

Esame Laboratorio Software Engineering (AA 2021/22)

02 Febbraio 2022, ore 13.30,

Enrico Tronci

*Computer Science Department, Sapienza University of Rome
Via Salaria 113 - 00198 Roma - Italy*

tronci@di.uniroma1.it

<http://mclab.di.uniroma1.it>

Esercizio 2 (25 punti)

L'unità di tempo per questo esercizio è il secondo.

Si vuole realizzare un sistema di monitoraggio per un impianto industriale. Il monitoraggio consiste nel leggere periodicamente i valori dei sensori con l'obiettivo di rilevare anomalie.

Ci sono N sensori, numerati da 1 ad N .

L'ambiente **Env** per il software consiste di un server che riceve come input l'identificatore $i \in \{1, \dots, N\}$ di un sensore e ritorna la coppia (τ, v) dove: v è il valore (numero reale) riportato dal sensore al tempo τ (numero reale) in cui è stata ricevuta la richiesta.

Il tempo τ è semplicemente il valore della variabile Modelica **time** al momento in cui **Env** riceve l'interrogazione.

Ai fini del vostro modello, il valore v riportato da **Env** per il sensore i è scelto uniformemente a random nell'intervallo $[a(i), b(i)]$, dove:

1. $a(i) = i - 1$
2. $b(i) = i + 1$

Cioè il segnale dal sensore i ha valor medio i .

Il server **Env** è connesso al client che realizza la rilevazione di anomalie attraverso due fifo: una dal server al client (che ritorna al client i valori (τ, v)) e l'altra dal client al server (che ritorna al server l'identificatore i del sensore di cui il client vuole sapere il valore).

Ogni minuto (cioè 60 secondi) il client chiede al server i valori di tutti i sensori (uno per volta, poichè così prevede il protocollo).

Il client calcola la media degli ultimi W valori letti per ogni sensore e rileva un'anomalia se questa media è minore di $0.8 \frac{a(i)+b(i)}{2}$ o maggiore di $1.2 \frac{a(i)+b(i)}{2}$. Cioè, differisce per più del 20% dal vero valore atteso ($\frac{a(i)+b(i)}{2}$).

Questo esercizio si focalizza sulla modellazione del server **Env** di cui sopra. A tal fine si sviluppano i seguenti blocchi.

1. Blocco **Env** nel file **env.mo** che modella l'environment descritto sopra.
2. Blocco **Server2Client** nel file **fifos2c.mo** che modella la fifo dal server al client.

3. Blocco `Client2Server` nel file `fifoc2s.mo` che modella la fifo dal client al server.
4. Blocco `ClientMockup` nel file `mockup.mo` che contiene un mock-up del client descritto sopra. Il `ClientMockup` semplicemente manda richieste a random al server e riceve la risposta dal server. La sua funzione è quella di permettervi di testare il server.

NOTA BENE

1. Tutti i parametri del vostro modello devono essere contenuti nel record `Prm` nel file `parameters.mo`. Oltre a quelli menzionati nel testo dell'esercizio potete aggiugnere dei vostri parametri, ma non dovete in alcun caso rimuovere quelli che ci sono poichè vengono usati per la correzione.
2. Il blocco `Probe` nel file `probe.mo` prende come input tutti gli outputs dei blocchi del vostro modello. Questo file viene usato per la correzione del progetto.
3. Il modello `System` nel file `system.mo` deve essere esteso come serve, ma non devono essere rimosso il contenuto già presente poichè viene usato per la correzione.
4. Potete aggiungere file a vostra discrezione ed estendere a vostra discrezione il contenuto dei file che vi sono forniti.
5. Salvo esplicita istruzione in senso contrario, non potete modificare in alcun modo il contenuto già presente nei file che vi sono forniti. Questi vengono usati per interfacciarsi con gli script di correzione. Una modifica delle interfacce fornite rende impossibile la correzione e quindi l'esercizio riceverà 0 punti.
6. Prima di consegnare accertarsi che il vostro modello compili. I modelli che non compilano ricevono 0 punti.