Lezione n.10
LPR- Informatica Applicata
RMI CallBacks

15/5/2006

Laura Ricci

RMI: IL MECCANISMO DELLE CALLBACK

Meccanismo RMI

- comunicazione unidirezionale (dal client al server)
- *comunicazione sincrona, rendez voz esteso*: il client invoca un metodo remoto e si blocca finchè il metodo non termina

In molte applicazioni

- il client è interessato ad un evento che si verifica sul server e notifica il suo interesse al server (ad esempio utilizzando RMI)
- il server **registra** che il client è interessato in quell'evento
- quando l'evento si verifica, **il server lo notifica** ai clients interessati l'accadimento dell'event

RMI : IL MECCANISMO DELLE CALLBACK

Esempi di applicazioni:

- un utente partecipa ad una chat e vuol essere avvertito quando un nuovo utente entra nella 'chat room'

- lo stato di un gioco multiplayer viene gestito da un server. I giocatori notificano al server le modifiche allo stato del gioco. Ogni giocatore deve essere avvertito quando lo stato del gioco subisce delle modifiche.

- gestione distribuita di un'asta: un insieme di utenti partecipa ad un'asta distribuita. Ogni volta che un utente fa una nuova offerta, tutti i partecipanti all'asta devono essere avvertiti

RMI: IL MECCANISMO DELLE CALLBACK

Soluzioni possibili per realizzare nel server un servizio di notifica di eventi al client:

- *Polling:* il client interroga ripetutamente il server, per verificare se l'evento atteso si è verificato. L'interrogazione avviene mediante *l'invocazione di un metodo remoto* (mediante RMI)

  Svantaggio: alto costo per l'uso non efficiente delle risorse del sistema

RMI: IL MECCANISMO DELLE CALLBACK

- *Registrazione* dei clients interessati agli eventi e successiva notifica (asincrona) del verificarsi dell'evento al client da parte del server

  Problema: quale meccanismo utilizza il server per risvegliare il client?

RMI: IL MECCANISMO DELLE CALLBACK

- E' possibile utilizzare RMI sia per l'invocazione client server (registrazione del client) che per quella server-client (notifica del verificarsi di un evento) utilizzando il *meccanismo delle callback*
- Il server definisce una interfaccia remota *ServerInterface* che definisce un metodo remoto che può essere utilizzato dal client per registrarsi
- Il client definisce una interfaccia remota *ClientInterface* che definisce un metodo remoto utilizzato dal server per notificare un evento al client
- Il server ha a disposizione la *ClientInterface* e lo stub relativo
- Il client ha a disposizione *la ServerInterface* e lo stub relativo

RMI: IL MECCANISMO DELLE CALLBACK

- Il client, al momento della registrazione sul server, passa al server un riferimento RC ad un oggetto che implementa la *CLientInterface*
- Il server memorizza RC in una sua struttura dati (ad esempio, un vector)
- al momento della notifica, il server utilizza RC per invocare il metodo remoto di notifica definito dal client.
- In questo modo rendo 'simmetrico' iol meccanismo di RMI, ma…
  il client non registra l'oggetto remoto in un rmiregistry, ma passa un riferimento a tale oggetto al server, al momento della registrazione

ESERCIZIO: GESTIONE FORUM

Si vuole implementare un sistema che implementi un servizio per la gestione di forum in rete. Un forum e' caratterizzato da un argomento su cui diversi utenti possono scambiarsi opinioni via rete. Il sistema deve prevedere un server RMI che fornisca le seguenti funzionalita':

a) apertura di un nuovo forum, di cui e' specificato l'argomento (esempio: giardinaggio)

b) inserimento di un nuovo messaggio indirizzato ad un forum identificato dall'argomento (es: e' tempo di piantare le viole, indirizzato al forum giardinaggio)

c) reperimento dell'ultimo messaggio inviato ad un forum di cui e' specificato l'argomento.

ESERCIZIO:GESTIONE FORUM

Si devono definire almeno le seguenti classi:

- una interfaccia *ForumInterface*, che definisce i metodi del server che possono essere invocati in remoto

- una classe *Forum* che implementa la struttura dati relativa ad un singolo forum. Il numero massimo di messaggi per ogni forum e'definito staticamente ed e' uguale per ogni forum.

- una classe *Forumimpl* che implementa i metodi definiti nella interfaccia ForumInterface. Il numero massimo di forum che possono essere gestiti da Server e' determinato staticamente.

ESERCIZIO:GESTIONE FORUM

- Una classe ForumObj che attiva una istanza dell'oggetto remoto Forumimpl

- una classe *Client* che interagisce con l'utente ed, in base alle richieste dell'utente, invoca i metodi di Server.

- una classe *Messaggio* che implementa la struttura dati che descrive il messaggio inviato al forum .

ESERCIZIO: CALLBACS

Nella lezione precedente è stato assegnato un esercizio per la gestione Elettronica di una elezione a cui partecipano un numero prefissato di candidati. Si chiedeva di realizzare un server RMI che consentisse al client di votare un candidato e di richiedere il numero di voti ottenuti dai candidati fino ad un certo punto.

Modificare l'esercizio in modo che il server *notifichi ogni nuovo voto ricevuto* a tutti i clients che hanno votato fino a quel momento. La registrazione dei clients sul server avviene nel momento del voto.