Descrizione Eventi/Azioni

Azioni dei treni:

***info\_treno(1,1).***

Fact che definisce l’ID e la priorità dell’agente Treno.

***goE :>***

***write('sto partendo'),***

***messageA(agentTreno2, send\_message(parti,Me)),***

***once(arrivo\_al\_passaggioA).***

L’agente scrive “sto partendo” e manda un messaggio al Treno successivo innescando la sua partenza, successivamente esegue l’azione *arrivo\_al\_passaggio*.

***arrivo\_al\_passaggioA :>***

***messageA(agentPassaggio1, send\_message(chiede\_passaggio(1,1), Me)).***

L’agente chiede di passare al suo relativo agente Passaggio, inviando un messaggio e passandogli il suo ID e la sua priorità.

***attraversaE:>***

***write('Il passaggio mi ha appena concesso ad attraversare'),***

***nl,***

***termino\_attraversamentoA.***

Una volta ricevuta l’autorizzazione per passare il treno scrive “il passaggio mi ha appena concesso di attraversare” ed esegue l’azione *termino\_attraversamento.*

***termino\_attraversamentoA:-***

***write('Ho attraversato incrocio'),***

***nl,***

***messageA(agentPassaggio1,***

***send\_message(termino\_attraversamento(1), Me)).***

L’agente scrive “ho attraversato l’incrocio” e invia un messaggio al suo agente Passaggio notificando l’avvenuto attraversamento e passando il suo ID.

Azioni dei passaggi:

***chiede\_passaggioE(X,Y) :>***

***now(Timestamp),***

***assert(treno\_prenotato(X,Y,Timestamp)),***

***format('il treno con id ~w ha chiesto di attraversare al tempo ~2f', [X, Timestamp]),***

***nl.***

L’agente controlla il Timestamp attuale e apprende l’ID la priorità e il Timestamp dell’Agente Treno che ha invocato l’evento. Successivamente stampa una stringa descrivendo l’id dell’Agente che ha richiesta di attraversare e il Timestamp.

***elabora\_codaA :-***

***write('Elaboro la coda...'),***

***nl,***

***treno\_prenotato(\_,Xp,\_),***

***not((treno\_prenotato(\_,Xp2,\_), Xp2 > Xp)),***

***treno\_prenotato(X,Xp,Xt),***

***not((treno\_prenotato(\_,Xp,Xt2), Xt2 < Xt)),***

***format('Treno successivo: ~d', X),***

***nl,***

***retract(strada\_libera(true)),***

***assert(strada\_libera(false)),***

***concedi\_autorizzazioneA(X).***

In questa azione viene elaborata la coda dei treni controllando la priorità e a parità di priorità il tempo di arrivo, viene poi rimossa la clausola *strada\_libera* precedentemente settata a True e viene aggiunta la stessa clausola settata a False, per indicare l’occupazione del passaggio, infine l’Agente esegue l’azione *concedi\_autorizzazione* passando come parametro l’ID dell’Agente Treno che può passare.

***concedi\_autorizzazioneA(X):-***

***format('Ora il treno con id ~d puo passare', X),***

***nl,***

***number\_chars(X, App),***

***atom\_chars(XString, App),***

***atom\_concat('agentTreno', XString, AgentName),***

***messageA(AgentName, send\_message(attraversa, Me)).***

L’Agente attraverso una serie di operazioni costruisce la Stringa contenente il nome dell’Agente Treno e il suo ID e invia un messaggio a tale Agente concedendo l’autorizzazione all’attraversamento.

***termino\_attraversamentoE(X):>***

***retract(treno\_prenotato(X,\_,\_)),***

***format('Il treno con id ~d mi ha comunicato di aver terminato di attraversare', X), nl,***

***retract(strada\_libera(false)),***

***assert(strada\_libera(true)).***

Una volta che il Treno ha attraversato il Passaggio, l’Agente Passaggio rimuove il treno e la variabile *strada\_libera* viene di nuovo settata a True, per indicare che il Passaggio è di nuovo libero e in caso il treno successivo può attraversare.

***strada\_liberaI :-***

***strada\_libera(Z),***

***true == Z,***

***treno\_prenotato(X,Y,Timestamp),***

***elabora\_codaA.***

Evento interno che sta ad indicare che l’azione *elabora\_coda* può partire solo quando la strada è libera e c’è un Treno che ha richiesto di passare.