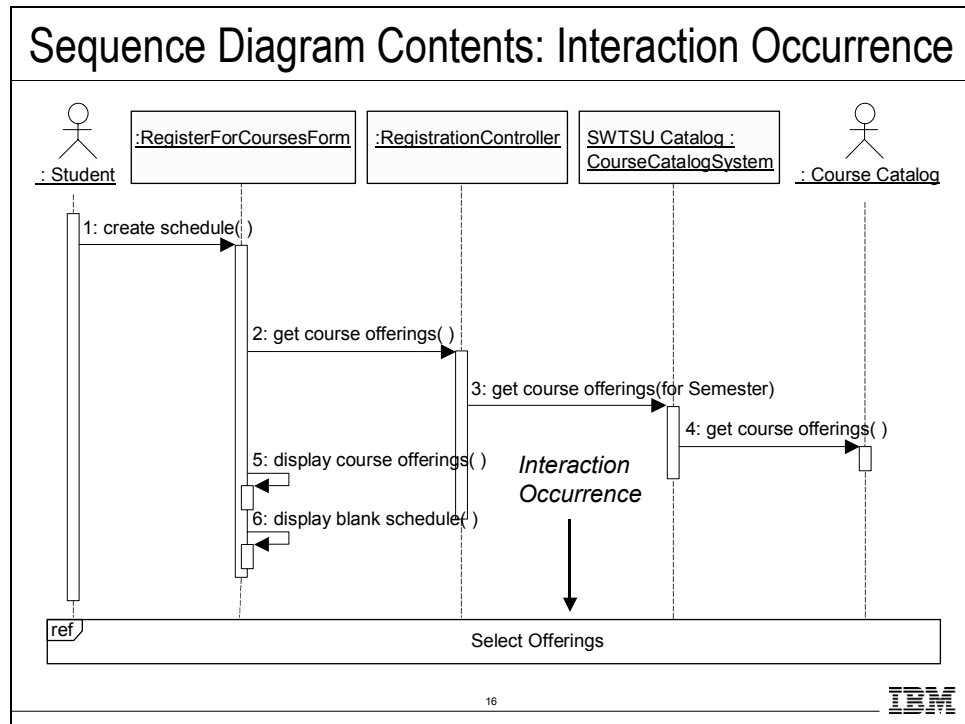The top of the rectangle is aligned with the start of the action. The bottom is aligned with its completion.

In earlier UML releases, the execution occurrence was called the focus of control. This changed with UML 2.

Sequence Diagram Contents: Event Occurrence

The **event occurrence** is the sending or receipt of a message by an object. An event occurrence is not explicitly shown as a separate modeling concept. It is normally shown by the intersection of the message with the lifeline. A message connects two event occurrences on two lifelines.

Sequence Diagram Contents: Interaction Occurrence

The **interaction occurrence** is a reference to an interaction within the definition of another interaction.

An interaction occurrence is shown in a sequence diagram as a rectangle with the tag **ref** (for reference). The rectangle covers the lifelines that are included in the referenced interaction. The name of the referenced interaction is placed in the rectangle

Where Are We?

Where Are We?

- ◆ Sequence diagrams
- ☆ ◆ Communication diagrams
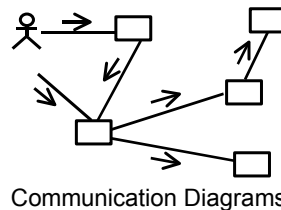- ◆ Interaction diagram comparison

17

IBM

What Is a Communication Diagram?

What Is a Communication Diagram?

- ◆ 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.
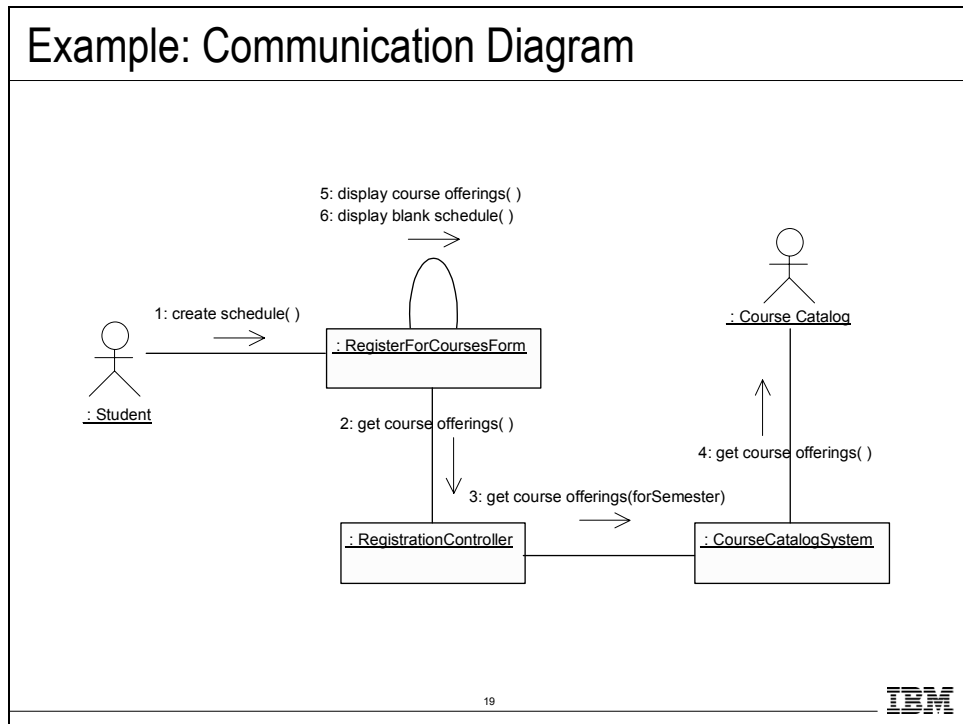 - ▪ Messages passed between the objects.

Communication Diagrams

18

IBM

A **communication diagram** shows how objects interact to perform the behavior of a particular use case or a part of a use case.

Like sequence diagrams, communication diagrams are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determine class responsibilities and interfaces.

Because of the communication diagram's format, they tend to be better suited for analysis activities. Specifically, they tend to be better suited to depict simpler interactions of a smaller number of objects.

As the number of objects and messages grows, the diagram becomes increasingly hard to read. It is also difficult to show additional descriptive information like timing, decision points, or other unstructured information that can be easily added to the notes in a sequence diagram.
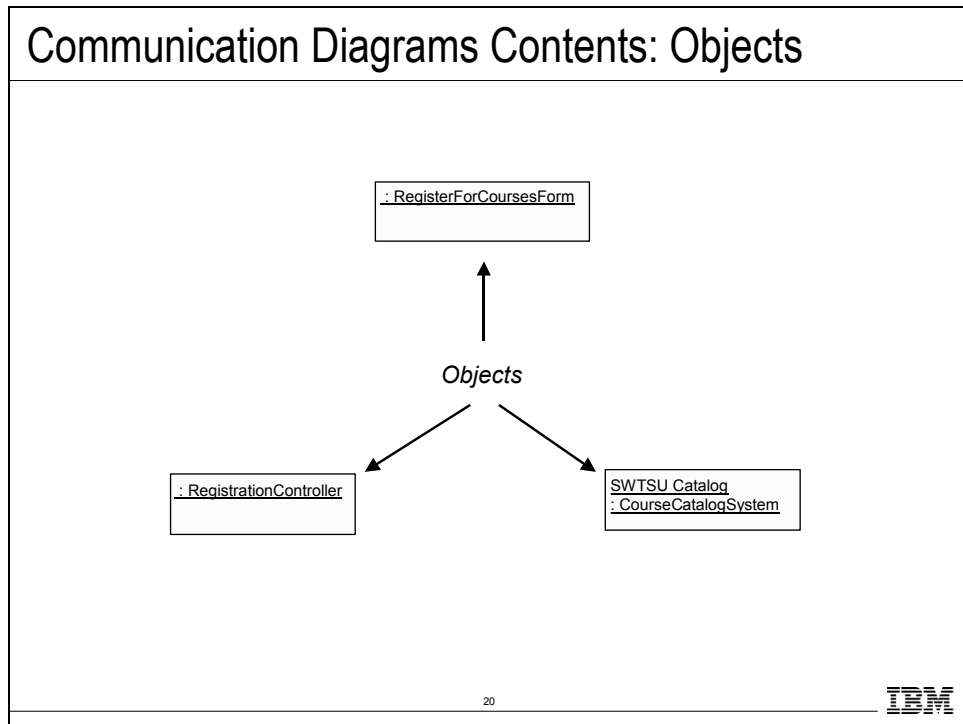
Example: Communication Diagram

You can have objects and actor instances in communication diagrams, together with links and messages describing how they are related and how they interact.

The diagram describes what takes place in the participating objects, in terms of how the objects communicate by sending messages to one another.

You can make a communication diagram for each variant of a use case's flow of events.

The above example shows the communication of objects to support the Register for Courses use case, Create a Schedule sub-flow. It is the "communication diagram equivalent" of the sequence diagram shown earlier.
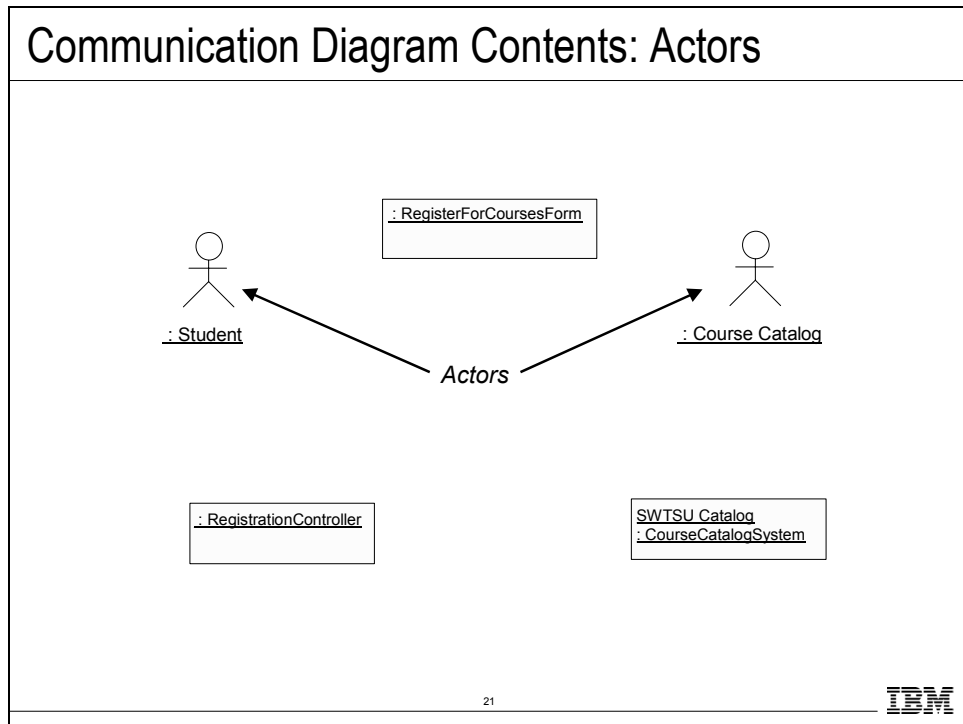
Communication Diagrams Contents: Objects

An **object** is represented by an object symbol, showing the name of the object and its class underlined, separated by a colon.

objectname : classname

You can use objects in communication diagrams in the following ways:

- An object's class can be unspecified. Normally, you create a communication diagram with objects first and specify their classes later.
- The objects can be anonymous. However, you should name them if you want to discriminate different objects of the same class.
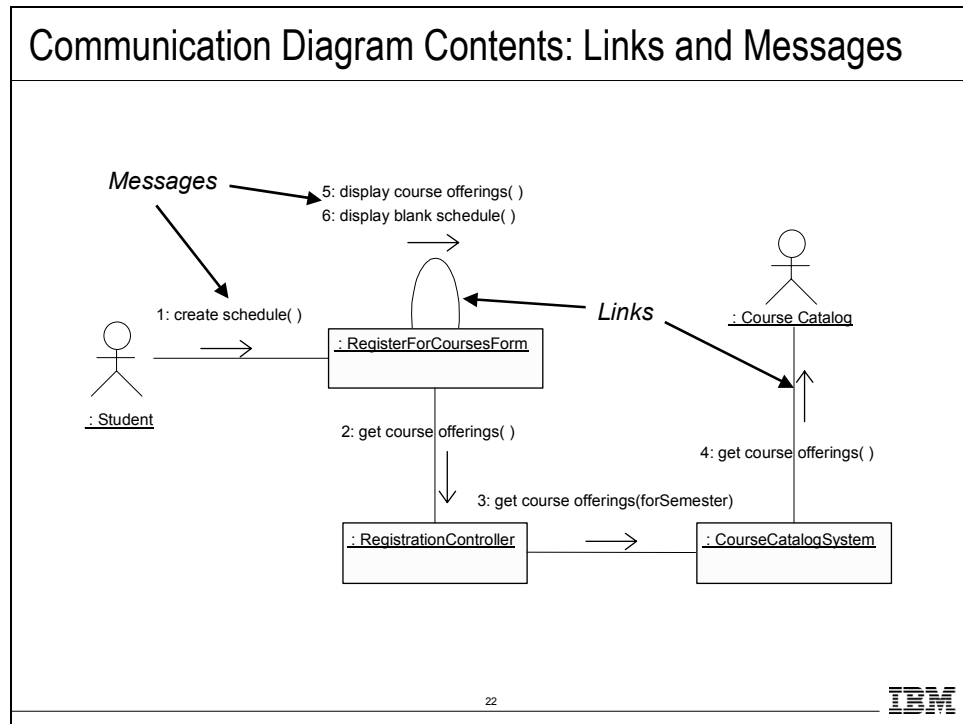
Communication Diagram Contents: Actors

Normally an **actor** instance occurs in the communication diagram as the invoker of the interaction.

If you have several actor instances in the same diagram, try to keep them in the periphery of the diagram.

Don't show the interaction between actors in a communication diagram because actors are, by definition, external to the system.

Communication Diagram Contents: Links and Messages

A **link** is a relationship between objects across which messages can be sent. In communication diagrams, a link is shown as a solid line between two objects. An object interacts with or navigates to other objects through its links to these objects.

A link can be an instance of an association. Or, it can be anonymous, meaning that its association is unspecified.

Message flows are attached to links. A **message** is a communication between objects that conveys information with the expectation that activity will ensue. In communication diagrams, a message is shown as a labeled arrow placed near a link. That is, the link is used to transport or otherwise implement the delivery of the message to the target object.

The arrow points along the link in the direction of the target object (the one that receives the message). The arrow is labeled with the name of the message and its parameters. The arrow may also be labeled with a sequence number to show the sequence of the message in the overall interaction. Sequence numbers are often used in communication diagrams because they are the only way to describe the relative sequencing of messages.

Where Are We?

Where Are We?

- ◆ Sequence diagrams
- ◆ Communication diagrams
- ☆ ◆ Interaction diagram comparison

23

IBM

Sequence and Communication Diagram Similarities

Because they both derive the same information from the UML's metamodel; sequence diagrams and communication diagrams are semantically equivalent. As a result, you can take a diagram in one form and convert it to the other without any loss of information.

Sequence and Communication Diagram Differences

Sequence and Communication Diagram 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

25   IBM

- Sequence and communication diagrams express similar information, but show it in different ways.
- Communication diagrams emphasize the structural communication of a society of objects and show a clearer picture of the pattern of relationships and control that exist among the objects participating in a use case.
- Communication diagrams also show more structural information, such as the relationships among objects.
- Communication diagrams are better for understanding all the effects of a given object and for procedural design.
- Sequence diagrams show the explicit sequence of messages and are better for real-time specifications and complex scenarios.
- A sequence diagram includes chronological sequences but does not include object relationships.
- Sequence numbers are often omitted in sequence diagrams, in which the physical location of the arrow shows the relative sequence.
- On sequence diagrams, the time dimension is easier to read, the operations and parameters are easier to present, and the larger number of objects are easier to manage than in communication diagrams.
- Both sequence and communication diagrams allow you to capture semantics of the use-case flow of events. They help identify objects, classes, interactions, and responsibilities, as well as validate the architecture.
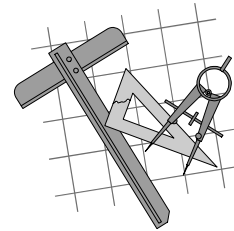
Review

Review

- ◆ What is the purpose of an interaction diagram?
- ◆ What is a sequence diagram? A communication diagram?
- ◆ What is a timing diagram? An interaction overview diagram?
- ◆ What are the similarities between sequence and communication diagrams?
- ◆ What are the differences between sequence and communication diagrams?

26

IBM

© Copyright IBM Corp. 2004

Exercise

Exercise

- ◆ Given:
  - ▪ A set of objects and their links and messages
- ◆ Produce:
  - ▪ A sequence diagram
  - ▪ A communication diagram

27

IBM

You are responsible for creating a sequence and communication diagram for the use case or use cases you are modeling.

Remember, an interaction diagram should model one scenario in the use case. If you are going to model multiple scenarios, you'll need to create an interaction diagram for each scenario. Refer to the following slides if needed.

- • What Is a Sequence Diagram? slides 9-16
- • What Is a Communication Diagram? slides 18-22

Draw sequence and communication diagrams using the following data:

1. The Prospective Buyer actor begins the sequence by requesting the Personal Planner Profile object (PPF) to maintain a profile.
2. The PPF requests the Personal Planner Controller object (PPC) to maintain a profile.
3. The PPC sends a message to the Buyer Record object asking it to find the planner record.
4. The PPF then displays the planner record.
5. The Prospective Buyer updates some information on the profile and asks the PPF to save the profile information.
6. The PPF takes the new information and requests that the PPC save the profile information.
7. The PPC asks the Buyer Record to update the record with the latest information that the actor has provided.
8. The PPC asks the Customer Profile object to create a new profile for the system.

Review your models and describe what the information says.