Proxy

Dmitri Nesteruk
QUANTITATIVE ANALYST

@dnesteruk http://activemesa.com

Overview

Motivation

Standard Library Smart Pointers

Virtual Proxy

Communication Proxy

Proxy vs Decorator

You are calling `foo.bar()`

This assumes that foo resides in the same process as bar

What if, later on, you want to put all Foo related operations into a separate process?

- How can you avoid changing all your code?

Proxy to the rescue!

- Same interface, entirely different behavior

This is a communication proxy

- Others: logging, virtual, guarding, …

Proxy

A class that is functioning as an interface to a particular resource. That resource may be remote, expensive to construct, or may require logging or some other added functionality.

How is Proxy different from Decorator?

- Proxy provides an identical interface; decorator provides an enhanced interface
- Decorator typically aggregates (or has reference to) what it is decorating; proxy doesn't have to
- Proxy might not even be working with a materialized object

Summary

A proxy has the same interface as the underlying object

To create a proxy, simply replicate the existing interface of an object

Add relevant functionality to the redefined member functions

- As well as constructor, destructor, etc.

Different proxies (communication, logging, caching, etc.) have completely different behaviors