

Messages

www.cs.uoi.gr/~zarras/soft-devII.htm

Sources: M. Fowler Refactorings Catalog
W. Wake Refactoring Workbook

Message Chains

Message Chains

Symptoms

You see calls of the form: **a.b().c().d()**

Causes

An object cooperates with other objects to get things done. That part is OK; the problem is that we're **coupled** both to the **objects** and *the path to get to them*.

This sort of coupling goes against two maxims of object-oriented programming: **"Tell, Don't Ask"** and the **Law of Demeter**.

"Tell, Don't Ask," says that instead of *asking for* objects so you can manipulate them, *tell them to just do the manipulation for you*.

It's phrased even more clearly in the Law of Demeter a method shouldn't talk to strangers.

Message Chains

What to Do

Use **Hide Delegate** to make the method depend on one object only. (So, rather than **a.b().c().d()**, **put a d() method on the a object.**)

Message Chains

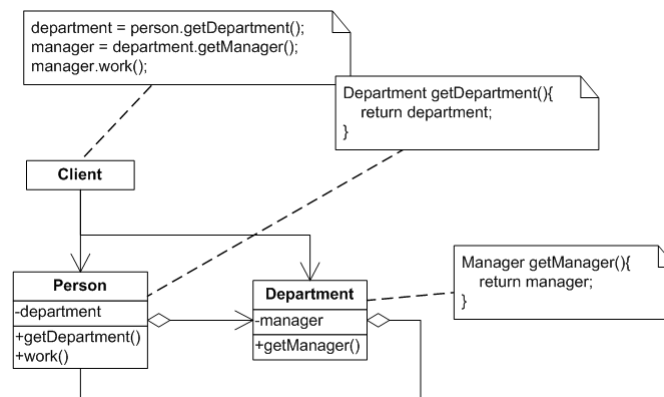
What to Do

If the **manipulations** actually belong on the **target object** (the one at the end of the chain), use *Extract Method* and *Move Method* to put them there.

Hide Delegate

A client is calling a delegate class of an object.

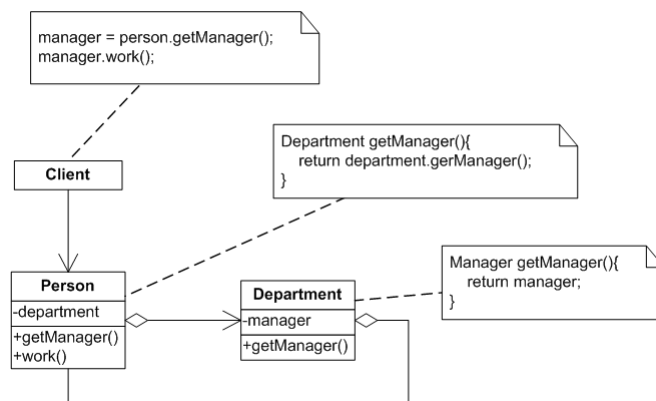
Create methods on the server to hide the delegate.



Hide Delegate

A client is calling a delegate class of an object.

Create methods on the server to hide the delegate.



Middle Man

Middle Man

Symptoms

Most methods of a class call the same or a similar method on another object:

```
f() {delegate.f();}
```

Causes

One way this happens is from applying [Hide Delegate](#) to address **Message Chains**. Perhaps other [features have moved out](#) since then, and we're [left with mostly delegating methods](#).

Middle Man

What to Do

Remove Middle Man by having the client call the delegate directly.

z1

Slide 10

z1 this means that for each middle object there is a different delegate object

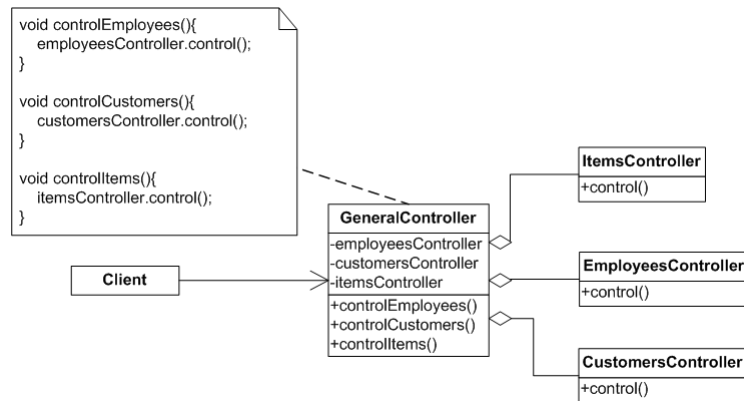
the objects always co-exist

zarras, 15/12/2017

Remove Middle Man

A class is doing too much simple delegation.

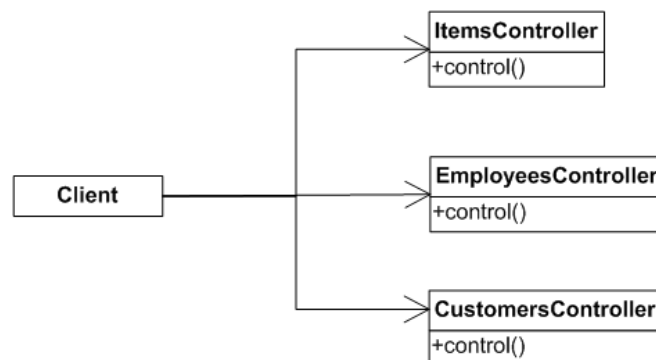
Get the client to call the delegate directly.



Remove Middle Man

A class is doing too much simple delegation.

Get the client to call the delegate directly.



Middle Man

What to Do

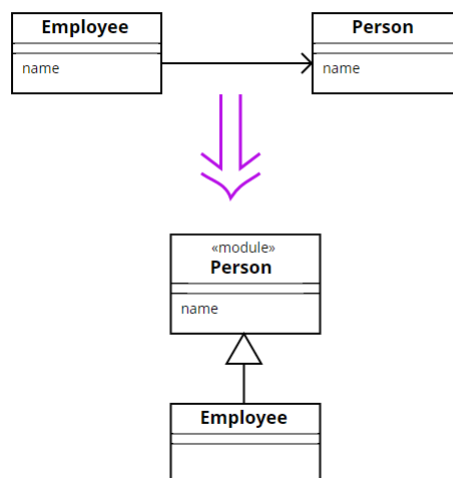
If the delegate is **owned** by the middle man or is **immutable**, and the **middle man** has behavior to add, and the **middle man can be seen as an example of the delegate**, you might **Replace Delegation with Inheritance**.

z2

Replace Delegation with Inheritance

You're using delegation and are often writing many simple delegations for the entire interface

Make the delegate a module and include it into the delegating class.



Slide 13

z2 this means that for each middle object there is a different delegate object

the objects always co-exist
zarras, 15/12/2017