

LABORATORIO DI INGEGNERIA DEI SISTEMI SOFTWARE

Introduction

L'oggetto di studio è un sistema formato da componenti che comunichino tra di loro con il meccanismo del ping pong

Requirements

Costruire un sistema software distribuito costruito da due qak che si inviino informazioni con il meccanismo del ping pong, ovvero, che ogni volta che uno dei due invia un messaggio all'altro, si aspetti al massimo una risposta. Inoltre, occorre che, una volta che uno dei due invii un messaggio, l'altro non invii altri messaggi.

Requirement analysis

1. I requisiti esprimono quale protocollo di comunicazione specifico sfruttano i componenti coinvolti
2. I requisiti esprimono quale linguaggio utilizzare: il linguaggio qak
3. I componenti sono enti computazionali attivi e autonomi

Problem analysis

Interazione logica

I requisiti specificano che la comunicazione tra i due componenti debba essere asincrona: in effetti, all'invio di un messaggio da parte di uno dei due componenti non è detto che vi sia una risposta da parte dell'altro. Tuttavia, se la risposta è presente, dev'essere una sola, e deve avvenire solamente dopo aver ricevuto un messaggio (ad eccezion fatta per il messaggio iniziale, che permette di inizializzare la comunicazione).

Architettura logica

I componenti vengono rappresentati come attori QAK: in questo modo questi saranno modellati come Automi a Stati Finiti, e questo permetterà di esprimere il problema utilizzando concetti di più alto livello, ovvero in termini di scambio di messaggi.

Test plans

Per poter predisporre un piano di test per questo tipo di applicazione si potrebbe pensare di far mandare un messaggio da uno o più Producer verso il Consumer, aggiungere un observer al

consumer e verificare che i messaggi vengano ricevuti correttamente.

Project

Si è deciso di implementare i messaggi scambiati come dispatch: in effetti, se fossero state usate le request, all'invio di un messaggio ogni componente si sarebbe aspettato una response. In ogni caso occorre fare attenzione affinché un componente (ad eccezione fatta di quando viene inviato il primo messaggio) invii un messaggio all'altro soltanto dopo averne ricevuto uno. Il componente che inizia la comunicazione nel suo stato iniziale dovrebbe inviare un messaggio all'altro e ad ogni messaggio ricevuto può inviarne un altro. Nell'implementazione fornita il primo messaggio inviato risulta essere il numero 0 e ad ogni messaggio successivo viene inviato il numero successivo rispetto al numero ricevuto. Quando un attore legge un numero sopra a 4 nel messaggio interrompe la comunicazione.

Testing

TODO

Deployment

Maintenance

By studentName email: gabriele.daga@studio.unibo.it,



GIT repo: <https://github.com/dagus01->

lab/issLab2024