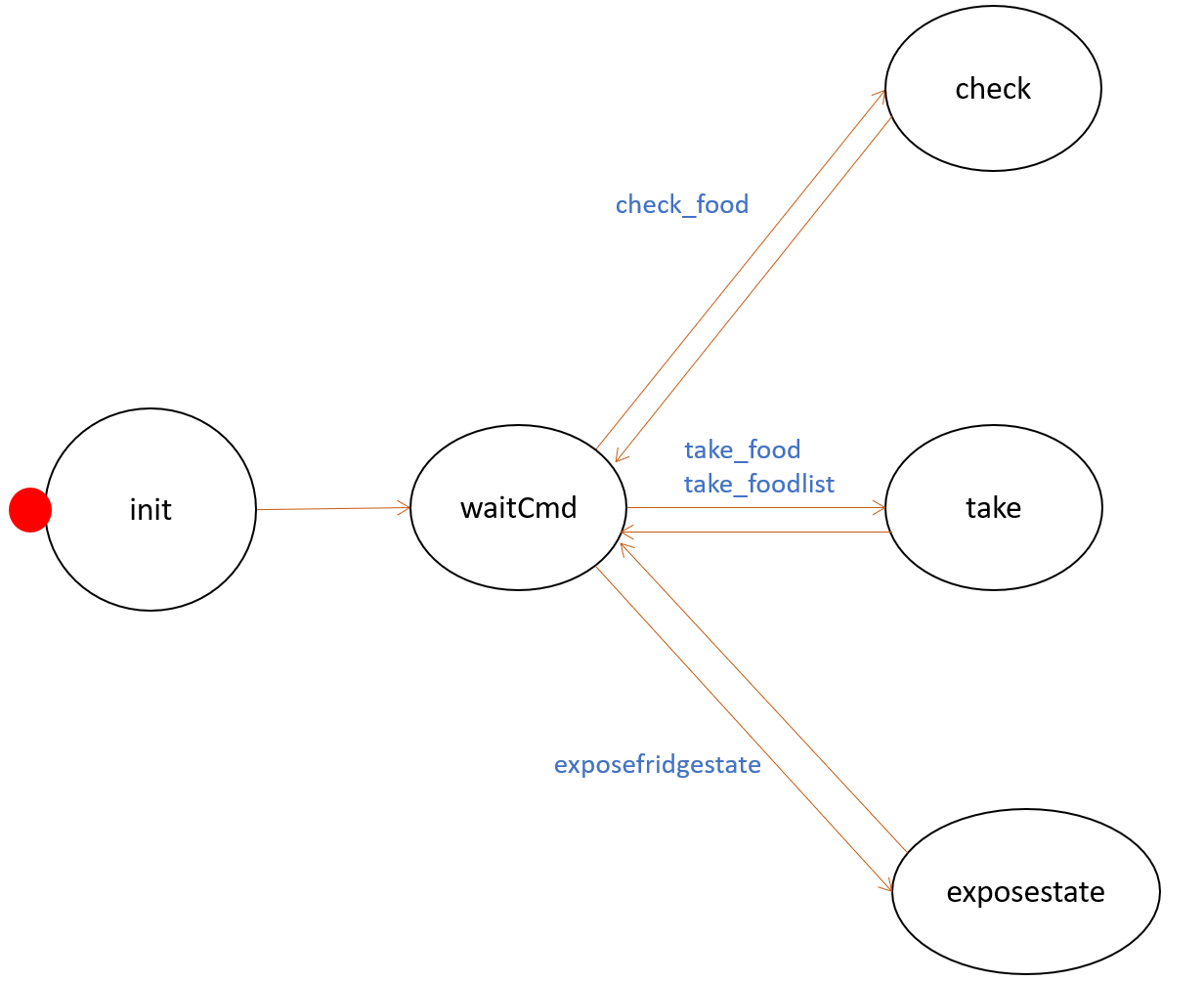
# Sprint 2 - Frigo smart

Il frigo smart è realizzato come un QActor il cui comportamento può essere modellato come una Finite State Machine:

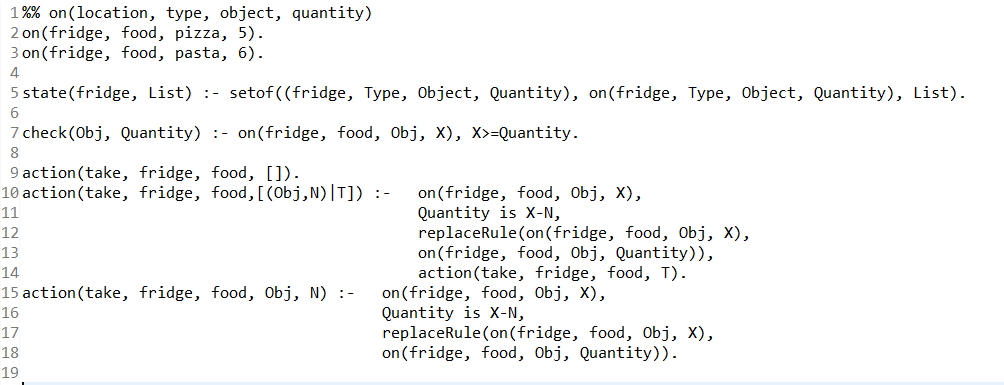


Le transizioni sono scatenate da messaggi del seguente tipo e id:

* check\_food(Foodcode, Quantity): messaggio di tipo Request, che permette di verificare la presenza di una certa quantità del cibo identificato da Foodcode. Nello stato check il frigo invia la risposta
* take\_food(Foodcode, Quantity): messaggio di tipo Dispatch, che permette di comunicare al frigo smart il prelievo di un cibo. È stata implementata anche una versione che accetta come argomento una lista di cibi con le relative quantità
* exposestate: messaggio di tipo Request con cui è possibile chiedere al frigo di esporre il suo contenuto. Nello stato exposestate il frigo invia la risposta

Il fatto di aver utilizzato messaggi di tipo Request/Reply disaccoppia il frigo dagli altri elementi del sistema e lo rende quindi in grado di offrire le informazioni sul suo contenuto anche ad eventuali elementi futuri.

Lo stato è mantenuto nel file prolog fridgestate.pl, insieme alle regole per accedervi e modificarlo:



**TestPlan**

Obiettivi:

* testare il funzionamento dell’interazione con il frigo

**ADD**

**Pre condizioni**: maître, robot e fridge sono in attesa di comandi

* si invia un evento “add\_button(pizza,4)” al maître

**Post condizioni**: il maître si deve trovare in uno stato di “add”, il robot in “add\_started” e il frigo in “check\_food”

**EXPOSE FRIDGE STATE**

**Pre condizioni**: il fridge è in attesa di comandi

* si invia un evento “exposefridgestate” al frigo

**Post condizioni**: il fridge si deve trovare in uno stato di “exposestate”

**TAKE FOOD**

**Pre condizioni**: il fridge è in attesa di comandi

* si invia un evento “take\_food(pizza,4)” al frigo

**Post condizioni**: il maître si deve trovare in uno stato di “takefood”