INTERAZIONE

* **local\_gui\_command : local\_gui\_command (COMMAND)**, evento generato dalla GUI in risposta alle interazioni dell’utente.

COMMAND cambia in base al bottone premuto sull’interfaccia:

* + Load Map --> *loadmap(PATH);*
  + Find Path –-> *findpath(START,GOAL);*
  + Start Navigation --> *navigate*;
  + Abort --> *abort*;
  + Manipulate --> *manipulate* (**TODO**)
* **navdata : navdata (DATA)**, messaggio inviato dalla Console al Robot per trasmettere i dati necessari per la navigazione (mappa, path e posizioni).

Ho scelto di utilizzare un messaggio per la loro natura buffered.

Mentre un evento può essere perso, i messaggi vengono mantenuti in un buffer di input. Sarebbe piuttosto problematico perdere uno step del plan o un elemento della mappa.

DATA cambia in base al tipo di informazione:

* + *mapdata ( map ( Xmax , Ymax ) )* e *mapdata ( element ( X , Y ) ),* usati per inviare le informazioni della mappa.
  + *plan ( robotmove )* e *plan ( robotspin ( DIR ) ),* usati per inviare il piano ottimale. È compito del Robot convertire i comandi in operazioni elaborabili;
  + *position ( X , Y )*, per START e GOAL.
* **abort : abort**, evento generato dalla Console, provoca la terminazione di qualsiasi operazione stia eseguendo il Robot;
* **end : end**, evento generato dal Robot per indicare la fine di un compito;
* **update : update (** *element ( X , Y )* **),** evento lanciato dal Robot quando rileva un nuovo ostacolo statico, in questo modo la Console può a sua volta aggiornare la Mappa.