

Design Patterns

Proxy

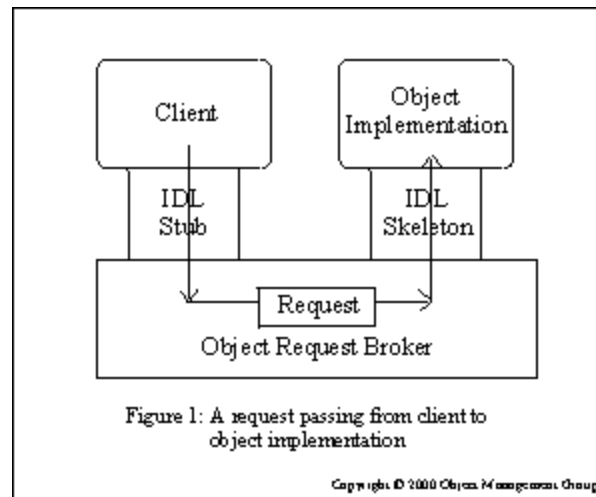
Dr. Chad Williams
Central Connecticut State University

Design pattern: Proxy

- **Category:** Structural design pattern
- **Intent:**
 - Provide surrogate or placeholder for another object to control access to it.
- **Motivation**
 - Control access to actual object
 - Actual object may be remote
 - Cost of actual operation is expensive so defer until needed
 - Monitor object access

Applicability

- Applicable wherever need for more versatile or sophisticated reference to an object
 - Remote proxy
 - Actual object is located in different address space, Proxy acts as ambassador for remote object. Ex. remote EJBs, Common Object Request Broker Architecture (CORBA)



Ex. CORBA interaction

Applicability cont.

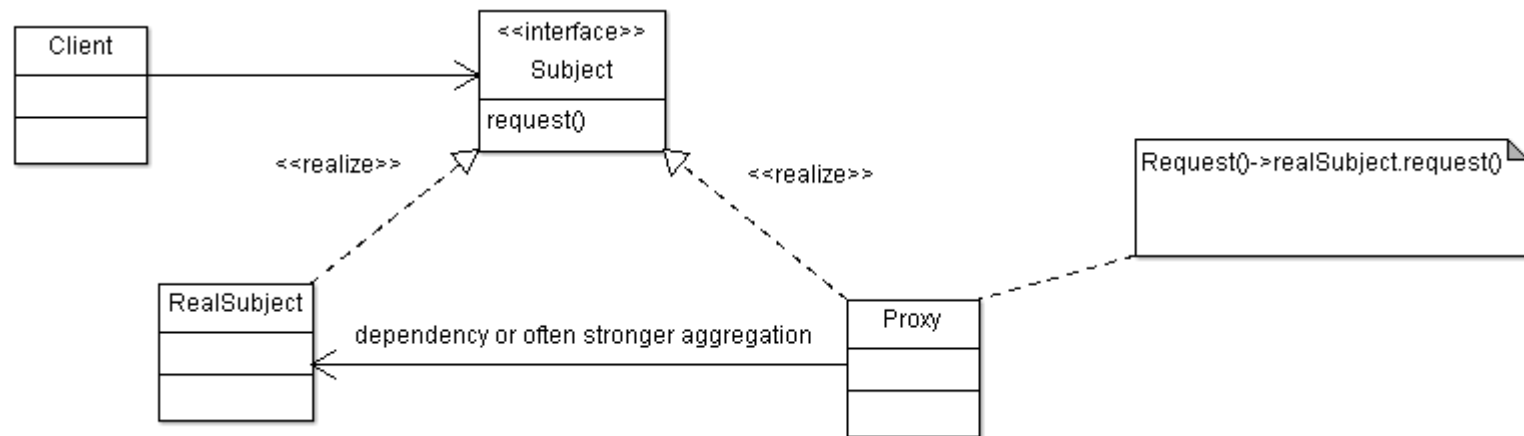
- Applicable wherever need for more versatile or sophisticated reference to an object
 - Virtual proxy
 - Create expensive objects on demand rather than up front (example to follow)
 - Protection proxy
 - Used to enforce when objects should have different access rights. Verify access before actual object called
 - Smart reference
 - Ex. Track number of references held or have multiple point to same object in memory until modification is made then split to new actual instance reference.

Participants

- Proxy
 - Maintains a reference that lets the proxy access the real subject – may refer to Subject or RealSubject
 - Provides interface identical to Subject's
 - Controls access to RealSubject may be responsible for creating/destroying
 - Others
 - *Remote proxy* – responsible for encoding request sent to RealSubject in different address space
 - *Virtual proxy* – optimizations such as on-demand
 - *Protection proxy* and *smart reference* – additional house cleaning

Participants cont.

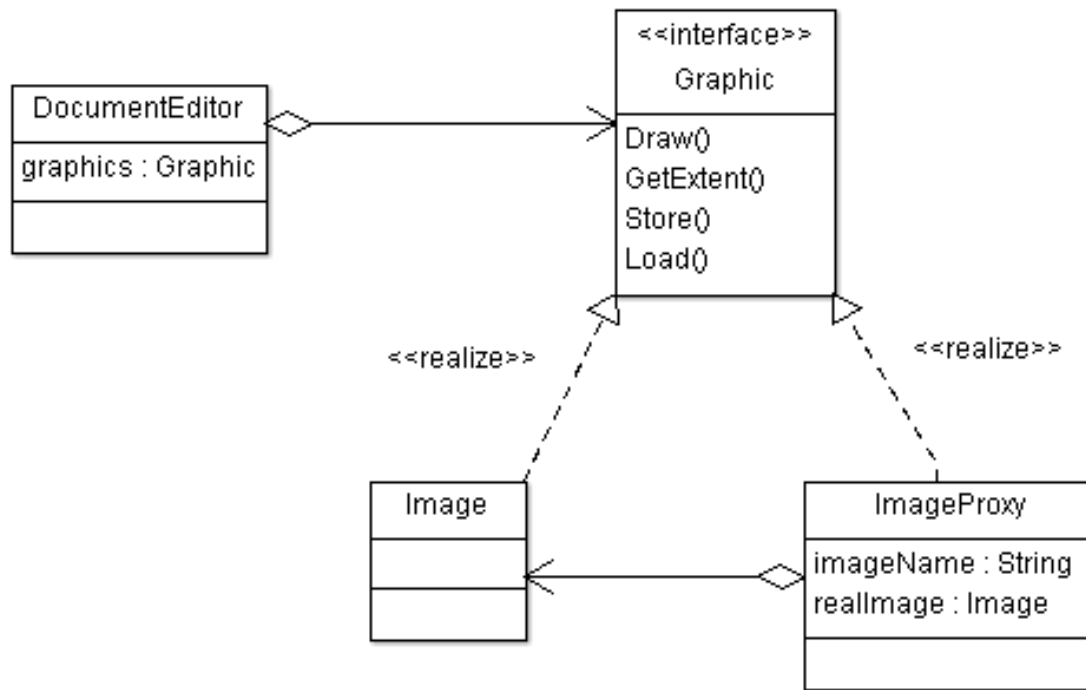
- Subject
 - Defines common interface for RealSubject and Proxy so Proxy can be used interchangeably
- RealSubject
 - Real object proxy represents



Motivation cont. - virtual proxy

- Motivation – Multi-page Word document with lots of embedded pictures
 - If load all when document opened very expensive
 - Time consuming
 - Resource consuming
 - Instead defer actual loading until page is visible
 - Load on demand

Proxy UML



More proxy examples

- Remote proxy
 - Ex. remote EJBs, Common Object Request Broker Architecture (CORBA)
- Protection proxy
 - Security protection
- Smart reference
 - All refer to same until modification made combining with flyweight pattern
- Decorator vs Proxy
 - Both provide identical interface
 - Decorator – goal add additional responsibilities or functionality dynamically without subclassing can recurse multiple decorators
 - Proxy – intent is to stand in when inconvenient or undesirable to access class directly – tied to single class not recursive

In class examples

- Identify two concrete example where different forms of Proxy would make sense
 - Remote proxy
 - Virtual proxy
 - Protection proxy
 - Smart reference