

Intelligens Elosztott Rendszerek VIMIAC02

Több Ágenses Elosztott Rendszer
Házi Feladat Dokumentáció

2018

Trkala Kristóf DTC7PH
Gyöngyössy Bence TBBFNC

1 TARTALOM

2	Feladatleírás	3
3	A megoldás összefoglalása	4
3.1	Az egyes ágensek és feladataik	4
3.2	Ágensek kapcsolata	4
3.3	Ágensek viselkedése	5
3.4	Összefoglaló ábra	7
4	A fejlesztés összefoglalása	8
4.1	Jason verzió	8
4.2	ASL szintű megvalósítások	8
4.3	JAVA szintű megvalósítások	8
5	A kifejlesztett program ismertetése	9
5.1	Felhasználói interfész	9
5.2	Ágensprogramok rövid összefoglalója BDI-jelleggel	10
6	A működést szemléltető videó elérése	11

2 FELADATLEÍRÁS

Intelligens raktár üzemeltető rendszer

Az intelligens tér egy raktár épület, amiben egységnyi méretű dobozokat lehet tárolni polcokon. A dobozok mozgatásáért, raktárba helyezéséért és kivételéért felelősek az ágensek.

A rendszerben a következő ágensek találhatóak:

- (Önvezető) Targonca, mozgatja a dobozokat
- Check in/out kapu, ellenőrző pont, egy kezelő személy dönti el, hogy átmehet-e rajta egy kamion
- Polc foglaltság figyelő szenzor tárolás kapacitás méréshez
- (Önvezető) Kamion a kapuknál, dobozokat lehet rá rakni vagy levenni róla

Misszió-kritikus esemény ötlet:

- Betelt a raktár

A GUI-n keresztül bevisszük a kezdeti adatokat, ezután megfigyelhető, ahogy az ágensek maguktól intézkednek folyamatosan. A raktár telítettsége leolvasható a GUI-ról (pl.: 99/100).

3 A MEGOLDÁS ÖSSZEFOGLALÁSA

3.1 AZ EGYES ÁGENSEK ÉS FELADATAIK

A feladat megoldásához a következő ágenseket hoztuk létre (zárójelben az implementációban található név szerepel):

- kapu (entryGate): Ez egy félig autonóm módon működő ágens. Feladata az érkező kamionok be-, és kiengedése. Ha egy kamion érkezik a bejárathoz arról jelzést küld, majd egy ember a felhasználói felületen megnyomja a kapu nyitása gombot, aminek hatására a várakozó kamion bemehet a raktárba, hogy elvégezzék rajta a le illetve felpakolást.
- kamion (truck): A kamion feladata az áru szállítása.
- targonca (forklift): A targonca feladata a dobozok lepakolása a kamionról, és azok polcra helyezése, valamint polcról való levétele és a kamionra pakolása.
- polc (shelf): A polc tárolja a dobozokat és számon tartja a raktár telítettségének állapotát. Ha nincs több hely a raktárban, akkor jelez, hogy baj van, betelt.

3.2 ÁGENSEK KAPCSOLATA

Az egyes ágensek rendre a következő ágensekkel állnak kapcsolatban (fekete ponttal az egyes ágensek felsorolása, üres karikával azok az ágensek, amelyekkel kapcsolatban állnak, azok mellett pedig a kapcsolatként szolgáló eseményeik neve szerepel):

- kapu:
 - kamion: arrived
 - targonca: truck
- kamion:
 - kapu: arrived
- targonca:
 - kamion: clear
 - polc: deposit, withdraw
- polc
 - targonca: forklift

3.3 ÁGENSEK VISELKEDÉSE

A következőkben bemutatásra kerülnek az egyes ágensek fő funkciói, interakciói.

Kamion:

- `arrived`: ez az esemény az environment-ben kerül triggerelésre, és hatására jelzi a kapunak, hogy megérkezett és be szeretne menni, valamint paraméterként átadja a rakományának adatait (*deposit*, *withdraw*).
- `needempty`: nincs szabad hely a raktárban, majd később visszajön üresen, hogy el tudjon szállítani belőle
- `finished`: kiírja, hogy végzett a pakolással, és elhagyja a raktárat, majd elindítja a reset goalját
- `reset`: újra generálja a rakományt, és az elvitendő áru mennyiségét

Kapu:

- `arrived`: a kamion ezen keresztül értesíti, hogy megérkezett
- `opengate`: a felhasználó által megnyomott kapunyitó gomb triggereli, és elindítja az `opengategoal`-t
- `opengategoal`: a kapu elkezd szabad targoncát keresni, ha nem talál csak küld egy üzenetet, hogy sajnos nincs szabad targonca, ha viszont talál akkor egyrészt jelez a kamionnak, hogy jöjjön be, másrészt a targoncának, hogy pakolja le a kamiont, aminek a rakodási adatait is átadja paraméterként

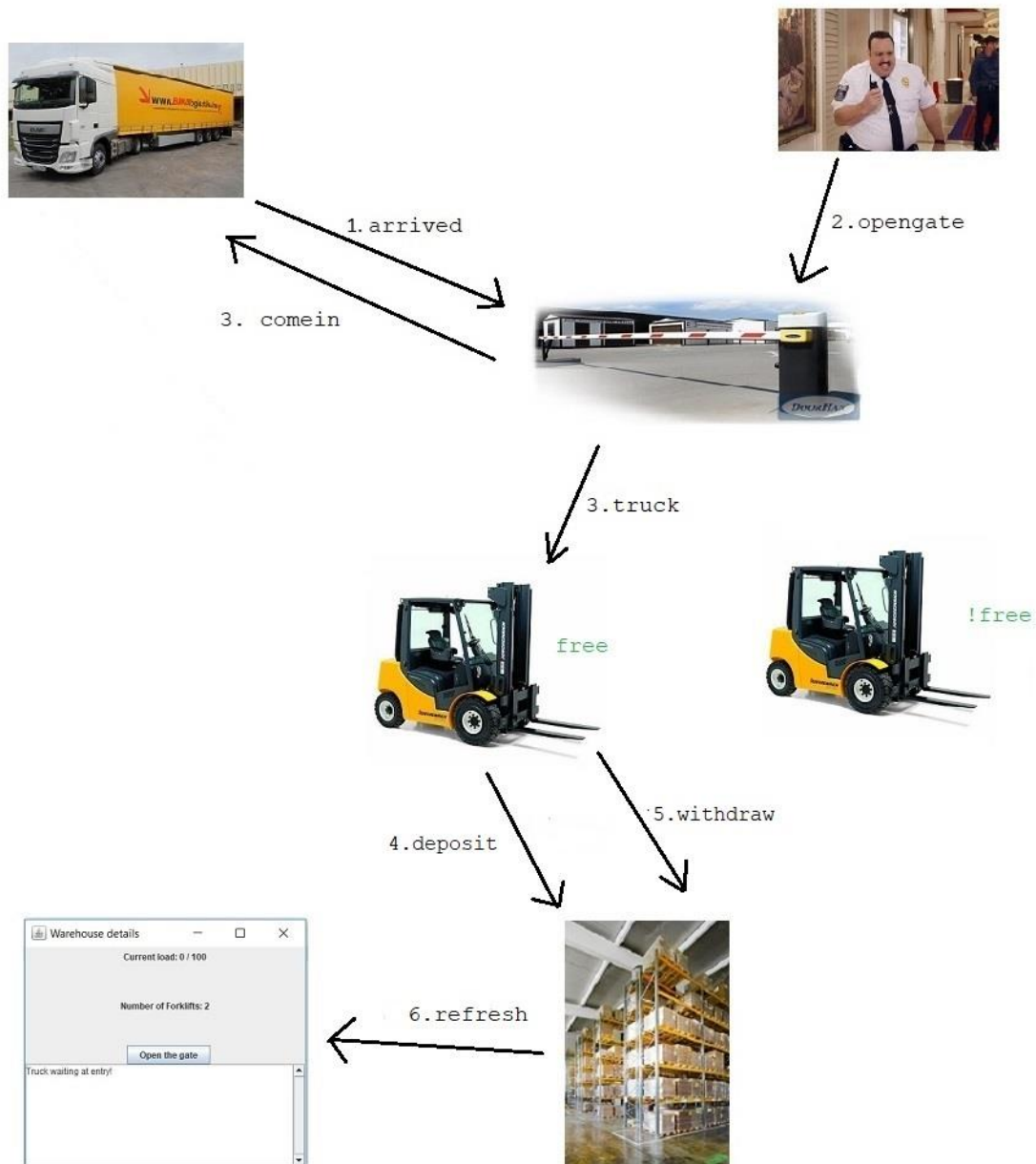
Targonca:

- `truck`: ha kamion érkezett, és szabad az adott targonca akkor hozzárendeli a kamionhoz, a targonca pedig jelzi, hogy most már nem lesz szabad
- `putSuccess`: a targonca miután elérte az `unload` goalját, abban meghívja a `polc deposit` eseményét, ami ha sikeresen végrehajtódik, akkor meghívja a `deposit` párját, vagyis a `withdraw`-t
- `putFailure`: jelzi, hogy nincs elég hely a raktárban, így a kamion később visszajön üresen, hogy fel lehessen rá pakolni
- `done`: jelez a kamionnak, hogy végzett a pakolással, és ki is írja azt
- `unload`: a targonca célja, hogy le/fel pakolja a kamiont, mikor ezt megkezdte azt kiírja a kimenetre, és szimulálva a lepakolás idejét, várakozik egy kicsit

Polc:

- `forklift`: ha egy új targonca kerül a raktárba, ezen keresztül értesíti róla
- `deposit`: a targonca triggereli, mikor rá akar pakolni, amennyiben van hely a doboz felkerül és a tároló kapacitása eszerint változik, amit ki is ír, ha pedig nincs hely akkor kiírja, hogy baj van, tele a raktár
- `withdraw`: hasonló a `deposit`-hoz csak épp az ellenkező irányba, a targonca le akar venni dobozokat, hogy azokat a kamionra pakolja, ami, ha van elég doboz, akkor teljesül és a kapacitást eszerint növeli, ha pedig nincs, akkor jelzi hogy nem sikerült, kiürült a raktár

3.4 ÖSSZEFOGLALÓ ÁBRA



4 A FEJLESZTÉS ÖSSZEFOGLALÁSA

4.1 JASON VERZIÓ

A feladat megoldása során a jason-2.2-a verzióját, valamint az Eclipse ehhez tartozó jason plugin-ját használtuk.

4.2 ASL SZINTŰ MEGVALÓSÍTÁSOK

ASL szinten került megvalósításra minden fő kommunikáció az ágensek között, leszámítva az emberi beavatkozást igénylő kapunyitást.

4.3 JAVA SZINTŰ MEGVALÓSÍTÁSOK

A program összesen két JAVA fájlt tartalmaz:

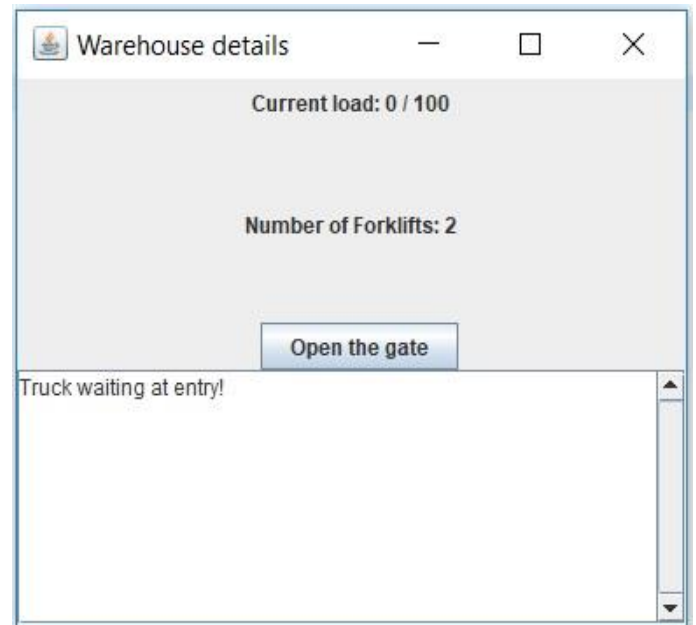
GUI.java : Ebben a fájlban került megvalósításra a grafikus interfész. Az osztály konstruktorában átvesz egy StoreHouseEnv példányt, hogy hozzáférjen a változóikhoz, valamint a kapu nyitásakor meghívja az ahhoz tartozó függvényt. A GUI-ban található elemek a következő pontban kerülnek bemutatásra.

StoreHouseEnv.java : Ez a fájl valósítja meg a feladathoz tartozó környezetet. Ebben az osztályban tároljuk a raktárat jellemző tulajdonságokat, a grafikus interfészen végbement interakciók eredményét, valamint itt szerepelnek az egyes ágensekhez tartozó literálok, például ha kamion érkezik az `arrived` észlelést kapja meg a kapu.

5 A KIFEJLESZTETT PROGRAM ISMERTETÉSE

5.1 FELHASZNÁLÓI INTERFÉSZ

A program felhasználói interfésze az oldalt található képen látható. Itt követhetjük nyomon, hogy áll a raktár telítettsége és hogy hány targonca dolgozik a raktárban. Továbbá amikor kamion érkezik az Open the gate gombbal tudjuk kinyitni a kaput. Miután ez megtörtént, amíg a kamion a raktárban tartózkodik, a gomb inaktívvá válik, így addig nem tudunk másik kamiont is beengedni. Mindezek alatt található még egy görgethető textview, ahol a kamionok érkezése/beengedése kerül kiírásra. A többi



esemény és ágens kommunikáció csak a MAS konzolon látszik, ezt azért raktuk ki külön ide is, hogy ne kerülje el a „kapunyitó” figyelmét, hogy kamion érkezett.

5.2 ÁGENSPROGRAMOK RÖVID ÖSSZEFOGLALÓJA BDI-JELLEGGEL

Ágens	Beliefs	Desires	Intentions
<i>Kapu</i>	X	<ul style="list-style-type: none"> • értesíteni egy targoncát hogy jöjjön lepakolni a kamiont • kinyitni a kaput a felhasználó jelzésére 	<ul style="list-style-type: none"> • beengedni a kamiont • értesíteni, hogy le tudják pakolni a kamiont vagy nem
<i>Kamion</i>	<ul style="list-style-type: none"> • aktuális szállítmány mennyisége (random szám, de maximum 25) • megrendelés mennyisége (random szám, maximum 25) 	<ul style="list-style-type: none"> • bejutni a raktárba • elvinni a raktárból valamennyi árut 	<ul style="list-style-type: none"> • elküldi a rakomány adatait a kapunak • értesít arról, ha bejutott
<i>Targonca</i>	<ul style="list-style-type: none"> • szabad e éppen, vagy foglalt 	<ul style="list-style-type: none"> • a targonca célja, hogy amint a kapu szólt neki, hogy jött egy kamion és van is hely a raktárban lepakolni, akkor ő levegye a szállítmányt és átrakja a polcra, majd visszapakoljon mást a kamionra 	<ul style="list-style-type: none"> • értesíti a kaput, hogy már nem szabad, és elkezd pakolni • felpakol a polcra • levesz a polcról
<i>Polc</i>	<ul style="list-style-type: none"> • tároló kapacitásának maximális értéke = 100 • kapacitás kezdő értéke 	<ul style="list-style-type: none"> • megakadályozni, hogy a raktárba többet akarjanak pakolni, mint amennyi fér, vagy többet kivenni, mint amennyi van 	<ul style="list-style-type: none"> • ha a raktár tele, szól róla a targoncáknak • ha a raktár üres, szintén szól róla a targoncáknak

6 A MŰKÖDÉST SZEMLÉLTETŐ VIDEÓ ELÉRÉSE

<https://youtu.be/pJCteA8PFRw>

A videó elején látható a misszió-kritikus esemény kialakulása: betelne a raktár, ha lepakolná a targonca a kamiont, ezért inkább csak elküldi, majd kicsit később üresen érkezik vissza a kamion, hogy csökkentse a raktár telítettségét.