

Ágens kommunikáció labor mérési útmutató

BME VIK BSc. Intelligens Rendszerek ágazat, Kooperatív rendszerek labor

Kidolgozta: Mészáros Tamás, BME MIT

A labor célja, megszerezhető ismeretek

A labor célja, hogy a hallgatók alapvető gyakorlati ismereteket szerezzenek az intelligens ágensek kommunikációjáról. A gyakorlat során a hallgatók megismerkednek az ágensek közötti interakciós protollokkal, az XML-alapú tartalomnyelvekkel és feldolgozásuk alapvető szintaktikai és szemantikai feladataival Java nyelven Eclipse környezetben.

Szükséges előismeretek

A felkészülést a labor elején ellenőrizzük!

- Az első labor anyaga (Jade használat, Eclipse integráció, ágens programok felépítése)
- a Russel-Norvig MI könyv 22. kommunikáció c. fejezetéből a 22.1 alfejezet
- a „Kooperatív és tanuló rendszerek” c. tárgy ágens kommunikációról szóló része („Beszédaaktusoktól az ACL-ig”), valamint a kapcsolódó jegyzetei
- XML alapismeretek (pl.) „Az XML alapjai” c. segédlet alapján
- A Schema deklarációs nyelv vázlatos ismerete a „” c. segédlet alapján
- a Jade könyv alábbi fejezetei: 4.2.5, 4.3, 5.4 bevezetés (5.4.1-től már nem)

Felhasznált eszközök

- Jade keretrendszer
- Eclipse fejlesztői környezet (EJade kiegészítővel)

Beszámoló készítése és tartalma

A laborfeladatok elvégzéséről írásbeli beszámolót kell készíteni. A labor honlapjáról letölthető egy minta jegyzőkönyv (lab02_agens_kommunikacio_jegyzokonyv), annak formátumát és tartalmi felépítését kell követni.

A beszámolónak tartalmaznia kell a feladatok során feltett kérdésekre adott válaszokat, valamint a feladatok elvégzése során tapasztalt jelenségek leírását, elkészített programrészleteket (a dokumentálandó részeket külön is beszámoltuk: J1, J2. stb.). Egy-egy ügyesen elkészített képernyőképpel sok szöveg pótolható...

A feladatok megoldásával párhuzamosan ajánlott készíteni a beszámolót, mivel azt a labor végén kell leadni.

A labor értékelése

Elégtelen felkészülés esetén a laboron a hallgató nem vehet részt.

A mérésvezető a laborfoglalkozáson mutatott teljesítmény és a leadott beszámoló formai és tartalmi értékelése után érdemjegyet ad.

A jeles osztályzathoz szükséges, hogy a hallgató legalább egy szorgalmi feladatot is megoldjon.

1. feladat: az ágens kommunikáció alapjai

A feladat célja: a laborkörnyezet megismerése, egyszerű ágens kommunikáció követése

Referencia idő: 40 perc

Részfeladatok:

Indítsa el az Eclipse környezetet, válassza ki a Jade labor munkaterületet, töltsse be az ir1lab nevű projektet és keresse meg a milab.lab02 csomagot!

Indítsa el az EJADE Jade környezetet (és az RMA ágenst), majd indítson el egy-egy BookBuyer és BookSeller ágenst! Előbbinek adjon meg egy könyvnevet, utóbbinak pedig a katalógus elérését (milab/lab02/BookSellerAgent/catalog/catalog.xml).

Ha minden jól megy, akkor megjelenik az eladó grafikus felülete, valamint a konzolon az alábbi üzenet olvasható:

```
Hello! Buyer-agent BookBuyerAgent@mitpc11:1099/JADE is ready.  
Target book is könyv Minta  
Catalogue database of BookSellerAgent is  
milab/lab02/BookSellerAgent/catalog/catalog.xml
```

Az ágenseket figyelje meg az RMA ágens felületén is.

Indítson el egy Sniffer ágenst az RMA felületén, és kezdje el megfigyelni az előbb létrehozott két ágenst. Várjon egy kicsit, amíg kommunikálni kezdenek, majd nézzon bele az eladó és vevő ágensok közötti üzenetek tartalmába (kattintson kétszer a Sniffer ágens felületén látható nyilakra)!

- J1. Dokumentálja a konzolon és a Sniffer ágensben látottakat! A két ágens között milyen üzenetek mozognak? Ismertesse a felépítésüket! Milyen kommunikációs és tartalomnyelvi elemeket azonosított?

Adja meg a könyvtárnak a vevőnél megadott könyvet árral együtt, és figyelje meg az adás-vételi tranzakciót a konzolon valamint a sniffer ágensben.

- J2. Dokumentálja az adás-vétel üzenetváltásait is (készítsen egy üzenetlistát a Sniffer ágenssel: CFP, PROPOSE, ACCEPT-PROPOSAL és INFORM üzenetek)! Milyen kommunikációs és tartalomnyelvi elemeket alkalmaztak az ágensok?

Nyissa meg a fejlesztői környezetben a két ágens programkódját! Keresse meg az ACL üzenetek összeállítására szolgáló részeket (ACLMessage)! Nézze meg, hogyan kell címezteket és kommunikáció típust beállítani!

- J3. Dokumentáljon egy üzenet létrehozási lépést a jegyzőkönyvében kódrészletekkel és magyarázatokkal!

Ha maradt idő (szorgalmi feladat):

Alakítson ki versenyhelyzetet 4 eladó és két vevő között ugyanarra a könyvre különböző árakon!

- J4. Dokumentálja az adás-vétel lefolyását az ágensok között (Sniffer képernyőképpel is)!

2. Feladat: interakciós protokollok vizsgálata

A feladat célja: ágensek közötti interakciós protokoll megvalósításának vizsgálata

Referencia idő: 60 perc

Részfeladatok:

Az előző feladatban készített egy üzenetlistát az adásvételről, most ezt a párbeszédet fogjuk megvizsgálni részletesebben.

A párbeszéd létrehozásának feltétele, hogy az egymás utáni közléseket össze tudjuk kapcsolni. Erre szolgálnak az üzenetekben található `conversation-id`, `reply-with` és `in-reply-to` azonosítók.

- J1. Nézze végig az üzenetlistát, majd az üzenetek performatívája, párbeszéd azonosítói, valamint a tartalom szintben látható adatok alapján részletesen magyarázza el az alkalmazott FIPA-Contract-Net protokoll működését!

Nézzük meg a párbeszéd megvalósítását az ágensek kódjában!

Az `ACLMessage` osztály különböző metódusai alkalmasak a párbeszéd fenti azonosítóinak beállítására. Az ágens üzenetsablonok (`MessageTemplate`) segítségével választhatja ki a számára megfelelő üzeneteket. Az eclipse részletes leírást is megjelenít az osztályokról (az egérkurzorral az osztály neve felett megállva).

A vevő ágens kódjában keresse meg az adás-vételre vonatkozó részt (`RequestPerformer` osztály)! Az ágens a `step` változója segítségével azonosítja a saját állapotát. A „0” állapot kezelésének végén nézze meg az üzenetsablon beállítását!

- J2. Dokumentálja az adás-vétel egyes lépéseit az ágensek programkódja alapján (step 0-3), és nézze meg, hogyan állítja be az ágens az üzenet azonosítóit, valamint hogyan működik az üzenetszűrés!

Egészítse ki a vásárlási protokollt fizetéssel! A vevő ágensnél a legjobb eladótól érkező visszaigazolás (INFORM) után építsen be egy pénzáttutalási lépést (step 3 – PROPOSE: fizetés, step 4 – ACCEPT_PROPOSAL: fizetés elfogadva)! Az eladó `PurchaseOrdersServer` osztályában kell egy új viselkedést megadnia (`addBehaviour`) a fizetésre (pl. `PaymentPerformer` – a vevő ágens használhatja mintának).

Az üzenetek tartalmával ne foglalkozzon (csak a performatívákat figyeljék az ágensek), de ügyeljen arra, hogy csak a kijelölt vevőtől fogadja el a fizetést! A viselkedés lezárását (`done` metódus) se felejtse el!

- J3. Dokumentálja a jegyzőkönyvében az új kódrészleteket és a teljes adás-vételi üzenetváltást (CFP, PROPOSE, ACCEPT-PROPOSAL, INFORM, PROPOSE, ACCEPT-PROPOSAL)!

Ha maradt idő (szorgalmi feladat):

Próbálja ki, mi történik, ha a vevő fizetési ajánlatára az eladó nem válaszol!

Egészítse ki a fizetést egy árat tartalmazó tartalommal (a tartalomnyelv az árat tartalmazza egy egész számként), majd készítsen egy „csaló” eladót!

Védje meg a vevőt a megrendelés és a fizetés közötti árváltozástól, vagy valamilyen más Ön által választott, a FIPA-Contract-Net protokoll megsértése miatt bekövetkező hibától!

- J4. Írja le a kérdésre adott választ, és „csaló” ágens viselkedését és a vevőt védő programkiterjesztések működését!

3. Feladat: tartalomnyelvek használata (szintaktikai szint)

A feladat célja: a Java – XML nyelvi kötés megismerése, tartalomnyelv alkalmazása

Referencia idő: 60 perc

Részfeladatok:

Az eddigi feladatokban a kommunikációs protollokat vizsgáltuk meg, most rátérünk az üzenettartalom szintaktikai feldolgozására.

A laborban az üzenetek tartalomnyelvét XML deklarációkkal kell megadni. Ezeket automatikusan Java osztályokra fordíthatjuk le az adattartalom egyszerű elérése érdekében.

Nyissa meg a nyelvek (milab.lab02.language) között a lang_books.xsd fájlt, valamint a könyvárus ágens katalógust (milab.lab02.BookSellerAgent.catalog: catalog.xsd)!

Utóbbi több XML típust is deklarál, amik közül az egyikre a lang_books tartalom nyelv is hivatkozik. A könyvárus ágens katalógusa (catalog.xml) mutat egy példát a deklarációk alkalmazására. Nyissa meg azt a fájlt is!

Ismerkedjen meg a megnyitott állományok „Design” és „Source” nézeteivel! Vizsgálja meg a deklarációkat, azok kapcsolatait, valamint a minta XML katalógust!

- J1. Rögzítse a megfigyeléseit a jegyzőkönyvben! A katalógus XML deklarációja alapján próbálja meg a jegyzőkönyvében leírni a bookType típust, majd vesse össze a leírását a BookType.java osztállyal! Dokumentálja az összehasonlítást!

Keresse meg a vevő ágens kódjában a keresett könyvre vonatkozó CFP üzenetet, majd nézze végig a tartalomnyelv alkalmazásának lépéseit (JAXBElemént létrehozása Java osztályból, majd az üzenet tartalmának előállítása)! Helyezzen el egy töréspontot (ctrl+shift +b) itt a kódban, majd indítsa el az ágenst (az eladó ágenst is, ha az nem futna)!

Vizsgálja meg a futó program változóit a programkód lépésenkénti futtatása során (reqBook, contentElement és cfp.content), hasonlítsa össze azokat egymással, illetve az XML deklarációkkal!

- J2. Rögzítse a megfigyeléseit a jegyzőkönyvben! Mennyiben hasonlítanak egymásra a változók?

Változtassa meg a „rating” tulajdonság típusát egészre (xsd:int) a catalog.xsd „Design” nézetében, majd próbálja meg futtatni az eladó ágenst!

- J3. A jegyzőkönyvben írja le hol és miért jelentkezett a hiba, és hol és hogyan javította ki az XML dokumentumot (használja a Validate funkciót az XML dokumentum érvényességének ellenőrzésére a javítás előtt és után)! Képernyőképekkel is illusztrálja a hibát és a javítást!

Az előbb az XML deklarációt változtatta meg, ami konfliktusba került az XML fájl tartalmával. Ennek kijavítása után azonban nem oldott meg minden problémát. A deklaráció megváltoztatása után a Java nyelvi kötéshez új Java osztályokat kell generálnia.

Töröljön ki a catalog csomagból minden .java állományt, és generálja őket újra a JAXB 2.0 – Run XJC parancs segítségével! A csomag tartalmának frissítése (F5) után újra megjelennek a generált Java állományok.

Javítsa ki a Java nyelvi kötés megváltozása miatt keletkezett hibákat az eladó kódjában!

- J4. A jegyzőkönyvben írja le hol és miért jelentkezett a hiba, és hogyan javította ki!

4. Feladat: tartalomnyelvek használata (szemantikai szint)

A feladat célja: új tartalomnyelv kialakítása és használata, egyszerű ontológia alkalmazása

Referencia idő: 60 perc

Részfeladatok:

Ebben a gyakorlatban a könyvkeresést egészítjük ki kulcsszavas kereséssel, amihez egy új tartalomnyelvet alkotunk, illetve a fogalmak értelmezését segítő ontológiát is használni fogunk.

A bookType típus rendelkezik egy keywords opcionális elemmel, amely keyword elemek (stringek) sorozatát tartalmazza, valamint egy ontológia név is rendelhető hozzá. A catalog.xml fájlban vannak könyvekhez rendelt kulcsszavak.

Hozzon létre egy új tartalom nyelvet kulcsszavas keresésre! A lang_search nyelv (lang_search.xsd) jelenleg üres a projektben. A searchDataType típust deklarálja úgy, hogy a könyv nevére és kulcsszavaira kereshessen (string típusú titleSearch és keywordSearch elemek opcionális megadásával), valamint a kereséshez egy ontológiát is megadhasson (egy ontology nevű attribútummal)!

A deklaráció elkészítése után hozza létre belőle a Java nyelvi kötés objektumait (JAXB 2.0 – Run XJC)! A lang_search csomag frissítése (F5) után láthatóvá válnak a generált fájlok.

J1. Dokumentálja a létrehozott új tartalomnyelvet a jegyzőkönyvben!

Módosítsa az eladó és a vevő ágenseket úgy, hogy az új nyelvet használják az első üzenetváltás során a könyv címe helyett (a BookType helyett használjon SearchDataType típusú változót)! (A kulcsszavak lehetnek a kódban rögzítettek az egyszerűség kedvéért.)

A vevő esetében a viselkedés 0. lépésében kell módosítani az üzenet tartalom nyelvét. Az eladó OfferRequestsServer viselkedésében kell a vett üzenet tartalomnyelvét lecserélnie. Ellenőrizze a searchCatalog metódus működését is!

J2. Dokumentálja a jegyzőkönyvében az új kódrészleteket, az ágensek párbeszédében alkalmazott nyelveket és a helyes működést pl. „philosophy” keresési mintára (konzol kimenetet és Sniffer adatokat, képernyőképet mellékeljen)!

Módosítsa a vevő ágensét úgy, hogy adjon meg a keresési mintához egy ontológiát is (lehet a kódban rögzített értékű, de az ágens paraméterezését is megváltoztathatja)!

Az eladó ágensben a searchCatalog metódust módosítsa úgy, hogy az csak a megfelelő ontológia szerinti kulcsszavak között keressen!

J3. Dokumentálja a jegyzőkönyvében az új kódrészleteket és a helyes viselkedést pl. az „education” ontológiára vonatkozó „philosophy” kereséssel!

Ha maradt idő (szorgalmi feladat):

Valósítson meg egy ügyesebb keresést a kérés-kiterjesztés (query expansion) technikájával! A találati lista bővíthető, ha a vevőtől érkező kulcsszavakat egy ontológia segítségével kiegészíti (felcseréli) kapcsolódó fogalmakkal.

Tanulmányozza a SimpleOntology osztály metódusait és ScienceOntology osztályban deklarált fogalmi kapcsolatokat, majd egészítse ki az eladó ágens katalóguskeresési eljárását ezzel a technikával! Próbaképpen keressen fizikával vagy filozófiával kapcsolatos könyvekre egy, a könyveknél kulcsszóként nem szereplő ontológia fogalom megadásával!

J4. Dokumentálja a keresési kiegészítéseit a jegyzőkönyvben!