

# Ro.Ma.S.

(Robot Management System)

## RMS\_R\_001

(ProcessManager - Requisiti)



## Storia del documento

<b>Data</b>	<b>Rilascio</b>	<b>Descrizione</b>
2009.01.29	0.0.1	Bozza 1
2009.02.09	0.0.2	Bozza 2
2009.02.20	0.0.3	Pre-release

## Scopo del documento

Stabilire i requisiti che deve avere il servizio ProcessManager.

# Requisiti

## Premessa ai requisiti

- ✓ Il servizio ProcessManager esegue le azioni secondo un tipo ed una priorità di azione.
- ✓ Tutti i servizi compiono un tipo specifico di azioni che vengono gestite dal servizio ProcessManager ed aggregate in uno scopo (SCOPE).
- ✓ Il servizio codifica le azioni in istruzioni da eseguire dalla ACTION\_TABLE (anagrafica delle azioni) per scriverle nella ACTION\_PROCESS\_TABLE.
- ✓ Il servizio ProcessManager esegue un ciclo di azioni per tipologia ed ordine decrescente di priorità (dalla più alta alla più bassa), questo significa che prima verranno eseguite tutte le azioni con priorità alta per ogni servizio.

Es.

PRIORITY	SCOPE	ACTION	COMMAND	PARAMS	ID_PROCESS	TYPE	OWNER
1	MOVEMENT	FORWARD	START		22	ORDINARY	12
2	MOVEMENT	FORWARD	STOP		23	ORDINARY	12
1	SENSOR	READ_DISTANCE	FRONT		25	ORDINARY	13
1	VIDEO	CLIENT CONNECTION	OPEN	1562	26	ORDINARY	14
1	SPEAK	SPEAKER	TEXT	"BUONGIORNO"		ORDINARY	16
2	SPEAK	SPEAKER	TEXT	"INTRUSIONE"	27	ORDINARY	16

Tabella 1 (ACTION\_PROCESS\_TABLE)

Come mostrato nella **Tabella 1** verranno eseguite le azioni 1002, 2105, 0066, 0034 in ordine di tipo e priorità.

La tabella dei servizi prevede che ci sia una priorità di esecuzione delle azioni dei servizi secondo il loro livello già definito nella **Tabella 1** del documento "RoMaS - Documento di progetto".

Particolare attenzione merita la descrizione della tabella 1:  
**PRIORITY:** Priorità dell'azione all'interno dello stesso SCOPE.  
**SCOPE:** Ambito applicativo o raggio d'azione di un servizio.  
**COMMAND / INSTRUCTION:** Istruzione o comando da eseguire.  
**PARAMS:** Parametri da passare all'istruzione.  
**ID\_ACTION:** Identificativo dell'azione.  
**ID\_PROCESS:** Processo incaricato di eseguire l'azione.  
**OWNER:** Azione proprietaria dell'azione stessa.

Esempio di ACTION\_TABLE dove immagazzinare l'anagrafica delle azioni lette dal file di configurazione del servizio.

ID_ACTION	SCOPE	ACTION	COMMAND	INSTRUCTION
00001	MOVEMENT	FORWARD	START	COM2 #1 P300 T3600000 (per un ora)
00002	SENSOR	FRONT	READ DISTANCE	COM2 #001 RS1 (esempio)
00003	VIDEO	OPEN CONNECTION	CLIENT {port}	new VideoSocketClient(port)

Tabella 2 (ACTION\_TABLE)

Il servizio prevede che le azioni vengano processate attraverso l'esecuzione del metodo del servizio ActionService => ExecuteAction che deve essere implementato.

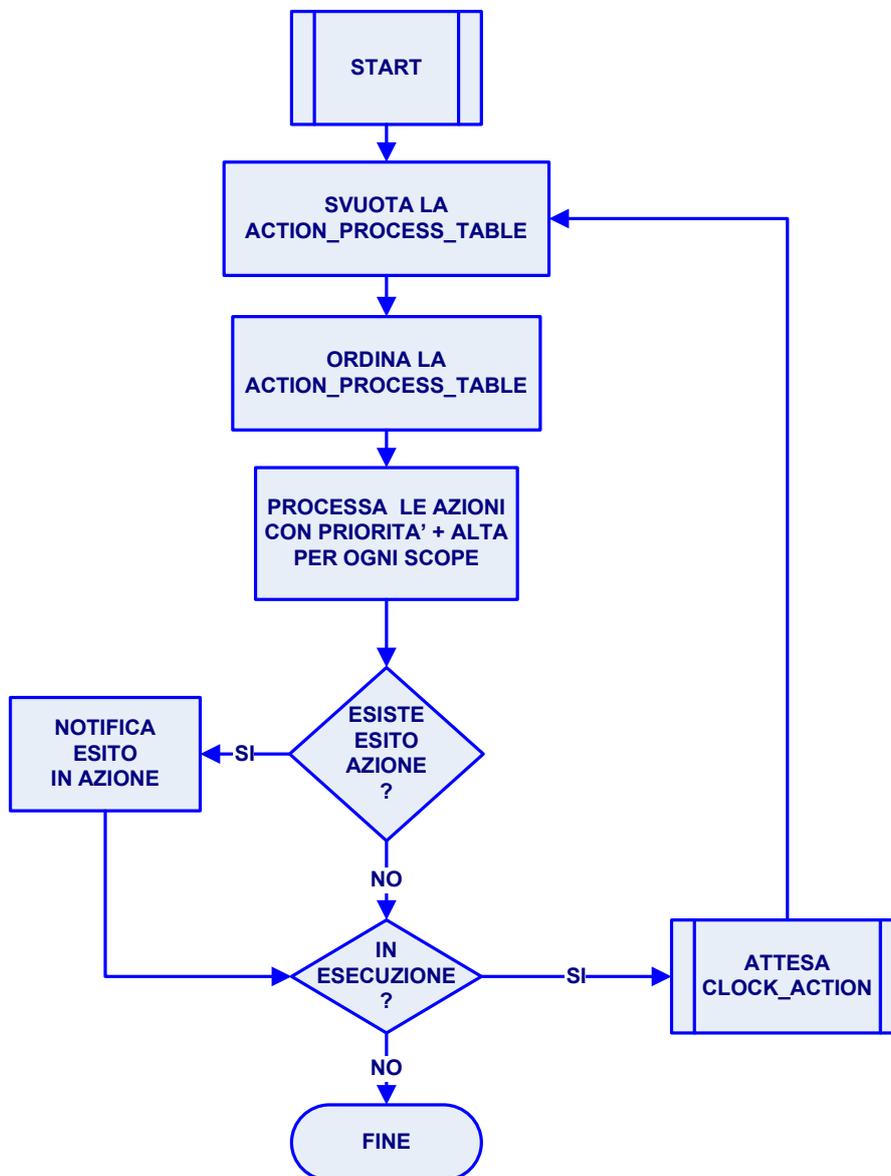


Figura 1

### Tabella dei requisiti

<b>Numero</b>	<b>Descrizione</b>
	Al suo avvio il servizio svuota la tabella ACTION_PROCESS_TABLE, per attuare un reset del sistema.
	Identifica le azioni dirette come di tipo ordinario (TYPE = 'ORDINARY').
	Esegue le azioni per cicli di tipo e priorità a scadenza di un tempo denominato CLOCK_ACTION.
	Dopo ogni ciclo di azioni il servizio riordina le azioni per priorità e tipo di modo da eseguire le azioni nel giusto ordine.
	Esegue prima le azioni "figlie" prima dell'azione padre (OWNER).
	Ripete le azioni un numero di volte definito dal valore ITERATION.
	Ripete l'azioni all'infinito se il valore ITERATION = -1.
	Lo stato di ogni azione cambia a seconda della fase del processo in cui si trova: CREATED, INITIATED, RUNNING, STOPPED, FINISHED, DELETING.
	Archivia le azioni cancellate dalla ACTION_PROCESS_TABLE nella ACTION_PROCESS_ARCHIVE_TABLE per un numero da definire (default= 1000).

Tabella 3