

LABORATORIO DI INGEGNERIA DEI SISTEMI SOFTWARE

Introduction

Requirements

1. Si desidera realizzare una versione osservabile (SonarObservable) del dispositivo Sonar introdotto in Il Sonar che soddisfi i seguenti requisiti.
2. Il SonarObservable deve inviare informazioni a tutti componenti software interessati alla rilevazione dei valori di distanza.
3. Il SonarObservable deve fornire valori di distanza solo quando questi si modificano in modo significativo.
4. I componenti interessati ai valori di distanza prodotti dal SonarObservable sono denominati Observer e possono risiedere sullo stesso nodo del SonarObservable (cioè sul RaspberryPi) o su un nodo remoto (ad esempio sul PC).
5. Il funzionamento del SonarObservable deve essere testato in modo automatizzato ponendo un ostacolo a distanza fissa DTESTING1 davanti ad esso, controllando che tutti gli Observers ricevano il valore DTESTING1. Dopo un qualche tempo, si modifica la posizione dell'ostacolo a una nuova distanza DTESTING2 e si controlla che tutti gli Observers ricevano il valore DTESTING2.

Requirement analysis

1. Con versione osservabile del Sonar intendiamo un dispositivo Sonar che sia in grado di restituire la distanza da lui misurata tra se stesso e un oggetto data e che sia in grado di mostrarla in qualche modo a coloro che intendono visualizzarla. La transizione ad un Sonar osservabile prospettata in Il pattern observer può essere affrontata pensando il SonarObservable in due modi:
2. Il SonarObservable deve essere in grado di fornire i dati a tutti i componenti software interessati, cioè significa che tali componenti dovranno inviare al SonarObservable una richiesta di invio dati che questo riceverà e alla quale risponderà con l'invio dei suddetti dati.
3. Il SonarObservable deve inviare i dati misurati ai componenti interessati alla ricezione di questi solamente quando variano in modo significativo. Ma la variazione significativa di una distanza è opinabile e dipende dal limite imposto sonarObservable. Quindi appena la differenza tra la distanza appena registrata e la distanza comunicata precedentemente sarà maggiore di un valore detto "limit", il sonarObservable comunicherà la nuova distanza a tutti gli observers.
4. Indipendentemente dalla loro allocazione, gli Observers devono essere in grado di comunicare con il SonarObservable, perciò questo dovrà poter essere in grado di ricevere richieste ed inviare dati sia in modo locale che non.
5. Indipendentemente dal tipo di Observer, il SonarObservable dovrà essere in

grado di eseguire positivamente i test automatizzati, in primo luogo deve essere in grado di fornire il dato misurato agli Observers (sia locali che non) e questi devono poter visualizzare correttamente il dato. Modificando la distanza di un valore scelto e noto, tutti gli Observer dovranno ricevere e visualizzare correttamente il nuovo valore.

Problem analysis

La transizione ad un Sonar osservabile prospettata in "Il pattern observer" può essere affrontata pensando il SonarObservable in due modi:

1. Il SonarObservable è visto come una classe che si interfaccia al Sonar e legge i dati misurati da quest'ultimo. Gli observers si "registrano" al SonarObservable (direttamente o tramite configuratore) e tramite un metodo UPDATE() il SonarObservable gli invia i dati richiesti secondo le distanze significative da loro desiderate.
2. Il SonarObservable e il Sonar (inteso come classe) sono la stessa cosa. In questo caso sono gli Observers che, tramite funzione READ() leggono e recuperano i dati derivanti dal sonar (la porta di input dell'osservatore si sincronizza con la porta di output del sonar).

Cosa abbiamo a disposizione:

- Sensore HC-SR04
- SonarAlone.c
- Progetto radarSystem22.domain (in particolari ISonar e IDistance)

Problema:

- Come gli Observers vogliono visualizzare i dati -> Display, Console, PC, ...,?

Come formalizzare questo approccio nel caso del SonarObservable?

- Con quale tipo di protocollo comunicano gli Observer esterni con il SonarObservable?
- Con quale tipo di comunicazione SonarObservable e Observers comunicano tra loro? Direttamente o in modo mediato?
- Cosa vuol dire "Osservabile" -> per un Observer osservare un dato significa ricevere il dato.
- se siamo convinti che si debbano osservare i valori prodotti piuttosto che il dispositivo in sé, come formalizzare questo approccio nel caso del SonarObservable? Possiamo formalizzarlo definendo la seguente interfaccia:

```
public interface IMisura extends IDistance {  
    public void setValue(IDistance d);  
    public IDistance getDistance();  
}
```

Test plans

L'obiettivo dei test e' verificare che l'osservatore riceva e visualizzi correttamente i dati che gli invia il SonarObservable. Analogamente il SonarObservable deve leggere correttamente i dati prodotti dal Sonar.

Test 1:

Verifica che il dato prodotto dal sonar venga letto correttamente dal sonarObservable

Il progetto viene eseguito e il sonarObservable entra in funzione. Si pone un oggetto ad una distanza M dal sonar. Il sonar produce un dato che verra' mandato al nostro sonarObservable. Il test e' positivo se il dato ricevuto dal sonarObservable e' uguale al dato inviato dal nostro sonar.

Test 2:

Verifica che il sonarObservable legga correttamente il dato del Sonar e che lo mandi correttamente ad un singolo osservatore. In particolare, il dato ricevuto dall'osservatore dovra' ugualirsi con il dato prodotto dal sonar.

Il progetto viene eseguito ed entra in funzione il sonarObservable. Ad esso si registra un osservatore. Mettiamo un oggetto a distanza M dal sonar. Il sonar produce un dato che viene mandato al sonarObservable che lo memorizza. A questo punto l'osservatore, tramite il metodo UPDATE, riceve la distanza che il sonar ha mandato. Il test e' positivo SE il dato mandato al SonarObservable e' uguale a quello che noi vediamo altrimenti il Test e' fallito.

Test 3:

Verifica che il sonarObservable legga correttamente il dato del Sonar e riesca a fare l'update della distanza qualora essa sia cambiata sensibilmente. In particolare, il nuovo dato ricevuto dall'osservatore dovra' ugualirsi con il nuovo dato prodotto dal sonar.

Ammettendo che il test uno sia riuscito, abbiamo un oggetto posizionato a distanza M e l'osservatore e il sonarObservable hanno questa distanza memorizzata.

Spostiamo sensibilmente l'oggetto a distanza N. Il Sonar manda il nuovo dato al sonarObservable e confronta il nuovo dato con quello vecchio per capire se la distanza e' cambiata sensibilmente. Il test e' positivo SE il sonarObservable cambia il dato che ha memorizzato con il dato appena ricevuto.

Project

DEFINIZIONE DEL TIPO DI PROGETTO

Testing

DEFINIZIONE DEL PIANO DI TESTING

Deployment

SVILUPPO DEL PROGETTO

Maintenance

MANTENIMENTO DEL PROGETTO

By studentName: Mirko Garofalo email:



mirko.garofalo@studio.unibo.it

github: <https://github.com/MJG07/issLab2022>