Proxy Design Pattern

Coding by indirection

Eileen Hollinger CSPP51023 Winter 2011

overview

- Generally speaking, a proxy pattern controls access to an object via a surrogate or placeholder.
- Common metaphor is as a check can be used as a placeholder for money in a bank account and controls access.

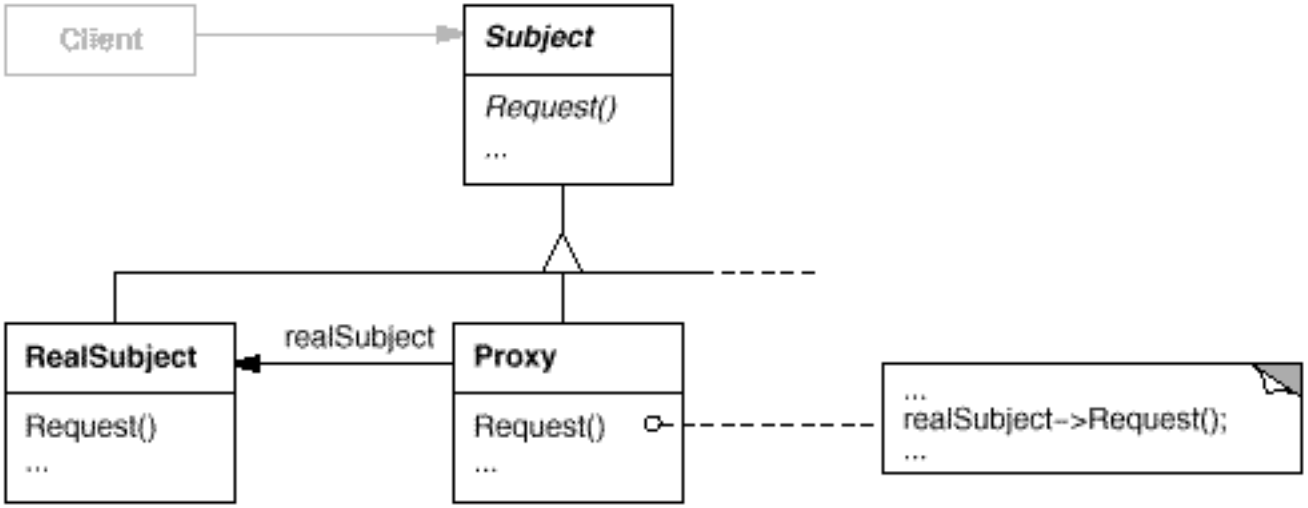
Contexts used

Typically used in one of four contexts:

- *Virtual proxy:* Delays instantiation of expensive objects until necessary
- *Remote proxy:* Provides local representation of a remote object
- *Protection proxy:* Provides control over sensitive master object
- *Smart proxy:* Adds additional functionality to master object

components

- Proxy: maintains reference allowing access to RealSubject
- Client only interfaces with Subject
- Both Proxy and RealSubject implement Subject interface
- Proxy class directs requests to RealSubject as needed.



Example in java

Implementation of a virtual proxy for remote image access

Terminal — emacs-i386 — 99×37

```
public interface Image {
    public void displayImage();
}



public class RealImage implements Image {

    public RealImage(URL url) {
        loadImage(url);
    }

    public void displayImage() {
        //displays image
    }

    private void loadImage(URL url) {
        //loads image. this method only exists in real subject class
    }
}

public class ProxyImage implements Image {
    private URL url;

    public ProxyImage(URL url) {
        this.url = url;
    }

    public void displayImage() {
        RealImage real = new RealImage(url);
        real.displayImage();
    }
}
```

```
-uuu:**-F1  Image.java     All L22     (Java/l Abbrev)----------
Auto-saving...done
```