Intelligens elosztott rendszerek

BMEVIMIAC02

2019

Az “Open Hybrid Lab Factory” gyár

komponensének modellezése

<https://www.youtube.com/watch?v=T2uBxjeF5jE>

<https://github.com/june9666/IER-HF>

Benkő Csaba (CMEB33)

Szalontai Jenő (JWCQK2)

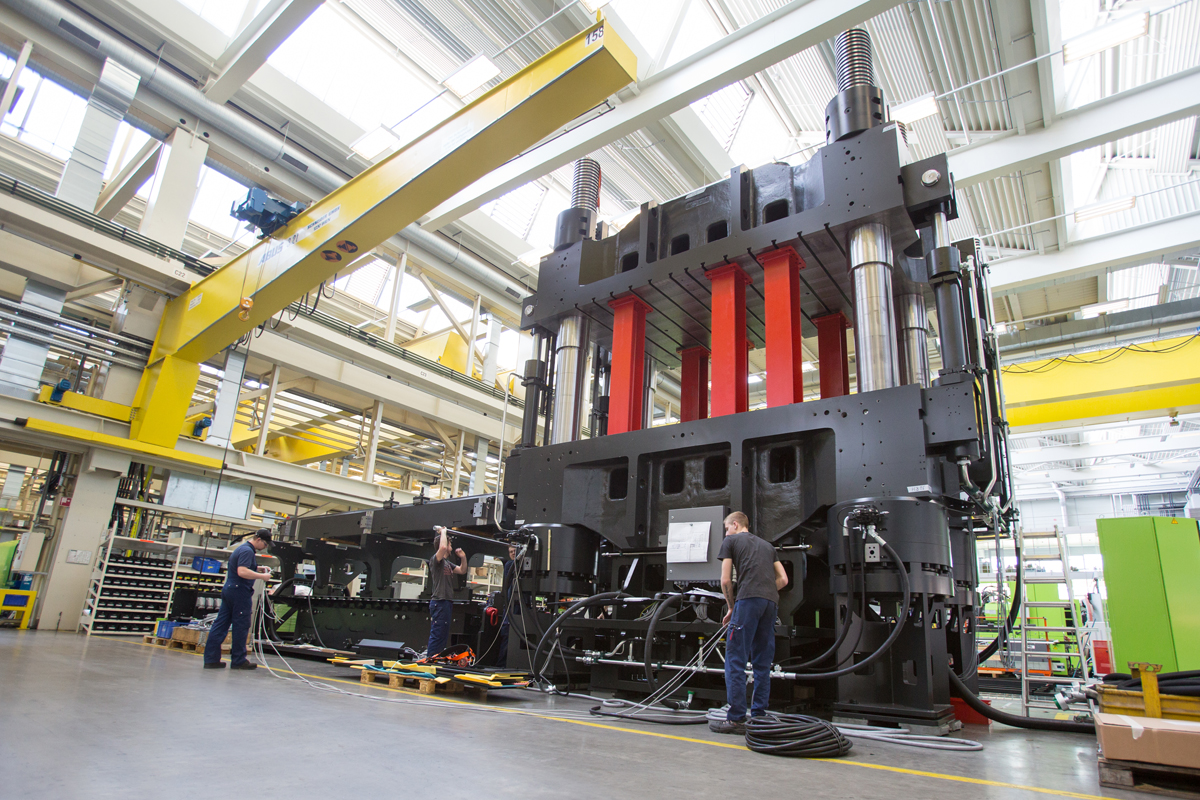
Tartalomjegyzék:

1. A feladat leírása
2. A megoldás összefoglalása
3. A fejlesztés összefoglalása
4. A kifejlesztett program ismertetése

A feladat leírása

A feladatban az intelligens tér szerepét egy ipar 4.0 gyár tölti be.

A gyár rendelkezik különböző érzékelőkkel, több “szalaggal”, melyek autonóm működnek. A problémát a kritikus jelenségek előrejelzése, kezelése szempontból közelítjük meg. Házi feladatként az [*Open Hybrid Lab Factory*](https://open-hybrid-labfactory.de) gyárat és egy benne lévő komponenst szeretnénk modellezni.



*A képen az ENGEL v-duo 3600, 3600 tonna erejű prés látható.*

Feladatunk, hogy a prés érzékelői és gyár rendszere segítségével jelezzük és kezeljük az esetleges meghibásodást.

Az eszközön különböző érzékelők találhatók, melyek az aktuális státuszt mutatják. Karbantartás nélkül az eszköz elromlik, illetve tűzesetek keletkezhetnek. A gyárban szimulálható tűzeset.

A megoldás összefoglalása

* *Ágensek:*

*https://lh5.googleusercontent.com/TXW5Ni4RRThYGHYud1DAaG8dFIYh8imp8Za-sUtb3qQ8WaTnYUI88T4qb7upabBVR4QbwxJZ2kWTkacSFSirGOtF1vBlIV3y_MfaQau9H-Aa2R23zfg-rUQq9rxhjA***Gépellenőr**: A gép szenzorai közt járva leolvasa azok értékét. Ha egy értéket rendellenesnek ítél, értesíti a karbantartót, majd félreáll a javítás elvégzéséig.

https://lh3.googleusercontent.com/rGBbohSWH60j1HbpKhogDDD9r8GASM8O57_On6mPnSgGW50dMJlILU3GZkKy6WpYu6iU-o8Dx4eGV8jSgqOj96kZe6RfifAPtHJVDgBTVO_jFYmJ6SRodI-Cf6gYag

**Karbantartó**: Ha értesítést kap a gép elromlásáról, a jelzett helyre megy, megjavítja a hibát, majd visszatér a szervízállomásra. (Innét tud alkatrészeket és szerszámokat vinni a következő javításhoz.)

https://lh3.googleusercontent.com/ZyHuMdUF7AUHsuvozhXxkD91pKqbz-jM23vzkJlW_f3CKg-UqMAyc_RfZzc3ub69tmzy4QSH-0ncqaF3QRojpVcbZhiSoilpASegqqyyTfnES2qaJXXTgA1g_i37Qw**Tűzoltó**: Tűz esetén kivonul a jelzett helyszínre, eloltja a tüzet, majd visszatér oltóanyagért a tűzoltóállomásra.

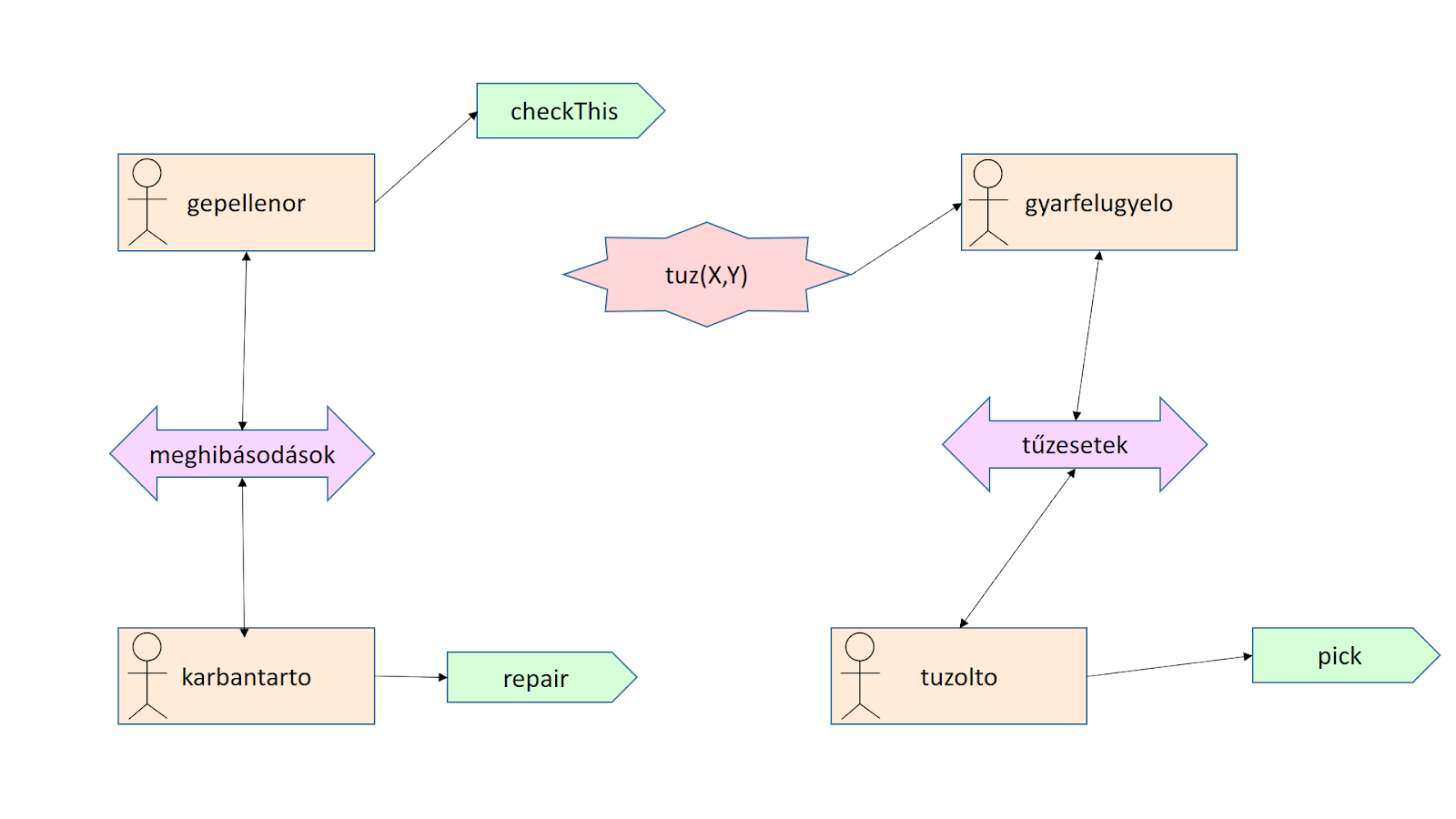
**Gyárfelügyelő**: A gyár területén figyeli a tüzek esetleges kialakulását. Tűz esetén tájékoztatást küld annak helyzetéről.

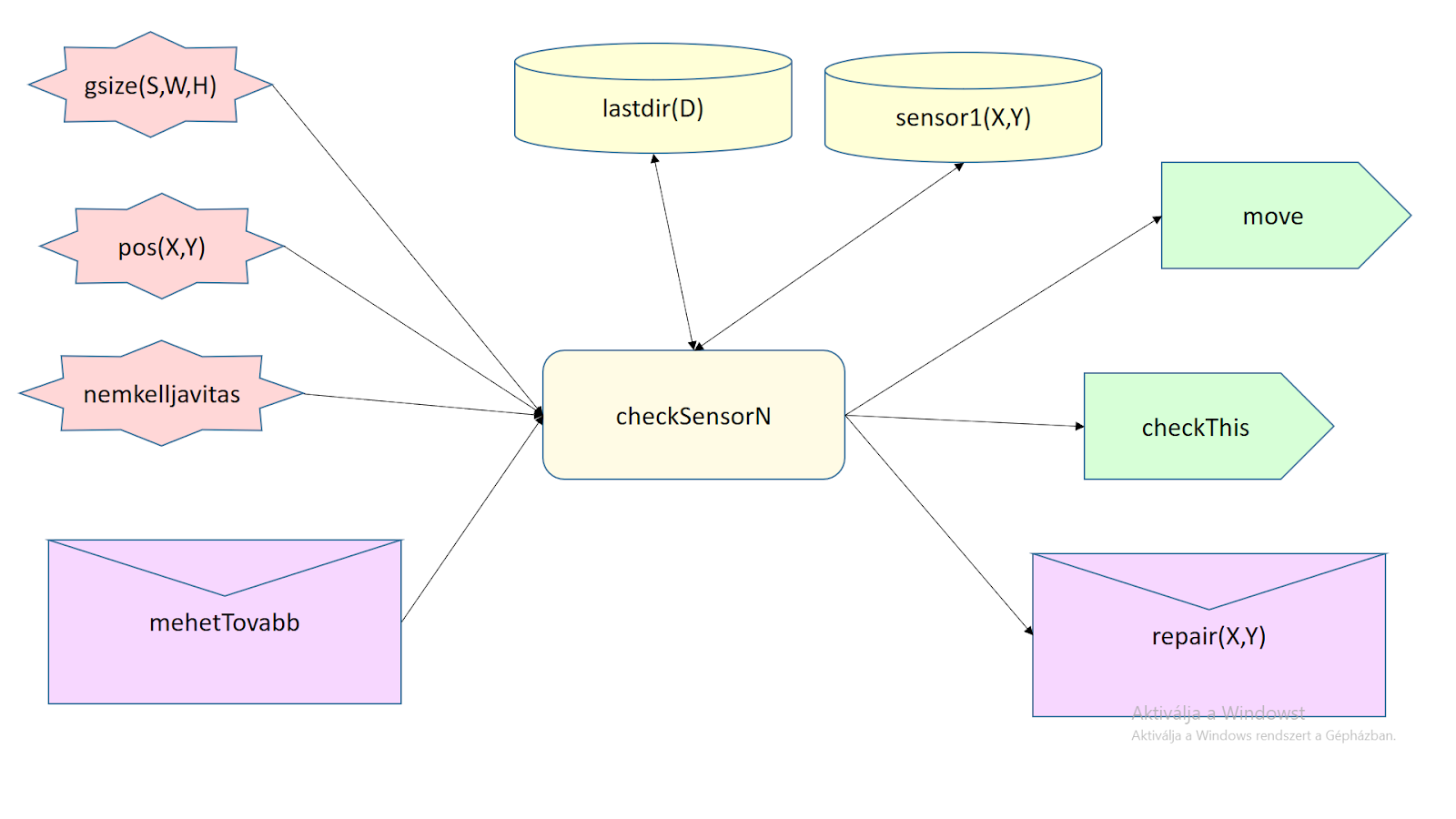
* *Ágensek kapcsolata, egymástól való függősége, ágensek viselkedése:*

Legszorosabban a gépellenőr és a karbantartó működik együtt. Értesítik egymást a gépek állapotának változásáról (javításra szorul/kijavítva).

A gyárfelügyelő a tűzoltóval működik együtt, értesíti őt a tűzesetekről.

* *A többágens rendszer összefoglaló ábrája:*

*A Gépellenőr össszefoglaló ábrája*

**

## A fejlesztés összefoglalása

### Milyen Jason verziót használtak?

Jason 2.3

### Mi készült ASL szinten?

Az ágensek feladatai, céljai, egymással való kommunikációja. A feladatok lépései is itt kerültek leírásra kivéve az olyan elemi műveleteket, melyeket a környezetnek kell biztosítania (pl.: szomszéd mezőre lépés, tűz eloltása).

A kommunikáció fajtái:

Gépellenőr –meghibásodás-> Karbantartó

Karbantartó –javítás vége-> Gépellenőr

Java kód –addPercept(tűz)-> Gyárfelügyelő

Gyárfelügyelő –tűzeset-> Tűzoltó

### Mi készült Java szinten?

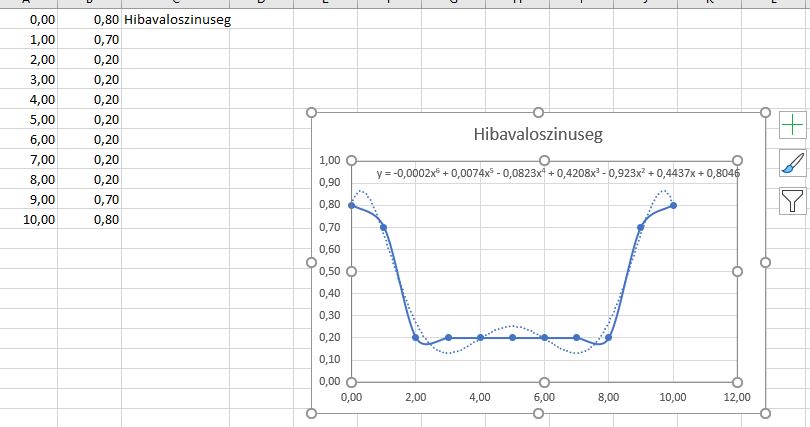
#### PhysicalModell.src / Intelligent Factory.java

A világ felépítése egy WorldModel alapján, majd a kezdeti hiedelmek létrehozása.

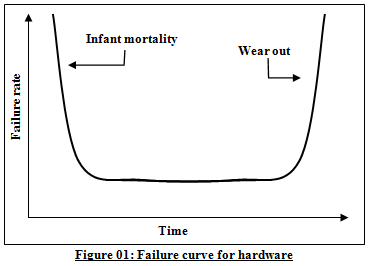
Itt található a tűzeset hiedelmek kezelése is.

#### PhysicalModell.src / Press.java

A fizikai prés dinamikus modellezése történik itt.



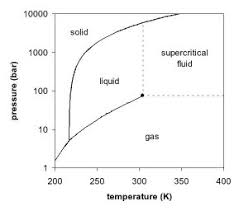
Ábra 2.) Hibavalószínűség függvény megalkotása

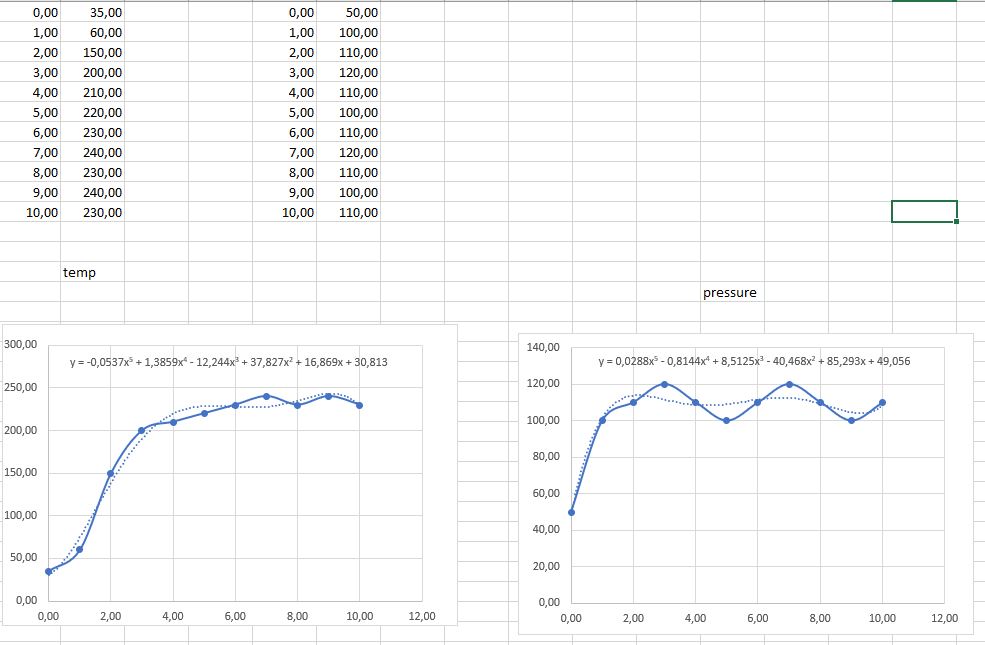


Ábra 1.) Hardwarehez tartozó hibafüggvény

Az (Ábra 1.) mutatja az alapvető hardwares hibagörbét, én létrehoztam a saját közelítő függvényem, amit a hibák generálására használok (Ábra 2.)

Ehhez hasonló módon, a hőmérsékletet(Ábra 3.) és a nyomást is modelleztem.



Ábra 3.) Hőmérséklet-Nyomás modellje

Ábra 4.) Saját modell Hőmérséklet és Nyomáshoz

#### JIA / getPress

Kiolvassa a prés szenzor értékeit, ha túl alacsony az adott szenzor által modellezett komponens kondíciója, akkor false-al tér vissza. Megj: a JIA packaget sikerült csak az ágensen belül hívni, ezért került ide.

### JIA / repairPress

Megjavítja az adott prés kondícióját.

#### PhysicalModell.src / WorldModel.java

A világ modelljének létrehozása, jelenleg a „World 1” tartalmazza a mi gyár modellünket, de tetszőleges gyár modell felépíthető vele.

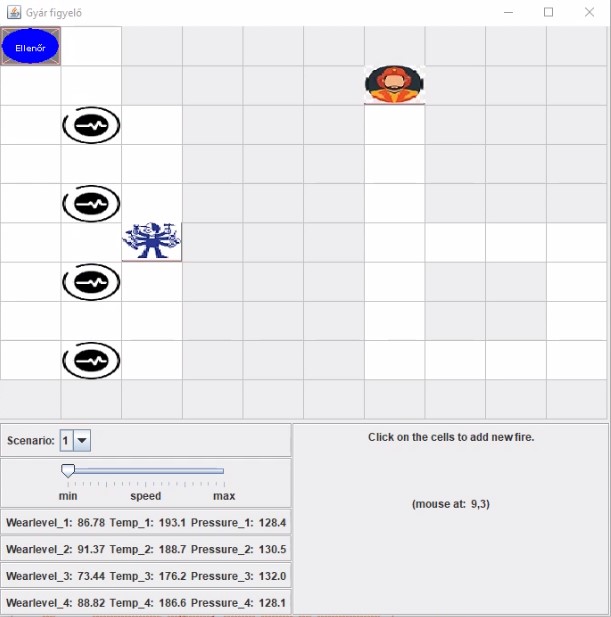
Ugyan akkor a világban navigáló függvény is itt van.

#### PhysicalModell.src / WorldView.java

A GUI- amit a felhasználó lát (Ábra 5.).

A felső 2/3-ban található a pálya, amin mozognak az ágensek, alul pedig a kezelő, monitorozó komponensek.

Itt található az egér-esemény kezelő is, aminek hatására tűz keletkezik a kattintás helyén.

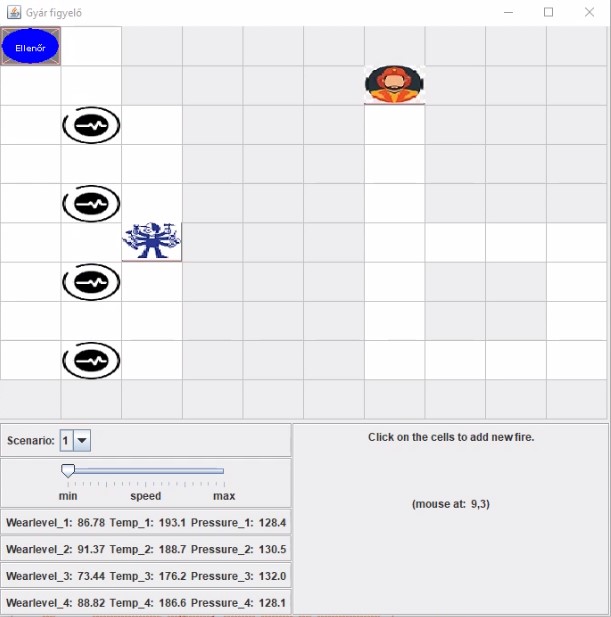


Ábra 5.) GUI

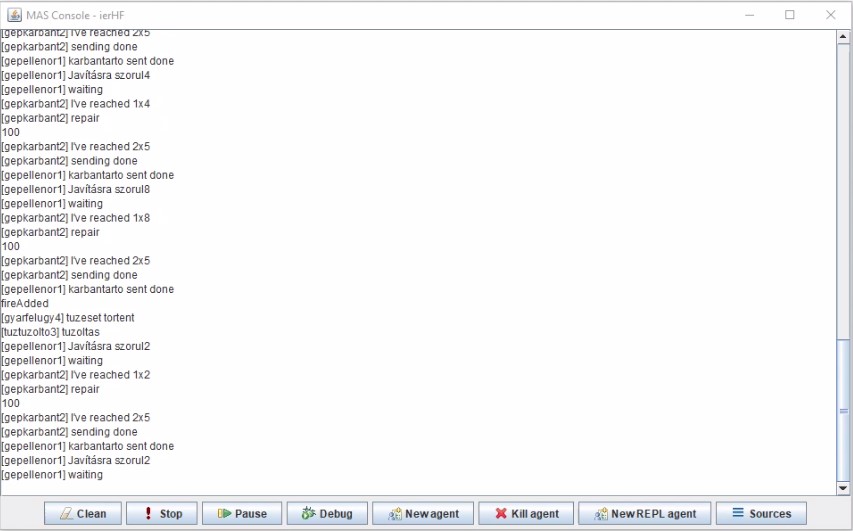
## A kifejlesztett program ismertetése

A program működése videó által bemutatva:

[https://www.youtube.com/watch?v=T2uBxjeF5jE](https://www.youtube.com/watch?v=T2uBxjeF5jE&fbclid=IwAR1IAhxkHESJmd6a3Hc8sACw39BHvElWdoAN8dIMjV7OGJUBLqcxqK4vf98)



A fenti képen látható az elkészült program felhasználói felülete. A középső csúszka segítségével állítható be az ágensek sebessége, illetve alattuk látható gépek különböző értékei. (Elhasználódás, hőmérséklet, nyomás). Mellettük pedig az egér pozíciója jelenik meg.



A fenti konzol ablakban láthatóak a különböző ágensek folyamatai időrendi sorrendben.

### Egyes ágensprogramok rövid összefoglalása (BDI jelleggel):

### Gépellenőr

Az ágens a működés kezdetén megkapja hiedelemnek a saját pozícióját, azt, hogy még nem mozgott, illetve, hogy jelenleg egyik eszköz sem szorul javításra, kezdeti feladatnak pedig az első szenzor ellenőrzését. A közös tudásbázisból kikeresi a szenzor helyét, odamegy, leellenőrzi, majd továbbmegy a következő szenzorra, majd a következőre, s. í. t. Ha valamelyik szenzornál hibát észlel, visszamegy a kiinduló pozíciójába (hogy ne zavarja a szerelők munkáját), üzenetet küld a szerelőknek, majd leáll. Hogyha üzenetet kap arról, hogy tovább folytathatja munkáját, ismét felveszi céljai közé az első szenzor ellenőrzését, ahonnét megint továbbmegy majd a következőkre. Megj.: Amíg ez az üzenet nem érkezik meg, addig az ágens még véletlenül sem tud mozogni, ugyanis a kezdőpozícióra érkezéskor eltávolította a saját tudásbázisából azt az ismeretet, hogy nincs megjavítandó szenzor. E nélkül pedig nem lesz olyan terve a mozgásra vonatkozóan, amit végre tudna hajtani.

### Karbantartó

Induláskor a karbantartó tudásbázisában az az információ van, hogy éppen nem ,,hazafelé” (a szervízállomás felé) tart, valamint hogy még nem mozgott. Kezdeti célja: várakozni. Ezt mindaddig folytatja is, amíg nem kap üzenetet arról, hogy meg kéne javítania valamit. Hogyha ez az üzenet érkezik, elmegy az üzenetben megjelölt helyre, megjavítja a prést, felveszi a tudásbázisába, hogy éppen hazafele tart, hazamegy, majd kiveszi a tudásbázisából, hogy hazafele tart. A hazamenetel fejben tartására azért van szükség, hogy egy esetleges új értesítés meghibásodásról ne térítse el egy másik irányba, ugyanis szerszámai és alkatrészei nélkül, melyek a szerelőállomáson találhatók, nem fogja tudni megoldani azt. Hazaérve üzenetet küld az ellenőrnek, hogy folytathatja ellenőrző körútját.

### Gyárfelügyelő

### Az ő feladata szintén várakozni. Ha a rendszer azt az érzetet közvetíti számára, hogy valahol tűz van, értesíti erről a tűzoltót.

### Tűzoltó

Kezdeti tudása, hogy a gyárban nincs tűz, kezdeti célja pedig várakozni. Hogyha üzenetet kap a gyárellenőrtők, hogy tűz van, kiveszi a tudásbázisából azt a hiedelmet, hogy nincs tűz, elmegy az üzenetben jelzett helyre, eloltja a tüzet, visszamegy a helyére, és ismét beteszi a tudásbázisába a hiedelmet, miszerint nincs tűz.

## Program egészének összefoglalása (többágenses rendszer jelleggel)

A rendszerben együttműködő ágensek észlelik és elhárítják a gyár működést meggátló potenciális hibák két legfontosabbikát, a gép hibáit és a tűzeseteket. Előbbiekkkel a Gépellenőr és a Karbantaró, míg utóbbiakkal a Gyárfelügyelő és a Tűzoltó foglalkozik. A Gépellenőr és a Karbantartó üzenetek segítségével kommunikál egymással, működésük pedig biztosítja, hogy nem tartózkodnak egyszerre a gépek között, így nem nehezítik egymás munkáját. A Gyárfelügyelő szerepe limitált, fizikai megjelenése nincs is a pályán, mintegy ,,felülről” szemléli a gyárépületbéli tűzeseteket, közvetítőként működve. A Tűzoltó, a Karbantartóhoz hasonlóan, amint értesül arról, hogy szükség van a szolgálataira, felhagy korábbi tevékénységével, és eloltja a Gyárfelügyelő által közvetített tüzet.