|  |  |
| --- | --- |
| Nome evento | Tasto Console |
| inputcmd : usercmd(executeInput(CMD)) | (RUN button) |
| usercmd : usercmd(robotgui(w(low))) | (FORWARD button) |
| alarm : alarm(fire) | (FIRE button) |
| obstacle : obstacle(X) | (OBSTACLE button) |
| cmd : cmd(start) | (START button) |
| cmd : cmd(stop) | (STOP button) |

msg( MSGID, MSGTYPE, SENDER, RECEIVER, CONTENT, SEQNUM )

|  |  |
| --- | --- |
| MSGID | Message identifier |
| MSGTYPE | Message type (e.g.:dispatch,request,invitation,event,token) |
| SENDER | Identifier of the sender |
| RECEIVER | Identifier of the receiver |
| CONTENT | Message payload |
| SEQNUM | Unique natural number associated to the message |

The msg/6 (numero di parametri) pattern può essere usato per esprimere le guards (see Subsection 8.2) to allow conditional evaluation of PlanActions.

Msg/6

Qualsiasicosa/Numero 🡪 numero di parametri che prende in input

Termini prolog:

|  |  |  |
| --- | --- | --- |
| Nome | Definizione | Esempio |
| answerEv | Ad esempio se sta riproducendo la musica, e gli arriva l’evento, lui devo rispondere fermando la musica |  |
| consult | Può fare il load del file prolog, che serve per aggiungere altri eventi nella base di conoscenza |  |
| context |  |  |
| continue |  |  |
| demand | Richiedi a VarOQactor un’azione o un messaggio |  |
| demo | Serve per risolvere dei goal di prolog |  |
| **Dispatch** | Messaggio, notizia?  Nomedispatch :  mqttmsg : mqttmsg( TOPIC,PAYLOAD ) | **Dispatch** polar : p( distance, Angle ) |
| emit | Manda l’evento ad esempio alarm |  |
| **EventHandler** |  | Pagina 18 qactors |
| **forward** | Richiedi a VarOQactor un valore |  |
| **-httpserver** | Serve per poter visualizzare direttamente sulla pagina localhost la shell |  |
| **memoCurrentEvent** |  |  |
| **Normal** | Primo plan che viene eseguito |  |
| **onFailSwitchTo** |  |  |
| **onMessage** |  |  |
| **printCurrentEvent** |  |  |
| **react** **event** |  |  |
| **receiveAndSwitch** |  |  |
| **replyToCaller -m** |  |  |
| **switchToPlan** |  |  |
| **solve** | Serve per risolvere dei goals di prolog( vedi anche demo) |  |
|  |  |  |
|  |  |  |
| **[!? goalResult(X)] println(x)** | Se quella confizione è vera non la rimuove dalla base di conoscenza (WorldTheory) |  |
| **[?? goalResult(X)] println(x)** | Se quella condizione è vera la rimuove dalla base di conoscenza (WorldTheory) |  |

Pagina 40 di qactors nuovo!!!

7.2.8 endActionInternal() operation.

The operation endActionInternal is executed when the action is terminated; it evaluates the action

execution time and emits the termination event with payload result(RES,EXECTIME):

The operation endActionInternal

|  |
| --- |
| /\* Calculate action execution time and emit the termination event \*/  2 protected T endActionInternal() throws Exception{  3 evalDuration();  4 T res = endOfAction();  5 if(terminationEvId != null && terminationEvId.length()>0) {  6 emitEvent( terminationEvId, res.toString() );  7 }  8 return res;  9 }  10 protected void emitEvent(String event, String res) throws Exception{  11 QActorUtils.raiseEvent((it.unibo.qactors.QActorContext) ctx, getName(), event, outS);  12 } |