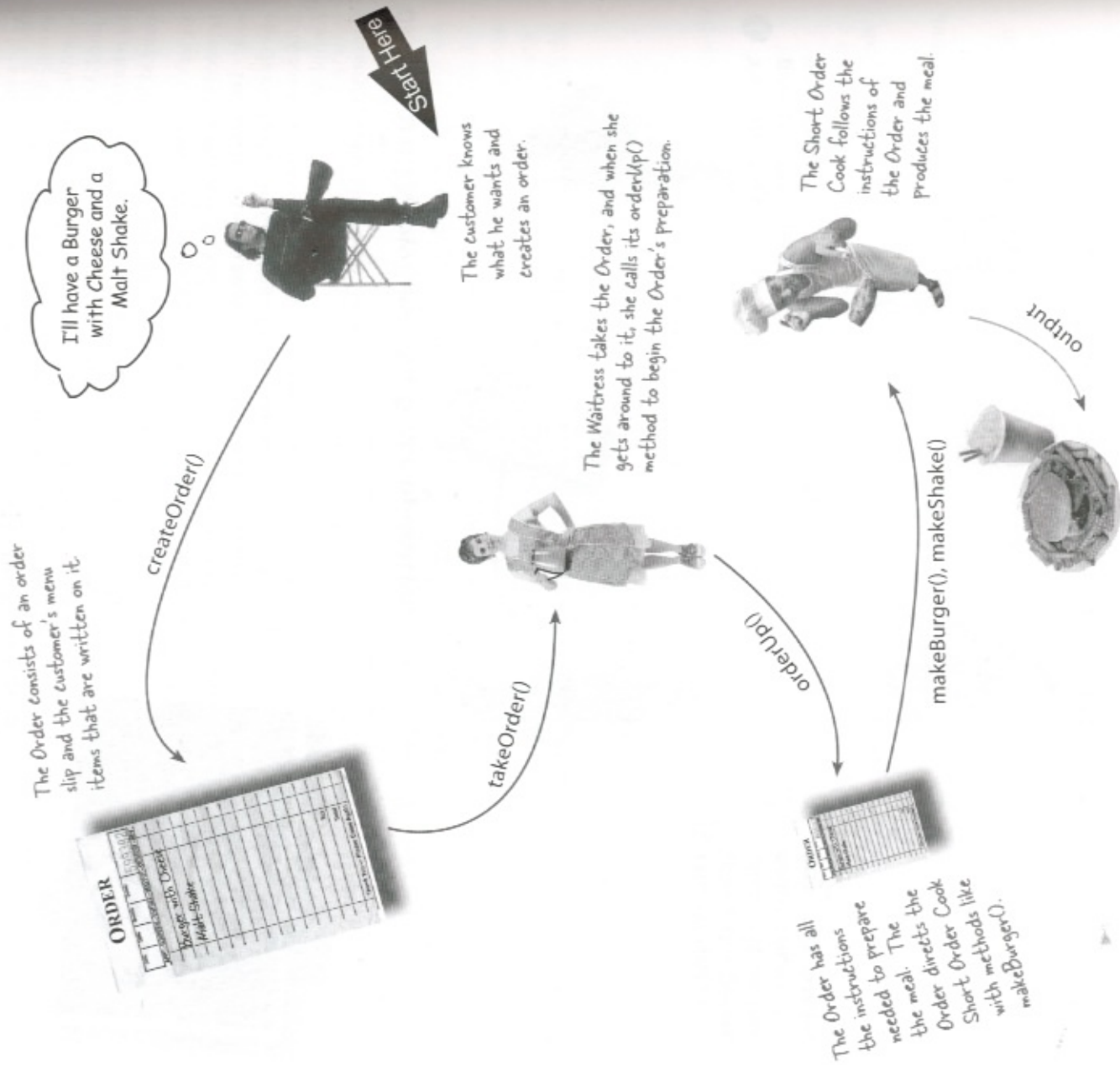


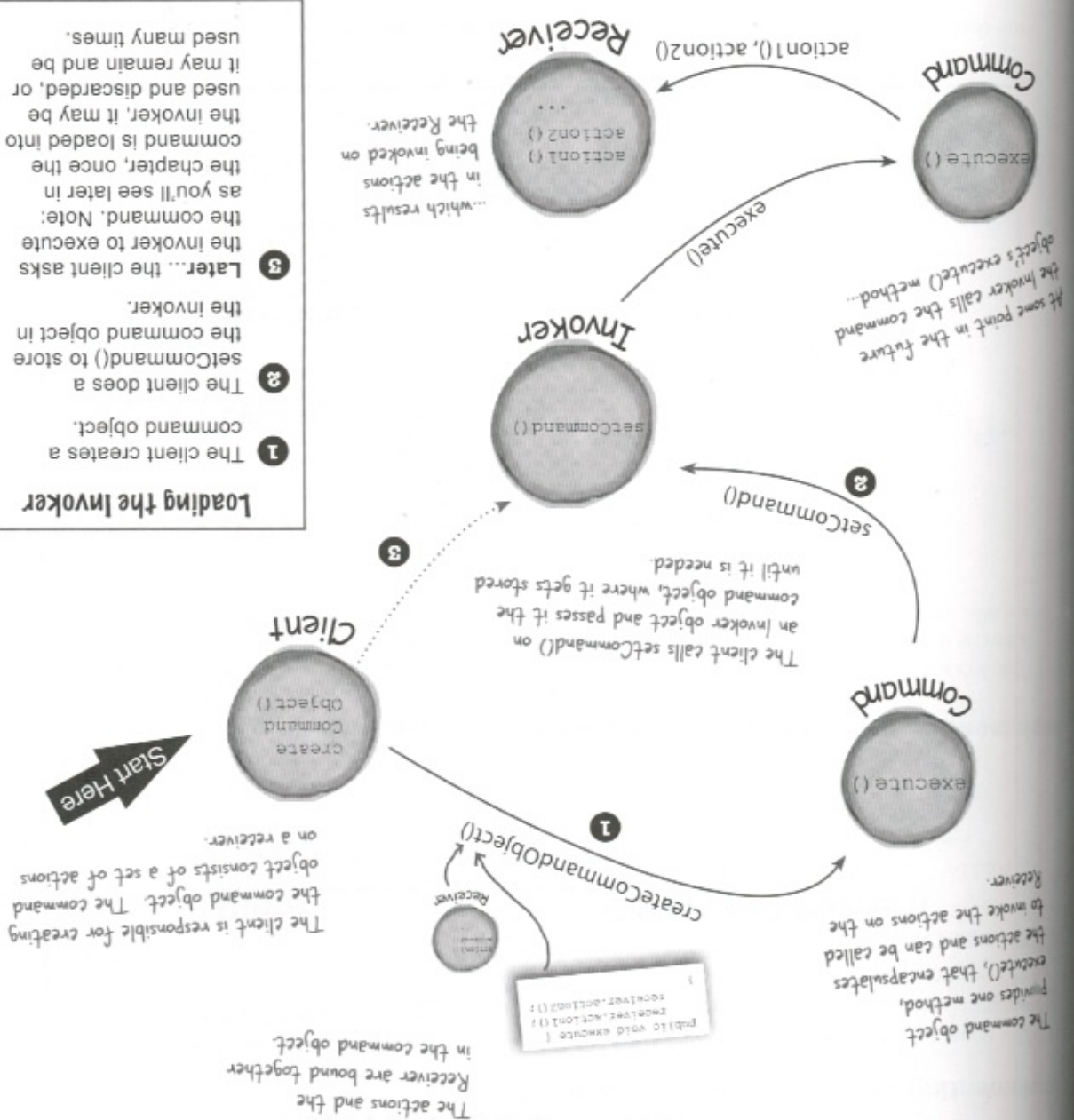
Let's study the interaction in a little more detail...

...and given this Diner is in Objectville, let's think about the object and method calls involved, too!



From the Diner to the Command Pattern

Okay, we've spent enough time in the Objectville Diner that we know all the personalities and their responsibilities quite well. Now we're going to rework the Diner diagram to reflect the Command Pattern. You'll see that all the players are the same; only the names have changed.



Loading the Invoker

- 1 The client creates a command object.
- 2 The client does a `setCommand()` to store the command object in the invoker.
- 3 Later... the client asks the invoker to execute the command, once the chapter, it may be used and discarded, or it may remain and be used many times.

Assigning Commands to slots

the command pattern

So we have a plan: We're going to assign each slot to a command in the remote control. This makes the remote control our *invoker*. When a button is pressed the *execute()* method is going to be called on the corresponding command, which results in actions being invoked on the receiver (like lights, ceiling fans, stereos).

