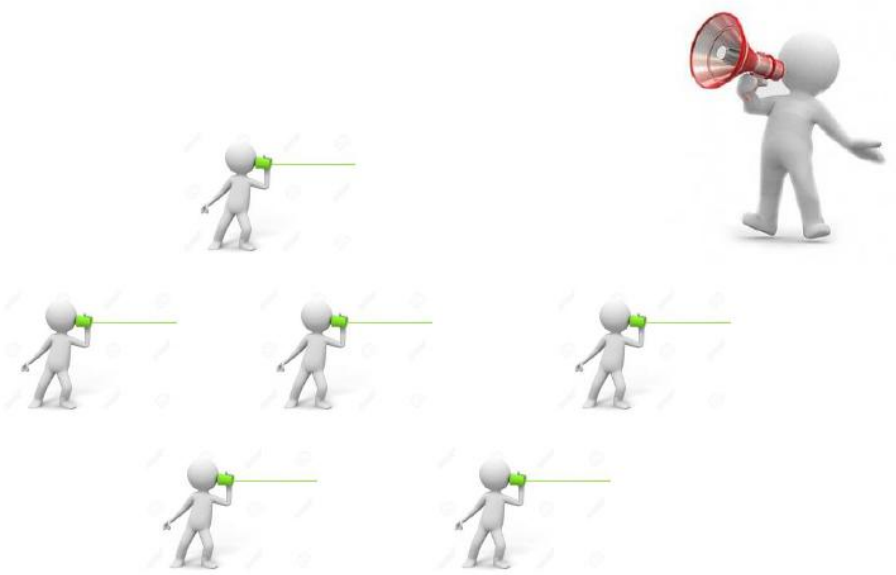
AGENTI INTELLIGENTI a.a 2016/2017



CONTRACT NET PROTOCOL

Studente:

Gabriella D'Andrea – 241568

Professore:

Giovanni De Gasperis

OVERVIEW

Per lo svolgimento del progetto si è deciso di implementare in DALI un esempio di esecuzione di uno dei più noti protocolli, utilizzati per la distribuzione dei compiti in sistemi multi-agente: il Contract Net Protocol. Tale protocollo è costituito da un insieme di agenti software, che formano il "Contract-Net" e ognuno di essi può, in momenti diversi o per compiti diversi, essere un Manager o un Seller.

Gli agenti di cui si parla sono principalmente tre:

- Organizer
- Manager
- Seller

L'agente Organizer invia periodicamente un task ad un agente Manager che, a sua volta inoltra il messaggio a tutti i Seller i quali valutano l'offerta e, in base alle proprie capacità, decidono se eseguire il task proposto oppure rifiutarlo. In particolare:

- se l'agente possiede i requisiti minimi per svolgere il task, i Seller inviano un'offerta al manager, indicando il profitto che vorrebbero ricevere;
- se l'agente non è in grado di soddisfare tutti i requisiti richiesti, i Seller rispondono con un messaggio di rifiuto.

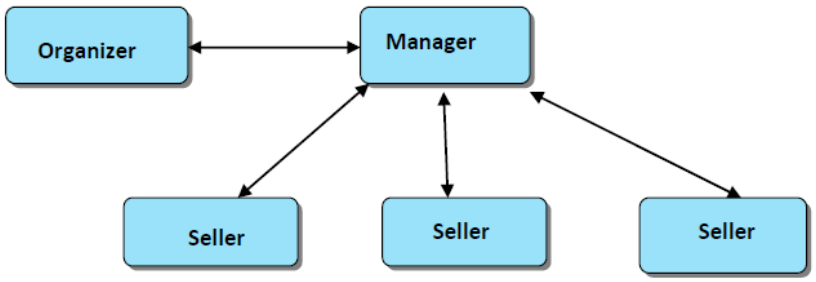


Figura 1 Rappresentazione scambio informativo

L'agente Manager dopo aver ricevuto una risposta da parte di tutti gli agenti Seller ha due possibilità:

1. *Almeno un agente soddisfa i requisiti*: in questo caso il Manager assegna il task all'agente che ha proposto un costo minore, inviandogli un messaggio di accettazione e inviando, di conseguenza, un messaggio di rifiuto a tutti agli altri agenti. Una volta completato il task, l'agente selezionato invia un messaggio per avvisare il Manager dell'avvenuta esecuzione.
2. *Nessun agente è in grado di eseguire il task*: se non vi è alcuna offerta per il task corrente si passa al successivo.

L'esecuzione termina quando l'agente Organizer non ha altri task da proporre all'agente Manager.

La comunicazioni tra gli agenti possono essere suddivise in 3 step:

1. **Announce**: step nel quale è previsto l'invio di un messaggio contenente le specifiche del task da eseguire, indicandone requisiti e vincoli.
2. **Bid**: step nel quale gli agenti che ricevono l'annuncio possono decidere se rifiutarlo o proporsi per eseguirlo,ovviamente l'agente prenderà la sua decisione basandosi sui vincoli ed i requisiti indicati nell'annuncio.
3. **Valutazione**: il Manager valuterà tutte le offerte ricevute ed assegnerà il task all'agente che meglio sembra conformarsi con i requisiti richiesti, ovviamente tale scelta sarà comunicata poi a tutti gli agenti che hanno formulato l'offerta.

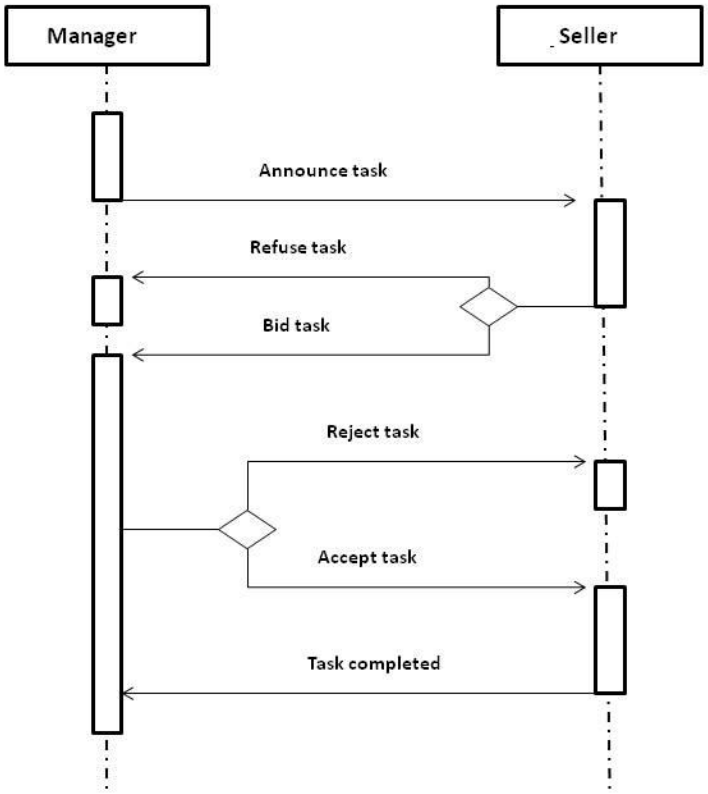


Figura 2 Scambio di messaggi tra Manager e Seller

CASO DI STUDIO

In questo progetto si è pensato di sviluppare una rivisitazione dell'applicazione "just-eat", dove si è preso in considerazione l'agente Organizer che riceve una serie di ordini da possibili utenti e li "inoltra" successivamente all'agente Manager, il quale decide il migliore "ristoratore" da cui acquistare il prodotto ordinato.

Al fine di implementare il caso di studio (come è possibile notare nella figura sottostante), si è deciso che ciascun utente potesse ordinare sei tipi di prodotti:

1. antipasto;
2. primi_piatti;
3. secondi_piatti;
4. contorni;
5. dolce;
6. bevanda;

ciascun ordine, inoltre, è composto da diversi parametri:

- Nome del Prodotto;

- Quantità richiesta espressa in termini di porzioni;

- Tempo indicante i minuti necessari per consegnare l'ordine;

- Prezzo che rappresenta il prezzo massimo che l'utente è disposto a pagare;

```
:- dynamic numOrder/1.
:- dynamic numDelivered/1.
:- dynamic numNotDelivered/1.
:- dynamic totOrder/1.
:- dynamic order/5.
:- dynamic on/1.
:- dynamic stop/1.

on(0).
totOrder(0).
numOrder(0).
numDelivered(0).
numNotDelivered(0).
stop(0).
order(antipasto,2,30,15,0).
order(primi_piatti, 3,10,30,0).
order(secondi_piatti, 3,20,30,0).
order(contorni, 2,25,20,0).
order(dolce, 1,5,15,0).
order(bevanda, 3,10,15,0).


activateApplicationE:> retract(on(0)), assert(on(1)),
                                    write('*****APPLICAZIONE AVVIATA CON SUCCESSO*****'),nl,nl,nl, sleep(2).


requestOrder:- on(1),stop(0),order(_,_,_,_,0).
requestOrderI:>         order(N,Q,T,M,0),
                                messageA(agentManager, send_message(newOrder(N,Q,T,M), organizer)),
                                numOrder(Num), N1 is Num+1,
                                retractall(numOrder(_)), assert(numOrder(N1)),
                                write('ORDINE UTENTE: '), write(N),nl,
                                write('RICHIESTA INVIATA CON SUCCESSO AL MANAGER'),nl,nl,
                                retractall(stop(_)), assert(stop(1)).


closeApplication:- totOrder(6), on(1).
closeApplicationI:>     nl, write('*****APPLICAZIONE TERMINATA CON SUCCESSO*****'), nl,
                                write(' Numero di ordini soddisfatti: '), numDelivered(Num1),
                                numNotDelivered(Num2), write(Num1/6),
                                messageA(agentManager, send_message(closeApplication('end'), organizer)),
                                retractall(on(_)), assert(on(0)).
```

Figura 3 Breve estratto del codice di "agentTypeOrganizer.txt" riportante i diversi ordini con i relativi parametri.

In tale implementazione sono previsti quattro Seller , ognuno dei quali ha dei vincoli sui tempi di consegna e quantità richieste.

Ciascun agente implementato si compone di eventi interni/esterni elencati nelle tabelle sottostanti:

- ***Organizer:***

Nome Evento	Descrizione
ActivateApplicationE	Messaggio inviato dall'utente per avviare l'applicazione
RequestOrderI	Messaggio inviato al Manager contenente l'ordine richiesto; tale messaggio, ovviamente, verrà inviato solo se c'è almeno un ordine da servire e se, e solo se, l'applicazione è ancora in esecuzione
orderDeliveredE	Messaggio ricevuto dal Manager con il quale si comunica che un determinato ordine è stato soddisfatto.
orderNotFoundE	Messaggio ricevuto dal Manager dove viene comunicato il fatto che nessun agente è in grado di soddisfare l'ordine richiesto.
closeApplicationI	Messaggio inviato al Managrer dove si comunica la chiusura dell'applicazione e riepilogo dell'attività (es: Numero di ordini soddisfatti:4/6). Prima di chiudere l'applicazione, però, è necessario verificare che tutti gli ordini siano stati gestiti.

- ***Manager:***

Nome Evento	Descrizione
newOrderE	Messaggio ricevuto dall'Organizer contenente l'ordine da soddisfare, successivamente, tale messaggio verrà inoltrato a tutti gli agenti Seller.
offerFromSellerE	Messaggio ricevuto da un agente Seller , contenente la proposta per soddisfare un ordine.

assignOrderToSellerI	Invio di un messaggio di accettazione all'agente selezionato per soddisfare l'ordine. Prima di inviare tale messaggio, però, si controlla sempre che sia arrivata una risposta (positiva/negativa) da tutti gli agenti Seller e che vi sia almeno un'offerta valida per soddisfare l'ordine.
orderNotSatisfableI	Invio di un messaggio all'Organizer per comunicare che nessun agente è in grado di soddisfare un particolare ordine.
orderAssignedE	Messaggio inviato da un Seller con il quale si comunica che l'ordine a lui assegnato è stato soddisfatto con successo.
refuseFromSellerE	Messaggio inviato da un Seller con il quale comunica di non avere i requisiti necessari per soddisfare l'ordine richiesto.
closeApplicationE	Messaggio ricevuto dall'Organizer con il quale viene comunicata la chiusura dell'applicazione in corso.

- ***Seller:***

Nome Evento	Descrizione
newOrderE	Messaggio ricevuto dall'Organizer contenente l'ordine da soddisfare con i relativi requisiti.
checkRequiredA	Azione scatenata come conseguenza dell'arrivo del messaggio "newOrder", in tale azione si verifica se l'agente è in grado o meno di soddisfare i requisiti richiesti per un determinato ordine. A seguito di tale controllo viene formulato un messaggio di risposta: 1) offerFromSeller: nel caso in cui il Seller ritiene di riuscire a soddisfare i requisiti richiesti per l'ordine; 2) refuseFromSeller: nel caso in cui il Seller ritiene di NON riuscire a soddisfare i requisiti richiesti per l'ordine;

offerRejectedE	Messaggio ricevuto dal Manager con il quale si informa il Seller che la sua offerta per soddisfare un ordine è stata rifiutata.
offerAcceptedE	Messaggio ricevuto dal Manager con il quale si informa il Seller che la sua offerta per soddisfare un ordine è stata accettata.
closeApplicationE	Messaggio ricevuto dal Manager con il quale viene comunicata la chiusura dell'applicazione.

AVVIO APPLICAZIONE:

Per avviare l' applicazione è necessario innanzitutto avviare "startmas.sh", con il quale vengono lanciate tutte le istanze degli agenti Seller nonché dell'Organizer e del Manager. Come è possibile notare nella figura sottostante viene, inoltre, aperto un terminale che l'utente può utilizzare per inviare all'agente Organizer il messaggio di avvio applicazione(send_message(activateApplication, NOMEUTENTE).

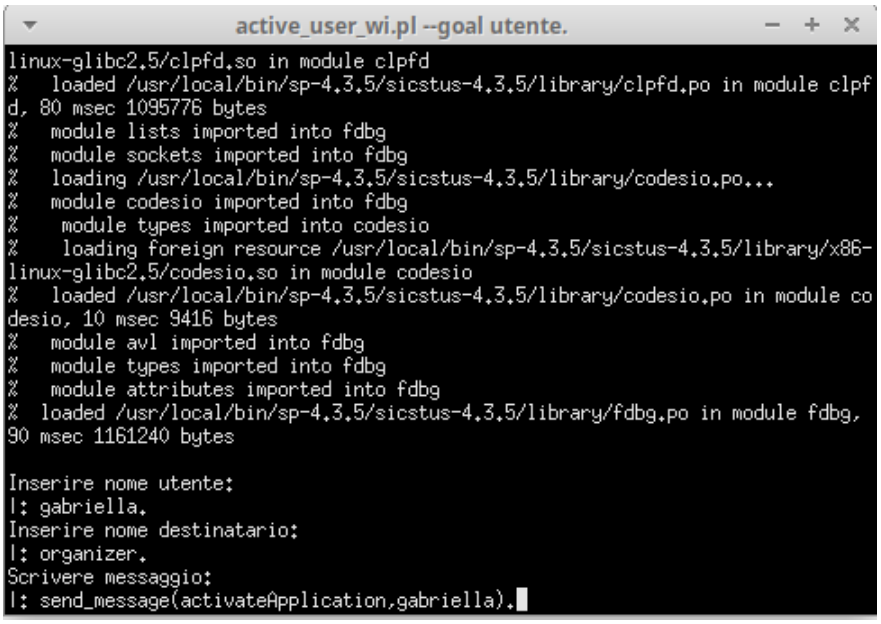


Figura 4 Terminale utilizzabile dall'utente per avviare l'applicazione.