

Progetto DALI DISIM univAQ: Intelligent Autonomous Agents A.A. 2021/2022



Membri del gruppo		
Nome e cognome	Matricola	Indirizzo e-mail
Daniele Fossemò	280966	daniele.fossemo@student.univaq.it
Mattia Masci	282311	mattia.masci@student.univaq.it

Sommario

Specifica	2
Sequence/collaboration diagram	3
Agent types	4
Consumatore	4
Venditore	5
Produttore	6
Class diagram	7
Events/action tables	8
Consumatore	8
Venditore	9
Produttore	10
Screenshot esecuzione progetto	11
Istruzioni di installazione e repository github	19

Specifica

Il dominio scelto è quello di un insieme di agenti, detti *consumatori*, che richiedono, sulla base di alcune necessità, dei *beni* ad un agente detto *venditore*. Il venditore, quando arriva un ordine da un consumatore, memorizza quel bene come bene da servire più tardi e restituisce un ticket all'ordine del consumatore. Il ticket è un numero identificativo che viene incrementato di volta in volta dopo essere stato assegnato ad un ordine. Ogni 40 secondi il venditore chiede a tutti i consumatori se hanno un ticket, ed in caso affermativo di fornirgli l'identificativo associato. Il venditore dunque servirà l'ordine associato all'identificativo con valore più basso, e scarterà i restanti ticket, chiedendo ai relativi consumatori di aspettare il loro turno. Per servire, il consumatore controlla se il bene richiesto è disponibile in magazzino. In caso il bene richiesto fosse esaurito, chiede al consumatore di ritornare più tardi, e chiede poi ad un terzo agente, detto *produttore*, rifornimenti di quel bene. Se invece il bene è disponibile serve il consumatore. Una volta servito il consumatore controlla nuovamente se tale bene è finito in magazzino ed in caso affermativo chiede al *produttore* un rifornimento di quel bene.

Le necessità dei consumatori possono essere "*affamato*", "*assetato*" ed "*ammalato*", ed i beni richiesti sulla base di queste necessità sono rispettivamente "*cibo*", "*acqua*", e "*medicina*". Se un consumatore non ha nessuna di queste necessità, allora è "*a posto*", e finché non sorgerà una nuova necessità non farà niente.

Per lo sviluppo di questa specifica abbiamo assunto che il venditore serva un ordine alla volta, così come i consumatori ne richiedono una alla volta; inoltre i consumatori richiedono una sola quantità del bene di cui hanno bisogno. Infine si è assunto che il produttore abbia una quantità infinita dei beni che possono essere richiesti dal venditore.

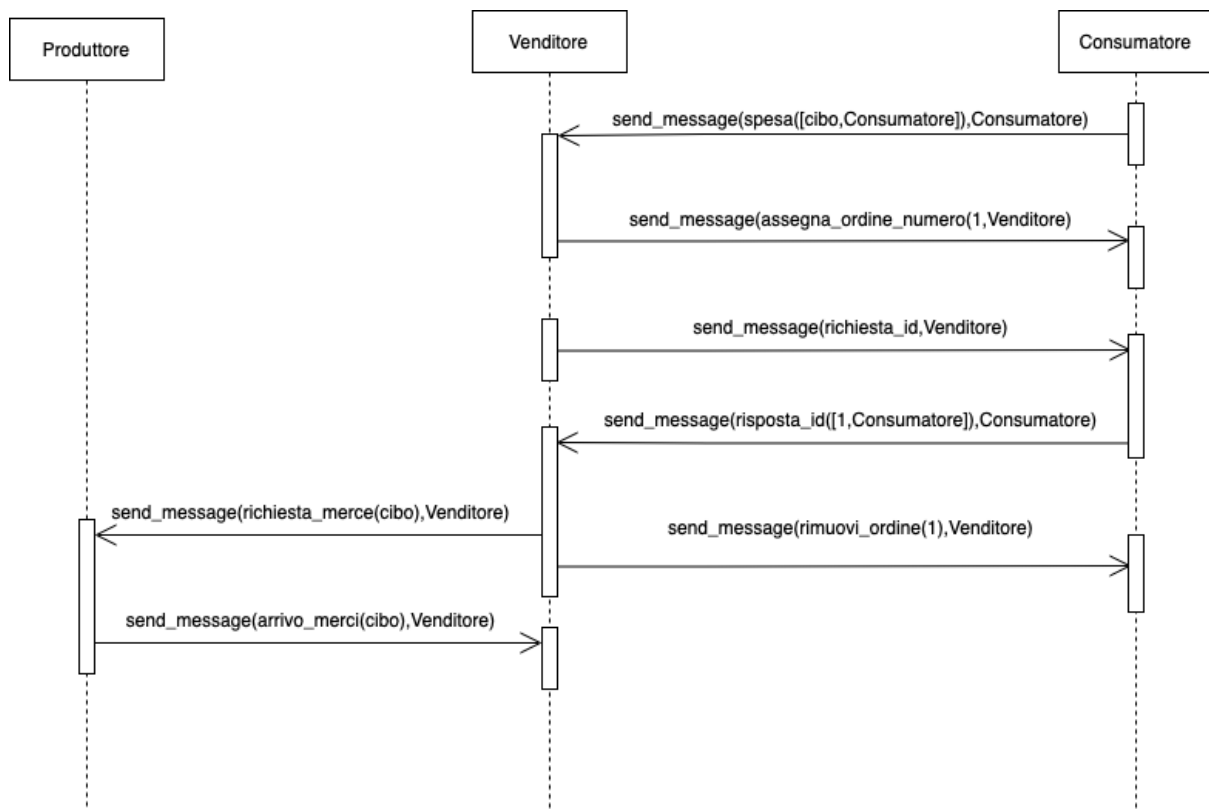
Sequence/collaboration diagram

Il seguente sequence diagram rappresenta un possibile esempio di esecuzione all'interno di un turno, ovvero l'intervallo di tempo all'interno del quale il venditore rimane in attesa di ordini, ai quali assegnerà il relativo ticket identificativo, comincia la sua routine in cui serve un cliente se ci sono richieste, per poi ricominciare da capo.

Al fine di mostrare una esecuzione facilmente comprensibile prendiamo in considerazione uno scenario in cui è presente un solo consumatore, in caso di più consumatori comunque il procedimento è analogo. Inoltre presentiamo l'esecuzione più articolata possibile, le alternative che possono avverarsi sono tutte realizzabili omettendo alcune parti di quella presa in esame.

In questo caso il consumatore è affamato e invia un messaggio al venditore chiedendogli del cibo, il venditore, da parte sua, assegna subito un ID all'ordine del consumatore e lascia trascorrere un intervallo di tempo per assicurarsi di ricevere ulteriori richieste nel caso ce ne fossero. Trascorso l'intervallo di tempo, il venditore richiede a tutti i consumatori eventuali identificativi dei ticket assegnati agli ordini richiesti, il consumatore preso in considerazione risponde inviando l'identificativo del ticket del proprio ordine e il venditore procede a effettuare tutti i controlli necessari e a servire la richiesta.

Nell'esempio, il venditore nota che servendo il cliente esaurirà le scorte della merce venduta e, per evitare di dover annullare eventuali ordini futuri riguardanti quella merce, invia al produttore una richiesta di rifornimento di cibo. Il produttore, in maniera analoga al venditore, recepisce la richiesta e dopo un certo quantitativo di tempo effettua i controlli necessari e invia al venditore i rifornimenti richiesti. Il venditore infine aggiorna l'inventario e chiude il turno.



Agent types

Consumatore¹

Ruolo: Consumatore
Descrizione: Questo ruolo richiede dei beni sulla base delle sue necessità e collabora con il venditore per ricevere questi beni.
Protocolli e attività: <ul style="list-style-type: none">- controllo_condizioni, ordine, stampa_richiesta, aggiorna, assegna_ordine_numero, ritorna_piu_tardi, richiesta_id, rimuovi_ordine, ritenta_ordine
Permessi: <ul style="list-style-type: none">- reads:<ul style="list-style-type: none">- id_ordine // id dell'ordine effettuato, se non c'è un ordine ha valore 0- condizione // la condizione corrente- fase // utilizzato per l'inizializzazione dell'agente- changes:<ul style="list-style-type: none">- id_ordine- condizione- fase
Responsabilità: <ul style="list-style-type: none">- Liveness: Consumatore = (controllo_condizioni. ((ordine. stampa_richiesta. assegna_ordine_numero.) aggiorna))^w. richiesta_id. ((rimuovi_ordine ritenta_ordine) ritorna_piu_tardi)- Safety: $id_ordine \geq 0$ $0 < condizione \leq 4$

¹ Per la definizione di queste tabelle è stata usata come fonte il seguente articolo: Wooldridge, M., Jennings, N. R., & Kinny, D. (2000). *The Gaia methodology for agentoriented analysis and design. Autonomous Agents and multi-agent systems*, 3(3), 285-312.

Citata all'interno delle slide di presentazione del progetto.

Venditore

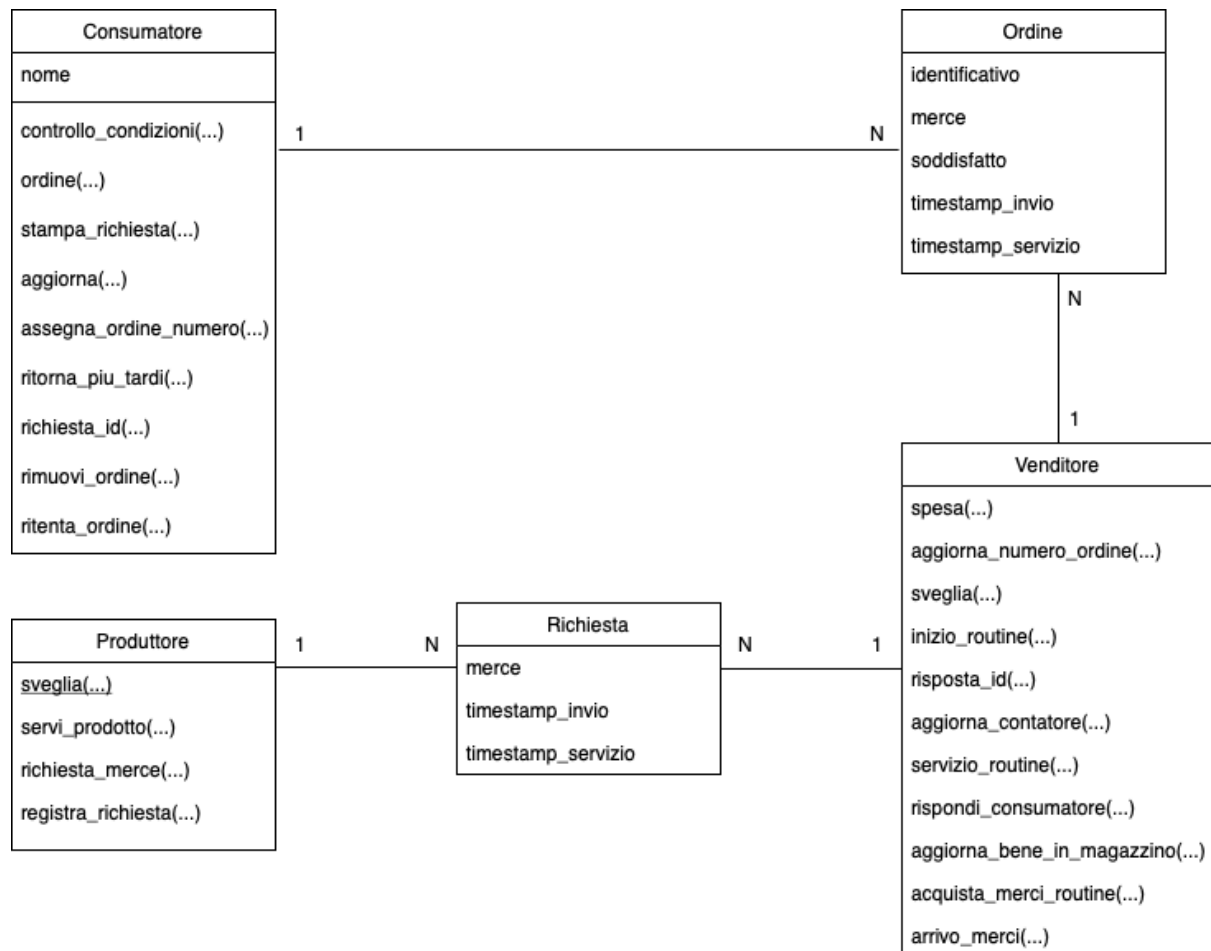
Ruolo: Venditore
Descrizione: Questo ruolo serve gli ordini dei consumatori, e richiede al produttore rifornimenti quando i beni in magazzino sono esauriti.
Protocolli e attività: <ul style="list-style-type: none"> - spesa, aggiorna_numero_ordine, sveglia, inizio_routine, risposta_id, aggiorna_contatore, servizio_routine, rispondi_consumatore, aggiorna_bene_in_magazzino, acquista_merci_routine, arrivo_merci
Permessi: <ul style="list-style-type: none"> - reads: <ul style="list-style-type: none"> - stato // utilizzato per l'inizializzazione dell'agente - ordine_numero // identificativo dei ticket da assegnare agli ordini - ordine // gli ordini che deve servire in futuro - pronto // utilizzato per gestire le fasi di routine dell'agente - contatore_risposte // conta le risposte durante la fase di richiesta degli id - bene_da_servire // l'ordine che sta servendo attualmente - bene_in_magazzino // i beni in magazzino - changes: <ul style="list-style-type: none"> - ordine_numero - ordine - pronto - contatore_risposte - bene_da_servire - bene_in_magazzino
Responsabilità: <ul style="list-style-type: none"> - Liveness: Venditore = (sveglia. inizio_routine. [risposta_id. (aggiorna_contatore)+. servizio_routine. [aggiorna_bene_in_magazzino. acquista_merci_routine]. rispondi_consumatore)^ω (spesa. aggiorna_numero_ordine)* (arrivo_merci)* - Safety: ordine_numero ≥ 0 0 ≤ contatore_risposte ≤ N dove N è il numero di consumatori

Produttore

Ruolo: Produttore
Descrizione: Questo ruolo fornisce i prodotti al venditore quando quest'ultimo li finisce in magazzino.
Protocolli e attività: <ul style="list-style-type: none">- sveglia, servi_prodotto, richiesta_merce, registra_richiesta
Permessi: <ul style="list-style-type: none">- reads:<ul style="list-style-type: none">- stato // utilizzato per l'inizializzazione dell'agente- richiesta // richieste arrivate dal venditore- contatore_richiesta // contatore di quante richieste sono arrivate- contatore_servito // contatore di quante richieste sono state servite- changes:<ul style="list-style-type: none">- richiesta- contatore_richiesta- contatore_servito
Responsabilità: <ul style="list-style-type: none">- Liveness: Produttore = (sveglia. servi_prodotto)^ω (richiesta_merce. registra_merce)*- Safety: contatore_richiesta ≥ 0 0 ≤ contatore_servito ≤ contatore_richiesta

Class diagram

Di seguito riportiamo le relazioni tra le entità fondamentali del dominio preso in considerazione.



Le varie componenti da cui è composto il class diagram sono già state descritte nelle sezioni precedenti, successivamente chiariamo alcuni nuovi concetti.

L'attributo 'merce' che compare nelle classi 'Richiesta' e 'Ordine' indica il tipo di merce richiesto (cibo, acqua, medicina).

L'attributo 'soddisfatto' nella classe 'Ordine' è un booleano che indica se l'ordine è stato effettivamente servito oppure per qualche motivo è stato annullato.

Events/action tables

Consumatore

Nome	Descrizione
controllo_condizioni	Il consumatore controlla se in un dato momento è presente una qualche necessità per la quale è necessario effettuare un ordine. Nel caso l'agente è "a posto" non fa niente. Questo evento è stato impostato tramite direttive per essere controllato ogni 20 secondi.
ordine(X)	Il consumatore invia un ordine al venditore. Se ha già inviato un ordine precedentemente a cui ancora non è stato servito aspetta prima di essere servito.
stampa_ordine(X)	Il consumatore stampa in output l'ordine appena effettuato.
assegna_ordine_numero(X)	Inserisce nella conoscenza il numero identificativo del ticket associato all'ordine appena effettuato.
ritorna_piu_tardi	Risponde affermativamente al venditore quando quest'ultimo gli chiede di ritornare più tardi perché è impegnato a servire un altro consumatore.
richiesta_id	Il consumatore fornisce al venditore l'identificativo associato all'ordine per cui sta aspettando di essere servito, così che quest'ultimo lo possa comparare a quello degli altri consumatori e definire chi servire. Nel caso in cui il consumatore non abbia alcun ordine per cui sta aspettando, restituisce l'identificativo 0.
rimuovi_ordine(X)	Il consumatore rimuove dalla conoscenza l'ordine per cui è stato appena servito ed aggiorna la sua condizione.
ritenta_ordine(X)	Risponde affermativamente al venditore quando quest'ultimo gli chiede di ritornare più tardi perché il bene richiesto è esaurito in magazzino.
aggiorna	Nel caso in cui con <i>controllo_condizioni</i> l'agente vede di essere "a posto", aggiorna le sue condizioni per verificare se lo sia ancora o meno.

Venditore

Nome	Descrizione
spesa	Quando arriva un ordine da parte di un consumatore, il venditore assegna un ticket all'ordine che restituisce al consumatore, il quale dovrà inserirlo nella sua conoscenza. Successivamente salva l'associazione ticket-ordine nella propria conoscenza per poterlo riprendere più tardi quando lo deve servire.
aggiorna_numero_ordine(X, Y)	Dopo aver assegnato un ticket all'ordine appena ricevuto, incrementa di uno questo ticket.
sveglia	Ogni 40 secondi il venditore inizia la propria routine.
inizio_routine	Il venditore inizia la routine inviando a tutti gli agenti un messaggio in cui chiede l'identificativo dei loro ordini.
risposta_id(X)	<p>Il venditore, quando riceve da un consumatore l'identificativo X associato al suo ordine, effettua dei controlli. Questi controlli li effettua tenendo aggiornato un contatore Q che segna quante risposte sono arrivate dai consumatori. Ponendo che il numero dei consumatori sia N, i controlli sono:</p> <ul style="list-style-type: none"> • Se il venditore non ha nella conoscenza un bene come "<i>bene_da_servire</i>" e se: <ul style="list-style-type: none"> ○ $X > 0$ allora inserisce nella conoscenza l'ordine come "<i>bene_da_servire</i>". ○ $X == 0$ allora scrive in output di aver ricevuto identificativo uguale a 0 (l'identificativo 0 viene usato dai consumatori per dire che non hanno ordini in sospeso). ○ $Q < N$ allora incrementa di uno Q. ○ $Q == N$, allora comincia la fase di servizio dell'ordine che ha passato. • Se il venditore ha nella conoscenza un bene come "<i>bene_da_servire</i>" con identificativo A, e se: <ul style="list-style-type: none"> ○ $0 < X < A$ allora sostituisce la conoscenza del "<i>bene_da_servire</i>" con identificativo A con quello con identificativo X. ○ $X == 0$ allora scrive in output di aver ricevuto identificativo uguale a 0. ○ $X > A$ allora scrive in output di aver ricevuto identificativo con priorità minore rispetto ad un altro che deve ancora servire, lasciando aspettare il consumatore con ordine con identificativo X. ○ $Q < N$ allora incrementa di uno Q ○ $Q == N$, allora comincia la fase di servizio dell'ordine che ha passato

aggiorna_contatore(X,Y)	Il venditore aggiorna il contatore delle risposte ricevute durante la fase di ricezione degli identificativi degli ordini relativi ai consumatori.
servizio_routine	Una volta scelto l'ordine da servire, il venditore comincia effettivamente a servire il consumatore collegato all'ordine. Se nessun ordine è arrivato nella fase precedente il venditore non fa nulla e torna a in attesa di ordini, altrimenti controlla se il bene richiesto è presente in magazzino. Se il bene richiesto non è presente in magazzino allora chiede al consumatore di aspettare e richiede al produttore un rifornimento di quel bene (il ticket del consumatore è ancora valido). Se invece il bene è presente in magazzino allora serve il consumatore, dopodiché controlla se il bene appena servito è esaurito e, nel caso, richiede al produttore un rifornimento di quel bene.
rispondi_consumatore(X,Y,Z)	Usata durante servizio_routine per comunicare al consumatore se deve ritentare l'ordine più tardi perché il bene non è presente in magazzino o se il servizio è andato a buon fine.
aggiorna_bene_in_magazzino(X,Y,Z)	Aggiorna la quantità in magazzino del bene appena servito.
acquista_merci_routine(X)	Richiede al produttore un rifornimento del bene X.
arrivo_merci(X)	Quando riceve le merci richieste precedentemente al produttore aggiorna la conoscenza relativa alla quantità disponibile in magazzino di quel bene.

Produttore

Nome	Descrizione
sveglia	Ogni 20 secondi il produttore controlla se ha dei prodotti da fornire al venditore.
servi_prodotto(X,Y)	Nel caso abbia prodotti da fornire al venditore li serve e rimuove dalla conoscenza base il fatto di dover fornire quel prodotto al venditore. Nel caso abbia più prodotti da fornire, dà la precedenza a quelli con identificativo associato minore.
richiesta_merce(X)	Quando riceve dal venditore una richiesta di rifornimento per un prodotto scrive in output di aver ricevuto la richiesta e la registra.
registra_richiesta(X,Y,Z)	Registra la richiesta con identificativo Y, e incrementa quest'ultimo di uno.

Screenshot esecuzione progetto

1) Quando il venditore comincia la sua routine di controllo, chiede ai consumatori gli identificatori dei loro ordini. I consumatori non hanno ordini, quindi restituiscono l'identificativo 0 e stampano in output di non aver effettuato ordini. Siccome il venditore vede che non sono arrivati ordini, allora non fa niente.

```
consumatore1.txt
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv'
* [To] - singleton variables
% compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv in module user, 33 msec 71328 bytes
..... Activated Agent consumatore1 .....
Non ho effettuato ordini.
[]

consumatore2.txt
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv'
* [To] - singleton variables
% compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv in module user, 30 msec 71328 bytes
..... Activated Agent consumatore2 .....
Non ho effettuato ordini.
[]

produttore.txt
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
% compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv in module user, 25 msec 66152 bytes
..... Activated Agent produttore .....
[]

venditore.txt
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv'
* [To] - singleton variables
% compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv in module user, 34 msec 77984 bytes
..... Activated Agent venditore .....
Eccomi! Inizio il turno.
Controllo ordini...
Non ho ricevuto ordini, chiudo il turno.
.....
```

2) Poco più tardi entrambi i consumatori richiedono una medicina mentre il venditore rimane in attesa di ordini. Il fatto che entrambi osservino la stessa condizione “malattia”, e quindi richiedano una medicina al venditore è una casualità. Come sarà fatto vedere successivamente possono avere necessità diverse nello stesso momento, in quanto sono due diverse istanze dello stesso ruolo. Il venditore, quando riceve questi ordini, assegna il ticket ai consumatori e torna in attesa di ordini. Il ticket è assegnato in ordine temporale, quindi poiché il consumatore 1 ha richiesto l'ordine prima del consumatore 2, allora assegnerà al consumatore 1 ticket con identificativo 1 e al consumatore 2 ticket con identificativo 2. Più è basso l'identificativo del ticket maggiore sarà la priorità del relativo consumatore di essere servito.

```

consumatore1.txt
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv'
% compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv in module user, 33 msec 71328 bytes
..... Activated Agent consumatore1 .....

Non ho effettuato ordini.

Salve, ho un ordine da inviare.
Ho ordinato: medicina.
L ID del mio ordine e: 1.[]

consumatore2.txt
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv'
% compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv in module user, 30 msec 71328 bytes
..... Activated Agent consumatore2 .....

Non ho effettuato ordini.

Salve, ho un ordine da inviare.
Ho ordinato: medicina.
L ID del mio ordine e: 2.[]

produttore.txt
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv'
% compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv in module user, 25 msec 65152 bytes
..... Activated Agent produttore .....

venditore.txt
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv'
* [To] - singleton variables
* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv'
% compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv in module user, 34 msec 77984 bytes
..... Activated Agent venditore .....

Eccomi! Inizio il turno.
Controllo ordini...
Non ho ricevuto ordini, chiudo il turno.

Ho ricevuto delle richieste mentre dormivo.
Ho ricevuto delle richieste mentre dormivo.

```

3) Quando il venditore ricomincia la routine di controllo chiede nuovamente ai consumatori di fornire gli identificatori degli ordini. I consumatori a questo punto hanno effettivamente degli ordini in sospeso, quindi restituiscono i corrispettivi identificatori. Poiché il consumatore 1 ha ticket con identificativo minore viene servito per primo. Il venditore serve il consumatore 1, osserva che ha ancora medicine in magazzino e quindi non richiede nulla al produttore, e torna a rimanere in attesa di ordini.

consumatore1.txt	consumatore2.txt
<pre>* [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv in module user, 33 msec 71328 bytes Activated Agent consumatore1</pre> <p>Non ho effettuato ordini.</p> <p>Salve, ho un ordine da inviare.</p> <p>Ho ordinato: medicina.</p> <p>L ID del mio ordine e: 1.</p> <p>Grazie per il servizio, a presto.</p>	<pre>* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv in module user, 30 msec 71328 bytes Activated Agent consumatore2</pre> <p>Non ho effettuato ordini.</p> <p>Salve, ho un ordine da inviare.</p> <p>Ho ordinato: medicina.</p> <p>L ID del mio ordine e: 2.</p>
produttore.txt	venditore.txt
<pre>* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv in module user, 25 msec 65152 bytes Activated Agent produttore</pre>	<pre>* [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/venditore.plv in module user, 34 msec 77384 bytes Activated Agent venditore</pre> <p>Eccomi! Inizio il turno.</p> <p>Controllo ordini...</p> <p>Non ho ricevuto ordini, chiudo il turno.</p> <p>Ho ricevuto delle richieste mentre dormivo.</p> <p>Ho ricevuto delle richieste mentre dormivo.</p> <p>Eccomi! Inizio il turno.</p> <p>Controllo ordini...</p> <p>Servo l ordine di: consumatore1 con id: 1.</p> <p>Fatto!</p> <p>Chiudo il turno.</p>

4) Poco più tardi il soddisfacimento della richiesta di consumatore 1, questo si ripresenta nuovamente con un ordine, richiedendo dell'acqua. Il venditore gli assegna ticket con identificativo 3. Successivamente quando il venditore ricomincia la routine di controllo serve il consumatore 2. A questo punto è da notare che dopo essere stato servito il consumatore 2 è "a posto", diversamente dal consumatore 1 che invece ha richiesto dell'acqua.

consumatore1.txt	consumatore2.txt
<pre>* [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv in module user, 33 msec 71328 bytes Activated Agent consumatore1</pre> <p>Non ho effettuato ordini.</p> <p>Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e: 1. Grazie per il servizio, a presto.</p> <p>Salve, ho un ordine da inviare. Ho ordinato: acqua. L ID del mio ordine e: 3.</p>	<pre>* [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv in module user, 30 msec 71328 bytes Activated Agent consumatore2</pre> <p>Non ho effettuato ordini.</p> <p>Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e: 2. Grazie per il servizio, a presto.</p> <p>Per adesso non ho bisogno di niente.</p>
produttore.txt	venditore.txt
<pre>* Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv in module user, 25 msec 65152 bytes Activated Agent produttore</pre>	<pre>..... Activated Agent venditore</pre> <p>Eccomi! Inizio il turno. Controllo ordini... Non ho ricevuto ordini, chiudo il turno.</p> <p>Ho ricevuto delle richieste mentre dormivo. Ho ricevuto delle richieste mentre dormivo.</p> <p>Eccomi! Inizio il turno. Controllo ordini... Servo l ordine di: consumatore1 con id: 1. Fatto! Chiudo il turno.</p> <p>Ho ricevuto delle richieste mentre dormivo.</p> <p>Eccomi! Inizio il turno. Controllo ordini... Servo l ordine di: consumatore2 con id: 2. Fatto! Chiudo il turno.</p>

5) A questo punto viene servito il consumatore 1, in quanto aveva ancora un ordine in sospeso. Dopo aver servito il consumatore 1, il venditore nota di non avere più acqua, quindi richiede al produttore un rifornimento. Il produttore riceve la notifica e stampa di aver ricevuto la richiesta di rifornimento per l'acqua.

consumatore1.txt	consumatore2.txt
<pre>* [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv in module user, 33 msec 71328 bytes Activated Agent consumatore1 Non ho effettuato ordini. Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e': 1. Grazie per il servizio, a presto. Salve, ho un ordine da inviare. Ho ordinato: acqua. L ID del mio ordine e': 3. Grazie per il servizio, a presto.]</pre>	<pre>oggettoFinal/work/consumatore2.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv in module user, 30 msec 71328 bytes Activated Agent consumatore2 Non ho effettuato ordini. Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e': 2. Grazie per il servizio, a presto. Per adesso non ho bisogno di niente. Non ho effettuato ordini.]</pre>
produttore.txt	venditore.txt
<pre>* [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv in module user, 25 msec 65152 bytes Activated Agent produttore Ho ricevuto la richiesta di rifornimenti per il bene: acqua.]</pre>	<pre>..... Eccomi! Inizio il turno. Controllo ordini... Servo l ordine di: consumatore1 con id: 1. Fatto! Chiudo il turno. Ho ricevuto delle richieste mentre dormivo. Eccomi! Inizio il turno. Controllo ordini... Servo l ordine di: consumatore2 con id: 2. Fatto! Chiudo il turno. Eccomi! Inizio il turno. Controllo ordini... Servo l ordine di: consumatore1 con id: 3. Fatto! Chiudo il turno.]</pre>

6) Mentre il venditore rimane in attesa di ordini, entrambi i consumatori richiedono dei nuovi beni, prima il consumatore 1 richiede una medicina, e dopo il consumatore 2 dell'acqua. Subito dopo il produttore invia i rifornimenti di acqua al venditore.

consumatore1.txt	consumatore2.txt
<pre> oggettoFinal/work/consumatore1.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/Pr oggettoFinal/work/consumatore1.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consum matore1.plv in module user, 33 msec 71328 bytes Activated Agent consumatore1 Non ho effettuato ordini. Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e: 1. Grazie per il servizio, a presto. Salve, ho un ordine da inviare. Ho ordinato: acqua. L ID del mio ordine e: 3. Grazie per il servizio, a presto. Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e: 4. </pre>	<pre> * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/Pr oggettoFinal/work/consumatore2.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/Pr oggettoFinal/work/consumatore2.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consum matore2.plv in module user, 30 msec 71328 bytes Activated Agent consumatore2 Non ho effettuato ordini. Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e: 2. Grazie per il servizio, a presto. Per adesso non ho bisogno di niente. Non ho effettuato ordini. Salve, ho un ordine da inviare. Ho ordinato: acqua. L ID del mio ordine e: 5. </pre>
produttore.txt	venditore.txt
<pre> * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/Pr oggettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/Pr oggettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/Pr oggettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/Pr oggettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/Pr oggettoFinal/work/produttore.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produ ttore.plv in module user, 25 msec 65152 bytes Activated Agent produttore Ho ricevuto la richiesta di rifornimenti per il bene: acqua. Invio rifornimenti di: acqua al venditore. </pre>	<pre> Chiudo il turno. ----- Ho ricevuto delle richieste mentre dormivo. ----- Eccomi! Inizio il turno. Controllo ordini... Servo l ordine di: consumatore2 con id: 2. Fatto! Chiudo il turno. ----- Eccomi! Inizio il turno. Controllo ordini... Servo l ordine di: consumatore1 con id: 3. Fatto! Chiudo il turno. ----- Ho ricevuto delle richieste mentre dormivo. Ho ricevuto delle richieste mentre dormivo. ***** Sono arrivati rifornimenti di acqua. ***** </pre>

7) Il venditore serve il consumatore 1, osserva che sono finite le medicine e quindi le richiede al produttore.

consumatore1.txt	consumatore2.txt
<pre>oggettoFinal/work/consumatore1.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore1.plv in module user, 33 msec 71328 bytes Activated Agent consumatore1</pre> <p>Non ho effettuato ordini.</p> <p>Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e': 1. Grazie per il servizio, a presto.</p> <p>Salve, ho un ordine da inviare. Ho ordinato: acqua. L ID del mio ordine e': 3.</p> <p>Grazie per il servizio, a presto.</p> <p>Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e': 4. Grazie per il servizio, a presto.</p>	<pre>oggettoFinal/work/consumatore2.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv in module user, 30 msec 71328 bytes Activated Agent consumatore2</pre> <p>Non ho effettuato ordini.</p> <p>Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e': 2.</p> <p>Grazie per il servizio, a presto.</p> <p>Per adesso non ho bisogno di niente.</p> <p>Non ho effettuato ordini.</p> <p>Salve, ho un ordine da inviare. Ho ordinato: acqua. L ID del mio ordine e': 5.</p>
produttore.txt	venditore.txt
<pre>* [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv' % compiled /home/daniele/Scrivania/DALI-master/Examples/ProgettoFinal/work/produttore.plv in module user, 25 msec 65152 bytes Activated Agent produttore</pre> <p>Ho ricevuto la richiesta di rifornimenti per il bene: acqua. Invio rifornimenti di: acqua al venditore.</p> <p>Ho ricevuto la richiesta di rifornimenti per il bene: medicina.</p>	<pre>Chiudo il turno. ----- Eccomi! Inizio il turno. Controllo ordini... Servo l ordine di: consumatore1 con id: 3. Fatto! Chiudo il turno. ----- Ho ricevuto delle richieste mentre dormivo. Ho ricevuto delle richieste mentre dormivo. ***** Sono arrivati rifornimenti di acqua. ***** Eccomi! Inizio il turno. Controllo ordini... Servo l ordine di: consumatore1 con id: 4. Fatto! Chiudo il turno.</pre>

8) il venditore riceve un ordine di nuovo dal consumatore 1, riceve poi i rifornimenti precedentemente richiesti dal produttore, e dopo ricomincia la routine di controllo per servire il consumatore 2. Da questo punto in poi gli agenti continueranno in maniera analoga a come si sono comportati fino ad ora.

consumatore1.txt	consumatore2.txt
<pre> Activated Agent consumatore1 Non ho effettuato ordini. Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e: 1. Grazie per il servizio, a presto. Salve, ho un ordine da inviare. Ho ordinato: acqua. L ID del mio ordine e: 3. Grazie per il servizio, a presto. Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e: 4. Grazie per il servizio, a presto. Salve, ho un ordine da inviare. Ho ordinato: acqua. L ID del mio ordine e: 6. </pre>	<pre> oggettoFinal/work/consumatore2.plv' % compiled /home/daniele/Scrivanis/DALI-master/Examples/ProgettoFinal/work/consumatore2.plv in module user, 30 msec 71328 bytes Activated Agent consumatore2 Non ho effettuato ordini. Salve, ho un ordine da inviare. Ho ordinato: medicina. L ID del mio ordine e: 2. Grazie per il servizio, a presto. Per adesso non ho bisogno di niente. Non ho effettuato ordini. Salve, ho un ordine da inviare. Ho ordinato: acqua. L ID del mio ordine e: 5. Grazie per il servizio, a presto. </pre>
produttore.txt	venditore.txt
<pre> * Approximate lines: 1-1, file: '/home/daniele/Scrivanis/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivanis/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivanis/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivanis/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables * Approximate lines: 1-1, file: '/home/daniele/Scrivanis/DALI-master/Examples/ProgettoFinal/work/produttore.plv' * [To] - singleton variables % compiled /home/daniele/Scrivanis/DALI-master/Examples/ProgettoFinal/work/produttore.plv in module user, 25 msec 65152 bytes Activated Agent produttore Ho ricevuto la richiesta di rifornimenti per il bene: acqua. Invio rifornimenti di: acqua al venditore. Ho ricevuto la richiesta di rifornimenti per il bene: medicina. Invio rifornimenti di: medicina al venditore. </pre>	<pre> Ho ricevuto delle richieste mentre dormivo. ***** Sono arrivati rifornimenti di acqua. ***** Eccomi! Inizio il turno. Controllo ordini... Servo l ordine di: consumatore1 con id: 4. Fatto! Chiudo il turno. Ho ricevuto delle richieste mentre dormivo. ***** Sono arrivati rifornimenti di medicina. ***** Eccomi! Inizio il turno. Controllo ordini... Servo l ordine di: consumatore2 con id: 5. Fatto! Chiudo il turno. </pre>

Istruzioni di installazione e repository github

Il progetto è stato sviluppato in sistemi operativi linux ubuntu e MacOS utilizzando la versione 4.7.1 di sicstus.

Nella repository è già presente la cartella src, ed essa è quella riferita dal file startmas.sh. In quest'ultimo file va aggiornato il percorso della variabile SICSTUS_HOME in base a dove è stato installato sicstus ed in base alla versione installata. Infine la riga 18 del file startmas.sh, ossia:

```
# rm -f work/* remove everything if you want to clear agent history
```

deve rimanere con il # all'inizio, ossia commentata. Questo perché sono state modificate le direttive riguardo alcuni eventi interni degli agenti. Togliere il # all'inizio di questa riga potrebbe influire sul corretto funzionamento degli agenti.

La repository github di questo progetto è presente al link:

<https://github.com/DanieleF198/IAA-DALproject>