

Object Oriented Analysis And Design

Aleenah Khan

Chapter 5

Discovering Object Interaction

Use Case Realization

- The use case diagram presents an outside view of the system.
- The functionality of the use case is captured in the flow of events.
- Scenarios are used to describe how use cases are realized as interactions among societies of objects.

Use Case Realization

Scenario:

- A scenario is an instance of a use case—it is one path through the flow of events for the use case.
- Scenarios are developed to help identify the objects, the classes, and the object interactions needed to carry out a piece of the functionality specified by the use case.
- Scenarios document decisions about how the responsibilities specified in the use cases are distributed among the objects and classes in the system.
- They also provide an excellent communication medium to be used in the discussion of the system requirements with customers.
- Scenarios speak the language of the end user and the domain expert, and therefore provide a means for them to state their expectations about the desired behavior of a system to its developers.

Use Case Realization

Each use case is a web of scenarios:

- **Primary Scenarios:** The normal flow for the use case.
- **Secondary Scenarios:** The what-if logic of the use case.

This means that there are numerous scenarios for any given system—all the primary and secondary scenarios for all the use cases.

Use Case Realization

- During the early stage of analysis it is safe to say that looking at the primary scenarios for each identified use case is enough.
- When you find that each new scenario is repeating a lot of steps from previously identified scenarios, then you have reached the finish line.
- This phase of analysis should be drawn to a close once the team has elaborated approximately 80 percent of a system's primary scenarios along with a representative selection of the secondary ones.

Use Case Realization

- *Elaborate upon any more, and your analysis will likely reach diminishing returns.*
- *Elaborate upon any fewer, and you won't have a sufficient understanding of the desired behavior of the system to properly understand the risks.*

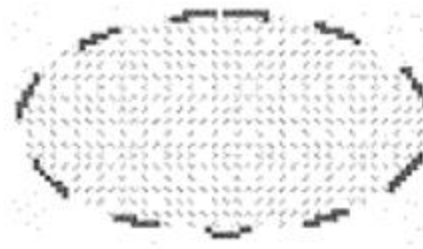
Use Case Realization

- In the Rational Unified Process, use case realizations are captured in the Logical View of the model.
- We use stereotype to show that the use cases that we create in the Logical View of our model are the realizations of the use cases contained in the Use Case View of our model.
- In other words, the use cases in the Logical View have the same name as the use cases in the Use Case View along with a stereotype of Realization.

Use Case Realization

- In the UML, use case realizations are drawn as dashed ovals as shown in Figure 5-1.
- These Logical View use cases typically are shown on a use case diagram (or set of use case diagrams) contained in the Logical View of your model.

Figure 5-1. UML Notation for a Use Case Realization



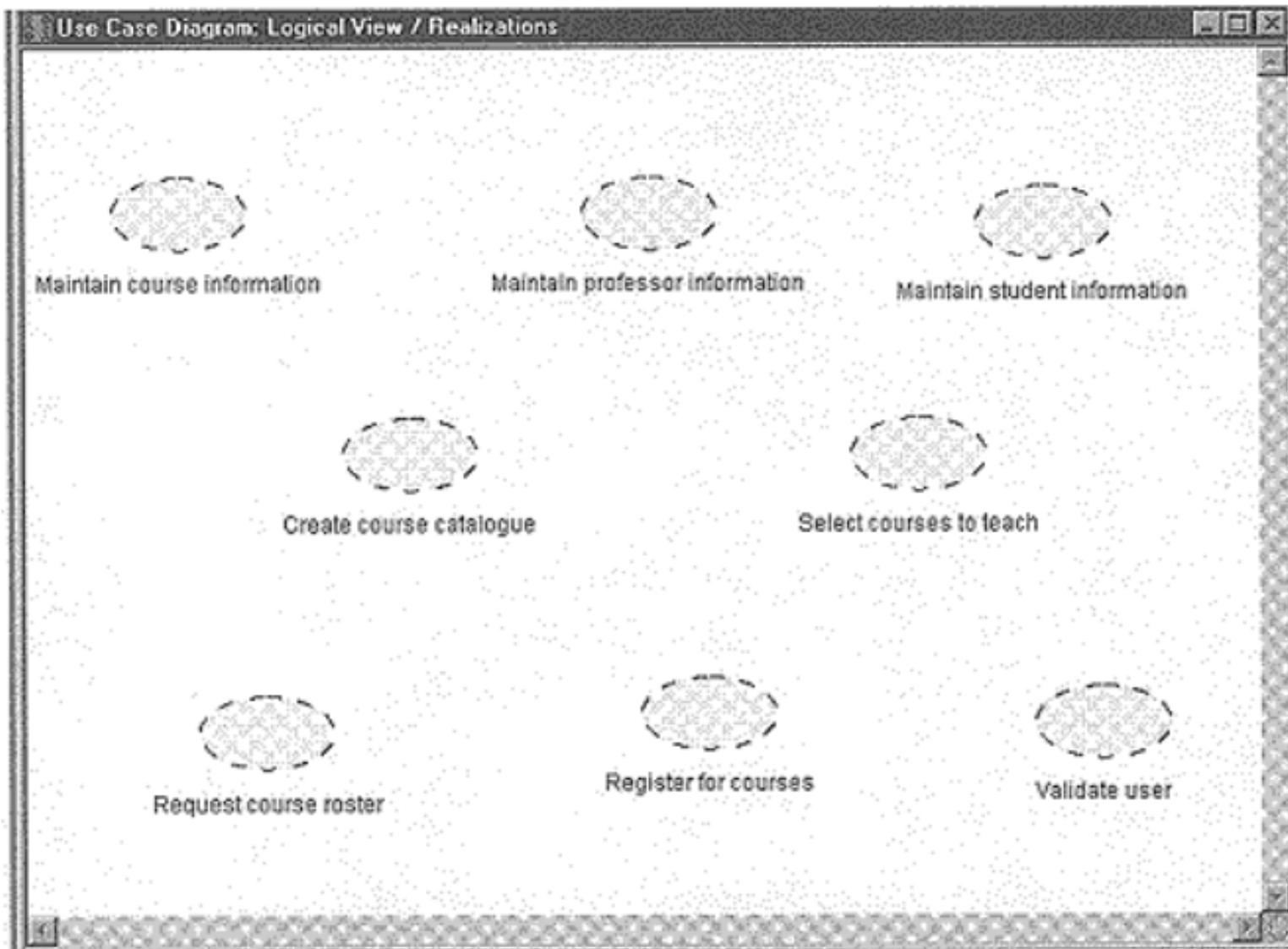
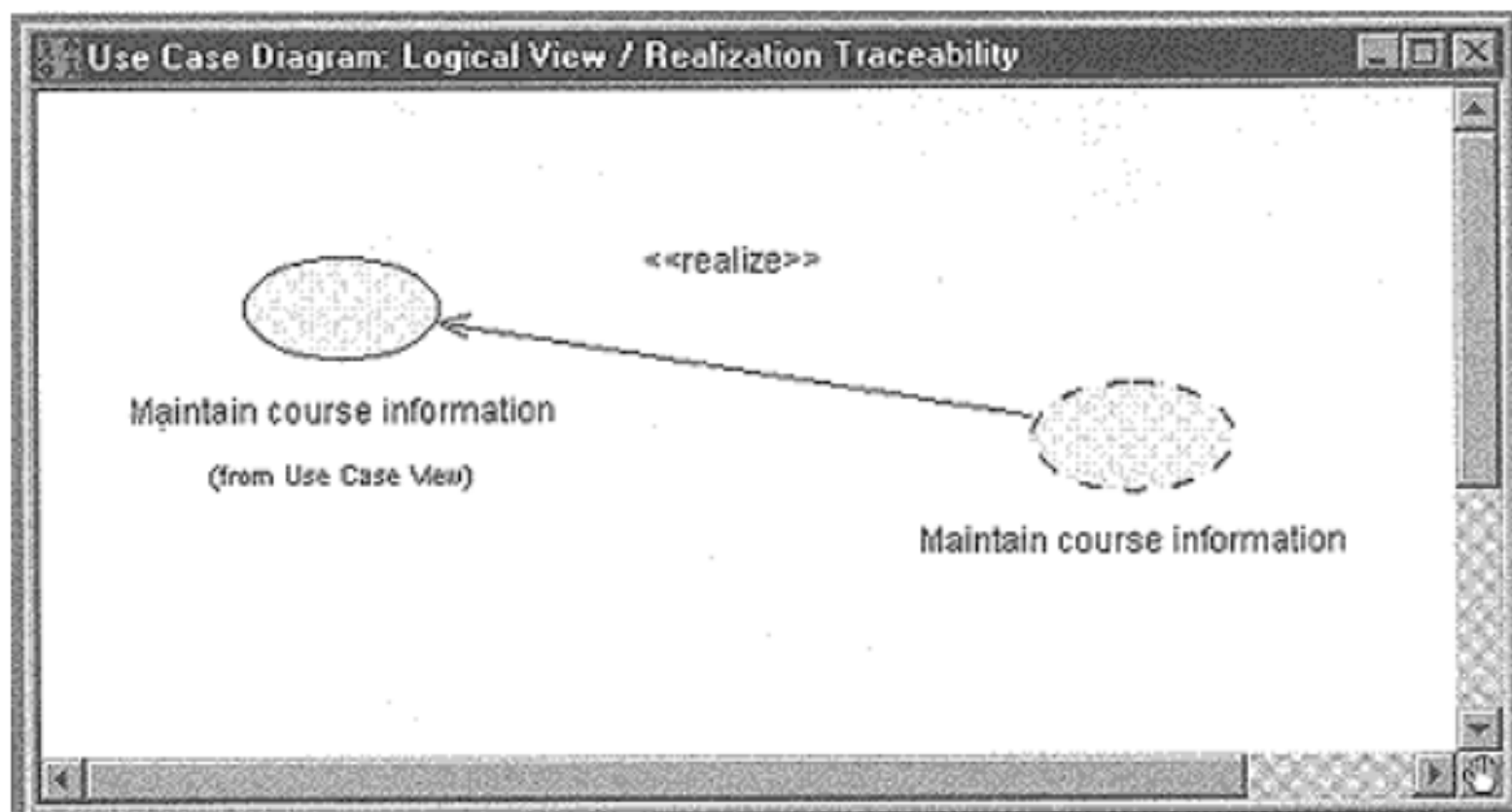


Figure 5-3. Use Case Realization Diagram

Traceability

- Traceability between the use cases in the Logical View and the use cases in the Use Case View is visualized by adding the Use Case View use cases to the Realizations diagram and connecting them to their realizations using a stereotyped unidirectional association.

Figure 5-4. Use Case Realization Diagram



Documenting Scenarios

- The flow of events for a use case is captured in text.
- Scenarios are captured in interaction diagrams.
- There are two types of interaction diagrams:
 - Sequence Diagrams
 - Collaboration Diagrams

Each diagram is a graphical view of the scenario.

Sequence Diagrams

- A sequence diagram shows object interactions arranged in time sequence.
- It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.
- Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development.

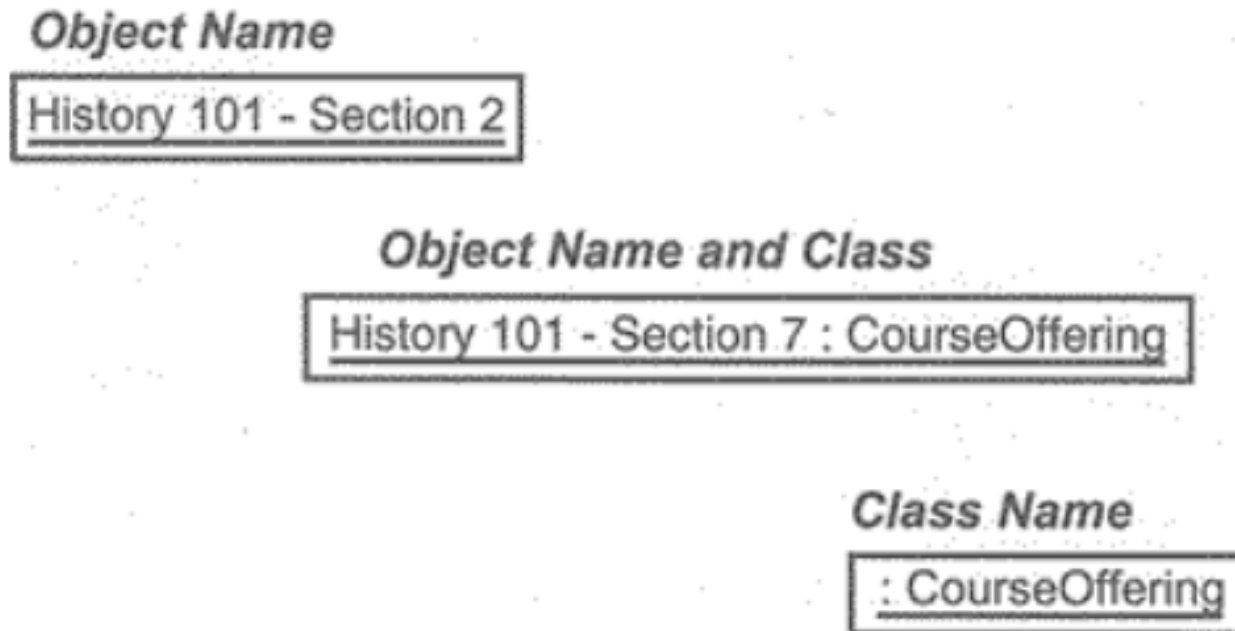
Sequence Diagrams

- In the UML, an object in a sequence diagram is drawn as a rectangle containing the name of the object, underlined.
- An object can be named in one of three ways:
 - the object name,
 - the object name and its class, or
 - just the class name (anonymous object)

Sequence Diagrams

- The three ways of naming an object are shown in Figure 5-5.

Figure 5-5. Naming Objects in a Sequence Diagram



Sequence Diagrams

- Each object also has its timeline represented by a dashed line below the object.
- Messages between objects are represented by arrows that point from the client (sender of the message) to the supplier (receiver of the message).
- The UML notation for objects and messages in a sequence diagram is shown in Figure 5.6.

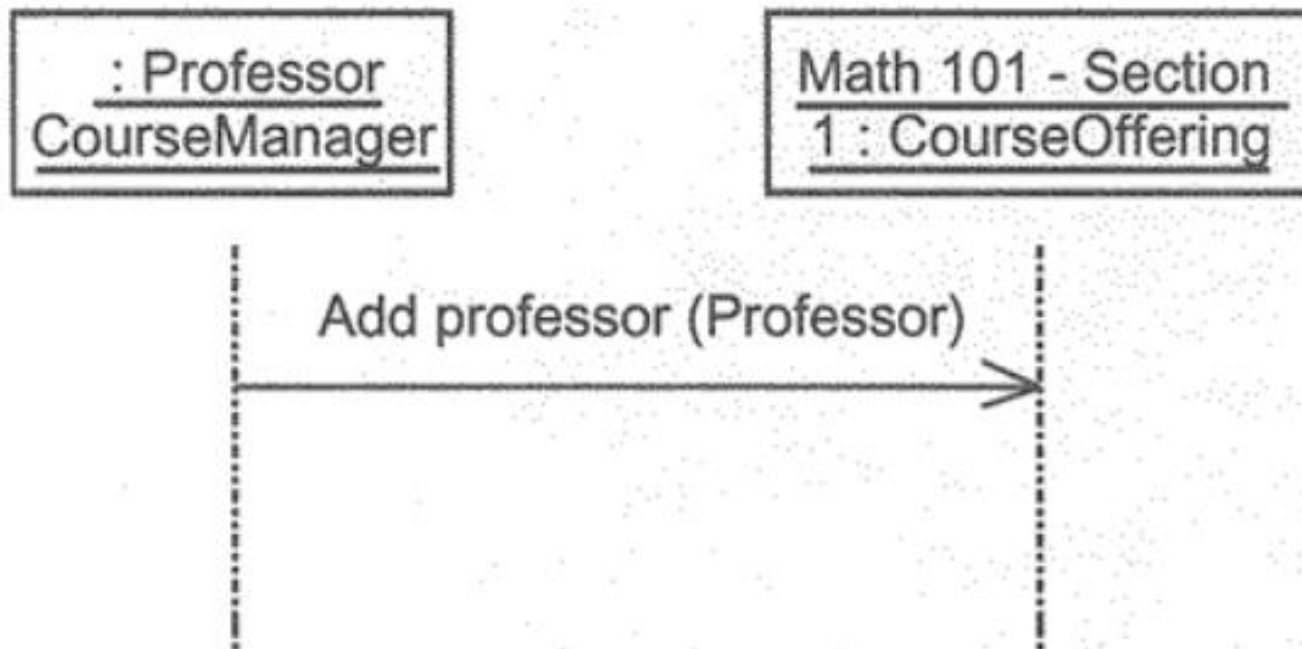
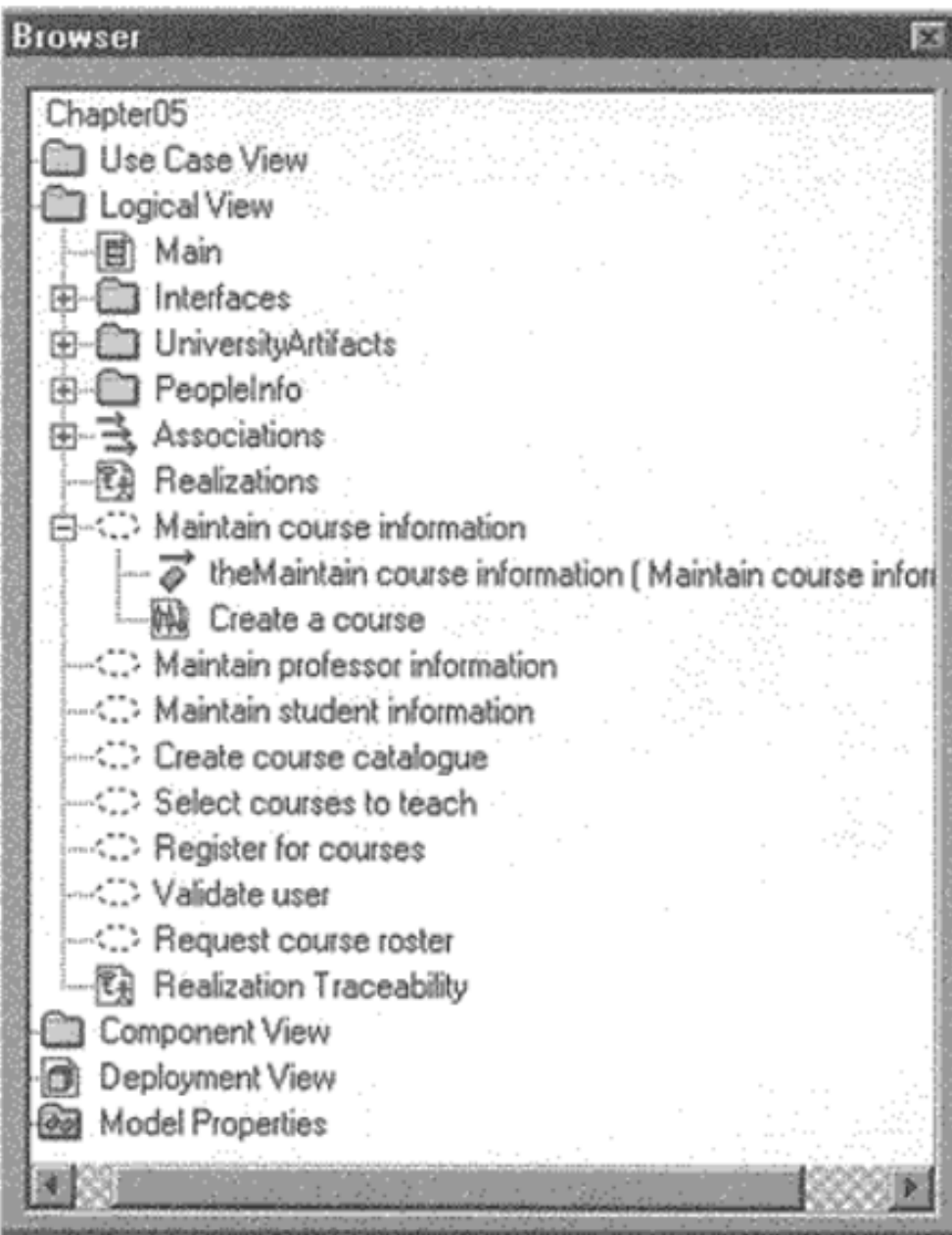
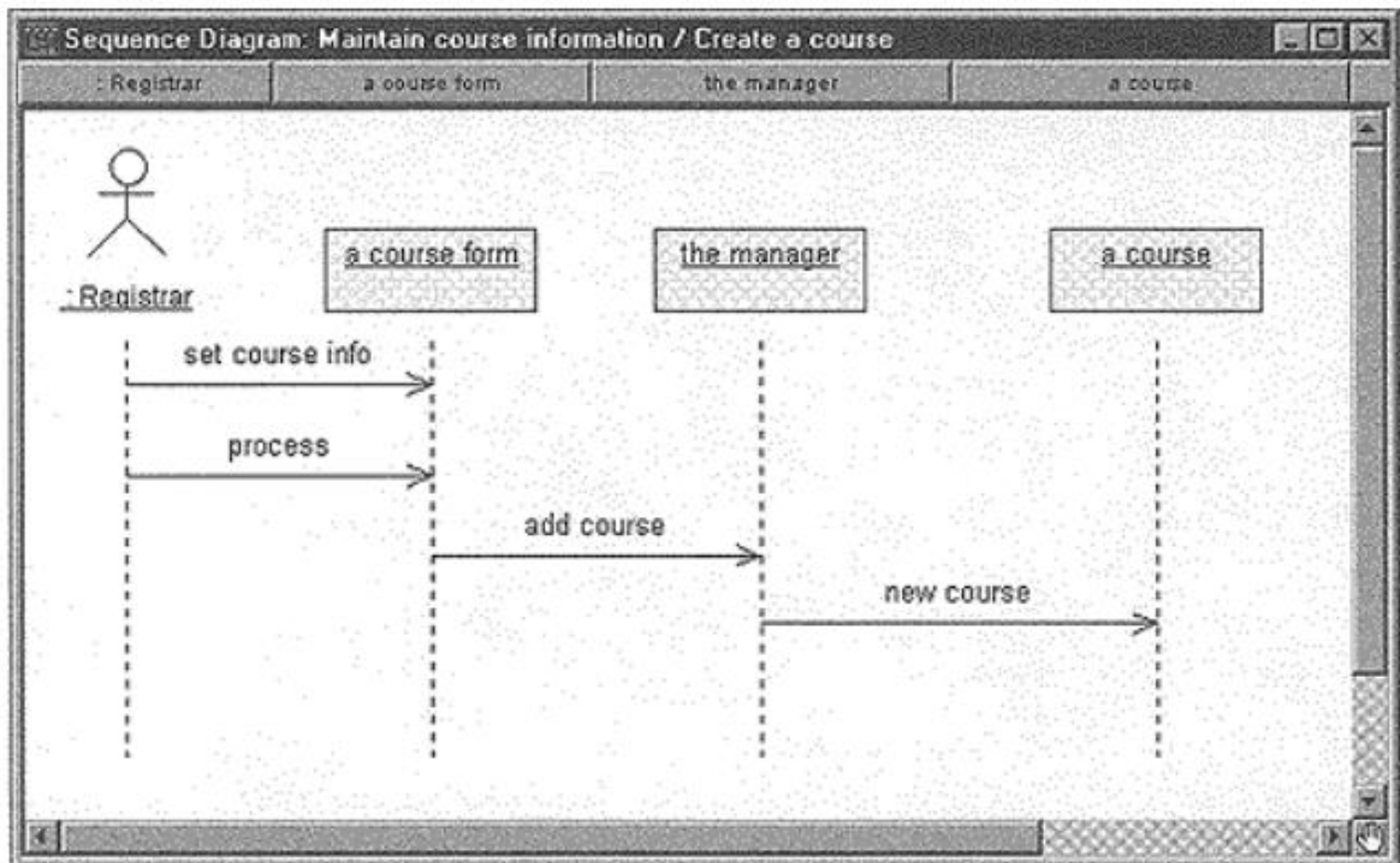


Figure 5.6. UML Notation for Objects and Messages in a Sequence Diagram



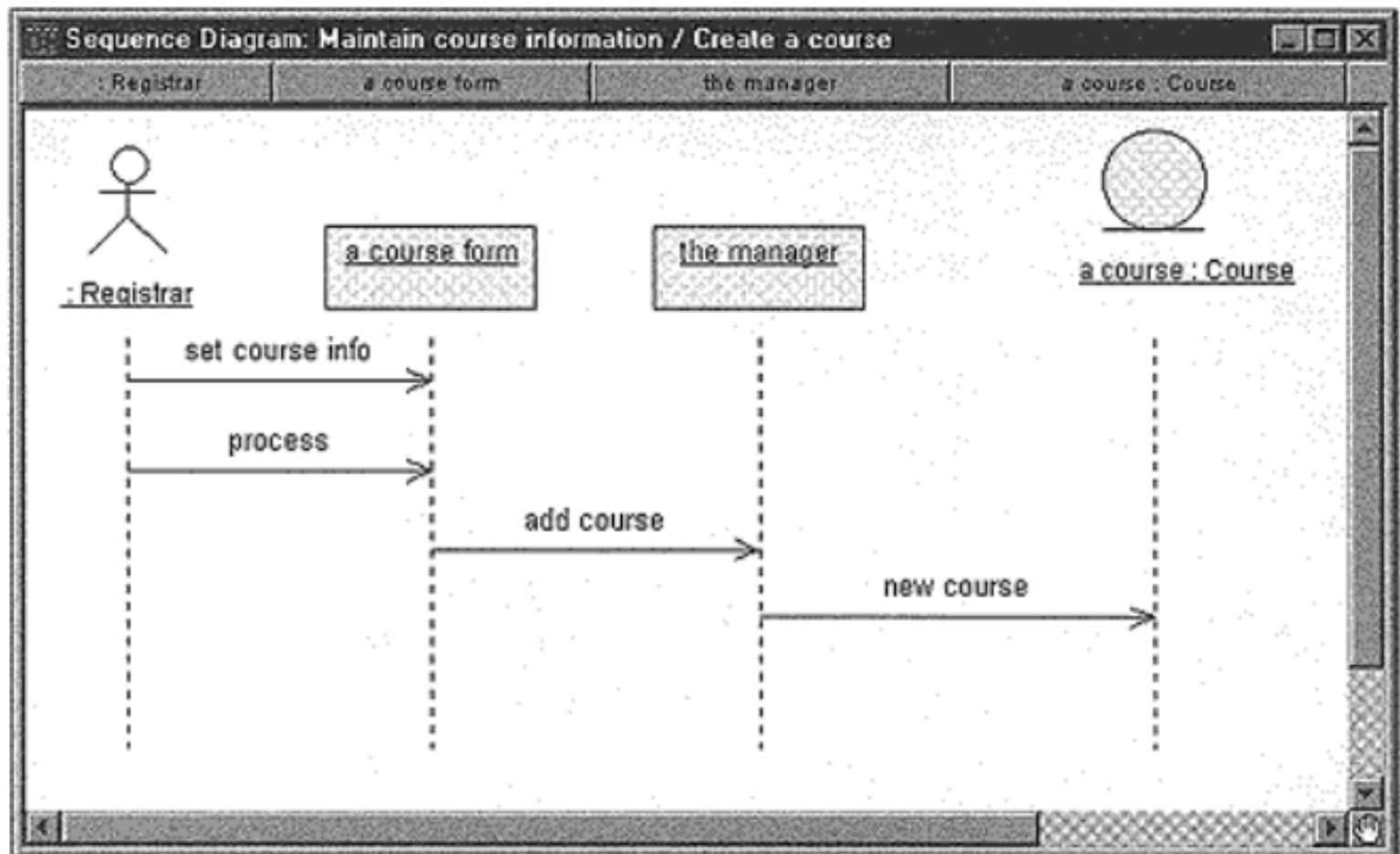
The sequence diagram for the *Create a Course* scenario is shown in [Figure 5.8](#)

Figure 5-8. Sequence Diagram



A sequence diagram with the object "a course" assigned to the Course class is shown in [Figure 5.9](#).

Figure 5-9. Sequence Diagram with an Object Assigned to a Class



Sequence Diagrams And Boundary Classes

- Boundary classes are added to sequence diagrams to show the interaction with the user or another system.
- In the early analysis phases, the purpose of showing boundary classes in sequence diagrams is to capture and document the **interface requirements**, *not* to show how the interface will be implemented.
- The actual messages from the actor to the boundary class along with their sequencing information are dependent upon the application framework that is chosen later in development.
- There will probably be change as more of the "how" is added to the system.

Complexity And Sequence Diagrams

- **"How complex can a sequence diagram be?"**
 - "Keep them simple."
 - The beauty of these diagrams is their simplicity—it is very easy to see the objects, the object interactions, the messages between the objects, and the functionality captured by the scenario.

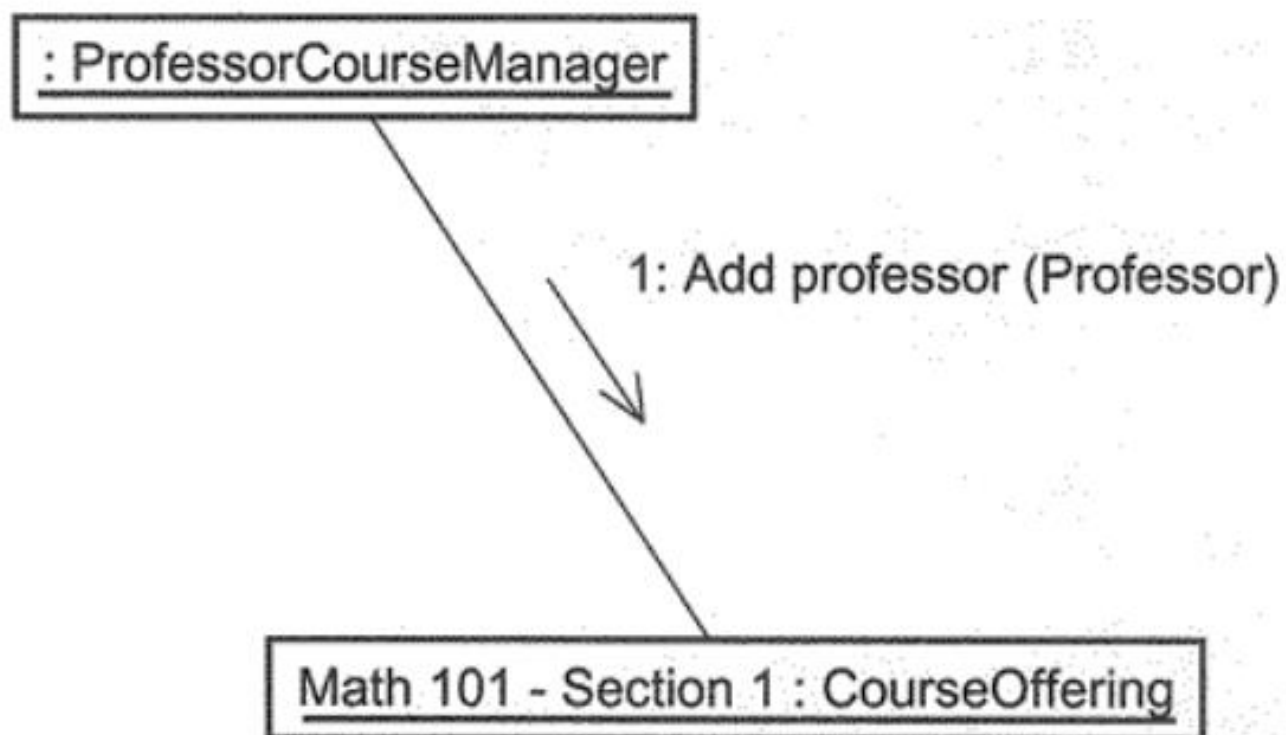
Complexity And Sequence Diagrams

- "What do I do about conditional logic?" (all the *if, then, else* logic that exists in the real world)
 - If the logic is simple, involving only a few messages, add the logic to one diagram and use notes and scripts to communicate the choices to be made.
 - On the other hand, if the *if, then, else* logic involves many complicated messages, draw a separate diagram—one for the *if* case, one for the *then* case, and one for the *else* case.
 - This is done to keep the diagrams simple.
 - These diagrams may be linked to one another.
 - This allows the user to navigate through a set of diagrams.

Collaboration Diagrams

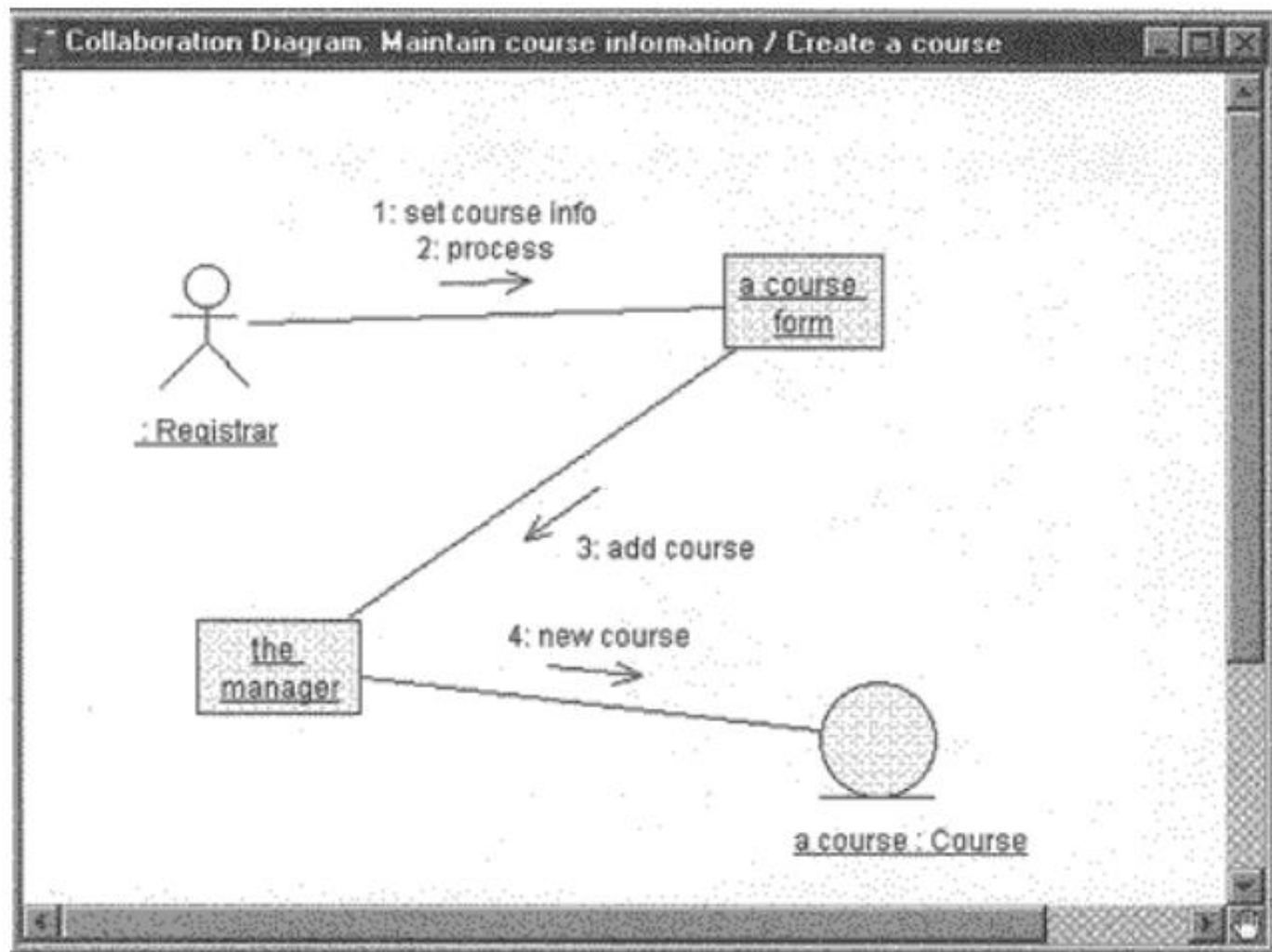
- A collaboration diagram is an alternate way to show a scenario.
- This type of diagram shows object interactions organized around the objects and their links to each other.
- A collaboration diagram contains:
 - Objects drawn as rectangles
 - Links between objects shown as lines connecting the linked objects
 - Messages shown as text and an arrow that points from the client to the supplier

Figure 5-10. UML Notation for Objects, Links, and Messages in a Collaboration Diagram



The collaboration diagram is shown in [Figure 5.11](#).

Figure 5-11. Collaboration Diagram



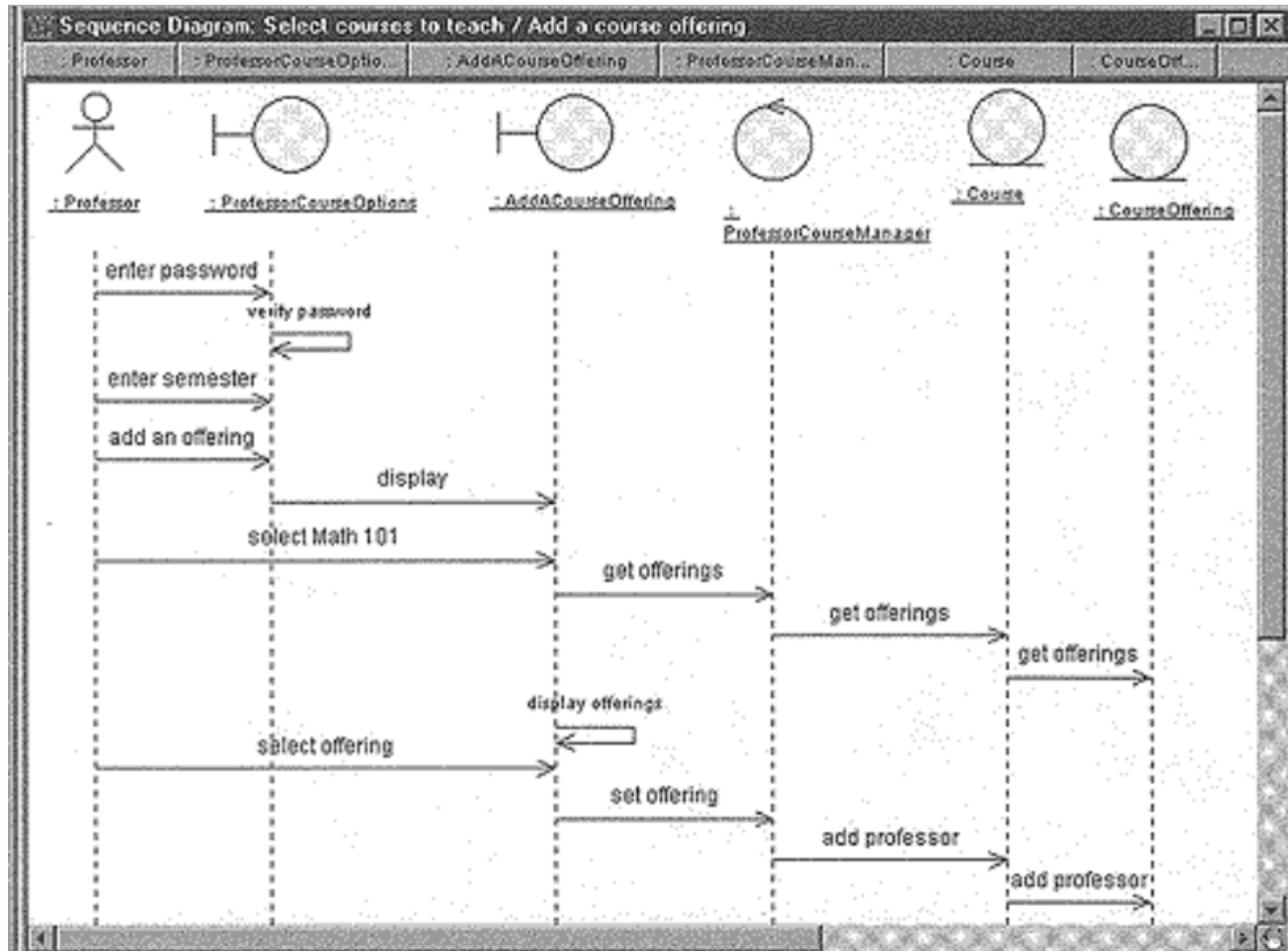
Collaboration Diagrams

- Collaboration diagrams can also be created from scratch. In that case, a sequence diagram can be created from the collaboration diagram.

Why Two Different Diagrams?

- **Sequence diagrams** provide a way to look at a scenario in a **time-based order**—what happens first, what happens next.
 - Customers easily can read and understand this type of diagram.
 - Hence, they are very useful in the early analysis phases.
- **Collaboration diagrams** tend to provide the big picture for a scenario since the collaborations are organized around the object links to one another.
 - These diagrams seem to be used more in the design phase of development when you are designing the implementation of the relationships.

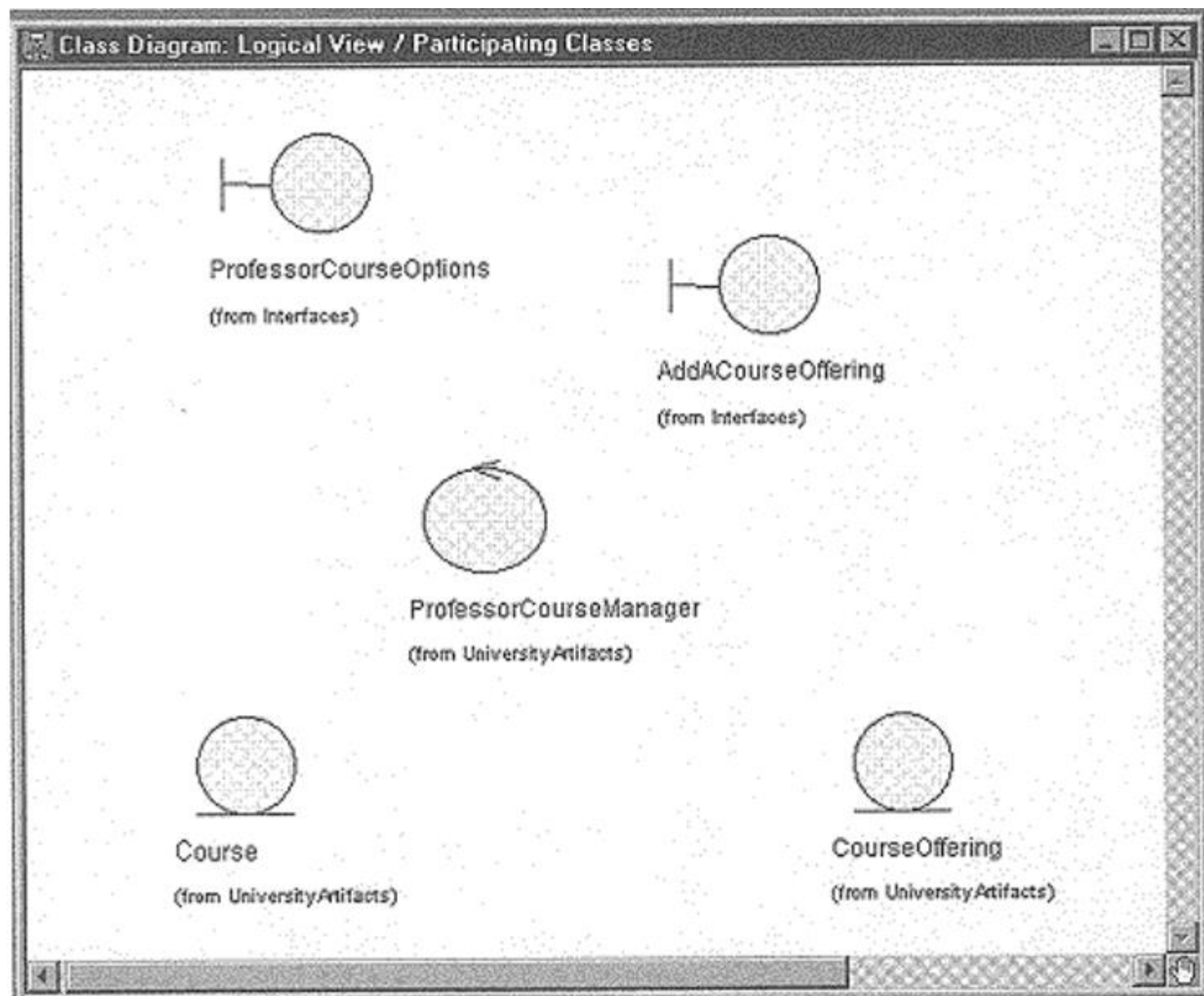
Figure 5.12. Sequence Diagram for the Add a Course Offering Scenario



Class Diagrams with Use Case Realizations

- Class diagrams may also be attached to use case realizations.
- These diagrams contain a view of the classes participating in the use case.

Figure 5-13. View of Participating Classes



Useful Links

- Sequence Diagram Tutorials

https://www.youtube.com/watch?v=XIQKt5Bs7II&fbclid=IwAR3om1J4UTjV_1RzYB3d5IGqmhXxG2jGtMDVe6-j-NtHWzkGCsP7u8BrEBk

https://www.youtube.com/watch?v=pCK6prSq8aw&fbclid=IwAR0c2M99N9_9wKaLYOj1wZsESQwMuokxtbc2An1Puc_9v1E2sU1afZUFWjE

- Collaboration Diagram Tutorials

https://www.youtube.com/watch?v=yPAB2M_fuGA&fbclid=IwAR1Autelj-8PqitRni-gYZ1q4iACo3df2X2LpzawR0EmTRrZgUSu2ej4dAM