Software Engineering

Lesson #09 - Practice

Lesson #09 - Practice

Agenda: Lesson #09 - Software Engineering - Practice

1. Component Diagrams
2. Collaboration
3. Class Work
4. Q & A

Agenda: Lesson #09 - Software Engineering - Practice

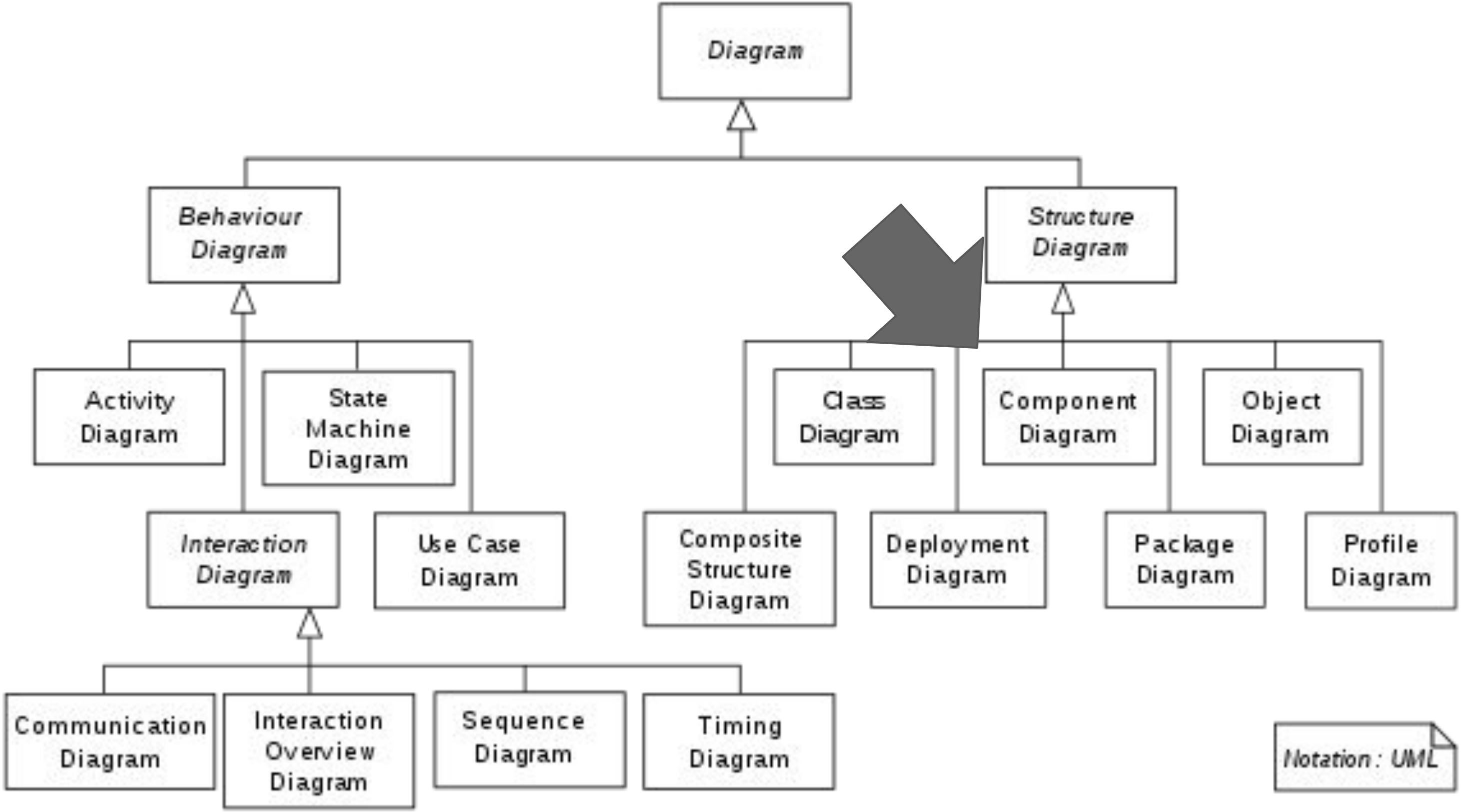
1 **Component Diagrams**

2 Collaboration

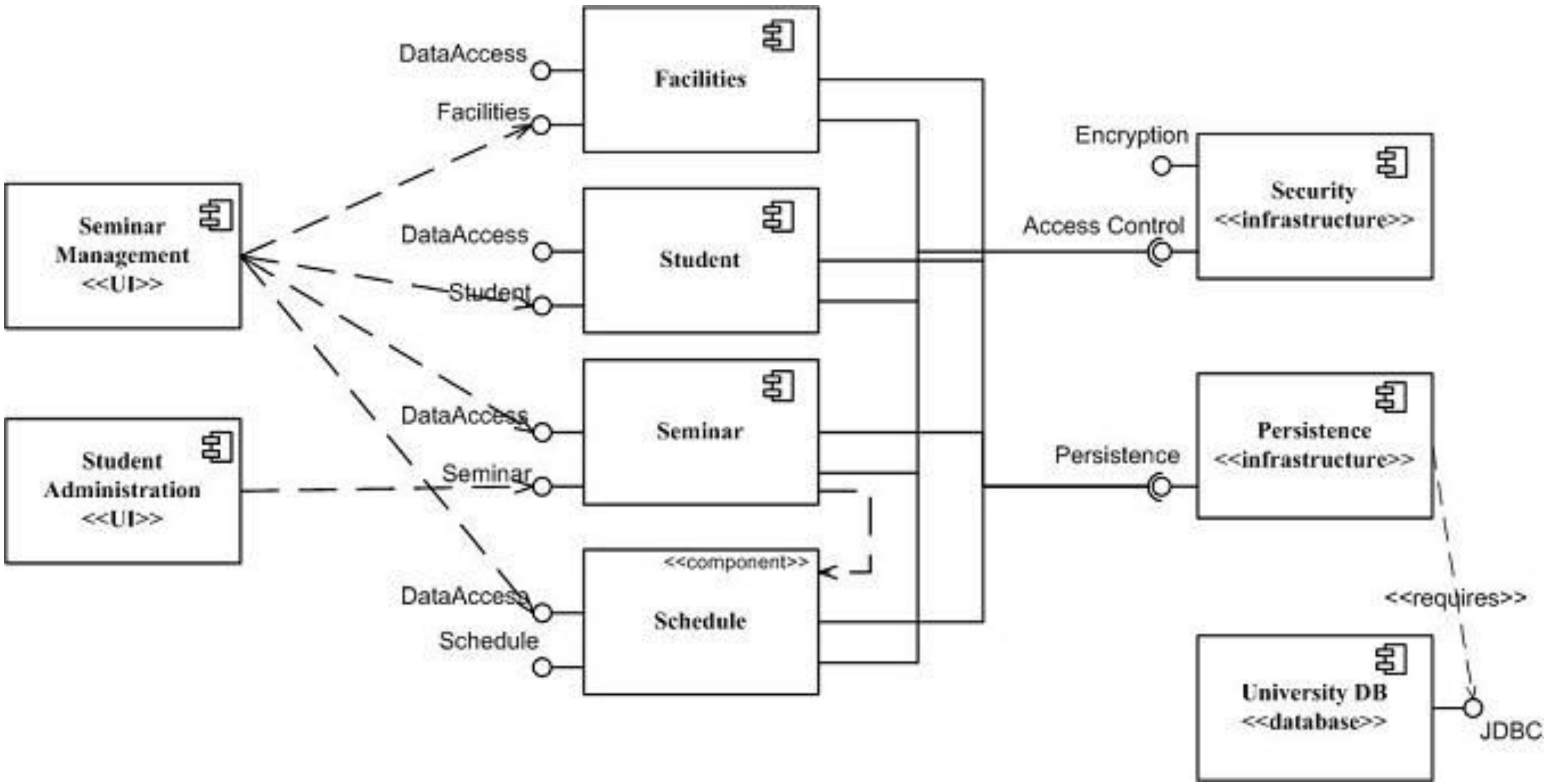
3 Class Work

4 Q & A

Component Diagrams



Component Diagrams

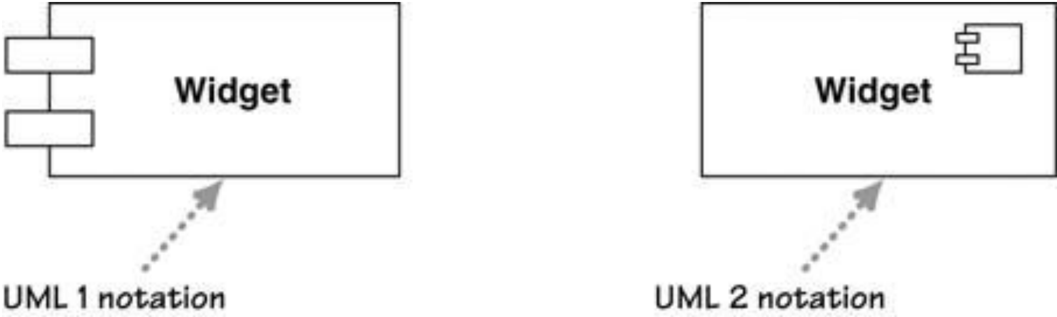


Component Diagrams

UML 1 had a distinctive symbol for a component as shown in picture below

UML 2 removed that icon but allows you to annotate a class box with a similar-looking icon

Alternatively, you can use the «component» keyword

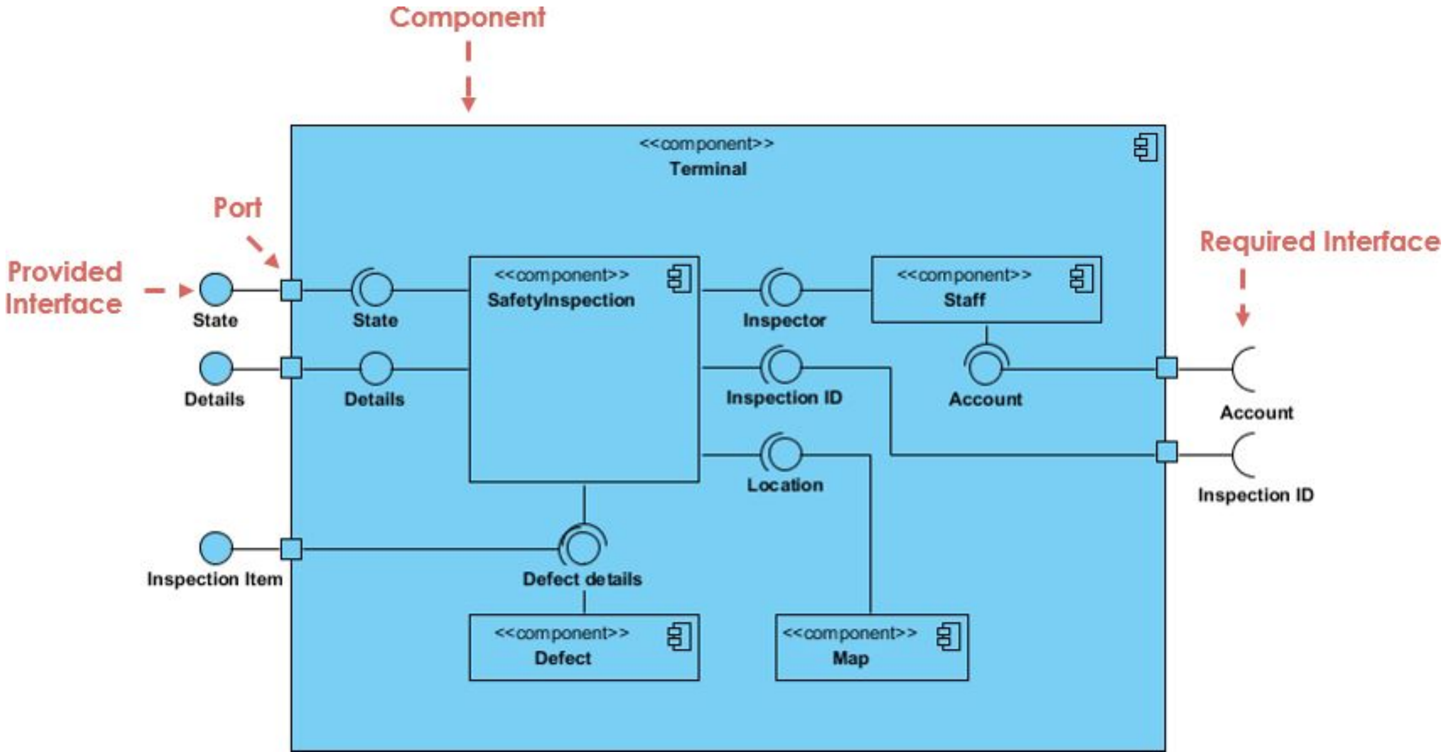


Component Diagrams

In Unified Modeling Language (UML), a component diagram depicts how components are wired together to form larger components or software systems

They are used to illustrate the structure of arbitrarily complex systems

Component Diagrams



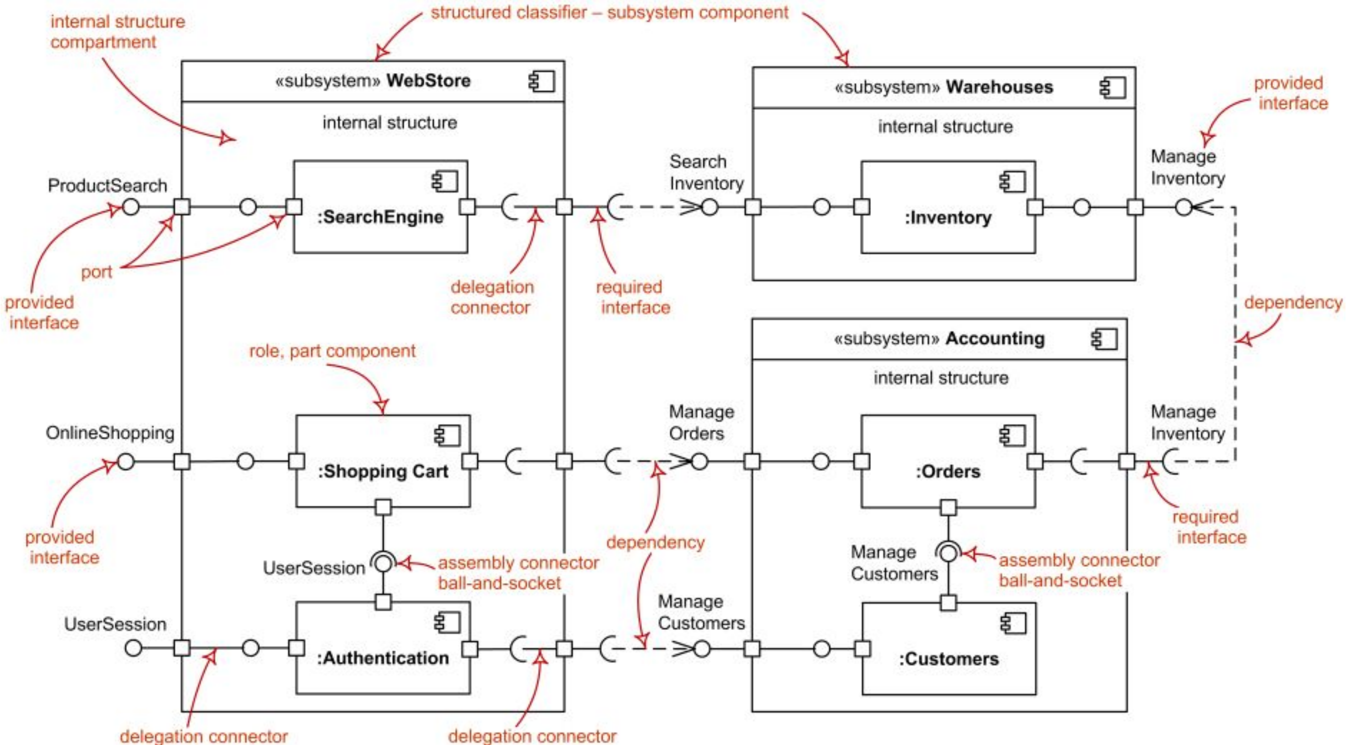
Component

Port

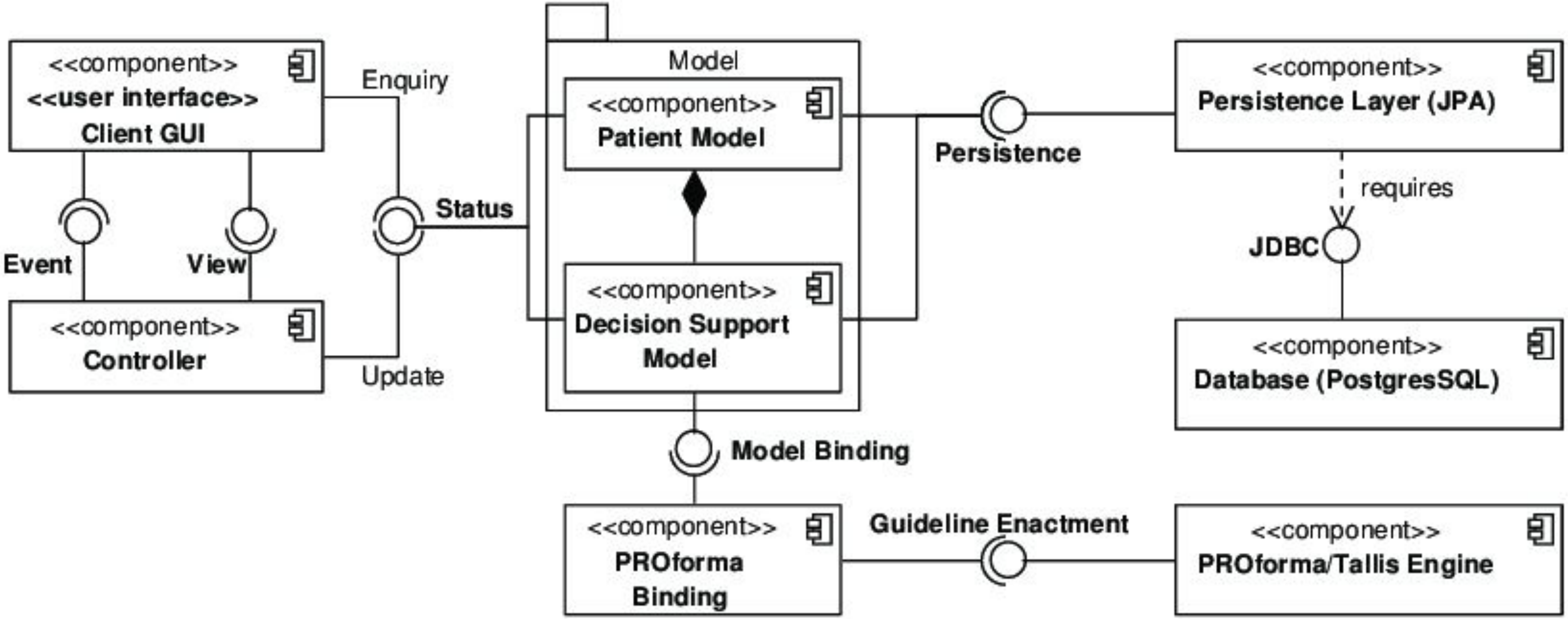
Provided Interface

Required Interface

Component Diagrams



Component Diagrams



Component Diagrams

A component diagram allows verification that a system's required functionality is acceptable. These diagrams are also used as a communication tool between the developer and stakeholders of the system.

Programmers and developers use the diagrams to formalize a roadmap for the implementation, allowing for better decision-making about task assignment or needed skill improvements

Component Diagrams

System administrators can use component diagrams to plan ahead, using the view of the logical software components and their relationships on the system

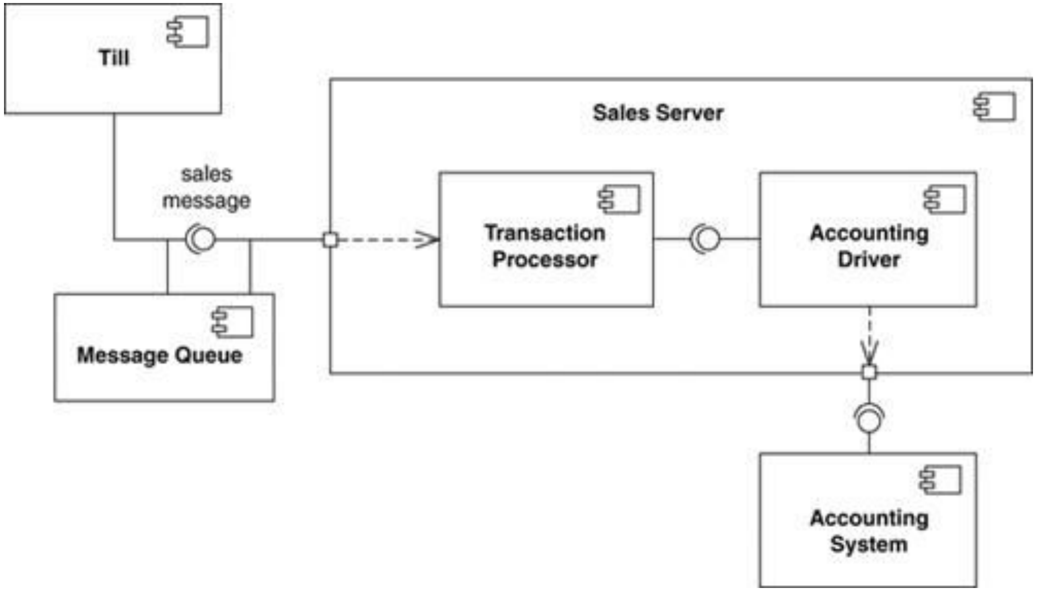
Component Diagrams

Structural UML diagrams

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

Component Diagrams

Figure shows an example component diagram. In this example, a sales till can connect to a sales server component, using a sales message interface



When to Use Component Diagrams

Use component diagrams when you are dividing your system into components and want to show their interrelationships through interfaces or the breakdown of components into a lower-level structure

Agenda: Lesson #09 - Software Engineering - Practice

1 Component Diagrams

2 Collaboration

3 Class Work

4 Q & A

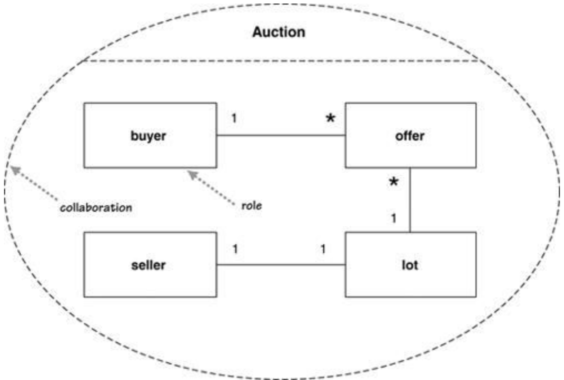
Collaboration

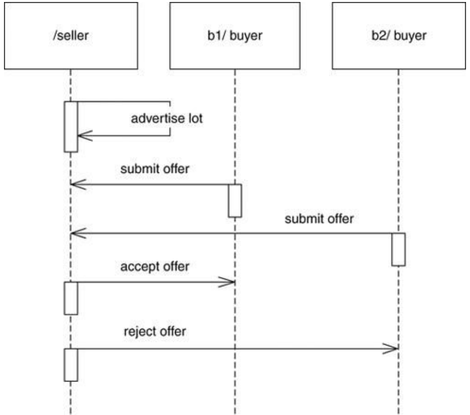
Unlike the other chapters in this book, this one does not correspond to an official diagram in UML 2.
The standard discusses collaborations as part of composite structures, but the diagram is really quite different and was used in UML 1 without any link to composite structures.

So we felt it best to discuss collaborations as their own chapter.

Collaboration

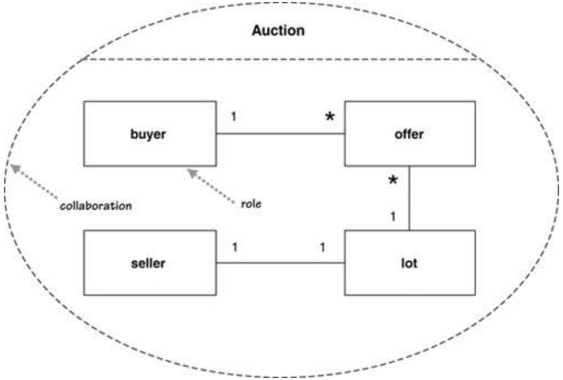
Let's consider the notion of an auction. In any auction, we might have a seller, some buyers, a lot of goods, and some offers for the sale. We can describe these elements in terms of a class diagram (Figure 1) and perhaps some interaction diagrams (Figure 2).





Collaboration

Figure 1 is not quite a regular class diagram. For a start, it's surrounded by the dashed ellipse, which represents the auction collaboration. Second, the so-called classes in the collaboration are not classes but roles that will be realized as the collaboration is applied - hence the fact that their names aren't capitalized. It's not unusual to see actual interfaces or classes that correspond to the collaboration roles, but you don't have to have them.

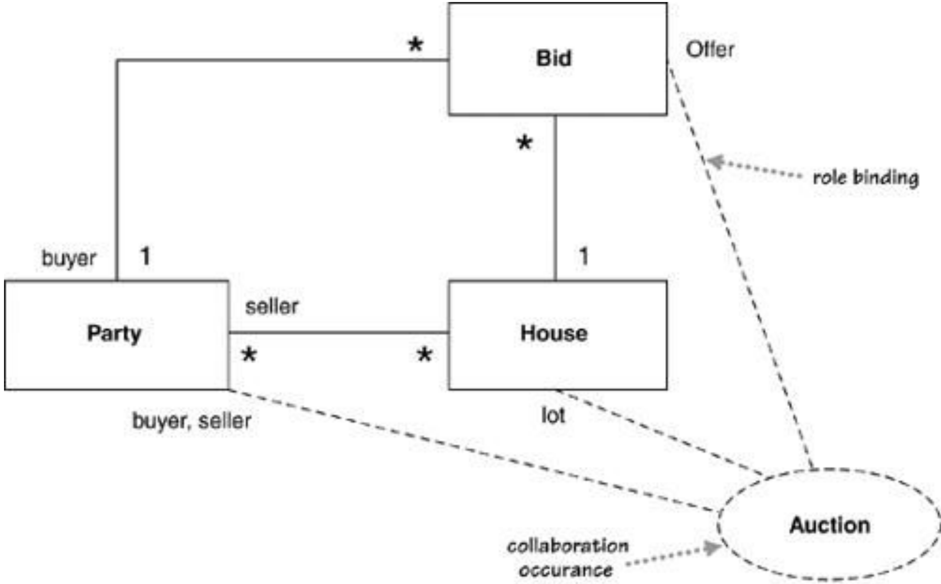


Collaboration

In the interaction diagram, the participants are labeled slightly differently from the usual case. In a collaboration, the naming scheme is participant-name /role-name : class-name. As usual, all these elements are optional.

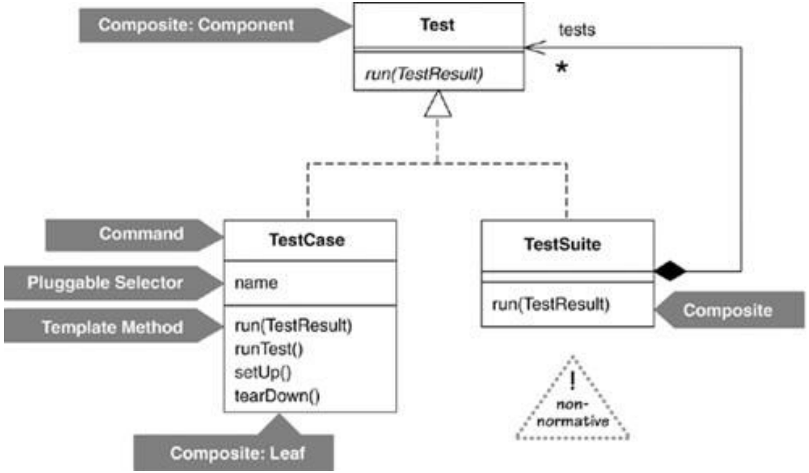
Collaboration

When you use a collaboration, you can show that by placing a collaboration occurrence on a class diagram, as in Figure 3, a class diagram of some of the classes in the application. The links from the collaboration to those classes indicate how the classes play the various roles defined in the collaboration



Collaboration

The UML suggests that you can use the collaboration occurrence notation to show the use of patterns, but hardly any patterns author has done this. Erich Gamma developed a nice alternative notation (Figure 4). Elements of the diagram are labeled with either the pattern name or a combination of pattern:role.



When to Use Collaborations

Collaborations have been around since UML 1, but I admit I've hardly used them, even in my patterns writing. Collaborations do provide a way to group chunks of interaction behavior when roles are played by different classes. In practice, however, I've not found that they've been a compelling diagram type.

Agenda: Lesson #09 - Software Engineering - Practice

1 Component Diagrams

2 Collaboration

3 Class Work

4 Q & A

Class work

Software Engineering, 10. Global Edition

No classwork for today

Agenda: Lesson #09 - Software Engineering - Practice

1 Component Diagrams

2 Collaboration

3 Class Work

4 **Q & A**

Agenda: Lesson #09 - Software Engineering - Practice

Q & A