**Design Patterns**

**Delegation**

Suppose we were building an app to manage a clinic. Our models might look like this to begin with:

class Office < ApplicationRecord

belongs\_to :doctor

end

class Doctor < ApplicationRecord

has\_one :office

has\_many :patients

end

class Patient < ApplicationRecord

belongs\_to :doctor

end

When documenting a patient's appointments we want to ensure we track which office and doctor the patient will be visiting.

A view of this might look something like this:

<%= @patient.doctor.name %>

<%= @patient.doctor.specialty %>

<%= @patient.doctor.office.number %>

<%= @patient.doctor.office.address %>

ActiveRecord is powerful. But with great power, comes great responsibility.

Remember when we talked about the Law of Demeter? It says you should only talk to your neighbors/only use one dot. The HTML above is loaded with LoD violations. Reaching from patient across to office will become a problem: when we want to some day change the address of the office, we'll be in a world of hurt.

One design pattern to overcome Law of Demeter violations is to use *delegation*. We can either write getters by hand like this:

class Patient < ApplicationRecord

belongs\_to :doctor

def doctor\_name

doctor.name

end

end

This allows us to access the doctors name via the patient like this:

<%= @patient.doctor\_name %>

See how we've only got one dot now? We've removed one Law of Demeter violation.

As you can imagine these getters are very common. Let's see how to use the delegate method so that we don't have to write these setters by hand.

We can add a woof method to an owner so that owner.woof calls the woof method of their dog by doing this:

class Owner < ApplicationRecord

has\_one :dog

delegate :woof, to: :dog

end

class Dog < ApplicationRecord

belongs\_to :owner

def woof

puts 'woof'

end

end

Now let's clean up our models so they delegate enough information to remove all of the LoD violations from above. We can use the prefix: true option to sensibly prefix our methods:

class Office < ApplicationRecord

belongs\_to :doctor

end

class Doctor < ApplicationRecord

has\_one :office

has\_many :patients

delegate :number,

:address,

to: :office,

prefix: true

end

class Patient < ApplicationRecord

belongs\_to :doctor

delegate :name,

:specialty,

:office\_number,

:office\_address,

to: :doctor,

prefix: true

end

We can now use our delegate methods in our view.

<%= @patient.doctor\_name %>

<%= @patient.doctor\_specialty %>

<%= @patient.doctor\_office\_number %>

<%= @patient.doctor\_office\_address %>