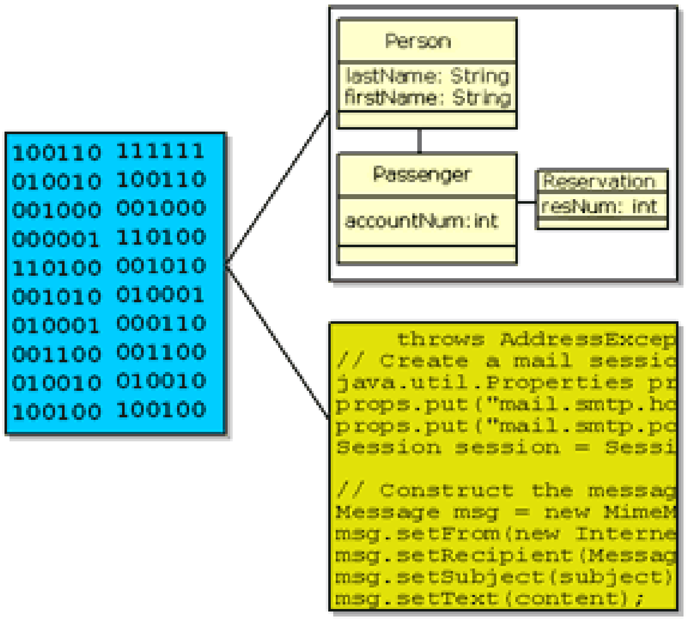
IBM Software Group

Rational Software Architect Workshop

UML Diagrams

What is a Model?

- A model is a semantically closed abstraction of a subject system.
  - A model is defined in RUP as “a complete description of a system from a particular perspective.”
- Examples of models:
  - UML model
  - Code
  - Data model

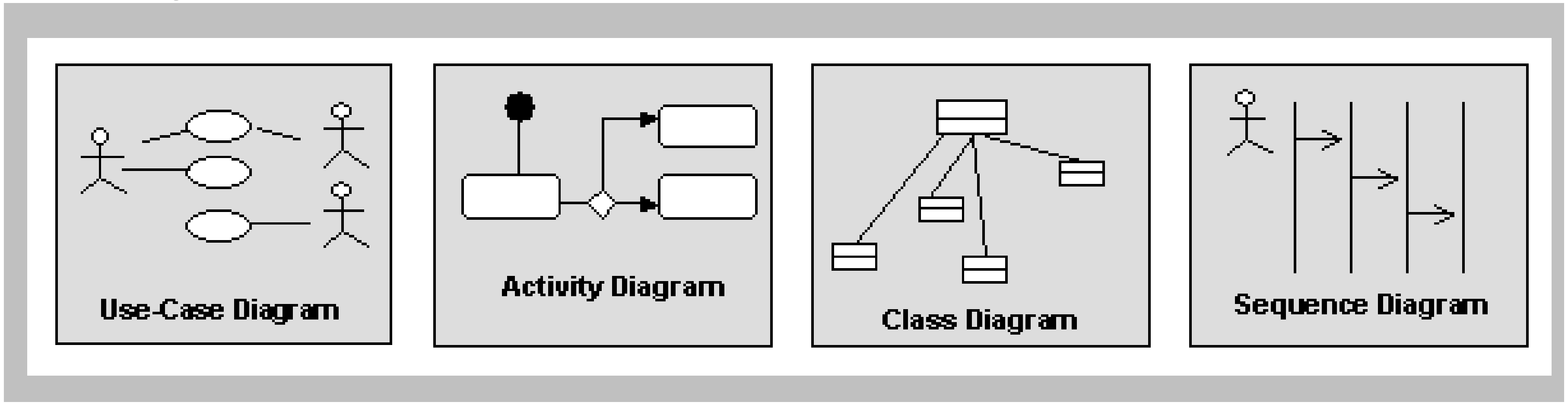


```
100110 111111
010010 100110
001000 001000
000001 110100
110100 001010
001010 010001
010001 000110
001100 001100
010010 010010
100100 100100
```

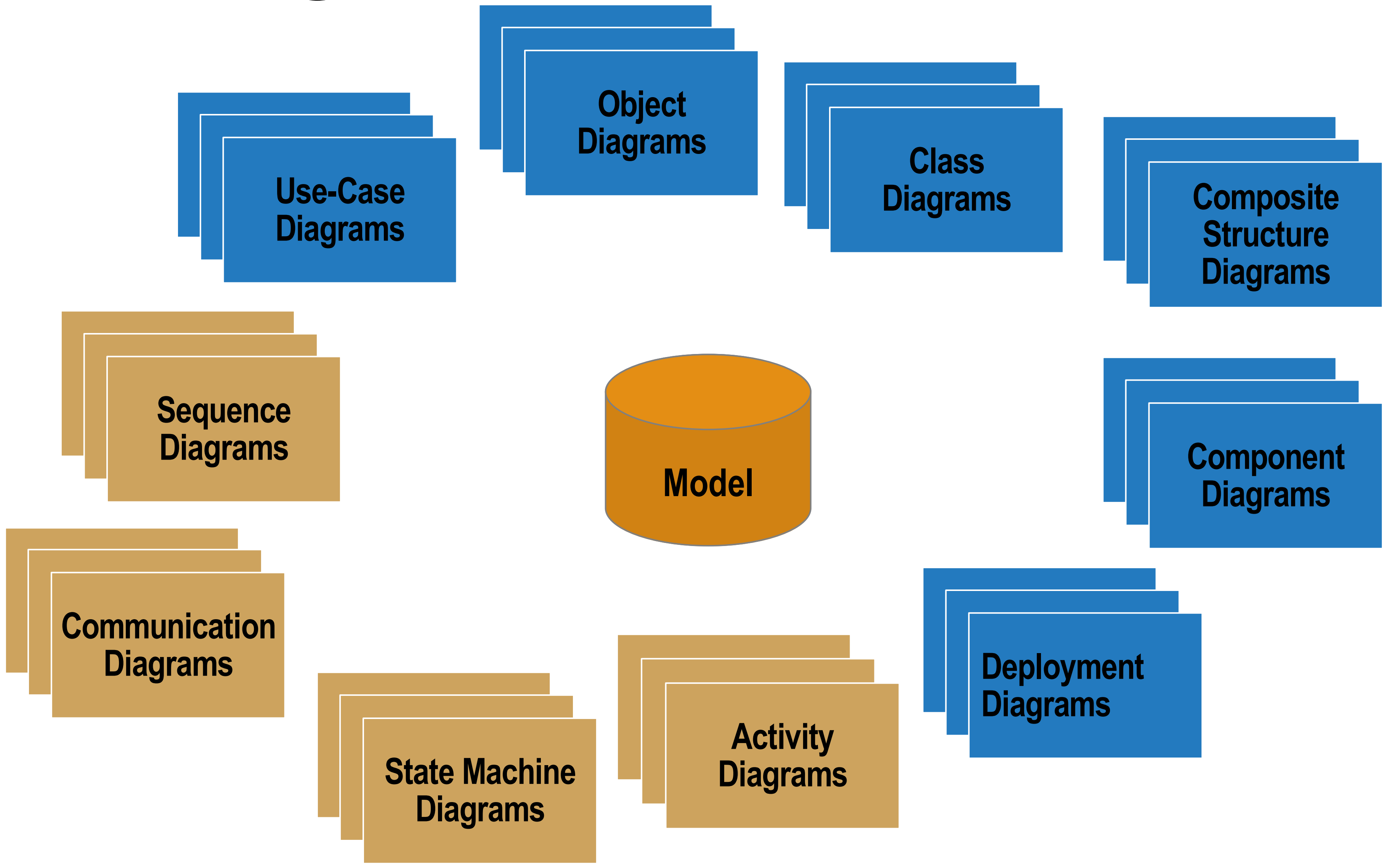
```
    throws AddressExcep
// Create a mail sessio
java.util.Properties pr
props.put("mail.smtp.ho
props.put("mail.smtp.po
Session session = Sessi

// Construct the messag
Message msg = new MimeM
msg.setFrom(new Interne
msg.setRecipient(Messag
msg.setSubject(subject)
msg.setText(content);
```

Diagrams

- Diagrams graphically depict a view of a part of your model.
- Different diagrams represent different views of the system that you are developing.
- A model element will appear on one or more diagrams.



UML Diagrams in Software Architect

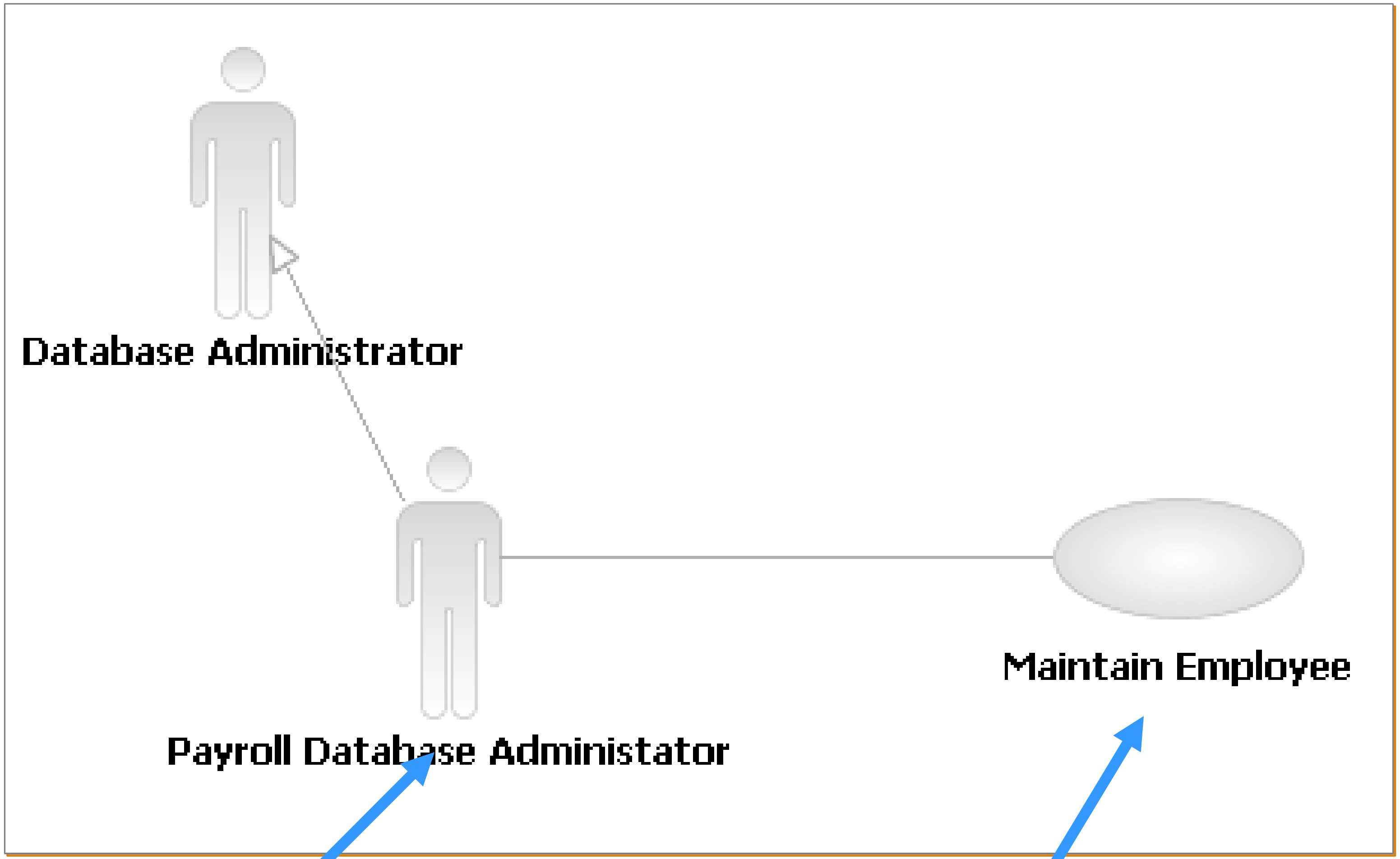


What is a Use-Case Model?

A use-case model:

- Is a model of a system’s intended functions and its environment
- Serves as a contract between the customer and the developers
- Contains the following diagrams:
  - Use case: Shows a set of use cases and actors and their relationships
  - Activity: Shows the flow of events within a use case
  - Sequence: Shows how a use case will be implemented in terms of collaborating objects

Use-Case Diagram



Actor
Someone or something that
Interacts with the system

Use Case
Units of system behavior

Activity Diagram

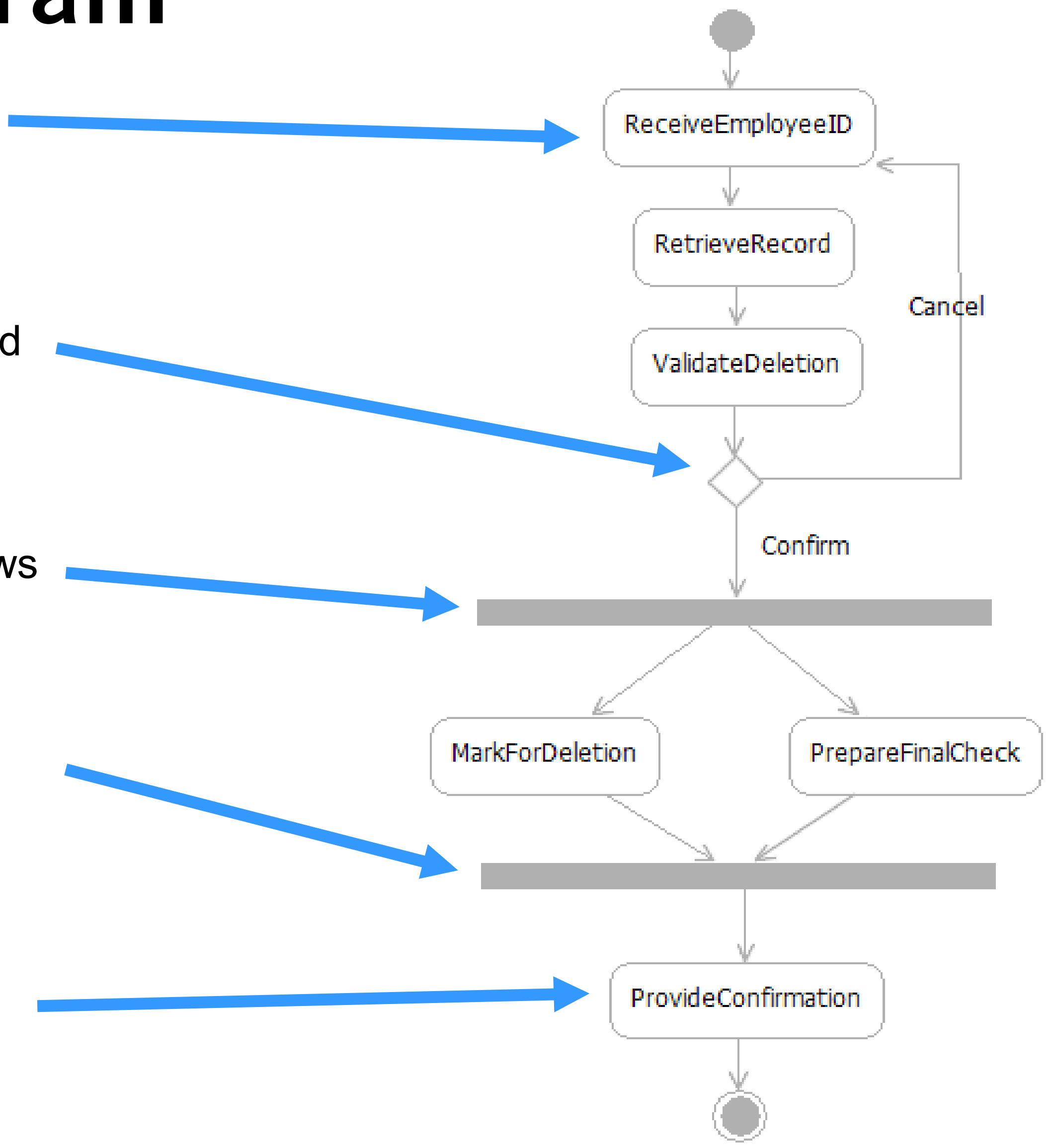
Action
A step in the flow of events

Decision
Flows split based on a guard condition

Fork
Beginning of concurrent flows

Join
End of concurrent flow

Flow
Show the sequence of activities



What is a Design Model?

A design model:

- Describes the realization of use cases in terms of design elements
- Describes the design of the application
- Contains the following diagrams:
  - Class: Shows UML classes and relationships
  - Component: Shows the structure of elements in the implementation model
  - Communication and Sequence: Show how objects and classes interact
  - State Machine: Shows event-driven behavior

Class Diagram (Design Model)

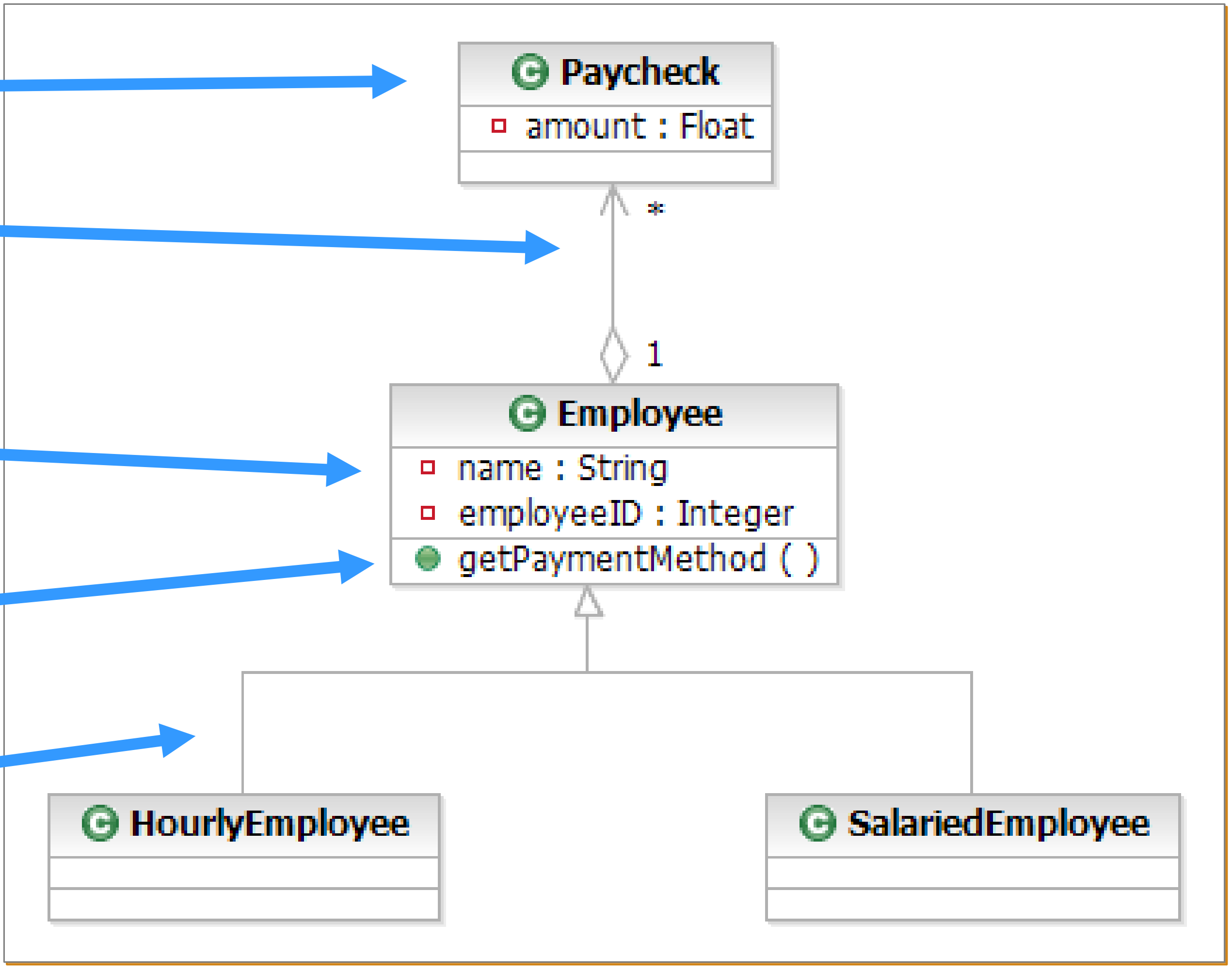
Class
A description of a set of objects

Aggregation
Represents a part-whole relationship

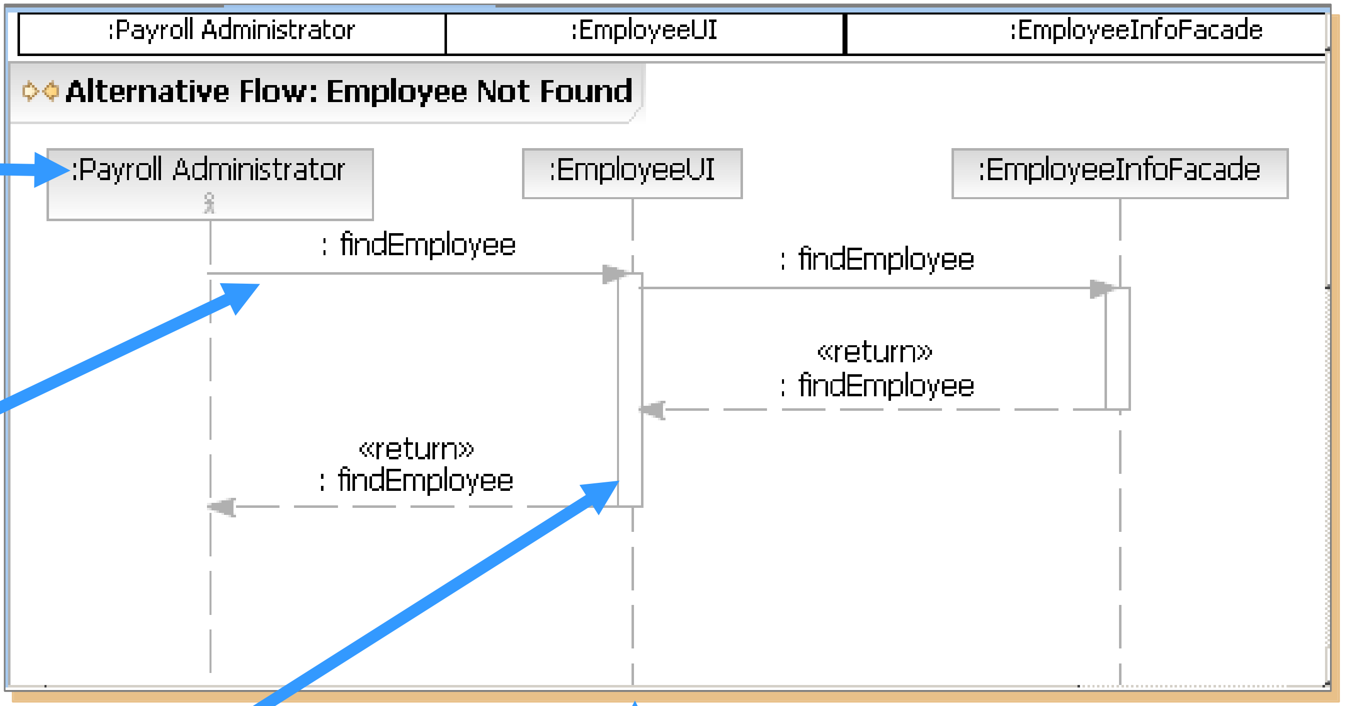
Attribute
Named property of a class

Operation
Class behavior

Generalization
Shows an inheritance relationship



Sequence Diagram



Object/Class
Shows the object/class involved in the interaction

Messages
Show data exchanged between objects

Execution Occurrence
Shows object executing

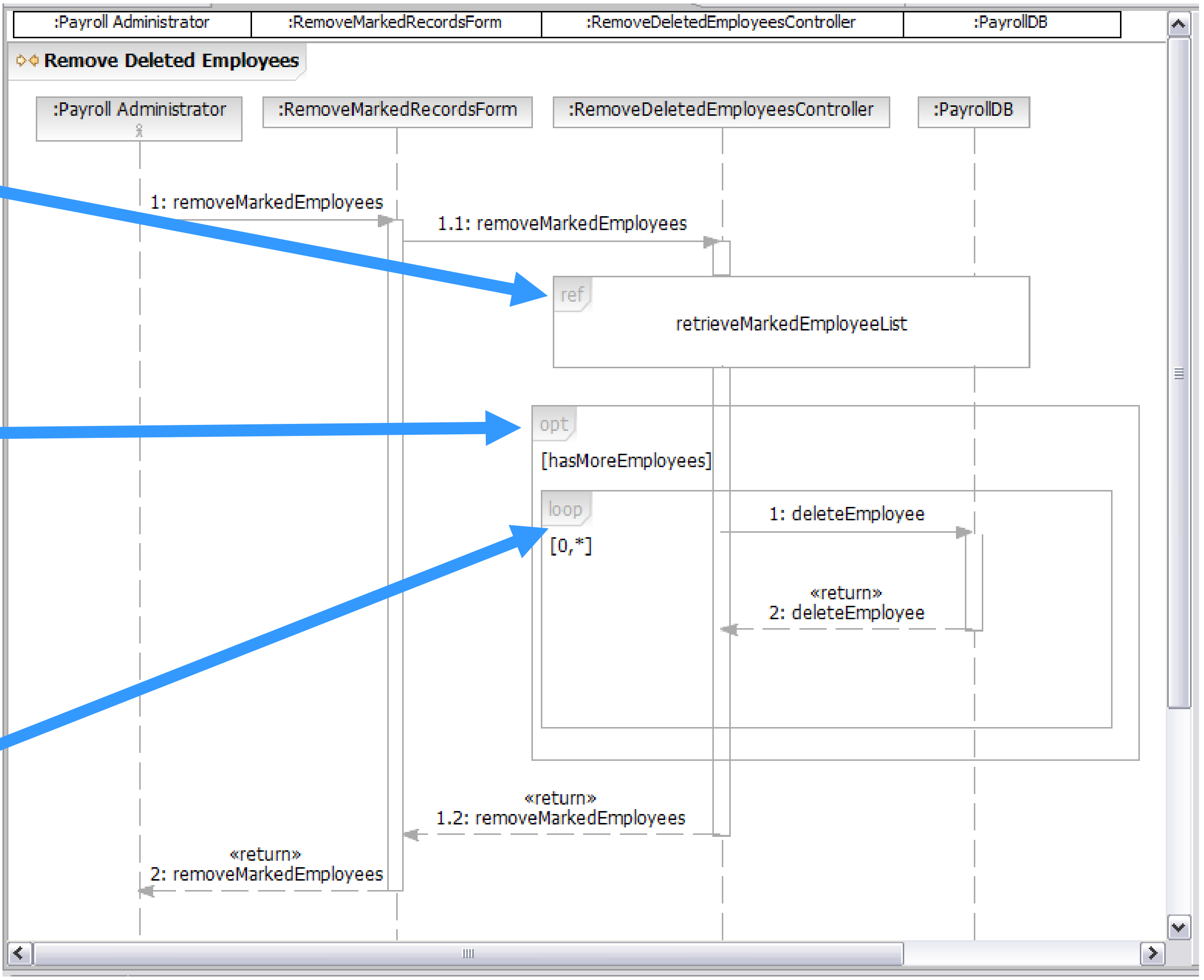
Lifeline
Shows the life of the object

Sequence Diagram: Combined Fragments

Interaction Use (ref)
References another interaction

Optional Fragment (opt)
Executed if guard condition evaluates to true

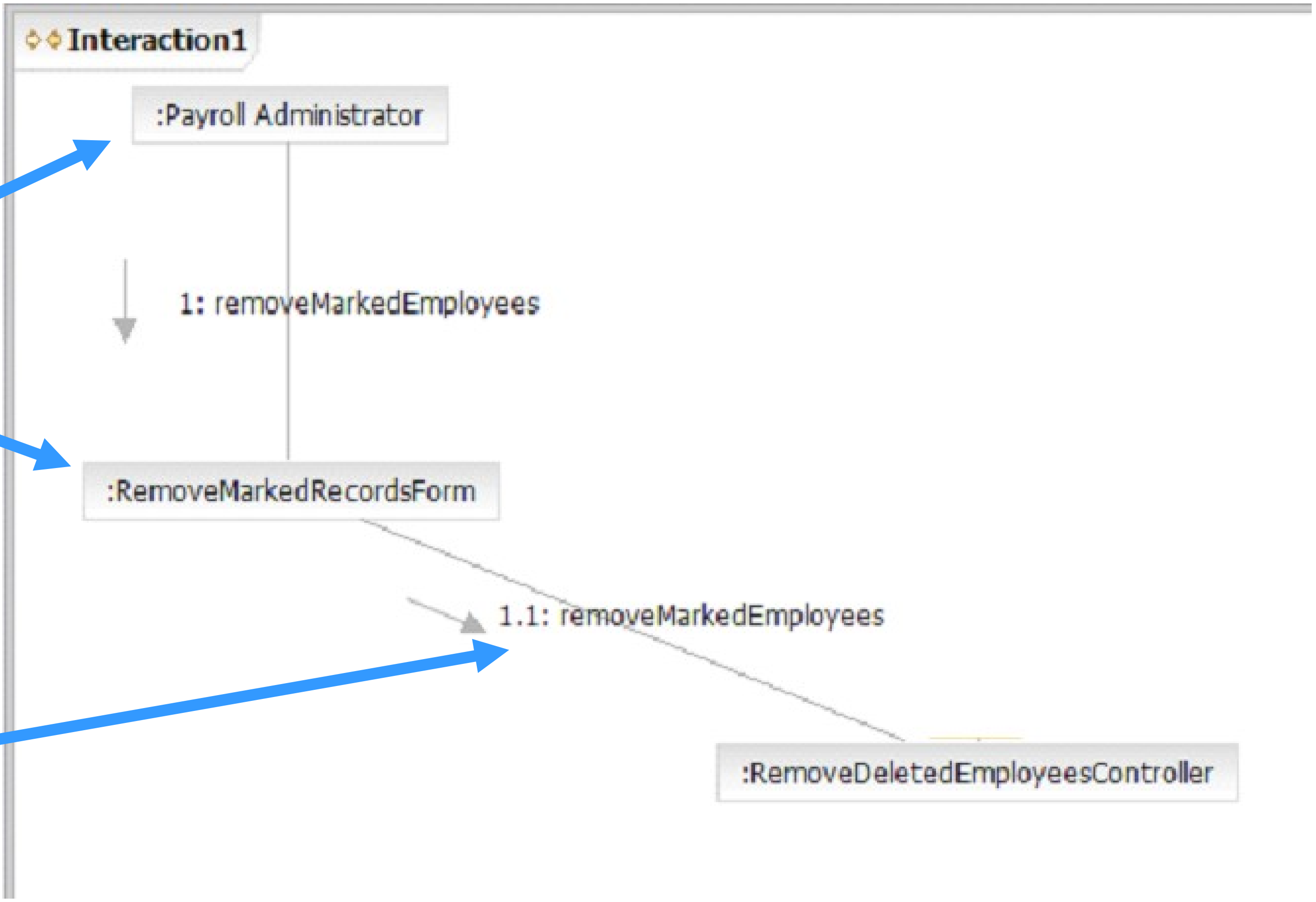
Loop (loop)
Executed as long as the first guard condition evaluates to true



Communication Diagram

Object/Class
Shows the object/class involved in the interaction

Message
Shows data exchanged between objects

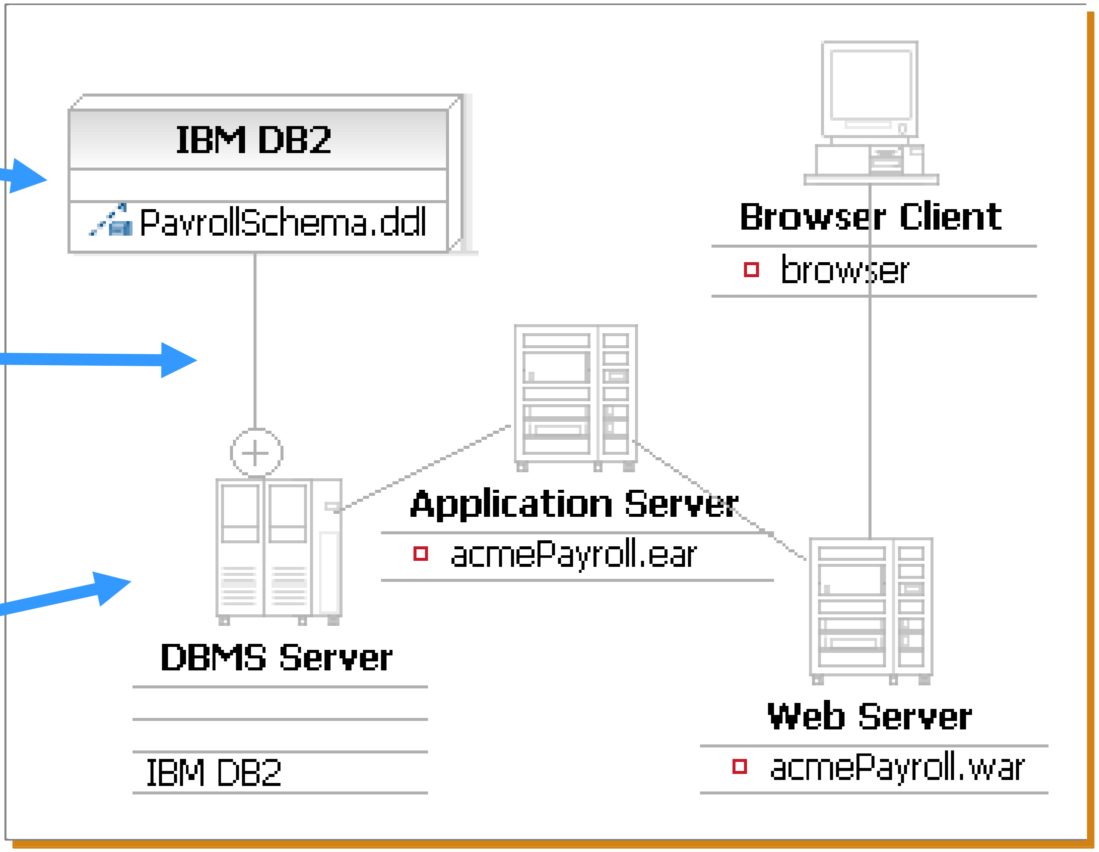


Deployment Diagram

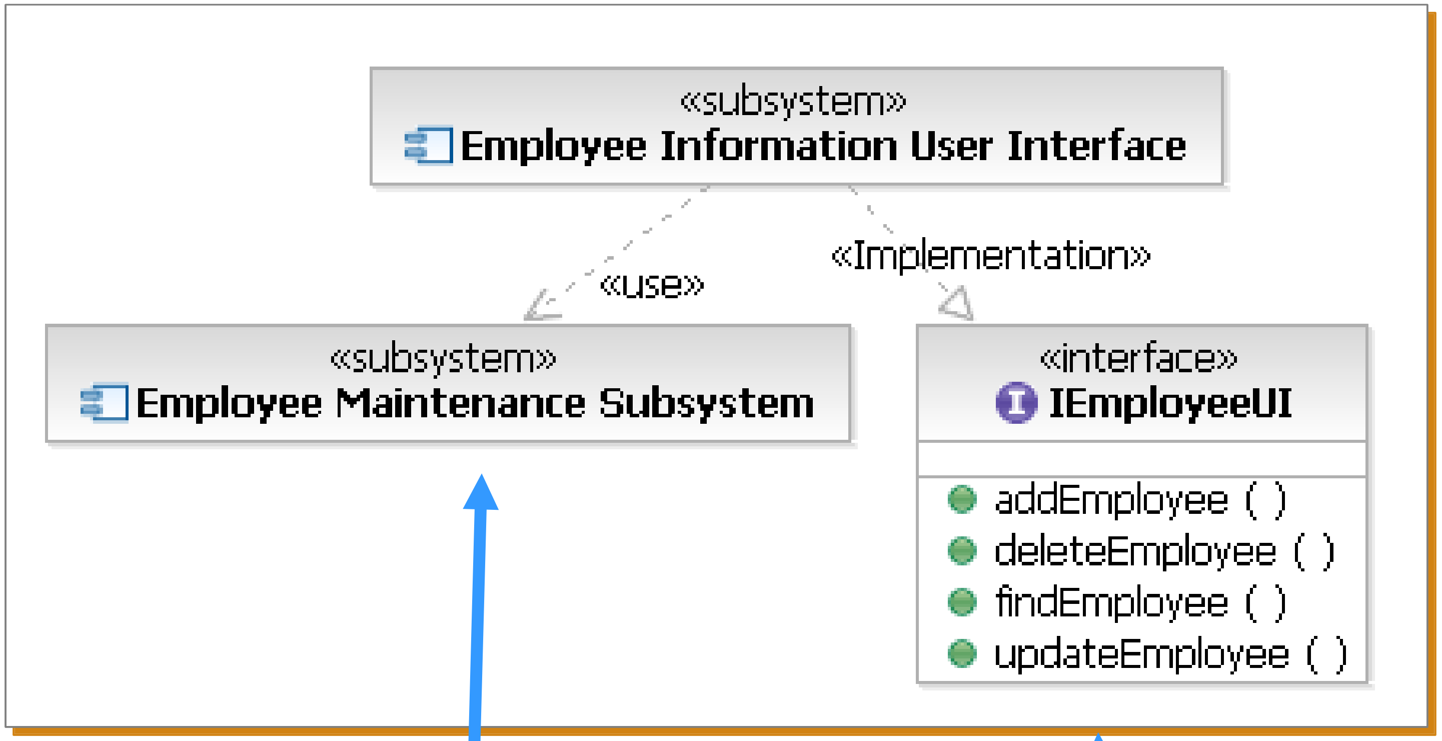
Artifact
Represents a physical file

Owned Element Relationship
Shows another way of showing nested elements

Node
Represents a physical machine



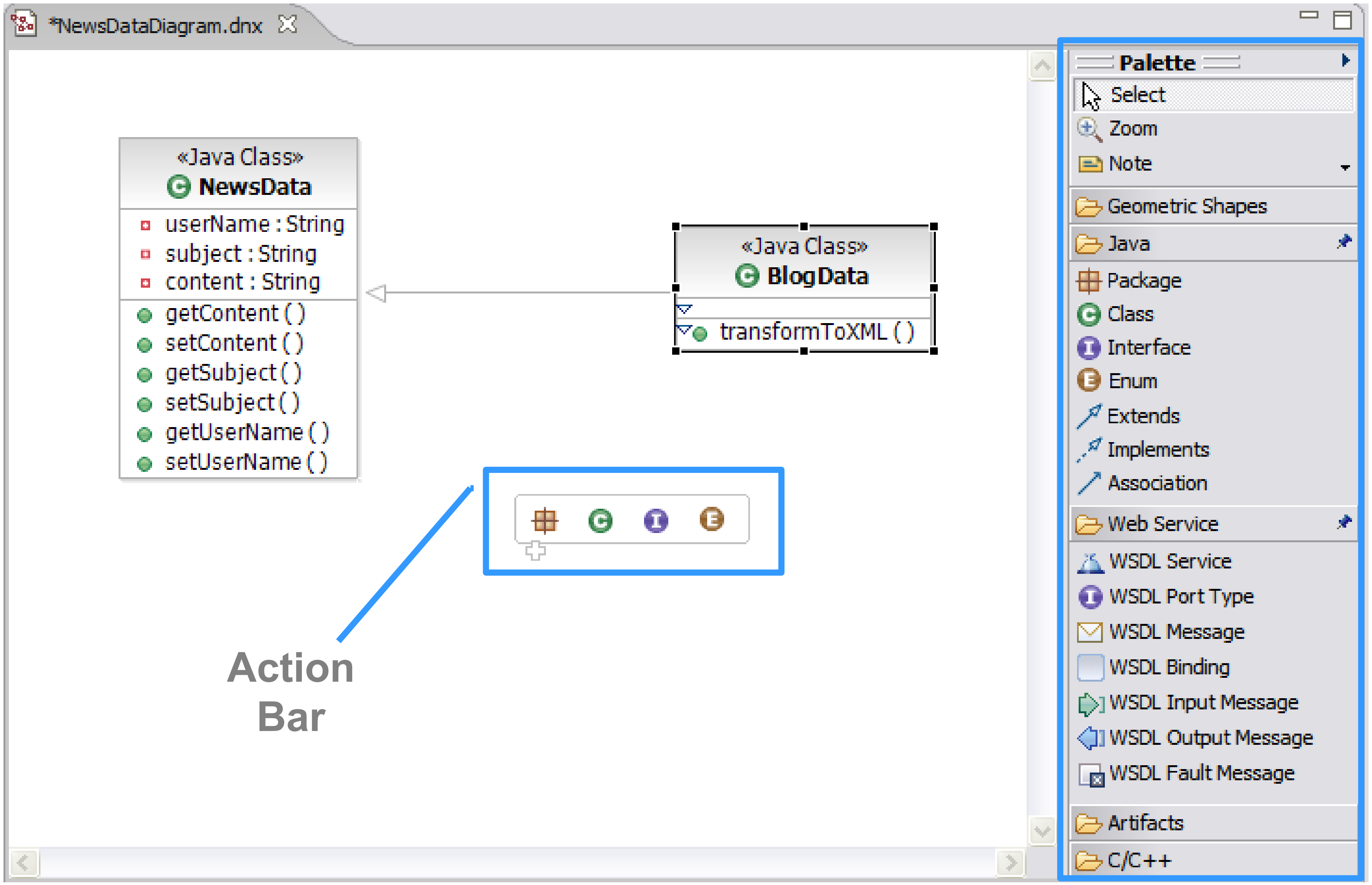
Component Diagram



Component
Modular parts of the system

Class
Included to show implementation relationships.

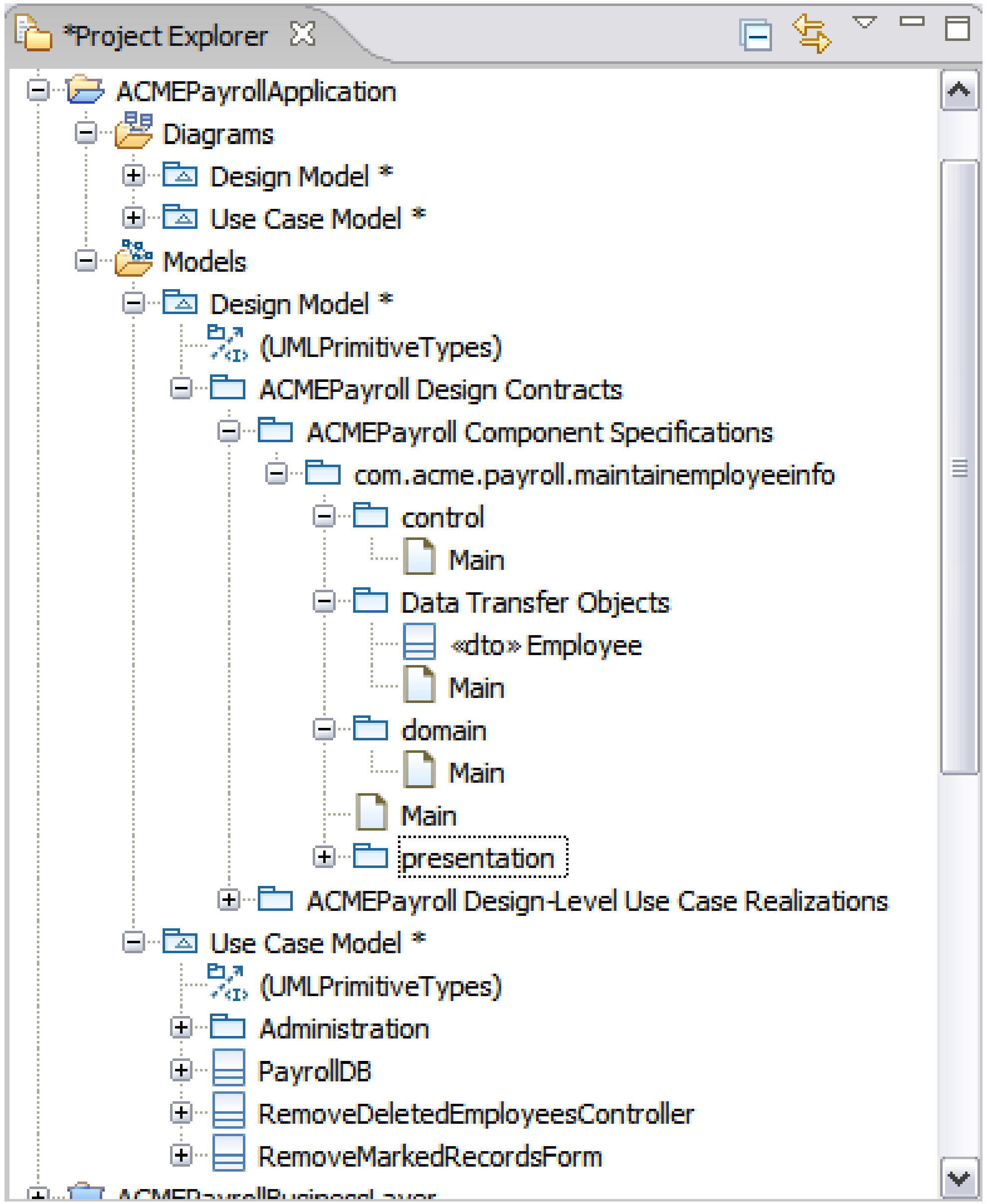
Drawing Diagrams in Software Architect



Project Explorer

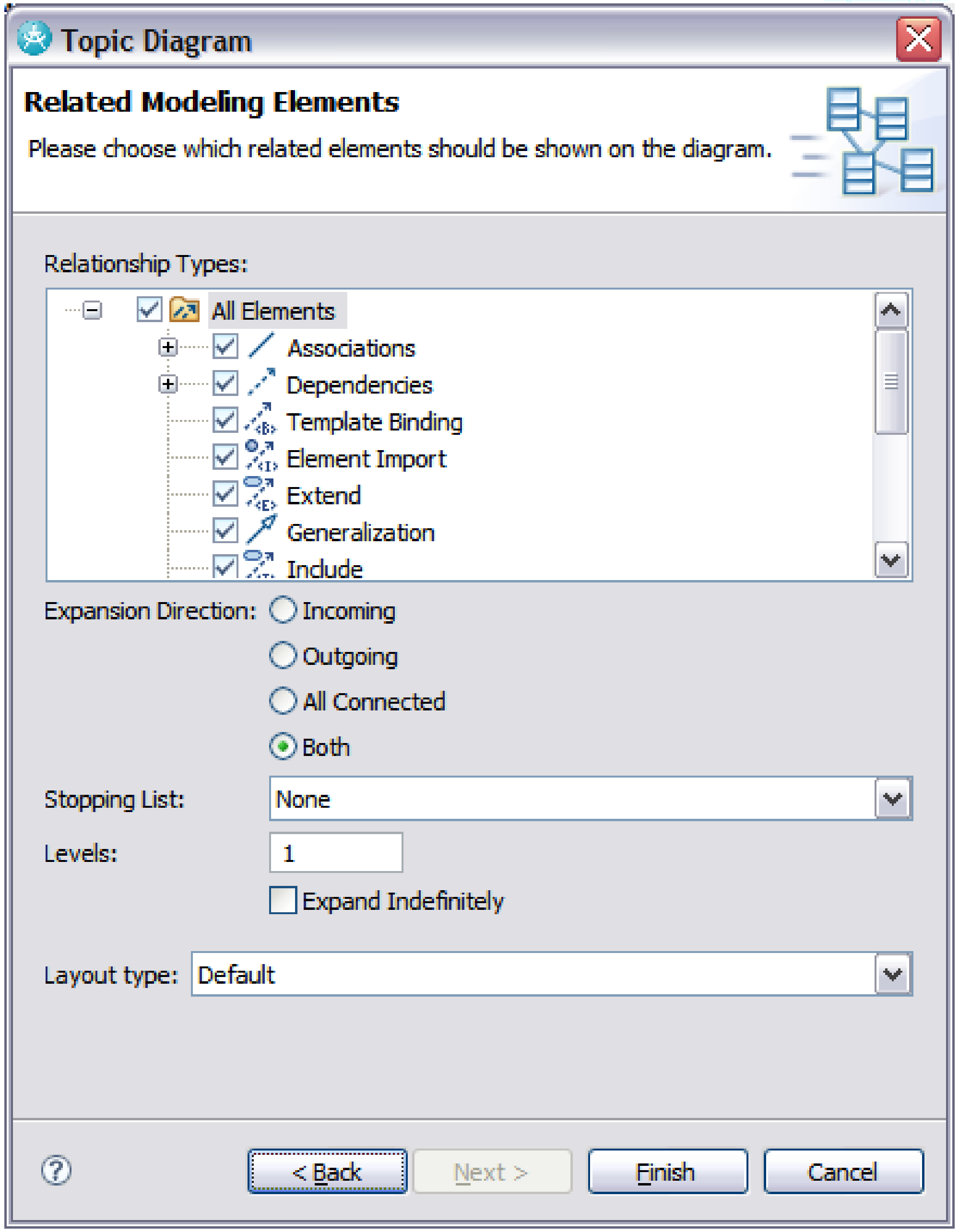
Diagrams organized by type:

- Use Case
- Design Model
- All diagrams and models are contained in the Project Explorer in their logical format.



Topic Diagrams

- Create Topic diagrams to depict key model elements and their relationships.
- Topic diagrams:
  - Are created by querying the model
  - Persist in the model
  - Are dynamically updated
  - Are most useful for visualizing code
  - Are used in architectural discovery



Topic Diagram

Related Modeling Elements

Please choose which related elements should be shown on the diagram.

Relationship Types:

- ✓ All Elements
  - ✓ Associations
  - ✓ Dependencies
  - ✓ Template Binding
  - ✓ Element Import
  - ✓ Extend
  - ✓ Generalization
  - ✓ Include

Expansion Direction: Incoming / Outgoing / All Connected / [✓ Both]

Stopping List: None

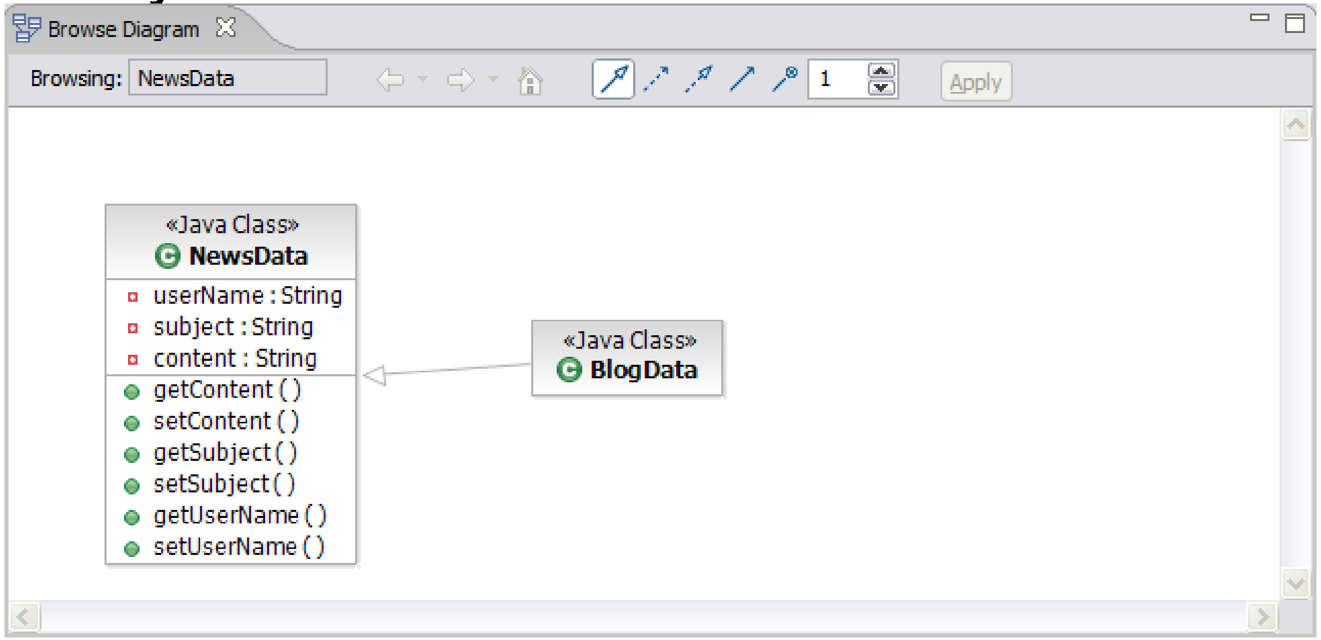
Levels: 1

☐ Expand Indefinitely

Layout type: Default

Browse Diagrams

- Browse diagrams can be used to:
  - Show the elements related to the selected element
  - Show the dependencies to the selected element
  - Gain a detailed understanding of the element in focus
- Browse diagrams are driven by parameters and filters that you control.

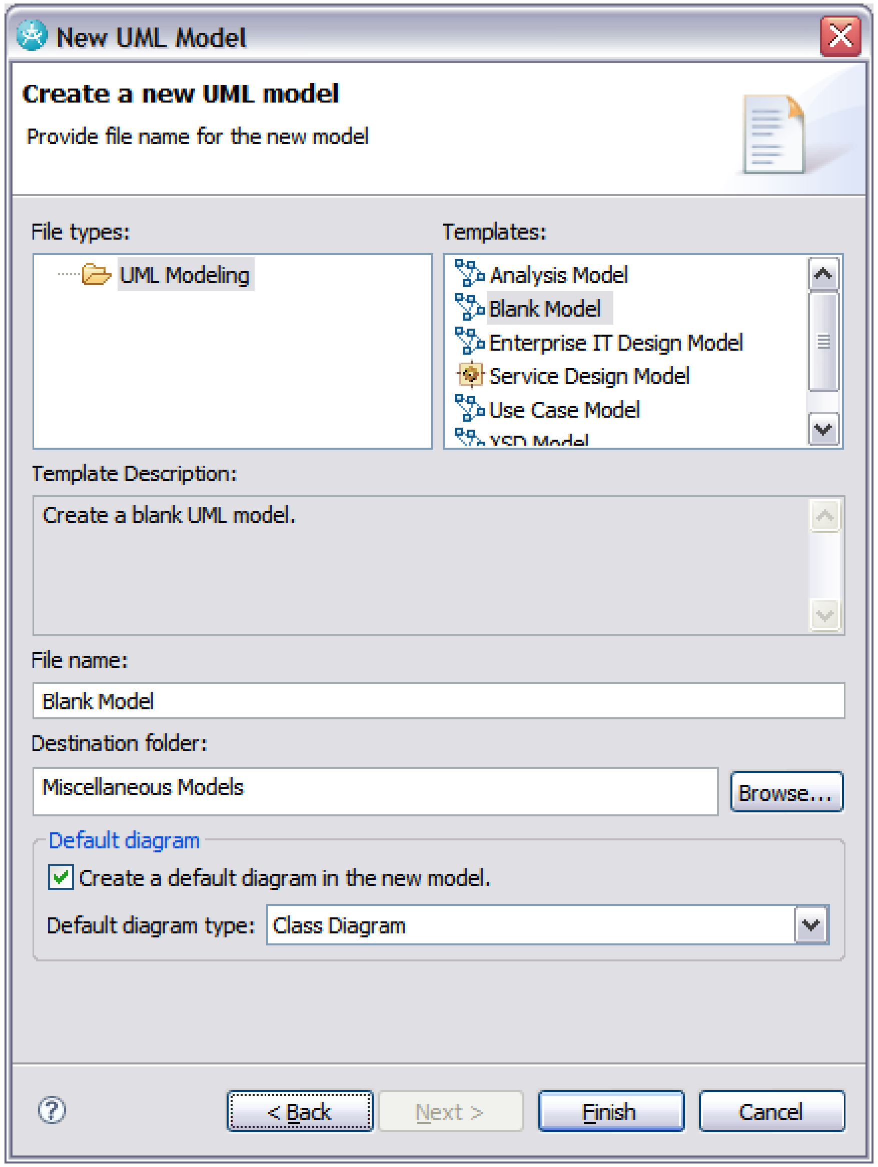


How Many Diagrams Need to be Created?

- Depends:
  - You use diagrams to visualize the system from different perspectives.
  - No complex system can be understood in its entirety from one perspective.
  - Diagrams are used for communication
- Model elements will appear on one or more diagrams.
  - For example, a class may appear on one or more class diagrams, be represented in a state machine diagram, and have instances appear on a sequence diagram.
  - Each diagram will provide a different perspective.

Model Templates in Software Architect

- Templates provide a starting point.
- Templates include:
  - Use-case model
  - Analysis model
  - Service Design Model
  - XSD Model
  - Enterprise IT design model
    - Design model specifically for n-tier business applications
  - Blank model
    - General design models
    - Freeform modeling


New UML Model

Create a new UML model

Provide file name for the new model

File types: UML Modeling

Templates:
- Analysis Model
- Blank Model
- Enterprise IT Design Model
- Service Design Model
- Use Case Model
- XSD Model

Template Description: Create a blank UML model.

File name: Blank Model

Destination folder: Miscellaneous Models

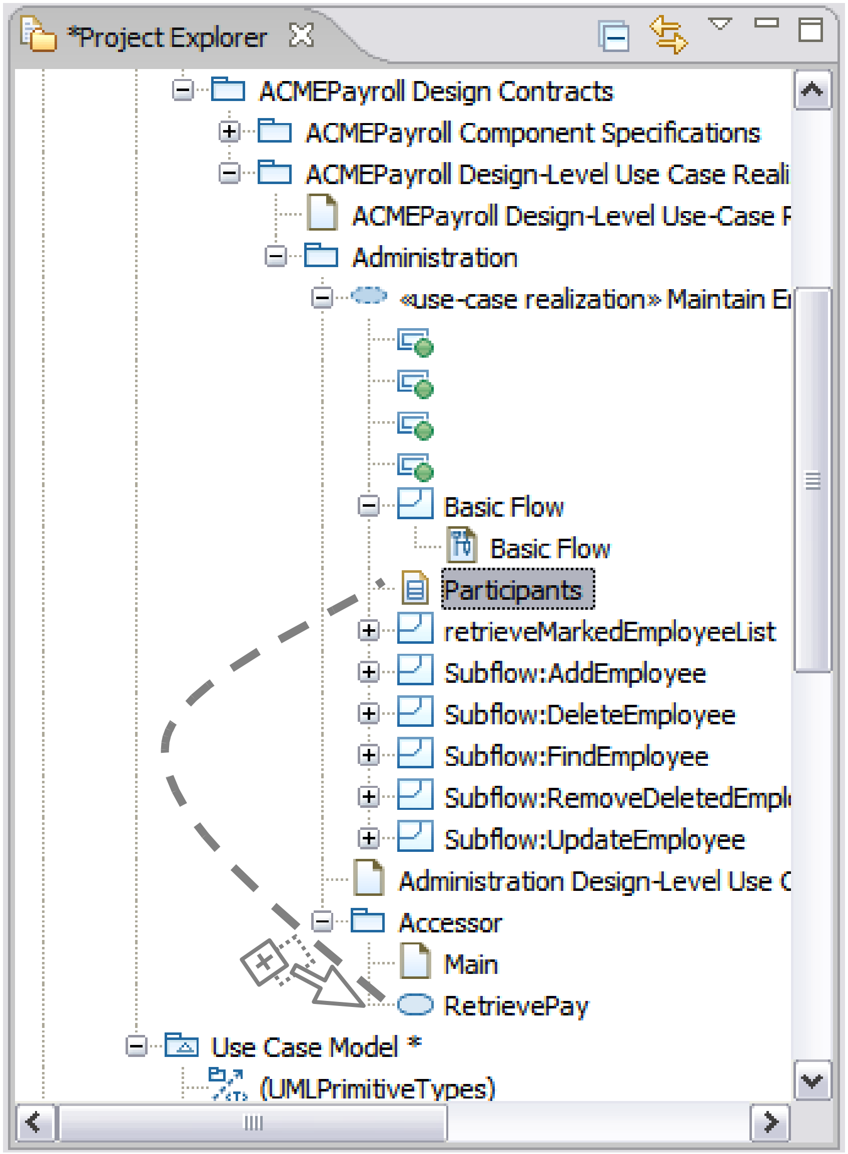
Default diagram

[✓] Create a default diagram in the new model.

Default diagram type: Class Diagram

Creating a Model Using a Model Template

- Create a UML project and select a template.
- Populate the model with:
  - UML packages
  - UML elements based on model building blocks provided


*Project Explorer

ACMEPayroll Design Contracts
ACMEPayroll Component Specifications
ACMEPayroll Design-Level Use Case Reali
ACMEPayroll Design-Level Use-Case F
Administration
«use-case realization» Maintain E
Basic Flow
Basic Flow
Participants
retrieveMarkedEmployeeList
Subflow:AddEmployee
Subflow:DeleteEmployee
Subflow:FindEmployee
Subflow:RemoveDeletedEmpl
Subflow:UpdateEmployee
Administration Design-Level Use C
Accessor
Main
RetrievePay
Use Case Model *
(UMLPrimitiveTypes)

IBM

Model Templates: Use-Case Model

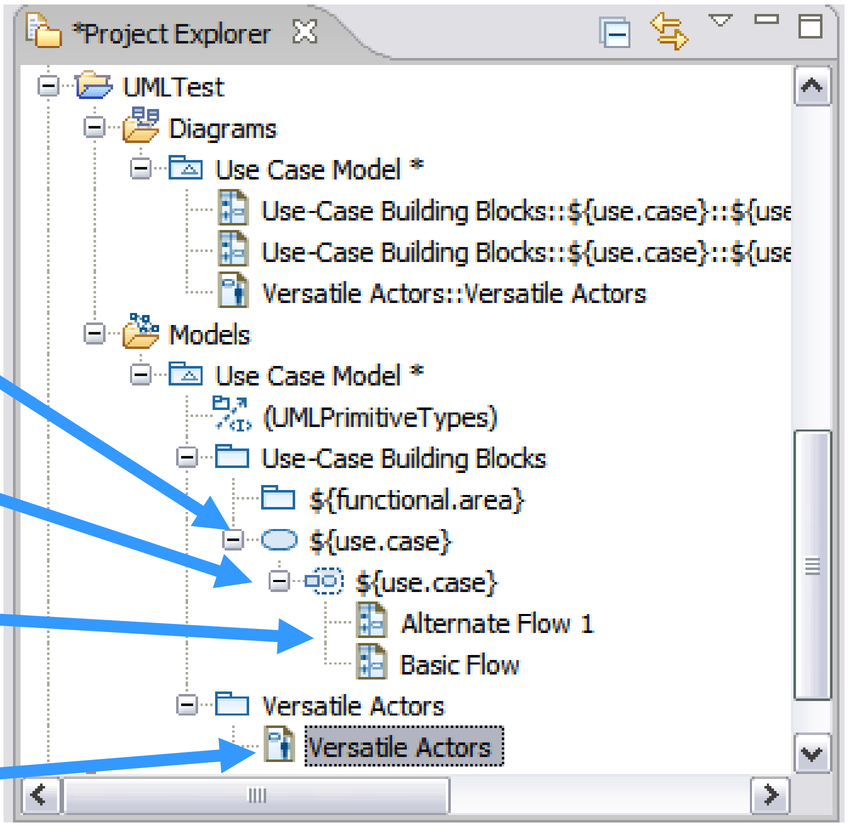
A Use-case model is a model of the system's intended functions and its environment.

Use Case
Unit of externally visible functionality

Activity
For gathering activity diagrams for the use case (optional)

Interactions
For modeling each flow of events in the use case using interaction diagrams

Versatile Actors
Actors that participate in multiple functional areas of the application


*Project Explorer

```
UMLTest
  Diagrams
    Use Case Model *
      Use-Case Building Blocks::${use.case}::${use
      Use-Case Building Blocks::${use.case}::${use
      Versatile Actors::Versatile Actors
  Models
    Use Case Model *
      (UMLPrimitiveTypes)
      Use-Case Building Blocks
        ${functional.area}
        ${use.case}
          ${use.case}
            Alternate Flow 1
            Basic Flow
      Versatile Actors
        Versatile Actors
```

Model Templates: Analysis Model

An analysis model describes the realization of use cases to analysis classes.

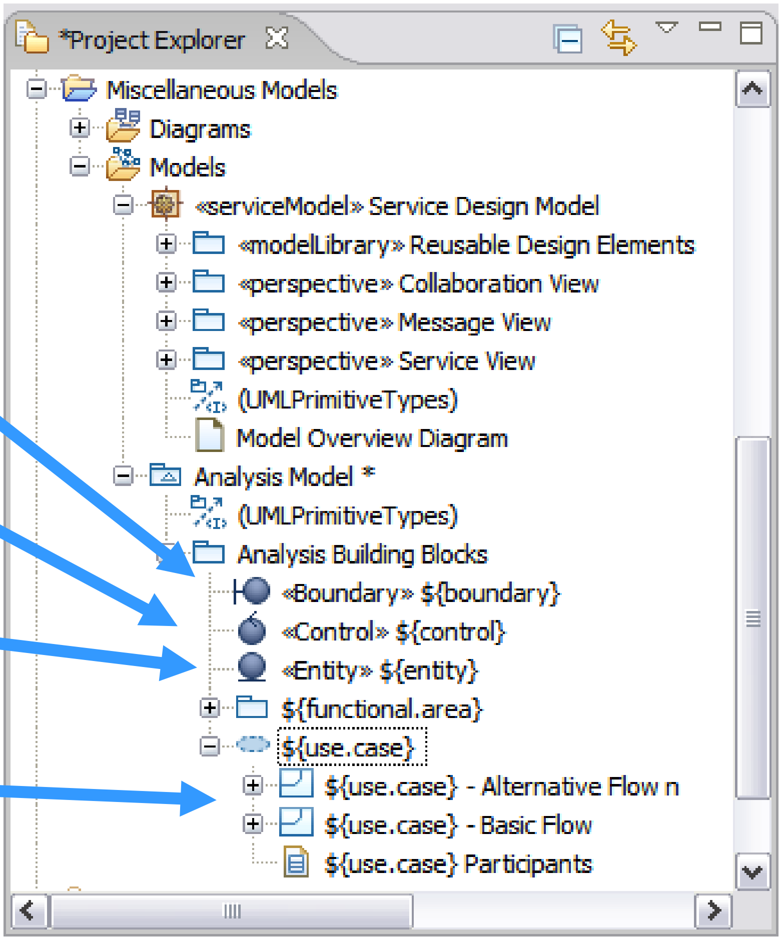
- Provides an abstraction of the design model

Boundary Class
Mediates between the system and something outside the system

Controller Class
Provides the coordinating behavior in the system

Entity Class
Reused in many use cases, often with persistent characteristics

Use-Case Realization
Shows how a use case are implemented in terms of collaborating objects



```
*Project Explorer
⊟ Miscellaneous Models
    ⊞ Diagrams
    ⊟ Models
        ⊟ «serviceModel» Service Design Model
            ⊞ «modelLibrary» Reusable Design Elements
            ⊞ «perspective» Collaboration View
            ⊞ «perspective» Message View
            ⊞ «perspective» Service View
              (UMLPrimitiveTypes)
              Model Overview Diagram
        ⊟ Analysis Model *
              (UMLPrimitiveTypes)
              Analysis Building Blocks
                  «Boundary» ${boundary}
                  «Control» ${control}
                  «Entity» ${entity}
                ⊞ ${functional.area}
                ⊟ ${use.case}
                    ⊞ ${use.case} - Alternative Flow n
                    ⊞ ${use.case} - Basic Flow
                      ${use.case} Participants
```

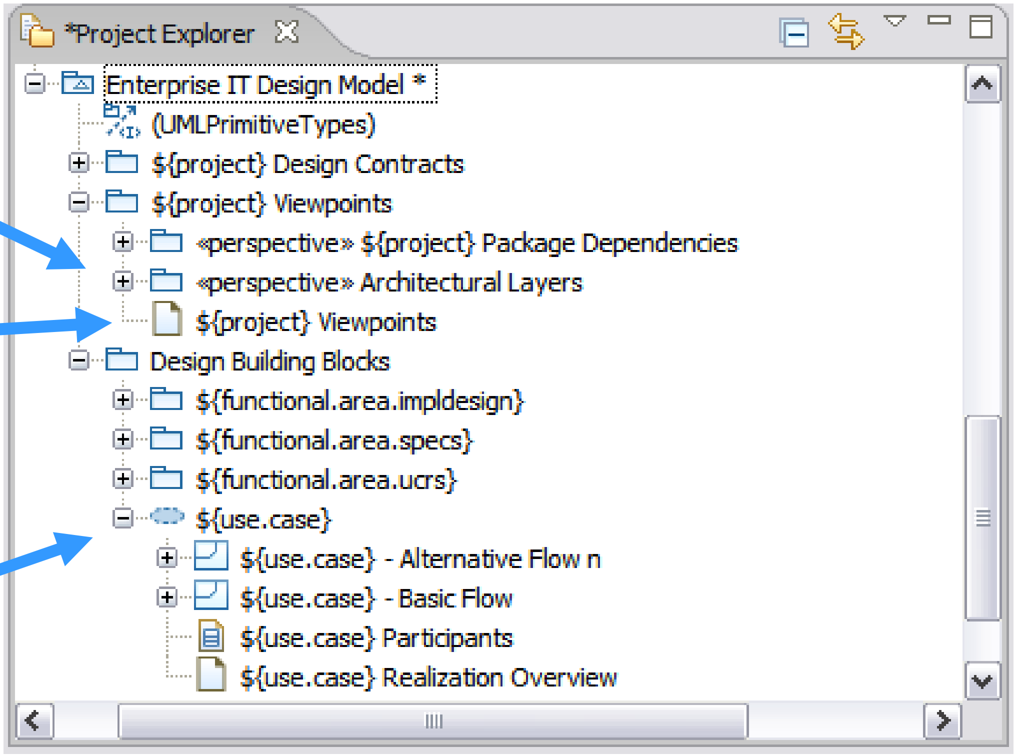
Model Templates: Enterprise IT Design Model

A design model describes the realization of use cases to design classes.

Architectural Layers
Separate business logic from data and user interface

Viewpoints
Contains diagrams presenting architecturally significant, cross-cutting views of the model

Use-Case Realization
Shows how a use case is implemented in terms of collaborating objects



```
Enterprise IT Design Model *
    (UMLPrimitiveTypes)
    ${project} Design Contracts
    ${project} Viewpoints
        «perspective» ${project} Package Dependencies
        «perspective» Architectural Layers
        ${project} Viewpoints
    Design Building Blocks
        ${functional.area.impldesign}
        ${functional.area.specs}
        ${functional.area.ucrs}
        ${use.case}
            ${use.case} - Alternative Flow n
            ${use.case} - Basic Flow
            ${use.case} Participants
            ${use.case} Realization Overview
```

Lab: UML Diagrams