

# Un Approccio Innovativo alla Programmazione di Robot Collaborativi

Sara Beschi, Daniela Fogli, Giovanni Guida, Fabio Tampalini

Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Brescia  
s.beschi005@studenti.unibs.it, {daniela.fogli, giovanni.guida, fabio.tampalini}@unibs.it

## Abstract

Una delle tecnologie abilitanti previste nel programma Industria 4.0 è costituita dai robot collaborativi. La programmazione di questi ultimi da parte di operatori non esperti di informatica è una delle principali sfide che interseca i campi dell'Intelligenza Artificiale e dell'Interazione Uomo-Macchina. Questo articolo presenta un approccio innovativo basato sul riconoscimento del linguaggio naturale e su un linguaggio di programmazione visuale per consentire a operatori non esperti di informatica di acquisire velocemente le abilità necessarie per la definizione di compiti per robot collaborativi di tipo pick-and-place.

## 1 Introduzione

I robot collaborativi sono robot industriali che condividono lo spazio di lavoro con gli operatori umani aiutandoli in attività ripetitive e faticose [Schou et al., 2018]. Questi robot devono dunque essere progettati per lavorare in collaborazione con gli operatori in maniera sicura e attraverso interfacce utente intuitive [Villani et al., 2018]. Grazie ai finanziamenti in ambito Industria 4.0, le piccole e medie imprese stanno sempre più introducendo robot collaborativi negli ambienti di lavoro per automatizzare compiti che non sarebbe conveniente affrontare con gli approcci tradizionali della robotica industriale. In questo contesto, uno dei benefici fondamentali che i robot collaborativi sono in grado di apportare è la flessibilità con cui è possibile modificare le attività che il robot è chiamato a svolgere, bilanciando fra le esigenze di automazione e quelle di una produzione fortemente variabile e caratterizzata da lotti di piccole o piccolissime dimensioni. Questo comporta tuttavia che gli operatori siano in grado di provvedere personalmente, senza il supporto di tecnici specializzati, a programmare i robot ogni volta che occorre definire un nuovo compito o modificare le modalità esecutive di un compito già definito. Pertanto, lo studio e la progettazione di interfacce operatore intelligenti e facili da usare per la programmazione di robot collaborativi da parte di utenti esperti del loro dominio applicativo ma non di robotica e programmazione diventa fondamentale.

In questo articolo illustriamo il sistema CAPIRCI (*Chat And Program Industrial Robots through Convenient Interaction*).

L'idea del sistema nasce da un lato dall'osservazione delle modalità di interazione con i sistemi utilizzate quotidianamente dagli utenti, come le chat per lo scambio di messaggi, e dall'altro dal successo osservato in letteratura dei linguaggi di programmazione basati sulla metafora del puzzle. Fra questi ultimi, Scratch, Alice e AppInventor rappresentano casi di successo impiegati nelle scuole di tutto il mondo per insegnare a bambini e adolescenti i concetti chiave della programmazione e del pensiero computazionale in un contesto coinvolgente e ricreativo [Franklin et al., 2017]. A questi sistemi si sono ispirati alcuni ricercatori che hanno affrontato il tema delle interfacce utente per la programmazione di robot collaborativi. Ad esempio, CodeBlox [Weintrop et al., 2018] è un'applicazione software il cui linguaggio visuale è basato su Blockly, lo stesso ambiente su cui si basano i sistemi sopra menzionati. CodeBlox permette agli utenti di programmare le operazioni di basso livello che vanno a costituire azioni di "pick" e "place" per un robot collaborativo. Similmente Code3 [Huang and Cakmak, 2017] prevede un'interfaccia visuale molto simile, basata anch'essa su Blockly, e altre due componenti per la creazione di blocchi di tipo "azione" e "oggetto" attraverso una tradizionale interazione basata su programming-by-demonstration [Argall et al., 2009].

## 2 Il sistema CAPIRCI

Il sistema CAPIRCI si compone di due parti principali in grado di interagire fra di loro. Una prima componente è a tutti gli effetti una chat in cui il sistema e l'utente dialogano con lo scopo di generare un programma per il robot. Funzionalità di riconoscimento del linguaggio naturale orientate alla comprensione di compiti basati su operazioni di pick-and-place, che coinvolgono oggetti e operazioni che rientrano nel dominio semantico e pragmatico del robot, consentono a un utente non esperto di informatica di creare semplici programmi (Fig. 1). La chat è inoltre collegata a un'interfaccia grafica a cui rimanda per la visualizzazione, la verifica, la modifica e l'eventuale estensione del programma creato (Fig. 2).

Un utente esperto di programmazione o che già ha dimestichezza con l'interfaccia grafica può tuttavia decidere di usare direttamente questo strumento anche per creare programmi complessi che includono diversi tipi di controlli (come cicli e istruzioni condizionali).

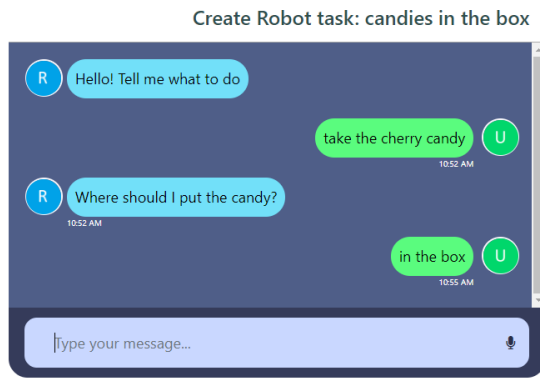


Figura 1: L'utente dialoga col sistema al fine di definire un compito per il robot collaborativo.

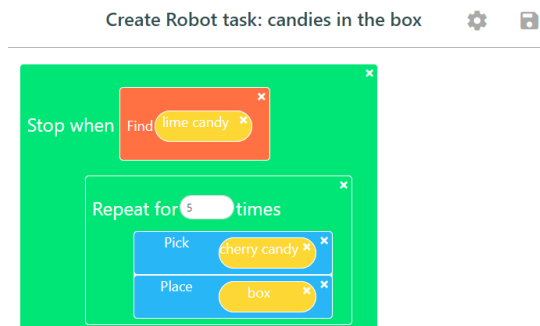


Figura 2: Il programma creato con la chat viene mostrato nell'interfaccia grafica.

L'interazione avviene con un semplice drag-and-drop dei diversi tipi di blocchi dalle librerie presenti in un menu predefinito (Fig. 3).

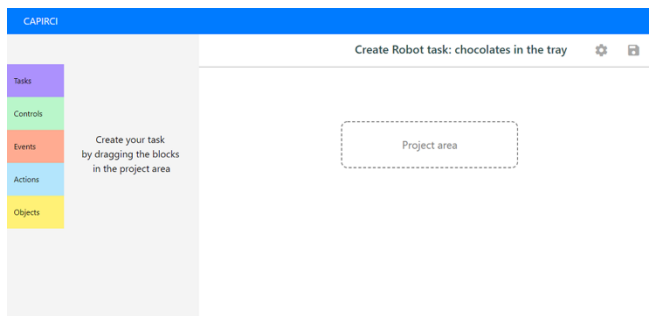


Figura 3: L'interfaccia grafica prima che l'utente inizi a creare un nuovo programma.

Il sistema interpreta le diverse azioni dell'utente ed è in grado di impedire combinazioni di blocchi errate e di suggerire soluzioni in caso di errori, consentendo quindi una programmazione visuale basata su vincoli (Fig. 4).

L'applicazione è sviluppata in Python e sfrutta librerie esistenti per il riconoscimento del linguaggio naturale (Natural Language Toolkit e Stanford Parser). Il programma generato è codificato in un file XML che può infine essere facilmente

tradotto nelle istruzioni di uno specifico linguaggio di programmazione per il robot collaborativo considerato.

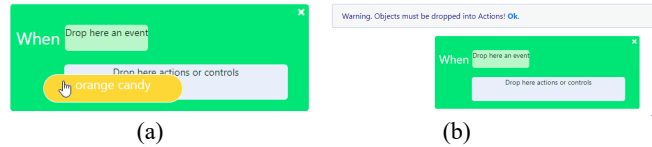


Figura 4: (a) L'utente cerca di posizionare un oggetto in un controllo invece di un'azione per il robot; (b) il sistema impedisce l'operazione indicandone il motivo.

### 3 Conclusioni

CAPIRCI è stato valutato mediante una sperimentazione con 20 utenti, di cui 10 esperti di programmazione e 10 privi di esperienza e di conoscenza relativa ai linguaggi di programmazione e alla creazione di programmi per robot. Agli utenti è stato chiesto di eseguire 5 compiti di complessità crescente riguardanti la programmazione di un robot collaborativo ipotizzando il suo utilizzo in una linea di produzione di una fabbrica di dolci.

Il test ha consentito di raccogliere sia dati quantitativi (tempi di esecuzione, numero di errori, percentuale di completamento dei compiti) che dati qualitativi, provenienti dall'osservazione diretta, dai commenti spontanei degli utenti e dal questionario sottomesso agli utenti alla fine del test.

I risultati sono molto incoraggianti: in particolare, la differenza fra i tempi medi di programmazione degli utenti esperti e di quelli non esperti non risulta statisticamente significativa per alcun task. Alcuni utenti non esperti di programmazione hanno commesso errori nel penultimo compito, che implicitamente richiedeva di specificare un ciclo al cui interno doveva essere inserita un'istruzione condizionale; tuttavia, tutti gli utenti sono poi riusciti a completare l'ultimo compito, denotando che il sistema risulta facile da apprendere.

Diversi utenti hanno espresso apprezzamenti per l'approccio misto, sottolineando che con la chat è possibile produrre una prima bozza di programma che poi può essere verificata e modificata facilmente per mezzo dell'interfaccia grafica, eventualmente risolvendo gli errori commessi.

Gli utenti hanno anche suggerito diversi miglioramenti a entrambe le componenti di CAPIRCI, come ad esempio l'auto-completamento nella chat o un'interazione alternativa al drag-and-drop nel caso dell'interfaccia grafica. Altre funzionalità sono già allo studio per l'estensione futura dell'applicazione.

### Riferimenti bibliografici

[Argall et al., 2009] Argall, B. D., Chernova, S., Veloso, M., Browning, B. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5), 469-483, 2009.

[Franklin et al., 2017] Franklin, D., Skifstad, G., Rolock, R., Mehrotra, I., Ding, V., Hansen, A., Weintrop, D., Harlow, D. Using Upper-Elementary Student Performance to Understand Conceptual Sequencing in a Blocks-based Curriculum. *In Proceedings of SIGCSE '17*, 231-236, 2017.

- [Huang and Cakmak, 2017] Huang, J., Cakmak, M. Code3: A System for End-to-End Programming of Mobile Manipulator Robots for Novices and Experts. In *Proceedings of HRI '17*, 453-462, 2017.
- [Schou et al., 2018] Schou, C., Andersen, R. S., Chrysostomou, Bøgh, S., Madsen, O. Skill-based instruction of collaborative robots in industrial settings. *Robotics and Computer Integrated Manufacturing* 53, 72-80, 2018.
- [Villani et al., 2018] Villani, V., Pini, F., Leali, F., Secchi, C. Survey on human-robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* 55, 248-266, 2018.
- [Weintrop et al., 2018] Weintrop, D., Afzal, A., Salac, J., Francis, P., Li, B., Shepherd, D. C., Franklin, D. Evaluating CoBlox: A Comparative Study of Robotics Programming Environments for Adult Novices. In *Proceedings of CHI '18*, Paper 366, 12 pages, ACM, 2018.