

Peer-Review 2: Protocollo di rete

Giuliano Crescimbeni, Mauro Leporace, Luca De Nicola, Cristian Dheskali

Gruppo AM27

Valutazione del protocollo di rete del gruppo AM36.

Lati positivi

Strutturazione del protocollo ben fatta e definita per l'intera partita. I messaggi scambiati sono pochi ed essenziali e risultano leggeri e agevoli.

Lati negativi

L'UML, a prima vista, non risulta chiarissimo per mancanza di informazioni sugli attori, lato server e lato client, che prendono parte allo scambio di messaggi. Non è presente alcuna indicazione sulla tecnologia di rete impiegata e non è possibile dedurlo dal diagramma. Teniamo a sottolineare che le informazioni mancanti ora riportate sono state fornite successivamente ad una richiesta via mail da parte del nostro gruppo.

La carenza di una buona strutturazione "Object Oriented", per quanto riguarda la messaggistica, è giustificata dall'impossibilità di serializzare gli oggetti java sulla rete per via dell'implementazione del server in C++.

Confronto tra le architetture

Le due architetture risultano diverse su una serie di specifiche: L'architettura del gruppo AM36 non prevede l'implementazione della tecnologia RMI, per il motivo sopra citato. Il nostro gruppo utilizza un set di classi "Command" e "Update" per lo scambio di messaggi sulla rete, quindi tramite serializzazione di oggetti, a differenza del gruppo AM36 che utilizza stringhe ed interi per il passaggio di dati. Inoltre, da quanto appreso dalle informazioni aggiuntive fornite tramite mail, viene implementato il multipartita, e quindi l'indirizzamento degli eventi al controller corretto, tramite la presenza di VirtualView sul server, nel nostro caso gestito da una classe apposita.