REAL WORLD AND VIRTUAL WORLD ARCHITECTURE FOR INTERCONNECTING FIRST AND SECOND LIFE

Filippo Sanfilippo

Università degli studi di Catania
Facoltà di Ingegneria

26 Gennaio 2009

Sommario

1. Introduzione
2. Connecting real world and SL
   - Middleware
   - Middleware: vantaggi
   - Middleware: svantaggi
3. Demo, una possibile applicazione
   - Architecture and protocol definitions
   - Dimostrazione
   - Possibili sviluppi
4. Virtual learning
   - Background
   - Possible solution
   - Real time data visualization in SL
5. Riconoscimenti e ringraziamenti

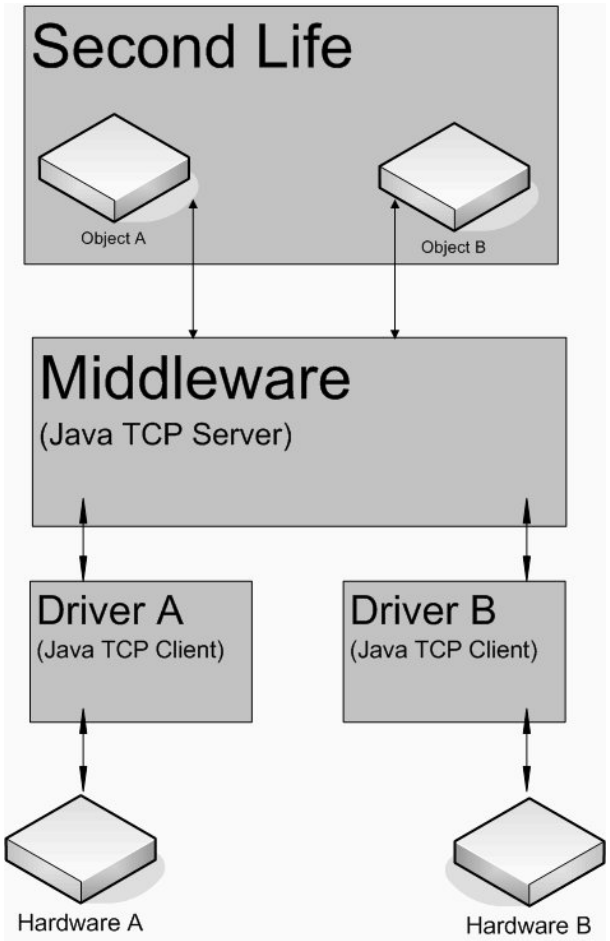
Introduzione

- Un mondo virtuale e' un ambiente interattivo simulato e accessibile da piu' utenti tramite un'interfaccia di rete. Un esempio eclatante e' certamente Second Life.
- Obiettivo: creare un'architettura per interconnettere questi due mondi in modo da stabilire una comunicazione real time tra di loro.
- L'interconnessione tra i due mondi consiste nel creare una relazione di equivalenza e di sincronia tra gli eventi reali e quelli virtuali:
  il verificarsi di un evento reale causa un avvenimento virtuale equivalente e viceversa.
- L'architettura che ho progettato e' il primo passo per risolvere questo problema.
- Utilita' del progetto:
  - controllo remoto
  - esperimenti remoti
  - virtual learning
  - domotica

Perche' ho scelto SL?

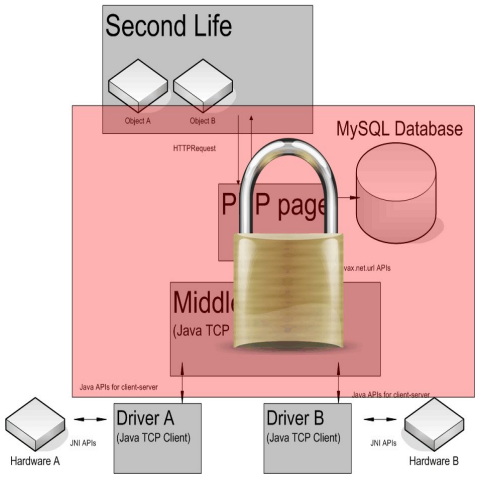
- E' estrmemente flessibile e aperto allo sviluppo, inoltre il client e' opensource e free-to-play
- E' molto semplice dare forma agli oggetti reali
- Tutti gli oggetti e i blocchi costruiti possono contenere codice eseguibile. Lo scripting degli oggetti rende possibile l'interazione tra di loro e con gli avatars. Gli oggetti virtuali possono, quindi, rilevare la presenza degli avatars vicini, sentire le loro parole e reagire al loro tocco.
- Gli oggetti possono anche sfruttare la possibilita' di scambiare dati con il mondo reale

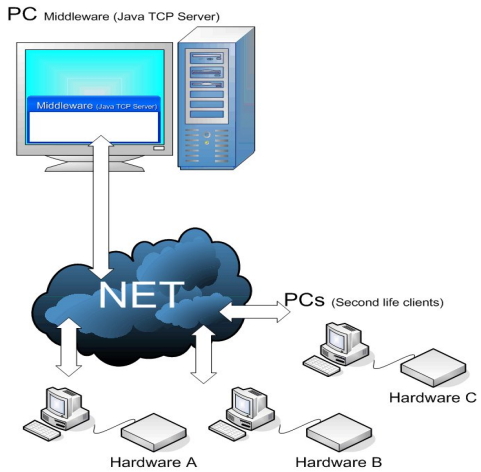
Middleware

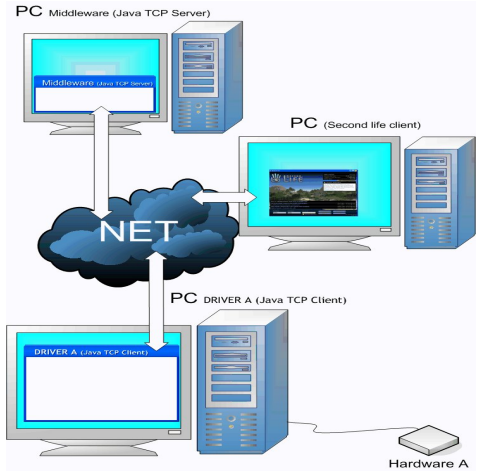

Second Life
Object A
Object B
Middleware
(Java TCP Server)
Driver A
(Java TCP Client)
Driver B
(Java TCP Client)
Hardware A
Hardware B

Middleware: vantaggi

- - Hardware Independent
- - Multi-platform
- - Easy plug-in
- - Multi Client
- - Distribuited System





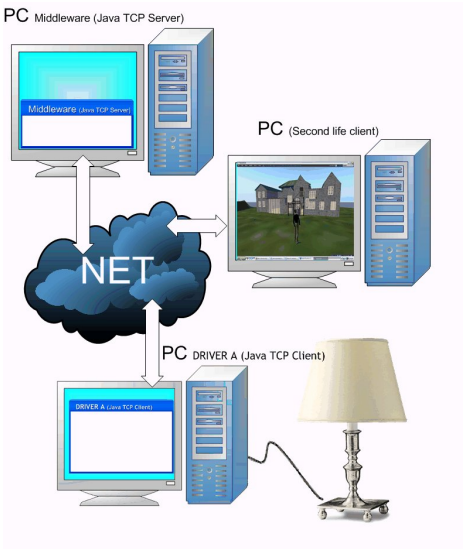


Middleware: svantaggi

- Dal lato di SL siamo costretti a scrivere codice utilizzando solo LSL Scripting che offre solo tre interfacce di comunicazione:
  - E-mail, possono essere scambiate tra SL e un certo indirizzo. Tuttavia, per ridurre il carico gravante sul server e per prevenire l'uso di spam, ci sono delle limitazioni sulla frequenza di invio/ricezione e sui ritardi
  - HTTP Request, e' veloce e easy-to-use, ma e' simplex (in downlink). Quando uno script chiama llHTTPRequest(), il gestore degli eventi associa automaticamente ad esso uno specifico evento httpresponse che catturera' il risultato della risposta
  - XML-RPC, permette agli oggetti virtuali di ricevere richieste dall'esterno, ma e' simplex (in uplink)

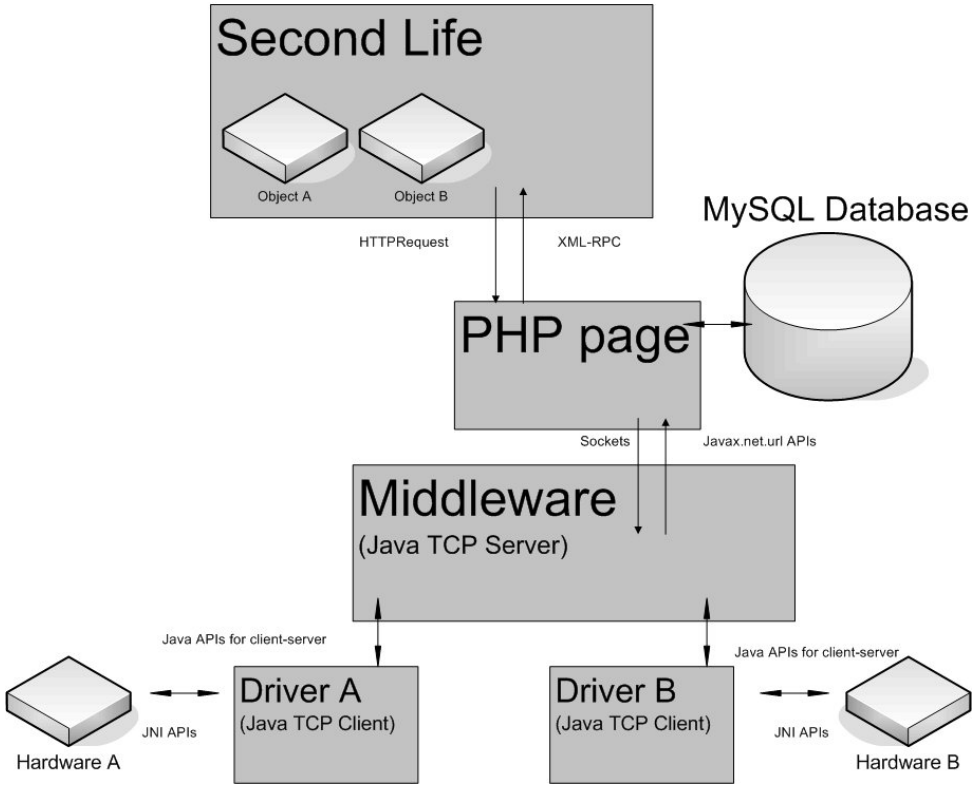
Demo, una possibile applicazione

- Per esempio, si pensi al problema di domotica: accensione/spegnimento delle luci di casa. Immaginiamo di trovarci fuori di casa, potremmo avere la possibilita' di accendere le luci della cucina o magari del viale del giardino direttamente da SL.



Architecture and protocol definitions

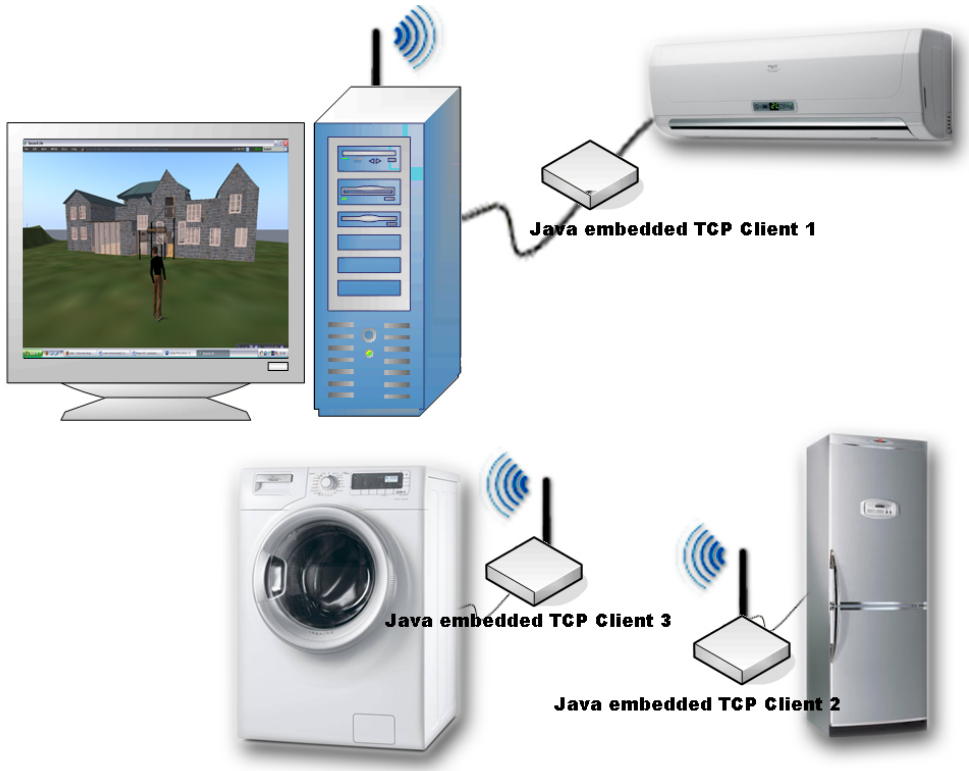
- Ciascun driver deve comunicare sia con il server che con l'hw. E' l'unico pezzo di software che occorre riscrivere se si vuole connettere un nuovo tipo di hw.



Dimostrazione

Possibili sviluppi

- Un possibile scenario applicativo potrebbe essere quello rappresentato in figura. Ciacun elettrodomestico e' reso intelligente tramite l'uso si una 'box' aggiuntiva su cui gira lo specifico driver.



Virtual learning

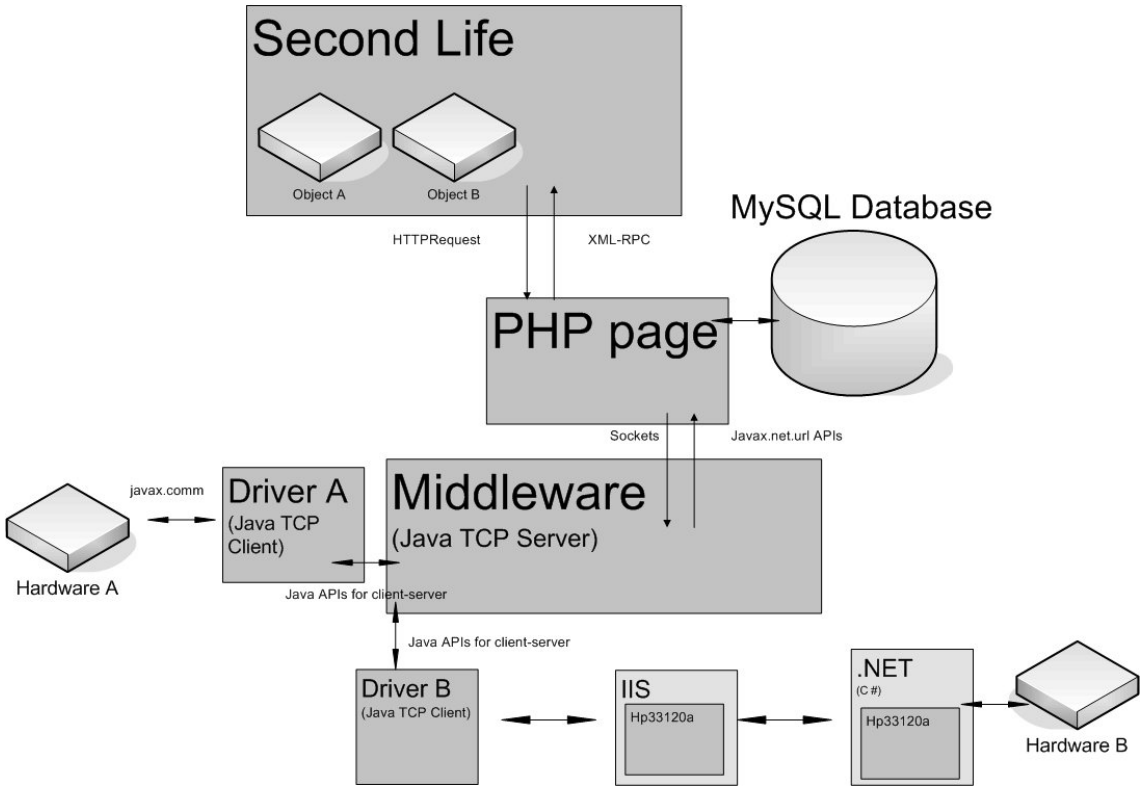
- Per dare un'idea delle enormi potenzialita' della mia architettura vi mostrero' un progetto al quale ho lavorato durante il mio tirocinio. Si tratta di un'applicazione che abbiamo sviluppato nell'ambito del virtual learning.
- La nostra intenzione era quella di utilizzare la stessa architettura del demo precedente per gestire hw piu' complesso. L'hw che abbiamo scelto era formato da un generatore di forma d'onda, un multimetro e un'oscilloscopio.

Background

- Disponevamo di drivers realizzati da un membro dell'University of Ulster. L'unico problema era che questo software era scritto in C mentre il nostro middleware e' scritto in Java.
- Abbiamo deciso di realizzare un'integrazione tramite Web Service. Questa soluzione mostra la notevole flessibilita' della nostra architettura e le sue quasi illimitate possibilita' di sviluppo.

Possible solution

- La tecnica del Web Service garantisce l'integrazione tra il nostro middleware (in Java) e i drivers (in C).



Real time data visualization in SL

- Quando l'oggetto virtuale riceve dati dal RW viene invocata la funzione 'llSetTexture()' su tutti i 'micro oggetti' che compongono lo schermo in modo da formare la stringa numerica ricevuta.
- Abbiamo reso possibile la visualizzazione dell'output dell'oscilloscopio reale dentro SL in quasi real time e anche la gestione dell'input da SL.

Riconoscimenti

- Partendo da quest'ultima parte del progetto abbiamo scritto un saggio: 'Experiential based learning in 3D Virtual Worlds: Visualization and data integration in Second Life' (Callaghan M.J., Harkin J., Sanfilippo F., Scibilia G., McCusker K., Wilson S.) che ha vinto il 'best paper award' alla REV Conference 2008 organizzata dalla University of Applied Sciences in Duesseldorf (Germany, June 2008).

Ringraziamenti

- Michael Callaghan della University of Ulster, Magee Campus
- Prof. Di Stefano, Ing. Morana
- I miei genitori, mio fratello, mia sorella, la mia ragazza
- Tutti i presenti per la cortese attenzione