Software Engineering

Lesson #08 - Practice

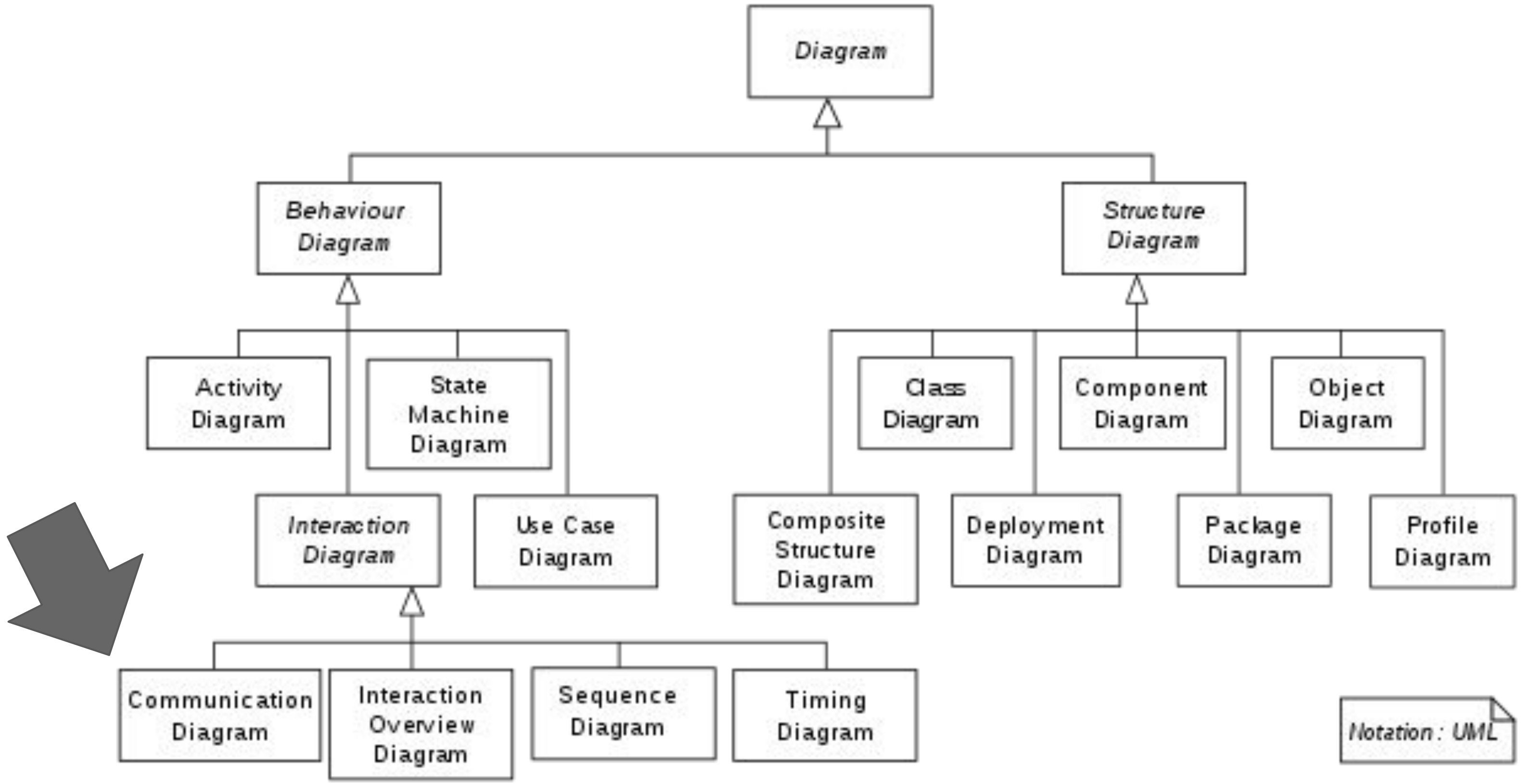
Lesson #08 - Practice

Agenda: Lesson #08 - Software Engineering - Practice

Agenda: Lesson #08 - Software Engineering - Practice

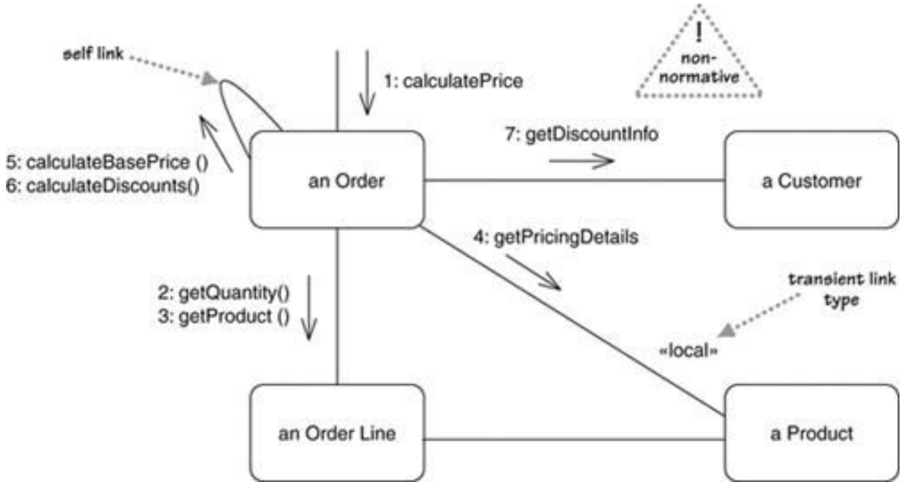
1	**Communication Diagrams**
2	Composite Structure Diagrams
3	Class Work
4	Q & A

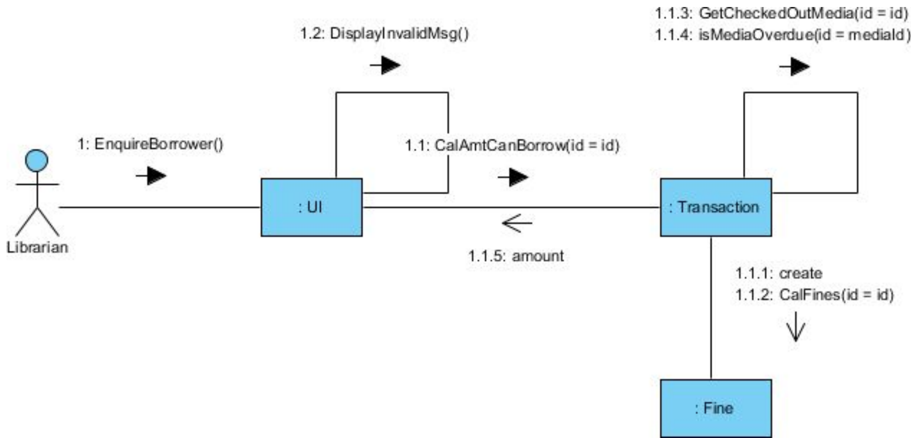
Communication Diagrams



Communication Diagrams

Communication diagrams, a kind of interaction diagram, emphasize the data links between the various participants in the interaction



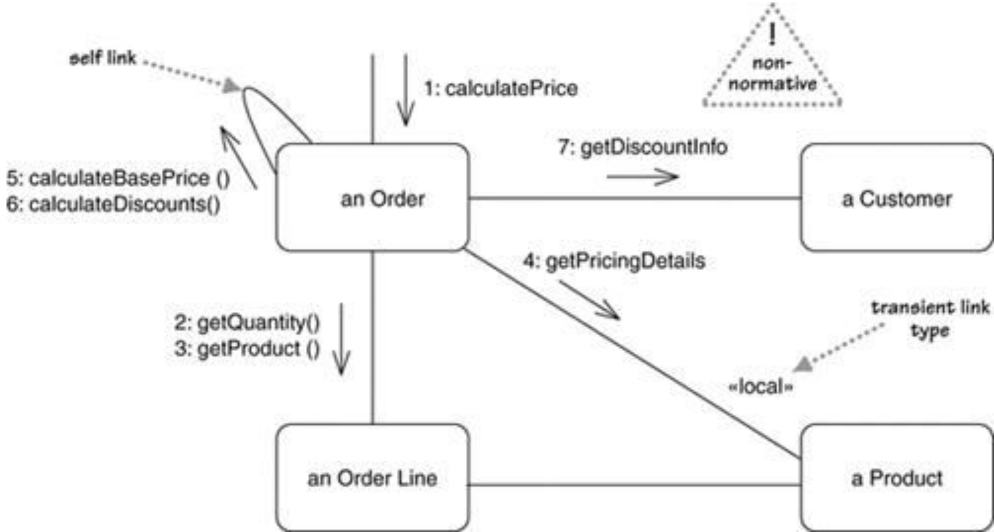


Communication Diagrams

Instead of drawing each participant as a lifeline and showing the sequence of messages by vertical direction as the sequence diagrams does, the communication diagram allows free placement of participants, allows you to draw links to show how the participants connect, and use numbering to show the sequence of messages

Communication Diagrams

Sample of communication diagram for centralized control
we can show how the participants are linked together



Communication Diagrams

As well as showing links that are instances of associations, we can also show transient links, which arise only the context of the interaction

In this case, the «local» link from Order to Product is a local variable; other transient links are «parameter» and «global»

Communication Diagrams

A communication diagram in the Unified Modeling Language (UML) 2.0, is a simplified version of the UML 1.x collaboration diagram

Communication Diagrams

UML has four types of interaction diagrams:

- Sequence diagram
- Communication diagram
- Interaction overview diagram
- Timing diagram

Communication Diagrams

Behavioural UML diagrams

- Activity diagram
- Communication diagram
- Interaction overview diagram
- Sequence diagram
- State diagram
- Timing diagram
- Use case diagram

Communication Diagrams

A Communication diagram models the interactions between objects or parts in terms of sequenced messages

Communication diagrams represent a combination of information taken from Class, Sequence, and Use Case Diagrams describing both the static structure and dynamic behavior of a system

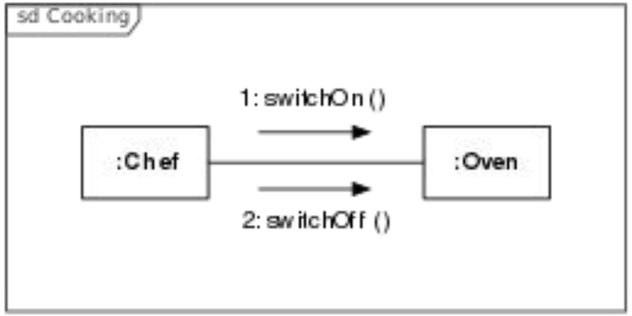
Communication Diagrams

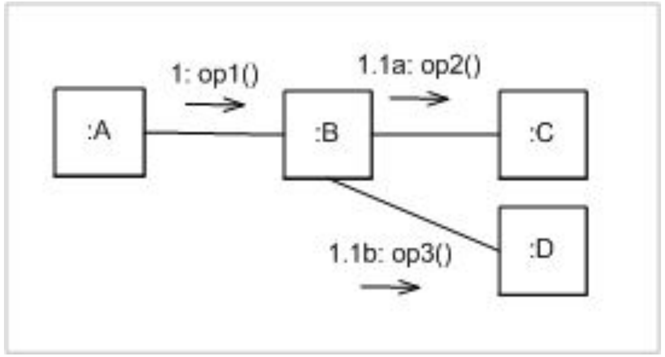
Communication diagrams show much of the same information as sequence diagrams, but because of how the information is presented, some of it is easier to find in one diagram than the other

Communication diagrams show which elements each one interacts with better, but sequence diagrams show the order in which the interactions take place more clearly

Communication Diagrams

Real life system design sample with the Communication diagram of UML -> Generic sample of a Communication diagram





When to use communication diagrams

The main question with communication diagrams is when to use them rather than the more common sequence diagrams

A strong part of the decision is personal preference: Some people like one over the other

Often, that drives the choice more than anything else. On the whole, most people seem to prefer sequence diagrams, and for once, I'm with the majority

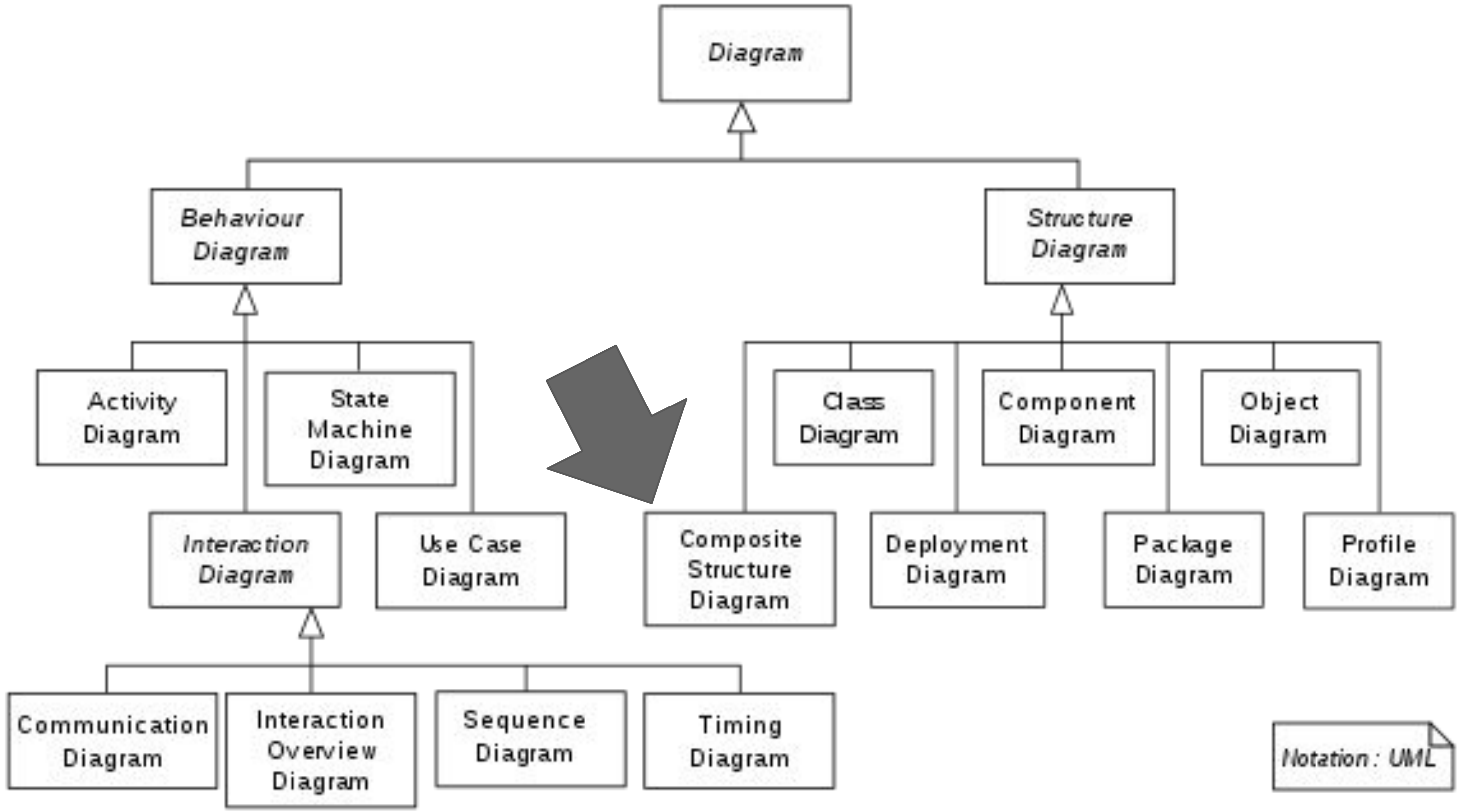
When to use communication diagrams

A more rational approach says that sequence diagrams are better when you want to emphasize the sequence of calls and that communication diagrams are better when you want to emphasize the links

Many people find that communication diagrams are easier to alter on a whiteboard, so they are a good approach for exploring alternatives, although in those cases, I often prefer CRC cards

Agenda: Lesson #08 - Software Engineering - Practice

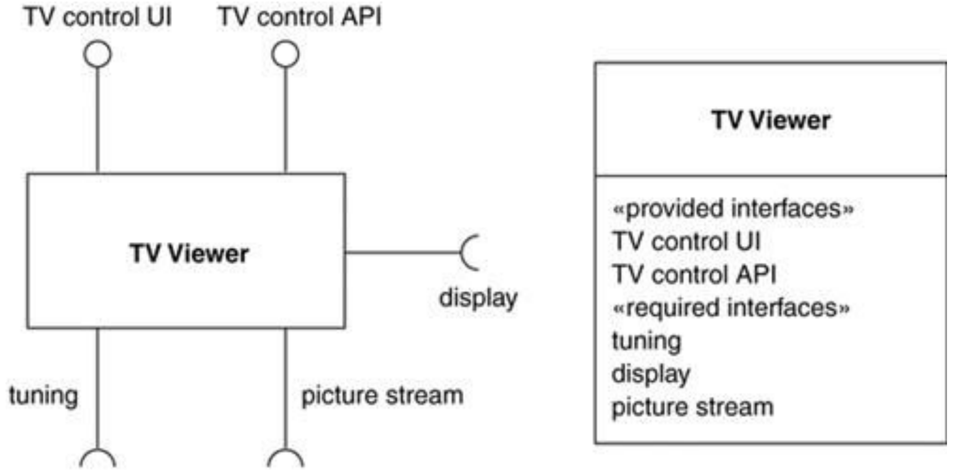
Composite Structures



Composite Structures

One of the most significant new features in UML 2 is the ability to hierarchically decompose a class into an internal structure

This allows you to take a complex object and break it down into parts



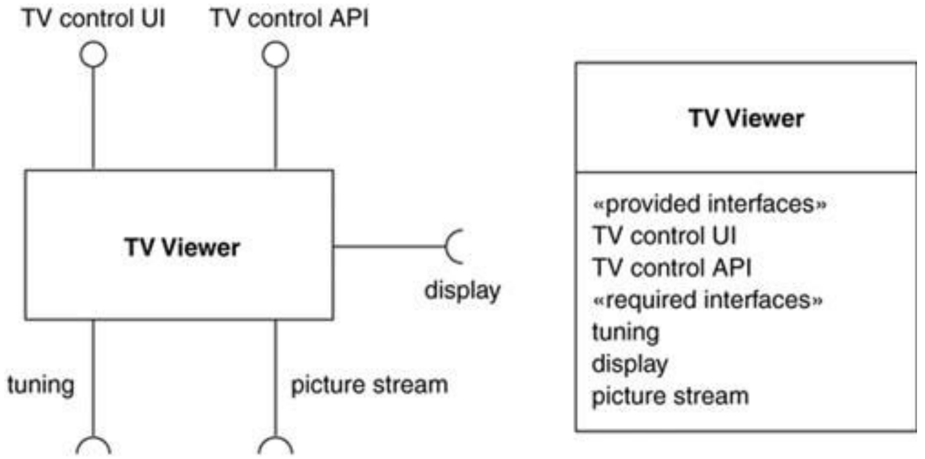
Composite Structures

Structural UML diagrams

- Class diagram
- Component diagram
- Composite structure diagram
- Deployment diagram
- Object diagram
- Package diagram
- Profile diagram

Composite Structures

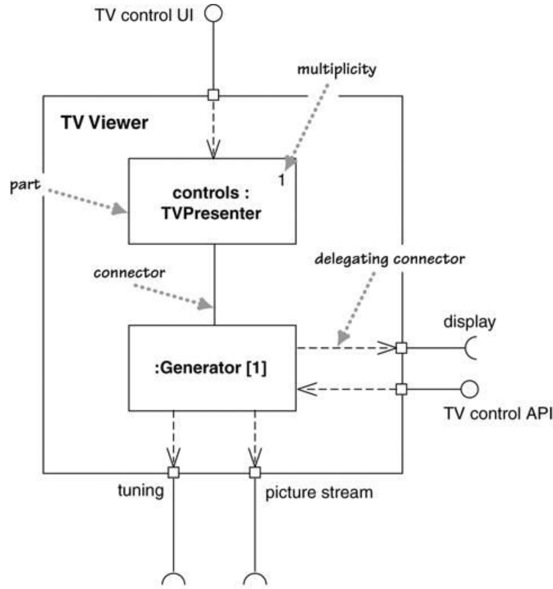
The picture shows a TV Viewer class with its provided and required interfaces I've shown this in two ways: using the ball-and-socket notation and listing them internally



Composite Structures

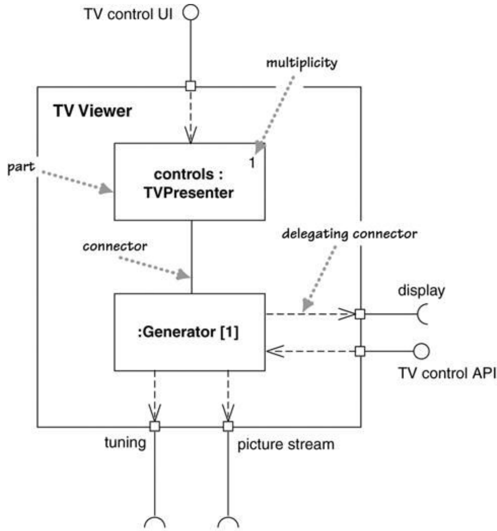
The picture shows how this class is decomposed internally into two parts and which parts support and require the different interfaces

Each part is named in the form name : class, with both elements individually optional
Parts are not instance specifications, so they are bolded rather than underlined



Composite Structures

You can show how many instances of a part are present
The picture says that each TV Viewer contains one generator part and one controls part



Composite Structures

To show a part implementing an interface, you draw a delegating connector from that interface

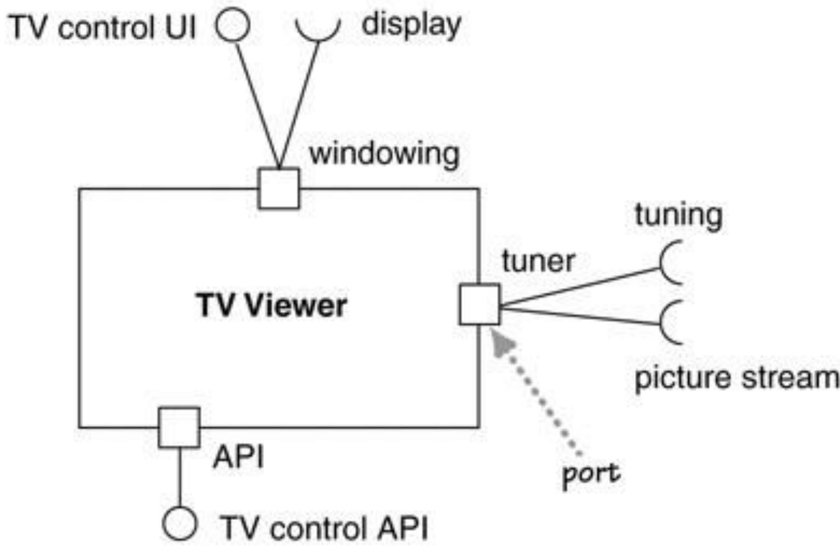
Similarly, to show that a part needs an interface, you show a delegating connector to that interface

You can also show connectors between parts with either a simple line, as I've done here, or with ball-and-socket notation

Composite Structures

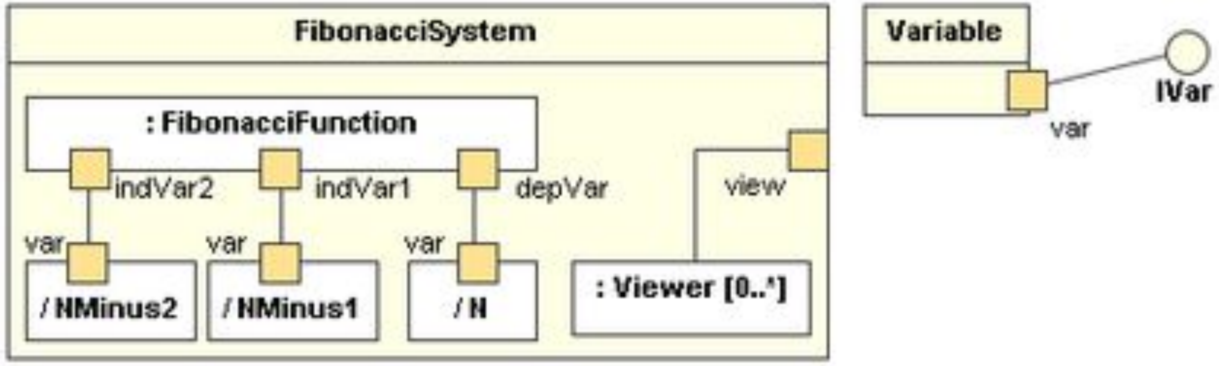
You can add ports to the external structure

Ports allow you to group the
required and provided interfaces
into logical interactions
that a component has with
the outside world



Composite Structures

Composite structure diagram in the Unified Modeling Language (UML) is a type of static structure diagram, that shows the internal structure of a class and the collaborations that this structure makes possible



Composite Structures

This diagram can include internal parts, ports through which the parts interact with each other or through which instances of the class interact with the parts and with the outside world, and connectors between parts or ports

A composite structure is a set of interconnected elements that collaborate at runtime to achieve some purpose

Each element has some defined role in the collaboration

When to Use Composite Structures

Composite structures are new to UML 2, although some older methods had some similar ideas

A good way of thinking about the difference between packages and composite structures is that packages are a compile-time grouping, while composite structures show runtime groupings

As such, they are a natural fit for showing components and how they are broken into parts; hence, much of this notation is used in component diagrams

When to Use Composite Structures

Because composite structures are new to the UML, it's too early to tell how effective they will turn out in practice; many members of the UML committee think that these diagrams will become a very valuable addition

Agenda: Lesson #08 - Software Engineering - Practice

Class work

Software Engineering, 10. Global Edition

No classwork for today

Agenda: Lesson #08 - Software Engineering - Practice

1 Communication Diagrams

2 Composite Structures

3 Class Work

4 **Q & A**

Agenda: Lesson #08 - Software Engineering - Practice

Q & A