

Peer-Review 2: UML Network

Martina Quaregna, Martina Riva, Andrea Pinessi, Alessandro Vai

Gruppo 24

Valutazione del diagramma UML riguardante la rete del gruppo 14.

Lati positivi

- Strategy pattern applicato nella scelta del tipo di connessione

Lati negativi

- Non è chiaro perché venga usato un thread all'interno dello schematico relativo alla rete (nel file nominato UML-MORE INFORMATION);
- Non abbiamo ben compreso come viene gestito il client al momento, probabilmente necessita ancora una vera e propria implementazione;
- Non esiste ancora un server generico che gestisce i client ma solo un'interfaccia e dei "server" RMI e Socket;
- Il Server RMI non possiede ancora un'interfaccia (necessaria per applicare il protocollo RMI);
- Sia "RMI_Strategy" che "SocketTcpStrategy_state" hanno come attributo una lista di indirizzi chiamata Registry, non siamo sicuri che questo tipo di soluzione rispetti il paradigma di RMI;
- Sequence diagrams non coerenti con l'UML (alcuni metodi richiedono parametri che nell'UML non sono specificati e nei sequence diagrams sono presenti classi assenti negli UML).

Confronto tra le architetture

Nell'UML del gruppo 14 non è ancora presente una vera propria struttura di rete (il client non è specificato e il network è solo parziale).

L'implementazione di RMI sembra più avanzata rispetto a quella di Socket, ma presenta comunque delle lacune.

L'UML del nostro gruppo presenta la divisione dei due protocolli gestita nel seguente modo:

- Il server e il client sono entrambi generici e in base alla scelta fatta dall'utente viene effettuata la chiamata ai metodi di client e server relativi al protocollo scelto;
- Per quanto riguarda RMI è presente un'interfaccia relativa al serverRMI che permette la connessione tra clientRMI e server;
- Per quanto riguarda Socket, invece abbiamo implementato una classe Connection che serve per lo scambio di messaggi tra clientSocket e Server.