LABORATORIO DI INGEGNERIA DEI SISTEMI SOFTWARE

Introduction

L'oggetto di studio è un sistema formato da a N Produttori che inviano informazione a 1 Consumatore

Requirements

Costruire un sistema software distribuito costituio da N (N>=1) Produttori che inviano informazione a 1 Consumatore, il quale deve elaborare tale informazione.La dislocazione dei componenti sui nodi di elaborazione può essere:

- OneNode: tutti i componenti operano nello stesso nodo;
- TwoNodes: gli N Produttori operano in uno stesso nodo, mentre il Consumatore opera in un diverso nodo;
- ManyNodes: il Consumatore opera in suo proprio nodo, mentre i Produttori operano su K nodi diversi (con N>=K>1)

Requirement analysis

1. I requisiti non esprimono quale protocollo di comunicazione specifico sfruttano consumatore e produttori
2. I requisiti non esprimono quale linguaggio specifico o tecnologia specifica utilizzare
3. Teoricamente, consumatori e produttori potrebbero anche sfruttare linguaggi diversi
4. Produttore e Consumatore sono enti computazionali attivi e autonomi

Problem analysis

Interazione logica

I requisiti non specificano se la comunicazione tra produttori e consumatore debba essere sincrona oppure asincrona. Se i producer dovessero aver bisogno di un feedback ad ogni invio da parte del consumer prima di continuare, allora sarebbe più opportuno sfruttare un protocollo sincrono (come HTTP), altrimenti si potrebbe valutare un protocollo asincrono

Architettura logica

A prescindere da come far avvenire la comunicazione, si potrebbe pensare di sfruttare un

modello in cui produttori e consumatori 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

Sono stati creati due producer e un consumer. Ciascuno dei due producer viene modellato come un actor con stato iniziale in cui invia due messaggi di dispatch e una request al consmer, e, successivamente, alla ricezione di una response, effettua una transizione di stato per poter ricevere la risposta alla request che è stata inviata. Il consumer invece viene modellato come attore con stato iniziale vuoto, da cui è possibile transitare in altri due stati: uno per ricevere dispatch, e uno per ricevere request e rispondere con opportuna response. Risulta evidente che lavorando con questo tipo di modello il focus dell'implementazione sia incentrato maggiormente sul comportamento delle entità coinvolte più che su caratteri prettamente implementativi.

Testing

Deployment

Maintenance

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

GIT repo: https://github.com/dagus01-lab/issLab2024