

# Standard Iec 61499 e simulazione di Dcs

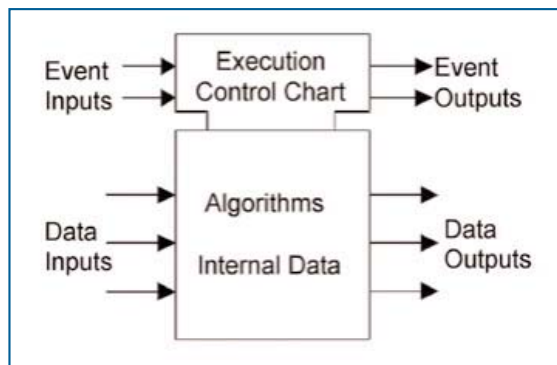


Figura 1 - Modello del blocco funzionale

Emanuele Carpanzano, Andrea Ballarino

Nel presente lavoro si adotta il modello funzionale per i sistemi di controllo proposto dallo standard Iec 61499, e si sfruttano le capacità di simulazione di due toolbox di Matlab, Simulink e Stateflow, per simulare tale modello. La descrizione delle soluzioni di controllo risulta semplificata e il loro riutilizzo è favorito dall'uso di modelli formali che sfruttino i concetti di modularità, incapsulamento e astrazione. L'interoperabilità e l'intercambiabilità sono garantite dall'adozione di uno standard internazionale; le soluzioni di controllo sono verificate tramite simulazione.

Per affrontare compiutamente il progetto di moderni sistemi di automazione, è necessario definire linee guida, metodi e strumenti che permettano di rappresentare sistemi di automazione complessi in maniera semplice e sicura, di mantenere la tracciabilità lungo le differenti fasi di progetto, e di avere la necessaria conoscenza sul comportamento del sistema definito [1, 2]. La modularità, sia in fase di progetto sia in fase di verifica, è un elemento cruciale per affrontare la complessità insita nei sistemi di controllo industriali.

In particolare, la facilità di riutilizzo e di riconfigurazione di soluzioni di automazione già sviluppate risulta accresciuta attraverso l'adozione di modelli formali modulari, riducendo i costi e i tempi richiesti per il progetto e la realizzazione di nuovi sistemi di produzione, o per la modifica di sistemi esistenti [3]. Una metodologia di progetto formale e strutturata è necessariamente basata su modelli di riferimento formali, che devono essere consistenti in tutte le fasi del ciclo di vita del progetto [4]. Inoltre, l'uso di standard internazionali garantisce l'interoperabilità e l'intercambiabilità delle soluzioni di controllo definite in progetti diversi [5]. Per raggiungere tali obiettivi nel presente lavoro viene adottato il modello proposto dallo standard Iec 61499, il quale definisce blocchi funzionali per sistemi di controllo e supervisione di impianti industriali [6]. Inoltre, l'impiego di tecniche basate sulla simulazione per la valutazione e la verifica di soluzioni di automazione è oggi sempre più

frequente nel mondo industriale [7, 8, 9]. Nell'ambito del presente lavoro saranno usati due toolbox di Matlab - Simulink e Stateflow - per simulare i modelli considerati [10, 11]. Quindi, mentre lo standard Iec 61499 fornisce un metodo adatto a modellare sistemi di controllo complessi, la simulazione su computer in Simulink/Stateflow è lo strumento per utilizzare tali modelli al fine di valutare le prestazioni del sistema [12]. L'articolo è strutturato come segue. Dapprima, sono illustrate le principali caratteristiche dello standard Iec 61499 e dei pacchetti Simulink e Stateflow. Quindi, vengono definite regole per rappresentare un modello Iec 61499 in Simulink/Stateflow. I metodi e gli strumenti presentati sono applicati al progetto e alla verifica del sistema di controllo per una cella manifatturiera di un impianto industriale. Infine, vengono tratte alcune considerazioni conclusive sui risultati raggiunti e sui possibili sviluppi futuri.

## Lo Standard Iec 61499

Lo standard Iec 61499 definisce i blocchi funzionali per sistemi di controllo e supervisione di impianti industriali [6]. Tale standard viene qui introdotto per supportare il progettista nello sviluppo di modelli dettagliati del software del sistema di automazione, e nella distribuzione delle funzioni di controllo e supervisione sui dispositivi di automazione. Lo standard Iec 61499 è basato su un modulo fondamentale: il blocco funzionale (Fb, Function Block),

E. Carpanzano, ricercatore, Istituto di Tecnologie Industriali e Automazione, Consiglio Nazionale delle Ricerche, Milano, e.carpanzano@itia.cnr.it. A. Ballarino, R&D, Torielli rag. Pietro & C. SpA, Vigevano (PV), research@torielli.com.

il quale rappresenta un'unità funzionale software, associata ad una risorsa hardware del sistema di controllo. Come mostrato in Figura 1, un'istanza di un Fb è caratterizzata da: nomi di tipo e di istanza, insiemi di eventi in ingresso/uscita e dati in ingresso/uscita, dati locali, un Execution Control Chart (Ecc), costituito da stati, transizioni e azioni, il quale invoca l'esecuzione di algoritmi in risposta a eventi in ingresso; un insieme di algoritmi, associati agli stati dell'Ecc. L'esecuzione degli algoritmi è invocata dall'Ecc di un Fb, in risposta ad eventi in ingresso. Quando l'esecuzione di un algoritmo viene pianificata, i valori degli ingressi e dei dati locali necessari sono letti e possono essere computati i valori nuovi da assegnare alle uscite ed ai dati locali. Inoltre, al completamento dell'esecuzione di un algoritmo, l'Ecc può generare zero o più eventi in uscita. Collegando tra loro in modo adeguato più Fb, si definisce un'applicazione.

In dettaglio, l'esecuzione di un algoritmo associato all'istanza di un blocco funzionale è invocata tramite una richiesta di esecuzione delle operazioni previste dall'algoritmo fatta alla funzione di pianificazione della risorsa da parte dell'Ecc di un Fb. Il meccanismo di gestione degli eventi in ingresso a un Ecc è illustrato tramite l'automa di Figura 2 e la Tabella 1, mentre il meccanismo di evoluzione dell'Ecc e di invocazione dell'esecuzione degli algoritmi è definito secondo l'automa di Figura 3 e la Tabella 2. Si osservi che la valutazione della condizione associata a una transizione dell'Ecc è disabilitata fintantoché gli algoritmi associati allo stato precedentemente attivo non abbiano completato la loro esecuzione. Inoltre, occorrenze multiple di un evento allo stesso ingresso possono essere perse nella transizione t2 in Figura 2. Il rilevamento e il trattamento di tale perdita sono caratteristiche dipendenti dalla specifica implementazione.

Per quanto concerne la configurazione, un'applicazione può essere distribuita su più dispositivi di controllo. Un dispositivo utilizza la relazione causale specificata dall'applicazione per determinare le risposte appropriate agli eventi. Inoltre, nello standard Iec 61499 una risorsa è considerata come una suddivisione logica all'interno della struttura software (e possibilmente hardwa-

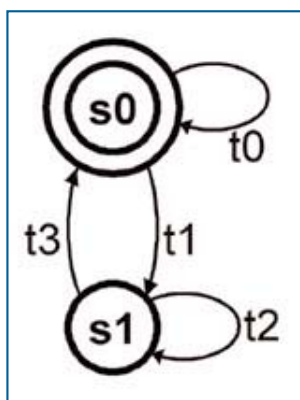


Figura 2 - Automa per la gestione di un evento in ingresso

Tabella 1 - Significato stati e transizioni dell'automa di Figura 2

Stato	Condizione
s0	in attesa dell'evento
s1	in attesa della terminazione dell'Ecc

Trans.	Condizione	Operazione
t0	mappatura ingressi	nessuna
t1	arriva un evento	richiesta di invocazione dell'Ecc
t2	arriva un evento	dipendente dall'implementazione
t3	mappatura ingressi	nessuna

re) di un dispositivo, dotata del controllo indipendente delle proprie operazioni. Secondo lo standard, ciascuna istanza di Fb è associata ad una singola risorsa. Basandosi su tali definizioni, l'architettura di un sistema di automazione industriale può essere modellata come una collezione di dispositivi, divisi in risorse, interconnessi e comunicati tra loro tramite una più reti di comunicazione. Mentre le funzioni eseguite da tale sistema sono modellate come applicazioni, costituite da reti di Fb allocati sulle risorse. L'adozione dello standard Iec 61499 come modello di riferimento formale favorisce la definizione di modelli riutilizzabili, dal momento che sono profondamente utilizzati i principi di modularità, incapsulamento e standardizzazione delle interfacce [4, 8].

### I toolbox Simulink e Stateflow

Simulink e Stateflow sono due toolbox dell'ambiente Matlab che supportano l'analisi di sistemi dinamici tramite la simulazione dinamica a tempo continuo ed ad eventi discreti. In particolare, Simulink supporta la modella-

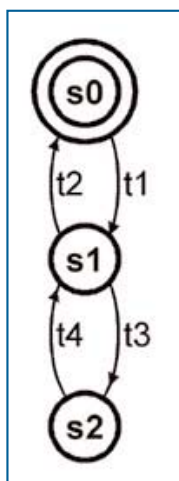


Figura 3 - Automa per l'evoluzione dell'Ecc

Tabella 2 - Significato stati e transizioni dell'automa per l'evoluzione dell'Ecc

Stato	Condizione
s0	pronto
s1	pianificazione algoritmi
s2	in attesa del completamento dell'algoritmo

Trans.	Condizione	Operazioni
t1	invocazione dell'Ecc	set variabili Ei, conferma mapp. ingressi, valut. transizioni
t2	nessuna transizione scatta	emissione eventi
t3	una transizione scatta	pianificazione algoritmo
t4	algoritmo completato	reset variabili Ei, set variabili Eo, valut. transizioni

zione e la simulazione di sistemi dinamici lineari e non lineari a tempo continuo ed a tempo discreto [10]. I modelli sono costruiti in un ambiente grafico tramite diagrammi causali a blocchi, utilizzando componenti disponibili in un insieme di librerie di segnali di ingresso, di componenti per la visualizzazione, di elementi lineari e non lineari, e di connettori. Inoltre,

l'utente ha la possibilità di creare nuovi blocchi e di personalizzare blocchi esistenti, organizzandoli eventualmente in nuove librerie. Un modello dinamico può essere costruito descrivendo il suo comportamento tramite equazioni matematiche o funzioni di trasferimento. Simulink supporta inoltre il concetto di gerarchia a più livelli, per esempio un modello può

essere diviso in sottosistemi a diversi livelli di aggregazione: è perciò possibile strutturare modelli in modo modulare, e riutilizzare parzialmente o completamente modelli già definiti. Si possono quindi effettuare simulazioni scegliendo il metodo di integrazione numerica, e specificando il grado di accuratezza desiderato. È possibile scegliere la simulazione in modalità interattiva o batch, per studiare specifici andamenti nel tempo o verificare automaticamente determinati casi di prova. I risultati della simulazione possono essere monitorati tramite appositi blocchi di visualizzazione grafica. D'altro canto, Stateflow è un ambiente grafico per il progetto strutturato di sistemi ad eventi discreti complessi, per esempio, sistemi per la supervisione e il controllo logico. Utilizzando Stateflow l'utente crea modelli a eventi basati su una rappresentazione grafica di una macchina a stati finiti, dove stati e transizioni sono i blocchi basilari costituenti il sistema. La caratteristica principale di tale toolbox è l'uso integrato di diagrammi di flusso e di tipiche notazioni "stato transizio-

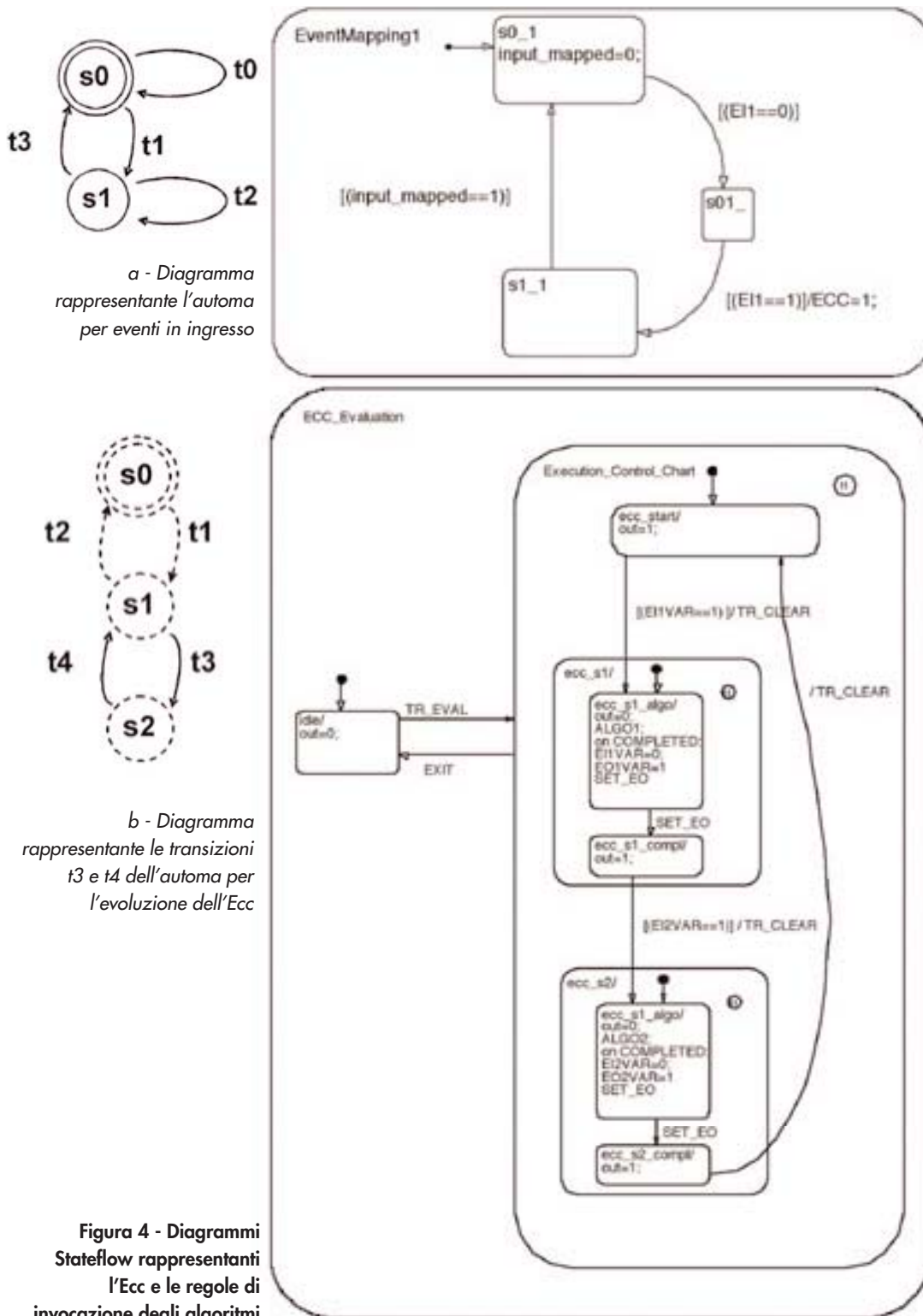


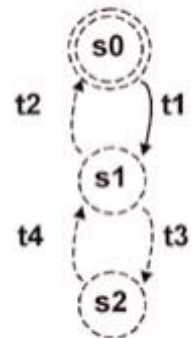
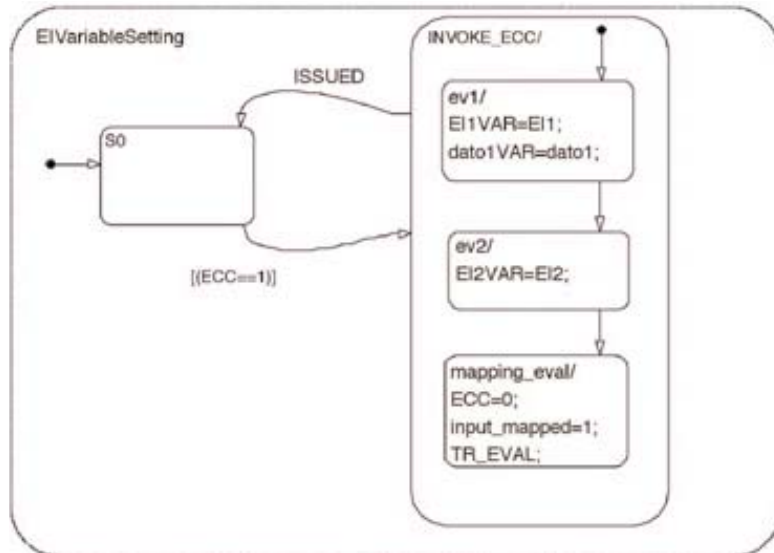
Figura 4 - Diagrammi Stateflow rappresentanti l'Ecc e le regole di invocazione degli algoritmi

ne” nella costruzione del modello Stateflow. Il primo formalismo permette di rappresentare efficacemente costrutti tipici della programmazione strutturata come cicli *for* e rami decisionali *if-then-else* usando *stateless decision points*. Il secondo formalismo è invece basato su una variante della teoria delle macchine a stati finiti, ovvero gli Statechart [11]. Tale formalismo sfrutta i concetti di gerarchia (permettendo di organizzare sistemi complessi definendo strutture padre-figlio), parallelismo (più stati indipendenti possono essere attivi contemporaneamente) e cronologia (l’evoluzione dello stato è funzione dell’informazione passata), semplificando così la modellazione di sistemi complessi ad eventi. I diagrammi Stateflow possono essere inseriti e simulati entro schemi Simulink, modellando così sistemi ibridi, caratterizzati cioè da dinamiche sia nel dominio del tempo che degli eventi discreti [1].

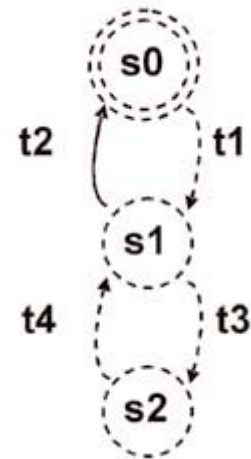
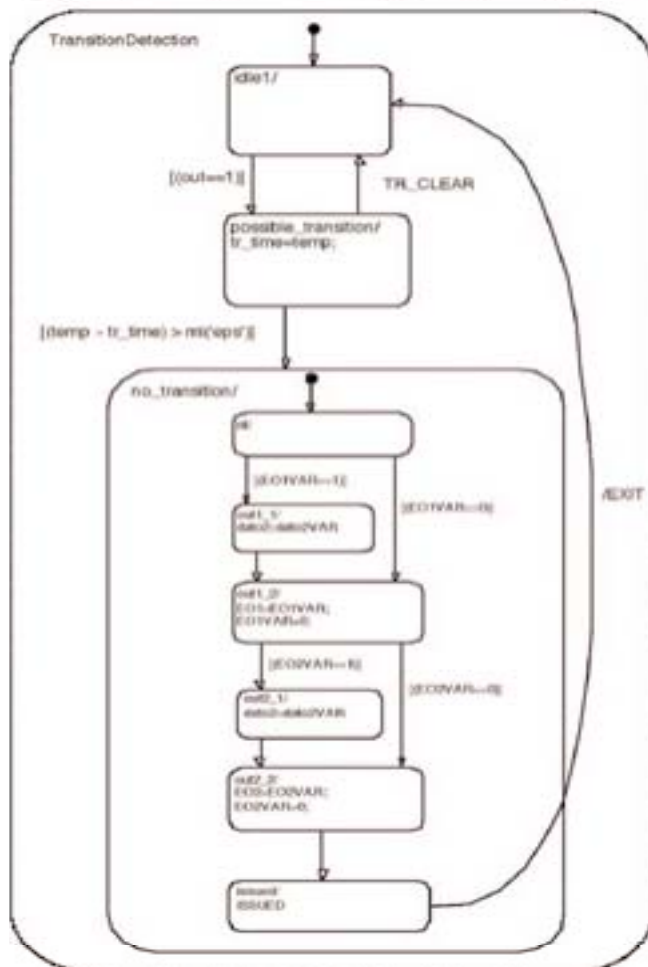
**Traduzione dal modello Iec 61499 a modelli Simulink/Stateflow**

La descrizione Simulink/Stateflow di un modello Iec 61499 è ottenuta modellando ogni Fb tramite un blocco Simulink, e connettendo adeguatamente tali blocchi secondo la struttura del modello Iec 61499 considerato [12]. In particolare, un singolo Fb è rappresentato attraverso un blocco Simulink costituito da un modello Stateflow, ove:

- i dati in ingresso e uscita sono rappresentati come segnali di ingresso e uscita di tipo double nel corrispondente blocco Simulink;
- gli eventi in ingresso e uscita sono rappresentati come fronti di salita di segnali di ingresso e uscita di tipo Booleano nel corrispondente blocco Simulink;



c - Diagramma rappresentante la transizione t1 dell’automa per l’evoluzione dell’Ecc



d - Diagramma rappresentante la transizione t2 dell’automa per l’evoluzione dell’Ecc

**Figura 4 - Diagrammi Stateflow rappresentanti l’Ecc e le regole di invocazione degli algoritmi**

- l'Ecc di un Fb è rappresentato tramite diagrammi Stateflow eseguiti in parallelo, come di seguito in dettaglio;
- gli algoritmi sono rappresentati tramite diagrammi Stateflow, dove possono essere eventualmente chiamate funzioni di Matlab;
- i dati locali possono essere rappresentati tramite variabili Simulink locali di tipo double.

In particolare, per modellare correttamente l'Ecc e le regole di invocazione degli algoritmi illustrate nel paragrafo dedicato allo standard Iec 61499, vengono introdotti differenti diagrammi Stateflow, i quali rappresentano la mappatura degli ingressi, l'invocazione dell'Ecc, la valutazione delle transizioni

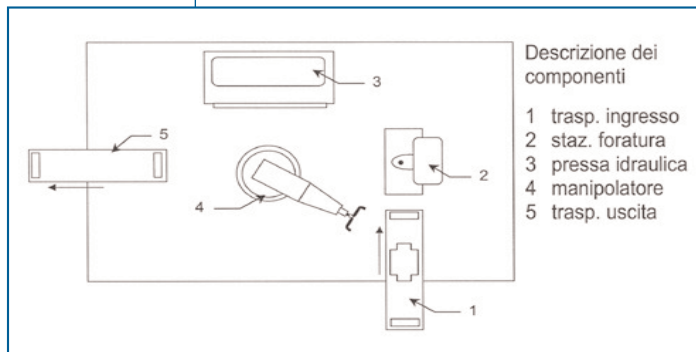


Figura 5 - Layout della cella manifatturiera considerata

dell'Ecc, lo scattare di nuove transizioni, la pianificazione degli algoritmi e il loro completamento, con la corrispondente generazione degli eventi in uscita. Tali diagrammi Stateflow e le loro relazioni con gli automi per eventi in ingresso ed evoluzione dell'Ecc, descritti in Figura 2 e 3 rispettivamente, sono riportati in Figura 4. Descrivendo ogni Fb tramite il corrispondente blocco Simulink, modelli Iec 61499 complessi, costituiti dall'aggregazione di più Fb a differenti livelli gerarchici, possono essere rappresentati agevolmente in Simulink realizzando un modello strutturalmente equivalente. Si possono quindi definire librerie di blocchi Simulink rappresentanti i Fb secondo la normativa Iec 61499. Tali librerie possono essere impiegate per tradurre modelli di sistemi di automazione dal formalismo dello standard Iec a modelli di simulazione. Si possono così sfruttare le funzionalità di gestione delle librerie fornite da Simulink, e favorire il riutilizzo di blocchi già definiti.

**Progetto e verifica del sistema di controllo di una cella di produzione, esempio applicativo**

In questa sezione i metodi e gli strumenti presentati vengono applicati al progetto e alla ve-

rifica del sistema di supervisione e controllo di una cella di produzione di arredi d'ufficio, dove in particolare sono prodotti cassette da scrivania. La cella considerata è descritta in Figura 5. Il ciclo di lavoro è il seguente: una lamina metallica entra nella cella su un nastro trasportatore (1), e viene afferrata tramite ventose da un manipolatore (4), che la trasporta a una stazione di foratura (2), dove viene praticato il foro per il pomello. Quindi la lamina viene sagomata dal manipolatore alla pressa idraulica (3) attraverso una sequenza di quattro pieghe. Infine, il cassetto viene posto su un nastro trasportatore (5) ed esce dalla cella.

Il sistema di automazione è strutturato come segue. Ogni componente della cella è descritto come un'unità: infeed (nastro trasportatore d'ingresso), manipolatore, stazione di foratura, pressa idraulica, e outfeeder (nastro trasportatore di uscita). Un supervisore di cella coordina le differenti unità, che non interagiscono tra loro, a eccezione della sincronizzazione tra manipolatore e pressa nell'operazione di piega. Inoltre è stato introdotto un semplice supervisore di area per segnalare la presenza di un nuovo pezzo da lavorare.

Il comportamento desiderato della cella è stato descritto dapprima tramite una specifica in linguaggio naturale, e quindi tradotta in regole di tipo "if-then-else". In particolare, ogni regola verifica l'occorrenza di azioni in corrispondenza di determinate condizioni; diversamente viene generato un segnale di errore. A titolo d'esempio, le regole relative alle specifiche del manipolatore sono:

- 1 - IF `piece_ready` Then `piece_at_drill`  
Else `infeeder_to_drill_error`
- 2 - IF `carry_to_press` Then `piece_at_press`  
Else `drill_to_press_error`
- 3 - IF `start_job` Then `job_end`  
Else `shape_drawer_error`
- 4 - IF `carry_to_outfeeder` Then `piece_at_outfeeder`  
Else `press_to_outfeeder_error`

Il sistema di automazione è stato progettato usando il formalismo dello standard Iec 61499 (Figura 6). Specificatamente, ogni unità, il supervisore di cella e il supervisore di area sono stati rappresentati come singoli Fb, interagenti fra loro tramite eventi. A titolo d'esempio l'Ecc del manipolatore è mostrato in Figura 7: tale Ecc è costituito da uno stato iniziale (start) e da quattro sequenze di stati e transizioni, che rappresentano le quattro possibili operazioni del manipolatore. Concordemente al modello Iec 61499 un'espressione Booleana è associata ad ogni transizione, mentre

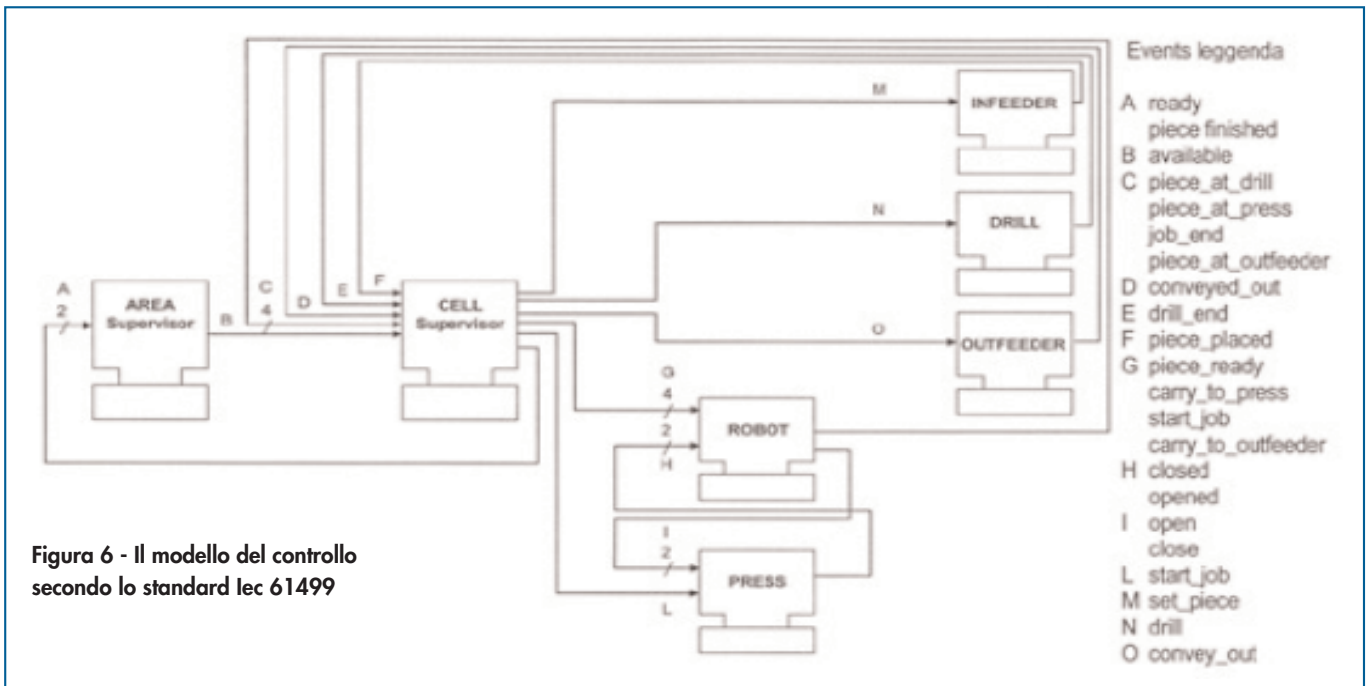
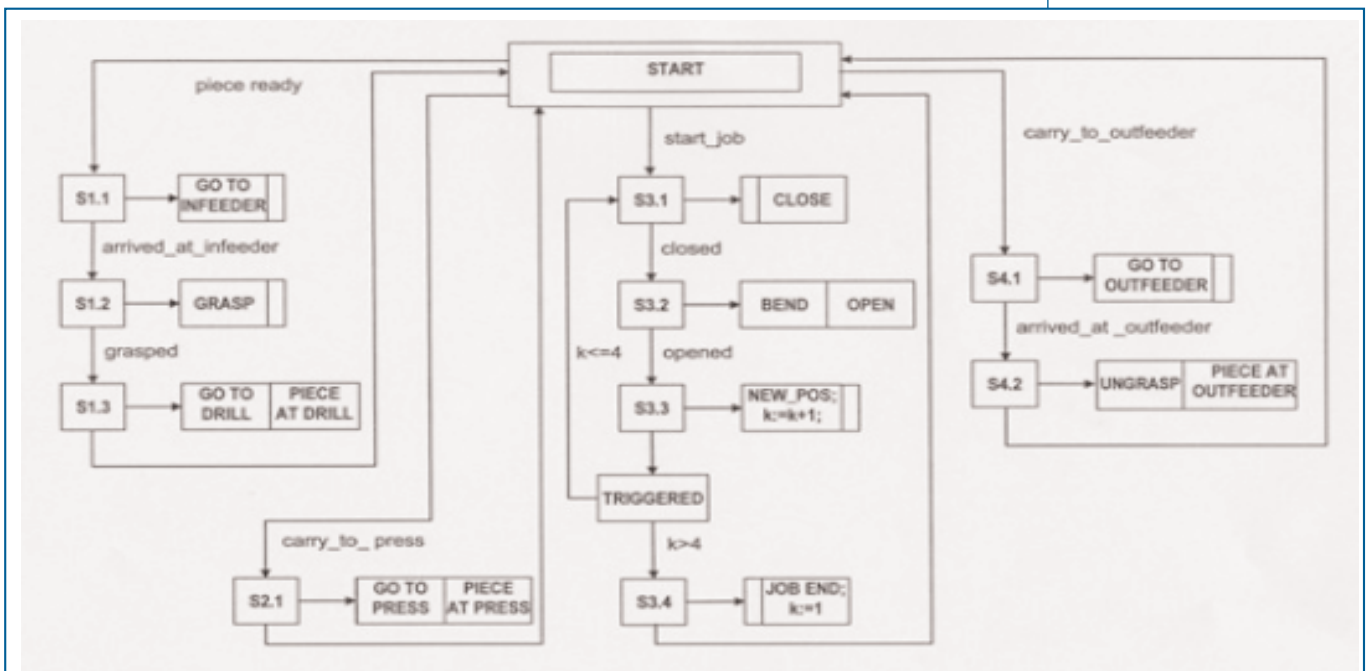


Figura 6 - Il modello del controllo secondo lo standard Iec 61499

azioni e/o eventi sono associati a ogni stato [6]. Seguendo le regole definite nel paragrafo *Traduzione dal modello Iec 61499 a modelli Simulink/Stateflow* per rappresentare un modello Iec 61499 in Simulink/Stateflow, ogni Fb è stato rappresentato in Simulink come modello Stateflow contenente più macchine a stati finiti. Ad esempio, in Figura 8 è mostrato uno dei diagrammi Stateflow definiti per rappresentare l'Ecc del manipolatore; in particolare sono evidenziati i dettagli concernenti la sequenza di operazioni da eseguire quando un

nuovo pezzo deve essere trasportato alla stazione di foratura. Tale diagramma riassume le regole principali per l'implementazione di un Fb in Stateflow. L'Ecc, rappresentato secondo lo standard Iec 61499 come mostrato in Figura 7, è arricchito in modo da favorirne la corretta valutazione in esecuzione. In particolare, un diagramma denominato *idle* è introdotto in mutua esclusione con il diagramma dell'Ecc per abilitare la valutazione di quest'ultimo solo quando necessario. Inoltre, per assicurare in ogni nuova valutazione l'evoluzione dei dia-

Figura 7 - L'Ecc del manipolatore



grammi Stateflow a partire dall'ultimo stato attivo, è incluso il concetto di cronologia in ogni stato dell'Ecc. Tale espediente è usato unitamente a una particolare struttura "stato-azione/evento" progettata per lavorare correttamente in caso di valutazioni multiple dell'Ecc (si veda lo stato denominato *ecc\_sl.1\_algo* in Figura 8). Focalizzando l'attenzione sulle transizioni, è possibile evidenziare la presenza di un evento di default, denominato *TR\_Clear*, associato alla condizione Booleana: tale evento segnala il soddisfacimento delle condizioni sulle transizioni, consentendo a un altro diagramma di rilevare se una nuova transizione scatta.

Il passo successivo è rappresentato dall'impiego di una tecnica modulare basata sulla simulazione definita in [7, 8] per la verifica automatica delle funzioni di controllo progettate. Seguendo tale tecnica, dopo il progetto di ogni singolo modulo del sistema di automazione, è possibile costruire uno schema Simulink per verificare la correttezza del modulo progettato rispetto alle sue specifiche [13]. Tale schema è

definito attraverso la corretta parametrizzazione di moduli predefiniti, che rappresentano il sistema in anello chiuso, ove il processo da controllare è opportunamente semplificato. Quindi, può essere avviata una procedura automatica per la verifica del modulo in oggetto. In particolare, il comportamento desiderato per il modulo di controllo testato è rappresentato attraverso diagrammi Stateflow, modellando ogni regola "if-then-else" della specifica tramite un opportuno diagramma (Figura 9). Inoltre, un modello semplificato del processo controllato è implementato in Simulink. Una volta verificati i Fb concernenti le singole unità, vengono verificati i Fb che rappresentano i moduli al livello di automazione più alto tramite l'approccio bottom-up presentato in [8].

Le operazioni di verifica descritte possono essere eseguite prima dell'implementazione finale sulla piattaforma di controllo, aumentando così l'affidabilità del software di controllo implementato sul sistema reale, e riducendo i tempi ed i costi di sviluppo in maniera significativa [1, 4]. Una volta verificato il sistema di

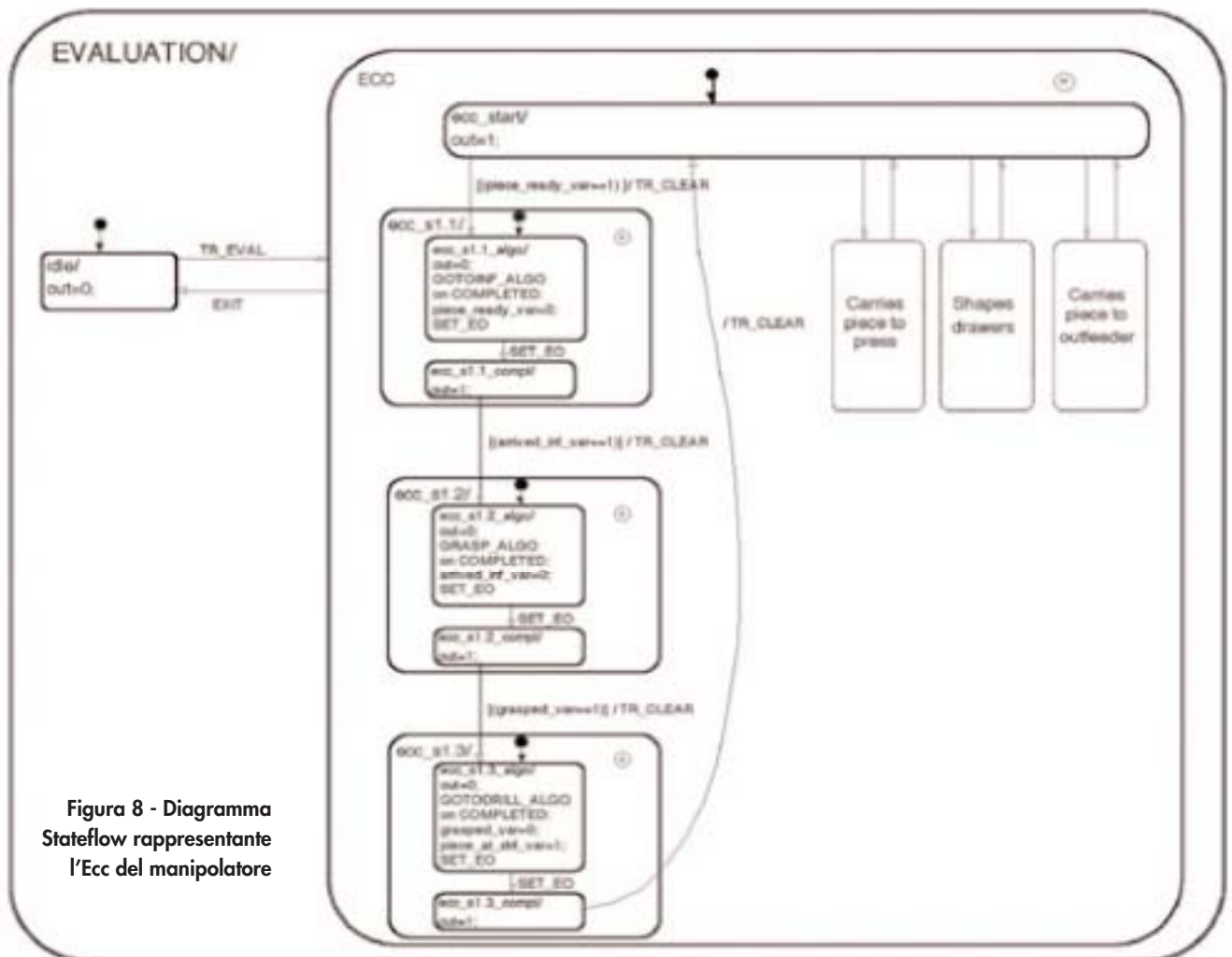
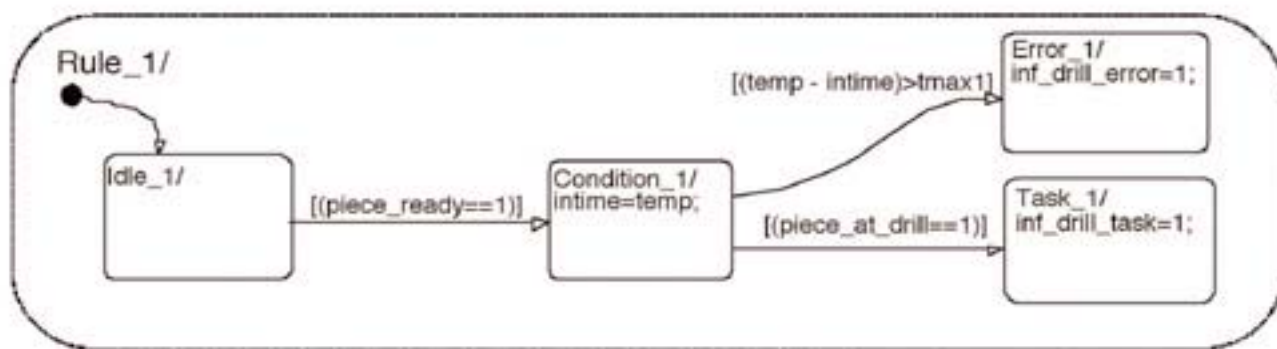


Figura 8 - Diagramma Stateflow rappresentante l'Ecc del manipolatore



automazione tramite simulazione, il software di controllo può essere generato in modo automatico a partire dal modello Iec 61499, e scaricato sulla piattaforma di destinazione.

### Considerazioni conclusive e sviluppi futuri

Nel presente lavoro sono state descritte le principali caratteristiche di una metodologia di progetto modulare, adottando il modello di riferimento per i sistemi di controllo industriali proposto dallo standard Iec 61499, e sfruttando le capacità di simulazione di Simulink e Stateflow. I principali benefici dell'approccio proposto consistono nella riduzione dei costi e dei tempi di sviluppo, e nell'affidabilità dei sistemi di controllo ottenuti.

In particolare, la descrizione di sistemi complessi risulta semplificata ed il riutilizzo di modelli è favorito attraverso l'uso di modelli formali che sfruttano i concetti di modularità, incapsulamento e astrazione; l'interoperabilità e l'intercambiabilità sono garantite dall'adozione di uno standard internazionale nella modellazione del sistema di automazione; le soluzioni di controllo sono verificate in fase di progetto tramite simulazioni. Gli sviluppi futuri del presente lavoro riguarderanno: la definizione di metodi di analisi formale per il modello di riferimento adottato; l'ottimizzazione del codice generato per operazioni di analisi, simulazione e controllo, partendo dal modello Iec 61499; e la creazione di strumenti Cacsd che supportino lo sviluppo del sistema di controllo basato sull'approccio presentato.

### Bibliografia

[1] Carpanzano E., 1999, *A Development Methodology for Hybrid Control Systems*, Ph.D. thesis, Politecnico di Milano.  
 [2] Elfving A., Kirchhoff U., 1991, *Design Methodology for Space Automation and Robotics Systems*, Esa Journal, vol. 15, pp. 149-164.

[3] Maffezzoni C., Ferrarini L., Carpanzano E., 1999, *Object-Oriented Models for Advanced Automation Engineering*, Control Engineering Practice, vol. 7, pp. 957-968.

[4] Ferrarini L., Carpanzano E., 2002, *A Structured Methodology for the Design and Implementation of Control and Supervision Systems for robotic applications*, Ieee Transactions on Control Systems Technology, vol. 10, n. 2, pp. 272-279.

[5] Schoop R., Strelzoff A., 1996, *Asynchronous and Synchronous Approaches for Programming Distributed Control Systems Based on Standards*, Control Engineering Practice, vol. 4, n. 6, pp. 855-861.

[6] Standard Iec 61499, *Function Blocks for Industrial-Process Measurements and Control Systems*, Iec TC65/WG6, 2001.

[7] Ballarino A., 2001, *Simulation based techniques for the functional verification of an industrial plant control software*, Master thesis, Politecnico di Milano.

[8] Carpanzano E., Ballarino A., *A structured approach to the design and simulation-based testing of factory automation systems*, Proceedings of the Ieee International Symposium on Industrial Electronics, Isie 2002, L'Aquila, July 8-11, 2002, pp. 181-186.

[9] Simon D., Espiau B., Castillo E., Kapellos K., 1993, *Computer-Aided Design of a Generic Robot Controller Handling Reactivity and Real-Time Control Issues*, Ieee Transactions on Control Systems Technology, vol. 1, n. 4, pp. 213-229.

[10] Using Simulink, 2001, The MathWorks, Version 4, [http://www.mathworks.com/access/helpdesk/help/pdf\\_doc/simulink/sl\\_using.pdf](http://www.mathworks.com/access/helpdesk/help/pdf_doc/simulink/sl_using.pdf)

[11] Stateflow User's Guide, 2001, The MathWorks, Version 4, [http://www.mathworks.com/access/helpdesk/help/pdf\\_doc/stateflow/sf\\_ug.pdf](http://www.mathworks.com/access/helpdesk/help/pdf_doc/stateflow/sf_ug.pdf).

[12] Ballarino A., Carpanzano E., *Modular automation systems design using the Iec 61499 standard and the Simulink/Stateflow toolboxes*, Proceedings of the Asme Japan-Usa Symposium on Flexible Automation, Jusfa 2002, Hiroshima, Japan, 14-19 July, 2002, pp. 465-472.

[13] Carpanzano E., Ferrarini L., Maffezzoni C., 2001, *Modular Testing of Logic Control Functions with Matlab*, Proceedings of the 13th European Simulation Symposium and Exhibition, Marseilles, Oct. 18-20, pp. 206-210.

Figura 9 – Chart Stateflow rappresentante la regola 1 per il manipolatore