# Composizione automatica dei servizi

Matteo Leonetti

## *Introduzione*

In questo lavoro viene affrontato il problema della composizione automatica dei servizi presentando degli esempi di realizzazione basati sulle tecniche introdotte in [1] e [2].

Il problema affrontato può essere formulato come segue: Dato un insieme di comportamenti non deterministici disponibili ed un comportamento deterministico obiettivo, stabilire se è possibile realizzare l'obiettivo utilizzando opportunamente i componenti ed in tal caso mostrare una realizzazione.

Il metodo utilizzato rappresenta i comportamenti come processi mediante macchine a stati finiti e non pone limitazioni sulla loro dinamica (ad esempio non impone che i componenti siano azioni atomiche). La conoscenza condivisa dai processi componenti e dall'obiettivo riguarda le azioni ed esprime lo stato dell'ambiente e le pre- e post-condizioni osservabili.

Il risultato della sintesi è un controllore, centralizzato o distribuito, che guida l'esecuzione dei processi componenti stabilendo le azioni che ad ogni passo ognuno deve eseguire per realizzare il comportamento obiettivo. Il controllore, se esiste, permettere la realizzazione di tutte le tracce dell'obiettivo gestendo il non determinismo dei processi componenti.

## *Definizioni*

**Ambiente $\xi$**: $<A, E, e_0, \delta_\xi>$

- A: insieme finito delle azioni
- E: insieme finito degli stati
- $e_0$: stato iniziale
- $\delta_\xi \subseteq E \times 2^A \times E$: relazione delle transizioni. Determina lo stato successivo dell'ambiente applicando un insieme di azioni ad uno stato sorgente. Sorgente e destinazione dell'azione rappresentano le pre- e post- condizioni osservabili dell'insieme delle azioni.

**Comportamento $\beta$:** $<S, s_0, G, \delta_\beta, F>$

- S: insieme finito degli stati
- $s_0$: stato iniziale
- G: insieme di guardie. Le guardie sono funzioni booleane definite sugli stati dell'ambiente ed associate ad ogni transizione. La transizione può essere eseguite se le precondizioni dell'azione sono verificate e la guardia è valutata a true.
- $\delta_\beta \subseteq S \times G \times 2^A \times S$: transizioni
- $F \subseteq S$: insieme degli stati finali

**Problema:** Dato un sistema $(\beta_1, ..., \beta_n, \xi)$ ed un comportamento obiettivo deterministico $\beta_0$ su $\xi$, sintetizzare un controllore P che realizzi $\beta_0$ assegnando opportunamente le azioni da eseguire ai comportamenti disponibili.

## *Sintesi*

E' possibile costruire una formula in PDL i cui modelli sono tutti i controllori che realizzano l'obiettivo. Se tale formula è soddisfacibile come effetto della prova si ottiene il controllore. La formula è la congiunzione della codifica degli stati iniziali, di tutti i processi e di un'insieme di assiomi che caratterizzano il dominio. Le variabili da istanziare realizzano le due funzioni di stato successivo e di etichettatura delle azioni con l'insieme di processi che le devono eseguire.

La formula non è categorica e se soddisfacibile il solutore restituisce uno qualsiasi dei possibili modelli, senza applicare un criterio di scelta.

Un controllore distribuito è costituito da un'insieme di controllori, uno per ogni processo componente, che accedendo all'ambiente e allo stato degli altri controllori mediante scambio di messaggi determinano la prossima azione da eseguire per il processo associato. Ad ogni controllore centralizzato ne corrisponde uno distribuito ottenibile dal primo.

# Coalizioni di robot

Quattro robot hanno differenti capacità e devono collaborare per esplorare l'ambiente in cui si trovano in cerca di vittime. I robot hanno le seguenti caratteristiche:

- Esploratore: localizzazione, navigazione, creazione della mappa
- Rivelatore di corpi umani: riconoscimento di una vittima basato sulla visione
- Rivelatore RFID: riconoscimento di oggetti mediante segnali RFID
- Rivelatore di vittime RFID: riconoscimento di vittime mediante segnali RFID.

Il comportamento obiettivo prevede l'esplorazione dell'area sia con la navigazione, sia con la ricerca di oggetti mediante RFID. Quando la mappa è stata creata, è possibile procedere con la ricerca delle vittime, in questo caso non contemporaneamente con i due metodi disponibili ma prima tramite i sensori ed in caso di fallimento tramite la visione.

L'ambiente è non deterministico ed il suo stato determina la fase attuale della ricerca (esplorazione o ricerca delle vittime).

In questo esempio una procedura generale viene applicata agli agenti disponibili i quali hanno capacità diverse che vengono utilizzate completamente. Nessuno dei robot singolarmente è in grado di realizzare il target ma la loro coalizione lo è.

## *Ambiente ξ*

L'ambiente individua i momenti in cui è stata completata la mappa e sono riuscite o fallite le ricerche della vittima.

**Stati *E*:** {Iniziale, Creata_mappa, RFID_fallito, Visione_fallita, Trovata_vittima}

**Azioni *A*:** {esplora, ricerca_oggetti, termina_esplorazione, ricerca_RFID, ricerca_Visione}

**Stato iniziale ($e_0$)**: Iniziale

**Transizioni:**

| Azione | Sorgente | Destinazione |
|---|---|---|
| esplora, ricerca_oggetti | Iniziale | Iniziale |
| esplora, ricerca_oggetti | Iniziale | Creata_mappa |
| esplora | Iniziale | Iniziale |
| esplora | Iniziale | Creata_mappa |
| ricerca_oggetti | Iniziale | Iniziale |
| termina_esplorazione | | lo stato *sorgente* |
| ricerca_RFID | Creata_mappa, Visione_fallita | Creata_mappa |
| ricerca_RFID | Creata_mappa, Visione_fallita | RFID_fallito |
| ricerca_RFID | Creata_mappa, Visione_fallita | Trovata_vittima |
| ricerca_visione | Creata_mappa, RFID_fallito | stato *sorgente* |
| ricerca_visione | Creata_mappa, RFID_fallito | Visione_fallita |
| ricerca_visione | Creata_mappa, RFID_fallito | Trovata_vittima |
| ricerca_fallita | RFID_fallito, Visione_fallita | lo stato *sorgente* |
| trovata_vittima | Trovata_vittima | lo stato *sorgente* |

## *Obiettivo $\beta_0$*

**Stati $S_0$:** {Esplorazione, Ricerca_RFID, Successo, Fallimento}

**Stato iniziale $s_0$:** Esplorazione

**Stati finali:** {Successo, Fallimento}

**Transizioni:**

| Azione | Sorgente | Destinazione | Guardia |
|---|---|---|---|
| esplora, ricerca_oggetti | Esplorazione | Esplorazione | Iniziale |
| esplora | Esplorazione | Esplorazione | Iniziale |
| ricerca_oggetti | Esplorazione | Esplorazione | Iniziale |
| termina_esplorazione | Esplorazione | Ricerca_RFID | Creata_mappa |
| ricerca_rfid | Ricerca_RFID | Ricerca_RFID | ¬ (RFID_fallito ∨ Trovata_vittima) |
| ricerca_fallita | Ricerca_RFID | Ricerca_visione | RFID_fallito |
| ricerca_fallita | Ricerca_visione | Fallimento | Visione_fallita |
| ricerca_visione | Ricerca_visione | Ricerca_visione | ¬ (Visione_fallita ∨ Trovata_vittima) |
| trovata_vittima | Ricerca_RFID, Ricerca_visione | Successo | Trovata_vittima |

## *Esploratore $\beta_1$*

**Stati $S_0$:** {Esplorazione, Fine}

**Stato iniziale $s_0$:** Esplorazione

**Stati finali:** {Fine}

**Transizioni:**

| Azione | Sorgente | Destinazione | Guardia |
|---|---|---|---|
| esplora | Esplorazione | Esplorazione | |
| termina_esplorazione | Esplorazione | Fine | Creata_mappa |

## *Riconoscitore oggetti $\beta_2$*

**Stati $S_0$:** {Esplorazione, Fine}

**Stato iniziale $s_0$:** Esplorazione

**Stati finali:** {Fine}

**Transizioni:**

| Azione | Sorgente | Destinazione | Guardia |
|---|---|---|---|
| ricerca_oggetti | Esplorazione | Esplorazione | |
| termina_esplorazione | Esplorazione | Fine | Creata_mappa |

## *Rivelatore vittime RFID $\beta_3$*

**Stati $S_0$:** Ricerca, Successo, Fallimento

**Stato iniziale $s_0$:** Ricerca

**Stati finali:** Successo, Fallimento

**Transizioni:**

| Azione | Sorgente | Destinazione | Guardia |
|---|---|---|---|
| ricerca_rfid | Ricerca | Ricerca | ¬ (RFID_fallito ∨ Trovata_vittima) |
| ricerca_fallita | Ricerca | Fallimento | RFID_fallito |
| trovata_vittima | Ricerca | Successo | Trovata_Vittima |

## *Rivelatore vittime visione $\beta_4$*

**Stati $S_0$:** Ricerca, Successo, Fallimento

**Stato iniziale $s_0$:** Ricerca

**Stati finali:** Successo, Fallimento

**Transizioni:**

| Azione | Sorgente | Destinazione | Guardia |
|---|---|---|---|
| ricerca_visione | Ricerca | Ricerca | ¬ (Visione_fallita ∨ Trovata_vittima) |
| ricerca_fallita | Ricerca | Fallimento | RFID_fallito |
| trovata_vittima | Ricerca | Successo | Trovata_Vittima |

## *Controllore centralizzato*

Il controllore condivide gli stati e la funzione di transizione con il comportamento obiettivo. L'etichettatura delle azioni è la seguente:

| Azione | Controllore | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\beta_4$ | $\xi$ | Agente |
|---|---|---|---|---|---|---|---|
| esplora, ricerca_oggetti | Esplorazione | Esplorazione | Esplorazione | Ricerca | Ricerca | Iniziale | $\beta_1$ - esplora<br>$\beta_2$ - ricerca_oggetti |
| esplora | Esplorazione | Esplorazione | Esplorazione | Ricerca | Ricerca | Iniziale | $\beta_1$ - esplora |
| ricerca_oggetti | Esplorazione | Esplorazione | Esplorazione | Ricerca | Ricerca | Iniziale | $\beta_2$ - ricerca_oggetti |
| termina_esplorazione | Esplorazione | Esplorazione | Esplorazione | Ricerca | Ricerca | Creata_mappa | $\beta_1$ - termina_esplorazione<br>$\beta_2$ - |

| Azione | Controllore | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\beta_4$ | $\xi$ | Agente |
|---|---|---|---|---|---|---|---|
| | | | | | | | termina_esplorazione |
| ricerca_rfid | Ricerca_RFID | Fine | Fine | Ricerca | Ricerca | Creata_mappa | $\beta_3$ - ricerca_rfid |
| trovata_vittima | Ricerca_RFID | Fine | Fine | Ricerca | Ricerca | Trovata_Vittima | $\beta_3$ - trovata_vittima<br>$\beta_4$ - trovata_vittima |
| ricerca_fallita | Ricerca_RFID | Fine | Fine | Ricerca | Ricerca | RFID_fallito | $\beta_3$ - ricerca_fallita |
| ricerca_visione | Ricerca_visione | Fine | Fine | Fine | Ricerca | RFID_fallito | $\beta_4$ - ricerca_visione |
| ricerca_fallita | Ricerca_visione | Fine | Fine | Fine | Ricerca | Visione_fallita | $\beta_4$ - ricerca_fallita |
| trovata_vittima | Ricerca_visione | Fine | Fine | Fine | Ricerca | Trovata_Vittima | $\beta_4$ - trovata_vittima |

Come descritto nel paragrafo *Sintesi* i modelli possibili per la formula PDL possono essere più di uno, e non è possibile controllare quale venga restituito. Però, tutti i controllori ammissibili devono verificare che ad uno stato finale dell'obiettivo corrispondano stati finali in tutti i processi componenti. In alcuni casi questo forza la realizzazione di certe azioni da parte di tutti i processi che non sono in uno stato finale. Un esempio è evidenziato nella tabella. I comportamenti 1 e 2 si trovano in uno stato non finale ed a seguito dell'azione termina_esplorazione nessuno dei due potrà più fare azioni. Se non eseguissero termina_esplorazione entrambi al termine dell'esecuzione uno dei due si troverebbe in uno stato non finale. In questo caso quindi tutti i modelli devono avere questa azioni associata ad entrambi i comportamenti. E' possibile sfruttare questa caratteristica dei modelli per ottenere che un insieme di agenti esegua certamente un'azione (cosa non verificabile in generale) utilizzando gli stati finali.

## *Controllore distribuito*

Il controllore distribuito è ottenuto da quello centralizzato sostituendo l'informazione sullo stato con la coppia “id : messaggio” da parte di ogni servizio componente e selezionando le sole azioni che riguardano l'agente controllato.

Il seguente controllore è relativo al robot di tipo Esploratore.

| Azione | Controllore | $\beta_1$ | 2 : | 3 : | 4 : | $\xi$ | Agente |
|---|---|---|---|---|---|---|---|
| esplora, ricerca_oggetti | Esplorazione | Esplorazione | Esplorazione | Ricerca | Ricerca | Iniziale | esplora |
| esplora | Esplorazione | Esplorazione | Esplorazione | Ricerca | Ricerca | Iniziale | esplora |
| ricerca_oggetti | Esplorazione | Esplorazione | Esplorazione | Ricerca | Ricerca | Iniziale | |
| termina_esplorazione | Esplorazione | Esplorazione | Esplorazione | Ricerca | Ricerca | Creata_mappa | termina_esplorazione |
| ricerca_rfid | Ricerca_RFID | Fine | Fine | Ricerca | Ricerca | Creata_mappa | |
| trovata_vittima | Ricerca_RFID | Fine | Fine | Ricerca | Ricerca | Trovata_Vittima | |

| Azione | Controllore | $\beta_1$ | 2 : | 3 : | 4 : | ξ | Agente |
|---|---|---|---|---|---|---|---|
| ricerca_fallita | Ricerca_RFID | Fine | Fine | Ricerca | Ricerca | RFID_fallito | |
| ricerca_visione | Ricerca_visione | Fine | Fine | Fine | Ricerca | RFID_fallito | |
| ricerca_fallita | Ricerca_visione | Fine | Fine | Fine | Ricerca | Visione_fallita | |
| trovata_vittima | Ricerca_visione | Fine | Fine | Fine | Ricerca | Trovata_Vittima | |

Si nota come in alcuni casi l'agente non debba eseguire azioni ma deve conoscere lo stato degli altri per procedere quando necessario.

# RobocupSoccer

Gli agenti di una squadra di robot calciatori hanno tutti lo stesso comportamento e sono in grado di ricoprire qualsiasi ruolo. I ruoli previsti sono “attaccante”, “difensore” e “supporto”. Si vuole considerare la possibilità di utilizzare la sintesi del controllore per assegnare i ruoli.

A differenza del caso precedente i comportamenti disponibili non vengono utilizzati del tutto ma solo nella parte relativa al ruolo scelto per l'agente. Il comportamento obiettivo è un piano “di squadra” applicabile ad un numero qualsiasi di robot. Il piano richiede che almeno un agente sia assegnato ad ogni ruolo (altrimenti l'obiettivo non sarebbe realizzabile) ma non è possibile specificare quanti lo siano effettivamente (esistono diverse assegnazioni che modellano la formula).

La sintesi del controllore viene realizzata offline prima dell'esecuzione determinando i ruoli definitivamente e senza possibilità di modificarli durante l'esecuzione.

## *Ambiente ξ*

**Stati *E*:** {PallaNonVista, PallaVista}

**Azioni *A*:** {cerca_palla, vai_versoPalla, vai_difesa, vai_supporto, prendi_palla, tira, vai_verso_porta, segui_palla, segui_attaccante}

**Stato iniziale ($e_0$)**: PallaNonVista

**Transizioni:**

| Azione | Sorgente | Destinazione |
|---|---|---|
| cerca_palla | PallaNonVista | PallaVista |
| tira | PallaVista | PallaNonVista |
| vai_versoPalla | PallaVista | PallaVista, PallaNonVista |
| vai_difesa | PallaVista | PallaVista, PallaNonVista |
| vai_supporto | PallaVista | PallaVista, PallaNonVista |
| prendi_palla | PallaVista | PallaVista, PallaNonVista |
| vai_verso_porta | PallaVista | PallaVista, PallaNonVista |
| segui_palla | PallaVista | PallaVista, PallaNonVista |
| segui_attaccante | PallaVista | PallaVista, PallaNonVista |

## *Obiettivo $\beta_0$*

**Stati $S_0$:** {Ricerca, Posizione, Attacco, Porta}

**Stato iniziale $s_0$:** Ricerca

**Stati finali:** {Ricerca}

**Transizioni:**

| Azione | Sorgente | Destinazione | Guardia |
|---|---|---|---|
| cerca_palla | Ricerca, Posizione,<br>Attacco, Porta | Ricerca | |
| vai_versoPalla,<br>vai_difesa,<br>vai_supporto | Ricerca | Posizione | |
| prendi_palla | Posizione | Attacco | |
| tira | Attacco,Porta | Ricerca | |
| vai_verso_porta | Attacco | Porta | |
| segui_palla,<br>segui_attaccante | Posizione, Attacco, Porta | lo stato *sorgente* | |

## *Comportamenti $\beta_{1..n}$*

**Stati $S_0$:** {Ricerca, Attacco, Difesa, Supporto, HoLaPalla, Porta}

**Stato iniziale $s_0$:** Ricerca

**Stati finali:** {Ricerca}

**Transizioni:**

| Azione | Sorgente | Destinazione | Guardia |
|---|---|---|---|
| cerca_palla | Ricerca, Attacco, Difesa,<br>Supporto, HoLaPalla, Porta | Ricerca | |
| vai_versoPalla | Ricerca | Attaccante | |
| vai_difesa | Ricerca | Difesa | |
| vai_supporto | Ricerca | Supporto | |
| prendi_palla | Attacco | HoLaPalla | |
| tira | HoLaPalla, Porta | Ricerca | |
| vai_verso_porta | HoLaPalla | Porta | |
| segui_palla, | Difesa | Difesa | |
| segui_attaccante | Supporto | Supporto | |

## *Controllore centralizzato*

| Azione | Controllore | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\xi$ | Agente |
|---|---|---|---|---|---|---|
| cerca_palla | Ricerca<br>Posizione<br>Attacco | Ricerca<br>Attacco<br>HoLaPalla | Ricerca<br>Difesa | Ricerca<br>Supporto | PallaNonVista | $\beta_1$ - cerca_palla<br>$\beta_2$ - cerca_palla<br>$\beta_3$ - cerca_palla |

| Azione | Controllore | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\xi$ | Agente |
|---|---|---|---|---|---|---|
| | Porta | Porta | | | | |
| vai_versoPalla<br>vai_difesa<br>vai_supporto | Ricerca | Ricerca | Ricerca | Ricerca | PallaVista | $\beta_1$ - vai_versoPalla<br>$\beta_2$ - vai_difesa<br>$\beta_3$ - vai_supporto |
| prendi_palla | Posizione | Attacco | Difesa | Supporto | PallaVista | $\beta_1$ - prendi_palla |
| tira | Attacco | HoLaPalla | Difesa | Supporto | PallaVista | $\beta_1$ - tira |
| tira | Porta | Porta | Difesa | Supporto | PallaVista | $\beta_1$ - tira |
| vai_verso_porta | Attacco | HoLaPalla | Difesa | Supporto | PallaVista | $\beta_1$ - vai_verso_porta |
| segui_palla<br>segui_attaccante | Posizione<br>Attacco<br>Porta | Attacco<br>HoLaPalla<br>Porta | Difesa | Supporto | PallaVista | $\beta_2$ - segui_palla<br>$\beta_3$ - segui_attaccante |

Nella prima riga evidenziata l'azione di ricerca della palla viene svolta da tutti gli agenti, come sembra ragionevole. Questa però è solo una delle realizzazione possibili e non c'è garanzia che ciò effettivamente avvenga. Come indicato nel caso precedente, anche in questo esempio c'è la certezza che in tutti i modelli un agente esegue l'azione se si trova in uno stato diverso da Ricerca perché è uno stato non finale.

Nella seconda riga evidenziata viene realizzata l'assegnazione dei ruoli. Tutte le azioni devono essere eseguite e nessun agente può svolgerne più di una, quindi almeno tre agenti sono richiesti affinché l'obiettivo sia raggiungibile. Non è però possibile specificare nel target in che modo distribuire gli agenti sui ruoli.

## *Controllore distribuito*

Controllore relativo al comportamento $\beta_2$:

| Azione | Controllore | 1 : | $\beta_2$ | 3 : | $\xi$ | Agente |
|---|---|---|---|---|---|---|
| cerca_palla | Ricerca<br>Posizione<br>Attacco<br>Porta | Ricerca<br>Attacco<br>HoLaPalla<br>Porta | Ricerca<br>Difesa | Ricerca<br>Supporto | PallaNonVista | cerca_palla |
| vai_versoPalla<br>vai_difesa<br>vai_supporto | Ricerca | Ricerca | Ricerca | Ricerca | PallaVista | vai_difesa |
| prendi_palla | Posizione | Attacco | Difesa | Supporto | PallaVista | |
| tira | Attacco | HoLaPalla | Difesa | Supporto | PallaVista | |
| tira | Porta | Porta | Difesa | Supporto | PallaVista | |
| vai_verso_porta | Attacco | HoLaPalla | Difesa | Supporto | PallaVista | |
| segui_palla<br>segui_attaccante | Posizione<br>Attacco<br>Porta | Attacco<br>HoLaPalla<br>Porta | Difesa | Supporto | PallaVista | segui_palla |

# Commercio elettronico

Realizzazione di un servizio responsabile del reperimento dei dati sui prodotti in commercio, della verifica della loro disponibilità e del perfezionamento dell'ordine con il pagamento presso un istituto bancario.

Le fonti di dati sui prodotti sono in un numero qualsiasi. Il problema affrontato in questo esempio è quello della coerenza dei dati, cioè le fonti di dati che li forniscono al primo passo *pubblica_prodotti* devono essere esattamente le stesse che eseguono *verifica_disponibilità*. Come già menzionato non è possibile controllare *quanti* agenti eseguono l'operazione ma ciò che qui importa è che tutti quelli che eseguono la prima effettuino anche la seconda. Per ottenere questo sono stati sfruttati ancora una volta gli stati finali.

## *Ambiente ξ*

Gli stati dell'ambiente sono tutti i sottoinsiemi di valori di verità per alcune proposizioni. Questo specifica le precondizioni senza indicare un ordine per le azioni. Lo stato destinazione è specificato in modo strips-like indicando le proposizioni modificate dall'azione.

**Stati *E*:** Tutti i sottoinsiemi delle proposizioni {Ricerca_ok, Ordine_ok, Pagamento_ok}

**Azioni *A*:** {pubblica_prodotti, mostra_elenco, ricerca, verifica_disponibilità, accettazione_ordine, avvia_pagamento, pagamento}

**Stato iniziale ($e_0$)**: ¬Ordine_ok ∧ ¬Ricerca_ok ∧ ¬Pagamento_ok

**Transizioni:**

| Azione | Sorgente | Destinazione |
|---|---|---|
| pubblica_prodotti | | lo stato *sorgente* |
| mostra_elenco | | lo stato *sorgente* |
| ricerca | ¬Ricerca_ok | -¬Ricerca_ok<br>+Ricerca_ok |
| ricerca | ¬Ricerca_ok | lo stato *sorgente* |
| verifica_disponibilità | ¬Ordine_ok | -¬Ordine_ok<br>+Ordine_ok |
| verifica_disponibilità | ¬Ordine_ok | lo stato *sorgente* |
| accettazione_ordine | Ordine_ok | lo stato *sorgente* |
| avvia_pagamento, pagamento | ¬Pagamento_ok | -¬Pagamento_ok<br>+Pagamento_ok |
| avvia_pagamento, pagamento | ¬Pagamento_ok | lo stato *sorgente* |
| conferma | | ¬Ordine_ok ∧ ¬Ricerca_ok ∧ ¬Pagamento_ok |
| annulla | | ¬Ordine_ok ∧ ¬Ricerca_ok ∧ ¬Pagamento_ok |

## *Obiettivo $\beta_0$*

**Stati $S_0$:** {Iniziale, Pronto, Ricerca_eseguita, Ordine, Accettato, Pagamento}

**Stato iniziale $s_0$:** Iniziale

**Stati finali:** {Iniziale}

**Transizioni:**

| Azione | Sorgente | Destinazione | Guardia |
|---|---|---|---|
| pubblica_prodotti | Iniziale | Pronto | |
| mostra_elenco | Pronto | Pronto | |
| ricerca | Pronto | Ricerca_eseguita | |
| verifica_disponibilità | Ricerca_eseguita | Ordine | Ricerca_ok |
| accettazione_ordine | Ordine | Accettato | |
| avvia_pagamento, pagamento | Accettato | Pagamento | |
| conferma | Pagamento | Iniziale | Pagamento_ok |
| annulla | Ricerca_eseguita<br>Ordine<br>Accettato | Iniziale | |
| annulla | Pagamento | Iniziale | ¬Pagamento_ok |

## *Presentazione informazioni $\beta_1$*

**Stati $S_0$:** {Presentazione, Ricerca, Accettazione, Pagamento}

**Stato iniziale $s_0$:** Presentazione

**Stati finali:** {Presentazione}

**Transizioni:**

| Azione | Sorgente | Destinazione | Guardia |
|---|---|---|---|
| mostra_elenco | Presentazione | Presentazione | |
| ricerca | Presentazione | Ricerca | |
| accettazione_ordine | Ricerca | Accettazione | |
| avvia_pagamento | Accettazione | Pagamento | |
| conferma | Pagamento | Presentazione | Pagamento_ok |
| annulla | Ricerca<br>Accettazione | Presentazione | |
| annulla | Pagamento | Presentazione | ¬Pagamento_ok |

## *Magazzino $\beta_2$*

**Stati $S_0$:** {Iniziale, Pronto}

**Stato iniziale $s_0$:** Iniziale

**Stati finali:** {Iniziale}

**Transizioni:**

| Azione | Sorgente | Destinazione | Guardia |
|---|---|---|---|
| pubblica_prodotti | Iniziale | Pronto | |
| verifica_disponibilita | Pronto | Iniziale | |
| annulla | Pronto | Iniziale | |

## *Banca $\beta_3$*

**Stati $S_0$:** {Attività}

**Stato iniziale $s_0$:** Attività

**Stati finali:** {Attività}

**Transizioni:**

| Azione | Sorgente | Destinazione | Guardia |
|---|---|---|---|
| pagamento | Attività | Attività | |

## *Controllore centralizzato*

| Azione | Controllore | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\xi$ | Agente |
|---|---|---|---|---|---|---|
| pubblica_prodotti | Iniziale | Presentazione | Iniziale | Attività | ¬Ordine_ok ∧ ¬Ricerca_ok ∧ ¬Pagamento_ok | $\beta_2$ - pubblica_prodotti |
| mostra_elenco | Pronto | Presentazione | Pronto | Attività | ¬Ordine_ok ∧ ¬Ricerca_ok ∧ ¬Pagamento_ok | $\beta_1$ - mostra_elenco |
| ricerca | Pronto | Presentazione | Pronto | Attività | ¬Ordine_ok ∧ ¬Ricerca_ok ∧ ¬Pagamento_ok | $\beta_1$ - ricerca |
| verifica_disponibilità | Ricerca_eseguita | Ricerca | Pronto | Attività | ¬Ordine_ok ∧ Ricerca_ok ∧ ¬Pagamento_ok | $\beta_2$ - verifica_disponibilità |
| annulla | Ricerca_eseg | Ricerca | Pronto | Attività | ¬Ordine_ok ∧ | $\beta_1$ - annulla |

| Azione | Controllore | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\xi$ | Agente |
|---|---|---|---|---|---|---|
| | uita | | | | ¬Ricerca_ok ∧ ¬Pagamento_ok | $\beta_2$ - annulla |
| accettazione_ordine | Ordine | Ricerca | Iniziale | Attività | Ordine_ok ∧ Ricerca_ok ∧ ¬Pagamento_ok | $\beta_1$ - accettazione_ordine |
| avvia_pagamento, pagamento | Accettato | Accettazione | Iniziale | Attività | Ordine_ok ∧ Ricerca_ok ∧ ¬Pagamento_ok | $\beta_1$ - avvia_pagamento $\beta_3$ - pagamento |
| conferma | Pagamento | Pagamento | Iniziale | Attività | Ordine_ok ∧ Ricerca_ok ∧ Pagamento_ok | $\beta_1$ - avvia_pagamento |
| annulla | Pagamento | Pagamento | Iniziale | Attività | Ordine_ok ∧ Ricerca_ok ∧ ¬Pagamento_ok | $\beta_1$ - annulla |
| annulla | Ricerca_eseguita Accettato | Ricerca Accettazione | Iniziale | Attività | * | $\beta_1$ - annulla |

# Riferimenti


[1] G. De Giacomo, S. Sardina. Automatic Synthesis of New Behaviors from a Library of Available Behaviors, IJCAI 2007

[2] G. De Giacomo . Automatic Synthesis of Global Behaviors from Multiple Distributed Existing Behaviors