

Intelligens Informatikai Épület

Intelligens elosztott rendszerek
(BMEVIMIAC02)

Németh Roland (B43SYE)
Takács Gergő (E7FCHF)

Budapest, 2017.05.19.

Tartalom

- Feladat leírása
- Ágensek
 - Security
 - Paramedic
 - Doorman
- Ágensek kommunikációja, kapcsolata
- Folyamat leírása
- Felhasználói felület
- Fejlesztői döntések
- Kifejlesztett program ismertetése, működése

Feladat leírása

A feladat célja egy intelligens informatika épület működésének szimulációja, melyben több, egymással összedolgozó ágens működik. Mindegyik ágensnek van feladata, de céljaikat csak egymás segítségével tudják elérni, tehát szükségük van egymás munkájára.

A szimulációs környezet az épület belseje, melynek alapterülete 10x10-es. A megvalósított feladatot tehát egy épület belső modellje, illetve a benne egymással kommunikáló, összedolgozó ágensek alkotják. A feladat alap koncepcióját úgy képzeltük el, hogy van egy hatalmas terület, amelyen emberek közlekedhetnek, ugyanakkor mivel a biztonság az egyik, ha nem a legfontosabb dolog, vannak járőreink, akik az épület belsejében egy meghatározott pálya mentén mozognak és figyelnek. Amennyiben azt látják, hogy egy orvosi segítségre szoruló ember van az épületen belül, értesítik az épület paramedikusát és folytatják útjukat. Természetesen amennyiben több sérült is található, ezeket a járőrök mind jelentik a paramedikusnak. A paramedikus, amennyiben szóltak neki, elindul a sérült felé, hiszen a járőrök feljegyezték a sérült helyét is, majd miután ellátta őt és nincs több sérült aki rá vár, visszatért a rendelőjébe. Amennyiben több sérülthöz is riasztották, a riasztás szerinti sorrendben látja őket el.

Ágensek

A szimulációban részvevő ágensek között megkülönböztetünk paramedikust, portást, biztonsági őröt és sérültet.

A biztonsági ágens egy emberi ágens, melynek feladata az épületen belüli sérültek felderítése. Alap esetben az épületen belül járőrözik. Ha látja, hogy valaki lesérült, azonnal értesíti a paramedikus ágenszt rádió eszközön keresztül, majd folytatja a járőrözést. A biztonsági ágensnek feladatainak elvégzésére mindössze egy rádió eszközre van tehát szüksége, amellett, hogy képesnek kell lennie érzékelni a körülötte folyó eseményeket, nagyrészt vizuális információ alapján (látás képessége).

A paramedikus ágens úgyszintén egy emberi ágens, aki ideje nagy részében utasításra vár. Ha szólnak neki rádió eszközön, hogy orvosi ellátásra szorul valaki az épületben, akkor a helyszínre siet és ellátja a sérültet. Amint sikerült ellátnia a sérültet, visszatér az alap helyzetébe és készenléti állapotban várja a következő utasítást. Feladatvégzéséhez nincs szüksége többre, mint egy rádió eszközre, melyen keresztül kommunikálhatnak vele, illetve orvosi felszerelésre, amivel elláthatja a sérülteket.

A porta ágensnek jelenleg bejelentkezésen kívül nincs funkciója.

Ezek az ágensek kívül léteznek sérültek is. Az épületben előre definiált helyeken helyezkednek el. Őket érzékeli a biztonsági ágens, jelenti be a helyét a paramedikusnak és látja el az orvosi ágens a bejelentési sorrendben.

Security

A security, vagyis biztonsági ágensek feladata, tehát a folyamatos járőrözés és a sérültek észlelése, illetve a sérültek jelenlétéről információ szolgáltatása a paramedikus ágensnek. A sérülteket járőrözés közben mindig figyeli egy 5x5-ös területen maga körül. Amint észlel egy sérültet, rádieszközzön szól a paramedikusnak, aki ettől kezdve figyeli a biztonsági ágensek által bejelentett sérülteket.

Paramedic

A paramedic, vagyis paramedikus ágens feladata, hogy a sérülteket ellássa. A járőr üzeneteire felfigyel és az első üzenet hatására aktívan kezdi figyelni a bejelentett sérülteket, majd bejelentési sorrendben hozzájuk siet és ellátja a sérülteket. Ha nincs több ápolásra szoruló sérült az épületben, akkor visszatér kezdeti pozíciójába. Ha elindult vissza a kezdeti pozíciójába és új sérültet jelentenek, akkor az új célja a sérült pozíciója lesz.

Doorman

A doorman, magyar nevén portás feladatai között szerepel a környezetének felügyelete, gond esetén a security értesítése. Sajnos idő hiányában ezen ágens implementációjának megvalósítása elmaradt. Az egyedüli funkciója az, hogy bejelentkezik a rendszerbe szimuláció indításakor.

Ágensek kommunikációja, kapcsolata

A szimuláció során a kommunikációban résztvevő ágensek a paramedikus, illetve a biztonsági ágens. A biztonsági ágens figyeli a környezetét és értesíti a paramedikus ágens, ha sérültet talál az épületben. Ebből tudja a paramedikus, hogy dolga lehet, így el kezdi figyelni a biztonsági ágens jelentéseit és ezekre reagál.

A biztonsági ágens jelez a paramedikusnak, aki erre el kezdi figyelni annak a jelentéseit. A biztonsági ágens a rendszerben jegyzi az észrevett sérülteket. Ezeket a jegyzéseket figyeli az első jelzés hatására a paramedikus. A rendszerben feljegyzett sérülteket mind ellátja, addig, míg a jelentések listája nem lesz üres.

Folyamat leírása

Egy futása a folyamatnak az alábbi szcenárió szerint zajlik:

A szimulációt elindítva láthatunk 3 sérültet, ezeket egy szürke négyzeten lévő fehér I (Injured) betű jelez, továbbá egy rögzített pozíciójú ágens nevezetesen a portást, akit egy sárga kör és rajta egy D betű (Doorman) szimbolizál. Két biztonsági ágens, amelyek folyamatosan egy meghatározott körpályán mozognak és a körülöttük lévő területet vizsgálják lépésről lépésre, őket egy kék körön lévő S betű jelez (Security). Amennyiben ezek a biztonsági ágensek a fentiekben leírt terület méreten sérültet látnak, a sérült helyét eltárolják, felveszik őket a sérültek listájára majd értesítik a paramedikust és haladnak tovább. A paramedikus, akit egy piros kör szimbolizál egy P betűvel (Paramedics) van ellátva. Ameddig az ellátásra szoruló emberek listája üres, ő egy helyben van és bevetésre kész. Amint a biztonsági ágens felvesz valakit a sérültek listájára, ezzel egy időben küld is egy értesítést a paramedikusnak, aki ezután a sérültet ellátja. Fontos megjegyezni, hogy a paramedikus az imént említett sérültek listájából tudja, hogy kihez kell menni először és pontosan hova. Amint szükség van rá, a paramedikus odamegy a sérülthöz és ellátja, ez a modellen úgy látszik, hogy egyszerűen eltűnik a sérült, hiszen már ellátták és ezzel egy időben a paramedikus törli is az adott sérültet a listáról. A sérültek listája tehát azon az elven működik, hogy a biztonsági ágens folyamatosan figyel, keres és ha talál, felvesz elemet a listára, míg a paramedikus törli ezt az elemet miután ellátta orvosi kötelezettségeit. A paramedikus addig van úton, és látja el a sérülteket amíg vannak. Amint ellátta az összeset és a lista kiürült, visszamegy a kiindulási helyére. Természetesen miután visszaért még találhat a biztonsági ágens új sérültet, ekkor az egész szcenárió kezdődik előlről és paramedikus útnak indul.

Felhasználó felület

A felhasználó felület az épület belsejében jelenlevő állapotot reprezentálja. Az egyes ágensek típusok szerint vannak színezve, illetve megnevezve a könnyebb felismerhetőség érdekében. A mozgások és egyéb műveletekről is vizuális információval szolgál a felhasználó számára. Előre definiált helyzetet mutat és old meg a felhasználó szeme láttára, így a felhasználó az eseményeket csak követni tudja. A felület kiegészítésében meg akartuk valósítani, hogy egy gomb megnyomásával a felhasználó egy véletlenszerű helyen helyezhessen el a rendszerben egy sérültet. Sajnos ez később nem került implementálásra.

Fejlesztési döntések

A fejlesztés során úgy döntöttünk, hogy ASL szinten valósítjuk meg a paramedikus részére szolgáló jelzést, melyben a biztonsági ágens jelez számára, hogy feladata lesz, így kezdje meg az aktív figyelést, ami szintén ASL szinten hívódik meg. Továbbá ASL szinten történik a biztonsági ágensek léptetésének folyamatos meghívása, illetve a paramedikus helyváltoztatására szolgáló függvényé is, aktív helyzetben.

Java szinten tároljuk a bejelentett sérülteket a rendszerben, végezzük az ágensek léptetését, ellenőrizzük a jelentési listát a paramedikus számára és döntünk annak függvényében a paramedikus további mozgatójáról. Java szinten valósul meg a sérültek gyógyítása is. Ezen felül a felhasználói felület is Java szinten lett megírva.

A fejlesztés során igyekeztünk egyenlő arányban elosztani a feladatokat. A kommunikáció megvalósításában Gergő aktívabban vett részt, míg a mozgatóban Roland. A döntéseket találkozókon hoztuk meg. Találkozót igen sok alkalommal tartottunk, az összehangolt munka érdekében. Igyekeztünk minél többet személyesen ötletelni, hiszen így egyrészt sokkal hatékonyabbak voltunk, másrészt könnyebben tudtunk ötleteket megvitatni, új funkciók előnyeit, hátrányait megtárgyalni, vagy éppen elvetni. Volt példa arra is, hogy otthonról próbáltuk a kódolás részét végezni és közben megbeszélni a saját döntéseinket (pl.: skype segítségével), ugyanakkor beláttuk, hogy ez nem járható út abban az esetben, ha egy olyan programnyelvben kell maradandót alkotni, amelyben egyikünk sem annyira jártas, tapasztalt. Ezen kívül természetesen az egész projekt fejlesztési procedúráját végigkövethetjük a commitok segítségével, hiszen a github grafikus megjelenítésére is alkalmas szoftvert használtuk, név szerint a Source Tree-t. Nem egyszer fordult elő az sem, hogy kódokat ilyen meetingeken írtunk meg és implementáltunk új funkciókat, működéseket.

A fejlesztést 2.1-es Jason verzióval végeztük.

Kifejlesztett program ismertetése, működése

A paramedikus csak egyszer reagál az injuredFound üzenetre, amelyet a biztonsági ágens valamelyikétől kap, közli, hogy kapott egy ilyen üzenetet és azt is, hogy melyik ágenstől, majd elindítja a startot. Onnantól start állapotban fut és folyamatosan meghívja a startWorking függvényt, amelynek saját maga nevét adja meg paraméterül.

A startWorking függvény a Java kódban a work terminológiával egyezik meg, melynek hatására a notifyParamedics függvény hívódik meg, melyben leellenőrizzük, hogy tényleg van-e még el nem látott sérült az épület területén, majd ha úgy látjuk hogy van, akkor kiszolgáljuk az első sérültet, felé mozgatójuk a paramedikust. Minden lépés után a sérült ellenőrizzük, hogy elértük-e a célunkat, ha igen, meghívjuk a healInj függvényt, mely a paramedikus jelenlegi pozícióján gyógyítja meg a sérülteket, ha vannak.

A biztonsági ágens a start állapotában közli, hogy elkezdett dolgozni, majd a move állapotba lép, ahol mindig meghívja a next függvényt, szintén a saját nevét átadva a függvénynek.,

meghívja a lookAround függvényt, megadva a saját nevét a függvénynek és megüzeni a paramedikusnak, hogy kezdje el vizsgálni a talált sérülteket az injuredFound üzeneten keresztül.

A next függvény az s1 és s2 terminológiának felel meg a java kódban, attól függően, hogy melyik biztonsági ágens hívta meg. Így elkülönítjük a két biztonsági ágens mozgását az adott függvényben, hogy ne kétszer mozgassunk egy cikluson belül. Ezen terminológiák bármelyikének az észlelésekor a nextSlot függvényt hívjuk meg paraméterrel aszerint, hogy melyik biztonsági ágens mozgását akarjuk végezni. A lookAround függvény a look1 és look2 terminológiának felel meg, szintén az alapján, hogy melyik biztonsági ágens hívta meg a kettő közül. Ez is elkülönítési érdeket szolgál. Így nem fogunk felesleges számításokat végezni egy adott cikluson belül. Ebben a függvényben az egyes ágensek egy 5x5-ös területet vizsgálnak meg maguk körül, sérültek után kutatva. Ha sérültet találnak, azt egy listában jegyzik, a pozíciójával együtt. Végül ASL-ben üzenünk a paramedikusnak, majd kezdjük a következő ciklust.

A jobb nyomonkövethetőség érdekében a szálat 1000 ms, vagyis 1 másodpercig altatjuk, majd folytatjuk a szimuláció futtatása minden updatePercets függvény hívásra, amelyet minden cselekmény után meghívunk.

A porta ágens egyszerűen csak beköszön start állapotba, majd hagyja a többi ágens dolgozni.

A program működése az alábbi linken tekinthető meg:

<https://www.youtube.com/watch?v=c3KMCdHliV8>