

Mini caso di studio relativo alla progettazione e alla realizzazione di un piccolo robot in grado di compiere missioni di navigazione autonoma, evidenziando come la progettazione ad oggetti sia applicabile alla robotica e l'intelligenza artificiale

Navigazione autonoma di robot: dalla progettazione object-oriented alla robotica

Introduzione

Imparare a progettare non richiede solamente l'acquisizione di solidi principi di ingegneria del software: richiede soprattutto esperienza. Esperienza per "saper fare". Ma cosa significa "saper fare"? Vuol dire aver già realizzato uno o più sistemi simili? Vuol dire conoscere un linguaggio di programmazione e un ambiente operativo sufficientemente da saper tradurre in codice un algoritmo? La mia personale risposta è no! In tutte le aziende che sviluppano software come attività principale questi risultati sono sempre in un qualche modo largamente acquisiti. Eppure i progetti continuano ad essere in crisi. Saper fare e, per estensione saper progettare, deve allora significare qualcosa di più. Un progettista esperto si costruisce nel tempo una specie di "cassetta degli attrezzi" composta da tecniche, principi di ingegneria, euristiche, ma anche intuito e convenzioni stilistiche (il lato più creativo di questo lavoro) che risultano utili al fine di:

- saper riconoscere i *sintomi* tipici che richiedono attenzione in un progetto software (come l'effetto domino causato da una modifica, oppure la difficoltà di aggiungere un nuovo concetto al sistema senza dover (ri)scrivere, modificare o rivedere un numero eccessivo di classi);
- saper agire sulle *vere cause* di tali problemi (scarsa coesione, elevato livello di accoppiamento, scarsa integrità concettuale dell'architettura e/o di un componente; complessità accidentale, ...);
- saper valutare approcci alternativi (metriche, trade-off, ...);
- avere in primo luogo delle alternative! (pattern di analisi, di design e di architettura, idiomi di programmazione);
- avere degli strumenti utili per ragionare sul sistema, soprattutto in chiave di manutenzione e di estensione (diagrammi UML espressivi, design basato sulle responsabilità, ...).

Dalla consapevolezza che "sapere la teoria" di per sé non significa automaticamente saperla applicare, nasce l'idea di lavorare ad una serie di progetti pilota la cui principale ambizione consiste nel mostrare *come* e *dove* attuare determinate tecniche. Tali progetti sono quindi da intendersi come piccole palestre, mini casi di studio, la cui dimensione e complessità strutturale è però sufficiente ad evidenziare applicazioni non banali di design del software. Come primo progetto ho scelto la progettazione e la realizzazione di un piccolo robot in grado di eseguire missioni di navigazione autonoma (eh eh, mi ero annoiato delle solite applicazioni di gestione del magazzino!). A parte gli scherzi, una delle sfide che ho voluto raccogliere in questo primo progetto consiste nell'applicare la progettazione ad oggetti in un'area in cui i sistemi sono stati storicamente costruiti (e vengono spesso ancora costruiti) secondo metodologie tradizionali.

A chi dunque è principalmente rivolto questo lavoro? Il taglio degli articoli è pensato per risultare accessibile a qualsiasi progettista del software che abbia una conoscenza di base della programmazione ad oggetti, senza richiedere particolari prerequisiti di intelligenza artificiale o di robotica. Le prime parti del progetto forniscono tutto il background teorico necessario per acquisire i concetti fondamentali che servono per implementare gli algoritmi e per capire il design. Ovviamente il materiale può essere utile anche ad una audience più allargata, comprendendo (a maggior ragione) professionisti che lavorano nel settore della robotica e dell'intelligenza artificiale.

Il progetto è idealmente suddiviso nelle seguenti quattro parti:

1. descrizione del progetto;
2. definizioni e concetti di base;
3. design: architettura e sottosistemi;
4. sistema finale.

L'intento principale che mi propongo di raggiungere in questo lavoro è sicuramente di tipo didattico: se dovrò esserci un trade-off tra efficienza e facilità di comprensione, verrà sempre preferita quest'ultima. A parte tale osservazione, il progetto può essere considerato abbastanza realistico (di fatto, alla fine, avremo un piccolo sistema funzionante!). Ovviamente lo sforzo richiesto per realizzare e soprattutto documentare tutte le fasi di progetto e sviluppo, anche considerando le dimensioni piuttosto limitate, richiederà un tempo piuttosto lungo. Ritengo tuttavia che, allo stesso modo in cui vedere (e scrivere) molti programmi risulta utile per imparare a programmare, vedere diversi buoni modelli è utile per imparare a progettare. Vedere poi anche il risultato della modellazione in un sistema finale "vivo", rafforza ulteriormente la padronanza di una metodologia complessa come quella orientata agli oggetti.

Prima parte: descrizione del progetto

L'obiettivo del progetto è quello di costruire un semplice robot in grado di eseguire alcune missioni di navigazione autonoma. Ogni missione richiede al robot di compiere una serie di azioni per raggiungere uno scopo prefissato (ad esempio partire da un punto in una stanza e raggiungere un altro punto nella medesima stanza, identificando un percorso tra i due punti, senza collidere con eventuali ostacoli). La navigazione non è supervisionata dall'utente, quindi il robot deve poter interagire con l'ambiente esterno per identificare eventuali ostacoli e reagire di conseguenza.

Durante lo sviluppo del progetto saranno discusse le seguenti tematiche:

- Che cos'è un robot e che cosa rende un robot intelligente;
- Quali sono i paradigmi di programmazione tipici di un robot;
- Come la progettazione ad oggetti ha influenzato la robotica e l'intelligenza artificiale applicata alla robotica;
- Quali sono le principali architetture e le relative metriche di valutazione nel design di robot intelligenti;
- Come progettare e realizzare un robot in grado di eseguire delle missioni di navigazione autonoma

Un vero robot tuttavia non è composto solamente da software: sono necessarie anche delle componenti hardware per assemblare lo chassis, gli organi di locomozione che permettono al robot di muoversi e gli organi sensori che permettono di percepire l'ambiente circostante. Per questo progetto è stato scelto un particolare tipo di hardware il cui costo e la cui reperibilità bilanciano le inerenti limitazioni: i LEGO Mindstorms. In particolare, il progetto si basa sul kit LEGO Mindstorms RIS (Robotics Invention System) 2.0 [Bagnall, 2002].

Bibliografia

[Bagnall, 2002] Bagnall, Brian – “*Core Lego Mindstorms Programming*”, Prentice Hall PTR, 2002