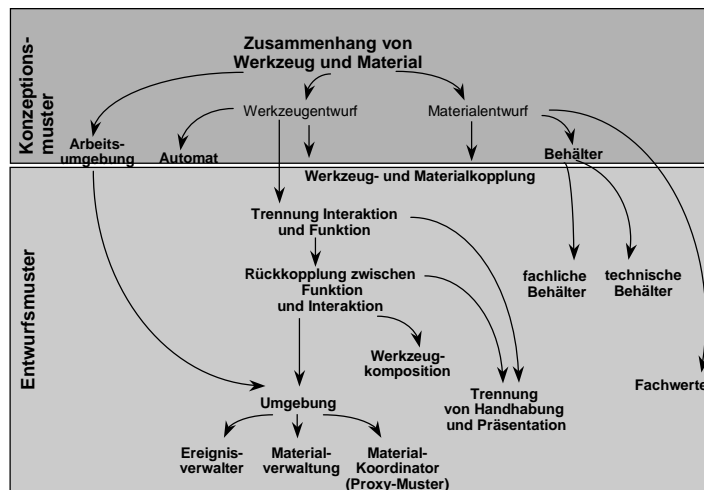


Konzeptions- und Entwurfsmuster



Das Proxy

Zweck

Kontrolliere den Zugriff auf ein Objekt mit Hilfe eines vorgelagerten Stellvertreterobjektes.

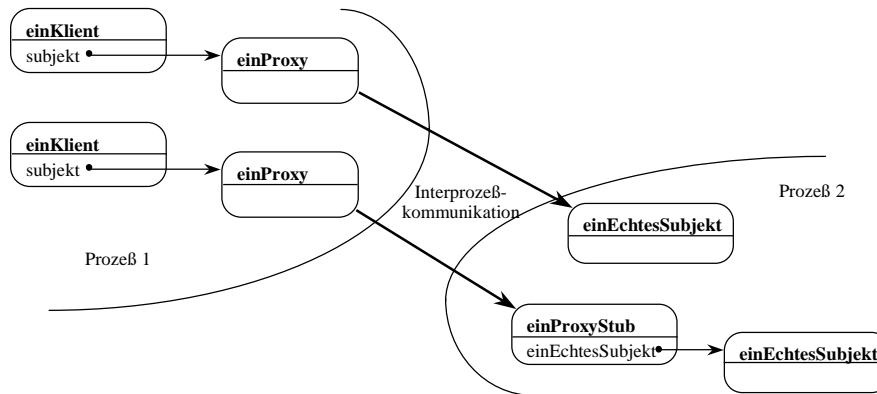
Anwendbarkeit

Es wird eine anpassungsfähigere und intelligentere Referenz auf ein Objekt als ein einfacher Zeiger benötigt:

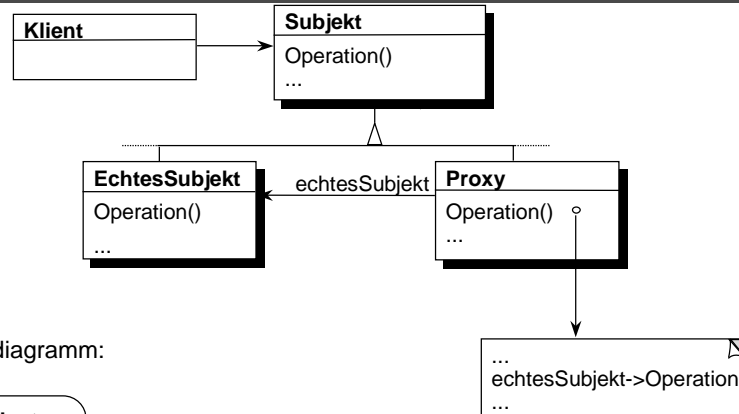
- Teure Objekte sollen erst auf Verlangen erzeugt und initialisiert werden (*Virtuelles Proxy*).
- Der Zugriff auf ein Originalobjekt soll kontrolliert erfolgen (*Schutzproxy*).
- Es soll ein lokaler Stellvertreter für ein Objekt eines anderen Adreßraumes bereitgestellt werden (*Remote Proxy*).
- Bei der Verfolgung von Referenzen sollen zusätzliche Aktionen (z.B. Laden persistenter Objekte bei ihrer ersten Dereferenzierung) ausgeführt werden (*Smart Pointer*).

Beispiel: Das Remote-Proxy

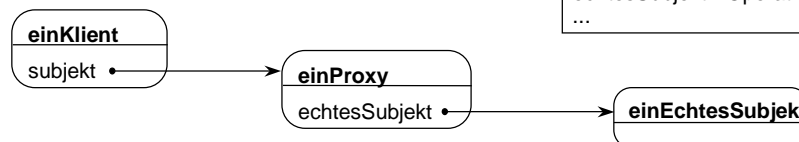
Das Remote-Proxy wird als lokaler Stellvertreter für potentiell verteilte Objekte verwendet. Beim Erhalt einer Nachricht codiert das Proxy sie zusammen mit ihren Argumenten und leitet die codierte Nachricht an das entfernt liegende Subjekt.



Struktur des Proxy-Musters



Objektdiagramm:



Diskussion des Proxy-Musters

- Das Proxymuster führt eine Ebene der Indirektion beim Zugriff auf ein Objekt ein:
 - Ein Remote-Proxy versteckt die Tatsache, daß sich ein Objekt in einem anderen Prozeßraum befindet.
 - Ein virtuelles Proxy kann Optimierungen ausführen, wie etwa das Kopieren schwerer Objekte auf Verlangen ("Copy-on-Write").
 - Sowohl Schutzproxies als auch Smart Pointer ermöglichen die Durchführung zusätzlicher Verwaltungsaufgaben, wenn auf das Objekt zugegriffen wird.
- Eine Proxyklasse muß nicht notwendig die konkrete Klasse seines EchtenSubjektes kennen, sondern kann mittels einer abstrakten Schnittstelle für eine ganze Familie echter Subjekte geeignet sein.