References

Wirfs-Brock, Designing Object-Oriented Software, chapters 1-5
Relations

Mr. White

employer

All Smart

Mr. White

employee

All Smart
If need both directions

Two Pointers

Mr. White

employer

All Smart

employee
If need both directions

Mr. White

works-for

All Smart
# Recording Responsibilities

<table>
<thead>
<tr>
<th>Class: Drawing</th>
<th>Class: Drawing</th>
</tr>
</thead>
<tbody>
<tr>
<td>List responsibilities here</td>
<td>Know which elements it contains</td>
</tr>
<tr>
<td></td>
<td>Maintain ordering between elements</td>
</tr>
</tbody>
</table>
Collaboration

Represents requests from a client to a server in fulfillment of a client responsibility

Interaction between objects
Finding Collaborations

Examine class responsibilities for dependencies

For each responsibility:
   Is class capable of fulfilling this responsibility?
   If not, what does it need?
   From what other class can it acquire what it needs?

For each class:
   What does this class do or know?
   What other classes need the result or information?
   If class has no interactions, discard it
Finding Collaborations

Examine scenarios

Interactions in the scenarios indicate collaboration
Common Collaboration Types

The is-part-of relationship

X is composed of Y's
   Composite classes
   Drawing is composed of drawing elements

Some distribution of responsibilities required

Container classes
   Arrays, lists, sets, hash tables, etc.
   Some have no interaction with elements
## Recording Collaborations

<table>
<thead>
<tr>
<th>Class: Drawing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Know which elements it contains</td>
</tr>
<tr>
<td>Maintain ordering between elements</td>
</tr>
</tbody>
</table>