

Implementazione e Analisi "TruDaMul Framework".

Francesco Rossi
Francesco Rombaldoni

1 Obiettivo del lavoro.

L'obiettivo del lavoro è stato quello di verificare alcune proprietà del "TruDaMul Framework" tramite una implementazione semplificata dello stesso, realizzata con l'ausilio del linguaggio "NuSMV".

2 Svolgimento del lavoro e strumenti usati

Il lavoro è stato svolto aprendo una "repository" su "GitHub" alla quale ci si è interfacciati utilizzando il programma "GitHubDesktop". La "repository" (da ora chiamata semplicemente "repo") è stata poi suddivisa in tre "branch", in ordine: "main branch" è stato usato per ospitare la presentazione del progetto e in conclusione il "merging" con il "branch" contenente la versione del "Framework" più rifinita; il "Case Base branch" è stato usato per ospitare una implementazione base del "Framework", dalla quale successivamente è stato fatto un "branch" per la costruzione di un caso più specifico da testare; l'ultimo "branch", chiamato "Relation" è servito per ospitare la relazione.

Terminata l'organizzazione della "repo", si è passati ad installare i programmi necessari per lo sviluppo del "Framework" con i relativi test e per la scrittura della relazione in \LaTeX . In Particolare per poter disporre di un ambiente integrato per lo sviluppo del "Framework" è stato installato il "JDK di Java" per fare in modo di installare successivamente "Eclipse" con l'estensione "NuSMV"; per fare in modo di eseguire il "Framework" è stata scaricata l'applicazione per "Bash" di "NuSMV"; infine per scrivere la relazione in \LaTeX è stato installato "TeX Live" e "TeXstudio", rispettivamente per compilare e per scrivere.

Una volta aver terminato di scrivere la presentazione del progetto si è passati all'implementazione del caso base, definendo i comportamenti ed il numero minimo degli attori, oltre che le prime proprietà da verificare. Il passo successivo è stato quello di creare un nuovo "branch" a partire dal caso base, in maniera tale da poter specializzare quest'ultimo per la costruzione di un caso completo.

3 Descrizione del 'Caso Base'

3.1 Motivazione del 'Caso Base'

Il 'Caso Base' è stato implementato per avere un'implementazione base funzionante del "Framework" ad un alto livello di astrazione, su cui poter in seguito aggiungere comportamenti più complessi. Viene inoltre utilizzato per verificare che non ci siano dei comportamenti anomali, come per esempio il fatto che il Client (privo di denaro) possa cercare di assoldare dei Muli.

3.2 Attori

Il caso base si compone dei seguenti attori:

- un Client.

- due Muli.
- un Proxy.
- un Server,.

3.3 Descrizione degli attori

- Il Client, che è in grado di comunicare soltanto con i Muli, è composto dai seguenti elementi:
 - Una variabile "State" che può assumere i seguenti valori:
 - * "Null" se il Client non ha un messaggio da inviare.
 - * "Request" se il Client ha un messaggio da inviare.
 - * "According" se il Client sta decidendo di assoldare il mulo.
 - * "Waiting" se il Client ha inviato il messaggio (dopo aver pagato il Mulo) e sta aspettando di ricevere una risposta.
 - * "Response" se un Mulo ha consegnato al Client una risposta.
 - * "Payment" quando il client sta pagando il mulo per aver ritirato il messaggio o per avergli consegnato la risposta.
 - Una variabile "Wallet" che contiene i soldi che il Client può spendere per assoldare i Muli. La variabile può assumere valori tra lo 0 e il 50 (estremi compresi).
 - Una variabile "Choice" che contiene l'id del Mulo a cui è stato affidato il messaggio, vale zero se il Client non ha affidato il messaggio ad un Mulo, altrimenti vale 1 se il messaggio è stato affidato al primo Mulo, 2 per il secondo.
 - Un comportamento che modifica la variabile "State":
 - * Inizialmente la variabile è inizializzata con lo stato "Null".
 - * Il Client entra nello stato "Request" quando decide (in maniera non predicibile) che vuole mandare un messaggio.
 - * Il Client entra nello stato di "According" quando ha un messaggio da spedire e un Mulo è nella sua zona.
 - * Il Client entra nello stato di "Response" se precedentemente ha inviato un messaggio e un Mulo gli consegna la risposta.
 - * Il Client entra nello stato di "Waiting" quando ha consegnato un messaggio ad un Mulo.
 - * Il Client entra nello stato di "Payment" quando sta pagando il mulo per aver consegnato il messaggio o per aver portato la risposta.
 - * Una volta aver ricevuto la risposta la variabile "State" viene ripristinata al valore di partenza.
 - Un comportamento che modifica la variabile "Wallet", la quale viene decrementata ad ogni pagamento effettuato dal Client.
 - Un comportamento che modifica la variabile "Choice":
 - * La variabile vale 0 se il Client non ha consegnato nessun messaggio.

- * La variabile vale 1 se il Client ha assoldato il Mulo 1.
- * La variabile vale 2 se il Client ha assoldato il Mulo 2.
- * Nel caso in cui entrambi i Muli siano in prossimità del Client la variabile assume un valore non deterministico tra 1 e 2.
- * Una volta aver ricevuto la risposta la variabile viene ripristinata a 0.

Il Client è un attore che trovandosi in una condizione di isolamento dalla rete e avendo necessità di mandare un messaggio al server, assolda un Mulo (sotto compenso) per spedire il messaggio. Una volta aver pagato il Mulo ed avergli consegnato il messaggio, il Client aspetta una risposta dal Server, la quale dovrà essere consegnata da un Mulo (non necessariamente deve essere lo stesso che ha preso in carico il messaggio), il quale dovrà essere pagato dal Client dopo aver mostrato la risposta del Server.

- I Muli che sono in grado di comunicare con il Client e con il Proxy e che inoltre conoscono il proprio codice identificativo sono composti da:
 - una variabile "State" che può assumere i seguenti valori:
 - * "Walking" se il Mulo è in cerca di un lavoro e cammina in modo casuale tra il Proxy ed il Client.
 - * "TalkingToC" se il Mulo è nella fase di contrattazione con il Client.
 - * "TalkingToP" se il mulo è nella fase di contrattazione con il Proxy.
 - * "GoingToC" se il Mulo si sta dirigendo verso il Client per la consegna della risposta data dal Proxy.
 - * "GoingToP" se il Mulo si sta dirigendo verso il Proxy per consegnare il messaggio del Client.
 - * "DeliverToC" se il Mulo sta consegnando la risposta al Client.
 - * "DeliverToP" se il Mulo sta consegnando il messaggio al Proxy.
 - * "Wait" se il Mulo sta attendendo di essere pagato dal Client.
 - Una variabile "Wallet" che contiene i soldi che il Mulo guadagna. La variabile può assumere valori tra lo 0 e il 50 (estremi compresi).
 - Una variabile "Position" che può assumere i seguenti valori:
 - * "Client" se il Mulo si trova in prossimità del Client.
 - * "Middle" se il Mulo si trova tra il Client ed il Proxy.
 - * "Proxy" se il Mulo si trova in prossimità del Proxy.
 - Un comportamento che modifica la variabile "State":
 - * Inizialmente la variabile è inizializzata con lo stato "Walking".
 - * Il Mulo entra in modalità "TalkingToC" se si trova in prossimità del Client e quest'ultimo ha un messaggio da inviare.
 - * Il Mulo entra in modalità "TalkingToP" se si trova in prossimità del Proxy e quest'ultimo ha una risposta da inviare.
 - * Il Mulo entra nello stato "GoingToC" se è stato assoldato dal proxy.

- * Il Mulo entra nello stato "GoingToP" se è stato assoldato dal client.
 - * Il Mulo entra nello stato "DeliverToC" se dopo aver raggiunto il Client, gli sta consegnando la risposta.
 - * Il Mulo entra nello stato "DeliverToP" se dopo aver raggiunto il Proxy, gli sta consegnando il messaggio.
 - * Il Mulo entra nello stato "Wait" quando sta aspettando di essere pagato dopo aver consegnato la risposta al Client.
 - * Alla fine del lavoro la variabile "State" viene ripristinata al valore di partenza.
- Un comportamento che modifica la variabile "Wallet", la quale viene incrementata ogni volta che il mulo viene pagato.
 - Un comportamento che modifica la variabile "Position":
 - * Se il Mulo è nello stato "Walking" la variabile può assumere in modo non predicibile uno dei tre valori precedentemente elencati.
 - * Se il Mulo deve dirigersi verso il "Proxy", la variabile assume il valore "Proxy".
 - * Se il Mulo deve dirigersi verso il "Client", la variabile assume il valore "Client".

Il compito dei Muli è quello di mettere in comunicazione il Client con il Proxy e viceversa, questa cosa la fanno grazie alla loro capacità di prendere un messaggio e di consegnarlo al destinatario. I Muli si muovono casualmente tra il Client ed il Proxy, in attesa di ricevere un messaggio da consegnare. I Muli si fanno pagare per il loro lavoro esclusivamente dal Client; infatti il Mulo si fa pagare dal Client dopo aver consegnato un messaggio e nel caso in cui il mulo consegna al Client una risposta data dal Server, il Client dovrà pagare il mulo dopo aver ricevuto la risposta.

- Il Proxy che è in grado di comunicare soltanto con i Muli e con il Server è composto dai seguenti elementi:
 - Una variabile "State" che può assumere i seguenti valori:
 - * "Null" se il Proxy è in attesa di ricevere un messaggio
 - * "Message" quando il Proxy ha ricevuto un messaggio da un Mulo.
 - * "Forward" quando il Proxy sta inoltrando il messaggio al Server.
 - * "Wait" quando il Proxy è in attesa di ricevere una risposta dal Server.
 - * "Response" quando il Proxy ha ricevuto la risposta dal Server.
 - * "WaitingMule" quando il Proxy è in attesa di un Mulo a cui consegnare la risposta.
 - * "According" quando il Proxy sta assoldando un Mulo.
 - Una variabile "Choice" che contiene l'id del Mulo a cui è stata affidata la risposta, vale zero se il Proxy non ha affidato la risposta ad un Mulo, altrimenti vale 1 se il messaggio è stato affidato al primo Mulo, 2 per il secondo.
 - Un comportamento che modifica la variabile "State":

- * Inizialmente la variabile viene inizializzata con lo stato "Null".
 - * Il Proxy entra nello stato "Message" se ha ricevuto un messaggio da un Mulo.
 - * Il Proxy entra nello stato "Forward" quando sta per inoltrare il messaggio al Server.
 - * Il Proxy entra nello stato "Wait" quando ha inoltrato il messaggio al Server e attende la risposta.
 - * Il Proxy entra nello stato "Response" quando ha ricevuto la risposta.
 - * Il Proxy entra nello stato "WaitingMule" quando aspetta un Mulo per inoltrargli la risposta data dal Server.
 - * Il Proxy entra nello stato "According" quando un Mulo si trova in prossimità del Proxy.
 - * Una volta aver inoltrato la risposta ad un Mulo la variabile ritorna allo stato di partenza.
- Un comportamento che modifica la variabile "Choice":
- * La variabile vale 0 il Proxy non ha consegnato nessun messaggio.
 - * La variabile vale 1 se il Proxy ha assoldato il Mulo 1.
 - * La variabile vale 2 se il Proxyent ha assoldato il Mulo 2.
 - * Nel caso in cui entrambi i Muli siano in prossimità del Proxy la variabile assume un valore non deterministico tra 1 e 2.
 - * Una volta aver inoltrato la risposta la variabile viene ripristinata a 0.

Il compito del Proxy è quello di inoltrare al Server il messaggio del Client che i Muli gli hanno consegnato. Una volta aver inoltrato il messaggio al Server, il Proxy, rimane in attesa che quest'ultimo gli inoltri la risposta da dover consegnare al Client che ha mandato il messaggio. Appena il Proxy riceve la risposta dal Server, resta in attesa di poter inoltrare la risposta ad un Mulo, il quale la dovrà poi recapitare al Client.

- Il Server che è in grado solo di comunicare con il Proxy è composto dai seguenti elementi:
 - Una variabile "State" che può assumere i seguenti valori:
 - * "Null" quando il Server è in attesa di ricevere un messaggio dal Proxy.
 - * "Message" quando il server ha ricevuto un messaggio dal Proxy.
 - * "Make" quando il Server sta preparando una risposta per il Proxy.
 - * "Forward" quando il Server sta inoltrando la risposta al Proxy.
 - Un comportamento che modifica la variabile "State":
 - * Inizialmente la variabile viene inizializzata con lo stato "Null".
 - * Il Server entra nello stato "Message" quando il Proxy gli ha inoltrato un messaggio.
 - * Il Server entra nello stato "Make" dopo aver ricevuto il messaggio dal server.

- * Il Server entra nello stato "Response" quando sta per consegnare la risposta al Proxy.
- * Il Server ritorna nello stato "Null" dopo aver inoltrato la risposta al Proxy.

Il compito del Server è quello di ricevere i messaggi dal Proxy, di elaborare una risposta e infine di inoltrare la risposta a quest'ultimo in modo che possa essere consegnata al Client.

3.4 Flusso di lavoro

1. All'inizio i Muli camminano casualmente tra il Client ed il Proxy, in attesa che qualcuno abbia bisogno di loro per consegnare un messaggio.
2. Se il Client ha bisogno di un Mulo per mandare un messaggio, entra nella modalità di richiesta, rimanendo in attesa di un Mulo a cui poter affidare il messaggio.
3. Quando un Mulo raggiunge la zona del Client, controlla se quest'ultimo vuole inviare un messaggio, nel caso in cui la risposta è positiva può iniziare la fase di according, dove il Client decide se affidare il lavoro al Mulo consegnandogli il messaggio da inoltrare al Proxy.
4. Il Mulo cammina fino al raggiungimento del Proxy.
5. Il Mulo consegna il messaggio al Proxy e successivamente ricomincia a muoversi casualmente tra il Proxy ed il Client.
6. Quando il Proxy riceve il messaggio, lo inoltra al Server e attende una risposta da quest'ultimo .
7. Quando il Server riceve un messaggio dal Proxy, inizia a elaborare una risposta da dover inoltrare a quest'ultimo.
8. Quando il Proxy ha ricevuto la risposta dal Server entra in modalità richiesta, attendendo il passaggio di un Mulo a cui poter inoltrare la risposta da dover consegnare al Client.
9. Una volta che il Mulo è nella zona del Proxy, controlla se quest'ultimo ha un messaggio da inviare al Client, se la risposta è affermativa, allora il mulo si prendere carico della risposta da consegnare al Client.
10. Il Mulo cammina fino a raggiungere la zona del Client.
11. A questo punto il Client paga il Mulo dopo aver ricevuto la risposta.

3.5 Test implementati

- È stato implementato il test per verificare se viene sempre soddisfatta la richiesta del Client, ovvero se il Client quando ha bisogno di inviare un messaggio viene sempre raggiunto da un mulo e se riceve sempre la risposta formulata dal Server.
Il risultato dimostra che siccome i muli si muovono in maniera causale, potrebbero non arrivare mai dal Client, come anche dal Proxy, motivo

per cui il Client potrebbe non venire sempre servito oppure potrebbe non ricevere la risposta da Server.

- È stato implementato un test per verificare che il portafoglio del Client possa scendere sotto lo 0 durante l'esecuzione.
Il risultato mostra che ciò non è possibile, perché se il Client non ha più soldi con cui assoldare i Muli, allora rimane sempre nello stato in cui non ha un messaggio da mandare.
- È stato implementato un test per verificare se durante l'esecuzione soltanto uno dei due Muli viene scelto dal Client per fare tutte le consegne.
Il risultato dimostra che siccome la scelta del Mulo non è predicibile allora può accadere in un caso estremo che soltanto un Mulo esegue tutte le consegne.
- È stato implementato un test per verificare se quando il Proxy entra nello stato di richiesta di un Mulo, la sua richiesta viene sempre "ascoltata".
il risultato dimostra che la richiesta del Proxy non viene sempre ascoltata perché potrebbe accadere che i muli non arrivino mai nella sua zona.

4 Descrizione del 'Caso Avanzato'

4.1 Motivazione del 'Caso Avanzato'

Il 'Caso Avanzato' è stato implementato per visualizzare i comportamenti del "Framework" nel caso in cui ci fosse un sistema di reputazione secondo il quale un Client che si trovi all'interno di una zona poco sicura per il Mulo possa essere rifiutato, siccome nella zona poco sicura aumenta la probabilità che il Mulo venga derubato del messaggio. Nel caso in cui il Mulo dovesse perdere il messaggio, viene inserito all'interno di una "black list", nella quale sono scritti i Muli segnalati come poco affidabili. I Client di a loro volta, possono accedere alla "black list" e sulla base di quest'ultima valutare di assoldare o meno un Mulo per spedire il proprio messaggio.

4.2 Attori

Il caso avanzato si compone dei seguenti attori:

- un Client.
- due Muli.
- un Proxy.
- un Server,.

4.3 Descrizione degli attori

- Il Client che è in grado di comunicare soltanto con i Muli è composto dai seguenti elementi:
 - Una variabile "State" che può assumere i seguenti valori:

- * "Null" se il Client non ha un messaggio da inviare.
 - * "Request" se il Client ha un messaggio da inviare.
 - * "According" se il Client sta decidendo se assoldare il Mulo.
 - * "Waiting" quando il Client ha inviato il messaggio e sta aspettando di ricevere una risposta.
 - * "Response" se un Mulo ha consegnato al Client una risposta.
 - * "Payment" quando il Client sta pagando il Mulo per avergli consegnato il messaggio o per avergli portato la risposta.
- Una variabile "Wallet" che contiene i soldi che il Client può spendere per assoldare i Muli. La variabile può assumere valori tra lo 0 e il 50 (estremi compresi).
 - Una variabile "Choice" che contiene l'id del Mulo a cui è stato affidato il messaggio, vale zero se il Client non ha affidato il messaggio ad un Mulo, altrimenti vale 1 se il messaggio è stato affidato al primo Mulo, 2 per il secondo.
 - Una variabile "Safety" che modella il caso in cui il Client si trovi o meno in una zona sicura per i Muli.
 - Comportamento che modifica la variabile "State":
 - * Inizialmente la variabile è inizializzata con lo stato "Null".
 - * Il Client entra nello stato "Request" quando decide (in maniera non predicibile) che vuole mandare un messaggio.
 - * Il Client entra nello stato di "According" quando ha un messaggio da spedire e un Mulo è nella sua zona.
 - * Il Client entra nello stato di "Response" se precedentemente ha inviato un messaggio e un Mulo gli consegna la risposta.
 - * Il Client entra nello stato di "Waiting" quando ha consegnato un messaggio ad un Mulo e attende la risposta.
 - * Il Client entra nello stato di "Payment" quando sta pagando il Mulo per aver consegnato il messaggio o la risposta
 - * Una volta aver ricevuto la risposta e pagato il mulo la variabile "State" viene ripristinata al valore di partenza.
 - Un comportamento per la variabile "Wallet" la quale viene decrementata ad ogni pagamento effettuato dal Client.
 - Un comportamento che modifica la variabile "Choice":
 - * La variabile vale 0 il Client non ha consegnato nessun messaggio.
 - * La variabile vale 1 se il Client ha assoldato il Mulo 1.
 - * La variabile vale 2 se il Client ha assoldato il Mulo 2.
 - * Nel caso in cui entrambi i Muli siano in prossimità del Client la variabile assume un valore non deterministico tra 1 e 2.
 - * Una volta aver ricevuto la risposta la variabile viene ripristinata a 0.
 - Un comportamento per la variabile "Safety" che alterna i valori booleani in modo non predicibile.

Il Client è un attore che trovandosi in una condizione di isolamento dalla rete e avendo necessità di mandare un messaggio al server, assolda un Mulo (sotto compenso) per spedire il messaggio. Dopo avergli consegnato il messaggio, il Client aspetta una risposta dal Server, la quale dovrà essere consegnata da un Mulo (non necessariamente deve essere lo stesso che ha preso in carico il messaggio), il quale verrà pagato dal Client dopo aver consegnato la risposta del Server.

- I Muli che sono in grado di comunicare con il Client e con il Proxy e che inoltre conoscono il proprio codice identificativo sono composti da:
 - una variabile "State" che può assumere i seguenti valori:
 - * "Walking" se il Mulo cammina in modo causale tra il Proxy ed il Client.
 - * "TalkingToC" se il Mulo è nella fase di contrattazione con il Client.
 - * "TalkingToP" se il mulo è nella fase di contrattazione con il Proxy.
 - * "GoingToC" se il Mulo si sta dirigendo verso il Client per la consegna della risposta data dal Proxy.
 - * "GoingToP" se il Mulo si sta dirigendo verso il Proxy per consegnare il messaggio del Client.
 - * "DeliverToC" se il Mulo sta consegnando la risposta al Client.
 - * "DeliverToP" se il Mulo sta consegnando il messaggio al Proxy.
 - * "Wait" se il Mulo sta attendendo di essere pagato dal Client.
 - Una variabile "Wallet" che contiene i soldi che il Mulo guadagna. La variabile può assumere valori tra lo 0 e il 50 (estremi compresi).
 - Una variabile "Position" che può assumere i seguenti valori:
 - * "Client" se il Mulo si trova in prossimità del Client.
 - * "Middle" se il Mulo si trova tra il Client ed il Proxy.
 - * "Proxy" se il Mulo si trova in prossimità del Proxy.
 - Una variabile "MessageState" che può assumere i seguenti valori:
 - * "Null" se il Mulo non ha ancora un messaggio.
 - * "Lost" se il Mulo ha perso il messaggio.
 - * "Owned" se il Mulo possiede ancora il messaggio e non l'ha ancora consegnato.
 - Una variabile booleana "BlackList" la quale modella il caso in cui il Mulo sia stato o meno inserito all'interno di una "black list".
 - Una variabile "Decision" che può assumere i seguenti valori:
 - * "Null" se il Mulo non ha ancora ricevuto una proposta di lavoro da parte di un Client.
 - * "Accept" se il Mulo ha accettato la proposta di lavoro del Client.
 - * "Refuse" se il Mulo non ha accettato la proposta di lavoro del Client.
 - Un comportamento che modifica la variabile "State":

- * Inizialmente la variabile è inizializzata con lo stato "Walking".
 - * Il Mulo entra in modalità "TalkingToC" se si trova in prossimità del Client e quest'ultimo ha un messaggio da inviare.
 - * Il Mulo entra in modalità "TalkingToP" se si trova in prossimità del Proxy e quest'ultimo ha una risposta da inviare.
 - * Il Mulo entra nello stato "GoingToC" se è stato assoldato dal proxy.
 - * Il Mulo entra nello stato "GoingToP" se è stato assoldato dal client.
 - * Il Mulo entra nello stato "DeliverToC" se dopo aver raggiunto il Client, gli consegna la risposta.
 - * Il Mulo entra nello stato "DeliverToP" se dopo aver raggiunto il Proxy, gli consegna il messaggio.
 - * Il Mulo entra nello stato "Wait" se sta attendendo di essere pagato per aver consegnato la risposta al Client o il messaggio al proxy.
 - * Alla fine del lavoro la variabile "State" viene ripristinata al valore di partenza.
- Un comportamento che modifica la variabile "Wallet" la quale viene incrementata ogni volta che il mulo viene pagato.
 - Un comportamento che modifica la variabile "Position":
 - * Se il Mulo è nello stato "Walking" la variabile può assumere in modo non predicibile uno dei tre valori precedentemente elencati.
 - * Se il Mulo deve dirigersi verso il "Proxy", la variabile assume i valori : middle → Proxy.
 - * Se il Mulo deve dirigersi verso il "Client", la variabile assume i valori : middle → client.
 - Un comportamento che modifica la variabile "MessageState":
 - * Inizialmente la variabile è inizializzata nello stato "Null" siccome il Mulo non ha prelevato ancora nessun messaggio.
 - * Se il Mulo accetta il lavoro proposto dal Client e se il Client accetta il Mulo, allora lo stato della variabile diventa "Owned".
 - * Se il Mulo non accetta il lavoro proposto dal Client, oppure se il Client non accetta il Mulo, la variabile rimane nello stato "Null".
 - * Se il Mulo si trova in corrispondenza di una zona poco sicura, c'è il pericolo che perda il messaggio (sia se si sta dirigendo verso la zona poco sicura sia nel caso in cui proviene da essa), in questo caso la variabile potrebbe assumere il valore "Lost" non deterministicamente.
 - * Se il Mulo si trova all'interno della zona a metà tra il Client ed il Proxy, c'è pericolo che perda il messaggio, in questo caso la variabile potrebbe assumere il valore "Lost" non deterministicamente.
 - * Una volta che il messaggio è stato consegnato la variabile torna al valore "Null".
 - Un comportamento che modifica la variabile "BlackList":

- * Se il Mulo durante una consegna perde il messaggio, la variabile assume il valore di 'TRUE'.
- * Se il Mulo ha fatto una consegna che è andata a buon fine (il Mulo non ha perso il messaggio), la variabile assume il valore di 'FALSE'.
- Un comportamento che modifica la variabile "Decision"
 - * Inizialmente la variabile è inizializzata con il valore "Null".
 - * Se il mulo è in prossimità del client (il quale ha un messaggio da inviare) e quest'ultimo è in una zona sicura, allora la variabile assume il valore "Accept".
 - * Se il mulo è in prossimità del client (il quale ha un messaggio da inviare) e quest'ultimo è in una zona poco sicura, allora la variabile può assumere non deterministicamente un valore tra "Accept" e "Refuse".
 - * Dopo aver preso la scelta, la variabile ritorna a valore "Null".

Il compito dei Muli è quello di mettere in comunicazione il Client con il Proxy e viceversa, questa cosa la fanno grazie alla loro capacità di prendere un messaggio e di consegnarlo al destinatario. I Muli si muovono casualmente tra il Client ed il Proxy, in attesa di ricevere un messaggio da consegnare. I Muli si fanno pagare per il loro lavoro esclusivamente dal Client; infatti il Mulo si fa pagare dal Client dopo aver consegnato il messaggio o la risposta. Il mulo non viene pagato se perde il messaggio.

- Il Proxy che è in grado di comunicare soltanto con i Muli e con il Server è composto dai seguenti elementi:
 - Una variabile "State" che può assumere i seguenti valori:
 - * "Null" se il Proxy è in attesa di ricevere un messaggio
 - * "Message" quando il Proxy ha ricevuto un messaggio da un Mulo.
 - * "Forward" quando il Proxy sta inoltrando il messaggio al Server.
 - * "Wait" quando il Proxy è in attesa di ricevere una risposta dal Server.
 - * "Response" quando il Proxy ha ricevuto la risposta dal Server.
 - * "WaitingMule" quando il Proxy è in attesa di un Mulo a cui consegnare la risposta.
 - * "According" quando il Proxy sta assoldando un Mulo.
 - Una variabile "Choice" che contiene l'id del Mulo a cui è stata affidata la risposta, vale zero se il Proxy non ha assoldato un Mulo, altrimenti vale 1 se il messaggio è stato affidato al primo Mulo, 2 per il secondo.
 - Un comportamento che modifica la variabile "State":
 - * Inizialmente la variabile viene inizializzata con lo stato "Null".
 - * Il Proxy entra nello stato "Message" se ha ricevuto un messaggio da un Mulo.
 - * Il Proxy entra nello stato "Forward" quando sta per inoltrare il messaggio al Server.

- * Il Proxy entra nello stato "Wait" quando ha inoltrato il messaggio al Server.
 - * Il Proxy entra nello stato "Response" quando il Server ha inoltrato la risposta.
 - * Il Proxy entra nello stato "WaitingMule" quando sta aspettando un Mulo per inoltrargli la risposta data dal Server.
 - * Il Proxy entra nello stato "According" quando un Mulo si trova in prossimità del Proxy (il quale sta attendendo un mulo).
 - * Dopo aver inoltrato la risposta ad un Mulo la variabile ritorna allo stato di partenza.
- Un comportamento che modifica la variabile "Choice":
- * La variabile vale 0 se Proxy non ha assoldato un Mulo.
 - * La variabile vale 1 se il Proxy ha assoldato il Mulo 1.
 - * La variabile vale 2 se il Proxy ha assoldato il Mulo 2.
 - * Nel caso in cui entrambi i Muli siano in prossimità del Proxy la variabile assume un valore non deterministico tra 1 e 2.
 - * Dopo aver inoltrato ad un Mulo la risposta la variabile viene ripristinata a 0.

Il compito del Proxy è quello di inoltrare al Server il messaggio del Client che i Muli gli hanno consegnato. Una volta aver inoltrato il messaggio al Server, il Proxy, rimane in attesa che quest'ultimo gli inoltri la risposta da dover consegnare al Client che ha mandato il messaggio. Appena il Proxy riceve la risposta dal Server, resta in attesa di poter inoltrare la risposta ad un Mulo, il quale la dovrà poi recapitare al Client.

- Il Server che è in grado solo di comunicare con il Proxy è composto dai seguenti elementi:
 - Una variabile "State" che può assumere i seguenti valori:
 - * "Null" quando il Server è in attesa di ricevere un messaggio dal Proxy.
 - * "Message" quando il server ha ricevuto un messaggio dal Proxy.
 - * "Make" quando il Server sta preparando una risposta per il Proxy.
 - * "Forward" quando il Server sta inoltrando la risposta al Proxy.
 - Un comportamento che modifica la variabile "State"
 - * Inizialmente la variabile viene inizializzata con lo stato "Null".
 - * Il Server entra nello stato "Message" quando il Proxy gli ha inoltrato un messaggio.
 - * Il Server entra nello stato "Make" dopo aver ricevuto il messaggio dal server.
 - * Il Server entra nello stato "Forward" quando sta inoltrando la risposta al Proxy.
 - * Il Server ritorna nello stato "Null" dopo aver inoltrato la risposta al Proxy.

Il compito del Server è quello di ricevere i messaggi dal Proxy, di elaborare una risposta e infine di inoltrare la risposta a quest'ultimo in modo che possa essere consegnata al Client.

4.4 Flusso di lavoro

1. All'inizio i muli incominciano a camminare casualmente tra il Client ed il Proxy.
2. Se il Client possiede un messaggio da inviare, entra nello stato di richiesta e attende il passaggio di un mulo nella sua zona, in modo da potergli consegnare il messaggio.
3. Quando un mulo arriva nella zona di un Client controlla se il Client è nello stato di richiesta, se questa condizione è vera allora il Mulo comincia a parlare con il Client.
4. Se il Mulo è stato inserito nella "black list" il Client può scegliere non deterministicamente di affidare o meno il messaggio al Mulo.
5. Se il Client si trova all'interno di una zona pericolosa, il Mulo può scegliere non deterministicamente di accettare o meno il messaggio dal Client.
6. Se il Client ha affidato il messaggio al Mulo e se il Mulo ha accettato il messaggio, quest'ultimo comincia a camminare per dirigersi verso il Proxy.
7. In questo momento può accadere che il Mulo perda il messaggio (la probabilità di perdere il messaggio aumenta se il mulo proviene da un Client che è all'interno di una zona poco sicura), se il Mulo perde il messaggio viene inserito nella "black list".
8. Se il Mulo riesce ad arrivare in prossimità del Proxy senza aver perso il messaggio, il Mulo consegna il messaggio al Proxy.
9. Se la consegna del Mulo è andata a buon fine (non ha perso il messaggio) e se prima della consegna il Mulo compariva all'interno della "black list", allora il Mulo viene cancellato dalla "black list".
10. Appena il Proxy riceve il messaggio dal Mulo, lo inoltra al Server, aspettando che il Server risponda al messaggio con una risposta.
11. Appena il Server riceve il messaggio inoltrato dal Proxy, elabora la risposta e la inoltra al Proxy.
12. Appena il Proxy riceve la risposta dal Server, entra nello stato di richiesta, attendendo un Mulo a cui poter affidare la risposta da consegnare al Client.
13. Appena un Mulo arriva in prossimità del Proxy, controlla se quest'ultimo ha necessità di spedire un messaggio, se il controllo dà esito positivo, il Mulo accetta il messaggio del Proxy.
14. Il mulo comincia quindi a camminare fino a raggiungere il Client.
15. Se durante il tragitto il Mulo perde il messaggio, viene inserito nella "black list". (la probabilità di perdere il messaggio aumenta se il Client da raggiungere è situato all'interno di una zona pericolosa).

16. Se il Mulo raggiunge il Client senza aver perso il messaggio, sotto pagamento consegna la risposta al Client. A questo punto se precedentemente il Mulo era stato inserito all'interno della "black list", viene rimosso dalla lista.

4.5 Test implementati

- È stato implementato il test per verificare se entrambi i Muli possiedono il messaggio nello stesso momento.
Il risultato dimostra che questa condizione non può accadere poiché il messaggio viene affidato ad un solo Mulo alla volta.
- È stato implementato il test per verificare se il portafoglio del Client può andare in negativo.
Il risultato dimostra che non è possibile perché quando il portafoglio del Client ha raggiunto la quota di zero, non può più fare richieste di spedire un messaggio.
- È stato implementato il test per verificare se il portafoglio del Mulo può superare il valore di 50.
Il risultato dimostra che questo non è possibile in quanto il Client ha una disponibilità di 50 monete con cui spedire i messaggi.
- È stato implementato il test per verificare se il portafoglio di un mulo è pieno e il portafoglio dell'altro mulo è vuoto.
Il risultato dimostra che questa condizione è possibile solo nel caso in cui un Mulo venga sempre assoldato a discapito dell'altro. (situazione di "fairness")
- È stato implementato il test per verificare se entrambi i Muli stanno camminando verso il Client per consegnargli la risposta.
Il risultato dimostra che questa condizione non può accadere poiché il messaggio viene affidato ad un solo Mulo alla volta.
- È stato implementato il test per verificare se entrambi i Muli stanno camminando verso il Proxy per consegnargli il messaggio.
Il risultato dimostra che questa condizione non può accadere poiché il messaggio viene affidato ad un solo Mulo alla volta.
- È stato implementato il test per verificare se il Client abbia sempre prima o poi un messaggio da inviare.
Il risultato dimostra che questa condizione non può accadere poiché il Client può finire i soldi oppure non avere un messaggio da inviare.
- È stato implementato il test per verificare se, una volta consegnato il messaggio, il Mulo ritorna a camminare prima o poi in modo causale in cerca di un nuovo lavoro.
Il risultato dimostra che il Mulo viene sempre pagato ogni volta che porta a termine il lavoro, quindi può ricominciare a camminare in modo causale in cerca di un nuovo lavoro.
- È stato implementato il test per verificare se quando il Client ha un messaggio da inviare, allora ci sarà sempre un Mulo con cui accordarsi.

Il risultato dimostra che questa condizione è falsa in quanto i Muli potrebbero non arrivare mai nella zona del Client, in quanto camminano in modo casuale.

- È stato implementato il test per verificare se, anche nelle migliori condizioni (quando il Client è nella zona sicura e i Muli non sono nella "Black list"), quando il Client ha un messaggio da inviare, allora ci sarà sempre un Mulo con cui accordarsi.

Il risultato dimostra che questa condizione è falsa in quanto i Muli potrebbero non arrivare mai nella zona del Client, in quanto camminano in modo casuale.

- È stato implementato il test per verificare se quando il Proxy ha un messaggio da inviare, allora ci sarà sempre un Mulo con cui accordarsi.

Il risultato dimostra che questa condizione è falsa in quanto i Muli potrebbero non arrivare mai nella zona del Proxy, in quanto camminano in modo casuale.

- È stato implementato il test per verificare se il Client ,dopo aver inviato un messaggio, riceverà sempre una risposta.

Il risultato dimostra che questa condizione è falsa in quanto i Muli potrebbero non arrivare mai nella zona del Proxy a ritirare la risposta da consegnare, perchè c'è la possibilità di perdere il messaggio.

- È stato implementato il test per verificare se quando il Proxy ha inoltrato il messaggio al Server, il Server risponde sempre al Proxy.

Il risultato dimostra che questa condizione è vera in quanto per costruzione il Server risponde sempre al Proxy.

- È stato implementato il test per verificare se quando il Mulo entra nello stato di consegna del messaggio, allora non ha perso quest'ultimo.

Il risultato dimostra che questa condizione è vera in quanto il Mulo non può entrare nello stato di consegna del messaggio se prima il messaggio è stato perso.