

La Riflessione computazione

Elisa Ferrando

Cos' è la Riflessione

La Riflessione



Sistema riflessivo



Sistema computazionale

Sistema Riflessivo

E' un sistema computazionale dove:

E' causalmente connesso

le modifiche di comportamento si riflettono

Sistema Riflessivo

Cosa significa causalmente connesso

Esempio

Sistema real-time in cui esiste:

- Del software di controllo di processi esterni
- Software riflessivo

Sistema Riflessivo

- Contiene strutture che permettono di effettuare azioni su se stesso

In ogni istante deve conoscere:

- La propria rappresentazione interna
- Lo stato e la computazione del sistema

Architettura riflessiva

Dotare un linguaggio di architettura riflessiva significa:

- Il compilatore e l'interprete forniscono strumenti per gestire la riflessione
- Il compilatore garantisce il mantenimento della proprietà di connessione causale

Sistema Riflessivo

- E' strutturato in 2 o più livelli
- Un primo livello D_n
- Un secondo livello D_{n+1}
- Ogni livello è base per il successivo e meta-livello per quello inferiore

Sistema Riflessivo

- D_n è formato da entità scritte in linguaggio L_n e da un Interprete scritto in linguaggio L_{n+1} interpretato da un Interprete I_{n+1} e così via..
- $L_n = L_{n+1} = \text{Torre Riflessiva}$

Sistema Riflessivo

Connessione tra entità base e meta:

- Le meta entità supervisionano il lavoro delle entità base tramite intercettazione
- Ogni azione è catturata dalla meta entità che esegue delle operazioni

Sistema Riflessivo

Caratteristiche:

- Transparency
- Separation of concerns

Riflessione

2 Tipi:

- Strutturale
- Computazionale vera e propria

La Riflessione

Applicata principalmente ai linguaggi
object-oriented

Inizialmente applicata ai linguaggi
funzionali

Primi esempi

META-PROLOG

- Linguaggio logico basato sulle meta-teorie
- Dotato di interprete meta-circolare

SOAR

- Linguaggio basato sul protocollo a meta-regole
- Interprete meta-circolare

Riflessione: modelli

Per i linguaggi object-oriented abbiamo diversi modelli i principali sono:

- Meta- classe
- Meta-oggetto

Meta-classe

- Ogni entità base è legata ad una meta entità
- Ogni meta entità è una meta-classe
- Implementabile solo nei linguaggi che gestiscono le classi come oggetti

Smalltalk

Meta-classe

=

informazione oggetto e riflessiva

Meta-oggetto

- Ogni entità base è legata ad una o più meta entità
- I meta oggetti sono istanze di una classe speciale
- Il meta oggetto può eseguire delle elaborazioni prima dell'operazione chiamata sull'oggetto e dopo dare il controllo all'oggetto

Java

Le librerie `java.lang` e `java.lang.reflect` offrono la capacita' di ispezionare classi e oggetti

Java

Tramite le librerie possiamo:

- Scoprire metodi presenti in una classe
- Istanziare una classe
- Invocare metodi, ispezionare e cambiare il contenuto di campi che scopriamo a run-time

Java

La classe di partenza e' `Java.lang.Class`

Le istanze di `Class` rappresentano classi e interfacce di una applicazione Java

Permette di avere tutte le informazioni riflessive riguardo una classe:

Superclasse, interfacce implementate e anche costruttori, campi e metodi definiti dalla classe

Java

Alcuni metodi di Class:

- `getConstructors()`
restituisce un array con i costruttori pubblici della classe rappresentata dall'oggetto su cui si invoca. Il valore restituito è una istanza di `java.lang.reflect.Constructor`
- `getFields()`
in modo simile alla lista di costruttori, restituisce la lista dei campi pubblici dell'istanza di Class. Il valore di ritorno è una istanza di `java.lang.reflect.Field`

Java

- `getMethods()`
restituisce tutti i metodi pubblici di una classe.
Il valore di ritorno è di tipo `java.lang.reflect.Method`
- `getSuperclass()`
restituisce la `Class` che rappresenta la superclasse della `Class` su cui è invocata.
- `newInstance()`
restituisce una nuova istanza della classe di cui viene fornito il nome come parametro stringa.
Permette di istanziare una classe non conosciuta al momento della compilazione.

Java

Vantaggi:

- Fornisce un modo per collegare ad un programma nuove classi, non conosciute a compile time.
- Permette di manipolare oggetti di una qualsiasi classe senza inserire nel codice la classe, quindi rinviando il binding fino a runtime.

Java

Svantaggi:

- Invocare metodi o accedere a campi con i meccanismi riflessivi è molto più lento che col codice diretto. Ma se l'uso della riflessione all'interno di un programma è limitato, l'overhead è sostenibile.
- Capire cosa sta facendo un programma leggendo il codice diventa molto più difficile, poiché il codice è più complesso.

Applicazioni della riflessione

- Calcolare statistiche sul numero e tipo di operazioni eseguite da un programma
- Debugging
- Sistemi di apprendimento
- Sistemi operativi