

AAI project 2021

| Nome | Matricola |
|---------------------|-----------|
| Leonardo Serilli | 274426 |
| Gabriele Colapelle | 274560 |
| Alessandro D'Orazio | |

Il problema che questo progetto risolve è quello di implementare, nel linguaggio **DALI**, un **Multy Agent System** in grado di automatizzare i processi di deposito e ritiro di merci da un magazzino, tramite la presenza di due tipi di agente; **un agente di tipo Magazzino**, in grado sia di contenere informazioni riguardanti le posizioni e numero delle merci presenti nel magazzino che di gestire le richieste esterne di deposito e ritiro inoltrandole agli **agenti di tipo Muletto**, incaricati della risoluzione di tali task.

Agente Magazzino

Un magazzino è diviso in **settori** e ogni settore può contenere vari tipi di **prodotti**. Per ogni prodotto l'agente conosce il **numero attuale di unità** e il **settore in cui si trovano**.

Nella knowledge base sono definiti **settori** e **prodotti**:

```
settore(a1).  
settore(a2).  
settore(b1).  
settore(b2).  
  
prodotto(tastiera, 50, a1).  
prodotto(tv, 30, a1).  
prodotto(telefono, 50, a2).  
prodotto(ventilatore, 70, a2).  
prodotto(mouse, 40, b1).  
prodotto(computer, 15, b1).  
prodotto(orologio, 100, b2).  
prodotto(monitor, 20, b2).
```

In seguito sono descritti i metodi per gestire:

(0) Controllo diponibilità prodotti

L'agente esegue un semplice check della quantità attuale del prodotto

```
disponibilita_prodottoE(Prodotto) :>
    prodotto(Prodotto, Qta, Settore),
    write(Prodotto),
    write(Qta),
    write(Settore).
```

(1) Deposito prodotti

1. Gestione richiesta di deposito per un **prodotto già presente**: l'evento

`richiesta_deposito_prodottoE(Agente, Prodotto)` è generato dall'agente `muletto` in quale è intenzionato il settore in cui si trova in prodotto da aggiungere (informazione posseduto nella KB dell'agente `magazzino`). infine al verificarsi di `prodotto_depositatoE(Prodotto, Qta, Settore)` la KB viene modificata con la nuova quantità per il prodotto.

```
richiesta_deposito_prodottoE(Agente, Prodotto) :>
    prodotto(Prodotto, QtaAttuale, Settore),
    format('Richiesta di deposito prodotto da agente ~w', Agente), nl,
    messageA(Agente, send_message(risposta_deposito_prodotto(Prodotto,
Settore), Me)).
```

```
prodotto_depositatoE(Prodotto, Qta, Settore) :>
    prodotto(Prodotto, QtaAttuale, Settore),
    NuovaQta is Qta + QtaAttuale,
    retract(prodotto(Prodotto, QtaAttuale, Settore)),
    assertz(prodotto(Prodotto, NuovaQta, Settore)),
    write('Nuova quantita: '),
    write(NuovaQta).
```

3. Gestione richiesta di deposito per un **prodotto NON presente**: a differenza del metodo precedente, una volta depositato il nuovo prodotto, l'agente creerà una nuova entry della KB per il nuovo prodotto.

```
richiesta_deposito_prodottoE(Agente, Prodotto) :>
    not(prodotto(Prodotto,_,_)),
    format('Richiesta di deposito di un nuovo prodotto da agente ~w', Agente),
    nl,
    messageA(Agente, send_message(risposta_deposito_prodotto(Prodotto, b1),
    Me)).
```

```
prodotto_depositatoE(Prodotto, Qta, Settore) :>
    not(prodotto(Prodotto, _, Settore)),
    assertz(prodotto(Prodotto, Qta, Settore)),
    write('Prodotto aggiunto al magazzino').
```