Sicurezza

- Al fine di sviluppare un sistema sicuro, oltre ai requisiti funzionali occorre includere i requisiti di sicurezza
- A tale scopo bisogna avere una rappresentazione delle regole di sicurezza dell'organizzazione che usa il sistema e mettere in essere un modello di sicurezza ai vari livelli di progettazione ed implementazione del sistema
- Vari design pattern si possono usare a tal fine
  - Authorization (semplice) e Reference Monitor (più completo) sono usati per definire le regole di accesso alle risorse
  - Role Based Access Control è adatto per organizzazioni complesse

Pattern Reference Monitor

aka Policy Enforcement Point [Schumacher]

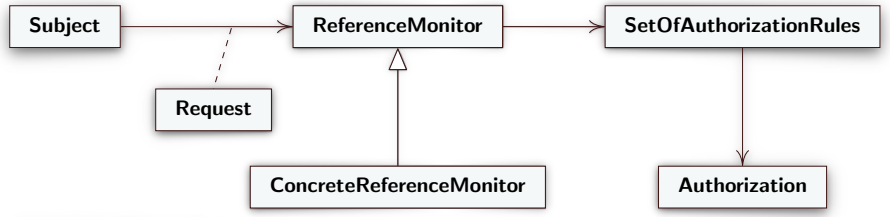
- Intento: in un ambiente in cui si hanno dati o risorse, bisogna imporre restrizioni di accesso. Il pattern descrive come definire un'astrazione che intercetta le richieste e controlla la conformità con le autorizzazioni
- Problema: se non imponessimo le autorizzazioni definite sarebbe come non averle, i richiedenti potrebbero effettuare qualsiasi azione non consentita. Per es., un utente potrebbe leggere da qualsiasi file. Inoltre, un'autorizzazione potrebbe essere espressa in modo complesso, per es. consistere nel consentire di leggere solo una parte di un file
- Forze
  - Le regole di autorizzazione definite devono essere imposte nel momento in cui vi è una richiesta per un oggetto protetto
  - Varie possibili implementazioni sono possibili, occorre un modello
  - Ci possono essere decisioni complesse e set di attributi da valutare

Reference Monitor

- Soluzione: definire una componente che valuta (intercetta) tutte le richieste, controlla la conformità con le autorizzazioni, prende decisioni in base alle autorizzazioni, e inoltra le richieste conformi
  - **ProtectionObject** è l'oggetto da proteggere
  - **Subject** è chi richiede di accedere a ProtectionObject
  - **Request** incapsula la richiesta da parte di Subject
  - **ReferenceMonitor** intercetta la richiesta e cerca fra le regole di autorizzazione una regola che può essere usata, in base alla richiesta
  - **ConcreteReferenceMonitor** implementa una specifica versione di ReferenceMonitor per un certo tipo di risorsa (per es. i file)
  - **SetOfAuthorizationRules** è un insieme di regole di autorizzazione, organizzate opportunamente
  - **Authorization** è la singola regola di autorizzazione

Reference Monitor

- Soluzione: diagramma UML delle classi



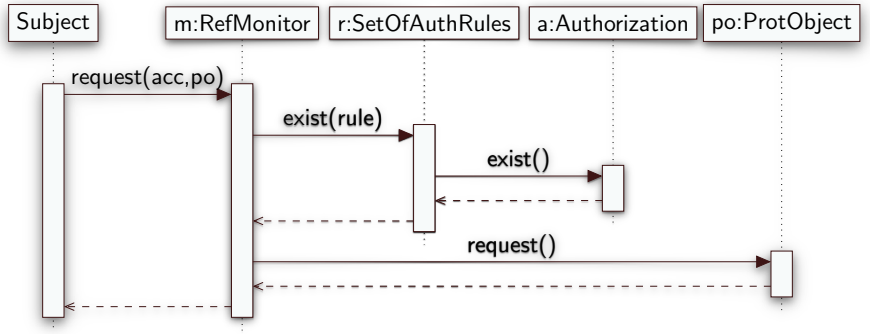
Design Pattern Reference Monitor

Considerazioni Su Request

- Request è una association class (prevista in UML)
- Request caratterizza l'associazione Subject e ReferenceMonitor
  - Poiché la richiesta non è un attributo né di Subject, né di ReferenceMonitor, creo un oggetto a parte
  - Le istanze di Request sono oggetti che permettono di legare Subject e ReferenceMonitor
- Implementazione di una association class
  - Request è una classe che contiene alcuni attributi che qualificano la richiesta
  - Tipicamente, Subject contiene un riferimento all'oggetto Request e lo passa a ReferenceMonitor

Reference Monitor

- Soluzione: diagramma UML di sequenza che verifica la presenza di autorizzazioni appropriate
  - acc indica il tipo accesso richiesto



Design Pattern Reference Monitor

Reference Monitor

- Conseguenze
  - Se tutte le richieste sono intercettate, possiamo imporre tutte le regole di accesso
  - Seguendo il pattern, varie implementazioni possono essere fatte (senza forzature)
  - Controllare tutte le richieste può far degradare le prestazioni. In alternativa, si possono tirar fuori i controlli su operazioni di un flusso di esecuzione, ed inserire **un solo controllo a monte** del flusso. Es., al momento di apertura di un file, e non per ogni operazione sul file
  - Si possono tenere in memoria le decisioni prese precedentemente, associandole alle richieste e al ProtectionObject, evitando quindi di valutare nuovamente le regole

Considerazioni

- Il ReferenceMonitor dovrebbe poter intercettare tutte le richieste per poterle esaminare
- Un modo per implementare la redirezione (intercettazione) delle richieste è implementare un Proxy per le risorse da proteggere (i ProtectionObject)
- Come assicurare che nessuna chiamata arrivi ad un ProtectionObject senza passare dal ReferenceMonitor?
  - Nascondendo il ProtectionObject (per es. rinominandolo)

Esempio Di Reference Monitor

- Per l'esempio precedente (applicazione Book), le classi hanno i seguenti ruoli per il design pattern Reference Monitor
  - `Book` ha il ruolo di ProtectionObject
  - `Protection` ha il ruolo di Proxy
  - `Client` ha il ruolo di Subject
  - `RefMonitor` ha il ruolo di ReferenceMonitor
  - `SetOfAuthRules` ha il ruolo di SetOfAuthorizationRules
  - `Authorization` ha il ruolo di Authorization
  - `Rule` è un parametro che individua la regola da applicare per valutare se si ha l'autorizzazione

Esempio