
PROJECTE FINAL DE CARRERA



UNIVERSITAT
ROVIRA I VIRGILI



Diseny i implementació d'un Sistema Multi-Agent per a l'assignació d'òrgans basat en Mètodes de decisió Multi-criteri

David Sánchez Ruenes, dsr.si@alumne.etse.urv.es

Enginyeria Tècnica Informàtica de Sistemes

Professors:

Aïda Valls, URV, avalls@etse.urv.es

Antonio Moreno, URV, amoreno@etse.urv.es

Escola Tècnica superior d'Enginyeria (ETSE)
Universitat Rovira i Virgili (URV), <http://www.etse.urv.es>

Curs 2000-2001

ÍNDIX

1. INTRODUCCIÓ

Els trasplantaments d'òrgans tenen una importància capdal dins del camp de la Medicina. Tot i això, la quantitat de variables que intervenen en el procés des que es disposa d'un donant fins que es fa el trasplantament, fan que l'elecció i coordinació de personal, equips i pacients sigui una feina força complexa que requereix temps (en alguns casos un factor crític) i diners. En aquest punt, el model de coordinació Espanyol, està demostrant ser un dels més efectius i que, juntament amb quantitat de donants, ens han situat com un país capdavanter dins del panorama mundial.

Una de les decisions més importants que s'ha de prendre en el moment de rebre un òrgan, és l'elecció del pacient que el rebrà. En aquest procés intervenen una gran quantitat de factors que s'han de tenir en compte per a cada cas. En la gestió d'aquest punt, el model Espanyol divideix el territori seguint una organització jeràrquica (hospital -> ciutat -> província -> zona -> país), de forma que s'intenti trobar el receptor ideal més proper, minimitzant el cost temporal i econòmic. Aquesta elecció es realitza de la manera tradicional: el personal de l'hospital intenta trobar un receptor dins del propi centre, el cas de no trobar-lo, es posa en contacte amb el personal d'una altre hospital dins del seu territori; si encara no es troba es passarà a un nivell més general dins de la jerarquia territorial. Com es veu, el procés d'elecció resulta força complicat (ja que s'hauran d'avaluar tots el pacients disponibles i decidir si són o no vàlids) i requereix un treball de coordinació important. És precisament en aquest punt on les tècniques d'intel·ligència artificial poden servir d'ajuda per tal d'aconseguir optimitzar el procés.

Tal com s'ha descrit anteriorment, els principals problemes amb què ens trobem en el moment d'escollir el receptor del donant són:

- Decidir quin serà el pacient o pacients més favorables per a l'òrgan disponible en aquest moments.
- Coordinar els diferents centres per tal d'avaluar les llistes de pacients.

Són precisament aquest dos punts els que tractarà aquest projecte, proposant un mecanisme que ajudi a l'automatització d'aquests processos, minimitzant el cost temporal (factor que en aquest punt es fa crític).

Dins del primer problema, ens trobem amb la dificultat de la comparació i avaluació de pacients vers el donant, ja que intervenen un quantitat considerable de factor (característiques físiques, temps d'espera, distància...) que tindran un pes més o menys important dins de la decisió final. Les persones encarregades d'aquest procés requereixen uns coneixements i una experiència considerables per tal d'analitzar conjuntament aquestes dades i extreure'n un resultat. El nostre objectiu no és substituir-les per un programa informàtic, doncs en un procés tan crític com el que estem tractant, les reticències i els prejudicis ho farien inviable, sinó que tractarem de construir una eina que ajudi a l'especialista a prendre la decisió més ràpidament i fàcilment.

La principal dificultat amb què ens enfrontem arribats a aquest punt es la manera com comparar les alternatives (pacients). Molts dels factors que intervenen en l'elecció d'un o un altre individu, no poden expressar-se d'una manera exacta, perquè no és disposa d'aquesta informació, no és necessària molta precisió, o són termes que resulten difícils d'expressar quantitativament. En aquest cas, a la persona li resulta més fàcil utilitzar termes més aviat 'psicològics' (qualitatius) que expressin de forma per general (i imprecisa) el que es vol representar. Davant d'aquest fet, ens trobem amb la dificultat d'establir rangs definits que ens permetin obtenir una transformació numèrica (fàcilment avaluable) de les dades ja que els límits entre uns i altres termes qualitatius són difusos.

Aquest fet, juntament amb la quantitat de factor que intervenen en el procés de decisió, fan que els algorismes estàndards d'ordenació no siguin aplicables i, per tant, es requereixin tècniques d'intel·ligència artificial.

A més, el resultat que obtingui el programa, no ha de ser taxatiu (doncs les persones són reticents a que una màquina pugui substituir el seu intel·lecte); per tant, ha d'oferir a l'especialista que supervisa el procés un rang de maniobra, de forma que l'última decisió sempre sigui seva. En aquest punt, el programa ha de ser capaç d'oferir alternatives i expressar-les també d'una forma 'amigable' per a la persona (qualitativament), així com ser capaç de rebre valoracions sobre els resultats i 'aprendre' d'ells.

Per tal d'aconseguir aquest comportament, s'aplicaran tècniques d'ajuda a la pressa de decisions multicriteri (MCDM: *Multi-criteria decision Making*) degut a la quantitat de factors que intervenen en el procés de decisió.

Per altra banda, un altre procés serà l'encarregat de gestionar i avaluar el seu comportament, de forma que sigui capaç d'adaptar-se al criteri de l'usuari. És en aquest punt on intervenen els Agents, sistemes que gràcies a les seves característiques (explicades més endavant), resulten adequats en la realització de tasques que simulen un comportament humà en situacions difícils.

Pel que fa a la coordinació entre personal i centres, es pretén automatitzar el procés de forma que la comunicació es realitzi mitjançant un xarxa global (amb la qual cosa es minimitzaria el cost temporal). Per aquesta raó, es proposa un organització basada en un sistema Multi-Agent, on cadascuna de les entitats estigui situada en un lloc determinat i encarregada d'una feina concreta (en aquest cas l'elecció del pacient adequat), sent capaç de comunicar-se (per consultar dades o transmetre resultats) a través de la xarxa.

Aconseguir un sistema efectiu basat en Agents que tracti tots els aspectes del trasplantament d'òrgans (elecció del pacient, transport, coordinació de personal mèdic...), seguint el model Espanyol és l'objectiu que desenvolupa actualment el grup de Recerca en Intel·ligència artificial de la Universitat Rovira i Virgili. Aquest projecte tracta d'un dels aspectes mencionats anteriorment (ajuda a la l'elecció del(s) pacient(s) ideals), definint i implementant un part del sistema Multi-Agent global.

2. AGENTS I SISTEMES MULTI-AGENT (SMA)

Com en la majoria dels casos en que s'intenta atribuir qualitats "intel·ligents" a un procés computacional, la definició d'Agent no és clara. Es podria descriure com un sistema que habita en un entorn complex i dinàmic; és capaç de sentir i actuar autònomament en el seu entorn, i té un conjunt d'objectius o motivacions que intentarà aconseguir a través de les seves accions [1].

2.1. Propietats dels agents

Les propietats que ha de complir un procés computacional per tal de considerar-se un Agent són:

- Autonomia: és la capacitat d'operació sense la intervenció directe dels humans o altres, i tenir algun tipus de control sobre les pròpies accions i l'estat intern.
- Sociabilitat/cooperació: els agents han de ser capaços d'interactuar amb altres agents i/o humans a través d'algun tipus de llenguatge de comunicació.
- Reactivitat: els agents perceben el seu entorn i responen en un temps raonable als canvis detectats.
- Pro-activitat o capacitat de prendre iniciativa: han de ser capaços d'exhibir algun tipus de comportament encaminat a assolir algun tipus d'objectiu prenent la iniciativa.
- Mobilitat: possibilitat de moure's a altres entorns a través d'una xarxa electrònica.
- Continuïtat temporal: els agents estan contínuament executant processos.
- Veracitat: es l'assumpció que un agent no comunicarà informació falsa premeditadament.
- Benevolència: es l'assumpció que els agents no tenen objectius conflictius, i que cada agent intentarà fer allò pel qual és requerit.
- Racionalitat: l'agent ha d'actuar per tal d'aconseguir el seu objectiu.

2.2. Arquitectures d'agents

Les architectures utilitzades en el procés d'implementació dels agents poden ser de tres tipus:

- Deliberatives: donada una representació del món real, es poden executar accions intel·ligents. Segueix el model BDI (*Beliefs Desires and Intentions*).
- Reactives: estructura on les accions es duen a terme responenent a estímuls. La combinació de diversos agents reactius proporciona un comportament "intel·ligent".
- Híbrides: estructures que barregen diferents tècniques.

2.3. Tipus d'agents

Segons la finalitat amb la que s'implementi un agent, podem classificar-lo en diferents tipus:

- D'informació: gestionen i manipulen dades; poden respondre a requeriments del usuari o altres agents. Un exemple molt comú són els cercadors d'informació a Internet.
- D'interfície: sistema on els agents col·laboren amb l'usuari per resoldre un problema. La interacció amb l'individu permet a l'agent desenvolupar un aprenentatge basat en les accions que es desenvolupen.
- De col·laboració: la característica principal és la comunicació i cooperació amb altres agents per resoldre un problema comú. Utilitzen tècniques d'IA distribuïda.
- Mòbils: la seva característica principal és la possibilitat de poder moure's per una xarxa electrònica recollint informació o interactuant amb altres hosts.
- Reactius: el seu comportament es basa en la resposta a estímuls segons un patró de l'estat actual en què es troben. Són molt simples, però la combinació de diversos agents reactius pot generar comportaments complexos.
- Híbrids: són combinacions de diversos tipus anteriors intentant reunir les avantatges d'uns i altres.
- Heterogenis: es refereixen a un conjunt integrat d'almenys dos agents que pertanyen a dues o més classes diferents.

2.4. Sistemes Multi-Agent (SMA)

Un sistema Multi-Agent serà aquell en el que un conjunt d'Agents cooperen, es coordinen i es comuniquen per tal d'aconseguir un objectiu comú.

2.4.1. Avantatges d'un SMA

Les principals avantatges de la utilització d'un sistema Multi-Agent en la gestió d'un problema són:

- Modularitat: es redueix la complexitat en treballar amb unitats més petites, permetent una programació més estructurada.
- Eficiència: la programació distribuïda permet repartir les feines entre els agents, aconseguint paral·lelisme (agents treballant en diferents màquines).
- Fiabilitat: el fet que un element del sistema deixi de funcionar no ha de significar que la resta també ho faci; a més, es pot aconseguir més seguretat replicant serveis crítics i així aconseguir redundància.
- Flexibilitat: es poden afegir i eliminar agents dinàmicament.

2.4.2. Administració d'un SMA

L'administració d'agents estableix el model lògic per a la creació, registre, comunicació, establiment, mobilitat i destrucció d'agents. La FIPA (*Foundation for Intelligent Physical Agents*, organització que promou el desenvolupament d'aplicacions basats en agents, definit estàndars) estableix el següent model per aquest entorn d'administració d'agent [10]:

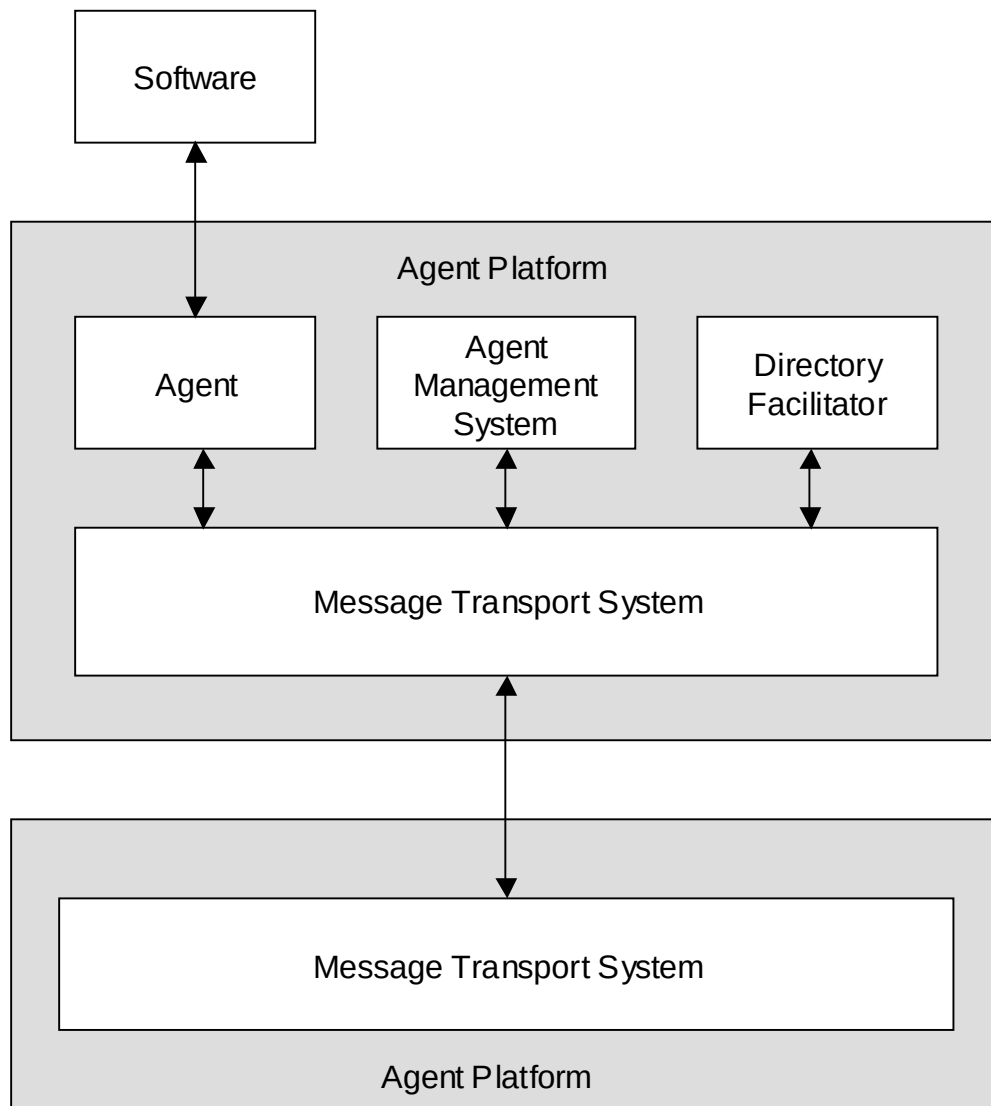


Figura 1- Aspecte intern d'un Sistema Multi-Agent

Els components lògics principals que formen part del sistema de la figura anterior són:

- *Agent*: unitat bàsica. Es pot descriure com un programa que encapsula una sèrie de serveis
- *Directory Facilitator (DF)*: és un agent que proporciona un servei de Pàgines Grogues dins del sistema (coneix els serveis que poden oferir els diferents agents del SMA). Els agents han de registrar-se en el DF per a oferir els seus serveis.

- *Agent Management System (AMS)*: és l'agent que controla l'accés i ús de la plataforma. Emmagatzema les direccions dels agents, oferint un servei de Pàgines Blanques.
- *Message Transport System (MTS)*: és el mètode de comunicació entre agents de diferents plataformes.
- *Agent Platform (AP)*: proporciona la infraestructura bàsica en al que poden crear-se i executar-se els agents.
- *Software*: descriu qualsevol programa accessible des d'un agent.

2.4.3. Llenguatge de comunicació entre agents (ACL)

Els agents individualment proporcionen una funcionalitat interessant, però el que els fa tant adequats per a determinats sistemes és la seva capacitat de cooperar per a resoldre problemes. Per poder fer-ho, els agents s'han de comunicar utilitzant un llenguatge comú : ACL (*Agent Communication Language*).

Els ACL estan basats en la teoria de la parla, en la qual les comunicacions individuals entre agents poden reduir-se a un petit nombre d'idees o més generalment actes comunicatius. Aquests actes donen forma al significat bàsic de la comunicació.

L'element bàsic de la comunicació és el missatge; la FIPA determina quina forma ha de tenir un missatge i la seva utilització [11].

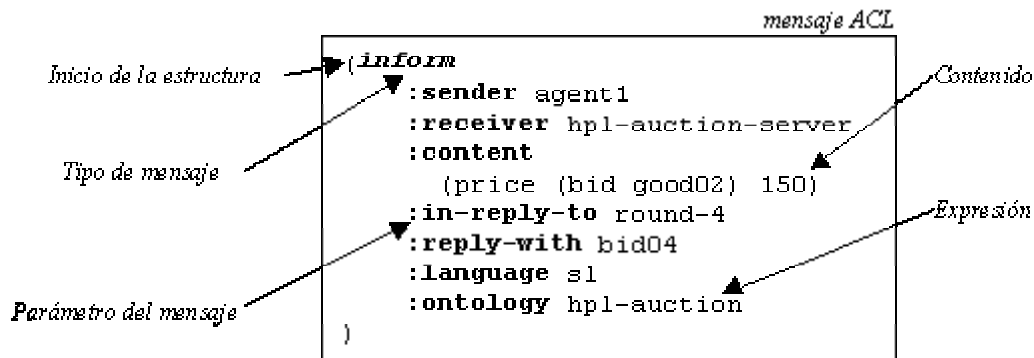


Figura 2 - Aspecte d'un missatge

2.4.3.1. Elements d'un missatge

Els elements que de forma estàndard formen part d'un missatge de comunicació entre agents són:

- Definició de tipus de missatge (*performative*): indica el tipus d'acte de comunicació. Les accions més habituals són:

- o **accept-proposal**: acceptació una proposta feta anteriorment per realitzar una acció.
 - o **agree**: acceptació d'una acció.
 - o **cancel**: cancel·lació d'una acció demanada anteriorment.
 - o **cfp**: crida per proposar la realització d'una acció.
 - o **confirm**: l'emissor informa al receptor que una proposició és certa, quan el receptor dubtava sobre la proposició.
 - o **disconfirm**: l'emissor informa al receptor que una proposició és falsa, quan el receptor creia que era certa.
 - o **failure**: indica que l'acció demanada anteriorment ha fallat per algun motiu.
 - o **inform**: l'emissor informa al receptor que una acció s'ha realitzar correctament.
 - o **not-understood**: l'emissor informa al receptor que no ha entès una petició realitzada anteriorment.
 - o **propose**: acció de proposar la realització d'una acció, donades certes precondicions.
 - o **query-if**: acció de preguntar a un altre agent si un fet és cert o no.
 - o **refuse**: negació a la realització d'una acció donada.
 - o **request**: l'emissor demana al receptor realitzar un acció.
 - o **subscribe**: l'emissor demana ser avisat en el moment que es compleixi una condició.
- Interlocutors: són l'identificadors d'agents. N'hi ha 3: **sender** (qui envia el missatge), **receiver** (qui el rep), **reply-to** (a qui ha d'anar destinat el següent missatge de la conversa).
 - Contingut del missatge: existeixen quatre tipus de continguts predefinits que s'utilitzen en funció de les necessitats que es tinguin.
 - o FIPA-CCL (*Constraint Choice Language*): defineix una semàntica que permet especificar predicats amb satisfacció de restriccions (CSP).
 - o FIPA-SL (*Semantic Language*): permet formar expressions lògiques, intercanviar coneixement de forma òptima i expressar accions a realitzar. És el llenguatge més general de tots i pot ser aplicat a molts dominis diferents. És el que s'utilitzarà en aquest projecte.
 - o FIPA-KIF (*Knowledge Interchange Format*): permet expressar objectes com a termes i proposicions com a sentències.
 - o FIPA-RDF (*Resource Description Framework*): permet representar objectes, interaccions i accions entre ells.
 - Descripció del contingut: permet que l'agent que rebí el missatge pugui identificar què se li està enviant i en quin format. Es defineixen 3 descriptors a tal efecte:
 - o **language**: en què està escrit el contingut, en aquest cas utilitzarem el FIPA-SL.
 - o **encoding**: indica si s'utilitza algun tipus de codificació especial.
 - o **ontology**: és el camp més important, ja que defineix la ontologia utilitzada per donar significat al contingut del missatge.

- Control de la conversa: està destinat a identificar quin tipus de converses es mantenen. Els seus camps informatius són:
 - o **protocol** utilitzat en el missatge.
 - o **conversation-id** identificador de conversa.
 - o **reply-with** identificador de missatge que s'utilitza per seguir els diferents passos de la conversa.
 - o **in-reply-to** identifica una acció passada a la qual aquest missatge és la resposta.
 - o **reply-by** marca de *timeout*: data/hora de caducitat del missatge.

2.4.3.2. Protocols de comunicació

Abans s'ha indicat que uns dels camps de control de conversa identifica el tipus de protocol que es fa servir. El protocol utilitzat és el que defineix una sèrie de regles o passos que s'han de seguir per a desenvolupar una conversa. N'hi ha de diferents tipus depenent de la seva finalitat. El conjunt de protocols disponibles i el seu funcionament també són especificats per la FIPA [7]:

- FIPA *Request*: permet demanar la realització d'accions i retornar els resultats
- FIPA *Query*: s'utilitza per fer preguntes
- FIPA *Contract Net*: és un protocol de negociació (es proposen i s'avaluen propostes)
- FIPA *Iterated Contract Net*: variant de l'anterior que permet la renegociació
- FIPA *Brokering*: gestiona els serveis disponibles pels agents
- FIPA *Recruiting*: variant de l'anterior
- FIPA *Subscribe*: permet la notificació de fets importants
- FIPA *Propose*: simplificació del *Contract Net*

3. JADE

JADE (*Java Agent Development Enviroment*) és una eina de programació que proporciona un conjunt de llibreries escrites en JAVA per al desenvolupament de Sistemes Multiagent.

A més de proporcionar les funcions de manegament d'agent a baix nivell i les interfícies que faciliten la programació, també ens presenta un entorn d'execució on els agents podran executar-se i interactuar.

3.1. Packages de JADE

JADE està compostat per una sèrie de paquets escrits en JAVA. Els principals són els següents:

- **jade.core**: és el nucli del sistema. Proporciona la classe **jade.core.Agent** que és imprescindible per a la creació de SMA. També conté el paquet **jade.core.behaviour** que conté les classes de comportaments suportats.
- **jade.lang**: conté les classes específiques per a suportar un llenguatge de comunicació entre agents (ACL). Està format pels paquets **jade.lang.acl** i **jade.lang.sl**.
- **jade.domain**: conté totes les classes per representar la plataforma d'agents i els models del domini, tals com entitats per a l'administració d'agents, llenguatges i ontologies (AMS, DF, ACC ...).
- **jade.proto**: paquet que conté els protocols d'interacció entre agents FIPA.
- **jade.tools**: trobem algunes eines que faciliten el desenvolupament d'aplicacions i l'administració de la plataforma:
 - o *Remote Management: Agent (RMA)*: actua com una consola gràfica per a l'administració i control del sistema.
 - o *Dummy Agent*: és una eina de monitorització i depuració, composta per una interfície gràfica i un agent JADE. Permet crear missatges ACL i enviar-los a altres agents, podent visualitzar-los.
 - o *Sniffer*: agent que intercepta missatges ACL i els visualitza.
 - o *SocketProxyAgent*: agent que actua com un gateway bidireccional entre la plataforma JADE i una connexió TCP/IP.

3.2 Components principals de JADE

Per JADE, els agents resideixen en un entorn predefinit anomenat plataforma. Cada plataforma està dividida en contenidors i cadascun d'ells contindrà agents. Els contenidors poden entendre's com dominis d'agents.

La plataforma s'engega en una màquina determinada, mentre que els agents podran executar-se en qualsevol estació.

Tal com es defineix a la FIPA [7], en cada plataforma s'inicien dos agents que tenen unes funcions vitals per al funcionament del sistema:

- *Domain Facilitator* (DF): pot ser definit com un servei de “pàgines grogues”, és a dir, contindrà tots els serveis que proporcionen els agents que hi estan donats d'alta.
- *Agent Management System* (AMS): es podria considerar com un servei de “pàgines blanques”, on es registren les adreces dels agents que s'han donat d'alta.

Gràcies a aquests serveis, es proporciona molta flexibilitat i transparència. D'aquesta manera, si un agent A vol enviar un missatge a un agent B que desenvolupa la tasca T, l'agent A preguntarà al DF quin o quins agents poder fer-se càrrec de la tasca T. Quan es vulgui conèixer l'adreça d'un agent determinat, preguntarà a l'AMS. Per tal que aquest sistema funcioni correctament, caldrà que, durant el procés d'inicialització de l'agent, aquest informi al DF de qui o quins serveis proporciona i quin tipus d'agent és (per tal de poder identificar-lo); en fer-ho, l'AMS li donarà una adreça física (depenent d'on s'executi), per tal que altres es puguin posar en contacte amb ell.

3.3. Agents i comportaments

Per poder crear agents en JADE, haurem d'extendre la classe **Agent** que ja es troba definida i anar omplint els camps i les funcions bàsiques dels quals ens proporciona la interfície. Dos mètodes molt importants que s'han d'implementar necessàriament són **setup()** i **takedown()** que inicialitzen i finalitzen l'agent respectivament.

Un cop creat l'agent, haurem de donar-li funcionalitat. Això s'aconsegueix mitjançant la definició de *behaviours* (comportaments) que són els que controlen les seves accions. Un *scheduler* intern serà l'encarregat d'anar controlant automàticament l'execució de tots els comportaments.

Arribats a aquest punt, JADE ens proporciona un conjunt de comportaments bàsics que permeten implementar diferents protocols de comunicació.

Dintre del grup de comportaments simples (sense fills) trobem els següents:

- **SimpleBehaviour**: representa un comportament atòmic que s'executa sense interrupcions. Pot ser de tipus **OneShotBehaviour** (si s'executa una sola vegada), o **CyclicBehaviour** (si executa una acció cíclicament).
- **ReceiverBehaviour**: permet engegar un comportament que espera la recepció d'un missatge.

- **SenderBehaviour**: és equivalent a l'anterior però des del punt de vista del qui envia el missatge.

El primer tipus ens permet implementar accions bàsiques que durà a terme l'agent de forma independent (consultar dades a una BD, interactuar amb l'usuari...), mentre que el segon i el tercer ens permet implementar protocols de comunicació entre agents.

A més, també existeixen comportaments complexos que ens permetran crear fills i atendre diverses peticions alhora. S'agrupen dins de la classe abstracte **ComplexBehaviour** i té dos subtipus:

- **SequentialBehaviour**: executa tots els seus fills de manera seqüencial i acaba quan tots els mètodes finalitzen correctament.
- **NonDeterministicBehaviour**: executa els seus fills segons una seqüència no determinista.

3.4. Missatges

Pel que fa a la comunicació, JADE segueix les especificacions de la FIPA en quant a format dels missatges [11]. Així, un missatge estarà dividit en *slots* que ens donen una informació (llenguatge, protocol, contingut...), i sobre aquests, es defineixen funcions de creació (**set**) i consulta (**get**).

Els mètodes bàsics de què disposem per al manegament dels missatges entre agents són:

- **Send (ACLMessage)**: utilitzat per enviar un missatge un cop omplerts tots els seus paràmetres.
- **ACLMessage receive()**: rep un missatge de la cua de l'agent. Es pot especificar quin tipus de missatge es vol rebre mitjançant els patrons de cerca (*MessageTemplate*).

El llenguatge que utilitzarem en el nostre cas per escriure els missatges serà el FIPA-SL (*Semantic Language*), que és el que té un àmbit més general: permet formar expressions lògiques, intercanviar informació entre agents i expressar accions a realitzar. La notació utilitzada és semblant al Lisp: es defineixen un conjunt de predicats genèrics i cadascun amb una sèrie d'atributs i paràmetres.

3.5. Ontologia

Un dels camps que haurem de definir en el moment d'enviar un missatge, és l'ontologia que utilitzarem per escriure el contingut. Aquest punt és molt important, ja

que defineix el vocabulari emprat (i per tant el significat i la forma d'accedir a les dades). Definir una ontologia adaptada al problema que volem tractar, és un dels elements més importants i que cal haver decidit completament abans d'implementar els agents. Una ontologia permet definir els següents punts:

- *Frames* disponibles per omplir el missatge.
- *Slots* que formen cada *frame*.
- *Slots* que alhora són *frames* (per poder representar dades complexes).
- Etiqueta (amb la que s'accedirà) i tipus de dades (String, long o compost) que conté cada *slot*.

A més de definir l'ontologia en un mòdul a banda (*frames*, *slots* i tipus de dades que contenen), també serà necessari definir i implementar una classe que s'encarregarà de manegar cada objecte o predicat (*frame*) creat. Aquesta classe haurà de contenir mètodes **set** i **get** per a cada *slot* que haguem definit. Aquestes funcions seran les que ens permetran consultar i omplir les dades del missatge.

3.6. Protocols de comunicació

Un cop definida i implementada la ontologia, ens caldrà escollir el protocol de comunicació entre agents que més s'adapti a les nostres necessitats. En aquest punt, JADE també ens proporciona els element necessaris (*behaviours* i *performatives*) per definir els protocols descrits per la FIPA [7]. Per a cada conversa, JADE distingeix dos rols: el que la inicia (*iniciator*) i el que la segueix (*responder*) i cada agent haurà d'extendre els comportament en funció del seu paper.

3.6.1. Tipus de protocols

Tot i això, no tots els protocols estan disponibles per a la seva implementació en JADE, només podrem utilitzar els següents:

3.6.1.1. FIPA Request

És el més comú i s'utilitza per demanar la realització d'accions. El comportament que tindran els agents quan utilitzen aquesta especificació és el següent:

- Un agent *iniciator* comença el procés quan vol que un agent *responder* realitzi alguna acció.
- L'agent *iniciator* envia un missatge de tipus **request** cap a l'agent amb el que vol interactuar (*responder*).

- El receptor del missatge llegeix el contingut del missatge i obté l'acció que se li demana. L'agent estudiarà la petició i determinarà si la pot fer o no.
- Si la pot realitzar, enviarà un missatge **agree** a l'agent inicial i començarà a realitzar l'acció.
- Si per alguna raó no pot realitzar-la, enviarà un missatge **refuse** que informi de la raó d'aquesta negació.
- En qualsevol cas, si l'*iniciator* no ha formulat bé el missatge i, per tant, no es pot desxifrar correctament, s'enviarà un missatge **not-understood**.
- En cas d'haver enviat un missatge d'**agree**, el *responder* haurà començat a realitzar l'acció. Quan l'acabi, enviarà un missatge de tipus **inform**. Aquest pot ser de dos tipus: **inform-done** que només notifica que l'acció ha estat finalitzada i **inform-ref**, que retorna un resultat.
- Si l'acció no s'ha finalitzat correctament, s'enviarà un missatge **failure** indicant la raó.

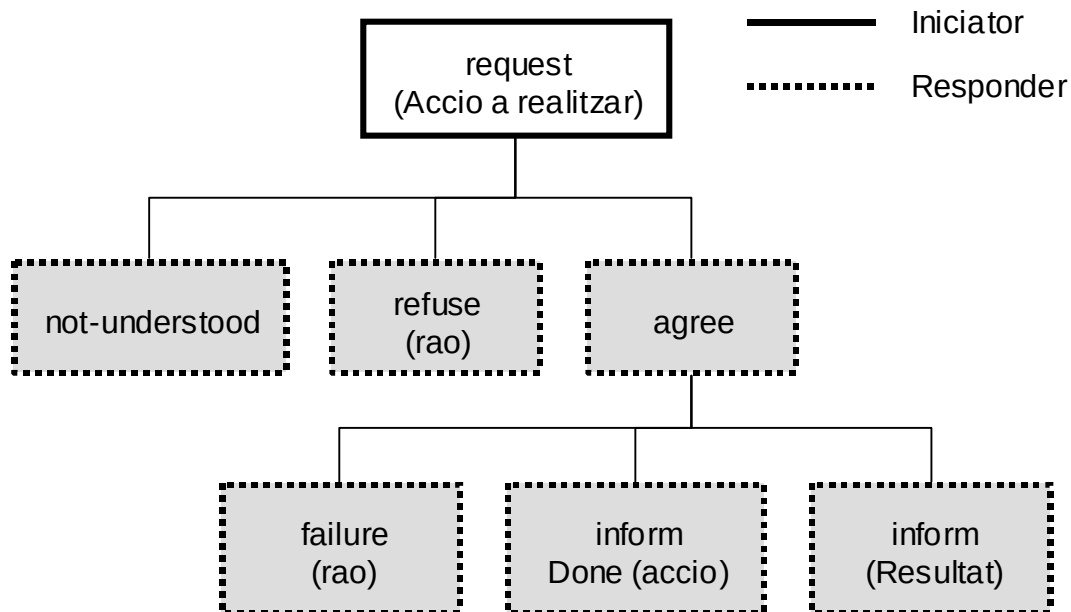


Figura 3 - Protocol FIPA-Request

Tenint en compte les característiques d'interacció entre agents que es necessiten en aquest projecte (descrites més endavant), aquest serà el comportament escollit per enviar missatges.

3.6.1.2. FIPA Query

S'utilitza quan es vol demanar algun tipus d'informació. Els passos que se segueixen durant la conversa són els següents:

- L'agent *initiator* formula una pregunta de tipus **query** o **query-ref**.
- L'agent *responder* podrà enviar la informació de la resposta o bé refusar la pregunta.
- Com sempre, si hi ha un error s'enviarà una fallada i si no se sap interpretar el contingut del missatge, un **not-understood**.

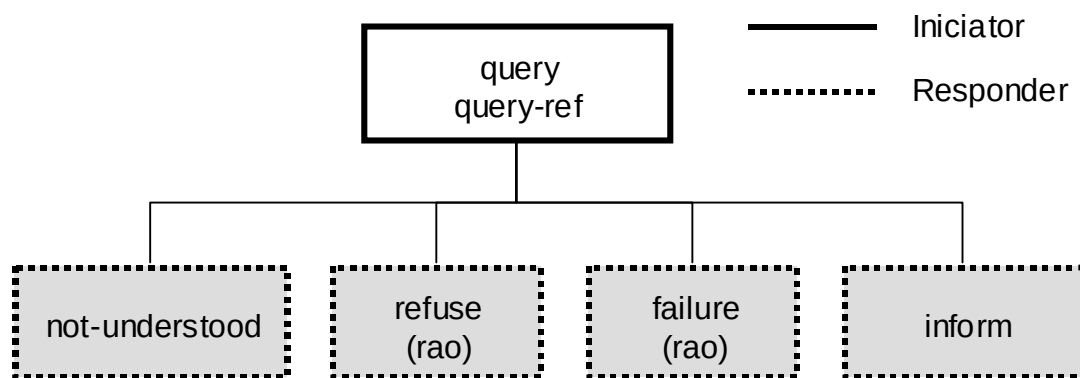


Figura 4 - Protocol FIPA-Query

3.6.1.3. FIPA Contract Net

Aquest protocol s'utilitza quan es vol realitzar una negociació d'accions entre diversos agents, per trobar aquell qui la pugui fer satisfent-la al màxim. Els passos que se segueixen són els següents:

- L'agent *initiator* envia una petició de proposta d'acció passant-li unes precondicions a complir.
- L'agent *responder* avaluarà les premisses rebudes i, en funció de quines siguin proposarà realitzar o no l'acció. En el primer cas, enviarà una proposta d'acció amb unes condicions determinades; en el segon, un missatge de refús. Per tal que l'agent inicial sàpiga quan ha acabat de rebre totes les propostes a la seva petició, es disposa d'un *time-out* màxim.
- Si s'ha enviat una proposta, l'agent inicial l'avaluarà, i podrà o no acceptar-la enviant a l'agent *responder* el missatge corresponent.
- Finalment, l'agent *responder* començarà a realitzar l'acció, enviant un missatge de finalització (**inform**) o una fallada (**failure**) en cas de produir-se algun error.

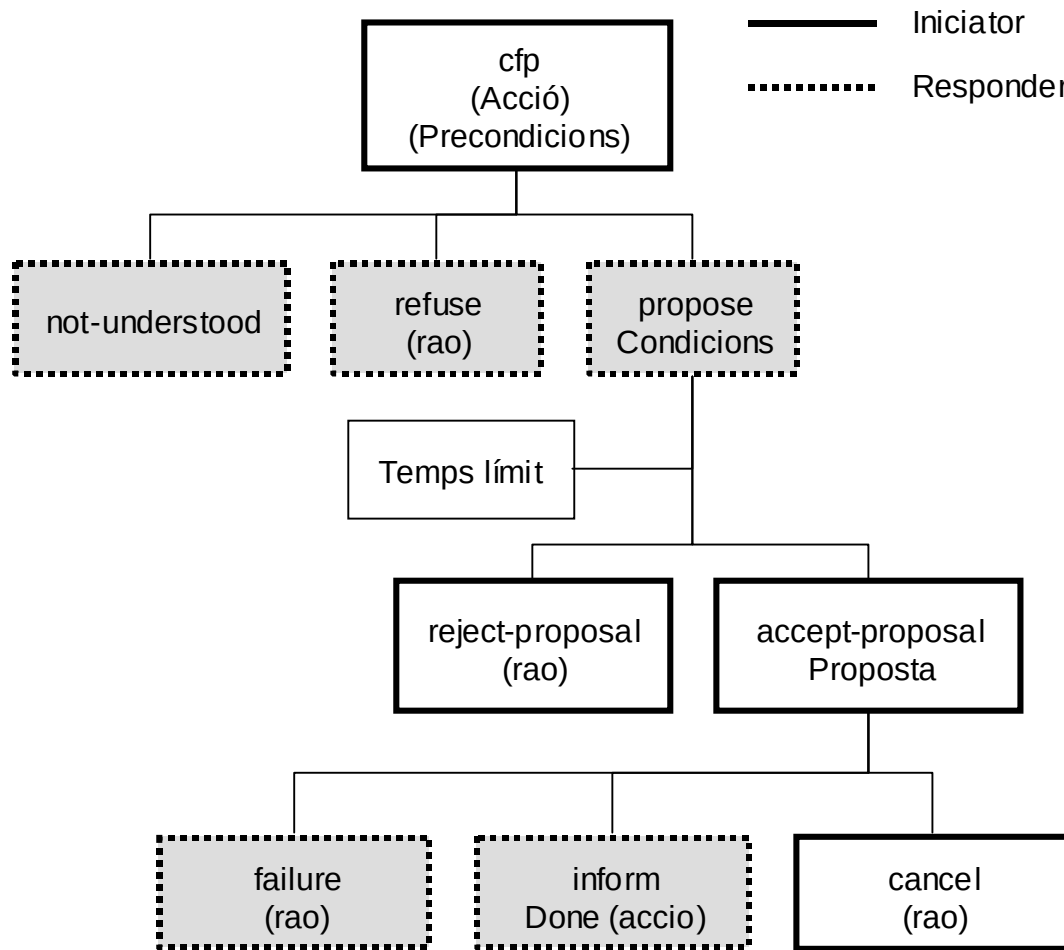


Figura 5 - Protocol FIPA-Contract Net

3.6.2. Implementació de protocols

Tal com s'ha indicat abans, per poder implementar aquest protocols, JADE els divideix en dos parts: la que fa la crida inicial (*initiator*) i la que manega la resposta (*responder*). Cadascuna d'aquestes parts és una classe abstracta de nom **xxxInitiatorBehaviour** i **xxxResponderBehaviour**, on xxx variarà segons el protocol utilitzat. Per a cada comportament, el programador, només haurà de sobreesciure aquells mètodes que manegen els diferents estats pels quals passa la conversa.

Per exemple, en cas d'utilitzar un protocol de tipus FIPA *Request* (el que s'ha implementat en aquest projecte), necessitarem dues subclasses:

- **FIPARequestInitiatorBehaviour**: la implementa l'agent *initiator* que comença l'enviament del **request** i manega les possibles respostes. En el protocol es considera una primera resposta on el *responder* informa si pot fer o no l'acció i, en cas afirmatiu, s'esperarà a rebre el resultat.

- **FIPAResponseResponderBehaviour**: implementa l'altre costat de la conversa. Esperarà a rebre *requests* i avaluarà l'acció: si la pot realitzar enviarà una confirmació i començarà a fer-la, enviant més tard el resultat; en cas contrari, la refusarà.

3.7. Consideracions generals

En resum els principals punts que s'han de considerar en la implementació d'agents en JADE són els següents:

- Definició i implementació d'un llenguatge de comunicació comú entre els agents del sistema -> ontologia.
- Associació i implementació de comportaments (*behaviours*) per a cada agent. La combinació d'aquests, ens permetrà definir protocols de comunicació.
- Cal que els agents publicitin al DF els serveis que proporcionen per tal que altre agents els puguin utilitzar.

3.8. Eines de JADE

JADE ofereix una interfície gràfica per a l'administració de la plataforma, així com eines per tal de facilitar la depuració i testeig de les aplicacions basades en ell.

Per poder iniciar una sessió i crear els agents, haurem d'escriure la següent comanda:

```
java jade.Boot -platform [opcions] [llista d'agents]
```

En el camp d'opcions podem especificar que volem visualitzar l'entorn gràfic amb el paràmetre `-gui`. Per que fa a la llista d'agents, serà una cadena de text on cadascun d'ells estarà separat per espais en blanc i seguirà el següent format: nom de l'agent, dos punt i la classe Java que l'implementa.

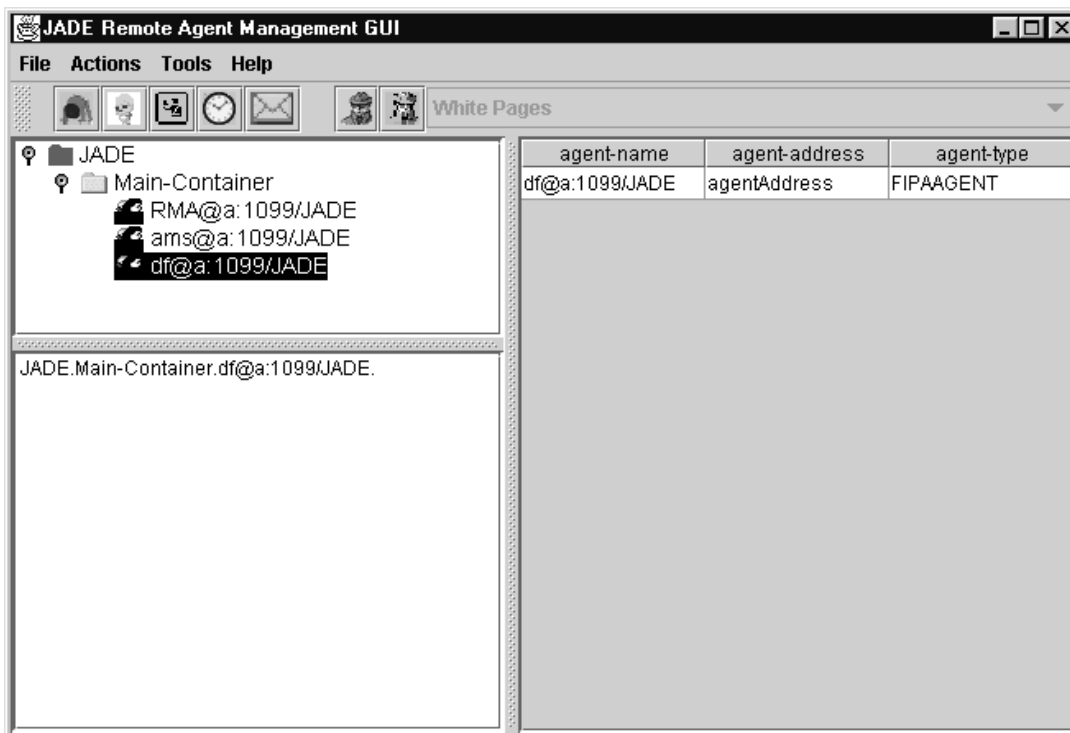


Figura 6 - Pantalla principal de JADE

Quan engeguem l'entorn gràfic, podem observar com, tot i que no s'inicialitzi cap agent propi, es creen uns quants automàticament:

- L'agent RMA serà l'encarregat de controlar la interfície
- El DF, on se subscriuran els serveis dels agents.
- L'AMS on es guardaran les adreces dels agents.

A més de la llista d'agents que hi ha al contenidor i la informació sobre el que hem seleccionat, disposem d'una sèrie d'opcions (botons o menú *tools*) que ens facilitaran el manegament i testeig dels agents:

- Afegir un nou agent: haurem d'indicar nom, la classe que l'implementa i el contenidor al que pertany.
- Matar els agents seleccionats.
- Executar els agents seleccionats.
- Suspendre els agents seleccionats.
- Continuar els agents seleccionats.

- Enviar un missatge als agents seleccionats: haurem d'omplir una finestra amb els camps corresponents: qui l'envia, qui el rep, protocol, acte comunicatiu, llenguatge, ontologia, contingut...

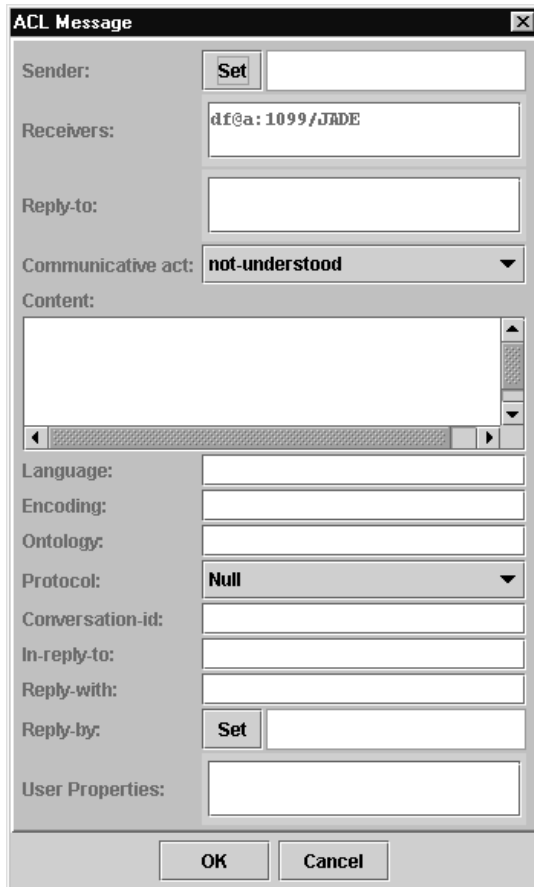
The image shows a Java Swing dialog box titled "ACL Message". It contains several input fields and buttons. The "Sender:" field has a "Set" button next to it. The "Receivers:" field contains the text "df@a:1099/JADE". The "Reply-to:" field is empty. The "Communicative act:" field is a dropdown menu with "not-understood" selected. The "Content:" field is a large text area. Below it are fields for "Language:", "Encoding:", "Ontology:", and "Protocol:" (which has a dropdown menu with "Null" selected). There are also fields for "Conversation-id:", "In-reply-to:", "Reply-with:", and "Reply-by:" (which has a "Set" button next to it). At the bottom is a "User Properties:" field. The dialog box has "OK" and "Cancel" buttons at the bottom right.

Figura 7 - Finestra per enviar missatge

- Executar l'agent sniffer: és una aplicació JAVA creada per rastrejar l'intercanvi de missatges entre els agents JADE. En activar-lo, s'obre una finestra on podem observar l'evolució temporal del pas de missatges. Si seleccionem alguna de les fletxes que simbolitzen aquest intercanvi, se'ns mostrarà el missatge amb tots els seus paràmetres. A més podrem seleccionar quins agents volem monitoritzar i guardar els missatges a disc.

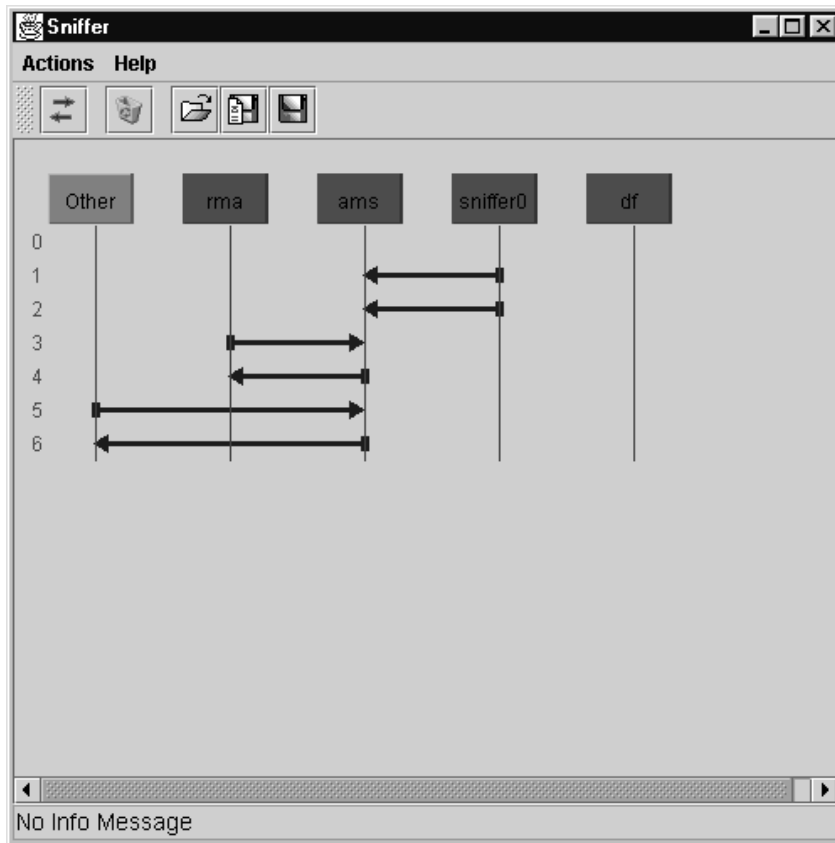


Figura 8 - Finestra de l'sniffer

- Obrir el Dummy-Agent: aquest permet enviar missatges ACL entre els agents, rebre'ls i inspeccionar-los i llegir i salvar-los des d'un fitxer.

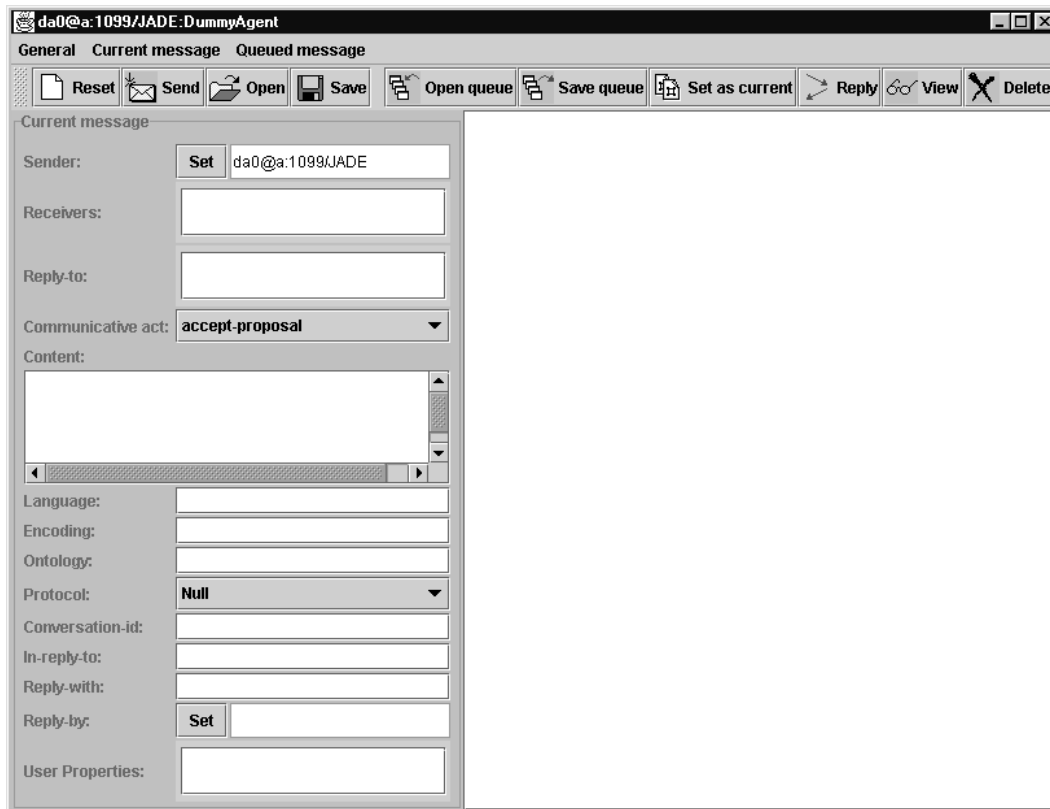


Figura 9 - Finestra de l'agent Dummy

- A més, també disposem de la possibilitat d'activar la interfície de l'agent DF (des del menú *tools*), on podrem observar els agents que s'han registrat i quin serveis ofereixen.

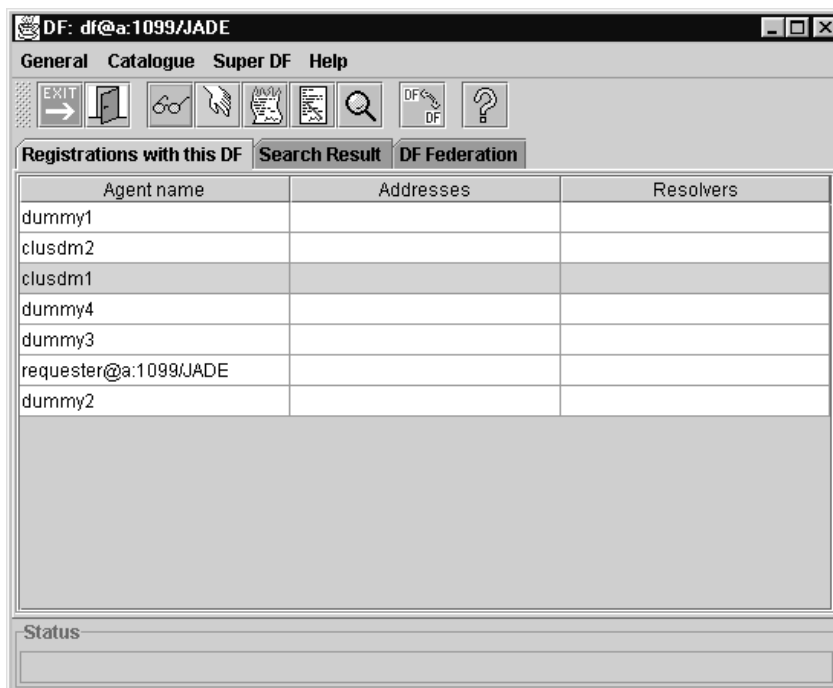


Figura 10 - Finestra amb la interfície del DF

4. JUSTIFICACIÓ DEL SMA

Del que s'ha explicat anteriorment sobre agents, es pot deduir la seva utilitat quan es treballa sobre entorns distribuïts o es requereix minimitzar el temps de càlcul mitjançant l'execució paral·lela.

Així, el sistema Multi-Agent que s'intentarà aconseguir es basarà en l'estructura del model Espanyol [13] de trasplantaments: el territori nacional està dividit en 6 zones; cada zona agrupa alguna o algunes de les 19 comunitats autònomes. En cada hospital, a més, existeix una persona encarregada de coordinar el procés d'assignació d'òrgans a pacients del seu centre. L'objectiu serà introduir un agent Coordinador a cada hospital que disposi de tota la informació i que realitzi la cerca del receptor seguint el model explicat anteriorment (primer dins l'hospital, després dins la seva zona i finalment a altres zones). Per aquest motiu, resulta molt important que els agents encarregats d'aquesta feina puguin prendre decisions i comunicar-se eficientment amb els seus homòlegs a altres centres mitjançant la xarxa.

Durant el procés d'assignació serà necessari trobar el receptor o receptors més adients per a l'òrgan, però aquest procés no és senzill, doncs cal tenir en compte molt criteris. És per això que per a l'avaluació de la llista de possibles receptors utilitzarem tècniques d'ajuda a la presa de decisions multicriteri (MCDM).

Donat que no existeix un únic MCDM que resolgui el problema, s'implementaran un conjunt d'agents que treballin en paral·lel. Cadascun d'aquest utilitzarà un mètode específic per tal d'ordenar de llista de pacients. Els resultats finals es passaran a l'agent Coordinador que disposarà de diferents punts de vista sobre les mateixes dades, facilitant l'elecció del millor receptor.

A més d'aquest agents MCDM, existirà un altre anomenat Especialista en Trasplantaments que disposarà de coneixement sobre algunes regles que han de complir els pacients per tal de rebre l'òrgan (ex: tipus de sang compatible). Aquest agent filtrarà la llista de pacients (eliminat aquells que no compleixin les condicions) abans d'enviar les dades al agents MCDM. A més, a partir dels resultats que s'obtinguin, l'Especialista rebrà una valoració sobre la seva bondat, de manera a mesura que es facin més proves, podrà detectar quins mètodes funcionen més o menys bé en aquest camp.

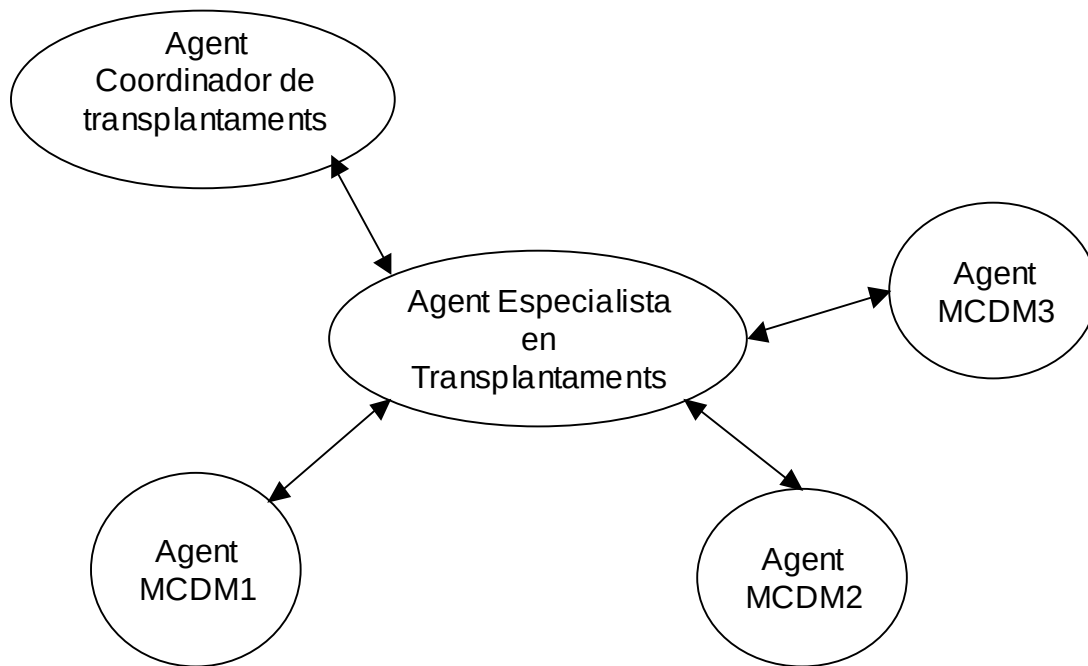


Figura 11 - Sistema Multi-Agent per a l'avaluació de pacients

Dels objectius definits anteriorment, podem observar com un Sistema Multi-Agent resulta molt convenient, ja que ens proporciona les següents avantatges:

- Modularitat: el problema es pot estructurar a diferents nivells tal com queda representat en l'esquema anterior, de forma que es pot dividir la feina i treballar les parts individualment.
- Eficiència: en aquest cas, ens permet que els diferents agents MCDM puguin executar-se paral·lelament (si es troben en diferents màquines) o concurrentment (a la mateixa), millorant el temps de càlcul global.
- Fiabilitat: gràcies a la incorporació de diversos agents MCDM, ens assegurem de que tot i que falli algun d'ells (per motius software o hardware), sempre tindrem altres alternatives.
- Flexibilitat: es poden afegir i eliminar fàcilment agents MCDM del sistema i donar preferència a certs mètodes (segons quins resultats obtinguin), gràcies a la capacitat d'adaptació i aprenentatge.

5. ARQUITECTURES

En aquest apartat es passaran a descriure les parts en les que es divideix el sistema implementat indicant quin és el seu propòsit i funcionament i quina estructura tenen.

Com ja s'ha dit, l'objectiu és simular una part d'un Sistema Multi-Agent que gestioni el trasplantament d'òrgans. Concretament ens centrarem en l'elecció dels receptors més adequats per al donant.

En aquest projecte, s'implementarà l'agent Especialista en Trasplantaments que es comportarà tal com s'ha descrit en l'apartat anterior tot i que les dades les introduirà l'usuari o a través d'un fitxer (doncs encara no existeix l'agent Coordinador). Pel que fa als agents MCDM, s'implementarà un que utilitzi un mètode basat en Funcions de Negació i *Clustering*, però permetent que s'afegeixin nous mètodes fàcilment (en concret se suporten mètodes que utilitzen semàntiques basades en *Negation Functions* [4], *Fuzzy sets* [5], *2-tuple* i *Ordinal*).

5.1 Estructura del sistema

L'estructura que segueix el sistema implementat és la següent:

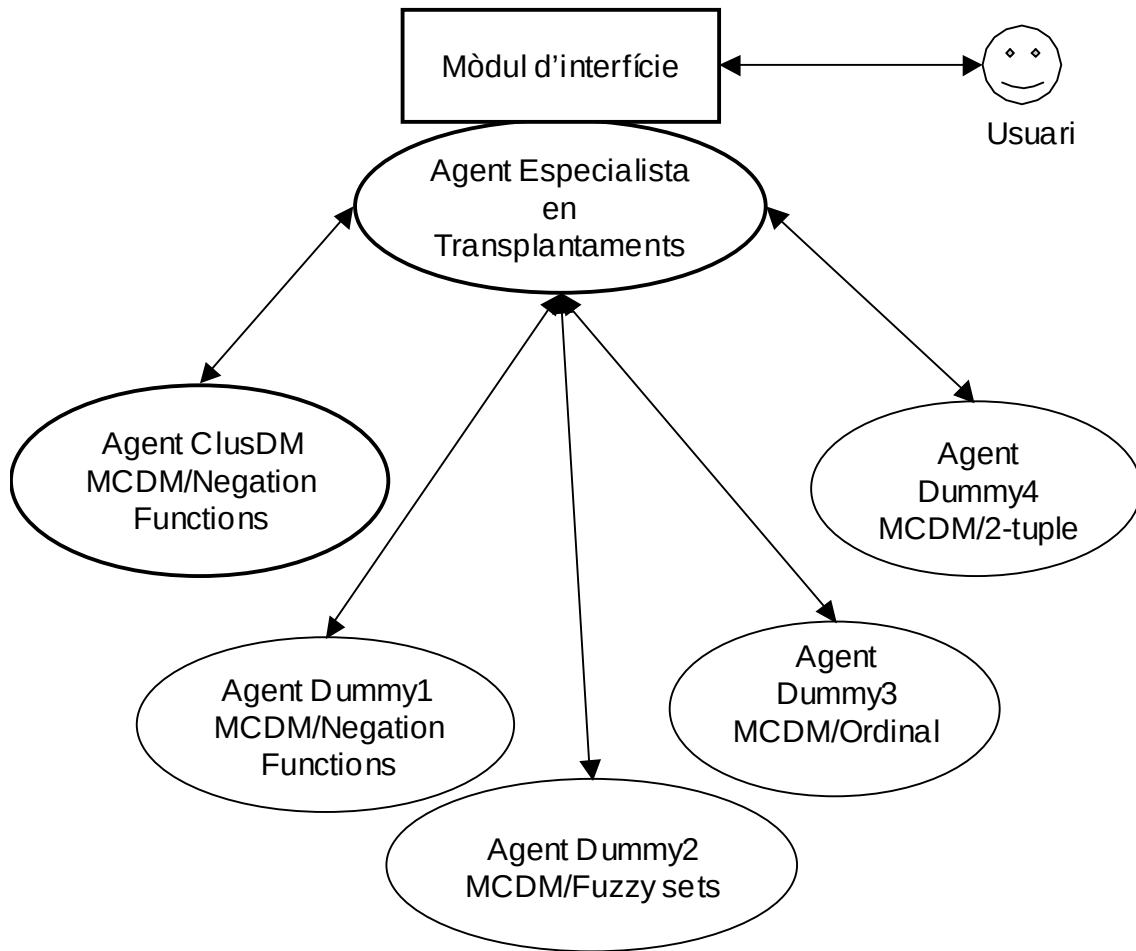


Figura 12 - Estructura del Sistema Multi-Agent implementat

Tal com s'observa a la figura, en l'estructura definida existeixen dos nivells principals:

- L'Agent Especialista: es podria considerar com un sistema expert en trasplantaments. La seva funció principal és la de realitzar un filtratge sobre la llista de receptors aplicant una sèrie de regles que s'han de complir per poder fer els trasplantament. A més, per cadascun dels atributs que es consideren, haurà de transformar les dades a criteris d'adequació amb el donant utilitzant valors qualitius. El conjunt d'atributs que s'avaluaran per a cada pacient són els següents:
 - o Distància
 - o Adequació del pes de l'individu
 - o Adequació del tamany de l'òrgan.
 - o Nombre d'antígens diferents
 - o Edat
 - o Temps d'espera

Un cop fet això, es comunicarà amb els diferents agents MCDM que existeixin al sistema, enviant-los les dades que ha obtingut i una mesura d'avaluació dels vocabularis segons les semàntiques utilitzades (negacions, conjunts difusos...). Originalment les dades les hauria de rebre d'un agent Coordinador de Trasplantaments que serà qui interactui amb l'usuari; a més també l'enviaria els resultats que li haguessin retornat els diversos agents MCDM amb que s'hagués comunicat. En funció d'aquests valors, el Coordinador rebria valoracions dels diferents mètodes i les enviaria a l'Especialista el qual els utilitzaria per extreure conclusions sobre el comportament de determinats agents (desestimació dels mètodes més dolents i recomanació dels més bons). En aquest projecte però, s'ha optat per "fusionar" aquests dos agents, de manera que l'Especialista disposarà d'una interfície amb l'usuari a través de la qual rebrà les dades inicials i les valoracions i des d'on es mostraran els resultats. D'aquesta forma es redueix la complexitat d'implementació d'un altre agent però es manté tota la funcionalitat del sistema complet.

- Agents MCDM (Métodes de decisió Multicriteri): el seu objectiu és, partint d'una llista de receptors i dels seus valors d'adequació al donant en funció de diversos atributs, proporcionar una llista ordenada segons les probabilitats d'èxit del trasplantament. Els algoritmes d'ajuda a la presa de decisions multicriteri no pretenen substituir a la persona en el procés d'elecció del donant, sinó facilitar-li la feina. D'aquesta forma, la major part de les vegades es proposaran diverses alternatives que l'individu haurà de valorar. A més, la forma d'expressar la viabilitat del trasplantament serà també utilitzant termes qualitius que simplifiquin la compressió dels resultats. Donant que bona part del procés es realitza mitjançant termes no numèrics, es necessitaran mesures d'avaluació i comparació d'aquests valors. Segons quins mètodes utilitzin els diferents agents, parlarem d'una o una altra semàntica. En aquest projecte s'ha implementat un algoritme d'ordenació basat en *Clustering* i funcions de negació associat a un agent, però també s'han definit la resta d'agents MCDM tot i que no tinguin cap algoritme associat (i per tant no tenen funcionalitat). Els diferents tipus d'agents MCDM que s'han implementat són:

- o MCDM/*Negation functions*
- o MCDM/*2-tuple*
- o MCDM/*fuzzy-sets*
- o MCDM/*ordinal*

D'aquesta forma, afegir nous algoritmes serà tant senzill com implementar-los i associar-los a l'agent del seu tipus.

5.2. Protocol de comunicació

Durant el procés de comunicació, és molt important el protocol utilitzat ja que aquest defineix quin comportament tindran els agents durant l'acte comunicatiu. En el nostre cas, l'objectiu principal és demanar la realització d'accions, enviar dades i retornar resultats. D'entre tots els protocols que es defineixen a la FIPA i permet implementar JADE, el que més s'adequa a aquest objectiu és el **FIPA-Request**. Les seves característiques ja s'han definit en la secció dedicada al JADE però adaptant-lo al nostre problema, l'evolució temporal de la conversa seguiria el següent esquema:

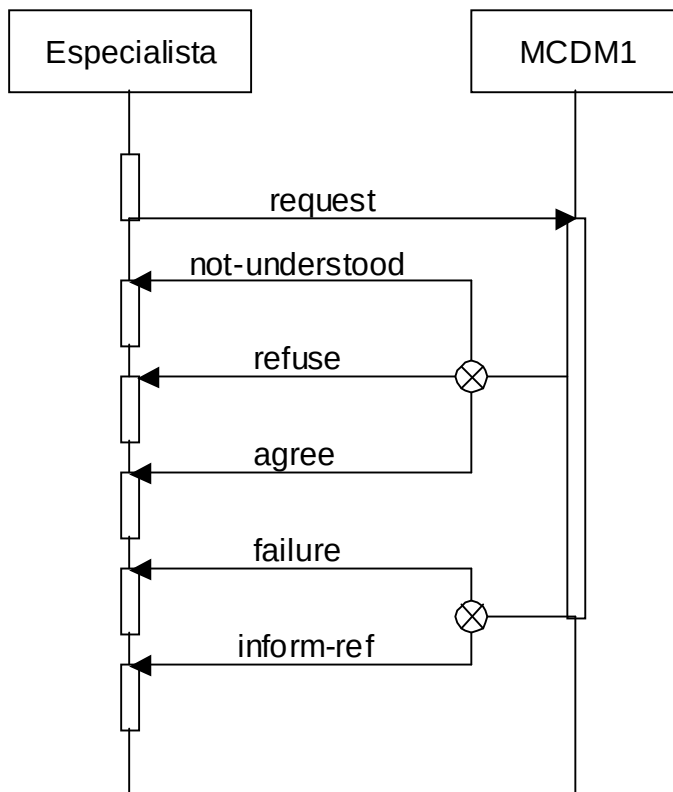


Figura 13 - Comunicació entre agents segons el protocol FIPA-Request

5.3 Missatges

Els diferents missatges que involucrem la conversa entre els agents són els següents:

- **Request:** a través d'aquest, l'especialista demana a un agent MCDM determinat que realitzi una acció (només pot demanar l'ordenació de la llista de pacients). A més també enviarà totes les dades que es necessitin en el procés (llista de pacients amb els seus atributs, informació requerida per la semàntica: funcions de negació, conjunts difusos...).
- **Not-understood:** indica que l'agent receptor del missatge no ha estat capaç d'extreure el contingut del missatge correctament i, per tant, no es poden interpretar les dades. Aquesta situació no es produirà mai, a menys que l'Especialista hagi compost malament el missatge.
- **Refuse:** a partir de les dades que ha rebut, l'agent MCDM pot avaluar-les i decidir si farà o no l'acció. En aquest cas, si alguna de les dades que s'han enviat no és correcta s'enviarà aquest missatge (concretament si els pesos dels atributs

en percentatges no sumen 100 o si se superen els rangs màxims definits en quant a nombre de pacients i d'atributs).

- **Agree:** indica que s'ha acceptat l'acció i que es comença a realitzar. Quan s'hagi completat el procés s'enviarà un nou missatge.
- **Failure:** en el cas que durant el procés de realització de l'acció es produeixi algun error, s'enviarà aquest missatge, indicant que no s'ha pogut arribar a obtenir un resultat. En el nostre cas, si durant el procés de *Clustering* que utilitza l'agent MCDM implementat no s'aconsegueix un nombre de classes finals vàlid (adequat al nombre de termes del vocabulari final), s'enviarà aquest missatge.
- **Inform-ref:** quan finalment s'ha completat l'acció correctament, s'enviarà aquest missatge en el qual s'adjuntarà el resultat obtingut. En el nostre cas es retornarà una llista de receptors ordenada i agrupada segons termes que indiquin la viabilitat del trasplantament.

5.3. Ontologia

Com es pot veure, l'intercanvi d'informació durant el procés de la conversa és molt abundant i la correcta interpretació de les dades resulta fonamental per aconseguir un resultat. Per aquest motiu, resulta necessari definir una sèrie de regles i estructures que hauran de seguir les dades de forma que totes dues bandes de la conversa siguin capaces d'entendre el significat.

És en aquest punt on es defineix l'Ontologia que s'utilitzarà per composar els continguts dels missatges. Les seves característiques ja s'han explicat en l'apartat d'Agents, però bàsicament es basa en la definició d'objectes o *frames* que estan formats per *slots* que poden ser tipus simples o altres *frames*.

En el nostre cas, s'enviaran dades durant el procés de **request** i en el moment de retornar un resultat. La informació que s'intercanviarà en cada cas serà la següent:

- **Request:**
 - o Nombre d'atributs
 - o Llista d'atributs:
 - Pes de l'atribut
 - Vocabulari de l'atribut:
 - Terme del vocabulari
 - Llista de negacions per a cada terme:
 - o Valor de cada negació
 - Valors difusos per a cada terme
 - o Funció de distància escollida
 - o Nombre de pacients a avaluar
 - o Llista de pacients:
 - Identificador
 - Llista d'atributs per a cada pacient:

- Valor de cada atribut

- **Inform:**

- o Nombre de pacients
- o Llista de pacients ordenada:
 - Identificador
 - Terme qualitatiu del vocabulari final assignat

A més, caldrà definir una sèrie d'objectes que indiquin situacions d'error que es puguin produir:

- Pesos no vàlids.
- Nombre de pacients màxim sobrepassat
- Nombre d'atributs màxim sobrepassat.
- Nombre de classes finals insuficients.

Com es pot veure, les dades segueixen una estructura que les fa ideals per a definir-les a l'ontologia. L'esquema general utilitzat amb notació definida per la FIPA és el següent [12]:

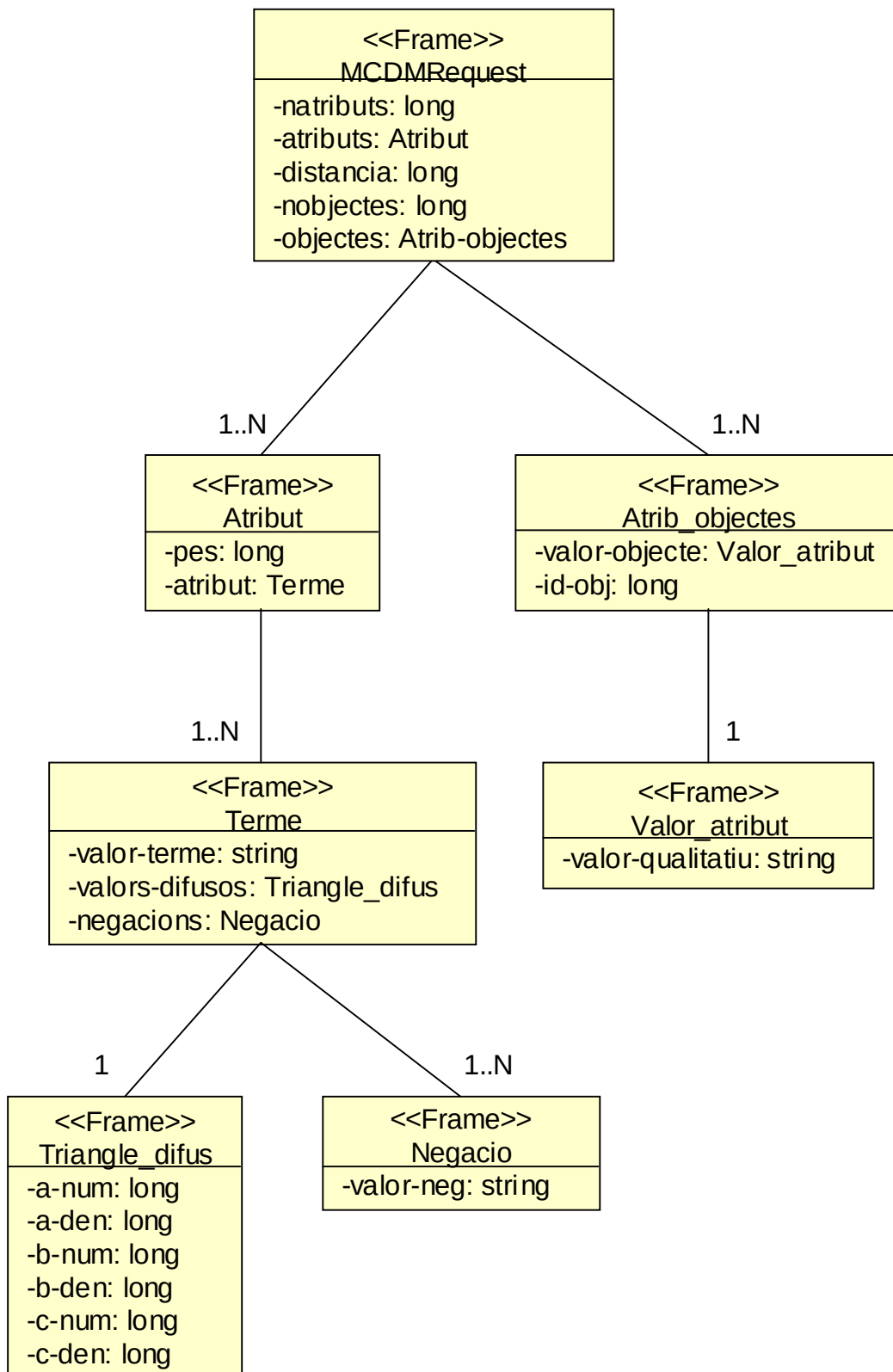


Figura 14 - MCDM Ontology

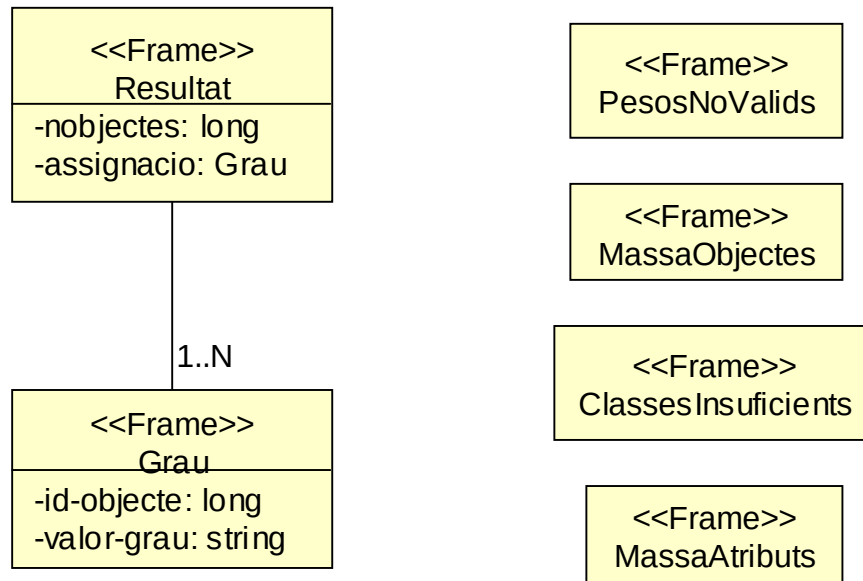


Figura 15 - MCDM Ontology (continuació)

6. ESPECIALISTA EN TRASPLANTAMENTS

Abans de fer les crides als agents MCDM (Mètodes de decisió multi-criteri) per a que ordenin la llista de pacients segons la seva adequació al donant, existeix un pas intermig on l'agent Especialista en Trasplantaments s'encarregarà de realitzar un filtratge sobre la llista de pacients i preparar les dades.

6.1. Dades d'entrada

Així, les dades que rebrà l'agent especialista (llegides des d'un fitxer d'entrada), seran les següents:

- Llista d'objectes pacient (definit a una classe Java). Cadascun d'ells haurà de proporcionar la següent informació:
 - o Hospital on s'està el pacient: es una dada de tipus *String*.
 - o Localitat on està situat l'hospital: dada de tipus *String*.
 - o Província on està la localitat: dada de tipus *String* que ens permetrà situar al pacient dins d'una zona del territori Espanyol.
 - o Tipus de sang: es un enumerat que pot prendre els següents valors: A, B, AB, O.
 - o Pes de la persona: valor decimal de doble precisió sense acotar.
 - o Tamany de l'òrgan del pacient: valor decimal de doble precisió sense acotar.
 - o Antigenes: cada individu disposa d'antigenes B i DR (2 en cada cas). Els valors seran quatre booleans segons els tingui o no.
 - o Data de naixement: es proporcionarà el dia, mes i any del naixement del pacient.
 - o Data d'entrada a la llista d'espera de l'hospital: ídem a l'anterior.
 - o Com a identificador de pacient (amb el qual se l'inclourà a la llista resultat) es prendrà l'ordre d'introducció al fitxer de pacients.
- Objecte donant amb el que comparar les alternatives. Contindrà totes les dades anteriors tret de la data d'entrada a la llista d'espera i l'identificador.
- Tipus d'òrgan del donant: se suposarà que tots els pacients de la llista necessiten aquest òrgan; es consideren els següent tipus: cor, pulmó, fetge i ronyó.

6.2. Restriccions sobre els receptors

El primer que es farà amb aquestes dades és aplicar uns filtres que desestimïn aquells pacients que, per determinades raons no se'ls pot trasplantar l'òrgan del donant.

Els filtres considerats en aquest cas són els següents:

- En quant al tipus de sang, cal considerar la següent taula de compatibilitat entre la sang del pacient i del donant. Si el receptor no té algun dels tipus vàlids en relació al donant, es desestimarà.

Donant	Receptor
O	O, A, B, AB
A	A, AB
B	B, AB
AB	AB

Taula 1 - Taula ABO

- Pes de la persona: cal que la diferència entre el pes del donant i del receptor no sigui massa gran, doncs els trasplantament no seria possible. En aquest cas, es calcula el rati de pesos (pes donant /pes pacient); si el resultat és inferior a 0.75 (pacient massa gras) o superior a 1.25 (pacient massa prim), es desestimarà aquest individu.
- Tamany de l'òrgan: en aquest cas s'actua de la mateixa forma que en l'anterior: el rati entre els tamanyes de receptor i donant ha d'estar entre 0.75 i 1.25, sinó es desestimarà el pacient.
- Data de naixement: la diferència d'edats entre el donant i el receptor no pot ser massa elevada, concretament, no ha de superar els 25 anys.
- Tipus d'òrgan: segons quin sigui, hi ha una limitació en l'edat del pacient que pot rebre'l:
 - o Cor: fins a 60 anys
 - o Pulmó: fins a 70 anys
 - o Fetge i Ronyó: fins a 80 anys.

Un cop fet això, ja haurem eliminat aquells pacients que violen aquestes restriccions de la llista de receptors. Però encara serà necessari transformar les dades per a que els agents d'ordenació puguin treballar-hi.

6.3. Obtenció de la matriu de preferències

Cada algoritme MCDM (Mètode de decisió multicriteri), no rebrà directament aquestes dades sinó que necessitarà una matriu de preferències. Aquesta estructura consisteix en una taula de dos dimensions, on les files es corresponen a les diferents alternatives de la decisió (pacients) i les columnes són els criteris que ens permeten expressar les preferències sobre aquestes alternatives segons diferents aspectes (propietats, característiques, punts de vista...).

	Pes	Alçada	Distància
Pacient X	Adequat	Alta	Província
Pacient Y	Baix	Factible	Ciutat
Pacient Z	Alt	Factible	província

Taula 2 - Exemple de matriu de preferències

Els valors que prenen els criteris s'anomenen Termes, i el conjunt d'aquests és el seu Vocabulari. Aquests valors estan ordenat de menys a més preferit, en relació a la decisió que s'ha de prendre. En el nostre cas, el valor òptim és el que hauria de tenir el pacient per tal de rebre l'òrgan amb èxit.

Així, per cadascun dels atributs considerats (tret dels que són excloents), es buscarà la relació entre cada pacient i el donant, expressant-la en termes qualitatiu (bo, fluix, factible...), tal com intentaria fer un Especialista humà. Per fer-ho, es defineix un vocabulari amb més o menys termes per a cada atribut que expressa el grau d'adequació al model. Les transformacions dels atributs a criteris de preferència són les següents:

6.3.1. L'atribut Distància

Es defineix el següent vocabulari (de menys a més adequació al model):

{ país, zona, província, ciutat, hospital }

Sota aquest criteri s'avaluaran les dades que ens informen sobre la localització geogràfica: hospital, localitat i província. L'assignació de valors dins del domini definit és la següent:

- Si observem que el pacient i el donant estan al mateix hospital, s'assignarà al receptor el terme "hospital".
- Si estan en la mateixa localitat, s'assignarà el terme "ciutat".
- Si estan a la mateixa província, utilitzarem el terme "província".
- Si no és cap d'aquest casos, l'agent avaluarà si les dues províncies (del donat i del pacient) es troben dins de la mateixa zona (veure figura), aplicant el terme "zona".
- Si no es compleix cap de les situacions anteriors, s'aplicarà el terme "país".

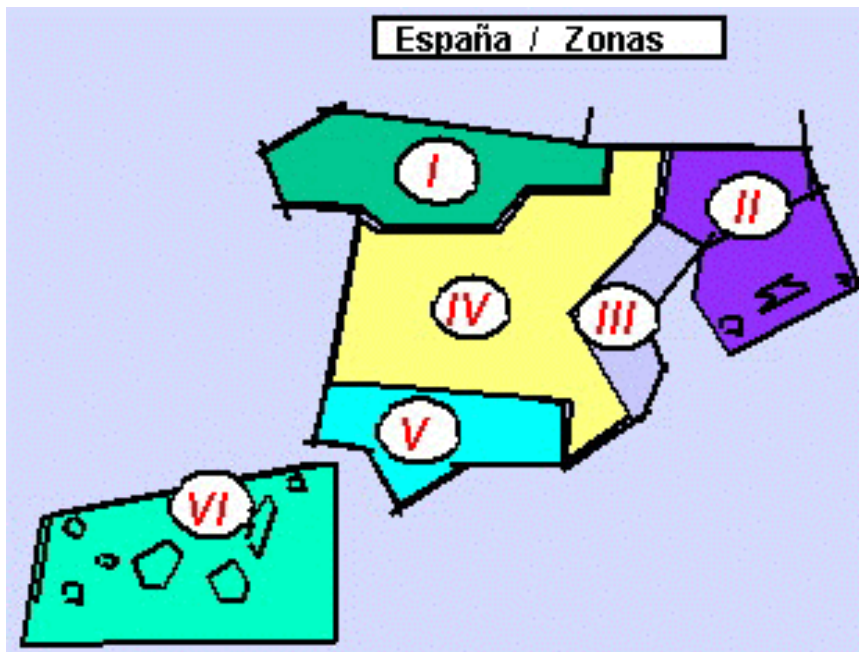


Figura 16 - Distribució de les províncies Espanyoles en zones

Detall de la distribució geogràfica per zones de les províncies del territori Espanyol segons la ONT (Organización Nacional de Trasplantes) [13]:

- Zona I: La Coruña, Pontevedra, Orense, Lugo, Asturias, Cantabria, Vizcaya, Guipuzcua, Álava, Navarra, La Rioja, León, Palencia, Burgos, Soria, Zamora, Valladolid, Segovia, Ávila, Salamanca
- Zona II: Lleida, Girona, Barcelona, Tarragona, Baleares
- Zona III: Castellón, Valencia, Alicante
- Zona IV: Huesca, Zaragoza, Teruel, Madrid, Guadalajara, Cuenca, Toledo, Ciudad Real, Albacete, Cáceres, Badajoz, Murcia
- Zona V: Huelva, Sevilla, Córdoba, Jaén, Granada, Almeria, Málaga, Cádiz
- Zona VI: Santa Cruz de Tenerife, Las Palmas

6.3.2. L'atribut Pes

El vocabulari definit en aquest cas (de pitjor a millor) és el següent:

{ fluixa, factible, bona, ideal }

Tal com s'ha explicat en l'apartat de filtres, es calcularà el rati entre el pes del donant i del pacient ($\text{pes_donant} / \text{pes_pacient}$). El resultat ha d'estar entre 0.75 i 1.25, assignat-se valors del domini segons la següent escala:

$0.208 \leq |\text{rati} - 1| < 0.25 \Rightarrow \text{fluixa}$
 $0.166 \leq |\text{rati} - 1| < 0.208 \Rightarrow \text{factible}$
 $0.042 \leq |\text{rati} - 1| < 0.166 \Rightarrow \text{bona}$
 $0.0 \leq |\text{rati} - 1| < 0.042 \Rightarrow \text{ideal}$

6.3.3. L'atribut Tamany de l'òrgan

Vocabulari definit per aquest criteri:

{ fluixa, factible, bona, ideal }

Al igual que en el cas anterior, es calcularà el rati entre el pes de l'òrgan de pacient i del donant, assignant valors del domini segons la següent escala.

$0.208 \leq |\text{rati} - 1| < 0.25 \Rightarrow \text{fluixa}$
 $0.166 \leq |\text{rati} - 1| < 0.208 \Rightarrow \text{factible}$
 $0.042 \leq |\text{rati} - 1| < 0.166 \Rightarrow \text{bona}$
 $0.0 \leq |\text{rati} - 1| < 0.042 \Rightarrow \text{ideal}$

6.3.4. L'atribut Nombre d'antígens diferents

Vocabulari definit per a aquest criteri:

{ dolent, adequat, correcte }

Ja s'ha explicat que cada individu té 4 antígens: 2 de tipus B i 2 de tipus DR amb valors booleans. En aquest cas, es tractarà de comparar dos a dos cada antigen del pacient i del donant i comptar el nombre de diferències. Segons aquestes, s'assignarà un o altre terme seguin la següent escala

Si no hi ha cap diferència ni en B ni en DR $\Rightarrow$ correcte
Si hi ha 1 diferència en total (en B o en DR) $\Rightarrow$ adequat
Si hi ha més d'1 diferència en total $\Rightarrow$ dolent

6.3.5. L'atribut Edat

Vocabulari:

{ mes_de_20, mes_de_17, mes_de_14, mes_de_11, mes_de_8, mes_de_5, iguals }

En aquest cas es calcularà la diferència d'edat en valor absolut entre el donant i el pacient (a partir de les dates de naixement) i se seguirà la següent escala:

$0 \leq \text{diferencia} < 6 \Rightarrow \text{iguals}$
 $6 \leq \text{diferencia} < 9 \Rightarrow \text{mes_de_5}$

9	$\leq$ diferencia < 12	=> mes_de_8
12	$\leq$ diferencia < 15	=> mes_de_11
15	$\leq$ diferència < 18	=> mes_de_14
18	$\leq$ diferència < 21	=> mes_de_17
21	$\leq$ diferència < 26	=> mes_de_20

6.3.6. L'atribut Temps d'espera

Vocabulari:

{ recent, normal, moderat, llarg, molt_llarg }

Segons la data d'introducció del pacient a la llista d'espera i la data actual, s'assignarà un terme del vocabulari segons el nombre de mesos que porti esperant:

	temps $\geq$	18 mesos	=> molt_llarg
18	> temps $\geq$	10 mesos	=> llarg
12	> temps $\geq$	6 mesos	=> normal
6	> temps $\geq$	1 mes	=> moderat
1	> temps $\geq$	0 mesos	=> recent

6.4. Mesures d'avaluació de termes qualitatius

La matriu de preferències que obtenim després de fer aquest procés, serà la que rebran tots els agent MCDM disponibles com a dades d'entrada. Tot i això, encara serà necessari definir algun tipus de mesura que sigui capaç de treballar amb termes qualitatius permetent fer comparacions. Aquesta mesura variarà segons la semàntica que utilitzi el mètode. En el nostre cas, utilitzarem dues semàntiques principals: funcions de negació i conjunts difusos.

6.4.1. Funcions de negació

Aquest mètode consisteix en definir una sèrie de funcions sobre els termes que formen el vocabulari d'un criteri, expressant així la relació existent entre els valors i permetent construir un domini dividit en intervals on, cadascun d'ells es refereix al rang de validesa d'un dels termes. Així, si considerem per exemple el cas de la distància i definim les etiquetes: *país*, *zona*, *província*, *ciutat* i *hospital*, podem observar com el pes del significat de cadascuna d'elles és diferent (definir una distància tipus "país" dona una informació molt més general que una tipus "hospital" i, per tant el rang de validesa dins del domini haurà de ser major que la segona). Per poder saber com estarà dividit el domini del criteri i la densitat dels termes lingüístics, s'utilitzaran les funcions de negació [4]. Aquesta es una forma d'expressar quin o quins són els termes oposats a un valor determinat. El fet de que una etiqueta tingui diverses negacions ens informará que el seu 'pes' informatiu és més gran que les altres.

Aquestes relacions ens permetran construir un domini on les etiquetes estiguin ordenades de menor a major conveniència. Si acotem aquest domini entre 0 i 1, els extrems de cada interval podran ser expressats com un valor numèric entre aquest rang (segons el nombre d'etiquetes i la seva densitat). D'aquesta forma, podrem trobar quin és el valor numèric que equival al punt mig de cada interval i que serà el que utilitzarem en lloc del terme lingüístic en fer els càlculs (doncs expressa el punt del domini –entre 0 i 1- on l'etiqueta té la màxima significació). La forma de calcular els extrems dels intervals per a cada terme es fa utilitzant la següent fórmula:

$$I(x_i) = [m_i, M_i] = \left[\frac{\sum_{x < x_i} |Neg(x)|}{\sum_{x \in X} |Neg(x)|}, \frac{\sum_{x \leq x_i} |Neg(x)|}{\sum_{x \in X} |Neg(x)|} \right]$$

on:

$I(x_i)$ és l'interval per al terme i del criteri x

m_i és la cota mínima de l'interval i

M_i és la cota màxima de l'interval i

Seguint aquesta metodologia, l'Especialista en Trasplantaments definirà una sèrie de funcions de negació per a cada criteri i passarà aquesta informació a l'agent MCDM que utilitzi aquesta semàntica. Aquest últim serà l'encarregat de fer la transformació de qualitatiu a numèric seguint el procés descrit anteriorment.

6.4.2. Conjunts difusos

En aquest cas, tractem amb algorismes que treballen amb valors difusos [5]. De la mateixa forma que en l'apartat anterior, partirem de les funcions de negació per poder definir els intervals del domini que avarca cada terme del vocabulari. A partir d'aquests podrem calcular quins són els extrems i el punt màxim per a cada etiqueta que anem obtenint, per tant, aconseguirem una tripleta de valors (mínim, màxim, mig).

Al contrari que en el cas anterior, els càlculs ja estaran fet abans i l'Especialista passarà a l'agent MCDM els tres valors que defineixen cada conjunt difús.

6.4.3. Obtenció de mesures per cada atribut

A continuació es descriuen les funcions de negació que s'han definit per a cadascun dels criteris, així com la distribució dels intervals de negació per als termes dins del domini 0-1 i el valor numèric que correspondrà a cada terme del vocabulari. També s'inclouen els conjunts difusos obtinguts aplicant la mateixa metodologia, definint així tripletes de valors (vértexs dels triangles que formen els extrems dels intervals), que seran els que rebrà l'agent.

6.4.3.1. Mesures per a la Distancia

Vocabulari

{ país, zona, província, ciutat, hospital }

Funcions de negació

Neg(país)=província, ciutat, hospital

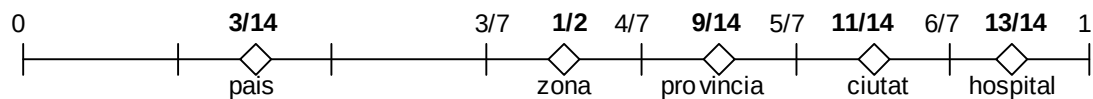
Neg(zona)=zona

Neg(província)=país

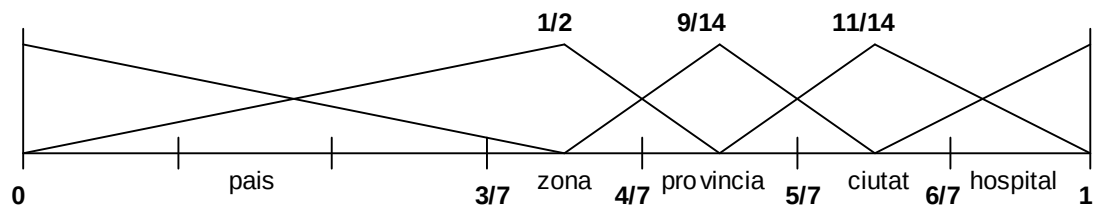
Neg(ciutat)=país

Neg(hospital)=país

Intervals de negació



Conjunts difusos



6.4.3.2. Mesures per al Pes

Vocabulari:

{ fluixa, factible, bona, ideal }

Funcions de negació

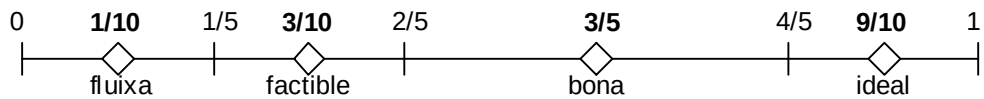
Neg(ideal)=fluixa

Neg(bona)=bona, factible

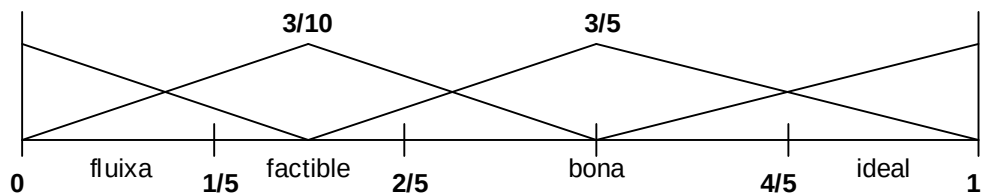
Neg(factible)=bona

Neg(fluixa)=ideal

Intervals de negació



Conjunts difusos



6.4.3.3. Mesures per al Tamany de l'òrgan

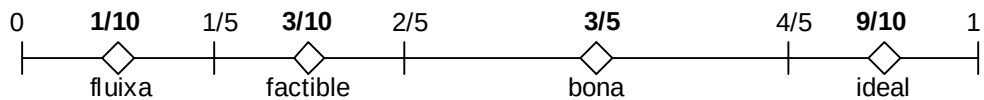
Vocabulari:

{ fluixa, factible, bona, ideal }

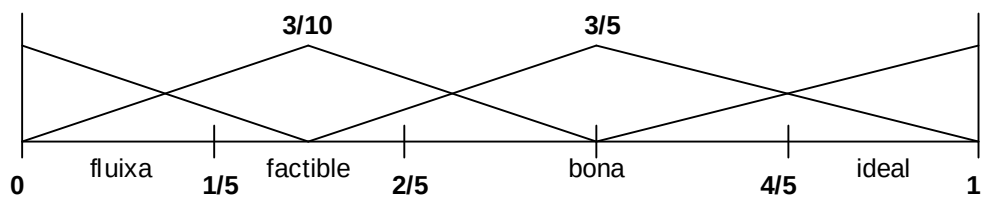
Funcions de negació

Neg(ideal)=fluixa
 Neg(bona)=bona, factible
 Neg(factible)=bona
 Neg(fluixa)=ideal

Intervals de negació



Conjunts difusos



6.4.3.4. Mesures per al Nombre d'antigens

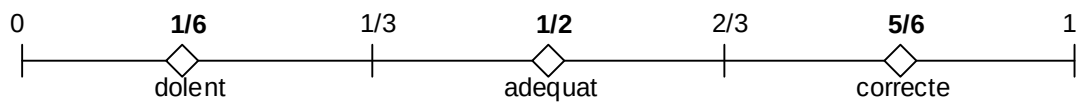
Vocabulari:

{ dolent, adequat, correcte }

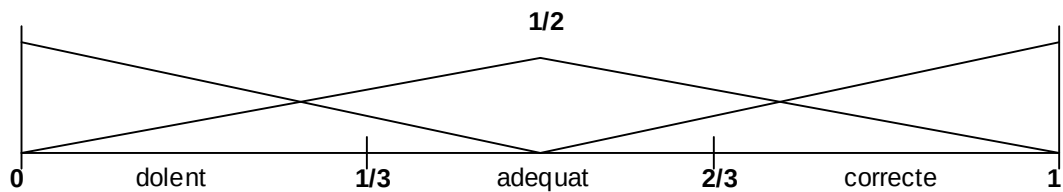
Funcions de negació

Neg(dolent)=correcte
Neg(adequat)=adequat
Neg(correcte)=dolent

Intervals de negació



Conjunts difusos



6.4.3.5. Mesures per a l'Edat

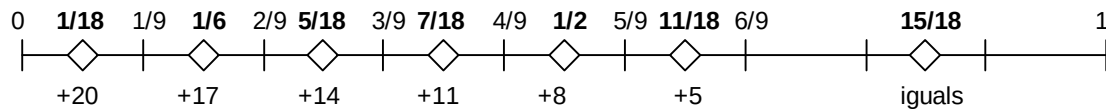
Vocabulari:

{ mes_de_20, mes_de_17, mes_de_14, mes_de_11, mes_de_8, mes_de_5, iguals }

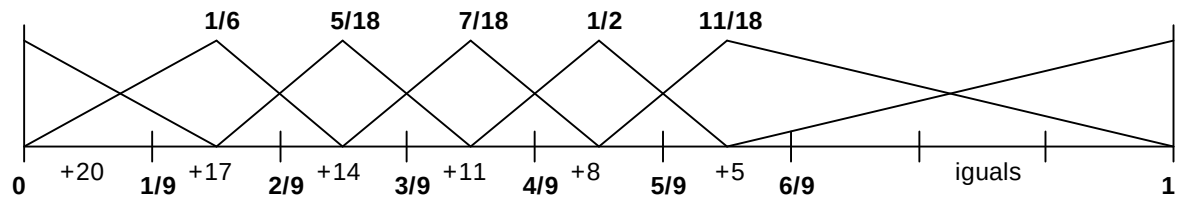
Funcions de negació

Neg(mes_de_20)=iguals
Neg(mes_de_17)=iguals
Neg(mes_de_14)=iguals
Neg(mes_de_11)=mes_de_5
Neg(mes_de_8)=mes_de_8
Neg(mes_de_5)=mes_de_11
Neg(iguals)=mes_de_14, mes_de_17, mes_de_20

Intervals de negació



Conjunts difusos



6.4.3.6. Mesures per al Temps d'espera

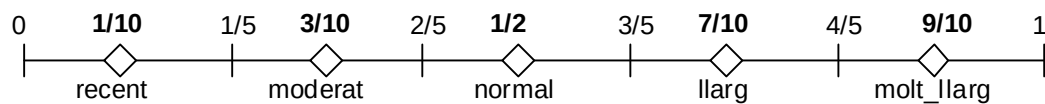
Vocabulari:

{ molt_llarg, llarg, normal, moderat, recent }

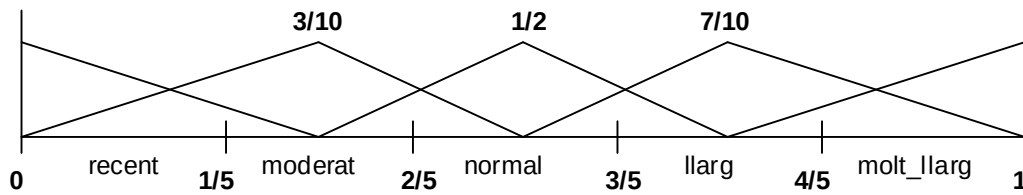
Funcions de negació

Neg(molt_llarg)=recent
 Neg(llarg)=moderat
 Neg(normal)=normal
 Neg(moderat)=llarg
 Neg(recent)=molt_llarg

Intervals de negació



Conjunts difusos



6.5. Avaluació dels resultats

A més de tot això, l'Especialista rebrà valoracions sobre el resultat que ha retornat l'agent MCDM amb qui s'ha comunicat, de forma que podrà crear-se una opinió sobre el funcionament de cada algoritme. Per fer això, es disposarà d'un vocabulari amb el que puntuar el resultat:

{ molt_bo, bo, acceptable, regular, dolent }

En funció del nombre de vegades que s'utilitzi un a altre terme, l'agent elaborarà una opinió seguint la següent escala:

- Si més de la meitat de les valoracions són:
 - o “dolent” : el mètode no es recomana.
 - o “regular” : el mètode no dona resultats satisfactoris.
 - o “bo” : el mètode és acceptable però s'han d'estudiar amb cura els resultats.
 - o “molt_bo” : es recomana el mètode.
- Si més d'un 80 % de les valoracions són “molt_bo” o “bo” : el mètode es recomana especialment.

7. Algoritme d'ordenació d'alternatives

Com ja s'ha dit, el nucli principal de tot el sistema serà el procés d'ordenació dels diversos receptors d'un òrgan vers el donant. Per fer-ho, es tenen en compte el conjunt de criteris (de tipus qualitatiu) explicats en el punt anterior.

A aquests algorismes se'ls anomena Mètodes d'ajuda a la presa de decisions multicriteri ja que el seu objectiu no és substituir el component humà en el procés d'elecció del receptor sinó ajudar-li a fer l'elecció. Així, el resultat serà una llista ordenada segons un criteri global de preferència (qualitatiu) on els receptors incompatibles s'hagin eliminat i on s'ofereixin diverses alternatives a l'elecció (poden existir varis pacients amb un nivell de preferència màxim). Com que es treballa interactuant amb una persona i molts dels criteris que es tenen en compte són difícils d'expressar de forma exacta per l'individu, durant una gran part del procés es treballaran amb termes qualitatius.

D'algorismes d'aquest tipus n'hi ha diversos i, segons la seva manera de tractar els termes qualitatius, utilitzen una o una altra semàntica. Les semàntiques suportades pel sistema Especialista en Trasplantaments (per tant podrà ser reconegut com a tal i se li podran enviar les dades inicials), ja descrites en anteriors seccions, són:

- Negation Functions [4]
- Fuzzy sets [5]
- 2-tuple
- Ordinal

En aquest projecte s'implementarà completament un d'aquests algorismes (i l'agent associat que es comunicarà amb l'Especialista) i es definiran els prototipus de la resta d'agents per als altres algorismes, de forma que sigui fàcil afegir-ne de nous (i així comparar-ne l'eficàcia).

7.1. ClusDM (*Clustering for decision making*)

L'algoritme a implementar, anomenat ClusDM, utilitza la semàntica de funcions de negacions i es basa en realitzar un procés de classificació o *clustering* amb les alternatives construint un arbre de classes que agruparà els diferents objectes a avaluar. A cadascuna de les classes finals obtingudes (conjunts d'alternatives inicials), se'ls assignarà un terme qualitatiu (d'un vocabulari definit) que expressi la seva relació vers del model.

Cal ressaltar el fet que, tot i que en aquest projecte l'algoritme s'utilitzi per avaluar pacients i òrgans, s'ha implementat de forma que es pugui adaptar a qualsevol domini que treballi amb criteris qualitatius (independentment del nombre de criteris, vocabularis emprats, funcions de negació associades....).

7.1.1. Funcions de distància

Durant el procés de classificació, serà necessari trobar mesures de dissimilitud entre les diferents alternatives (a partir dels valors numèrics que hàgim obtingut per a cada atribut mitjançant les funcions de negació). Aquestes mesures són les que ens permetran anar agrupant aquells objectes que a cada pas siguin semblants (*clustering*).

Per obtenir-les, caldrà utilitzar algun tipus de funció pugui comparar els objectes a partir dels valors dels seus atributs, aplicant a cadascun d'ells un pes relatiu.

A continuació es descriuen les diferents funcions de distància implementades que farà servir el ClusDM per comparar les alternatives [2]. Cal considerar que, per a cadascun dels atributs que descriuen un objecte es veu afectat per un pes més o menys gran en el resultat final (ponderació).

La nomenclatura utilitzada en la fórmules és:

- X_{ij} : valor de l'atribut i per a l'objecte j
- W_i : pes de l'atribut i
- n : nombre d'atributs

7.1.1.1. Diferències

És un coeficient basat en la mitja de les diferències entre valors.

El seu principal desavantatge està en què una distància positiva d'igual magnitud a una de negativa dona com a resultat final zero, és a dir, objectes diferents seran considerats iguals

La seva fórmula és:

$$\left| \frac{\sum_{i=1}^n W_i (X_{ij} - X_{ik})}{n} \right|$$

Els valors que assigna pertanyen a $[0, \sqrt{2}]$

7.1.1.2. M.C.D (Mean Character Difference)

Aquest coeficient agafa el valor absolut de les distància entre valors, d'aquesta manera evita la indistingibilitat dels valors amb signes contraris comentada abans.

Malgrat tot, també té alguns desavantatges, com el fet de que sempre fa una subestimació de la distància Euclidiana.

La seva fórmula és:

$$\frac{\sum_{i=1}^n W_i |X_{ij} - X_{ik}|}{n}$$

7.1.1.3. Manhattan o city Blocks

Es basa en el mateix principi que les funcions anteriors, però no té en compte el nombre d'atributs utilitzats per calcular el sumatori de les diferències entre valors

La fórmula és:

$$\sum_{i=1}^n W_i |X_{ij} - X_{ik}|$$

Funció limitada inferiorment a 0.

7.1.1.4. Camberra Metric

És un variació de la fórmula Manhattan.

Té unes propietats matemàtiques molt bones per calcular semblances, com la característica de no ser afectada pels rangs dels dominis, a més és sensible a les distàncies relatives i absolutes, especialment si treballem amb valors positius.

La seva fórmula és:

$$\sum_{i=1}^n W_i \frac{|X_{ij} - X_{ik}|}{|X_{ij}| + |X_{ik}|}$$

Funció limitada inferiorment a 0.

7.1.1.5. Taxonomic Distance

Es tracta de la generalització de la distància Euclidiana a un espai n-dimensional.

Obtenim la fórmula:

$$\sqrt{\frac{\sum_{i=1}^n W_i (X_{ij} - X_{ik})^2}{n}}$$

Funció limitada inferiorment a 0.

7.1.2. Dades d'entrada

Les dades d'entrada que són necessàries per realitzar el procés d'ordenació són les següents:

- Taula amb els termes del vocabulari de cada atribut a avaluar (tipus String).
- Taula amb les negacions dels termes del vocabulari de cada criteri (tipus String).
- Taula amb els objectes a avaluar (pacients), que conté els valors qualitius que l'Especialista ha assignat per cada atribut (tipus String).
- Funció de distància que s'utilitzarà per comparar les alternatives i pesos relatius dels diversos atributs.

7.1.3. Etapes del procés

El procés de classificació de les dades es divideix en diverses etapes que es passen a descriure a continuació.

7.1.3.1. Transformació de les dades

El primer pas consistirà en transformar les dades dels objectes d'entrada (pacients) que són del tipus atribut:terme_qualitatiu a valors amb els que es pugui treballar (quantitatius). Per fer-ho caldrà utilitzar les funcions de negació descrites anteriorment, de forma que trobem una equivalència numèrica a cada terme dels diversos vocabularis utilitzats en el diferents criteris.

Així, es construiran els intervals de negació tal com s'ha explicat en l'apartat de l'Especialista i obtindrem el valor a assignar a cada terme (el punt mig de l'interval definit per a cada etiqueta).

D'aquesta forma, obtindrem una nova matriu per als objectes a avaluar on cadascun dels seus atributs tindrà un valor entre 0 i 1 que indica el seu grau d'adequació al model (donant).

	Distància	Pes	Tamany	Antígens	Edat	Espera
Pacient_X	Hospital	Factible	Bona	Correcte	Iguals	Normal

	Distància	Pes	Tamany	Antígens	Edat	Espera
--	-----------	-----	--------	----------	------	--------

Pacient_X	13/14	3/10	3/5	5/6	15/18	1/2
-----------	-------	------	-----	-----	-------	-----

Taula 3 - Exemple de transformació de terme qualitatiu a valors numèrics utilitzant els intervals definits per les funcions de negació de l'apartat anterior

7.1.3.2. Classificació

El mètode de classificació utilitzat pertany a la família SAHN (*sequential agglomerative hierarchical non-overlapping*). Aquests algorismes generen un arbre de classificació del qual es farà un tall per obtenir una partició amb les classes resultants (grups d'objectes als que s'assignarà un terme del vocabulari final).

El passos que segueix el mètode de classificació són els següents:

- Càlcul de dissimilitud entre tots els parells d'alternatives: a partir dels valors qualitatiu per a cada objecte obtinguts en el pas inicial, utilitzarem una mesura de distància (la que se'ns indiqui per paràmetres), per construir una matriu de dissimilitud on s'expressi la diferència entre els parells d'alternatives.

	Pac_A	Pac_B	Pac_C	Pac_D	Pac_E	Pac_F	Pac_G
Pac_A	0	0.4	0.8	0.8	0.3	0.5	0.6
Pac_B	0.4	0	0.3	0.7	0.5	0.4	0.8
Pac_C	0.8	0.3	0	0.4	0.6	0.3	0.4
Pac_D	0.8	0.7	0.4	0	0.7	0.9	0.5
Pac_E	0.3	0.5	0.6	0.7	0	0.7	0.6
Pac_F	0.5	0.4	0.3	0.9	0.7	0	0.6
Pac_G	0.6	0.8	0.4	0.6	0.6	0.6	0

Taula 4 - Exemple de matriu de dissimilituds: matriu triangular (dist A-B=dist B-A) amb la diagonal principal a 0 (dist X-X=0)

- Agrupació dels objectes segons el grau de dissimilitud: buscarem aquells parells d'alternatives que tinguin un valor de dissimilitud mínim i els agruparem. Cal tenir en compte que les agrupacions es faran de manera incremental. L'algoritme utilitzat és el següent:
 - o Buscar el valor més petit de la matriu.
 - o Formar una classe amb cada parell d'objectes que tenen aquest valor de dissimilitud. Si dos parells o més d'alternatives tenen elements en comú (ex: B-C i C-F) es fusionaran en una sola classe.
 - o Eliminar els objectes que formen part de les noves classes de la matriu de dissimilitud.
 - o Calcular el prototipus de les noves classes: per cada atribut es fa la mitja entre els valors dels atributs dels objectes que la formen.
 - o Afegir a la matriu les noves classes creades com si fossin un objecte més. Per fer-ho caldrà calcular la seva distància (segons la funció escollida) vers la resta d'alternatives.

	Pac_A	Pac_B	Pac_C	Pac_D	Pac_E	Pac_F	Pac_G
Pac_A	0	0.4	0.8	0.8	0.3	0.5	0.6
Pac_B	0.4	0	0.3	0.7	0.5	0.4	0.8
Pac_C	0.8	0.3	0	0.4	0.6	0.3	0.4
Pac_D	0.8	0.7	0.4	0	0.7	0.9	0.5
Pac_E	0.3	0.5	0.6	0.7	0	0.7	0.6
Pac_F	0.5	0.4	0.3	0.9	0.7	0	0.6
Pac_G	0.6	0.8	0.4	0.6	0.6	0.6	0

Taula 5 - Exemple de matriu de dissimilituds: selecció de valors mínims

En la matriu de l'exemple, el valor mínim és 0.3. Com que hi ha 3 parells d'objectes que tenen aquest valor, es formaran les següents classes noves:

- o A-E: Classe 1.
- o B-C; C-F tenen elements en comú (C), per tant es fusionaran: B-C-F: Classe 2.

	Pac_D	Pac_G	Classe_1	Classe_2
D	0	0.5	??	??
G	0.5	0	??	??
Classe_1	??	??	0	??
Classe_2	??	??	??	0

Taula 6 - Exemple de matriu de dissimilituds: actualització dels objectes

Aquest és l'aspecte de la matriu un cop eliminats els objectes que formen part de les noves classes i afegint aquestes. Caldrà calcular el prototipus de les classes noves per poder trobar els valors de dissimilitud vers la resta d'objectes

- Es torna a repetir el procés anterior partint de la nova matriu de dissimilitud. En la nova iteració serà possible que diverses classes es fusionin entre elles o amb altre objectes.
- Cada cop que creem una nova classe, caldrà guardar quins objectes en formen part, per tal d'anar construint un arbre de classes.

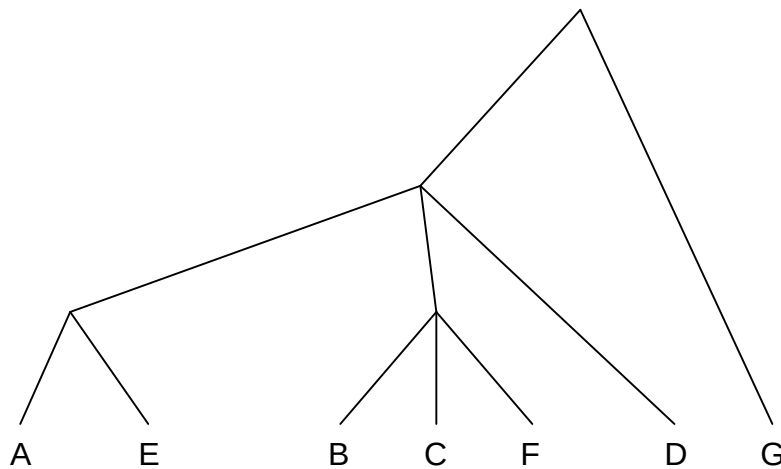


Figura 17- Exemple d'arbre de classes

- El procés acaba quan el nombre d'alternatives que queden sigui el desitjat: tantes com termes tingui el vocabulari final (7). Donat que no podem assegurar que sempre obtinguem el nombre de classes ideal (7), oferirem més possibilitats seguint un ordre de preferència: 7 -> 8 -> 6 -> 9 -> 5. Si el nombre final aconseguit no és cap dels anteriors es s'avortarà el procés i es retornarà un error.

7.1.3.3. Ordenació de les classes

El següent pas consistirà en ordenar les classes obtingudes després del procés de *clustering* comparant-les amb el model (donant). Això es fa així, ja que com més s'assemblin els receptors al donant, més possibilitats hi ha que els trasplantament sigui un èxit.

Per fer-ho, s'utilitzarà la mateixa mesura de distància emprada en el procés anterior entre el prototipus de les classes finals i el model. Al donant li assignarem sempre els valors corresponent a les etiquetes màximes per a cada atribut, ja que ell seria el receptor ideal per al seu òrgan.

7.1.3.4. Semàntica de les classes

En aquest punt, s'assignarà un terme d'un vocabulari final a cada classe (conjunt d'objectes inicials), que expressi el seu grau d'adequació al model, segons la distància vers el donant obtinguda en el pas anterior.

Donat que el nombre de classes finals pot variar segons la regla definida anteriorment (7 -> 8 -> 6 -> 9 -> 5), caldrà haver definit un vocabulari prou extens, utilitzant en cada cas uns o altres termes.

Vocabularis definits en funció del nombre de termes a utilitzar:

Voc_9 = {pèssim,dolent,desfavorable,fluix,factible,adequat,bo,molt_bo,òptim}

Voc_8 = { pèssim, dolent, desfavorable, factible, adequat, bo, molt_bo, òptim }

Voc_7 = { pèssim, dolent, desfavorable, factible, adequat, bo, òptim }

Voc_6 = { pèssim, desfavorable, factible, adequat, bo, òptim }

Voc_5 = { pèssim, desfavorable, factible, adequat, òptim }

Per expressar el resultat, es recuperaran tots els objectes que formen part de cada classe i se'ls assignarà el terme del vocabulari final que correspon al seu grup.

Com a resultat final, es retornarà una llista ordenada segons aquest vocabulari on s'inclouen els identificadors dels pacients i el terme qualitatiu assignat.

8. Distribució modular del sistema

Per tal d'estructurar tot el sistema descrit anteriorment i permetre la seva execució en entorns distribuïts, el codi s'ha dividit en una sèrie de mòduls que es passen a descriure a continuació

8.1. ClusDM.java

És l'objecte Java que implementa l'algoritme de *Clustering* utilitzat en el procés d'ordenació de les alternatives. Aquest mòdul és manegat per l'agent corresponent (**ClusDMResponderAgent**), que serà l'encarregat de comunicar-se amb l'Especialista i preparar les dades rebudes a través dels missatges per a que l'algoritme les pugui interpretar.

L'objecte no té funcionalitat pròpia (no té programa principal) i requereix de la crida al seu constructor per obtenir-ne una instància. Les funcions més importants que es troben dins d'aquest mòdul són:

- **ClusDM()**: és el constructor buit de l'objecte, permet que l'agent associat el pugui referenciar.
- **Clustering (DadesClusDM dades, ResultatMCDM resultat[])**: funció principal accessible externament que realitza tot el procés de classificació: trobar els intervals de negació, transformar les dades de les alternatives a numèric, realitzar el procés de *clustering* i assignar el vocabulari final al resultat. Com a informació d'entrada rep un objecte tipus **DadesClusDM** que l'agent corresponent s'haurà encarregat d'omplir adequadament a partir de les dades que ha rebut per missatge. Un cop s'hagi completat l'algoritme, es retornarà una taula d'objectes **ResultatMCDM** que conté els identificadors dels objectes i els termes assignats. En cas que no s'obtinguin un nombre de classes vàlid (entre 5 i 9), es retornarà un error.
- Funcions internes a l'objecte:
 - o **double distancia (double pes[], double obj1[], double obj2[], int distancia)**: aquesta funció implementa totes les funcions de distancia utilitzades per comparar alternatives (*diferència, taxonòmic, manhattan...*). Rep una taula amb els pesos dels atributs a comparar, els valors dels atributs del parell d'objectes a avaluar i un identificador del tipus de distància a utilitzar. Retorna la distància entre les dues alternatives.
 - o **void CalculaNeg (String neg[][], double val[])**: aquesta funció calcula l'equivalent numèric (punt mig de l'interval de negació) per a cada etiqueta. Rep un matriu amb la llista de termes i les seves negacions i retorna el seu corresponent valor.
 - o **void Clustering ()**: aquesta funció realitza una iteració del procés de *clustering*: trobar la matriu de dissimilituds, buscar els valors mínims,

formar les classes, trobar els prototipus i actualitzar les alternatives per a la nova iteració. L'execució d'aquest mètode és controlada per la funció principal que avaluarà en tot moment el nombre de classes totals que queden, aturant el procés quan s'arribi al nombre final (idealment 7).

8.2. ClusDMResponderAgent.java

Aquest és l'agent associat a l'algoritme de *clustering* anterior. Té un comportament simple que rep peticions d'ordenació, les processa, prepara les dades, les passa a l'objecte **ClusDM** i amb el resultat, construeix el missatge de resposta. El protocol de comunicació que utilitza és de tipus **FIPA-Request**.

Les parts principals que el formen són:

- **class ClusDMResponseBehaviour**: subclasse que exten el comportament **FipaRequestResponderBehaviour** per tal de poder manegar peticions de tipus **Request** enviades per l'agent Especialista. Aquest és un comportament simple que s'instancia, s'executa i es destrueix cada vegada que l'*scheduler* de JADE rep una petició de **Request**. Les funcions principals dins del comportament són:
 - o **ClusDMResponseBehaviour (Agent agent, ACLMessage requestMsg)**: constructor del comportament.
 - o **action()**: són les accions que s'executaran un cop rebuda la petició. El primer que fa és extreure les dades que s'han enviat al missatge i omplir un objecte tipus **DadesClusDM** que és el que rebrà l'algoritme **ClusDM**. Si es troba algun error durant l'avaluació d'aquestes dades (pesos dels atributs erronis), es retornarà un missatge de refús de la petició i es finalitzarà el comportament. Si totes les dades són correctes, s'instanciarà un objecte tipus **ClusDM** i s'executarà la funció de *clustering*. Si el resultat retornat és correcte, es compondrà un missatge tipus **inform** adjuntat les dades; si s'ha produït algun error, es s'enviarà un missatge tipus **failure** indicant la rao de fallada.
 - o **boolean done()**: conjunt d'accions a realitzar quan s'acaba el comportament: en aquest cas, cap.
 - o **void reset()**: accions a realitzar quan es reinicialitza el comportament. Donat que aquest s'executa només un sol cop, no és necessari definir-ne cap.
- **ClusDMResponderAgent()**: constructor de l'agent.
- **Setup()**: aquest mètode es crida un sol cop quan s'arrenca l'agent i és l'encarregat de realitzar el procés de configuració previ a qualsevol acció. Aquí s'inicialitza l'ontologia, el llenguatge, els comportaments i es fa el registre al DF. Aquest últim pas és molt important, doncs és on es defineix quin tipus d'agent és i permetrà a l'Especialista trobar-lo i posar-se en contacte. Per tal que s'identifiqui correctament a l'agent, s'especificarà que és de tipus MCDM i el servei que proporciona utilitza una semàntica de funcions de negació.

- **void takeDown():** funció que conté les accions a realitzar quan l'agent acabi la seva execució. En aquest punt, es realitzarà el desregistre al DF.
- **int Parse_missatge (MCDMRequest cr, DadesClusDM dades):** funció auxiliar a l'agent que s'encarrega d'extreure les dades d'un missatge de **Request** i omplir un objecte tipus **DadesClusDM** que serà el que rebrà l'algoritme d'ordenació. Durant el procés, es comproven que les dades són correctes: pesos vàlids, i dades dintre del rang (nombre d'alternatives i d'atributs).

8.3. DadesClusDM.java

Objecte que defineix totes les estructures de dades que ha de rebre l'algoritme **ClusDM** per realitzar el procés d'ordenació. A més, inclou una funció d' inicialització (**void Inicialitza (int nObjectes, int nAtributs)**) que defineix la dimensió de les taules dinàmicament.

El conjunt de dades requerides són:

- **int nObjectes:** nombre d'objectes a avaluar.
- **int nAtributs:** nombre d'atributs per a cada alternativa.
- **String Atributs[][]:** taula amb els vocabularis dels atributs.
- **String NegAtributs[][][]:** taula amb les negacions dels termes dels vocabularis dels atributs.
- **double []pesos:** pesos dels atributs (en percentatges).
- **int tipus_dist:** funció de distancia escollida.
- **String objectes[][]:** taula d'objectes amb els valors de cada atribut.
- **int idObj[]:** taula d' identificadors d' objectes.

8.4. DadesEspecialista.java

Defineix el conjunt d'estructures de dades que ha de rebre l'objecte Especialista per poder realitzar el procés de filtratge i comparació dels pacient. També es defineixen les estructures on es guardaran els resultats, que seran les que necessitaran els diversos algoritmes d'ordenació. S'inclou una funció d'inicialització (**void Inicialitza (int n)**) que defineix la dimensió de les taules dinàmicament. El conjunt de dades definides són:

- Dades internes a l'especialista:
 - o **int taulaABO[][]:** taula de tipus de sang. Cada fila indica un tipus de sang. Cada columna de cada fila és un tipus compatible amb el de la fila.
 - o **String zones[][]:** taula amb les zones del país. Cada fila indica una zona. Cada columna és una província que pertany a la zona de la fila
- Dades d'entrada:
 - o **String NomAtributs[]:** nom dels atributs.
 - o **int nPacients:** nombre de pacients.
 - o **int nAtributs:** nombre d'atributs.
 - o **Pacient pacients[]:** taula d'objectes Pacient.

- o **Donant donant:** objecte Donant.
- Dades a enviar als agents d'ordenació:
 - o **String Atributs[][]:** taula de vocabularis de cada atribut. Cada fila és un atribut. Cada columna és un terme del vocabulari de l' atribut.
 - o **String NegAtributs[][][]:** taula de negacions dels elements del vocabulari dels atributs. La primera dimensió és l' atribut. La segona és el terme dins del vocabulari de l' atribut. La tercera és la llista de negacions per al terme del vocabulari de l' atribut.
 - o **int ValorsDifusos[][][][]:** taula de valors difusos (extrems del triangle que defineix cada interval) per als elements del vocabulari dels atributs. La primera dimensió és l' atribut. La segona és el terme del vocabulari de l' atribut. La tercera és la llista de vèrtexs del triangle (3). La quarta són els valors del vèrtex (2): numerador i denominador (per mantenir la precisió, doncs a través dels missatges entre agents no es poden enviar nombre decimals, només enters).
 - o **double pesos[]:** pesos dels atributs.
 - o **int tipus_dist:** funció de distància escollida.
 - o **String pacients_atrib[][]:** taula de pacients amb els valors de cada atribut. Les files són els pacients. Les columnes són els valors assignat per a cada atribut (6).
 - o **int idPac[]:** taula d' identificadors de pacient.

8.5. Donant.java

Objecte que reuneix tots els camps amb la informació que ha de rebre l'Especialista en Trasplantaments sobre el Donant. Es defineixen les estructures de dades i un constructor que les inicialitza. El conjunt de dades que s'inclouen són les següents:

- **String Hospital:** hospital on es troba l'òrgan.
- **String Localitat:** localitat on està situat l'hospital.
- **String Provincia:** província a la que pertany la localitat.
- **int Tipus_sang:** tipus de sang: A, B, AB, O.
- **double Pes:** pes de la persona (en kilograms).
- **double Tamany:** tamany de l'òrgan.
- **boolean Antigens[][]:** taula que indica l'existència o no dels 4 antígens.
- **Calendar Data_naix:** data de naixement.
- **String Organ:** tipus d'òrgan: ronyó, pulmons, fetge a cor.

Constructor:

Donant(String Hospital, String Localitat, String Provincia, int Tipus_sang, double Pes, double Tamany, boolean[][] Antigens, Calendar Data_naix, String Organ)

8.6. Dummy2TupleAgent.java

Agent de prova que es registra al DF amb tipus MCDM i semàntica **2-tuple**, permetent l'addició senzilla de nous algoritmes basats en ella. Estructuralment és idèntic a l'agent **ClusDMResponderAgent** però, donat que no s'ha implementat un algoritme d'aquest tipus, per cada petició que rebí sempre retornarà **Failure**.

8.7. DummyFuzzyAgent.java

Agent de prova que es registra al DF amb tipus MCDM i semàntica **Fuzzy-sets**, permetent l'addició senzilla de nous algoritmes basats en ella. Estructuralment és idèntic a l'agent **ClusDMResponderAgent** però, donat que no s'ha implementat un algoritme d'aquest tipus, per cada petició que rebí sempre retornarà **Failure**.

8.8. DummyNegationAgent.java

Agent de prova que es registra al DF amb tipus MCDM i semàntica **Negation Functions**, permetent l'addició senzilla de nous algoritmes basats en ella. Estructuralment és idèntic a l'agent **ClusDMResponderAgent** però, donat que no s'ha implementat un algoritme d'aquest tipus, per cada petició que rebí sempre retornarà **Failure**.

8.9 DummyOrdinalAgent.java

Agent de prova que es registra al DF amb tipus MCDM i semàntica **Ordinal**, permetent l'addició senzilla de nous algoritmes basats en ella. Estructuralment és idèntic a l'agent **ClusDMResponderAgent** però, donat que no s'ha implementat un algoritme d'aquest tipus, per cada petició que rebí sempre retornarà **Failure**.

8.10. DummyResponderAgent.java

Agent de prova que es registra al DF amb tipus MCDM i qualsevol semàntica, permetent l'addició senzilla de nous tipus. Estructuralment és idèntic a l'agent **ClusDMResponderAgent** però, donat que no té funcionalitat, sempre retornarà **Failure**.

8.11. Pacient.java

Objecte que reuneix tots els camps amb la informació que ha de rebre l'Especialista en Trasplantaments sobre cada possible receptor. Es defineixen les estructures de dades i un constructor que les inicialitza. El conjunt de dades que s'inclouen són les següents:

- **String Hospital:** hospital on es troba l'òrgan.
- **String Localitat:** localitat on està situat l'hospital.

- **String Provincia:** província a la que pertany la localitat.
- **int Tipus_sang:** tipus de sang: A, B, AB, O.
- **double Pes:** pes de la persona (en kilograms).
- **double Tamany:** tamany de l'òrgan.
- **boolean Antigens[][]:** taula que indica l'existència o no dels 4 antígens.
- **Calendar Data_naix:** data de naixement.
- **Calendar Data_esp:** data d'entrada a la llista d'espera.
- **String Organ:** tipus d'òrgan: ronyó, pulmons, fetge a cor.

Constructor:

Donant(String Hospital, String Localitat, String Provincia, int Tipus_sang, double Pes, double Tamany, boolean[][] Antigens, Calendar Data_naix, Calendar Data_esp, String Organ)

8.12. ResultatMCDM.java

Objecte que defineix les dades que s'han de retornar com a resultat del procés d'ordenació per a cada pacient:

- **String grau:** terme assignat al objecte segons la seva adequació al model.
- **int id:** identificador (es pren com a ID l'ordre d'arribada dels objectes).

Constructor:

ResultatMCDM(String grau, int id)

8.13. Especialista.java

És l'objecte Java que implementa les funcions de filtratge dels receptors (aplicant les regles explicades anteriorment) i de transformació de les dades dels pacients, comparant-les amb el donant i obtenint mesures de bondat per a cada atribut.

Aquest mòdul no té funcionalitat pròpia i requereix que un altre l'instancii a través del seu constructor per tal d'accedir a les funcions. Els mètodes principals són:

- **Especialista (DadesEspecialista dades):** és el constructor i l'únic accessible externament. A partir d'una referència a un objecte amb les dades d'entrada, realitzarà tot el procés (filtratge i transformació) i guardarà els resultats (relació entre pacient i donant per cadascun dels atributs) a les estructures que s'han definit en aquell objecte. Els identificadors utilitzats per referir-se als receptors seran l'ordre d'introducció a la llista i es mantindran en el moment de passar-los als algoritmes d'ordenació (doncs pot ser que alguns pacients quedin desestimats i caldrà utilitzar un referent comú).
- A més, existeixen una sèrie de funcions internes auxiliars que s'encarreguen de tractar cada atribut que es tindrà en compte:
 - o **String Avalua_distancia (Pacient pacient, Donant donant, String zones[][]):** com a paràmetres d'entrada, rep un objecte Pacient i el

Donant, juntament amb la taula de zones (grups de províncies que pertanyen a una mateixa zona). Després de comparar la localització geogràfica de donant i receptor, es retornarà un *String* amb el terme del vocabulari de l'atribut "distància" corresponent.

- o **boolean Avalua_sang (Pacient pacient, Donant donant, int taulaABO[][]):** rep un objecte Pacient i l'objecte Donant, així com la taula de compatibilitat sanguínia. En aquest cas, donat que l'atribut representa un filtre (excloent), només es retorna un booleà que indica si la sang de donant i receptor són o no compatibles.
- o **String Avalua_pes (Pacient pacient, Donant donant):** a partir dels pesos del pacient i donant contingut als objectes respectius, es compararan per obtenir la mesura de relació entre ells. Com a resultat, es retorna el terme del vocabulari de l'atribut "pes de l'individu" corresponent o el terme "null" (pacient no vàlid) en cas que els valors superin el rang màxim.
- o **String Avalua_tamany (Pacient pacient, Donant donant):** el mateix comportament que l'anterior però en aquest cas, treballant sobre l'atribut "tamany de l'òrgan".
- o **String Avalua_antigens(Pacient pacient, Donant donant):** en aquesta funció es farà un recompte de les diferències entre els antígens del donant i del pacient (entre 0 i 4), retornant el terme del vocabulari corresponent.
- o **String Avalua_edat (Pacient pacient, Donant donant):** avaluant les dates de naixement del pacient i el donant, es troba quina és la diferència d'edat. Segons aquest valor, es retornarà un terme del vocabulari de l'atribut "edat" o "null" (pacient no vàlid) si se supera el rang màxim.
- o **String Avalua_espera (Pacient pacient):** en aquest cas només és necessària la data d'entrada del pacient a la llista d'espera. Comparant-la amb la data actual, s'obté el nombre de mesos que porta i se li assigna un terme del vocabulari corresponent.
- o **boolean Avalua_organ (Pacient pacient):** una altra funció de filtratge: obté l'edat del pacient a partir de la seva data de naixement i, en funció del tipus d'òrgan que vulgui rebre, es decideix si el trasplantament és o no possible.

8.14. Gui.java

En aquest mòdul es defineix la interfície gràfica d'interacció amb l'usuari: components gràfics i accions de resposta a events (pulsació de botons, selecció de valors...) sobre aquests. En la finestra corresponent es permet introduir les dades d'entrada (fitxer de pacients/donant, funció de distància, pesos dels atributs); escollir l'agent MCDM amb qui es vol interactuar d'entre els disponibles, realitzar peticions d'ordenació i, en funció dels resultats obtinguts, donar una valoració que permetí a l'Especialista recomanar o no la utilització de determinats mètodes.

Disposa d'un constructor (**public gui(MCDMRequesterAgent cra)**) al que li haurem de passar una referència de l'agent que manega la comunicació amb els MCDM

(el que fa el paper d'Especialista) per poder accedir als seus mètodes (cerca d'agents, inici de la comunicació...)

8.15. Info.java

Objecte auxiliar a la GUI que permet mostrar finestres d'informació quan es produeixen determinats events: errors en les dades, valors no introduïts, recomanacions de determinats agents MCDM....

Disposa d'un constructor (**public info()**) que l'inicialitza i d'una funció (**void Missatge (String s)**) que permet escollir el contingut del missatge a mostrar.

8.16. MCDMRequesterAgent.java

Aquest mòdul afegeix la funcionalitat d'Agent a l'Especialista. Així, el conjunt de *Especialista.java* (nucli), *gui.java* (interacció amb l'usuari) i *MCDMRequesterAgent.java* (agent), és el que forma l'anomenat Agent Especialista en trasplantaments. Com tots els de la seva classe, implementarà una sèrie de comportaments que definiran la seva manera d'actuar. En aquest cas són dos:

- **class MCDMInputBehaviour:** classe que exten el comportament simple **SequentialBehaviour** que s'utilitza per l'execució lineal d'una sèrie d'accions. Aquest serà el comportament principal i s'encarregarà de demanar les dades d'entrada a l'usuari (mitjançant la instància del mòdul d'interfície *gui.java*), realitzar les comunicacions amb els agents d'ordenació disponibles, mostrar els resultats (a la finestra) i captar la valoració de l'usuari sobre aquests, que el permetrà en un futur realitzar recomanacions sobre determinats agents (a través de la interfície). Les funcions principals que troben dintre d'aquest comportament són:
 - o **MCDMInputBehaviour (Agent myAgent):** constructor de l'agent.
 - o **void preAction():** espera a que les dades d'entrada s'hagin introduït (mitjançant la finestra gràfica) i crea el missatge de **Request** a enviar a l'agent escollit. Com a últim pas, instanciarà el subcomportament que s'encarregarà d'implementar el protocol **FIPA-Request** amb l'agent d'ordenació.
 - o **ACLMessage createMCDMRequest():** funció auxiliar que inicialitza els camps necessaris a un missatge de **Request**: agent que l'envia, ontologia, llenguatge, protocol... També és l'encarregat de fer la instància i la crida al mòdul Especialista (que realitzarà el filtratge dels receptors i les transformacions de les dades), a partir de l'objecte **DadesEspecialista** que s'haurà omplert a mesura que l'usuari introdueixi les dades. Finalment, amb el resultat retornat per l'Especialista, s'omplirà el contingut del missatge seguint les regles definides a l'ontologia i es retornarà.
 - o **void postAction():** conté el conjunt d'accions realitzar després de cada execució del comportament principal. En aquest cas es reinicialitzarà de nou.

- o **void reset()**: reinicia el comportament principal comprovant que no existeix cap subcomportament secundari en execució.
- **class MCDMBehaviour**: classe que exten el comportament **FipaRequestInitiatorBehaviour** que implementa la banda *Iniciator* del protocol **FIPA-Request**. En aquest cas, s'ompliran una sèrie de funcions (definides per JADE) que s'encarreguen de manejar els possibles missatges que puguin arribar (**inform**, **failure**, **refuse**...) i generar respostes. Així, dins d'aquest, trobem les següents funcions:
 - o **MCDMBehaviour (Agent agent, ACLMessage requestMsg)**: constructor del comportament.
 - o **void handleAgree(ACLMessage msg)**: manegador de missatges **Agree**.
 - o **void handleInform(ACLMessage msg)**: manegador de missatges **Inform**: es recupera el resultat retornat per l'agent d'ordenació i es mostra per pantalla.
 - o **void handleNotUnderstood(ACLMessage msg)**: manegador de missatges **not-understood**.
 - o **void handleFailure(ACLMessage msg)**: manegador de missatges **Failure**: es recupera la causa de la fallada i es mostra per pantalla.
 - o **void handleRefuse(ACLMessage msg)**: manegador de missatges **Refuse**: es recupera la raó del refús i es mostra a l'usuari.
- **MCDMInputBehaviour (Agent myAgent)**: constructor de l'agent.
- **void setup()**: aquest mètode és cridat un sol cop al principi i s'encarrega d'inicialitzar l'ontologia i el llenguatge, així com realitzar el registre de l'agent al DF, indicat de quin tipus és i quin servei proporciona. A més, des d'aquí es realitza la instància del component gràfic associat (passant-li la pròpia referència) i la creació del comportament principal (**MCDMInputBehaviour**).
- **void takeDown()**: aquesta funció s'encarrega de finalitzar correctament l'agent quan s'acaba d'executar. En aquest cas, s'eliminaran els subcomportaments que hi puguin haver i es farà el desregistre al DF.
- **Vector BuscarAgents ()**: és una funció auxiliar que s'encarrega de buscar tots els agents de tipus MCDM disponibles al sistema. Com a resultat, retorna un vector amb la llista de Noms+semàntica dels agents trobats.
- **void Parse_Missatge(MCDMRequest cr, DadesEspecialista dades)**: una altra funció auxiliar que s'encarrega d'omplir el contingut d'un missatge de **Request** amb les dades retornades per l'Especialista, utilitzant les regles definides a l'ontologia.
- **void Llegir_Fitxer (DadesEspecialista dades, Properties prop)**: aquesta funció rep una referència a l'objecte *Properties* (parells clau-valor) corresponent al fitxer d'entrada de dades i una altra a l'objecte de dades de l'Especialista. S'encarregarà d'omplir els camps del segon amb les dades contingudes al primer.

8.17. El paquet Ontology

Com ja s'ha definit anteriorment, l'ontologia és el que permet als agents "entendre" el contingut dels missatges que s'envien. La formen un conjunt de *frames* i *slots* que s'han de seguir durant la creació del missatge per tal que el receptor sigui capaç d'extreure la informació. Aquest és un dels punts que cal haver definit abans d'implementar els agents, i està formada per dues parts principals:

- Definició: en aquest punt cal especificar per quins *frames*, accions i predicats estarà composta l'ontologia. Dintre de cadascun d'ells haurem d'incloure el conjunt d'*slots* que els formen (indicat el tipus de dada que són) i la classe que s'encarregarà de manejar-los (implementarà els mètodes d'inicialització i accés sobre les dades). L'objecte on s'inclouen aquestes definicions és **ontology/MCDMOntology.java**. L'estructura de l'ontologia utilitzada està definida a la secció Arquitectures.
- Implementació: per cada element definit a l'ontologia, haurem de crear una classe Java que implementi els mètodes d'inicialització (SETxxx) i consulta (GETxxx) per a cadascun dels *slots* que contingui. El conjunt de fitxers Java que implementen cadascun dels elements definits són:
 - o **ontology/Atrib_objectes.java**: permet manejar les dades dels objectes a classificar: valors dels seus atributs i identificadors.
 - o **ontology/Atribut.java**: permet accedir a les dades dels atributs: pesos, i vocabulari.
 - o **ontology/ClassesInsuficients.java**: implementa l'error produït quan el nombre de classes finals obtingudes són insuficients per al vocabulari definit.
 - o **ontology/Domain_OntologyObj.java**: implementa l'interface **OntologyEntity** i dona suport a les variables. La resta de classes l'extendran per simplificar la implementació.
 - o **ontology/Grau.java**: permet l'accés a les dades de cadascun dels objectes del resultat: terme del vocabulari final assignat i identificador.
 - o **ontology/MassaAtributs.java**: implementació de l'error que es produeix quan el nombre d'atributs supera el límit.
 - o **ontology/MassaObjectes.java**: implementació de l'error que es produeix quan el nombre d'objectes supera el límit.
 - o **ontology/MCDMRequest.java**: implementa l'acció principal: enviar un **request** a un agent MCDM. El conjunt de dades que s'envien són: nombre d'atributs, atributs amb els seus vocabularis, funcions de negació i valors difusos, funció de distància escollida, nombre d'objectes i llista d'objectes a avaluar amb els valors dels seus atributs i identificadors.
 - o **ontology/Negacio.java**: permet accedir a les negacions de cada terme d'un vocabulari determinat.
 - o **ontology/PesosNoValids.java**: implementació de l'error que es produeix quan el valor dels pesos no és vàlid (no suma 100 %).
 - o **ontology/Resultat.java**: dona accés a les dades del resultat: nombre d'objectes i llista d'aquests amb el seu identificador i el terme del vocabulari final assignat.

- o **ontology/Terme.java**: permet accedir a les dades de cada terme d'un vocabulari necessàries per al manegament: el seu valor, les seves negacions i els valors difusos corresponents.
- o **ontology/Triangle_difus.java**: implementa l'estructura amb la tripleta de valors difusos corresponent per a cada terme del vocabulari.
- o **ontology/Valor_atribut.java**: permet accedir als valors dels atributs per a cada objecte a avaluar.

9. Manual d'usuari

En aquest apartat es descriuen els aspectes que cal conèixer i tenir en compte l'usuari del sistema per poder interactuar-hi correctament.

9.1. Interfície

Donat que aquest projecte només és una part de tot el sistema, no hi hauria d'haver interacció amb l'usuari ja que aquesta es realitza a través del Agent d'Interfície/Coordinador de trasplantaments. Així, l'agent Especialista rebria les peticions d'aquest agent, les processaria i enviaria als diversos agents MCDM que trobés al sistema i li retornaria els resultats. A més, esperaria a rebre una valoració de les dades que ha enviat i, en funció d'aquesta escolliria o desestimaria determinats agents d'ordenació en el següent pas.

Tot i això, tenint en compte que el mòdul d'elecció del receptor de l'òrgan encara ha de desenvolupar-se abans d'integrar-se amb la resta del sistema (afegint nous agents amb altres algorismes, calibrar els paràmetres, valorar adequadament els resultats per escollir-ne els millors...), s'ha implementat una interfície gràfica a nivell de l'Especialista que “simuli” l'agent Coordinador principal. D'aquesta forma, les diferents proves sobre el sistema es podran fer més fàcilment i còmodament.

Per arrencar el sistema des del JADE iniciant tots els agents, utilitzarem la següent comanda:

```
java jade.Boot -platform clusdm1:ClusDMResponderAgent
dummy1:DummyNegationAgent dummy2:DummyFuzzyAgent
dummy3:Dummy2TupleAgent dummy4:DummyOrdinalAgent
requester:MCDMRequesterAgent
```

En fer-ho s'iniciarà la plataforma i es carregaran els diferents agents registrant els seus serveis al DF (en aquest cas, hem arrencat un agent per a cada tipus de semàntica, un agent *ClusDM* i l'agent Especialista o *Requester*).

A continuació es mostrarà la següent finestra des d'on podrem interactuar amb el sistema:

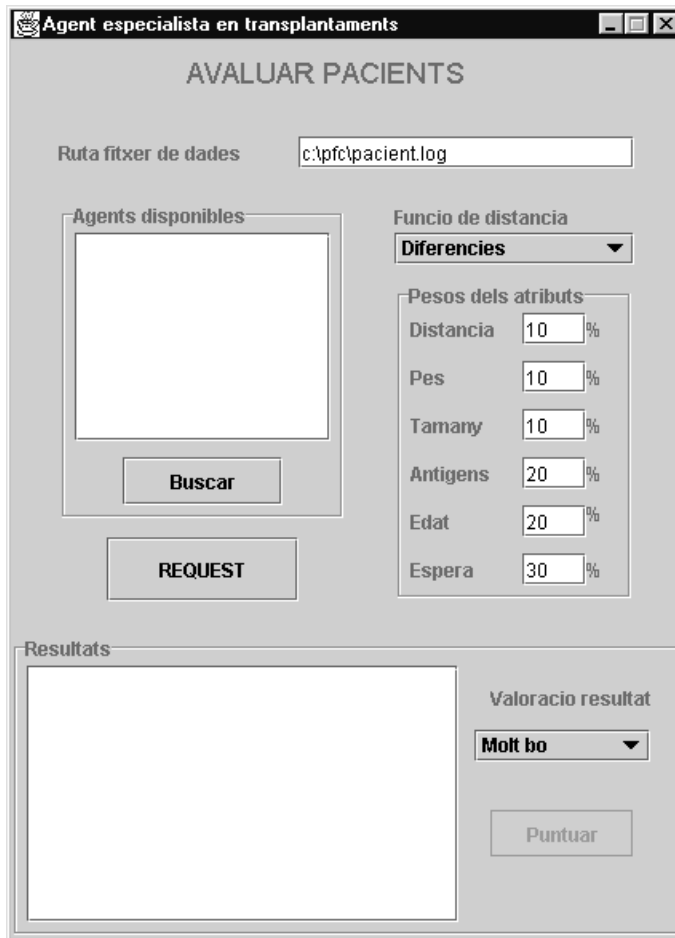


Figura 18 - Aspecte de la finestra principal de l'Agent Especialista en Trasplantaments

Observem com l'àrea gràfica està dividida en dos parts: introducció de les dades i avaluació dels resultats.

En la primera regió tenim les següent possibilitats:

- Introducció de la ruta del fitxer amb les dades dels pacients i el donant. Caldrà indicar una ruta absoluta o relativa la directori actual.

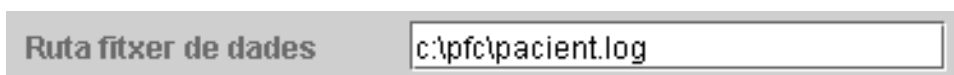


Figura 19 - Camp d'introducció de la ruta del fitxer de dades

- Cerca i selecció dels agents MCDM disponibles al sistema. En aquest punt si pitgem sobre el botó **Buscar**, es mostrarà una llista amb tots els agents que s'han trobat (preguntant-li al DF aquells que són del tipus MCDM). Al costat del nom, també s'indicarà la semàntica que utilitza (*Negacion functions, 2-tuple, fuzzy-sets, ordinal*). Seleccionant un agent qualsevol de la llista escollirem aquell amb el qual ens comunicarem.

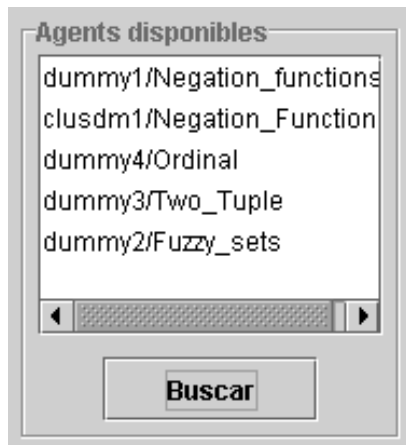


Figura 20 - Finestra de selecció de l'agent MCDM

- Un cop seleccionat l'agent MCDM, el següent pas serà configurar els paràmetres que s'aplicaran durant el procés d'ordenació de les alternatives: els pesos dels atributs i la funció de distància per comparar els objectes. Per al primer cas, haurem d'omplir la llista de valors disponibles per cada atribut amb un número entre 0 i 100. Cal tenir en compte que, per a que el procés es realitzi correctament, la suma de tots els pesos dels atributs haurà de ser 100.

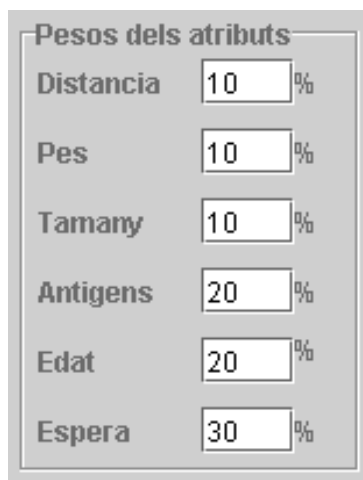


Figura 21 - Finestra d'introducció dels pesos de cada atribut

- Per escollir la funció de distància a utilitzar d'entre les disponibles, només caldrà desplegar la llista i seleccionar la que es desitgi.



Figura 22 - Finestra de selecció de la funció de distància

- Si s'intenta iniciar la comunicació sense haver introduït alguna dada o aquesta té un valor erroni, es mostrarà una finestra on se'ns informa d'aquesta situació.



Figura 23 - Finestra d'error al obrir el fitxer de dades

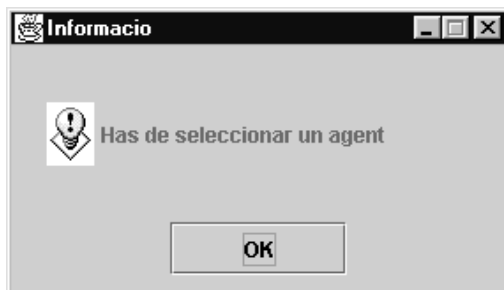


Figura 24 - Finestra d'error al selecció un agent

- Un cop introduïdes totes les dades i seleccionat l'agent, només ens caldrà pulsar el botó **Request** per iniciar el procés de comunicació. Durant aquesta etapa, s'inhabilitaran els botons de cerca i de **request** per tal que no s'entorpeixi el procés. A més, a la finestra de resultats, se'ns mostrarà informació de com evoluciona la comunicació. Un cop finalitzat el procés, s'escriuran els resultats (grups d'identificadors de pacients amb la mateixa valoració) i s'habilitarà el botó de valoració (fins ara estava desactivat).

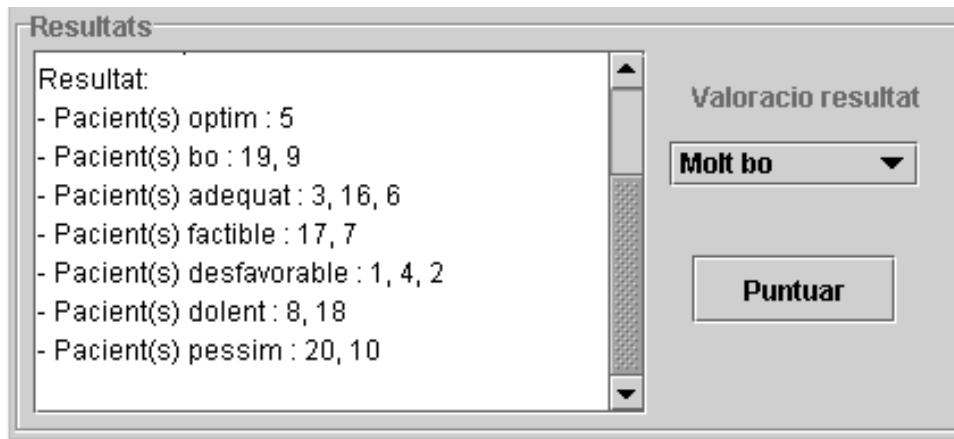


Figura 25 - Finestra de resultats i valoracions

- A partir d'aquestes dades, podrem escollir una valoració dintre del vocabulari que està disponible i polsant sobre el botó **Puntuar** s'emmagatzemarà aquest valor. Just a continuació el botó es desactivarà, assegurant que només es puntua un sol cop per cada ordenació.



Figura 26 - Finestra de selecció de la valoració del resultat

- Amb les successives puntuacions que rebrà l'Especialista per a cadascun dels agents que s'utilitzin (les quals es guarden a un fitxer per recuperar-les d'una sessió a una altra), es podrà avaluar el comportament de cadascun dels algoritmes i desestimar els més dolents o recomanar els més bons. D'aquesta forma, un cop l'Especialista disposi d'informació sobre un agent MCDM, en el moment de seleccionar-lo per realitzar el **request**, es mostrarà una finestra on s'informarà de com s'ha comportat en les proves anteriors.

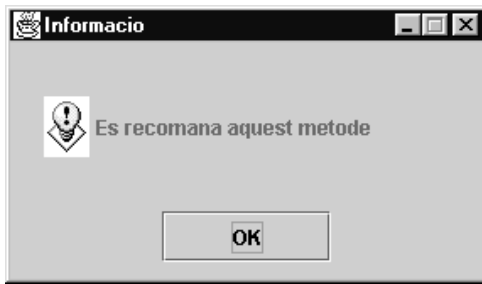


Figura 27 - Exemple de finestra de recomanació d'un agent MCDM

A banda de la interfície que permet introduir les dades inicials, enviar missatges i avaluar els resultats, també es disposa del terminal des del que s'arrenca el JADE on s'imprimirà informació detallada de tot el procés: missatges complerts, dades recuperades del contingut, evolució dels algoritmes, resultats... Aquesta última possibilitat resulta molt útil per *debugar*.

9.2. El fitxer de dades

Per poder introduir les dades dels pacients i del donant que volem avaluar, haurem d'utilitzar un fitxer amb una estructura especial per a que l'agent Especialista les pugui interpretar. Concretament es tracta d'un fitxer de *Properties*, que emmagatzema parells de 'clau = valor' per a cada atribut d'un individu. Les regles que s'han de complir per afegir dades són:

- Sempre haurà de començar amb un parell de claus que l'indiquin el nombre de pacients totals que es volen avaluar i l'òrgan que tenen en comú pacients i donant.

```
#DADES GLOBALS  
Numero_pacients=20  
Organ=cor
```

- A continuació hi hauran d'haver les dades referents al donant. Per a cada informació (hospital, pes, tamany...), haurem d'escriure el nom de la propietat seguit d'un signe "_" i el número "0" que indica que les dades pertanyen al donant.

```
#DONANT  
  
Hospital_0=Joan XXIII  
Localitat_0=Tarragona  
Provincia_0=Tarragona  
Tipus_sang_0=A  
Pes_persona_0=70.4  
Tamany_organ_0=34.2  
Antigens_B0_0=false  
Antigens_B1_0=false  
Antigens_DR0_0=true  
Antigens_DR1_0=false
```

```
Data_naix_dia_0=10
Data_naix_mes_0=10
Data_naix_any_0=1970
```

- Finalment haurem d'introduir les dades de cada pacient seguint el mateix esquema que en el pas anterior però substituint el número '0' a cada dada per l'identificador del pacient (hauran de ser valors consecutius entre 1 i Numero_pacients). En aquest cas, a més dels atributs inclosos amb el donant, s'haurà d'especificar també la data d'entrada a la llista d'espera.

```
#PACIENT 1
```

```
Hospital_1=Joan XIII
Localitat_1=Tarragona
Provincia_1=Tarragona
Tipus_sang_1=A
Pes_persona_1=60.8
Tamany_organ_1=30.0
Antigens_B0_1=true
Antigens_B1_1=true
Antigens_DR0_1=true
Antigens_DR1_1=false
Data_naix_dia_1=1
Data_naix_mes_1=1
Data_naix_any_1=1976
Data_esp_dia_1=10
Data_esp_mes_1=10
Data_esp_any_1=1999
```

9.3. El fitxer de resultats

Com ja s'ha explicat, l'Especialista va emmagatzemant informació respecte a les valoracions que rep cada agent que s'utilitza per tal de realitzar recomanacions en passos següents. Les dades les guarda a un fitxer també de tipus *properties*, que pot ser consultat i interpretat fàcilment.

Per a cada agent que s'hagi puntuat, es guardaran 5 claus: cadascuna d'elles correspon a un terme de valoració seguint la següent escala:

```
0 -> molt_bo
1 -> bo
2 -> acceptable
3 -> regular
4 -> dolent
```

Cada clau estarà formada pel nom de l'agent i el tipus de semàntica que fa servir, juntament amb l'identificador del terme del vocabulari de valoració a que es refereix. El valor que es guarda és el nombre de vegades que s'ha puntuat amb cada terme.

```
#Valoracions dels resultats obtinguts pels agents MCDM
#Sat Aug 18 11:53:59 CEST 2001
```

```
clusdm1/Negation_Functions_4=0  
clusdm1/Negation_Functions_3=0  
clusdm1/Negation_Functions_2=1  
clusdm1/Negation_Functions_1=1  
clusdm1/Negation_Functions_0=3
```

9.4. Constants

També s'inclouen un conjunt de constants que regulen els rangs de les estructures i que s'han de respectar per tal que el procés pugui realitzar-se correctament:

- **final static int MAX_PAC=100:** nombre màxim de pacients des del punt de vista de l'especialista.
- **final static int MAX_OBJ=100:** nombre màxim d'objectes (pacients) des del punt de vista del ClusDM.
- **final static int MAX_ATRIB=10:** nombre màxim d'atributs per cada alternativa.
- **final static int MAX_VAL_ATRIB=10:** nombre màxim de termes del vocabulari d'un atribut.
- **final static int MAX_NEG_VAL_ATRIB=5:** nombre màxim de negacions per a un terme del vocabulari d'un atribut.
- **final static int MAX_AGENTS=10:** nombre màxim d'agents MCDM que poden manegar-se alhora.

10. Avaluació del sistema

A continuació es descriu l'evolució d'un exemple complet (incloent els missatges enviats entre els agents i el seu contingut), així com un conjunt de proves realitzades per comprovar el correcte funcionament del sistema.

10.1. Seguiment d'un exemple

Per tal d'aconseguir un cas el més general possible s'han introduït un conjunt de dades inicials sobre pacients que intenten abarcar el major nombre de variacions.

Contingut del fitxer de pacients/donant

```
#Fitxer d'introduccio de dades referents al donant i els pacients
#Cada individu te un identificador numeric
#Sempre es prendra com a donant aquell qui tingui d'identificador 0
(NomAtribut_0)
#Els pacients s'avaluen segons el seu identificador entre 1 i
N_Pacients
#Els resultats es retornen juntament amb l'identificador corresponent
#Tots els pacients i donant corresponen a un tipus d'organ
(especificat a l'inici)
```

```
#DADES GLOBALS
Numero_pacients=20
Organ=cor
```

```
#DONANT
```

```
Hospital_0=Hospital Joan XXIII
Localitat_0=Tarragona
Provincia_0=Tarragona
Tipus_sang_0=A
Pes_persona_0=70.4
Tamany_organ_0=34.2
Antigens_B0_0=false
Antigens_B1_0=false
Antigens_DR0_0=true
Antigens_DR1_0=false
Data_naix_dia_0=10
Data_naix_mes_0=10
Data_naix_any_0=1960
```

```
#PACIENT 1
```

```
Hospital_1=Clinica Universitaria de Navarra
Localitat_1=Pamplona
Provincia_1=Navarra
Tipus_sang_1=A
Pes_persona_1=60.8
Tamany_organ_1=30.0
Antigens_B0_1=true
Antigens_B1_1=true
Antigens_DR0_1=true
```

Antigens_DR1_1=false
Data_naix_dia_1=1
Data_naix_mes_1=1
Data_naix_any_1=1956
Data_esp_dia_1=4
Data_esp_mes_1=11
Data_esp_any_1=1999

#PACIENT 2

Hospital_2=Hospital Vall d'Hebron
Localitat_2=Barcelona
Provincia_2=Barcelona
Tipus_sang_2=AB
Pes_persona_2=65.8
Tamany_organ_2=31.2
Antigens_B0_2=true
Antigens_B1_2=true
Antigens_DR0_2=true
Antigens_DR1_2=true
Data_naix_dia_2=2
Data_naix_mes_2=1
Data_naix_any_2=1970
Data_esp_dia_2=12
Data_esp_mes_2=5
Data_esp_any_2=2001

#PACIENT 3

Hospital_3=Hospital Ramon y Cajal
Localitat_3=Madrid
Provincia_3=Madrid
Tipus_sang_3=A
Pes_persona_3=75.0
Tamany_organ_3=39.64
Antigens_B0_3=false
Antigens_B1_3=false
Antigens_DR0_3=true
Antigens_DR1_3=false
Data_naix_dia_3=4
Data_naix_mes_3=7
Data_naix_any_3=1977
Data_esp_dia_3=9
Data_esp_mes_3=8
Data_esp_any_3=2000

#PACIENT 4

Hospital_4=Hospital de Sagunto
Localitat_4=Valencia
Provincia_4=Valencia
Tipus_sang_4=AB
Pes_persona_4=65.9
Tamany_organ_4=37.1
Antigens_B0_4=true
Antigens_B1_4=true
Antigens_DR0_4=true
Antigens_DR1_4=false
Data_naix_dia_4=1
Data_naix_mes_4=2

Data_naix_any_4=1945
Data_esp_dia_4=12
Data_esp_mes_4=5
Data_esp_any_4=2001

#PACIENT 5

Hospital_5=Hospital Universitario de Valme
Localitat_5=Sevilla
Provincia_5=Sevilla
Tipus_sang_5=A
Pes_persona_5=77.2
Tamany_organ_5=40.3
Antigens_B0_5=false
Antigens_B1_5=false
Antigens_DR0_5=true
Antigens_DR1_5=false
Data_naix_dia_5=23
Data_naix_mes_5=9
Data_naix_any_5=1950
Data_esp_dia_5=31
Data_esp_mes_5=12
Data_esp_any_5=1999

#PACIENT 6

Hospital_6=Hospital de Cabueñes
Localitat_6=Gijon
Provincia_6=Asturias
Tipus_sang_6=AB
Pes_persona_6=80.0
Tamany_organ_6=39.1
Antigens_B0_6=true
Antigens_B1_6=true
Antigens_DR0_6=true
Antigens_DR1_6=true
Data_naix_dia_6=4
Data_naix_mes_6=8
Data_naix_any_6=1965
Data_esp_dia_6=1
Data_esp_mes_6=1
Data_esp_any_6=2001

#PACIENT 7

Hospital_7=Hospital de Laredo
Localitat_7=Laredo
Provincia_7=Cantabria
Tipus_sang_7=A
Pes_persona_7=78.0
Tamany_organ_7=28.3
Antigens_B0_7=true
Antigens_B1_7=false
Antigens_DR0_7=true
Antigens_DR1_7=false
Data_naix_dia_7=1
Data_naix_mes_7=1
Data_naix_any_7=1976
Data_esp_dia_7=1
Data_esp_mes_7=12

Data_esp_any_7=2000

#PACIENT 8

Hospital_8=Hospital Universitario La Fe
Localitat_8=Valencia
Provincia_8=Valencia
Tipus_sang_8=AB
Pes_persona_8=65.0
Tamany_organ_8=37.0
Antigens_B0_8=true
Antigens_B1_8=false
Antigens_DR0_8=true
Antigens_DR1_8=true
Data_naix_dia_8=2
Data_naix_mes_8=1
Data_naix_any_8=1975
Data_esp_dia_8=12
Data_esp_mes_8=5
Data_esp_any_8=2001

#PACIENT 9

Hospital_9=Hospital Joan XXIII
Localitat_9=Tarragona
Provincia_9=Tarragona
Tipus_sang_9=A
Pes_persona_9=70.0
Tamany_organ_9=35.4
Antigens_B0_9=false
Antigens_B1_9=false
Antigens_DR0_9=true
Antigens_DR1_9=false
Data_naix_dia_9=9
Data_naix_mes_9=9
Data_naix_any_9=1961
Data_esp_dia_9=10
Data_esp_mes_9=12
Data_esp_any_9=1999

#PACIENT 10

Hospital_10=Hospital Sierrallana
Localitat_10=Torrelavega
Provincia_10=Cantabria
Tipus_sang_10=AB
Pes_persona_10=65.4
Tamany_organ_10=29.2
Antigens_B0_10=true
Antigens_B1_10=true
Antigens_DR0_10=false
Antigens_DR1_10=true
Data_naix_dia_10=2
Data_naix_mes_10=1
Data_naix_any_10=1975
Data_esp_dia_10=12
Data_esp_mes_10=5
Data_esp_any_10=2000

#PACIENT 11

Hospital_11=Hospital Universitario La Paz
Localitat_11=Madrid
Provincia_11=Madrid
Tipus_sang_11=0
Pes_persona_11=60.8
Tamany_organ_11=30.0
Antigens_B0_11=true
Antigens_B1_11=true
Antigens_DR0_11=true
Antigens_DR1_11=true
Data_naix_dia_11=17
Data_naix_mes_11=9
Data_naix_any_11=1967
Data_esp_dia_11=17
Data_esp_mes_11=9
Data_esp_any_11=2000

#PACIENT 12

Hospital_12=Hospital Universitario de Canarias
Localitat_12=La Laguna
Provincia_12=Tenerife
Tipus_sang_12=AB
Pes_persona_12=100
Tamany_organ_12=40.2
Antigens_B0_12=false
Antigens_B1_12=true
Antigens_DR0_12=true
Antigens_DR1_12=true
Data_naix_dia_12=12
Data_naix_mes_12=8
Data_naix_any_12=1974
Data_esp_dia_12=13
Data_esp_mes_12=7
Data_esp_any_12=2001

#PACIENT 13

Hospital_13=Hospital Marques de Valdecilla
Localitat_13=Santander
Provincia_13=Cantabria
Tipus_sang_13=A
Pes_persona_13=80.0
Tamany_organ_13=50
Antigens_B0_13=false
Antigens_B1_13=false
Antigens_DR0_13=true
Antigens_DR1_13=true
Data_naix_dia_13=18
Data_naix_mes_13=9
Data_naix_any_13=1952
Data_esp_dia_13=19
Data_esp_mes_13=5
Data_esp_any_13=2000

#PACIENT 14

Hospital_14=Hospital Obispo Polanco
Localitat_14=Teruel

Provincia_14=Teruel
Tipus_sang_14=AB
Pes_persona_14=65.9
Tamany_organ_14=37.1
Antigens_B0_14=true
Antigens_B1_14=true
Antigens_DR0_14=false
Antigens_DR1_14=false
Data_naix_dia_14=1
Data_naix_mes_14=2
Data_naix_any_14=1990
Data_esp_dia_14=22
Data_esp_mes_14=4
Data_esp_any_14=2001

#PACIENT 15

Hospital_15=Hospital de Cruces
Localitat_15=Baracaldo
Provincia_15=Vizcaya
Tipus_sang_15=A
Pes_persona_15=64.2
Tamany_organ_15=35.3
Antigens_B0_15=true
Antigens_B1_15=false
Antigens_DR0_15=true
Antigens_DR1_15=false
Data_naix_dia_15=23
Data_naix_mes_15=9
Data_naix_any_15=1935
Data_esp_dia_15=30
Data_esp_mes_15=6
Data_esp_any_15=1999

#PACIENT 16

Hospital_16=Hospital General de Vic
Localitat_16=Vic
Provincia_16=Barcelona
Tipus_sang_16=AB
Pes_persona_16=60.0
Tamany_organ_16=36.1
Antigens_B0_16=true
Antigens_B1_16=true
Antigens_DR0_16=true
Antigens_DR1_16=false
Data_naix_dia_16=4
Data_naix_mes_16=8
Data_naix_any_16=1965
Data_esp_dia_16=11
Data_esp_mes_16=1
Data_esp_any_16=2001

#PACIENT 17

Hospital_17=Hospital Universitario son Dureta
Localitat_17=Palma de Mallorca
Provincia_17=Baleares
Tipus_sang_17=A
Pes_persona_17=58.0

Tamany_organ_17=33.3
Antigens_B0_17=true
Antigens_B1_17=false
Antigens_DR0_17=true
Antigens_DR1_17=true
Data_naix_dia_17=1
Data_naix_mes_17=11
Data_naix_any_17=1966
Data_esp_dia_17=14
Data_esp_mes_17=5
Data_esp_any_17=2000

#PACIENT 18

Hospital_18=Hospital San Sebastian
Localitat_18=Ecija
Provincia_18=Sevilla
Tipus_sang_18=AB
Pes_persona_18=81.0
Tamany_organ_18=37.0
Antigens_B0_18=false
Antigens_B1_18=false
Antigens_DR0_18=false
Antigens_DR1_18=true
Data_naix_dia_18=2
Data_naix_mes_18=12
Data_naix_any_18=1955
Data_esp_dia_18=1
Data_esp_mes_18=7
Data_esp_any_18=2001

#PACIENT 19

Hospital_19=Hospital San Agustin
Localitat_19=Aviles
Provincia_19=Asturias
Tipus_sang_19=A
Pes_persona_19=90.1
Tamany_organ_19=43.4
Antigens_B0_19=true
Antigens_B1_19=true
Antigens_DR0_19=false
Antigens_DR1_19=true
Data_naix_dia_19=9
Data_naix_mes_19=9
Data_naix_any_19=1943
Data_esp_dia_19=15
Data_esp_mes_19=8
Data_esp_any_19=2001

#PACIENT 20

Hospital_20=Hospital Vall d'Hebron
Localitat_20=Barcelona
Provincia_20=Barcelona
Tipus_sang_20=AB
Pes_persona_20=63.4
Tamany_organ_20=39.2
Antigens_B0_20=true
Antigens_B1_20=true

```
Antigens_DR0_20=true  
Antigens_DR1_20=false  
Data_naix_dia_20=2  
Data_naix_mes_20=1  
Data_naix_any_20=1969  
Data_esp_dia_20=12  
Data_esp_mes_20=4  
Data_esp_any_20=2001
```

Observant les dades anteriors es poden destacar els següents casos:

- S'ha definit el pacient número 9 com el receptor ideal per a l'òrgan del donant. Tots els seus atributs obtindran la màxima qualificació durant l'avaluació que realitzi l'Especialista. Així doncs, sota totes les circumstàncies, aquest pacient hauria d'obtenir el terme "òptim" després del procés d'ordenació.
- Oposadament a aquest, el pacient número 19 és el que està més allunyat del donant en tots els atributs (tot i que compleix les restriccions que imposa l'Especialista sobre la viabilitat del trasplantament). D'aquesta forma, aquest hauria d'obtenir sempre la qualificació de "pèssim" després del procés d'ordenació.
- S'han definit un conjunt de pacients que violen les regles sobre alguns dels criteris imposats per l'Especialista. Això fa que el trasplantament sigui impossible i els individus es desestimin. El conjunt de pacient definits són els següents:
 - o Pacient número 11: la seva sang és incompatible amb la del donant (veure taula ABO).
 - o Pacient número 12: la diferència entre el seu pes i el del donant és massa gran.
 - o Pacient número 13: la diferència entre el tamany del seu òrgan i el del donant és massa gran.
 - o Pacient número 14: la diferència d'edat vers el donant és excessiva.
 - o Pacient número 15: tot i que la diferència entre la seva edat i la del donant estigui dintre dels rangs definits, l'individu és massa gran per rebre l'òrgan en qüestió (cor).
- La resta de pacients s'han definit amb valors a l'atzar però que intenten avarcar totes les possibilitats.

Segons aquestes característiques, en totes les proves els pacients 9 i 19 haurien d'ocupar sempre la mateixa posició (òptim i pèssim respectivament), els pacients de 11 a 15 no haurien d'entrar en el procés d'ordenació (doncs no compleixen les restriccions) i la resta aniran variant lleugerament de qualificació final segons quins siguin els pesos dels atributs que es defineixin i la funció de distància utilitzada.

Un cop definides les dades inicials al fitxer, arrencarem el sistema amb la següent comanda:

```
java jade.Boot -platform -gui clusdm1:ClusDMResponderAgent  
dummy1:DummyNegationAgent dummy2:DummyFuzzyAgent
```

```
dummy3:Dummy2TupleAgent dummy4:DummyOrdinalAgent  
requester:MCDMRequesterAgent
```

Quan s'hagi iniciat l'interfície gràfica, haurem d'escollir l'agent amb el qual ens volem comunicar. En aquest cas, només disposem d'un amb funcionalitat completa (el que implementa l'algoritme de *clustering*), ja que la resta només són prototipus.

Deixant els valors per defecte, iniciarem un **request** sobre l'agent. El primer pas que es produirà serà l'avaluació de les dades per part de l'Especialista en trasplantaments, que les llegirà de fitxer. Els resultats que obté (mostrats al terminal) després del procés de filtratge i avaluació de les dades són els següents (només depenen de les dades del fitxer, no de les variables –pesos i distància-):

```
Sang del pacient 1 compatible  
L'organ i l'edat del pacient 1 son compatibles  
Pacient 1 valid  
Atributs del pacient: 1  
    Distancia: pais  
    Pes: bona  
    Tamany: bona  
    Antigens: dolent  
    Edat: iguals  
    Espera: molt_llarg
```

```
Sang del pacient 2 compatible  
L'organ i l'edat del pacient 2 son compatibles  
Pacient 2 valid  
Atributs del pacient: 2  
    Distancia: zona  
    Pes: bona  
    Tamany: bona  
    Antigens: dolent  
    Edat: mes_de_8  
    Espera: moderat
```

```
Sang del pacient 3 compatible  
L'organ i l'edat del pacient 3 son compatibles  
Pacient 3 valid  
Atributs del pacient: 3  
    Distancia: pais  
    Pes: bona  
    Tamany: bona  
    Antigens: correcte  
    Edat: mes_de_14  
    Espera: llarg
```

```
Sang del pacient 4 compatible  
L'organ i l'edat del pacient 4 son compatibles  
Pacient 4 valid  
Atributs del pacient: 4  
    Distancia: pais  
    Pes: bona  
    Tamany: bona  
    Antigens: dolent  
    Edat: mes_de_14  
    Espera: moderat
```

Sang del pacient 5 compatible
L'organ i l'edat del pacient 5 son compatibles
Pacient 5 valid
Atributs del pacient: 5
 Distancia: pais
 Pes: bona
 Tamany: bona
 Antigens: correcte
 Edat: mes_de_8
 Espera: molt_llarg

Sang del pacient 6 compatible
L'organ i l'edat del pacient 6 son compatibles
Pacient 6 valid
Atributs del pacient: 6
 Distancia: pais
 Pes: bona
 Tamany: bona
 Antigens: dolent
 Edat: iguals
 Espera: normal

Sang del pacient 7 compatible
L'organ i l'edat del pacient 7 son compatibles
Pacient 7 valid
Atributs del pacient: 7
 Distancia: pais
 Pes: bona
 Tamany: fluixa
 Antigens: adequat
 Edat: mes_de_14
 Espera: normal

Sang del pacient 8 compatible
L'organ i l'edat del pacient 8 son compatibles
Pacient 8 valid
Atributs del pacient: 8
 Distancia: pais
 Pes: bona
 Tamany: bona
 Antigens: dolent
 Edat: mes_de_14
 Espera: moderat

Sang del pacient 9 compatible
L'organ i l'edat del pacient 9 son compatibles
Pacient 9 valid
Atributs del pacient: 9
 Distancia: hospital
 Pes: ideal
 Tamany: ideal
 Antigens: correcte
 Edat: iguals
 Espera: molt_llarg

Sang del pacient 10 compatible
L'organ i l'edat del pacient 10 son compatibles
Pacient 10 valid
Atributs del pacient: 10
 Distancia: pais

Pes: bona
Tamany: factible
Antigens: dolent
Edat: mes_de_14
Espera: llarg

Sang del pacient 11 incompatible

Sang del pacient 12 compatible
L'organ i l'edat del pacient 12 son compatibles
Pes del pacient 12 fora de rang

Sang del pacient 13 compatible
L'organ i l'edat del pacient 13 son compatibles
Tamany de l'organ del pacient 13 fora de rang

Sang del pacient 14 compatible
L'organ i l'edat del pacient 14 son compatibles
La diferencia de edat del pacient 14 es massa gran

Sang del pacient 15 compatible
L'organ i l'edat del pacient 15 no son compatibles

Sang del pacient 16 compatible
L'organ i l'edat del pacient 16 son compatibles
Pacient 16 valid
Atributs del pacient: 16

Distancia: zona
Pes: factible
Tamany: bona
Antigens: dolent
Edat: iguals
Espera: normal

Sang del pacient 17 compatible
L'organ i l'edat del pacient 17 son compatibles
Pacient 17 valid
Atributs del pacient: 17

Distancia: zona
Pes: fluixa
Tamany: ideal
Antigens: dolent
Edat: mes_de_5
Espera: llarg

Sang del pacient 18 compatible
L'organ i l'edat del pacient 18 son compatibles
Pacient 18 valid
Atributs del pacient: 18

Distancia: pais
Pes: bona
Tamany: bona
Antigens: dolent
Edat: iguals
Espera: moderat

Sang del pacient 19 compatible
L'organ i l'edat del pacient 19 son compatibles
Pacient 19 valid
Atributs del pacient: 19

Distancia: pais
Pes: fluixa
Tamany: fluixa
Antigens: dolent
Edat: mes_de_14
Espera: recent

Sang del pacient 20 compatible
L'organ i l'edat del pacient 20 son compatibles
Pacient 20 valid
Atributs del pacient: 20

Distancia: zona
Pes: bona
Tamany: bona
Antigens: dolent
Edat: mes_de_8
Espera: moderat

Nombre de pacients valids: 15

Com es pot observar, es compleix el que s'havia definit a l'inici:

- El pacient 9 obté la màxim puntuació en tots els atributs.
- El pacient 19 és qui té la puntuació més baixa en tots els criteris.
- Els pacient 11 a 15 es desestimen per no complir alguna de les restriccions.

Aquest pas serà comú a totes les proves que es realitzin, ja que sempre partirem del mateix fitxer de dades.

La següent etapa del procés serà l'enviament de les dades que ha processat l'Especialista a l'agent MCDM que ha escollit l'usuari juntament amb els paràmetres de configuració (pesos i funció de distància). Tal com es defineix al protocol utilitzat, el primer missatge serà un **request** en el qual s'adjuntarà l'acció a realitzar (en aquest cas només pot ser una) i el conjunt de dades necessàries per dur-la a terme. El missatge enviat serà molt gran, doncs caldrà adjuntar totes les dades de tots els pacients així com la informació referent als atributs i les funcions d'avaluació (conjunts difusos i funcions de negació). El missatge corresponent a les dades descrites anteriorment és el següent:

```
((action (agent-identifier :name clusdm1 :addresses (sequence )
:resolvers (sequence ) ) (MCDMRequest :atributs (set (atribut
:vocabulari (set (terme :valor-terme pais :negacions (set (negacio
:valor-neg provincia ) (negacio :valor-neg ciutat ) (negacio :valor-
neg hospital ) (negacio :valor-neg null ) ) :valors-difusos (set
(triangle-difus :a-num 0 :a-den 1 :b-num 0 :b-den 1 :c-num 3 :c-den 7
) ) ) (terme :valor-terme zona :negacions (set (negacio :valor-neg
zona ) (negacio :valor-neg null ) ) :valors-difusos (set (triangle-
difus :a-num 3 :a-den 7 :b-num 1 :b-den 7 :c-num 4 :c-den 7 ) ) )
(terme :valor-terme provincia :negacions (set (negacio :valor-neg pais
) (negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus
:a-num 4 :a-den 7 :b-num 9 :b-den 7 :c-num 5 :c-den 7 ) ) ) (terme
:valor-terme ciutat :negacions (set (negacio :valor-neg pais )
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-
num 5 :a-den 7 :b-num 11 :b-den 7 :c-num 6 :c-den 7 ) ) ) (terme
```

```
:valor-terme hospital :negacions (set (negacio :valor-neg pais )
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-
num 6 :a-den 7 :b-num 1 :b-den 7 :c-num 1 :c-den 1 ) ) ) (terme
:valor-terme null :negacions (set (negacio :valor-neg null ) )
:valors-difusos (set ) ) :pes 17 ) (atribut :vocabulari (set (terme
:valor-terme fluixa :negacions (set (negacio :valor-neg ideal )
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-
num 0 :a-den 1 :b-num 0 :b-den 1 :c-num 1 :c-den 5 ) ) ) (terme
:valor-terme factible :negacions (set (negacio :valor-neg bona )
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-
num 1 :a-den 5 :b-num 3 :b-den 5 :c-num 2 :c-den 5 ) ) ) (terme
:valor-terme bona :negacions (set (negacio :valor-neg bona ) (negacio
:valor-neg factible ) (negacio :valor-neg null ) ) :valors-difusos
(set (triangle-difus :a-num 2 :a-den 5 :b-num 3 :b-den 5 :c-num 4 :c-
den 5 ) ) ) (terme :valor-terme ideal :negacions (set (negacio :valor-
neg fluixa ) (negacio :valor-neg null ) ) :valors-difusos (set
(triangle-difus :a-num 4 :a-den 5 :b-num 1 :b-den 5 :c-num 1 :c-den 1
) ) ) (terme :valor-terme null :negacions (set (negacio :valor-neg
null ) ) :valors-difusos (set ) ) ) :pes 17 ) (atribut :vocabulari
(set (terme :valor-terme fluixa :negacions (set (negacio :valor-neg
ideal ) (negacio :valor-neg null ) ) :valors-difusos (set (triangle-
difus :a-num 0 :a-den 1 :b-num 0 :b-den 1 :c-num 1 :c-den 5 ) ) )
(terme :valor-terme factible :negacions (set (negacio :valor-neg bona
) (negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus
:a-num 1 :a-den 5 :b-num 3 :b-den 5 :c-num 2 :c-den 5 ) ) ) (terme
:valor-terme bona :negacions (set (negacio :valor-neg bona ) (negacio
:valor-neg factible ) (negacio :valor-neg null ) ) :valors-difusos
(set (triangle-difus :a-num 2 :a-den 5 :b-num 3 :b-den 5 :c-num 4 :c-
den 5 ) ) ) (terme :valor-terme ideal :negacions (set (negacio :valor-
neg fluixa ) (negacio :valor-neg null ) ) :valors-difusos (set
(triangle-difus :a-num 4 :a-den 5 :b-num 1 :b-den 5 :c-num 1 :c-den 1
) ) ) (terme :valor-terme null :negacions (set (negacio :valor-neg
null ) ) :valors-difusos (set ) ) ) :pes 17 ) (atribut :vocabulari
(set (terme :valor-terme dolent :negacions (set (negacio :valor-neg
correcte ) (negacio :valor-neg null ) ) :valors-difusos (set
(triangle-difus :a-num 0 :a-den 1 :b-num 0 :b-den 1 :c-num 1 :c-den 3
) ) ) (terme :valor-terme adequat :negacions (set (negacio :valor-neg
adequat ) (negacio :valor-neg null ) ) :valors-difusos (set (triangle-
difus :a-num 1 :a-den 3 :b-num 1 :b-den 3 :c-num 2 :c-den 3 ) ) )
(terme :valor-terme correcte :negacions (set (negacio :valor-neg
dolent ) (negacio :valor-neg null ) ) :valors-difusos (set (triangle-
difus :a-num 2 :a-den 3 :b-num 1 :b-den 3 :c-num 1 :c-den 1 ) ) )
(terme :valor-terme null :negacions (set (negacio :valor-neg null ) )
:valors-difusos (set ) ) ) :pes 17 ) (atribut :vocabulari (set (terme
:valor-terme mes_de_20 :negacions (set (negacio :valor-neg iguals )
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-
num 0 :a-den 1 :b-num 0 :b-den 1 :c-num 1 :c-den 9 ) ) ) (terme
:valor-terme mes_de_17 :negacions (set (negacio :valor-neg iguals )
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-
num 1 :a-den 9 :b-num 1 :b-den 9 :c-num 2 :c-den 9 ) ) ) (terme
:valor-terme mes_de_14 :negacions (set (negacio :valor-neg iguals )
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-
num 2 :a-den 9 :b-num 5 :b-den 9 :c-num 3 :c-den 9 ) ) ) (terme
:valor-terme mes_de_11 :negacions (set (negacio :valor-neg mes_de_5 )
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-
num 3 :a-den 9 :b-num 7 :b-den 9 :c-num 4 :c-den 9 ) ) ) (terme
:valor-terme mes_de_8 :negacions (set (negacio :valor-neg mes_de_8 )
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-
num 4 :a-den 9 :b-num 1 :b-den 9 :c-num 5 :c-den 9 ) ) ) (terme
:valor-terme mes_de_5 :negacions (set (negacio :valor-neg mes_de_11 )
```

```
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-  
num 5 :a-den 9 :b-num 11 :b-den 9 :c-num 6 :c-den 9 ) ) ) (terme  
:valor-terme iguals :negacions (set (negacio :valor-neg mes_de_14 )  
(negacio :valor-neg mes_de_17 ) (negacio :valor-neg mes_de_20 )  
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-  
num 6 :a-den 9 :b-num 1 :b-den 9 :c-num 1 :c-den 1 ) ) ) (terme  
:valor-terme null :negacions (set (negacio :valor-neg null ) )  
:valors-difusos (set ) ) ) :pes 16 ) (atribut :vocabulari (set (terme  
:valor-terme recent :negacions (set (negacio :valor-neg recent )  
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-  
num 0 :a-den 1 :b-num 0 :b-den 1 :c-num 1 :c-den 5 ) ) ) (terme  
:valor-terme moderat :negacions (set (negacio :valor-neg moderat )  
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-  
num 1 :a-den 5 :b-num 3 :b-den 5 :c-num 2 :c-den 5 ) ) ) (terme  
:valor-terme normal :negacions (set (negacio :valor-neg normal )  
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-  
num 2 :a-den 5 :b-num 1 :b-den 5 :c-num 3 :c-den 5 ) ) ) (terme  
:valor-terme llarg :negacions (set (negacio :valor-neg llarg )  
(negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus :a-  
num 3 :a-den 5 :b-num 7 :b-den 5 :c-num 4 :c-den 5 ) ) ) (terme  
:valor-terme molt_llarg :negacions (set (negacio :valor-neg molt_llarg  
) (negacio :valor-neg null ) ) :valors-difusos (set (triangle-difus  
:a-num 4 :a-den 5 :b-num 1 :b-den 5 :c-num 1 :c-den 1 ) ) ) (terme  
:valor-terme null :negacions (set (negacio :valor-neg null ) )  
:valors-difusos (set ) ) ) :pes 16 ) ) :natributs 6 :objectes (set  
(atrib-objectes :valor-objecte (set (valor-atribut :valor-qualitatiu  
pais ) (valor-atribut :valor-qualitatiu bona ) (valor-atribut :valor-  
qualitatiu bona ) (valor-atribut :valor-qualitatiu dolent ) (valor-  
atribut :valor-qualitatiu iguals ) (valor-atribut :valor-qualitatiu  
molt_llarg ) ) :id-obj 1 ) (atrib-objectes :valor-objecte (set (valor-  
atribut :valor-qualitatiu zona ) (valor-atribut :valor-qualitatiu bona  
) (valor-atribut :valor-qualitatiu bona ) (valor-atribut :valor-  
qualitatiu dolent ) (valor-atribut :valor-qualitatiu mes_de_8 )  
(valor-atribut :valor-qualitatiu moderat ) ) :id-obj 2 ) (atrib-  
objectes :valor-objecte (set (valor-atribut :valor-qualitatiu pais )  
(valor-atribut :valor-qualitatiu bona ) (valor-atribut :valor-  
qualitatiu bona ) (valor-atribut :valor-qualitatiu correcte ) (valor-  
atribut :valor-qualitatiu mes_de_14 ) (valor-atribut :valor-qualitatiu  
llarg ) ) :id-obj 3 ) (atrib-objectes :valor-objecte (set (valor-  
atribut :valor-qualitatiu pais ) (valor-atribut :valor-qualitatiu bona  
) (valor-atribut :valor-qualitatiu bona ) (valor-atribut :valor-  
qualitatiu dolent ) (valor-atribut :valor-qualitatiu mes_de_14 )  
(valor-atribut :valor-qualitatiu moderat ) ) :id-obj 4 ) (atrib-  
objectes :valor-objecte (set (valor-atribut :valor-qualitatiu pais )  
(valor-atribut :valor-qualitatiu bona ) (valor-atribut :valor-  
qualitatiu bona ) (valor-atribut :valor-qualitatiu correcte ) (valor-  
atribut :valor-qualitatiu mes_de_8 ) (valor-atribut :valor-qualitatiu  
molt_llarg ) ) :id-obj 5 ) (atrib-objectes :valor-objecte (set (valor-  
atribut :valor-qualitatiu pais ) (valor-atribut :valor-qualitatiu bona  
) (valor-atribut :valor-qualitatiu bona ) (valor-atribut :valor-  
qualitatiu dolent ) (valor-atribut :valor-qualitatiu iguals ) (valor-  
atribut :valor-qualitatiu normal ) ) :id-obj 6 ) (atrib-objectes  
:valor-objecte (set (valor-atribut :valor-qualitatiu pais ) (valor-  
atribut :valor-qualitatiu bona ) (valor-atribut :valor-qualitatiu  
fluixa ) (valor-atribut :valor-qualitatiu adequat ) (valor-atribut  
:valor-qualitatiu mes_de_14 ) (valor-atribut :valor-qualitatiu normal  
) ) :id-obj 7 ) (atrib-objectes :valor-objecte (set (valor-atribut  
:valor-qualitatiu pais ) (valor-atribut :valor-qualitatiu bona )  
(valor-atribut :valor-qualitatiu bona ) (valor-atribut :valor-  
qualitatiu dolent ) (valor-atribut :valor-qualitatiu mes_de_14 )
```

```
(valor-atribut :valor-qualitatiu moderat ) ) :id-obj 8 ) (atrib-  
objectes :valor-objecte (set (valor-atribut :valor-qualitatiu hospital  
) (valor-atribut :valor-qualitatiu ideal ) (valor-atribut :valor-  
qualitatiu ideal ) (valor-atribut :valor-qualitatiu correcte ) (valor-  
atribut :valor-qualitatiu iguals ) (valor-atribut :valor-qualitatiu  
molt_llarg ) ) :id-obj 9 ) (atrib-objectes :valor-objecte (set (valor-  
atribut :valor-qualitatiu pais ) (valor-atribut :valor-qualitatiu bona  
) (valor-atribut :valor-qualitatiu factible ) (valor-atribut :valor-  
qualitatiu dolent ) (valor-atribut :valor-qualitatiu mes_de_14 )  
(valor-atribut :valor-qualitatiu llarg ) ) :id-obj 10 ) (atrib-  
objectes :valor-objecte (set (valor-atribut :valor-qualitatiu zona )  
(valor-atribut :valor-qualitatiu factible ) (valor-atribut :valor-  
qualitatiu bona ) (valor-atribut :valor-qualitatiu dolent ) (valor-  
atribut :valor-qualitatiu iguals ) (valor-atribut :valor-qualitatiu  
normal ) ) :id-obj 16 ) (atrib-objectes :valor-objecte (set (valor-  
atribut :valor-qualitatiu zona ) (valor-atribut :valor-qualitatiu  
fluixa ) (valor-atribut :valor-qualitatiu ideal ) (valor-atribut  
:valor-qualitatiu dolent ) (valor-atribut :valor-qualitatiu mes_de_5 )  
(valor-atribut :valor-qualitatiu llarg ) ) :id-obj 17 ) (atrib-  
objectes :valor-objecte (set (valor-atribut :valor-qualitatiu pais )  
(valor-atribut :valor-qualitatiu bona ) (valor-atribut :valor-  
qualitatiu bona ) (valor-atribut :valor-qualitatiu dolent ) (valor-  
atribut :valor-qualitatiu iguals ) (valor-atribut :valor-qualitatiu  
moderat ) ) :id-obj 18 ) (atrib-objectes :valor-objecte (set (valor-  
atribut :valor-qualitatiu pais ) (valor-atribut :valor-qualitatiu  
fluixa ) (valor-atribut :valor-qualitatiu fluixa ) (valor-atribut  
:valor-qualitatiu dolent ) (valor-atribut :valor-qualitatiu mes_de_14  
) (valor-atribut :valor-qualitatiu recent ) ) :id-obj 19 ) (atrib-  
objectes :valor-objecte (set (valor-atribut :valor-qualitatiu zona )  
(valor-atribut :valor-qualitatiu bona ) (valor-atribut :valor-  
qualitatiu bona ) (valor-atribut :valor-qualitatiu dolent ) (valor-  
atribut :valor-qualitatiu mes_de_8 ) (valor-atribut :valor-qualitatiu  
moderat ) ) :id-obj 20 ) ) :nobjectes 15 :distancia 0 ) ) )
```

La resposta a aquest missatge podrà ser afirmativa o negativa en funció de les dades que s'enviïn. En aquest punt, l'agent receptor avaluarà el valor dels pesos dels atributs (han de sumar 100) i el nombre d'atributs i d'objectes pacient a avaluar. Si tots són correctes, s'enviarà un missatge d'**agree** que indica que començarà a realitzar l'acció. Com a contingut d'aquest missatge s'enviaran les dades inicials de forma que es pugui comprovar que han arribat correctament (obviades en la següent transcripció) i la clausula *true* (per indicar que s'accepta la petició).

```
((action (agent-identifier :name clusdm1 :addresses (sequence )  
:resolvers (sequence ) ) (MCDMRequest :  
.....dades.....) true )
```

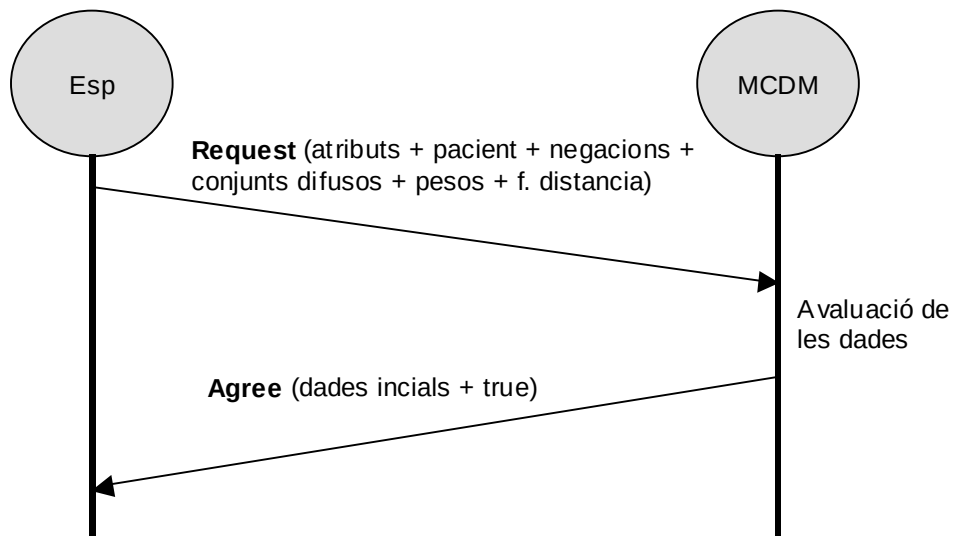


Figura 28 - Acceptació d'una petició de Request

En cas que la resposta sigui negativa, s'enviarà un missatge **refuse** que provocarà la finalització de l'acte comunicatiu. En aquest cas també s'envien les dades inicials però el tipus de missatge en què s'inclouen informa de la causa del refús (tal com s'hagi definit a l'ontologia).

```
((action (agent-identifier :name clusdm1@a:1099/JADE :addresses
(sequence ) :resolvers (sequence ) ) (MCDMRequest :
.....dades.....) Pesos-no-valids )
```

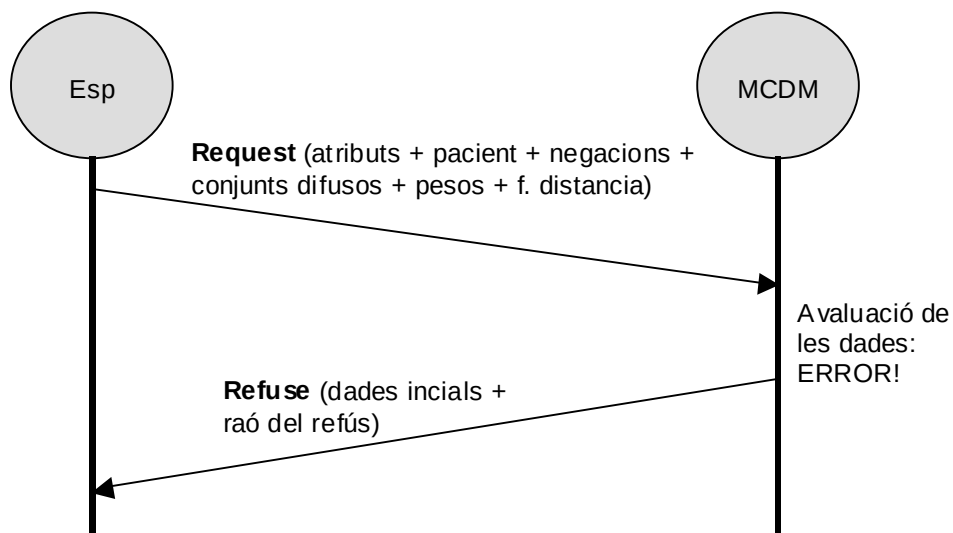


Figura 29 - Refús d'un missatge de request

Si s'ha produït la situació inicial, l'agent MCDM haurà començat a realitzar l'acció demanda. Quan s'hagi completat, s'enviarà un missatge tipus **inform** en el qual s'adjuntarà a les dades inicials el resultat de l'acció. En aquest cas serà una llista amb els parells identificador de pacient + valoració final.

```
((result (action (agent-identifier :name clusdm1 :addresses (sequence ) :resolvers (sequence ) ) (MCDMRequest : .....dades.....) (set (resultat :assignacio (set (grau :valor-grau optim :id-objecte 9 ) (grau :valor-grau bo :id-objecte 5 ) (grau :valor-grau bo :id-objecte 17 ) (grau :valor-grau bo :id-objecte 3 ) (grau :valor-grau adequat :id-objecte 1 ) (grau :valor-grau adequat :id-objecte 18 ) (grau :valor-grau factible :id-objecte 16 ) (grau :valor-grau factible :id-objecte 6 ) (grau :valor-grau desfavorable :id-objecte 20 ) (grau :valor-grau desfavorable :id-objecte 2 ) (grau :valor-grau dolent :id-objecte 8 ) (grau :valor-grau dolent :id-objecte 4 ) (grau :valor-grau dolent :id-objecte 7 ) (grau :valor-grau dolent :id-objecte 10 ) (grau :valor-grau pessim :id-objecte 19 ) ) :nobjectes 15 ) ) ) )
```

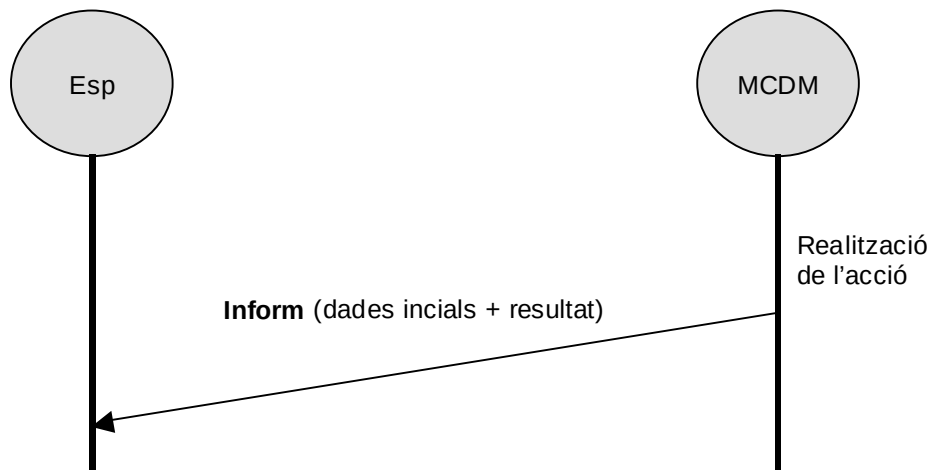


Figura 30 - Finalització correcta de l'acció demanda

Si per qualsevol raó l'acció ha fallat, llavors l'agent enviarà un missatge tipus **failure** adjuntant la raó que ha provocat l'avortament de l'acció.

```
((action (agent-identifier :name clusdm1 :addresses (sequence ) :resolvers (sequence ) ) (MCDMRequest : .....dades..... ) Classes-insuficients )
```

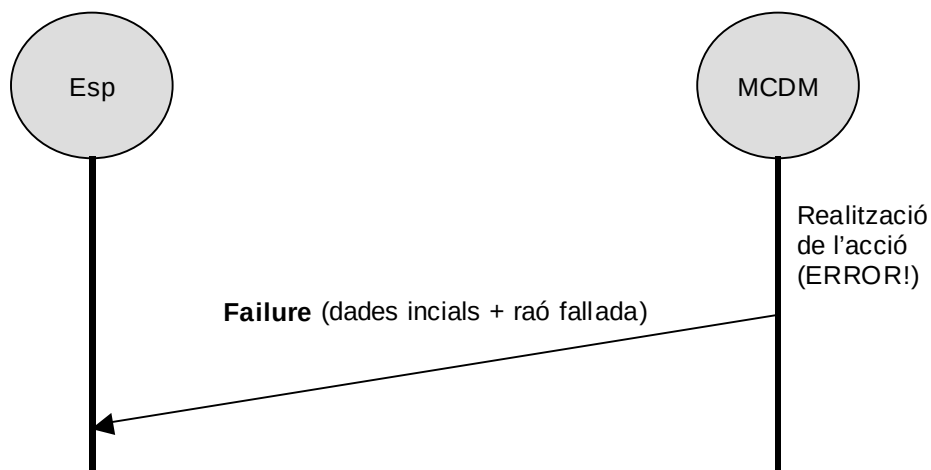


Figura 31 - Avortament de l'acció demanda

10.2. Joc de proves

A continuació es recullen els resultats finals obtinguts al realitzar request sobre l'agent ClusDM partint de les dades descrites anteriorment sota diverses condicions d'execució (variant els valors dels pesos i la funció de distància).

10.2.1. Proves sobre les funcions de distància

Mantenint els mateixos pesos per tots els atributs (de forma que sumen 100), anem variant la funció de distància.

Pesos:

Distància	17 %
Pes	17 %
Tamany	17 %
Antigens	17 %
Edat	16 %
Espera	16 %

Prova 1

Funció de distància: *Diferències*

Resultat:

- Pacient(s) optim : 9
- Pacient(s) bo : 5, 17, 3

- Pacient(s) adequat : 1, 18
- Pacient(s) factible : 16, 6
- Pacient(s) desfavorable : 20, 2
- Pacient(s) dolent : 8, 4, 7, 10
- Pacient(s) pessim : 19

Prova 2

Funció de distància: *M.C.D.*

Resultat:

- Pacient(s) optim : 9
- Pacient(s) bo : 5, 3
- Pacient(s) adequat : 17
- Pacient(s) factible : 18, 6, 1, 16
- Pacient(s) desfavorable : 8, 4, 20, 2
- Pacient(s) dolent : 10, 7
- Pacient(s) pessim : 19

Prova 3

Funció de distància: *Manhattan*

Resultat:

- Pacient(s) optim : 9
- Pacient(s) bo : 5, 3
- Pacient(s) adequat : 17
- Pacient(s) factible : 18, 6, 1, 16
- Pacient(s) desfavorable : 8, 4, 20, 2
- Pacient(s) dolent : 10, 7
- Pacient(s) pessim : 19

Prova 4

Funció de distància: *Camberra*

Resultat:

- Pacient(s) optim : 9
- Pacient(s) bo : 3, 1, 5
- Pacient(s) adequat : 17
- Pacient(s) factible : 7, 16
- Pacient(s) desfavorable : 18, 6, 8, 4, 20, 2
- Pacient(s) dolent : 10
- Pacient(s) pessim : 19

Prova 5

Funció de distància: *Taxonomic*

Resultat:

- Pacient(s) optim : 9
- Pacient(s) bo : 5, 3
- Pacient(s) adequat : 16
- Pacient(s) factible : 8, 4, 1, 20, 2, 18, 6
- Pacient(s) desfavorable : 17
- Pacient(s) dolent : 10, 7
- Pacient(s) pessim : 19

Conclusions

Com es pot observar, els resultats obtinguts per totes les proves tenen un punt en comú: el pacient número 9 sempre obté la màxima valoració i el 19 la mínima, tal com s'havia provocat en la confecció de les dades inicials. Pel que fa a la resta de pacients, alguns mantenen sempre la seva valoració (5, 10, 3, 16...) mentre que la resta varien un nivell a dalt o a baix segons quina sigui la funció escollida.

Això demostra que tot i utilitzar una o altra fórmula per avaluar les distàncies, sempre s'obtingran els mateixos resultats per a aquells elements prou distants dels altres i l'elecció només afectarà a aquells pacients que tinguin característiques molt semblants (que variarant un nivell a dalt o a baix).

Resulta interessant observar que la funció de distància *M.C.D.* i la *Manhattan* obtenen els mateixos resultats. Això és així perquè les dues funcions tenen una fórmula molt semblant (l'una té en compte el nombre de pacient i l'altra no però les distàncies relatives es mantenen).

10.2.2. Proves sobre els pesos

En aquest cas mantindrem la funció de distància fixa (*Manhattan*) i anirem variant els pesos dels atributs, donant major importància a un d'ells i mantenint la resta.

Prova 1

Pesos:

Distància	30 %
Pes	14 %
Tamany	14 %
Antigens	14 %
Edat	14 %

Espera	14 %
--------	------

Resultat:

- Pacient(s) optim : 9
- Pacient(s) bo : 5, 3
- Pacient(s) adequat : 17
- Pacient(s) factible : 20, 2, 16
- Pacient(s) desfavorable : 8, 4, 10, 18, 6, 1
- Pacient(s) dolent : 7
- Pacient(s) pessim : 19

Prova 2

Pesos:

Distància	14 %
Pes	30 %
Tamany	14 %
Antigens	14 %
Edat	14 %
Espera	14 %

Resultat:

- Pacient(s) optim : 9
- Pacient(s) bo : 5, 3
- Pacient(s) adequat : 8, 4, 20, 2, 18, 6, 1
- Pacient(s) factible : 16
- Pacient(s) desfavorable : 17
- Pacient(s) dolent : 10, 7
- Pacient(s) pessim : 19

Prova 3

Pesos:

Distància	14 %
Pes	14 %
Tamany	30 %
Antigens	14 %
Edat	14 %
Espera	14 %

Resultat:

- Pacient(s) optim : 9
- Pacient(s) bo : 5, 3
- Pacient(s) adequat : 17

- Pacient(s) factible : 18, 6, 1, 16
- Pacient(s) desfavorable : 7, 8, 4, 20, 2
- Pacient(s) dolent : 10
- Pacient(s) pessim : 19

Prova 4

Pesos:

Distància	14 %
Pes	14 %
Tamany	14 %
Antigens	30 %
Edat	14 %
Espera	14 %

Resultat:

- Pacient(s) optim : 9
- Pacient(s) bo : 5, 3
- Pacient(s) adequat : 17
- Pacient(s) factible : 18, 6, 1, 16
- Pacient(s) desfavorable : 7, 8, 4, 20, 2
- Pacient(s) dolent : 10
- Pacient(s) pessim : 19

Prova 5

Pesos:

Distància	14 %
Pes	14 %
Tamany	14 %
Antigens	14 %
Edat	30 %
Espera	14 %

Resultat:

- Pacient(s) optim : 9
- Pacient(s) bo : 3, 1, 5
- Pacient(s) adequat : 18, 6, 16
- Pacient(s) factible : 17
- Pacient(s) desfavorable : 20, 2
- Pacient(s) dolent : 8, 4, 10, 7
- Pacient(s) pessim : 19

Prova 6

Pesos:

Distància	14 %
Pes	14 %
Tamany	14 %
Antigens	14 %
Edat	14 %
Espera	30 %

Resultat:

- Pacient(s) optim : 9
- Pacient(s) bo : 17
- Pacient(s) adequat : 5, 7
- Pacient(s) factible : 8, 4, 1, 3
- Pacient(s) desfavorable : 18, 6, 16, 20, 2
- Pacient(s) dolent : 10
- Pacient(s) pessim : 19

Conclusions

Novament es manté la característica que comentàvem al l'inci: el pacients 9 i 19 sempre ocupen la millor i pitjor posició respectivament. Això resulta lògic doncs, per a l'atribut pel qual hem augmentat el pes per cada cas, ells poseeixen la millor i pitjor valoració. Pel que fa a la resta de pacients, les variacions són més significatives, doncs per cada cas influirà bastant més el valor de l'atribut maximitzat que la resta.

10.2.3. Proves extremes

El conjunt de proves següents, pretenen demostrar el funcionament del sistema sota condicions extremes en quant al valor dels pesos

Prova 1

En la següent prova augmentarem el pes d'un atribut fins al valor màxim, deixant la resta a 0. La funció de distància escollida és Manhattan.

Pesos:

Distància	0 %
Pes	0 %
Tamany	0 %
Antigens	0 %
Edat	0 %

Espera	100 %
--------	-------

Resultat:

- Pacient(s) optim : 9, 5, 1
- Pacient(s) adequat : 17, 10, 3
- Pacient(s) factible : 16, 7, 6
- Pacient(s) desfavorable : 20, 18, 8, 4, 2
- Pacient(s) pessim : 19

Observem com en aquest cas, el nombre de termes del vocabulari final s'ha reduït a 5 (el millor nombre de classes obtingudes). Això es deu a que, al tenir en compte només un dels atributs, el procés de *clustering* haurà agrupat a tots els objectes que tinguin la mateixa valoració; com que en aquest cas el nombre de valoracions diferents per a aquest atribut és 5 (nombre de termes del seu vocabulari: *recent*, *moderat*, *llarg*, *normal*, *llarg*), el resultat és una assignació directa entre terme de l'atribut i terme del vocabulari final seguint la següent regla:

Recent => pèssim
Moderat => desfavorable
Normal => factible
Llarg => adequat
Molt_llarg => òptim

Prova 2

En el cas anterior, l'assignació del vocabulari final era possible perquè el vocabulari de l'atribut al que hem donat tota la significació tenia un nombre de termes mínim. Observem ara què passa si donem significació màxima a un atribut que tingui menys de 5 termes (el nombre mínim per al vocabulari final).

Pesos:

Distància	0 %
Pes	100 %
Tamany	0 %
Antigens	0 %
Edat	0 %
Espera	0 %

Resultat:

Iniciant la sollicitud de Request...
Request iniciat...
MCDM acceptat, esperant resultat...
MCDM fallat
La rao de la fallada es: Classes-insuficients

Els missatges anteriors són els que retorna el sistema a través de la pantalla de resultats. Com que l'atribut pes només té 4 termes (*fluixa, factible, bona, ideal*), només s'obtidran 4 classes finals i, per tant, no podrem assignar el vocabulari.

Prova 3

En la següent prova, definirem uns pesos erronis, de forma que la seva suma no sigui 100.

Pesos:

Distància	10 %
Pes	10 %
Tamany	10 %
Antigens	10 %
Edat	10 %
Espera	10 %

Resultat:

```
Iniciant la sollicitud de Request...
Request iniciat...
MCDM refusat
La rao del refus es: Pesos-no-valids
```

A diferència del cas anterior, aquí el procés de *clustering* ni tan sols a començat ja que l'agent receptor del missatge inicial ha avaluat les dades i les ha refusat directament.

10.2.4. Altres proves

A banda dels casos definits anteriorment que afecten al comportament de l'algoritme d'ordenació, també s'han realitzat altres comprovacions que afecten al funcionament en general:

- Realització d'un **request** sobre un agent tipus *dummy* (de qualsevol de les semàntiques definides) -> donat que aquests no implementen funcionalitat (només són un prototipus), retornen una fallada durant el procés d'execució (en aquest cas la que està definida a l'ontologia: classes insuficients).
- Introducció d'una ruta de fitxer incorrecte o deixar-la buida -> es mostra un missatge d'error.
- Valor buit per a alguns dels pesos -> es mostra un missatge d'error.
- Intent de realitzar un **request** sense haver seleccionat un agent MCDM -> es mostra un missatge d'error.

- Definició d'un fitxer de pacients amb menys de 5 objectes -> es produirà un abortament del procés degut a la falta de classes.
- Definició d'un fitxer de pacients amb un nombre superior al màxim definit (només caldrà indicar un número més gran a 100 en la capçalera) -> es refusarà el procés d'ordenació argumentant un nombre d'objectes excessiu.
- El nombre de pacients indicats a la capçalera del fitxer de dades és més petit que el nombre de registres inclosos - > només s'avaluaran els N primers pacients.
- Inici d'un nombre d'agents superior al màxim que s'ha definit en el procés de cerca -> només es podrà accedir als N primers agents trobats.
- Després de fer diverses proves sobre l'agent ClusDM i donar una valoració al resultat, comprovem com en el següents pasos, se'ns informarà de la seu comportament en funció dels valors que hàgim escollit.

11. Codi

11.1. ClusDM.java

11.2. ClusDMResponderAgent.java

11.3. DadesClusDM.java

11.4. DadesEspecialista.java

11.5. Donant.java

11.6. Dummy2TupleAgent.java

11.7. DummyFuzzyAgent.java

11.8. DummyNegationAgent.java

11.9. DummyOrdinalAgent.java

11.10. DummyResponderAgent.java

11.11. Pacient.java

11.12. ResultatMCDM.java

11.13. Especialista.java

11.14. Gui.java

11.15. Info.java

11.16. MCDMRequesterAgent.java

11.17. El paquet Ontology

11.17.1. ontology.MCDMOntology.java

11.17.2. ontology/Atrib_objectes.java

11.17.3. ontology/Atribut.java

11.17.4. ontology/ClassesInsuficients.java

11.17.5. ontology/Domain_OntologyObj.java

11.17.6. ontology/Grau.java

11.17.7. ontology/MassaAtributs.java

11.17.8. ontology/MassaObjectes.java

11.17.9. ontology/MCDMRequest.java

11.17.10. ontology/Negacio.java

11.17.11. ontology/PesosNoValids.java

11.17.12. ontology/Resultat.

11.17.13. ontology/Terme.java

11.17.14. ontology/Triangle_difus.java

11.17.15. ontology/Valor_atribut.java

12. Conclusions i valoració

Els agents suposen un gran avanç en IA. El fet de que diversos individus treballin conjuntament per aconseguir un objectiu els confereix una capacitat de reacció i de decisió força complexes. Una entitat de sí sola no tindria gaire èxit imitant el comportament humà, però la combinació i cooperació d'un grup d'ells amb diversos comportament fa que, globalment, les seves actuacions siguin força intel·ligents (tal com fan algunes comunitats d'insectes). És en aquest punt precisament on el Sistemes Multi-Agent (SMA) demostren com poden ser d'útils per resoldre determinades tasques que individualment serien massa complexes.

El camp dels SMA és una tècnica molt nova dins de la IA i encara està en procés de desenvolupament però, gràcies a organismes internacionals sense ànim de lucre com la FIPA, que defineix estàndards i en promou la utilització, s'aconsegueixen avenços molt ràpidament.

Pel que fa a la realització d'aquest projecte, ha suposat tot un repte, doncs requeria d'un estudi i aprenentatge d'un món completament nou del qual només en l'assignatura d'IA s'havien mostrat algunes pinzellades. Tot i això, gràcies al GruSMA (Grup de SMA) creat per Aïda Valls i Toni Moreno, que m'ha proporcionat la base teòrica i gran quantitat de documentació sobre el tema, les dificultats inicials s'han superat més fàcilment.

A més, el fet de treballar sobre un problema real com és la gestió de trasplantaments d'òrgans, intentant aconseguir un sistema que en millori la seva eficàcia, ha suposat un estímul afegit. En aquest punt, s'ha intentat en tot moment que la manera de treballar amb les dades sigues la més real possible (utilitzant atributs i variables que tindria en compte un Especialista i seguint el model geogràfic definit per la ONT).

Per altra banda també ha estat interessant l'experiència en la implementació de Mètodes d'ajuda a la presa de decisions multicriteri (MCDM), i les tècniques apreses per a l'avaluació de les dades basades en termes qualitius (conjunts difusos i funcions de negociació). En aquest punt ha estat especialment útil l'ajuda prestada per Aïda Valls que realitza una tesis sobre aquest tema.

Com a punt final, voldria agrair l'interès demostrat pels meus tutors Aïda Valls i Toni Moreno i als meus companys de GruSMA per l'aportació dels seus projectes, en especial a David Isern i els seus treballs d'investigació sobre SMA que m'han resultat de gran ajuda.

13. Treballs futurs

Com ja s'explicava a l'inici, l'objectiu d'aquest projecte i del sistema global era molt ambiciós: implementar un sistema que gestioni eficientment el procés de trasplantament d'òrgans. El fet de treballar amb un cas real, fa que les variables i factors a tenir en compte durant la implementació es multipliquin i compliquin el procés. Per tant, la realització d'aquest sistema és un procés llarg que encara requerirà de força temps per completar-se.

Pel que fa a aquest projecte, encara queden força apartats sobre els que treballar en un futur:

- Tot i que s'ha implementat completament un algoritme MCDM, existeixen molts d'altres que caldria avaluar per poder comparar-los sota diferents circumstàncies. En aquest punt, s'han definit una sèrie de semàntiques extres que podrien utilitzar aquests mètodes i els agents que les manegen.
- Quan es disposin d'altres mètodes d'ordenació seria molt interessant realitzar proves exhaustives sobre la conveniència d'uns i altres i dotar a l'Especialista de més criteris de decisió del mètode adequat a través de l'aprenentatge.
- Per poder insertar aquest mòdul dintre del conjunt global serà necessari implementar un agent Coordinador que farà de pont amb el sistema.
- Es podrien millorar i complicar els comportaments dels agents del sistema per tal que puguin atendre a peticions més complexes o donar explicacions sobre els resultats obtinguts.
- Donat que els algoritmes MCDM treballen amb a dades generals, es podria utilitzar aquest sistema per avaluar altres matèries (que es basin en termes qualitius), incorporant un Especialista amb una base de coneixement sobre el tema.
- Aprofitant la capacitat de treball distribuït i de comunicació a través de xarxa dels agents, es podria provar el sistema utilitzant diverses màquines.
- Incorporar més variables i factors a tenir en compte en el procés de decisió aproximant-lo al màxim a un cas real. En aquest punt seria molt recomanable la col·laboració amb un especialista en el tema.

Com es pot veure, la feina a realitzar encara és molt extensa i el fet que es treballi sobre una tecnologia tant nova que encara està en procés de desenvolupament fa que el procés de l'aprenentatge sigui continu i que es replanteixin contínuament les idees i objectius.

14. Bibliografia

- [1] Isern D., Moreno A., Valls A. (2001), *Desenvolupament de sistemes Multi-Agent en JADE*, Report de Recerca, Universitat Rovira i Virgili
- [2] Valls A. (1995), *Sedàs, Mòdul general de classificació de dominis poc estructurats*, Projecte Final de Carrera, Universitat Rovira i Virgili
- [3] Valls A., Moreno A., Alonso H., Sánchez D. (2001), *A Multi-Agent System to Support the Assignment of Transplantable Organs*, Universitat Rovira i Virgili
- [4] Torra V., *Negation functions based semantics for ordered linguistic labels*, Universitat Rovira i Virgili
- [5] Valls A., Torra V. (1998), *Fuzzy sets and systems: Using classification as an aggregation tool in MCDM*, Universitat Rovira i Virgili
- [6] Foundation for Intelligent Physical Agents (1998), *Agent Communication Language*, FIPA
- [7] Foundation for Intelligent Physical Agents (1998), *Agent Management*, FIPA
- [8] Foundation for Intelligent Physical Agents (1998), *FIPA97 Developers Guide*, FIPA
- [9] Foundation for Intelligent Physical Agents (2000), *FIPA SL Content Language Specification*, FIPA
- [10] Foundation for Intelligent Physical Agents (2000), *FIPA Agent Management Specification*, FIPA
- [11] Foundation for Intelligent Physical Agents (2000), *FIPA ACL Message Structure*, FIPA
- [12] Foundation for Intelligent Physical Agents (2001), *FIPA Device Ontology*, FIPA
- [13] Organización Nacional de Trasplantes, *¿Qué es la ONT?*, Ministerio de Sanidad y Consumo
- [14] GruSMA (2000), *Lenguaje de comunicación entre agentes FIPA*, Universitat Rovira i Virgili
- [15] GruSMA (2000), *Especificación FIPA de la Gestión de Agentes*, Universitat Rovira i Virgili
- [16] GruSMA (2000), *Agents i Sistemes Multi-Agents*, Universitat Rovira i Virgili
- [17] GruSMA (2000), *Jade 1.3*, Universitat Rovira i Virgili

- [18] Isern D., (1999), *Avaluació d'entorns de desenvolupament de sistemes Multiagent*, Projecte Final de Carrera, Universitat Rovira i Virgili
- [19] Bocio J. (2000), *Esquips mèdics per a transplantaments*, Projecte Final de Carrera, Universitat Rovira i Virgili
- [20] Ribes A. (2000), *Ruta de transport d'un òrgan*, Projecte Final de Carrera, Universitat Rovira i Virgili
- [21] Isern D. (2000), *Desenvolupament de Sistemes Multiagent: JADE*, Universitat Rovira i Virgili
- [22] Isern D. (2001), *JADE: Conceptes fonamentals*, Universitat Rovira i Virgili