Tecniche di Progettazione: Design Patterns

GoF: Façade

Amplifier

tuner
dvdPlayer
cdPlayer

on()
off()
setCd()
setDvd()
setStereoSound()
setSurroundSoud()
setTuner()
setVolume()

Tuner

amplifier

on()
off()
setAm()
setFm()
setFrequency()

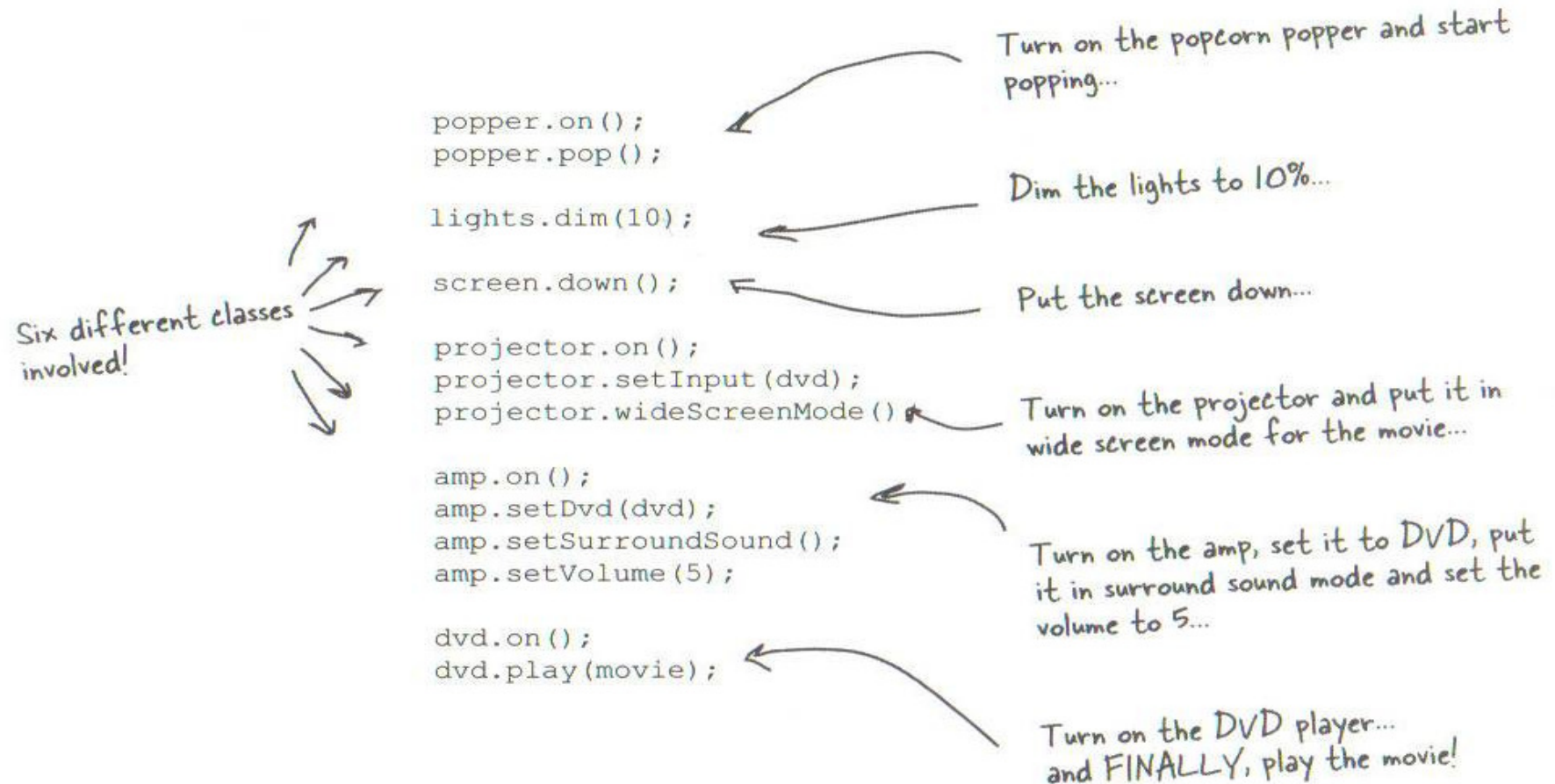**DvdPlayer**

amplifier

on()
off()
eject()
pause()
play()
play()
setSurroundAudio()
setTwoChannelAudio()
stop()
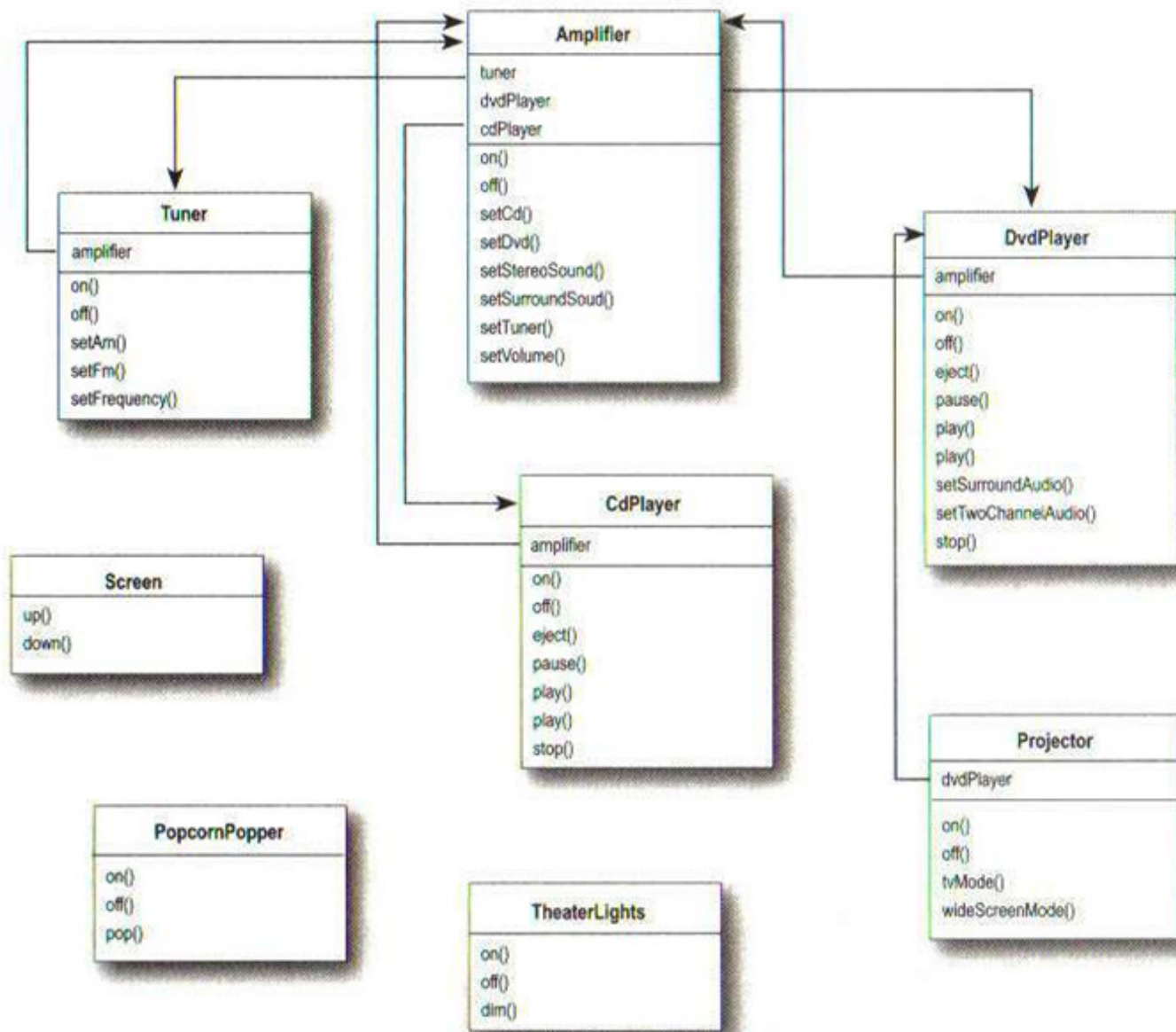
That's a lot of classes, a lot of interactions, and a big set of interfaces to learn and use

Screen

up()
down()

CdPlayer

amplifier

on()
off()
eject()
pause()
play()
play()
stop()

Projector

dvdPlayer

on()
off()
tvMode()
wideScreenMode()

PopcornPopper

on()
off()
pop()

TheaterLights

on()
off()
dim()

Design patterns, Laura Semini, Università di Pisa, Dipartimento di Informatica.
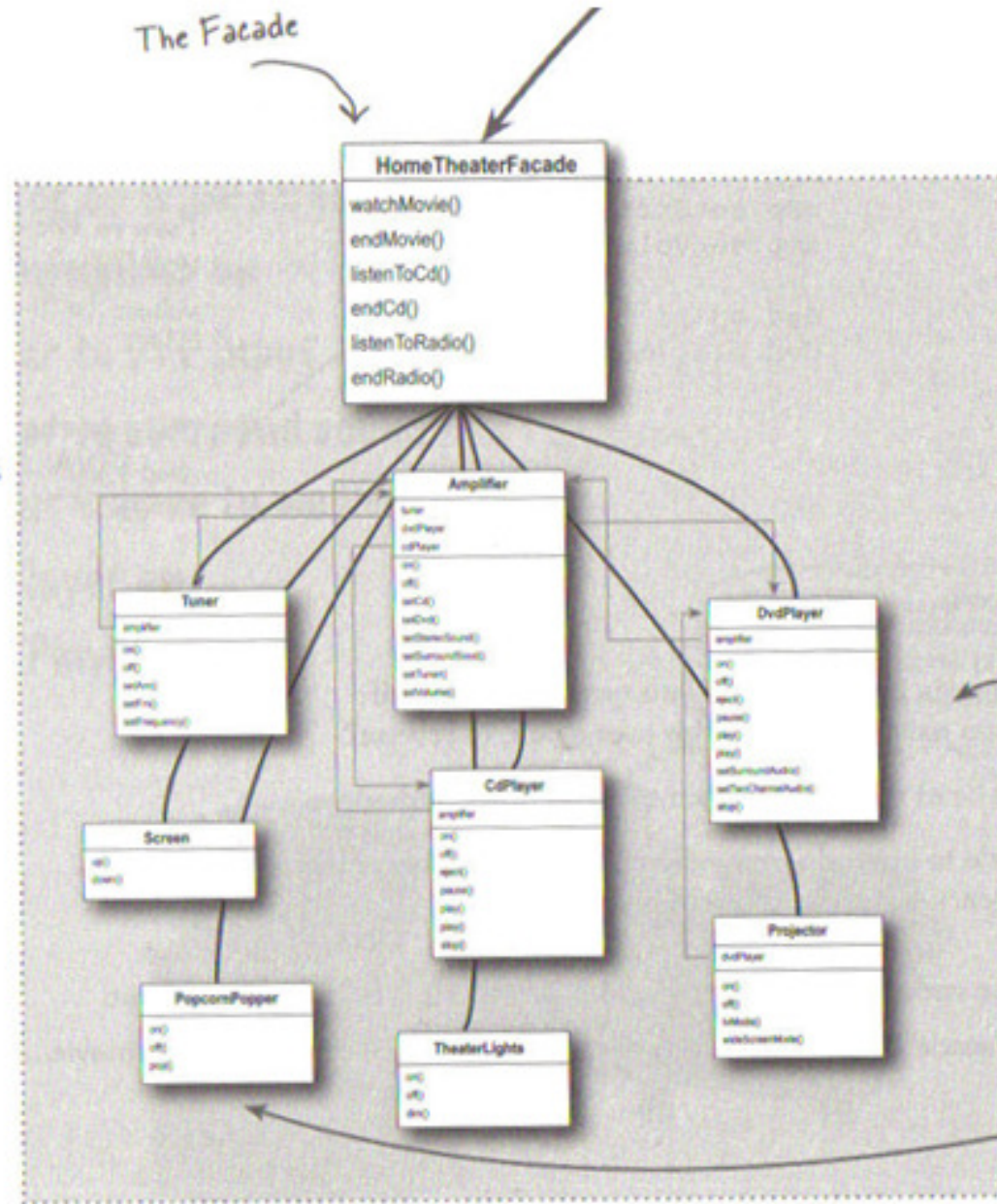
Watching the movie the hard way....

① Turn on the popcorn popper

② Start the popper popping

③ Dim the lights

④ Put the screen down

⑤ Turn the projector on

⑥ Set the projector input to DVD

⑦ Put the projector on wide-screen mode

⑧ Turn the sound amplifier on

⑨ Set the amplifier to DVD input

⑩ Set the amplifier to surround sound

⑪ Set the amplifier volume to medium (5)

⑫ Turn the DVD Player on

⑬ Start the DVD Player playing

What needs to be done to watch a movie....

Turn on the popcorn popper and start popping...

```
popper.on();
popper.pop();
```

Dim the lights to 10%...

```
lights.dim(10);
```

```
screen.down();
```

Put the screen down...

Six different classes involved!

```
projector.on();
projector.setInput(dvd);
projector.wideScreenMode();
```

Turn on the projector and put it in wide screen mode for the movie...

```
amp.on();
amp.setDvd(dvd);
amp.setSurroundSound();
amp.setVolume(5);
```

Turn on the amp, set it to DVD, put it in surround sound mode and set the volume to 5...

```
dvd.on();
dvd.play(movie);
```

Turn on the DVD player... and FINALLY, play the movie!

The Facade

HomeTheaterFacade

watchMovie()

endMovie()

listenToCd()

endCd()

listenToRadio()

endRadio()

❶ Okay, time to create a Facade for the home theater system. To do this we create a new class HomeTheaterFacade, which exposes a few simple methods such as watchMovie().

❷ The Facade class treats the home theater components as a subsystem, and calls on the subsystem to implement its watchMovie() method.

play()

on()

The subsystem the Facade is simplifying.

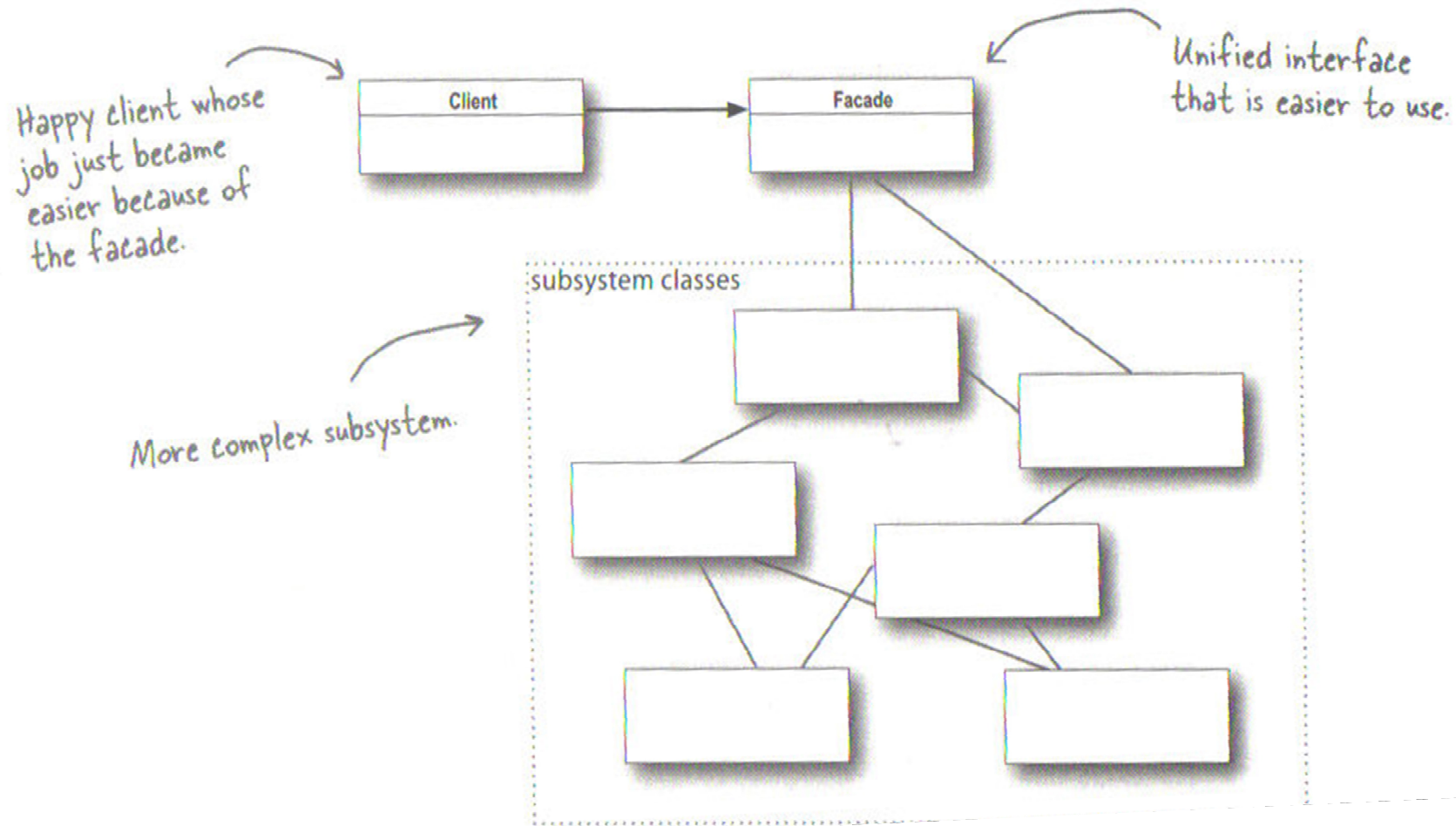**Design patterns, Laura Semini, Università di Pisa, Dipartimento di Informatica.**
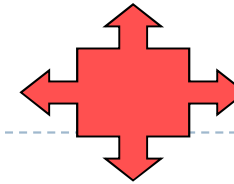
Façade Pattern defined

The Façade Pattern provides a unified interface to a set of interfaces in a subsystem. Façade defines a higher level interface that makes the subsystem easier to use.

Façade pattern – Class Diagram

Happy client whose job just became easier because of the facade.

Unified interface that is easier to use.

More complex subsystem.

Client

Facade

subsystem classes

Design Principle

Principle of Least Knowledge
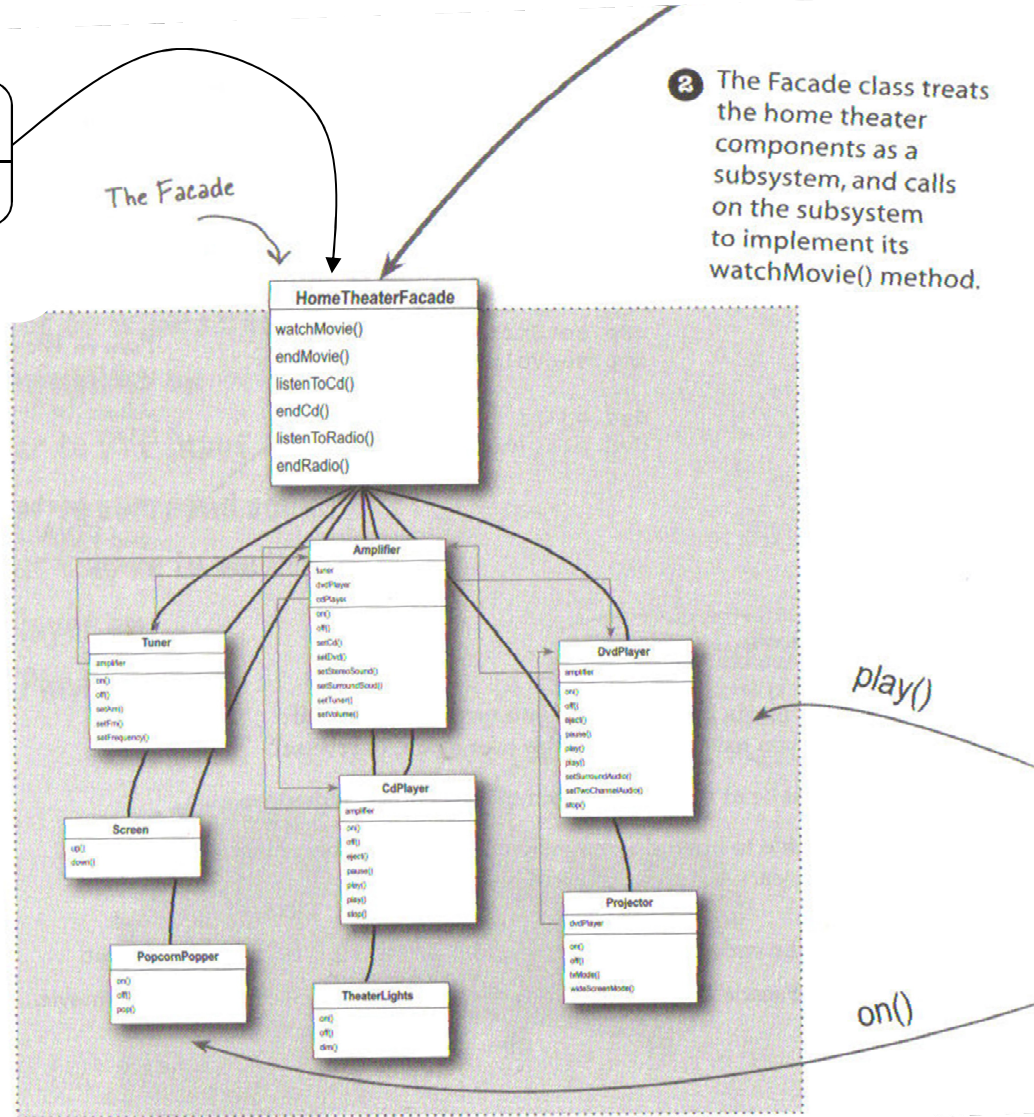
  talk only to your immediate friends

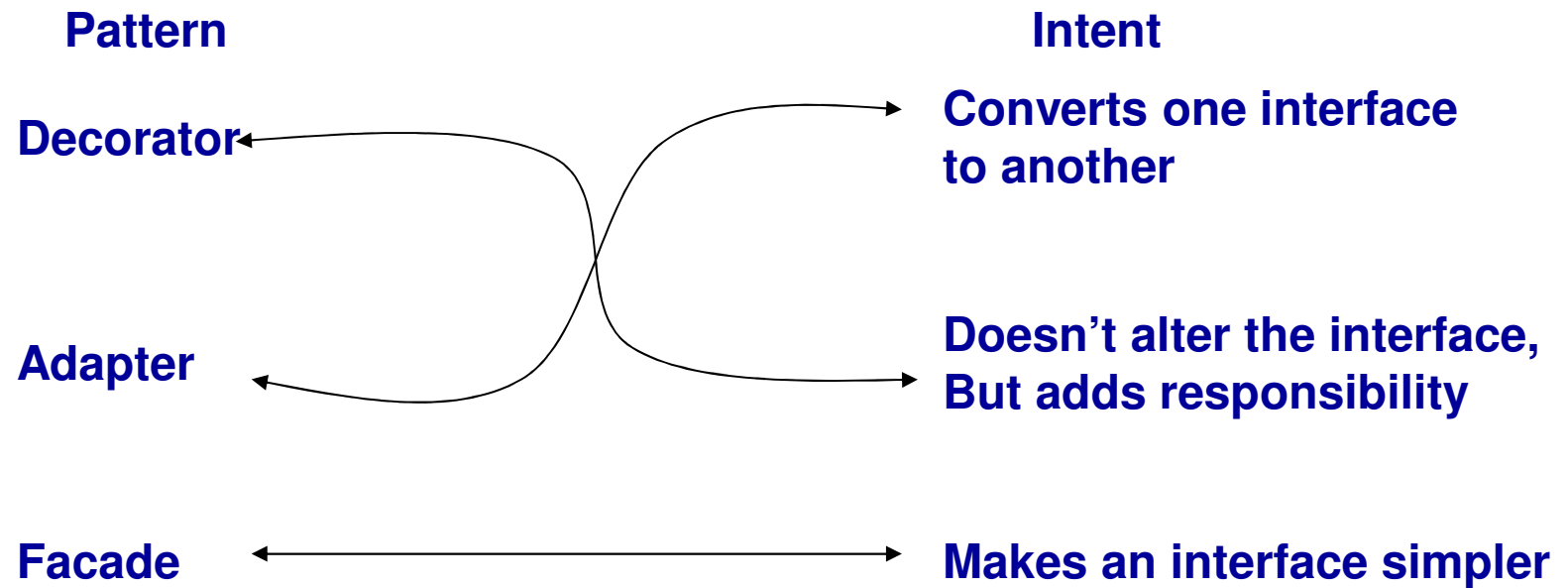Basically this says minimize your dependencies

Client

The client only has
one friend – and
that is a good thing

If the subsystem
gets too complicated
One can recursively
apply the same
principle.

The Facade

2 The Facade class treats
the home theater
components as a
subsystem, and calls
on the subsystem
to implement its
watchMovie() method.

HomeTheaterFacade

watchMovie()
endMovie()
listenToCd()
endCd()
listenToRadio()
endRadio()

Amplifier

tuner
dvdPlayer
cdPlayer

on()
off()
setCd()
setDvd()
setStereoSound()
setSurroundSound()
setTuner()
setVolume()

Tuner

amplifier

on()
off()
setAm()
setFm()
setFrequency()

DvdPlayer

amplifier

on()
off()
eject()
pause()
play()
play()
setSurroundAudio()
setTwoChannelAudio()
stop()

play()

CdPlayer

amplifier

on()
off()
eject()
pause()
play()
play()
stop()

Screen

up()
down()

Projector

dvdPlayer

on()
off()
tvMode()
wideScreenMode()

PopcornPopper

on()
off()
pop()

TheaterLights

on()
off()
dim()

on()

Design patterns, Laura Semini, Università di Pisa, Dipartimento di Informatica.

A little comparison

Pattern

Decorator

Adapter

Facade

Intent

Converts one interface to another

Doesn't alter the interface, But adds responsibility

Makes an interface simpler

Discussion

▶ One common problem experienced by software development teams who use the Façade Pattern occurs when the Façade class is used to represent the entire system on which the team is working.

▶ A team of 20-30 people sends every method call to the system through the Façade, with each team member making several changes to the system per day.

▶ Because of the heavy dependency on the Façade class, however, the team's schedule is frequently delayed because the Façade class is often locked by a particular developer for quite some time.

▶ Discuss how this problem might be overcome without sacrificing the use of the Façade Pattern