

LABORATORIO DI INGEGNERIA DEI SISTEMI SOFTWARE

Introduction

SonarObservable

Requirements

Si desidera realizzare una versione osservabile (**SonarObservable**) del dispositivo Sonar 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;
- 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);

Requirement analysis

- Il SonarObservable deve poter inviare informazioni (la distanza misurata) a componenti software interessate, che chiameremo Observer.
- I processori che saranno interessati alla distanza idealmente potrebbero essere ad esempio il PC o il RaspberryPi.
- Gli Observer possono risiedere sullo stesso nodo del SonarObservable, o su un nodo remoto.
- Il SonarObservable deve inviare il valore della distanza misurata solo quando quest'ultimo si è modificato in modo significativo.
- La versione osservabile (SonarObservable) deve utilizzare il dispositivo Sonar introdotto e definito precedentemente.

- La notifica nel nuovo valore andrà effettuata solo nel momento in cui tale valore sarà significativo..

Problem analysis

La transizione ad un Sonar può essere affrontata pensando il **SonarObservable** in due modi:

- **Modo1:** risorsa che ad ogni cambiamento modifica il proprio stato ed invia una notifica agli observer sul nuovo stato (inviando una richiesta al Sonar, che gli fornisce un dato come risposta)
- **Modo2:** creazione di un oggetto implementato come risorsa osservabile che memorizza la distanza misurata e un processo che avrà il compito di aggiornare tale oggetto

Si ritiene preferibile la seconda, cioè optare per un processo che aggiorna l'oggetto senza notificare direttamente sul nuovo stato. Tramite tale scelta, oltre ai vantaggi di avere una risorsa osservabile, si potranno notificare i dati in modo da agevolare l'interoperabilità senza dover mandare l'informazione direttamente agli Observer ma saranno questi a richiedere tale informazione.

Test plans

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.

Project

Volendo impostare il Sonar come un dispositivo osservabile, introduciamo un nuovo contratto, che estende il precedente attraverso la creazione di un interfaccia **ISonarObserver** con relativa classe **SonarObserver** che la implementa.

Visto l'idea che il Sonar-observable sia un processo che aggiorna il valore di una distanza osservabile (distanza misurata) che implementa l'interfaccia **IDistanceMeasured** con relativa classe **DistanceMeasured** che la implementa.

Creato un nuovo model `SonarObservableModel` con relative classi concrete e Mock `SonarObservableConcrete` e `SonarObservableMock`.

La nascita del nuovo tipo di Sonar ci induce a introdurre nuovi metodi in DeviceFactory.

Testing

Deployment

Maintenance

By **Lorenzo Di Palma**

ISS repo: <https://github.com/lollofred/iss2022>

Contacts:

e-mail: lorenzo.dipalma2@studio.unibo.it

GitHub: lollofred

