

LABORATORIO DI INGEGNERIA DEI SISTEMI SOFTWARE

Introduction

Sonar domain pattern observer

Requirements

Si desidera realizzare una versione osservabile (SonarObservable) che soddisfi i seguenti requisiti:

- Il SonarObservable deve inviare informazioni a tutti componenti software interessati alla rilevazione dei valori di distanza;
- Il SonarObservable deve fornire valori di distanza solo quando questi si modificano in modo significativo;
- Per valori modificati in modo significativo intendiamo i valori che si discostano dall'ultimo valore rilevato al sonar (iniziale punto oscuro dei requisiti)
- 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);
- 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

Domande del committente:

- Cosa significa osservare un dispositivo? Significa osservare i valori prodotti dal dispositivo (nel caso del Sonar significa la distanza rilevata)
- Si può dire che il valore della distanza sia una proprietà del dispositivo di misura (cioè del Sonar)? No, indica un valore esterno rilevato.
- Come formalizzare questo approccio nel caso del SonarObservable? Mediante l'utilizzo di modelli, ovvero di un linguaggio comprensibile alla macchina. (Per esempio nel caso di java mediante l'utilizzo di classi e interfacce)

Problem analysis

Un sonar osservabile può essere implementato mediante il pattern observer. Il pattern observer può essere sviluppato in due modi diversi:

- Modo 1: SonarObservable che registra le entità observer che devono essere aggiornate e le notifica con l'aggiornamento del valore mediante il metodo update
- Modo 2: SonarObservable che comunica con una entità terza che a sua volta registra le entità observer che devono essere aggiornate e le notifica con l'aggiornamento del valore mediante il metodo update

Il secondo approccio risulta preferibile avendo un miglior disaccoppiamento dei compiti ed andando quindi a rispettare il "single responsibility principle"

UPDATE LEZIONE (05/04/2022)

A seguito della lezione è stata eliminata la prima versione dello observer mentre è stata modificata la seconda versione implementata (quella facente uso dell'elemento intermedio)

Dovendo osservare i valori prodotti dal dispositivo e non il dispositivo stesso è stata introdotta una interfaccia osservabile IDistanceMeasured.

Il sonar (interfaccia ISonar) quindi comunicherà i valori prodotti a IDistanceMeasured che a sua volta aggiornerà gli Observer (interfaccia ISonarObserver)

Questo porta a i seguenti vantaggi:

- Single responsibility Principle
- Maggiore disaccoppiamento

Test plans

Test plan sonar osservabile:

- 1) Viene posto un ostacolo a distanza fissa DTESTING1 davanti al sonar
- 2) Viene poi controllato che tutti gli Observer ricevano il valore DTESTING1.
- 3) Attendo qualche istante
- 4) Modifico la posizione dell'ostacolo a una nuova distanza DTESTING2
- 5) Controllo che tutti gli Observer ricevano il valore DTESTING2.

Project

Modo 1:

- 1) Costruisco la classe SonarObservable, implementando l'interfaccia ISonar.
- 2) Costruisco la classe SonarObservableMock e SonarObservableConcrete, estendendo SonarObservable.

- 3) Costruisco la classe SonarObserver, implementando l'interfaccia ISonarObserver.
- 4) Creo la classe di test TestSonarObservableMock

Modo 2:

- 1) Costruisco la classe SonarObservablePubSub, implementando l'interfaccia ISonar.
- 2) Costruisco la classe SonarObservableMockPubSub e SonarObservableConcretePubSub, estendendo SonarObservablePubSub.
- 3) Costruisco la classe SonarObserverBroker
- 4) Costruisco la classe SonarObserver, implementando l'interfaccia ISonarObserver.
- 5) Creo la classe di test TestSonarObservableMockPubSub

UPDATE LEZIONE (05/04/2022)

Partendo da come base il "modo 2" è stata implementata la seguente struttura:

- 1) costruisco l'interfaccia osservabile (mediante l'interfaccia IObservable) IDistanceMeasured, implementata dalla classe DistanceMeasured
- 2) costruisco la classe SonarObservable, implementando l'interfaccia ISonar, che produce i valori comunicando con DistanceMeasured
- 3) costruisco l'interfaccia ISonarObserver, implementata dalla classe SonarObserver, che si metterà ad osservare IDistanceMeasured
- 4) Creo la classe di test TestSonarObserverMock seguendo le richieste della analisi dei requisiti

Testing

Deployment

Maintenance

By Davide Guidetti email: davide.guidetti3@studio.unibo.it
Git:
https://github.com/ilmada99/issLab2022_DavideGuidetti

