

TAULA DE CONTINGUTS.

1	INTRODUCCIÓ	3
2	AGENTS I SISTEMES MULTIAGENT (SMA)	5
2.1	Propietats dels agents	5
2.2	Tipus d'agents	6
2.3	Arquitectures d'agents	7
2.4	Sistemes Multi-Agent (SMA)	8
2.4.1	Beneficis d'un SMA	8
2.4.2	Especificació FIPA de la gestió d'agents	8
2.4.3	El llenguatge de comunicació entre agents	9
2.4.3.1	Components d'un missatge	10
2.4.3.2	Protocols de comunicació	12
3	JADE	13
3.1	Plataforma d'agents	14
3.2	Agents	14
3.3	Comportaments	15
3.4	Comunicació	15
3.5	Ontologia	16
3.6	Protocols de comunicació	16
3.6.1	FIPA Request	16
3.6.2	Protocol FIPA Query	17
3.6.3	Protocol FIPA Contract Net	18
3.7	Eines de desenvolupament JADE	19
3.7.1	RMA	19
3.7.2	Sniffer	20
3.7.3	Dummy Agent	20
3.7.4	El DF	21
4	DESCRIPCIÓ DEL SISTEMA	22
4.1	Per què un SMA?	23
4.2	Arquitectura del sistema	23
4.3	Comunicació entre agents	27
4.3.1	Protocol	27
4.3.2	Ontologia	28
4.3.2.1	Objectes	28
	L'objecte Person	28
	L'objecte Application	30
	L'objecte Work	31
	L'objecte Applicants	31
4.3.2.2	Les accions	31
	L'acció SendApplication	31
	L'acció NewApplicant	32
	L'acció NewWork	32
	L'acció FindApplicant	32
	L'acció SendData	33
	L'acció AcceptWork	33
	L'acció HaveWorker	34
	L'acció HaveWork	34

5	<i>L'AGENT EMPRESA</i>	39
5.1	Descripció	39
5.2	Ordenació de candidats	40
5.2.1	Filtratge de candidats	40
5.2.2	Obtenció dels termes qualitatius	41
5.2.2.1	L'atribut qualificació	41
5.2.2.2	Mesures per l'experiència	41
5.2.2.3	Mesures per les capacitats específiques	41
5.2.2.4	Mesures per les capacitats genèriques	42
5.2.2.5	Mesures per salary i hours	42
5.2.2.6	Mesures per change residence i travel	43
5.2.3	Funcions de negació	43
5.2.3.1	Negacions per a la qualificació	44
5.2.3.2	Negacions per la experiencia	44
5.2.3.3	Negacions per les capacitats específiques	44
5.2.3.4	Negacions per les capacitats genèriques	45
5.2.3.5	Negacions per salary i hours	45
5.2.3.6	Negacions per change residence i travel	45
5.2.4	La funció de distancia	46
5.2.5	Els pesos	46
6	<i>L'AGENT EMPRESA DE SELECCIÓ DE PERSONAL</i>	47
6.1	Descripció	47
6.2	Disseny de les BDs	49
6.3	Connexió amb la base de dades	50
	La classe Statement	50
	Obtenció de resultats	51
6.4	Obtenir un origen de dades ODBC	51
7	<i>INICIALITZACIÓ DEL SISTEMA</i>	52
7.1	Fitxers d'entrada	52
7.1.1	Fitxers d'entrada per l'agent personal	52
7.1.2	El fitxer d'entrada per l'agent ESP	53
7.1.3	Fitxer d'entrada per l'agent Empresa	53
7.2	Vocabulari utilitzat	54
7.3	Els fitxers de resultats	56
8	<i>JOC DE PROVES</i>	58
9	<i>TREBALLS FUTURS</i>	78
10	<i>VALORACIÓ</i>	79
11	<i>BIBLIOGRAFIA</i>	80

1 INTRODUCCIÓ

Trobar feina el més ràpid possible és un tema que preocupa a moltes persones. Per la seva banda, les empreses volen trobar les persones més adequades per ocupar els llocs de treball. És per això que persones i empreses acostumen a treballar amb empreses intermediàries de selecció de personal.

En molts casos, quan una persona cerca feina es dirigeix a una oficina d'una o varies empreses de selecció de personal. Cada empresa de selecció de personal té un imprès de sol·licitud de feina diferent, això implica que la persona haurà d'omplir tants impresos diferents, però amb informació similar, com a empreses de selecció de personal és vulgui apuntar, i retornar-los a l'empresa de selecció corresponent.

Per la seva banda, les empreses que busquen treballadors, també es posen en contacte amb una o varies empreses de selecció de personal, i els donen les característiques de la feina o feines que ofereixen i les característiques que busquen en un treballador.

Les empreses de selecció de personal acostumen a introduir tant les dades de la persona, com les de la feina, en una base de dades d'una manera més o menys manual, amb el temps que això implica. Després, l'empresa de selecció ha de buscar els possibles candidats per a una feina, i ha d'enviar els seus currículums a l'empresa que cerca treballadors, depenent del nivell d'informatització potser hi haurà currículums que s'hagin de revisar manualment.

Quan les empreses que cerquen treballadors reben els currículums dels candidats a una feina des de totes les empreses de selecció de personal amb les que treballen, s'han de mirar tots els currículums un a un per a ordenar-los de millor a pitjor segons el grau d'adequació que tinguin per la feina i eliminar els currículums repetits. Llavors es contacta, normalment de forma telefònica, amb els candidats que l'empresa consideri millors per a obtenir més dades necessàries per la selecció o per a oferir-los la feina.

Quan un a persona trobi feina, haurà d'anar trucant a totes les empreses de selecció de personal a les que s'havia apuntat per a donar-se de baixa. Paral·lelament l'empresa també haurà d'anar informant a totes les empreses de selecció que el lloc de treball ja ha estat ocupat.

Per solucionar el problema de la lentitud de procés de les tasques i de la comunicació entre persones, empreses que cerquen treballadors i empreses de selecció de personal, es proposa una metodologia de sistemes multi-agent. Els sistemes multi-agent permeten treballar de forma distribuïda i permeten una comunicació eficient. Escollir els diferents candidats per una feina i decidir quin o quins són els més adequats es pot solucionar amb l'ús d'algunes tècniques d'intel·ligència artificial, de forma que l'elecció es realitzi de manera més eficient.

En aquest projecte es pretén crear un sistema multi-agent (SMA) que serveixi per modelar la interacció entre empreses intermediàries de selecció de personal (ESPs), persones que busquen feina, i empreses que cerquen personal.

Hem fet un prototipus centrat en el cas concret de la selecció de personal en el món de la informàtica, ja que, és aquest, el que ens és més proper, tanmateix, com veurem, el sistema és aplicable a qualsevol altre entorn.

En el sistema tindrem cinc tipus diferents d'agents.

- ESP: agent que representa una empresa de selecció de personal.
- Personal: agent que representa la persona que cerca feina.
- Empresa: agent que representa una empresa que ofereix llocs de treball.
- Simulador, agent que ens ajuda a simular l'arribada d'agents al sistema en un cas real.
- ClusDMA: agent per l'ajuda de presa de decisions multi-criteri.

Aquest document s'estructura de la següent manera: al capítol 2 es fa una introducció dels conceptes bàsics d'agents i Sistemes Multi-Agent ; al capítol 3 s'explica, JADE, que és l'eina de desenvolupament i programació de Sistemes Multi-Agent utilitzada; a partir d'aquí ens centrarem en la feina feta en el projecte, en el capítol 4 es descriurà l'arquitectura del sistema i la comunicació entre els agents; en el capítol 5 es parlarà detalladament de l'agent Empresa; al capítol 6 es descriurà l'agent ESP (empresa de selecció de personal) i la seva interacció amb la base de dades; el capítol 7 es dedicarà a la descripció del funcionament del programa; en el capítol 8 trobem un joc de proves; en el capítol 9 es descriuen possibles línies de treball futur , en el capítol 10 hi ha una valoració del projecte i, finalment, trobem la bibliografia en el capítol 11.

2 AGENTS I SISTEMES MULTIAGENT (SMA)

La definició d'agent genera confusió, existeixen diferents definicions del concepte d'agent. Un agent es pot definir com un sistema que habita en un entorn complex i dinàmic; és capaç de sentir i actuar de manera autònoma en el seu entorn, i té un conjunt d'objectius o motivacions que intentarà aconseguir a través de les seves accions.

2.1 Propietats dels agents

Com que no es pot definir d'una manera clara que és una agent, s'acostuma a donar una llista de propietats necessàries perquè un sistema hardware o software sigui considerat un agent. Les propietats més importants que caracteritzen als agents són les següents:

- ❑ **Autonomia:** els agents han d'operar sense la intervenció directa dels humans o altres, i han de 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. La comunicació entre agents pot ser descrita com un contracte de col·laboració on l'usuari dóna unes especificacions que hauran de ser portades a terme per l'agent.
- ❑ **Reactivitat:** els agents perceben el seu entorn i responen en un temps raonable als canvis detectats.
- ❑ **Pro-activitat o capacitat de prendre iniciativa:** Els agents no només han de respondre al seu entorn, sinó que han de ser capaços d'exhibir algun tipus de comportament encaminat a assolir algun tipus d'objectiu prenent la iniciativa.
- ❑ **Mobilitat:** els agents poden moure's a altres entorns a través d'una xarxa electrònica. Poden traslladar dades amb instruccions que poden ser executades remotament.
- ❑ **Continuïtat temporal:** els agents estan contínuament executant processos, no són només un procés que s'acaba.
- ❑ **Veracitat:** és l'assumpció que un agent no comunicarà informació falsa premeditadament.
- ❑ **Benevolència:** és l'assumpció que els agents no tenen objectius conflictius, i que cada agent intentarà fer allò pel qual és requerit.
- ❑ **Racionalitat:** és l'assumpció que l'agent ha d'actuar per tal d'aconseguir el seu objectiu.

2.2 Tipus d'agents

S'ha fet una possible classificació del tipus d'agents. Veurem les seves característiques fonamentals:

- ❑ **Agents d'Internet o d'informació:** agents que tenen accés a les fonts d'informació (Internet), gestionen i manipulen dades; poden respondre a requeriments imposats pels usuaris o altres agents. Un exemple molt comú són els cercadors d'informació a Internet.
- ❑ **Agents d'interfície:** sistema on els agents col·laboren amb l'usuari per resoldre un problema. L'agent observa les accions de l'usuari i realitza una interacció aprenentatge per suggerir a l'usuari millors maneres de dur a terme una tasca.
- ❑ **Agents de col·laboració:** la característica principal és la comunicació i cooperació amb altres agents per resoldre un problema comú. La capacitat de resoldre problemes entre un grup d'agents s'ha de dur a terme utilitzant tècniques d'IA distribuïda.
- ❑ **Agents mòbils:** la seva característica principal és la possibilitat de poder moure's per una xarxa electrònica. Aquesta mobilitat es fa interactuant amb altres host, permet als agents recollir informació i realitzar serveis requerits per l'usuari.
- ❑ **Agents reactius:** aquests tipus d'agents no treballen amb un model simbòlic de l'entorn. El seu comportament es basa en la resposta a estímuls segons un patró de l'estat actual en què es troben. Aquests agents i les seves interaccions amb d'altres agents són molt simples, però la combinació de diversos agents reactius pot generar comportaments complexos.
- ❑ **Agents híbrids:** combinen diversos dels tipus d'agents anteriors per tal de minimitzar els desavantatges de cada tipus d'agent per separat.
- ❑ **Agents heterogenis:** es refereixen a un conjunt integrat d'almenys dos agents que pertanyen a dues o més classes diferents.

2.3 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 responent a estímuls. La combinació de diversos agents reactius proporciona un comportament “intel·ligent”.
- ❑ Híbrides: estructures que barregen diferents tècniques.

2.4 Sistemes Multi-Agent (SMA)

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

2.4.1 Beneficis d'un SMA

Els beneficis que aporta utilitzar un sistema Multi-Agent són els que s'enumeren a continuació:

- Modularitat: es redueix la complexitat distribuint les tasques, i permet una programació més estructurada.
- Eficiència: repartir la feina entre diversos agents permet paral·lisme quan els agents estan treballant en màquines diferents.
- Fiabilitat: el sistema no ha de deixar de funcionar si un dels seus elements deixa de fer-ho. Es pot aconseguir més seguretat replicant serveis i aconseguint redundància.
- Flexibilitat: es poden incorporar i eliminar agents al sistema dinàmicament.

2.4.2 Especificació FIPA de la gestió d'agents

L'administració d'agents facilita una normativa de treball dins de la qual els agents existeixen i operen. 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) estableix el següent model per a aquest entorn administratiu d'agents [FIPA, 2000b]:

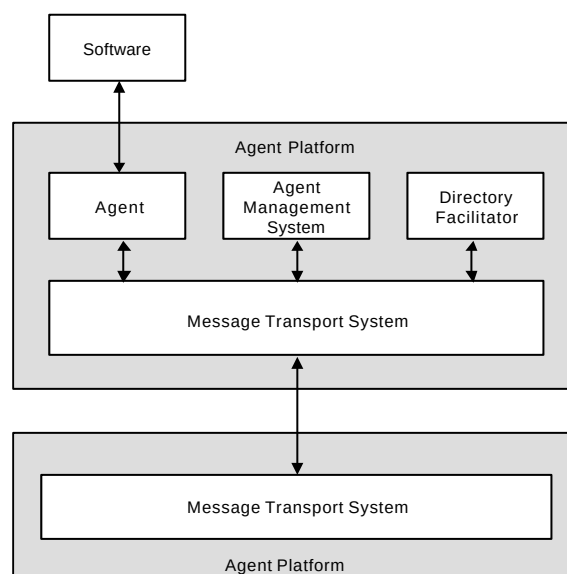


Fig. 1

Els components lògics que componen l'entorn de manipulació d'agents i que hem vist a la **Fig. 1** són:

- ❑ **Agents**: és una unitat bàsica del sistema o l'element actor fonamental. És pot descriure com un programa que encapsula una sèrie de serveis.
- ❑ **Agent Platform (AP)**: proporciona la infraestructura física en la que es poden crear-se i executar-se els agents.
- ❑ **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 al DF per a oferir els seus serveis.
- ❑ **Agent Management System (AMS)**: és un agent que exerceix un control de supervisió sobre l'accés i l'ús de la plataforma d'agents. Només un AMS existirà en un *Agent Platform (AP)*. Emmagatzema els noms lògics i les adreces de transport dels agents, i ofereix un servei de *Pàgines Blanques*.
- ❑ **Message Transport Service (MTS)**: facilita la comunicació entre agents de diferents plataformes.
- ❑ **Software**: qualsevol programa accessible des d'un agent.

2.4.3 El llenguatge de comunicació entre agents

Els agents individualment proporcionen una funcionalitat interessant, però el que fa que siguin molt útils per resoldre determinats problemes, que sols potser no podrien solucionar, és la seva capacitat de cooperar. Normalment, el coneixement que té cada agent és limitat i potser erroni o incomplet, per això cal que cooperi amb altres agents per a poder beneficiar-se'n. Per poder realitzar la cooperació, 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.

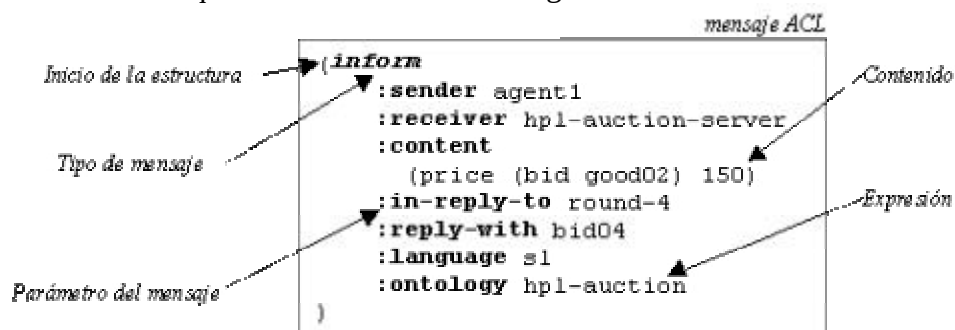


Fig. 2

L'element bàsic de la comunicació és el missatge; la FIPA determina quina forma ha de tenir un missatge i la seva utilització [FIPA, 2000a]. En la **Fig. 2** observem els components bàsics que ha de tenir un missatge.

2.4.3.1 Components d'un missatge

Els missatges estan representats com expressions, un missatge es compon d'una instrucció per a identificar el tipus d'acte comunicatiu i de diferents paràmetres, anem a veure el significat d'aquests paràmetres:

Definició del tipus de missatge

Dins d'aquesta categoria només es té un element: *performative*. Indica el tipus d'acte de comunicació. Es tenen diferents actes de comunicació:

- **accept-proposal**: acceptació una proposta per realitzar una acció.
- **agree**: acceptació d'una acció per realitzar una acció.
- **cancel**: cancel·lació d'alguna petició anterior que encara no ha finalitzat.
- **cfp**: crida per proposar la realització d'una acció.
- **confirm**: l'emissor informa al receptor que una proposició és certa, quan el receptor dubtava sobre la proposició.
- **disconfirm**: l'emissor informa al receptor que una proposició és falsa, quan el receptor creia que era certa.
- **failure**: s'indica la fallada d'una petició feta amb anterioritat.
- **inform**: l'emissor informa al receptor que una acció s'ha realitzat correctament.
- **inform-if**: s'informa al receptor si una proposició és certa o no.
- **inform-ref**: s'informa al receptor de si un objecte al qual es fa referència, és cert o no.
- **not-understood**: l'emissor informa al receptor que no ha entès una petició realitzada anteriorment.
- **propagate**: Un agent envia un missatge i un conjunt de AIDs a qui es reenviarà el missatge; l'agent primer tracta el missatge i torna a enviar.
- **propose**: acció de proposar la realització d'una acció, donades certes precondicions.
- **proxy**: l'emissor espera que el receptor identifiqui als agents que poden satisfer una descripció donada, i els hi faci arribar.
- **query-if**: acció de preguntar a un altre agent si un fet és cert o no.
- **query-ref**: acció de preguntar sobre un objecte.
- **refuse**: acció de refusar la petició de realitzar una acció donada, explicant la raó de la negativa.
- **request**: l'emissor demana al receptor realitzar una acció.
- **request-when**: l'emissor vol que el receptor realitzi alguna acció quan alguna proposició esdevingui certa.
- **request-whenever**: es demana a un agent que realitzi una acció tant bon punt es donin unes condicions.
- **subscribe**: l'emissor demana ser avisat en el moment que es compleixi una condició.

Participant de la comunicació:

Un participant de la comunicació és un identificador d'un agent. N'hi ha 3:

- **sender** : identificador de l'agent que envia el missatge.
- **receiver**: identificador de l'agent que 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.

- FIPA-CCL (FIPA *Constraint Choice Language*): defineix una semàntica que permet especificar predicats amb satisfacció de restriccions (CSP).
- FIPA-SL (FIPA *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.
- FIPA-KIF (FIPA *Knowledge Interchange Format*): permet expressar objectes com a termes i proposicions com a sentències. No permet l'expressió d'accions.
- FIPA-RDF (FIPA *Resource Description Framework*): permet representar objectes, interaccions i accions entre ells. Semblant al FIPA-SL

Descripció del contingut

La FIPA defineix tres descriptors per al contingut d'un missatge:

- **language**: nom del llenguatge en el qual està escrit el contingut del missatge, en aquest cas utilitzarem el FIPA-SL.
- **encoding**: indica si s'utilitza algun tipus de codificació especial.
- **ontology**: defineix el vocabulari usat en el contingut dels missatges.

Control de la conversa

Els *slots* de control de conversa estan destinats a identificar quin tipus de converses que es mantenen.

- **protocol**: indica el protocol usat en els missatges.
- **conversation-id** identificador de conversa.
- **reply-with**: introdueix una expressió que serà usada per l'agent que respongui per identificar el missatge.
- **in-reply-to** identifica una acció passada a la qual aquest missatge és la resposta.
- **reply-by** marca de *timeout*: data/hora de caducitat del missatge.

2.4.3.2 Protocols de comunicació

Els protocols de comunicació defineixen uns passos que s'ha de seguir per a determinar el curs d'una conversa. Hi ha diferents tipus de protocols depenent de la seva finalitat. Els diferents protocols definits per la fipa són els següents [FIPA, 1998b]:

- FIPA *Request*: permet a un agent demanar a un altre que realitzi unes accions i retorni els resultats
- FIPA *Query*: s'usa per demanar a un agent informació sobre algun fet.
- FIPA *Contract Net*: permet realitzar negociacions entre dos o més agents.
- FIPA *Iterated Contract Net*: variant de l'anterior que permet la renegociació
- FIPA *Brokering*: gestiona una demanda cap a un conjunt d'agents que poden donar un cert servei.
- FIPA *Recruiting*: variant de l'anterior
- FIPA *Subscribe*: es basa en el fet que un agent demana a un altre que quan succeeixi un cert fet, li sigui notificat.
- FIPA *Propose*: simplificació del *Contract Net*

3 JADE

JADE (Java Agent Development Enviroment) és un entorn de construcció de Sistemes Multiagent, conté un conjunt de llibreries escrites en JAVA i proporciona un conjunt de *packages* JAVA que permeten crear agents. A més de proporcionar les funcions de manegament d'agents 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.

La versió utilitzada per a la implementació d'aquest projecte és la 2.01.

JADE es compona d'un conjunt de paquets que descriurem a continuació:

- ❑ **jade.core:** conté 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 inclou 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:
 - *Remote Management Agent (RMA):* actua com una consola gràfica per a l'administració i control de la plataforma.
 - *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.
 - *Sniffer:* agent que intercepta missatges ACL i els visualitza.
 - *SocketProxyAgent:* agent que actua com un *gateway* bidireccional entre la plataforma JADE i una connexió TCP/IP.
- ❑ **jade.gui:** conté eines per construir interfícies gràfiques.
- ❑ **jade.mtp:** conté mètodes de transport de missatges
- ❑ **jade.ont:** trobem la definició d'ontologies. Inclou la classe *frame*.

3.1 Plataforma d'agents

Per JADE, els agents resideixen en un entorn predefinit que s'anomena 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 poden residir en màquines diferents o no. Jade proporciona un entorn segur perquè els agents es puguin comunicar.

Tal com es defineix a la FIPA [FIPA, 1998b], en cada plataforma es defineixen dos agents vitals pel funcionament del sistema: el *Domain Facilitator* (DF), que proporciona un servei de *Pàgines Grogues* i l' *Agent Management System* (AMS, que proporciona un servei de *Pàgines Blanques*.

Gràcies a aquests serveis, 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 que és qui coneix tots els agents de la plataforma.

3.2 Agents

Una agent JADE és un objecte de la classe *Agent*. Per crear un agent hem d'estendre la classe *Agent* i implementar els mètodes que són obligatoris de la classe.

Dos mètodes molt importants que s'han d'implementar necessàriament són:

- ❑ **setup()** : aquest mètode s'executa quan l'agent entra al sistema per primera vegada. Inicialitza l'agent i posa en marxa comportaments.
- ❑ **takedown()** : aquest mètode s'executa quan se surt del sistema. Finalitza l'agent.

Un agent pot definir diversos comportaments (behaviours). Per cada servei que hagi de donar un agent s'implementarà un o més comportaments. Un *scheduler* intern controlarà l'execució de tots els comportaments.

3.3 Comportaments

Per a donar funcionalitat a un agent s'hauran de definir comportaments (behaviours). Cal implementar com a mínim un behaviour per agent. Jade proporciona un conjunt de comportaments amb els quals podem treballar. N'hi ha de dos tipus:

- Comportaments simples: no tenen fills. Aquest behaviours són:
 - **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.
- Comportaments complexos: poden tenir fills, s'agrupen en la classe abstracta **ComplexBehaviour**. En té dues subclasses:
 - **SequencialBehaviour**: executa tots els seus fills de manera seqüencial i acaba quan tots els fills acaben.
 - **NonDeterministicBehaviour**: no executa els fills en seqüència. Segueix una política Round Robin. Cal una condició d'acabament per acabar el comportament principal.

3.4 Comunicació

Jade segueix les especificacions de la FIPA [FIPA,2000a] i permet implementar tots els missatges. Les operacions bàsiques sobre un missatge són crear-lo (usa el mètode `set`) i consultar-lo (una el mètode `get`).

Per l'enviament i recepció de missatges existeixen dos mètodes bàsics:

- `send (ACLMessage)`: s'usa per enviar un missatge.
- `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 FIPA –SL (*Semantic Language*) és un llenguatge formal amb el qual es pretén poder especificar una gran varietat de problemes. En FIPA-SL es poden 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

El contingut dels missatges entre agents són d'una determinada ontologia i s'especifica al camp *ontology*. L'ontologia és molt important, perquè el vocabulari que s'usarà en els missatges. Una ontologia està formada per tres tipus d'elements:

Una ontologia està formada per objectes o *frames*: cada *frame* està compost per atributs. Cada atribut o *slot* és un camp d'informació en un objecte, un *slot* pot ser un altre frame, un objecte bàsic o un conjunt d'elements. Cada *slot* està etiquetat i s'hi podrà accedir pel nom i pel tipus.

Els objectes estan formats per dues parts:

- Definició dels slots: Els slots contenen el nom se l'slot, el tipus, l'estructura de dades que el manegarà, i si és un slot que serà obligatori o opcional en els missatges.
- Definició de la classe Java: s'ha de definir una classe que s'encarregarà de manegar cada objecte o predicat. 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

JADE proporciona els elements necessaris (*behaviours* i *performatives*) per definir els protocols descrits per la FIPA [FIPA, 1998b]. Bàsicament els protocols tenen dos parts, la part de qui fa la crida del protocol (*initiator*) i la part de qui manega la resposta d'aquest (*responder*). Cada agent haurà d'estendre els comportaments en funció del seu paper.

JADE no implementa tots els protocols que defineix la FIPA. JADE implementa aquest tres:

3.6.1 FIPA Request

En aquest protocol un agent demana a un altre agent que realitzi una acció i que li retorni els resultats.

El protocol s'implementa en dues classes:

- **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.

- **FIPARequestResponderBehaviour:** 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.

En la Fig. 3 podem veure el protocol FIPA Request. Els quadres blancs representen l'*initiator* i els grisos el *responder*.

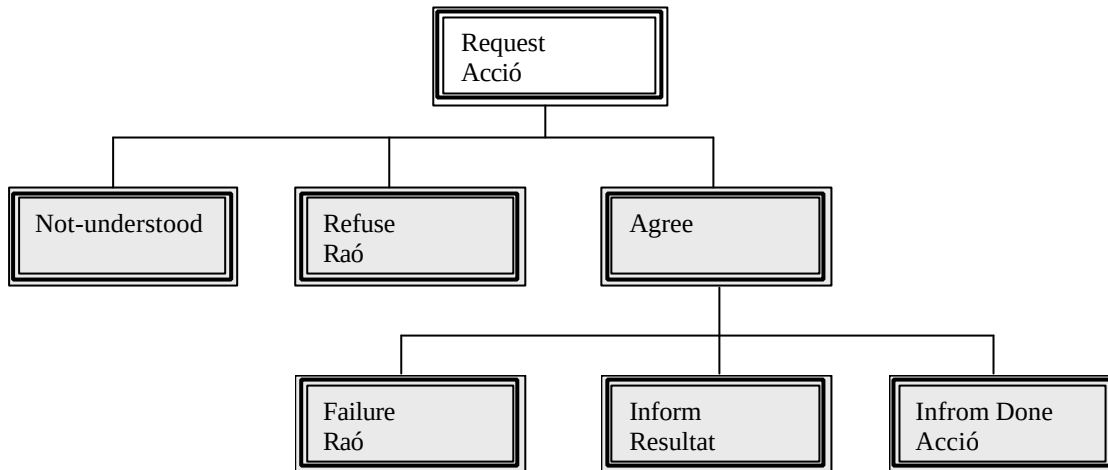


Fig. 3

3.6.2 Protocol FIPA Query

S'utilitza quan un agent vol demanar algun tipus d'informació a un altre agent. El protocol s'implementa en dues classes:

- **FIPAQueryInitiatorBehaviour:** la implementa l'agent *initiator* que comença l'enviament de la pregunta i manega les possibles respostes.
- **FIPAQueryResponderBehaviour:** implementa l'altre costat de la conversa. Usualment aquest pot funcionar com un servidor de *queries*.

En la Fig. 4 podem veure el protocol FIPA Query. Els quadres blancs representen l'*initiator* i els grisos el *responder*.

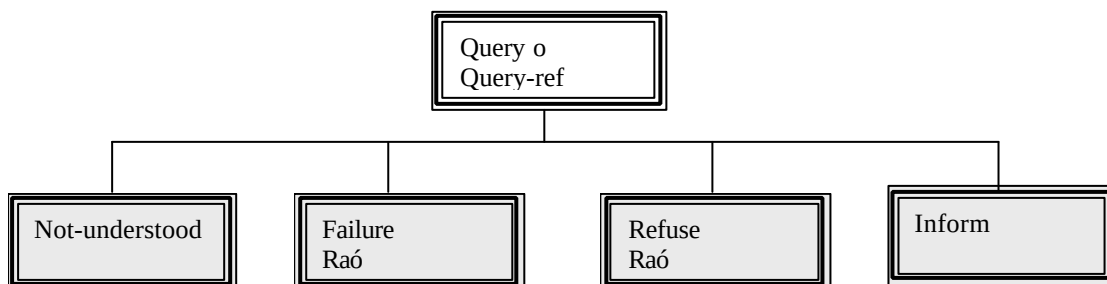


Fig. 4

3.6.3 Protocol FIPA Contract Net

Aquest protocol s'utilitza quan es vol realitzar una negociació entre diversos agents.

El protocol s'implementa en dues classes:

- **FIPACONTRACTNETInitiatorBehaviour:** la implementa l'agent *initiator* que comença l'enviament del *call for proposal* (cfp) i manega les possibles respostes. El protocol considera les propostes que li arriben i accepta les més adients. Totes les respostes s'ha de contestar sigui per a acceptar o per refusar. Si l'agent accepta una proposta, espera els resultats.
- **FIPACONTRACTNETResponderBehaviour:** implementa l'altre costat de la conversa, l'agent *responder*. Té dues funcions bàsiques: 1) esperar missatges del tipus cfp i generar propostes adients. 2) Si accepta la proposta generada, realitzar l'acció associada.

En la Fig. 5 podem veure el protocol FIPA Contract Net. Els quadres blancs representen l'*initiator* i els grisos el *responder*.

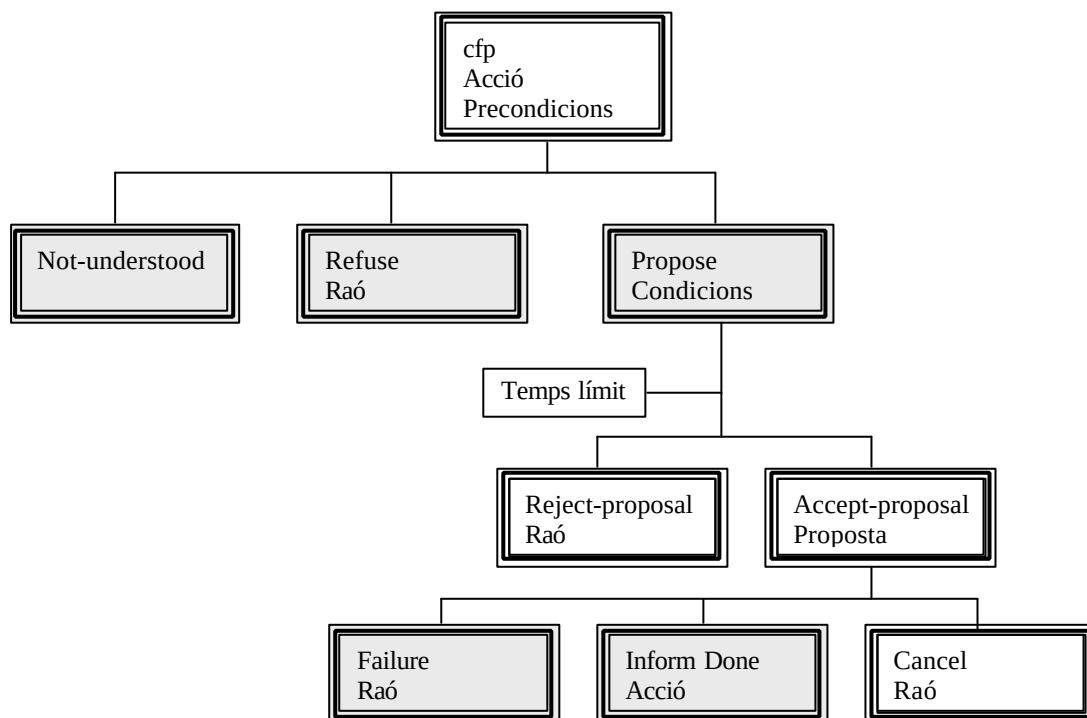


Fig. 5

3.7 Eines de desenvolupament JADE

JADE ofereix una interfície gràfica per a l'administració de la plataforma, així com eines per seguir el funcionament del sistema Multi-Agent.

Per a iniciar la sessió i els agents hem d'escriure la comanda:

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

Si es vol inicialitzar l'entorn gràfic cal posar a les opcions el paràmetre `-gui`. En engegar-se l'entorn gràfic es creen uns quants automàticament:

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

3.7.1 RMA

L'agent RMA (Remote Agent Management) s'ocupa de la interfície gràfica d'una plataforma d'agents. Mostra l'estat de la plataforma a la que pertany i proporciona eines per a demanar accions i per depurar i testejar aplicacions basades en JADE.

A continuació podem veure la interfície gràfica d'un RMA:

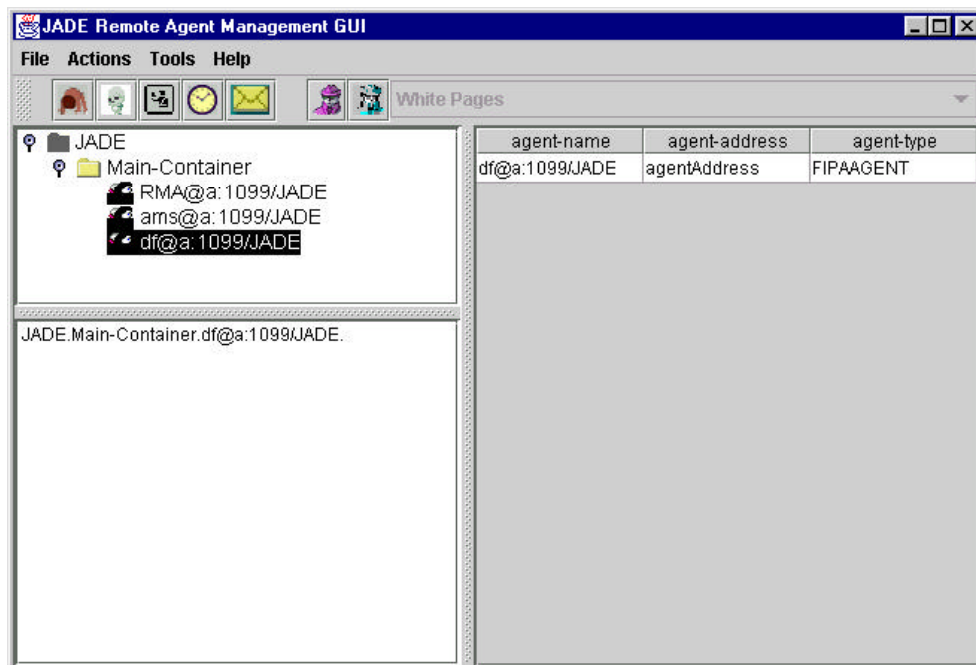


Fig. 6

3.7.2 Sniffer

És una aplicació JAVA que rastreja l'intercanvi de missatges en un entorn basat en JADE. Permet veure en la seva interfície gràfica (Fig. 7) els missatges que es van passant entre els agents. 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. És una aplicació JAVA que rastreja l'intercanvi de missatges en un entorn basat en JADE..

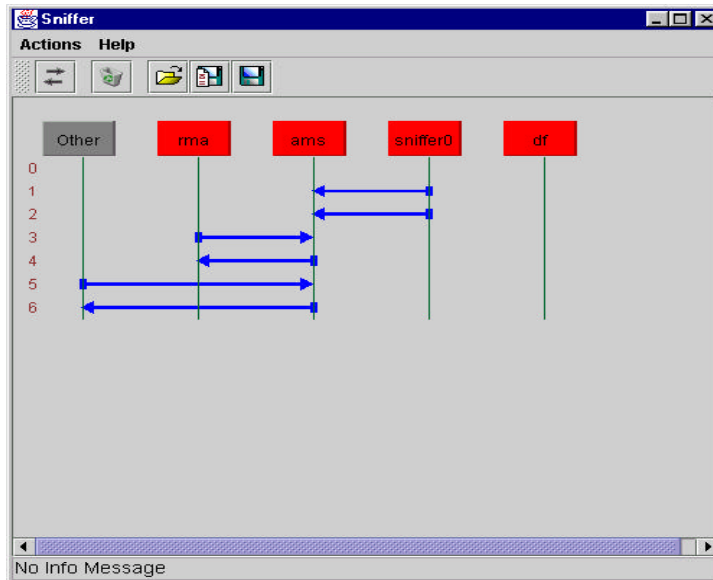


Fig. 7

3.7.3 Dummy Agent

Aquest agent permet enviar missatges ACL entre els agents, rebre'ls i inspeccionar-los i llegir i salvar-los a un fitxer.

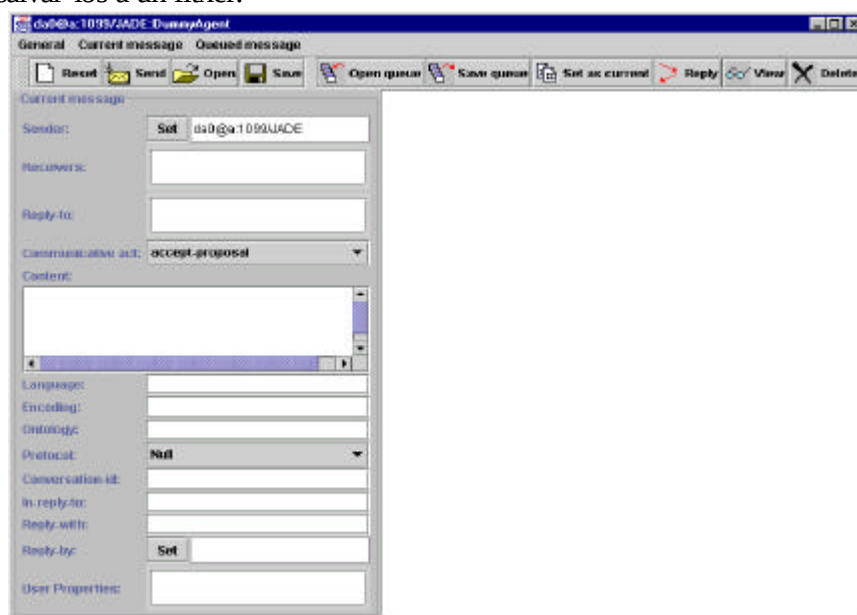


Fig. 8

3.7.4 EI DF

També podem activar la interfície de l'agent DF (des del menú *tools*), on podrem observar els agents que s'han registrat i quin serveis ofereixen.

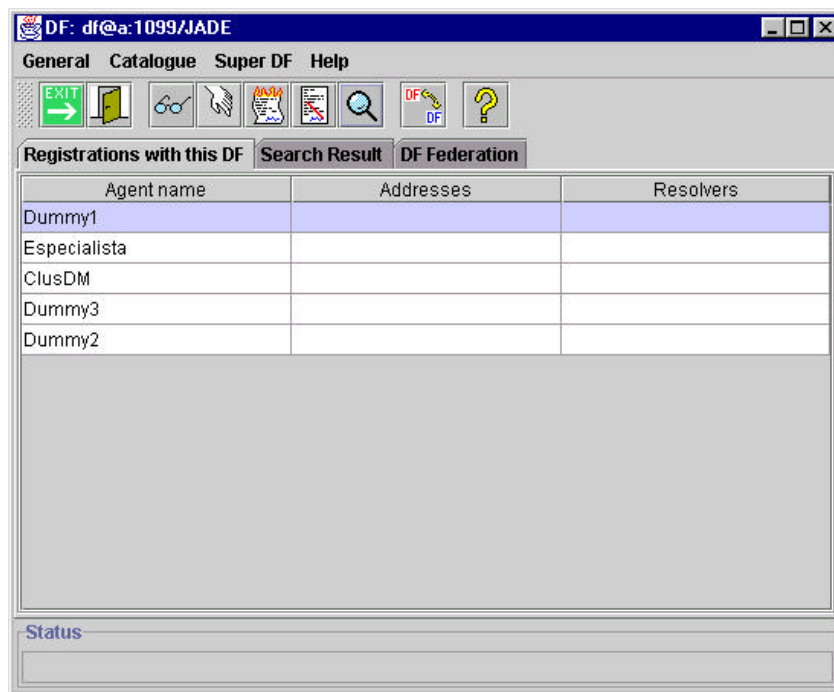


Fig. 9

4 DESCRIPCIÓ DEL SISTEMA

Com ja s'ha explicat al capítol 1, l'objectiu d'aquest projecte és implementar una aplicació basada en agents que realitzi la interacció entre empreses de selecció de personal, persones que cerquen feina i empreses que ofereixen un lloc de treball.

Quan una persona vol buscar feina, el seu agent Personal s'encarregarà de posar-se en contacte amb una o varies ESPs (Empreses de Selecció de Personal) perquè cada una li envii una sol·licitud de feina. L'agent Personal les omplirà amb les dades curriculars i les retornarà. A més d'enviar les dades a les ESPs, l'agent Personal també enviarà informació a l'agent Empresa quan aquesta estigui interessada en ella. Finalment pot decidir si accepta o no una proposta de feina. En cas afirmatiu, l'agent es torna a posar en contacte amb les ESPs per a informar-les.

Quan una empresa ofereix una feina, el seu agent Empresa contactarà amb les ESPs i els enviarà les característiques de la feina o feines. També, demanarà a les ESPs que li proposin candidats per una feina. Quan una empresa rep la llista de possibles candidats, pot ser que les dades estiguin incompletes. Si una empresa ha demanat candidats a varies ESPs intentarà completar les dades dels candidats entre tots els que ha rebut de les ESPs, i si és necessari, es posarà en contacte amb l'agent Personal de cada candidat directament. Amb totes les dades completes ja podrà comunicar-se amb l'agent ClusDMA per a què aquest li retorni una valoració dels candidats, de millor a pitjor. Finalment, preguntarà als agents Personals d'un en un si volen la feina. Si l'accepten, l'agent Empresa informarà a les ESPs que ja ha ocupat el lloc de treball.

Els agents ESPs (Empreses de Selecció de Personal) faran de mitjancers entre les persones que busquen feina i les empreses que necessiten treballadors. Cada ESP selecciona un subconjunt d'atributs que considera important per crear la seva pròpia sol·licitud de feina. Quan l'agent Personal demani una sol·licitud la ESP li enviarà, quan li retornin omplerta emmagatzemarà a una base de dades les dades personals de la persona. A més, la ESP també guardarà a una base de dades les feines que les empreses vagin oferint i farà una pre-selecció de personal per una feina i li enviarà a l'empresa. Quan una persona trobi feina o una empresa trobi un treballador la ESP actualitzarà la seva base de dades.

L'agent simulador, engegarà agents Personals, ESP i d'Empresa cada un cert temps per a simular quin seria el funcionament en un cas real.

L'agent ClusDMA realitzarà el procés d'ordenació de candidats.

4.1 Per què un SMA?

Els motius o avantatges que ens ofereix utilitzar un Sistema Multi-Agent per solucionar aquest problema són els següents:

- Autonomia: permet treballar sense intervenció directa dels humans o altres, els agents tenen control sobre les seves accions.
- Flexibilitat: el poder incorporar i eliminar agents Personals, ESPs i d'Empresa fàcilment del sistema, i adaptar-los a les necessitats de cada empresa, donen una gran flexibilitat.
- Fiabilitat: com que aquest és un problema distribuït, hi ha vàries entitats diferents, el fet que algun dels ordinadors que formen el sistema deixi de funcionar, no implica que els altres agents no puguin continuar fent la seva tasca.
- Eficiència: un agent Personal o d'Empresa es pot connectar a vàries ESP alhora, i permet que es puguin executar paral·lelament (si es troben en diferents màquines) o concurrentment (a la mateixa) millorant el temps necessari per a realitzar la selecció.

4.2 Arquitectura del sistema

L'arquitectura del sistema és la que podem veure a la Fig. 10

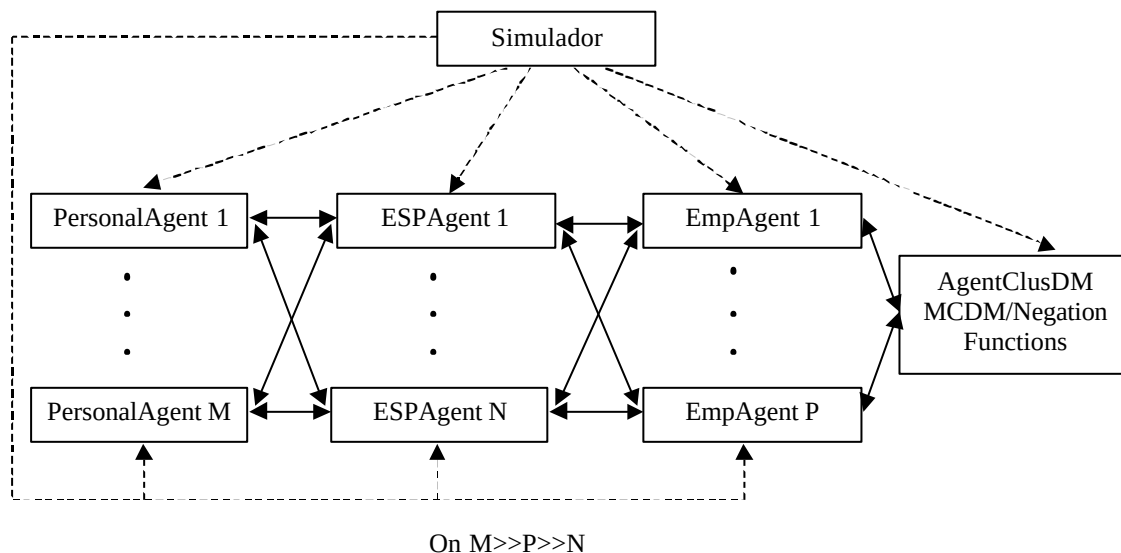


Fig. 10

Com podem veure els tipus d'agents que componen el SMA són els següents:

- Agent Personal: representa a una persona. L'objectiu d'un agent personal és trobar una feina per al seu propietari. En el nostre prototipus l'agent personal llegeix d'un fitxer les dades de la persona a la que representa, a continuació busca quins agents ESPs hi ha al sistema i en tria un o varis de manera aleatòria. L'agent Personal es posa en contacte amb les ESPs triades i els demana que li enviïn una sol·licitud de feina (en rep una de cada ESP), un cop l'obté omple totes les dades demanades i les retorna als agents ESP perquè, aquests, guardin la informació a les seves respectives bases de dades. Un cop fet això, l'agent Personal, només, haurà d'esperar que algun agent Empresa s'interessi per ell. L'agent empresa pot demanar-li les dades personals que li manquen per a poder realitzar el procés d'ordenació de candidats, o pot demanar-li si accepta o no una feina. En aquest segon cas, si l'agent Personal accepta, informarà a tots els agents ESPs als que s'havia apuntat que ja ha trobat una feina.

L'esquema en seqüència de la comunicació entre un agent Personal i un agent ESP és el que podem veure a la figura Fig. 11 .

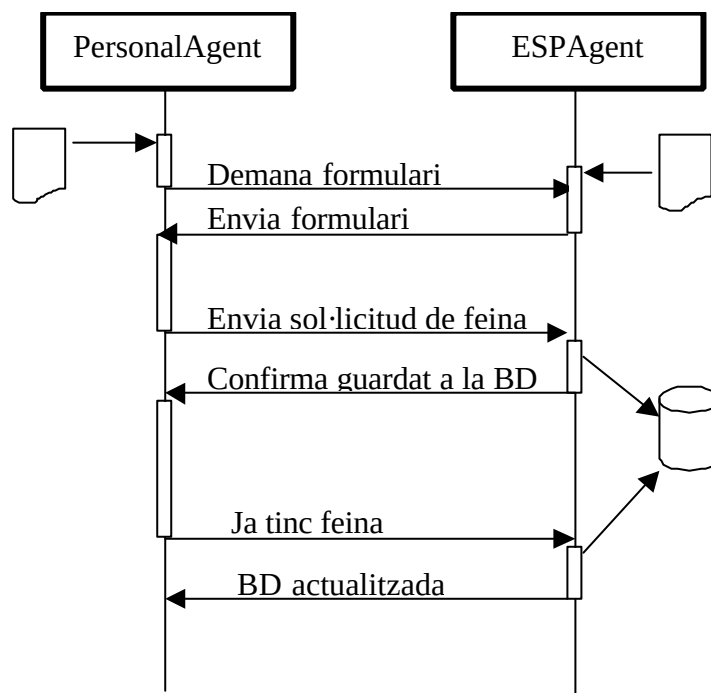


Fig. 11

- L'agent ESP (Empresa de Selecció de Personal): l'agent ESP representa a una empresa de selecció de personal. Aquest agent es registra al DF perquè els agents Personals i d'Empresa puguin demanar els seus serveis. Cada ESP té un formulari de sol·licitud de feina diferent, creat a partir d'un subconjunt dels atributs, que en aquest prototipus es llegeixen d'un fitxer. Les ESPs són les mitjanceres entre les persones que busquen feines i les empreses que cerquen treballadors. Cada ESP manté una base de dades locals amb les dades o currículums de les persones que cerquen feina i de les feines que han estat ofertes per les empreses que busquen ocupar un lloc de treball. Una altra funció de la ESP és fer un pre-filtratge dels possibles candidats a una feina quan una empresa li ho demani, és a dir, obtindrà, a partir de la seva base de dades, una llista de les persones que podrien encaixar amb una feina, i informará dels resultats a l'agent Empresa. Per últim, en el moment en què una empresa trobi un treballador, i una persona trobi una feina, l'agent ESP actualitzarà la seva base de dades, marcant que la persona ha trobat feina i que la feina ha estat ocupada.
- L'agent Empresa: l'agent Empresa representa l'empresa que ofereix una feina. El seu objectiu és trobar un candidat el millor possible perquè ocupi una vacant. L'agent empresa pot oferir una o varies feines, segons les seves necessitats. En el nostre prototipus, el primer que fa és llegir la feina o feines des d'un fitxer i es comunica amb els agents ESP perquè les guardi una a una a la base de dades. En aquest prototipus l'agent empresa treballarà amb totes les ESP perquè n'engegarem poques, però el sistema està pensat per a triar-ne un subconjunt aleatòriament. Posteriorment, demana a la ESPs que li enviïn una llista de tots els candidats que poden ocupar aquella plaça. Però com ja hem dit cada ESP no guarda tots els atributs de la persona, sinó que en guarden un subconjunt, per tant les dades del candidat no estaran complertes, llavors l'empresa intentarà unir totes les dades que ha rebut de les diferents ESP amb les que treballa, i si tot i així, no aconsegueix completar les dades, confeccionarà un objecte per a cada agent personal, no complet, amb les dades que li falten i les demanarà a l'agent personal. Quan, finalment, ja tingui totes les dades necessàries, per cadascun dels atributs que es consideren en el procés d'ordenació, haurà de transformar les dades de les persones a criteris d'adequació amb la feina utilitzant atributs qualitius. Un cop fet això, es comunicarà amb l'agent ClusDMA, enviant-los les dades que ha obtingut, juntament amb una manera d'interpretar-les usant funcions de negoció. Quan tingui els resultats mirarà si hi ha algun agent dins d'un rang mínim d'acceptació, si és així demanarà a l'agent corresponent a aquesta persona si accepta la feina. Si l'accepta informará a totes les ESP amb les que treballa que la vacant està coberta.

Podem veure l'esquema seqüencial de la comunicació de l'agent empresa amb els altres agents del sistema en la figura Fig. 12

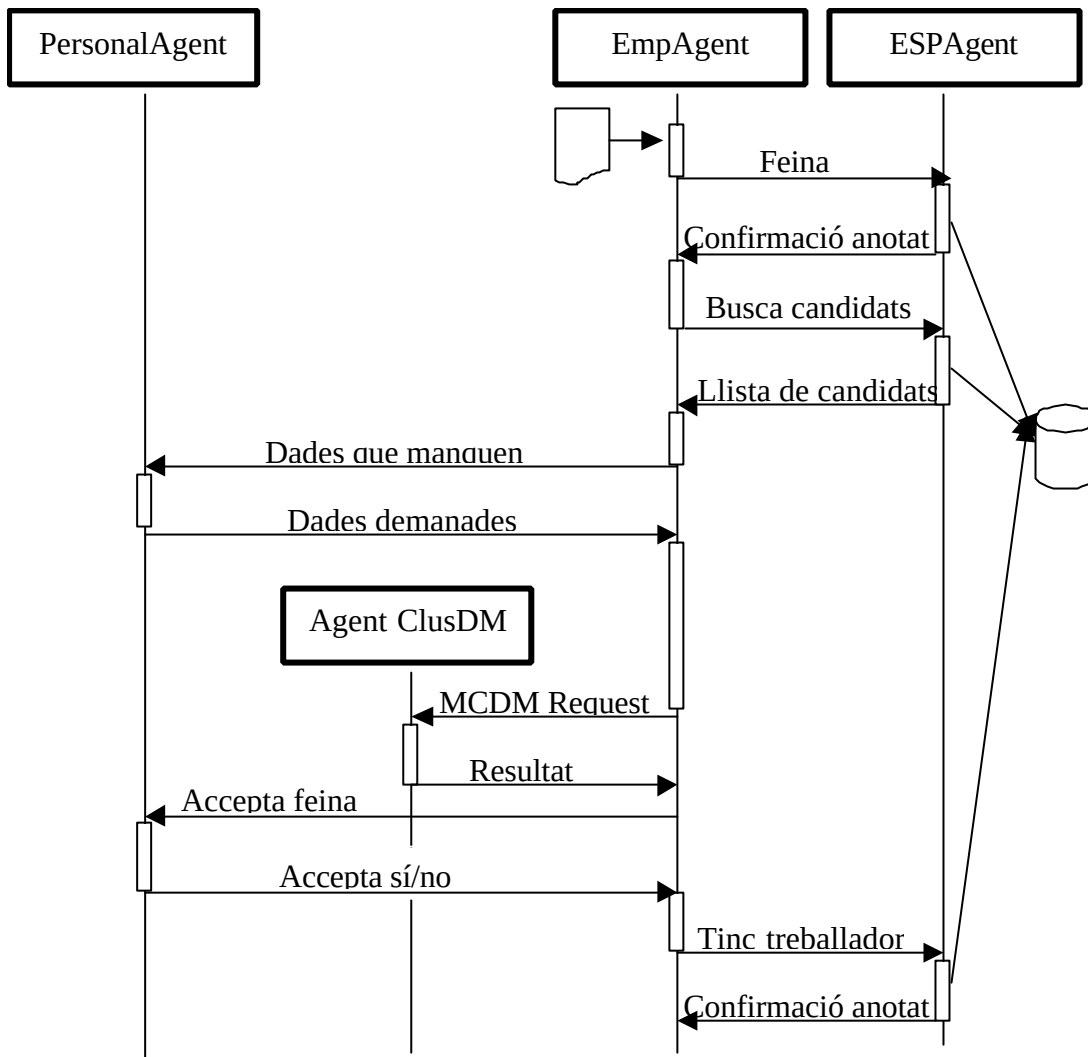


Fig. 12

- L'agent simulador: aquest agent l'hem inclòs per a simular com funcionaria els prototipus en un cas real. Els agents no s'engegarien tots a l'hora, sinó que anirien apareixen al sistema nous agents personals, d'empresa o ESP al llarg del temps. Aquest agent simulador engegarà tots els altres agents que intervindran en les proves, però no tots alhora sinó que cada un cert temps engegarà un o varis agents indicant quin és el seu nom i de quin tipus és.
- L'agent ClusDMA: és un agent capaç d'analitzar una llista d'objectes descrits mitjançant atributs qualitius i ordenar-los de millor a pitjor. Aquest problema es coneix com a "Presa de decisions multicriteri" (MCDM) ja que per a fer l'ordenació s'ha de considerar molts atributs alhora. En el ClusDMA cal que els agents siguin qualitius, és a dir, lingüístics, i els valors usats han d'estar ordenats (establir preferències). Aquest agent va ser implementat per David Sánchez en el seu projecte de final de carrera [Sánchez, 2001] seguint el mètode ClusDM desenvolupat per Àida Valls a la seva tesi doctoral [Valls et al., 2002].

En el capítol 5 s'explicarà més detalladament el funcionament de l'agent Empresa, mentre que en el capítol 6 es parlarà de l'agent ESP.

4.3 Comunicació entre agents

L'acte comunicatiu en un SMA es basa en el pas de missatges entre agents, per a comunicar-se correctament, els agents necessiten uns protocols, un vocabulari i un llenguatge comú.

El llenguatge que usarem serà el llenguatge definit per la FIPA: FIPA-SL [FIPA, 2000c].

L'ontologia l'hem definit nosaltres mateixos i l'explicarem a la secció 4.3.2.

4.3.1 Protocol

El protocol de comunicació defineix quins patrons determinats segueixen els agents quan realitzen l'acte comunicatiu, serveix per portar un control de la conversa entre agents de manera que cada agent sap en tot moment quins missatges pot enviar i quins pot rebre.

La FIPA com a organisme normalitzador ha definit una sèrie de protocols [FIPA,1998a]. En el nostre cas, un agent determinat demana a un altre que realitzi una acció i que li retorni uns resultats. Del conjunt de protocols definits per la FIPA, i que implementa JADE, el que més s'adequa a les nostres necessitats és el protocol *FIPA-Request*. Podem veure l'esquema seqüencial de pas de missatges del protocol *FIPA-Request*, adaptat al nostre cas, entre l'agent Personal i l'agent ESP a la Fig. 13 .

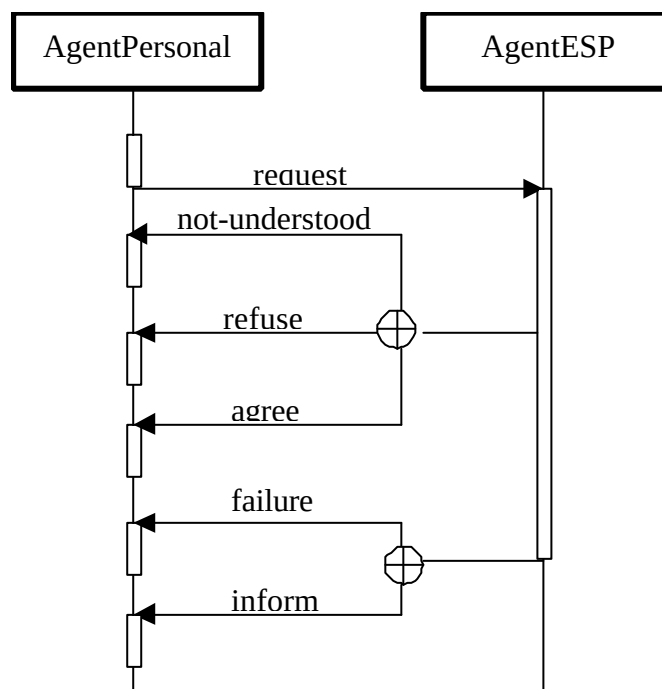


Fig. 13

4.3.2 Ontologia

L'ontologia defineix el vocabulari (objectes), el conjunt de predicats que podem formar i les accions que es poden sol·licitar a un agent.

En la nostra ontologia només necessitarem objectes i accions. La llengua usada en l'ontologia ha estat l'anglès. Així doncs, aquesta ontologia consistirà en la definició dels següents objectes o *frames*, formats pels camps o *slots* de tipus simple, altres frames o un conjunt d'elements, que es detallen a continuació.

4.3.2.1 Objectes

L'objecte Person

L'objecte person conté totes les dades referents a la persona que busca feina.

Person			
Paràmetre	Tipus	ús	Descripció
obl_pers_data	obl-personal-data	M	Dades personals obligatòries
other_pers_data	other-personal-data	O	Altres dades personals
studiesD	studies	M	Dades acadèmiques
job	job-part	M	Dades Laborals
specific_skills	specific-skills	M	Capacitats específiques
generic_skills	generic-skills	O	Capacitats genèriques
additional_data	additional-data	O	Altres dades
agent_name	AID	M	Identificador de l'agent

- obl-personal-data: dades personals de la persona que busca feina

OblPersonalData			
Paràmetre	Tipus	ús	Descripció
name	String	M	El nom de la persona que busca feina
surname1	String	M	El primer cognom
surname2	String	M	El segon cognom
nif	String	M	El NIF
address	postal_address	M	L'adreça

- postal-address: l'adreça postal de qualsevol edifici

PostalAddress			
Paràmetre	Tipus	ús	Descripció
country	String	M	El país
zip_code	String	M	El codi postal
city	String	M	La ciutat
street	String	M	El carrer
street_number	String	M	El portal

- other-personal-data: altres dades personals d'interès, com el telèfon o adreça de correu electrònic de contacte, la data de naixement, el sexe, i l'estat civil.

OtherPersonalData			
Paràmetre	Tipus	ús	Descripció
phone	String	O	El número de telèfon
mobile_phone	String	O	El número de telèfon mòbil
email	String	O	L'adreça de correu electrònic
date_of_birth	date	M	La data de naixement
sex	String	O	El sexe
married	Long	O	Estat civil

- date: data de naixement de la persona, es podria usar per a altres funcions

Data			
Paràmetre	Tipus	ús	Descripció
day	Long	M	Dia
month	Long	M	Mes
year	Long	M	Anys

- studies: estudis realitzats i qualificació obtinguda

Studies			
Paràmetre	Tipus	ús	Descripció
degree	String	M	El nivell d'estudis
especiality	String	M	L'especialitat estudiada
qualification	String	O	La qualificació obtinguda

- job-part: dades referents a les condicions de feina desitjades i l'experiència

JobPart			
Paràmetre	Tipus	ús	Descripció
experience	String	M	Anys treballats
salary	Long	O	Sou al que espira la persona que busca feina
hours	Long	O	Hores de dedicació setmanal a la feina
contract_type	String	O	Tipus de contracte laboral desitjat

- specific_skills: són les capacitats específiques de la persona

SpecificSkills			
Paràmetre	Tipus	ús	Descripció
Access	String	M	Nivell de coneixements en Access
Informix	String	M	Nivell de coneixements en Informix
Oracle	String	M	Nivell de coneixements en Oracle
MySQL	String	M	Nivell de coneixements en MySQL
Ada	String	M	Nivell de coneixements en Ada
C++	String	M	Nivell de coneixements en C++
Java	String	M	Nivell de coneixements en Java
Lisp	String	M	Nivell de coneixements en Lisp
Pascal	String	M	Nivell de coneixements en Pascal
Prolog	String	M	Nivell de coneixements en Prolog
MS DOS	String	M	Nivell de coneixements en MS DOS
OS/2	String	M	Nivell de coneixements en OS/2
Unix	String	M	Nivell de coneixements en Unix
Windows	String	M	Nivell de coneixements en Windows

typing	String	M	Nivell de coneixements en mecanografia
accountancy	String	M	Nivell de coneixements en comptabilitat
english	String	M	Nivell de coneixements en anglès
french	String	M	Nivell de coneixements en francès
german	String	M	Nivell de coneixements en alemany

- generic-skills: altres capacitats d'indole més general

GenericSkills			
Paràmetre	Tipus	ús	Descripció
leadership	String	O	Capacitat de lideratge
motivation	String	O	Capacitat de motivació
team_work	String	O	Capacitat de treball en equip
project_management	String	O	Capacitat de gestió de projectes

- additional-data: dades addicionals

AdditionalData			
GenericSkills			
change_residence	String	O	Disponibilitat per canviar de residència
travel	String	O	Disponibilitat per viatjar
destination	String	O	Destí on prefereix treballar
driving_licence	Long	O	Indica si té carnet de conduir
car	Long	O	Indica si té vehicle propi

L'objecte Application

L'objecte Application és l'objecte que utilitzen les ESPs per a definir quin subconjunt d'atributs són els que formaran la sol·licitud de feina que envien als agents Personals quan es volen donar d'alta en aquella ESP.

Application			
Paràmetre	Tipus	ús	Descripció
phone	Long	O	El número de telèfon
mobile_phone	Long	O	El número de telèfon mòbil
email	Long	O	L'adreça de correu electrònic
sex	Long	O	El sexe
married	Long	O	Estat civil
qualification	Long	O	La qualificació obtinguda
salary	Long	O	Sou al que espira la persona que busca feina
hours	Long	O	Hores de dedicació diària a la feina
contract_type	Long	O	Tipus de contracte laboral
leadership	Long	O	Capacitat de lideratge
motivation	Long	O	Capacitat de motivació
team_work	Long	O	Capacitat de treball en equip
project_management	Long	O	Capacitat de gestió de projectes
change_residence	Long	O	Disponibilitat per canviar de residència
travel	Long	O	Disponibilitat per viatjar
driving_licence	Long	O	Indica si té carnet de conduir
car	Long	O	Indica si té vehicle propi

L'objecte Work

Aquest objecte conté totes les característiques d'una feina que ofereix una empresa, i les condicions i capacitats mínimes que es requereixen als candidats

Work			
Paràmetre	Tipus	ús	Descripció
work_id	Long	M	Identificador de la feina
contract_type	String	M	Tipus de contracte laboral
hours	Long	O	Hores setmanals de treball
salary	Long	O	Sou aproximat ofert
city	String	M	Ciutat en la qual s'ofereix la feina
studies	String	M	Estudis mínims necessaris per a poder optar a la feina
especiality	String	M	Especialitat estudiada
experience	String	M	Experiència mínima necessària per a optar a la feina
age	String	M	Edat que ha de tenir el candidat
travel	Long	O	Disponibilitat per viatjar
residence	Long	O	Disponibilitat per canviar de residència
driving_licence	Long	O	Indica si cal carnet de conduir o no
car	Long	O	Indica si cal carnet tenir vehicle propi
obl_constraints	specific_skills	M	Condicions obligatòries mínimes que ha de complir el candidat
non_obl_constraints	generic_skills	O	Capacitats que es valorarien en el candidat

L'objecte Applicants

És una llista de persones que poden optar a una feina, és l'objecte que dóna un agent ESP com a resultat de l'acció FindApplicants (que expliquem més avall).

Applicants			
Paràmetre	Tipus	ús	Descripció
applicants	Set of Person	M	Persones que encaixen amb una feina

4.3.2.2 Les accions

Amb els objectes definits anteriorment no en tenim prou per a realitzar la comunicació entre els agents, ens calen unes accions per a poder realitzar les tasques pròpies de cada agent. Anem a veure les accions necessàries.

L'acció SendApplication

Mitjançant aquesta acció els agents Personals sol·liciten a un agent ESP que envii la sol·licitud de feina que ha d'omplir.

SendApplication			
Paràmetre	Tipus	ús	Descripció
application	Application	M	Sol·licitud de feina

La informació que s'intercanviarà serà la següent:

- **Request:**
 - ☐ Application: la sol·licitud de feina
- **Inform:**
 - ☐ Done

L'acció NewApplicant

Sol·licitada per l'agent Personal, envia la sol·licitud de feina complerta, mitjançant l'objecte persona, i demana a la ESP que guardi les dades a la seva base de dades.

NewApplicant			
Paràmetre	Tipus	ús	Descripció
person	Person	M	Dades personals sol·licitades

La informació que s'intercanviarà serà la següent:

- **Request:**
 - ☐ Person: Totes les dades demanades en la sol·licitud de la persona
- **Inform:**
 - ☐ Done

L'acció NewWork

La sol·licita l'empresa, demana a l'agent ESP que guardi les dades de la feina a la base de dades.

NewWork			
Paràmetre	Tipus	ús	Descripció
work	Work	M	Dades de la feina

La informació que s'intercanviarà serà la següent:

- **Request:**
 - ☐ Work: Dades de les condicions i característiques de la feina
- **Inform:**
 - ☐ Done

L'acció FindApplicant

Sol·licitada pels agents Empresa, demana a les ESPs que busquin possibles candidats per cobrir la feina indicada amb el seu identificador. Les ESP retornaran una llista de les persones seleccionades.

FindApplicant			
Paràmetre	Tipus	ús	Descripció
work_id	Long	M	Identificador de la feina

La informació que s'intercanviarà serà la següent:

- **Request:**
 - ❑ work_id: identificador de la feina per la qual es demana que es busquin candidats.
- **Inform:**
 - ❑ Applicants: llista amb totes les persones que podrien optar a la feina.

L'acció SendData

Sol·licitada pels agents Empresa, demana als agents Personals que li proporcionin les dades personals que li manquen.

SendData			
Paràmetre	Tipus	ús	Descripció
application	Application	M	Les dades que falten per completar les dades de la persona

La informació que s'intercanviarà serà la següent:

- **Request:**
 - ❑ application: objecte Application amb les dades que manquen a l'Empresa per completar a les dades de la persona.
- **Inform:**
 - ❑ Person: les dades demanades per l'agent Empresa

L'acció AcceptWork

La sol·liciten els agents Empresa, demana als agents Personals que decideixin si accepten o no la feina.

AcceptWork			
Paràmetre	Tipus	ús	Descripció
work_id	Long	M	Identificador de la feina
salary	Long		Salary que la empresa està disposada a pagar
hours	Long	M	Hores setmanals de feina

La informació que s'intercanviarà serà la següent:

- **Request:**
 - ❑ work_id: identificador de la feina
 - ❑ salary: salari que l'empresa està disposada a pagar
 - ❑ hours: hores setmanals de feina
- **Inform:**
 - ❑ accepta: indica si accepta o no la feina.

L'acció HaveWorker

La sol·liciten els agents Empresa, demana als agents ESP que actualitzin la base de dades, perquè ja ha cobert el lloc de treball.

HaveWorker			
Paràmetre	Tipus	ús	Descripció
work_id	Long	M	Identificador de la feina

La informació que s'intercanviarà serà la següent:

- **Request:**
 - ☐ work_id: identificador de la feina
- **Inform:**
 - ☐ Done

L'acció HaveWork

La sol·liciten els agents Personals, demana als agents ESP que actualitzin la base de dades, perquè ja han trobat una feina.

HaveWork			
Paràmetre	Tipus	ús	Descripció
nif	String	M	Nif de la persona que ha trobat feina
work_id	Long	M	Identificador de la feina

La informació que s'intercanviarà serà la següent:

- **Request:**
 - ☐ nif: de la persona que ha trobat feina
 - ☐ work_id: identificador de la feina ocupada
- **Inform:**
 - ☐ Done

MCDMOntology

A més a més, per a poder realitzar la comunicació amb l'agent ClusDMA, ha calgut incorporar l'ontologia (MCDMOntology) realitzada per David Sánchez en el seu PFC [Sánchez, 2001].

Cal apuntar que el que en MCDMOntology s'anomena *objecte*, correspon, en el nostre cas, a una persona o candidat, el *nobjectes* serà el número de persones i els *atributs* són les dades de les persones, característiques i coneixements.

L'esquema de l'ontologia (SPOntology) és el que es mostra a continuació i a la Fig. 15

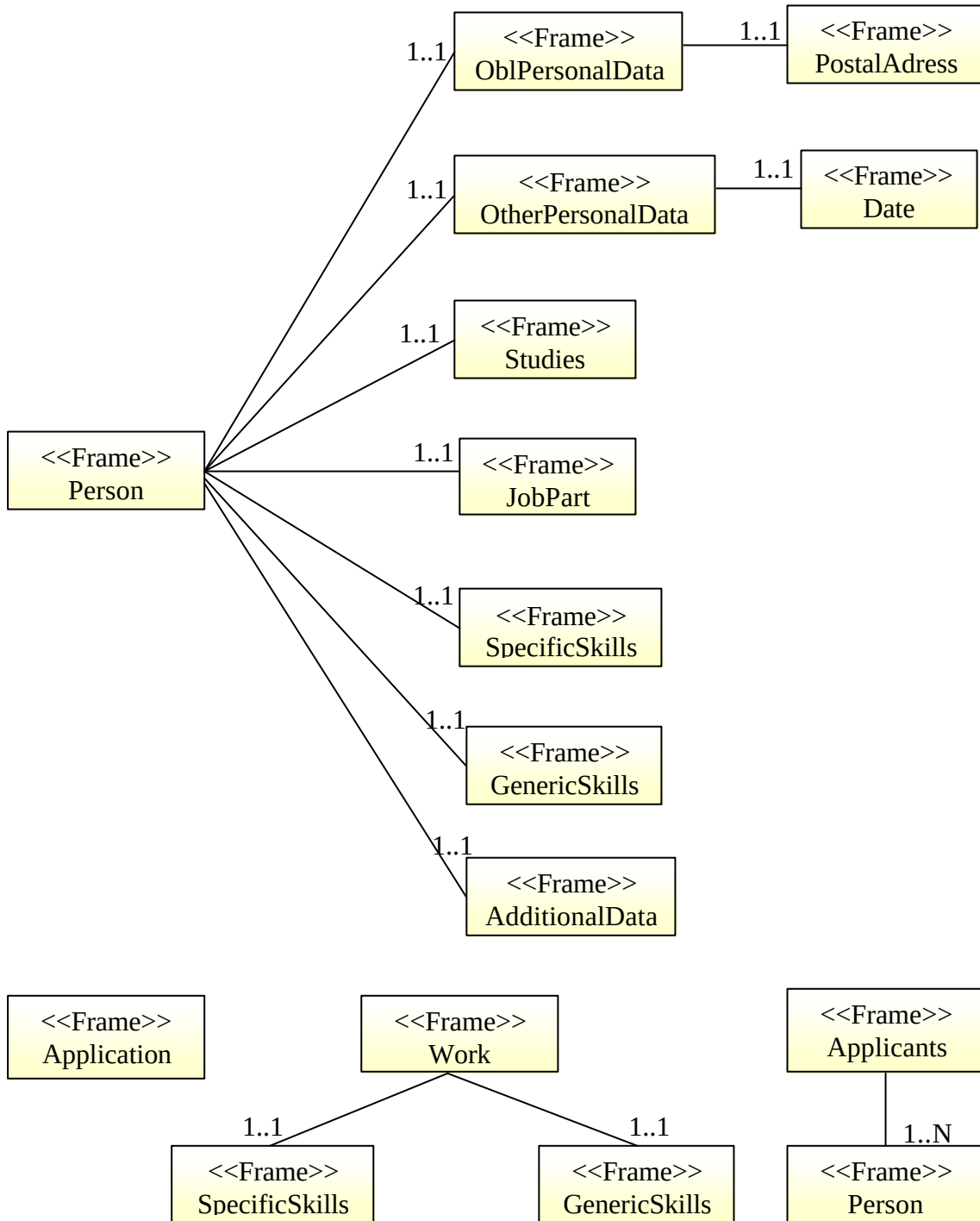


Fig. 14

Continuació de la SPontology:

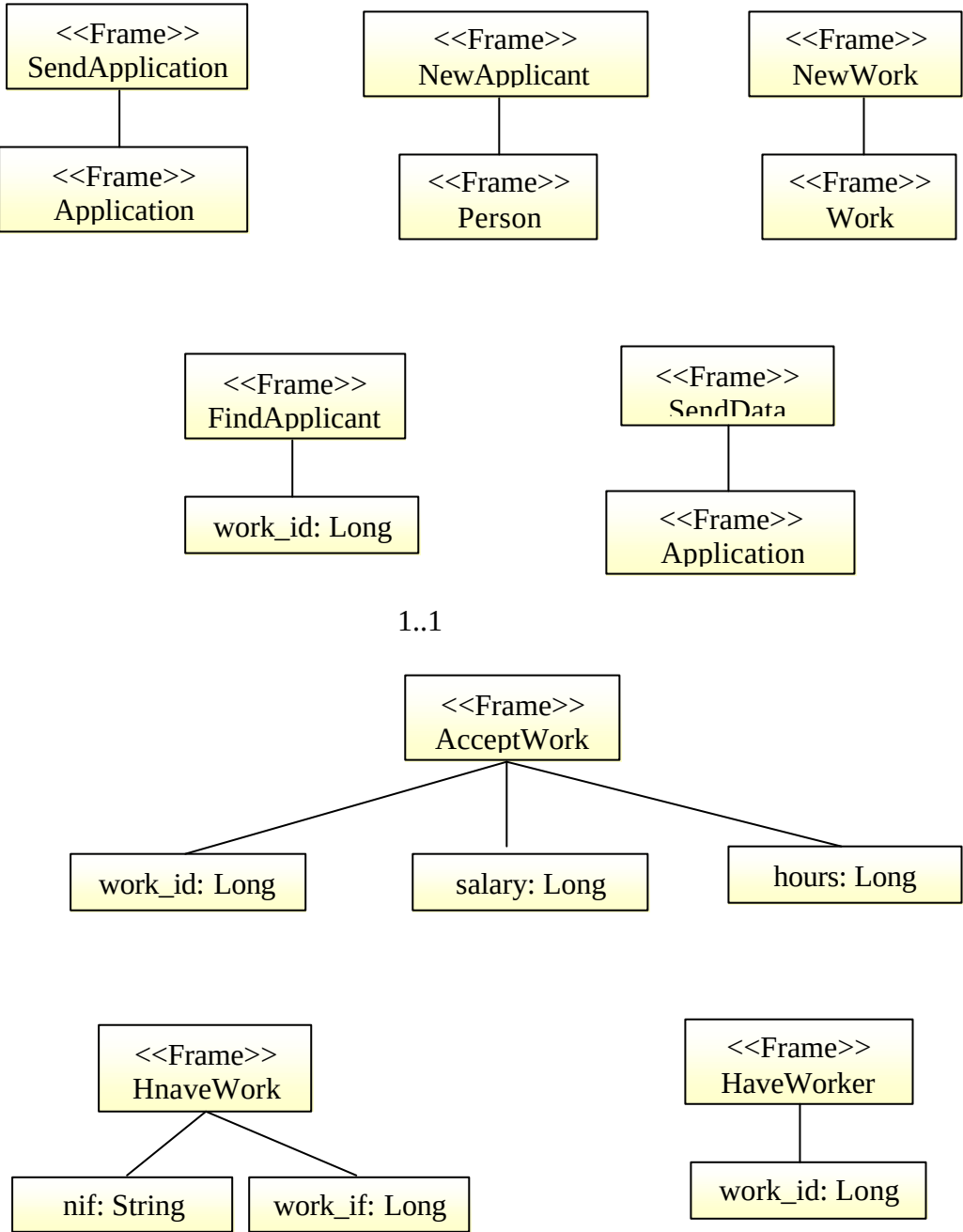


Fig. 15

L'esquema de l'ontologia (MCDMOntology) utilitzada per la comunicació amb l'agent ClusDMA es mostra en les Fig. 16 i Fig. 17

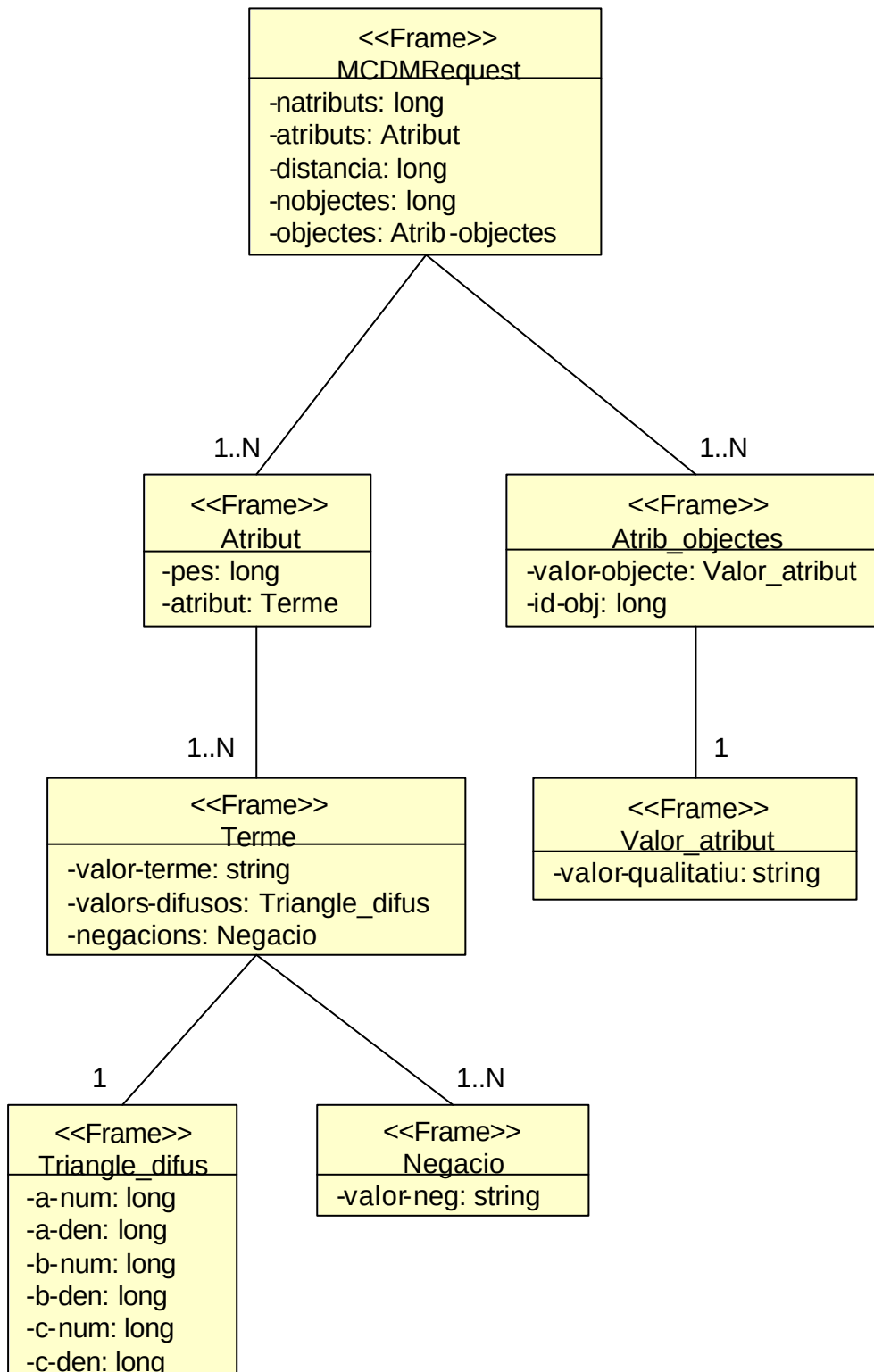


Fig. 16

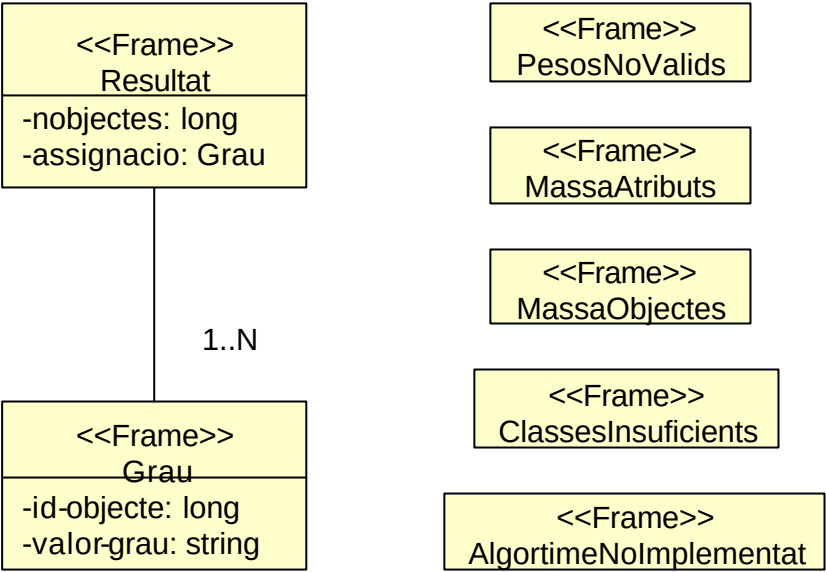


Fig. 17

5 L'AGENT EMPRESA

5.1 Descripció

Com ja hem comentat anteriorment, l'agent Empresa té com a funció aconseguir trobar treballadors per les feines que ofereix.

En aquest prototipus tenim les dades de la feina o feines en un fitxer, un per a cada agent Empresa. El primer que fa l'agent Empresa és llegir aquesta informació i guardar-la en un objecte *feina* (l'hem anomenat Work a l'ontologia), seguidament, busca al DF (Domain Facilitator) quins agents ESP hi ha al sistema i n'escull un subconjunt aleatòriament, en principi la meitat. Com que en les proves que farem engegarem pocs agents ESP l'agent empresa treballarà amb tots ells. A continuació, es busca al DF quins agents ClusDMA hi ha al sistema. En aquest cas només hi haurà un agent ClusDMA per a totes les empreses que busquen treballadors.

Un cop l'agent Empresa tingui les ESPs amb les que vol treballar, engegarà un comportament per a cada feina, de manera que no calgui esperar a trobar un treballador per una feina per a començar a buscar-ne per una altra. En cada un dels comportaments engegats per a una feina diferent, aquest agent es comunicarà amb les ESPs i els enviarà un objecte *feina*, perquè el guardin a la base de dades. Un objecte *feina* consta de les característiques de la feina i dels requeriments mínims que ha de complir un candidat. Fet això, l'agent Empresa demanarà a les ESPs que confeccionin una llista dels candidats que es puguin adequar a la feina.

Quan l'agent Empresa ja ha rebut la llista de candidats des de totes les ESPs amb les que treballa, potser, que no tingui totes les dades necessàries, sinó només una part, això es deu a que cada agent ESP confecciona una sol·licitud de feina diferent a partir d'un subconjunt dels atributs d'una persona. Com que, probablement, l'agent Empresa haurà demanat candidats a varies ESP, l'empresa podrà unir aquestes dades incompletes de la persona. Si tot i així no ha obtingut totes les dades que li calien, comprovarà quines són dades que li manquen i les demanarà directament als candidats o candidats. Pot passar que les ESPs no trobin cap candidat adequat per la feina, llavors, l'agent Empresa esperarà un temps (durant aquest temps poden arribar noves persones que sí que siguin adequades per la feina) i tornarà a fer la petició a les ESPs perquè busquin candidats per la feina.

Quan finalment, pugui completar totes les dades es començarà a construir el missatge que cal enviar a l'agent ClusDMA (calen un mínim de cinc candidats), que és, qui farà l'ordenació dels candidats de millor a pitjor a partir de termes qualitatius. Per a construir el missatge que ha de rebre l'agent ClusDMA, primer es filtraran els agents que no compleixin algunes condicions que són excloents (en parlarem en més detall a la secció 5.2.1) i a continuació buscarà els termes qualitatius per a cada atribut (s'explica més detalladament a la secció 5.2.2) . Quan ja s'hagin preparat les dades, ja es podrà demanar a l'agent d'ordenació que faci la seva tasca.

El resultat de la petició a l'agent ClusDMA serà una llista amb l'identificador del candidat i el grau d'adequació a la feina. Les dades que s'han preparat abans de fer el procés d'ordenació i els resultats obtinguts es guardaran en un fitxer. El vocabulari per a indicar el grau d'adequació a la feina d'un candidat és el següent.

```
vocabulari={"the_worst", "terrible", "very_bad", "bad", "not_recommendable",  
"inappropriate", "borderline", "acceptable", "adequate", "recommendable",  
"fairly_good", "good", "very_good", "excellent", "optimum"}
```

Si no hi ha cap candidat que es trobi entre els graus d'adequació "optimum" i "acceptable", es considerarà que no hi ha cap candidat suficientment adequat per la feina, llavors l'agent Empresa esperarà un temps i tronarà a demanar a les ESPs que busquin candidats, ja que des de l'última vegada que ho va fer podrien haver-se incorporat noves persones que cerquen feina i que, aquestes sí, podrien ser més adequades per a ocupar el lloc de treball.

Si l'agent Empresa troba algun candidat adequat, li pregunta si està disposat a acceptar la feina, si aquest l'accepta s'informarà a les ESPs de que ja es té el lloc de treball ocupat. Si no accepta, preguntarà al següent candidat de la llista, si és que ni ha algun més o el que hi ha és adequat. Si no n'hi ha més o tots tenen un grau d'adequació inferior a "acceptable", tornarà a demanar a les ESPs que busquin candidats per la feina.

5.2 Ordenació de candidats

El procés d'ordenació de candidats necessita de l'ús d'un agent anomenat ClusDMA. Aquest agent utilitza Mètodes de Decisió Multi-Criteri (MCDM) per a ordenar la llista de candidats segons l'adequació a la feina. L'agent Empresa, primer, haurà de filtrar tots aquells candidats que per determinades raons no són adequats per aquella feina, a continuació, haurà de preparar les dades perquè l'agent d'ordenació pugui treballar-hi.

5.2.1 Filtratge de candidats

Com ja hem dit, el primer que haurà de fer l'agent Empresa és filtrar els candidats que per diferents motius no siguin adequats per a la feina.

Els atributs que hem considerat excloents són els següents:

- Contract type: cal que el tipus de contracte que ofereix l'empresa que cerca treballador per a aquella feina sigui el mateix que el que desitja la persona que busca feina, si no és així s'eliminarà el candidat de la llista
- Car: cal que, en el cas, que l'empresa que ofereix la feina necessiti que la persona que ocupi el lloc de treball tingui un vehicle propi, aquest el tingui.
- Driving licence: cal que, en el cas, que l'empresa necessiti que la persona que ocupi el lloc de treball tingui carnet de conduir, aquest el tingui.

5.2.2 Obtenció dels termes qualitatius

L'algorisme d'ordenació, no rebrà les dades de les persones directament, sinó que abans haurem de preparar les dades, buscant la relació entre cada candidat i la feina en termes qualitatius. Per a cada característica de la persona hem definit un vocabulari de termes qualitatius. Com que, el número de característiques és bastant important i en molts casos són similars, les hem agrupat de la següent manera:

5.2.2.1 L'atribut qualificació

Per l'atribut *qualification* és considerará que el millor candidat és el que ha obtingut una qualificació en els seus estudis més alta.

Vocabulari de termes qualitatius:

Pass, Good, Excellent

5.2.2.2 Mesures per l'experiència

Per l'atribut *experience* també es considerará millor el candidat que tingui més experiència.

Vocabulari de termes qualitatius:

No_experience, Less_than_a_year, One_year, Two_years, Three_years,
Four_years, Between_5_and_10_years, More_than_10_years

5.2.2.3 Mesures per les capacitats específiques

Concretament es tracta dels atributs: *Access, Informix, Oracle, MySql, Ada, C++, Java, Lisp, Pascal, Prolog, MS DOS, OS/2, Unix, Windows, typing, accountancy, english, french, german.*

Tots aquests atributs ens informen del nivell de coneixements del candidat en cada una d'aquestes matèries.

El vocabulari de termes qualitatius que s'ha definit és el següent:

Bad, Borderline, Acceptable, Good

Els termes del vocabulari s'assignen de la següent manera, posarem com a exemple l'atribut Java:

- Si el nivell de coneixements mínims de Java demanats en una feina són “No”, siguin quin siguin els coneixements de Java de la persona, s'assignarà el terme “Good”.
- Si el nivell de coneixements mínims demanats són baixos, i el candidat no té coneixements de Java, s'assignarà el terme “Borderline”. Altrament, s'assignarà el terme “Good”.
- Si els coneixements mínims demanats en una feina són mitjans, i el candidat no té coneixements de Java, s'assignarà el terme “Bad”. Si els seus coneixements són baixos, s'assignarà el terme “Borderline”. Altrament s'assignarà el valor “Good”.
- Finalment, si els coneixements mínims demanats són alts, i el candidat no té coneixements, s'assignarà el valor “Bad”. Si els seus coneixements són baixos, s'assignarà el valor “Borderline”. Si el nivell de coneixements són mitjans, s'assignarà el valor “Acceptable”. I si són alts, s'assignarà el valor “Good”.

5.2.2.4 Mesures per les capacitats genèriques

Els atributs on s'aplica són: *leadership, team work, motivation, project management*

Vocabulari de termes qualitatiu

Bad, Acceptable, Good

Els termes del vocabulari s'assignen de la següent manera, posarem com a exemple l'atribut “leadership”:

- Si en la feina no es busca algú amb aquestes capacitats, s'assignarà el terme “Good”.
- Si en la feina es busca algú amb una certa capacitat de lideratge, i el candidat considera que no en té, s'assignarà el valor “Acceptable”. Altrament, s'assignarà el valor “Good”.
- Si en la feina es busca algú amb una clara capacitat de lideratge, i el candidat considera que no en té, s'assignarà el valor “Bad”. Si el candidat considera que té una certa capacitat de lideratge, s'assignarà el valor “Acceptable”. Altrament, s'assignarà el valor “Good”.

5.2.2.5 Mesures per salary i hours

En aquest cas, es calcula el rati entre el salari o hores de treball setmanal que ofereix l'empresa i el salari o hores de treball que demana el candidat

Vocabulari de termes qualitatiu

Very_Good, Good, Acceptable, Not_recomendable, Bad

Els termes del vocabulari s'assignen de la següent manera:

- Si $0.0 \leq | \text{rati}-1 | < 0.042$, s'assigna el valor " Very_Good "
- Si $0.042 \leq | \text{rati}-1 | < 0.166$, s'assigna el valor " Good"
- Si $0.166 \leq | \text{rati}-1 | < 0.208$, s'assigna el valor " Acceptable "
- Si $0.208 \leq | \text{rati}-1 | < 0.25$, s'assigna el valor "Not_recomendable"
- Altrament, s'assigna el valor "Bad"

5.2.2.6 Mesures per change residence i travel

Vocabulari de termes qualitatiu

Good, Acceptable, Borderline, Bad

Els termes del vocabulari s'assignen de la següent manera, posarem com a exemple l'atribut "change residence":

- Si en la feina no es posa com a condició canviar de residència, s'assigna el valor "Good".
- Si en la feina es posa aquesta condició, i el candidat està disposat a canviar de residència, s'assignen el valor "Good".
- Si es posa la condició, i el candidat està disposat a canviar de residència depenen de diverses circumstàncies, s'assignen el valor "Acceptable".
- Si es posa la condició, però el candidat no està especialment disposat a canviar de residència s'assignen el valor "Borderline"
- Si es posa la condició, però el candidat no està disposat a canviar de residència, s'assigna el valor "Bad".

5.2.3 Funcions de negació

Una vegada tenim assignats els valors qualitatiu per a tots els termes que utilitzarem per a fer l'ordenació de candidats, definirem les seves funcions de negació. Les funcions de negació són una forma d'expressar quin o quins són els valors oposats per a valor determinat. Les funcions de negació per a cada criteri són les que s'ha definit més avall.

5.2.3.1 Negacions per a la qualificació

Vocabulari

Pass, Good, Excellent

Funcions de negació

Neg(Pass)=Excellent, Good

Neg(Good)=Pass

Neg(Excellent)=Pass

5.2.3.2 Negacions per la experiencia

Vocabulari

No_experience, Less_than_a_year, One_year, Two_years, Three_years,
Four_years, Between_5_and_10_years, More_than_10_years

Funcions de negació

Neg(No_experience)= More_than_10_years, Between_5_and_10_years

Neg(Less_than_a_year)= Four_years, Three_years

Neg(One_year)= Three_years, Two_years

Neg(Two_years)= One_year

Neg(Three_years)= One_year, Less_than_a_year

Neg(Four_years)= Less_than_a_year

Neg(Between_5_and_

5.2.3.3 Negacions per les capacitas especificques

Concretament es tracta dels atributs: Access, Informix, Oracle, MySQL, Ada, C++, Java, Lisp, Pascal, Prolog, MS DOS, OS/2, Unix, Windows, typing, accountancy, english, french, german.

Vocabulari

Bad, Borderline, Acceptable, Good

Funcions de negació

Neg(Bad)= Good, Acceptable, Borderline

Neg(Borderline)= Acceptable, Borderline

Neg(Acceptable)= Borderline, Bad

Neg(Good)=Bad

5.2.3.4 Negacions per les capacitats genèriques

Els atributs on s'aplica són: leadership, team work, motivation, project management

Vocabulari

Bad, Acceptable, Good

Funcions de negació

Neg(Bad)= Good, Acceptable

Neg(Acceptable)= Acceptable

Neg(Good)= Bad

5.2.3.5 Negacions per salary i hours

Vocabulari

Very_Good, Good, Acceptable, Not_recomendable, Bad

Funcions de negació

Neg(Very_Good)= Bad, Not_recomendable

Neg(Good)= Not_recomendable, Acceptable

Neg(Not_recomendable)= Good

Neg(Acceptable)= Good

Neg(Bad)= Very_Good

5.2.3.6 Negacions per change residence i travel

Vocabulari

Good, Acceptable, Borderline, Bad

Funcions de negació:

Neg(Good)= Bad

Neg(Acceptable)= Borderline

Neg(Borderline)= Acceptable

Neg(Bad)= Good

5.2.4 La funció de distància

L'agent ClusDMA implementa diferents funcions de distància per a comparar els candidats entre si. Les funcions que considera aquest agent són:

- Diferències
- MCD (Mean Character Difference)
- Manhattan
- Canberra Metric
- Taxonomica

En el nostre projecte s'utilitzarà la funció de distància MANHATTAN perquè sembla ser més adequada per comparar valors qualitatius.

5.2.5 Els pesos

Un altre paràmetre que admet l'agent ClusDMA és el pes dels criteris en l'ordenació. Per al seu correcte funcionament, cal que la suma de tots ells sigui igual a 100.

En aquest cas, s'han definit tots els pesos amb el mateix valor, dividint 100 del número d'atributs que hi intervenen.

6 L'AGENT EMPRESA DE SELECCIÓ DE PERSONAL

6.1 Descripció

L'agent ESP (Empresa de Selecció de Personal), fa de mitjancer entre les persones que cerquen un lloc de treball i les empreses que necessiten treballadors.

Cada ESP mantindrà una base de dades local amb les persones que cerquen feina i les empreses que cerquen treballadors, així doncs tindrem tantes bases de dades com agents ESP hi hagi.

Cada ESP selecciona un cert subconjunt dels atributs rellevants per a buscar feina i crea una sol·licitud de feina pròpia. Com a restricció, hi ha dades que es consideren obligatòries (com el nom, el nif, l'edat, etc) i que l'agent ESP sempre rebrà. En el prototipus, els agents ESP llegeixen d'un fitxer quines dades formaran part de la sol·licitud i quines no.

Després de llegir el fitxer i crear la sol·licitud, els agents ESP intenten carregar el driver JDBC-ODBC per mitjà del mètode *forName* de la classe *Class*. Si no és troba el driver és mostrarà un missatge indicant el problema i es matarà l'agent. A continuació, els agents ESP es registren al DF (Directory Facilitator) perquè els agents Personals i d'Empresa puguin demanar els seus serveis. Cada agent engega la seva interfície gràfica on es mostrarà informació de quines són les peticions que estan sent ateses i de quin agents són els que fan aquestes peticions.

Els agents ESP engeguen diversos comportaments diferents del tipus *FipaRequestResponderBehaviour_sl* per manejar *request*. Anem a detallar cada un d'aquests comportaments:

□ El comportament *AppliRespBehaviour*

Aquest comportament és el que manega peticions de tipus *request* quan una agent Personal demana a la ESP que li enviï la sol·licitud de feina. L'agent ESP agafa la sol·licitud de feina que havia creat en llegir del fitxer, i l'envia a l'agent Personal que li havia demanat.

□ El comportament *NewAppliRespBehaviour*

Aquest comportament és el que manega peticions de tipus *request* quan un agent Personal li envia les dades de la sol·licitud perquè les guardi a la base de dades. L'agent ESP intenta establir una connexió amb la base de dades. Primer, comprova a la taula *People* que no existeixi cap persona amb el mateix nif i que encara no hagi trobat feina. Pot ser que una persona hagués trobat feina en el passat mitjançant aquella ESP i que ara torni a necessitar aquest servei. Per comprovar-ho executa la consulta per mitjà del mètode *executeQuery*. A aquest mètode li passem un string que conté la consulta SQL:

```
SELECT DISTINCT P.*
FROM PEOPLE P
WHERE P.nif =nif and P.workfind=false
```

Si no hi és a la base de dades, s'inserta:

```
INSERT INTO PEOPLE VALUES name, surname, surname2, nif, country,
zip_code, city, street, street_number, phone, mobile_phone, email, day, month,
year, married, sex, degree, especiality, qualification, experience,
salary, hours, contract_type, Access, Informix, Oracle,
MySQL, Ada, Cpp, Java, Lisp, Pascal, Prolog, MS_DOS, OS2, Unix,
Windows, typing, accountancy, english, french, german,
leadership, motivation, team_work, project_management,
change_residence, travel, destination, driving_licence, car,
agent_name, workfind
```

Un cop ha guardat les dades a la base de dades, es tanca la connexió amb la base de dades i s'envia la confirmació de que ha finalitzat.

Comportament NewWrkRespBehaviour

Aquest comportament és el que manega el missatge de request quan un agent Empresa li envia les dades d'una feina perquè les guardi a la base de dades. L'agent ESP estableix una connexió amb la base de dades. Primer, fa una consulta a la taula de feines (*Work*) per a comprovar que aquella feina no estigui ja a la base de dades. La consulta SQL és la següent:

```
SELECT DISTINCT W.*
FROM Work W
WHERE W.work_id =id_feina and W.workerFind =false
```

Si no hi és s'inserta a la base de dades la nova feina. En acabar, tanca la connexió i envia la confirmació de que la feina ha estat guardada.

```
INSERT INTO WORK VALUES work_id, contract_type, hours, salary,
city, studies, especiality, experience, age, travel, residence,
driving_licence, car, Access, Informix, Oracle, MySQL, Ada, Cpp,
Java, Lisp, Pascal, Prolog, MS_DOS, OS2, Unix, Windows, typing,
accountancy, english, french, german, leadership, motivation,
team_work, project_management, workerFind
```

El comportament FindApplRespBehaviour

Comportament que manega la petició d'un agent Empresa perquè es busquin els candidats per a una feina. Aquest comportament rep l'identificador de la feina per la que ha de buscar els candidats. Primer, s'estableix la connexió amb la base de dades i es fa una consulta per a obtenir les característiques de la feina i els requisits que ha de complir un candidat.

```
SELECT DISTINCT W.*
FROM Work W
WHERE W.work_id =identificador_feina
```

Quan ja es tenen les dades de la feina, ja tenim els criteris que hem de fer servir per fer la consulta a la taula de persones (*People*). La consulta SQL és la que hi ha a continuació:

```
SELECT DISTINCT P.*
FROM PEOPLE P "
WHERE p.especiality=especialitat_demanada_feina and P.year
BETWEEN Any_minim and Any_max and P.workerfind = false
```

Un cop realitzada la consulta s'obté un objecte del tipus *ResultSet* amb totes les persones que complien els requisits per la feina. S'omple un objecte *Person* per a cada candidat trobat, s'envia la llista dels candidats a l'agent Empresa i es tanca la connexió.

Comportament HaveWorkerRespBehaviour

Quan un agent Empresa avisa a l'ESP que ha trobat un treballador, l'ESP es connecta amb la seva base de dades i actualitza el camp workerFind de la taula *Work*, que és el camp que indica si s'ha trobat o no un treballador. La sentència SQL és la següent:

```
UPDATE WORK W
SET W.workerFind=true
WHERE W.work_id=identificador_feina
```

Comportament HaveWorkRespBehaviour

Quan un agent Personal avisa a l'ESP que ha trobat una feina., l'ESP es connecta amb la seva base de dades i actualitza el camp workerfind i el de l'identificador de la feina trobada de la taula *People*, que és el camp que indica que s'ha trobat una feina i el seu identificador. La sentència SQL és la següent:

```
UPDATE PEOPLE P
SET P.workerfind=true, P.work_id=id_feina
WHERE P.nif=nif_persona
```

6.2 Disseny de les BDs

Les bases de dades dels agents ESP estan formades per dues taules: una taula de persones, anomenada ***People***, on es guarda les dades referents a la persona, i una taula de feines, anomenada ***Woks***, on el guarden totes les dades de les característiques de la feina i dels requisits que ha de complir un candidat.

Per al correcte funcionament del prototipus cal que les bases de dades i els seus orígens de dades tinguin el mateix nom que l'agent al qual pertanyen.

6.3 Connexió amb la base de dades

Abans de connectar-se amb una base de dades cal registrar els controladors.

Java utilitza la classe *DriverManager* per carregar els controladors disponibles que es desitgen utilitzar. Aquesta classe serà l'encarregada de tornar-nos la connexió amb la base de dades.

Per a carregar un controlador del tipus pont JDBC/ODBC s'ha d'usar el mètode *forName()* de la classe *Class*, que retorna un objecte *Class* associat a la classe que es passa com a paràmetre. Exemple:

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

Un cop registrat el controlador, podem establir la connexió amb la base de dades mitjançant el mètode *getConnection()* de la classe *DriverManager* determinada per la *url* que segons l'agent serà "jdbc:odbc:ESP1", "jdbc:odbc:ESP2", etc. aquest mètode ens retornarà un objecte de tipus *Connection*. Per exemple:

```
Connection con=DriverManager.getConnection(url);
```

El paràmetre URL ha de tenir la següent sintaxi:

```
jdbc:subprotocol//servidor:port/base de dades
```

Exemple:

```
String url="jdbc:odbc:ESP1";
```

La classe *Connection* és la que s'encarrega de la connexió amb la base de dades. Aquesta classe conté tots els mètodes que ens permetran llençar sentències SQL i obtenir els resultats de les sentències llançades.

La classe *Statement*

És l'encarregada d'enviar sentències SQL a la base de dades. Els resultats obtinguts es guarden en un objecte del tipus *ResultSet*. Per a obtenir una instància de la classe *Statement* associada a la connexió s'ha d'usar el mètode *createStatement()* de la classe *Connection* que ens retorna un objecte de la classe *Statement*. Exemple:

```
Statement stmt=con.createStatement();
```

Un cop instanciat, podem realitzar sentències SQL sobre la base de dades. Hi ha dos tipus de sentències:

- Sentències de modificació (UPDATE). Per a fer això ens cal el mètode de la classe *Statement*:

```
public abstract int executeUpdate(String sentènciaSQL);
```

- Sentències de consulta (QUERY), són sentències del tipus SELECT.

```
public abstract ResultSet executeQuery(String sentènciaSQL);
```

Obtenció de resultats

Quan llencem una sentència del tipus “*query*” obtenim un resultat que s’emmagatzema a un objecte del tipus *ResultSet*. Aquesta classe proporciona tots els mètodes necessaris per obtenir els resultats que s’han generat en llençar el “*query*”.

La interfície *ResultSet* contindrà les files obtingudes de l’execució de la sentència “*query*”. A la seva vegada les files es divideixen en columnes.

La interfície *ResultSet* conté un punter que apunta a la fila actual. Per passar d’una fila a una altra cal el mètode *next()*, aquest retorna un valor bolean indicant si existeixen més files. Un cop a la fila, podem obtenir les dades de les columnes, d’esquerra a dreta, amb el mètode *getXXX()*, on “XXX” és el tipus de dada Java de la columna. Exemple:

```
ResultSet rs=stmt.executeQuery( "SELECT nif FROM Clients C WHERE C.ciutat = “Reus”");
While (rs.next())
{
    String nif=rs.getString(“nif”);
}
```

Un cop finalitzat, s’aconsella cridar al mètode *close()* en tots els casos. Això alliberarà recursos abans no ho faci el recolector de brossa.

6.4 Obtenir un origen de dades ODBC

Els passos per registrar la base de dades són els que s’enumeren a continuació:

Passos:

- 1) Panel de control
- 2) ODBC Data Source Administrator
- 3) Etiqueta: User DSN
- 4) Add
- 5) Microsoft Access Driver->Finalizar
- 6) Seleccionar-> Indicar la base de dades que es vulgui registrar
- 7) Nombre del origen de datos-> posar el nom de la BD. Exemple ESP1
- 8) Aceptar

7 INICIALIZACIÓ DEL SISTEMA

7.1 Fitxers d'entrada

En aquest prototipus els agents Personals, ESP i d'Empresa necessiten d'uns fitxers per a obtenir les dades. Aquest fitxer, són fitxers de *Properties*, que emmagatzemen parells clau=valor per a cada atribut.

7.1.1 Fitxers d'entrada per l'agent personal

Per a poder obtenir les dades personals d'una persona que busca feina s'ha utilitzat un fitxer. Com ja hem dit, serà un fitxer de *Properties*. El nom del fitxer ha de ser el mateix que el de l'agent al qual pertany. L'estructura del fitxer ha de ser com la del següent exemple.

#Fitxer d'introduccio de dades referents a la persona

```
Name="Marc"
Surname1="Marti"
Surname2="Molet"
Nif="39723145"
Country="Spain"
Zip_code="49008"
City="Vic"
Street="Castell"
Street_number="39"
Phone="933231554"
Mobile_phone="639234724"
Email="mmarti@terra.es"
Day=23
Month=4
Year=1979
Sex=Male
Married=0
Degree=Enginyeria_tecnica
Especiality=ETIG
Qualification=Good
Experience=Two_years
Salary=1100
Hours=40
Contract_type=Permanent
Access=Middling
Informix=No
Oracle=Middling
MySQL=No
Ada=High
C++=High
Java=High
Lisp=Middling
Pascal=No
Prolog=No
MS_Dos=No
OS/2=No
Unix=Middling
```

```
Windows=Middling
Typing=No
Accountancy=Middling
English=Spoken
French=No
German=No
Leadership=Not_much
Motivation=Yes
Team_work=No
Project_management=Yes
Change_residence=Depend
Travel=Depend
Destination="Tarragona"
Driving_licence=1
Car=1
```

7.1.2 El fitxer d'entrada per l'agent ESP

Els agents ESP creen la sol·licitud de feina a partir d'uns fitxers en el que s'indica amb el valor 0 quin són els atributs que no s'inclouran a la seva sol·licitud, i amb el valor 1 els que si que s'inclouran. El nom del fitxer ha de ser el mateix que el de l'agent al qual pertany. Veiem-ne un exemple:

```
#Fitxer d'introduccio de dades referents a la sollicitud fe feina
```

```
Phone=1
Mobile_phone=0
Email=1
Sex=1
Married=0
Qualification=0
Salary=0
Hours=1
Contract_type=1
Leadership=0
Motivation=0
Team_work=1
Project_management=0
Change_residence=0
Destination=0
Travel=1
Driving_licemce=1
Car=0
```

7.1.3 Fitxer d'entrada per l'agent Empresa

Cada agent empresa tindrà un fitxer de *Properties* amb el mateix nom que l'agent al qual pertany on s'introduiran totes les dades referents a la feina.

Sempre s'haurà de començar amb una clau que indiqui el nombre de feines totals que es volen oferir. A continuació, hi haurà d'haver les dades referents a les feines, per a cada atribut haurem d'escriure el nom de la propietat seguit del caràcter “_” i el número de feina. Per exemple, l'identificador de la feina per la primera feina serà `work_id_0=10`,

mentre que per la segona feina serà `work_id_1=11`. Un exemple d'un fitxer d'aquest tipus és el que es pot veure a continuació.

```
#Fitxer d'introduccio de dades referents a la feina
#Cada feina te un identificador numeric
```

```
#DADES GLOBALS
Number_works=1
#Organ=cor
```

```
#WORK1
```

```
Work_id_0=10
Contract_type_0=Permanent
Hours_0=40
Salary_0=1000
City_0="Tarragona"
Studies_0=Enginyeria_tecnica
Especiality_0=ETIG
Experience_0=Two_years
Age_0=Between_18_and_25_years
Travel_0=1
Residence_0=0
Driving_licence_0=0
Car_0=0
Access_0=No
Informix_0=No
Oracle_0=Middling
MySQL_0=No
Ada_0=High
C++_0=Middling
Java_0=High
Lisp_0=Low
Pascal_0=No
Prolog_0=No
MS_Dos_0=No
OS/2_0=No
Unix_0=No
Windows_0=No
Typing_0=No
Accountancy_0=No
English_0=Read
French_0=No
German_0=No
Leadership_0=No
Motivation_0=Yes
Team_work_0=No
Project_management_0=No
```

7.2 Vocabulari utilitzat

En alguns casos els valors per a cada clau dels fitxers no poden ser qualssevol, sinó que han de ser uns valors concrets, si omplim els fitxers a partir d'una interfície gràfica creada expressament per introduir les dades no tindríem que preocupar-nos per a res, però si creem els fitxers manualment hem de tenir en compte que és molt important seguir el vocabulari que detallarem a continuació:

Valors possibles per la clau **Qualification**

Pass
Good
Excellent

Valors possibles per la clau **Experience**

No_experience
Less_than_a_year
One_year
Two_years
Three_years
Four_years
Between_5_and_10_years
More_than_10_years

Valors possibles per la clau **Age**

Between_18_and_25_years
Between_25_and_30_years
More_than_30_years

Valors possibles per la clau **Contract_type**

Part_time
Temporary
Permanent

Valors possibles per la clau **Degree**

Diplomatura
Enginyeria_tecnica
Llicenciatura
Enginyeria_superior
Doctorat

Valors possibles per la clau **Studies**

ETIG
ETIS
EI

Valors possibles per la clau **Access, Informix, Oracle, MySql, Ada, C++, Java, Lisp, Pascal, Prolog, MS DOS, OS/2, Unix, Windows, Typing, Accountancy**

No
Low
Middling
High

Valors possibles per la clau **English, French i German**

No
Read
Spoken
Written

Valors possibles per la clau **leadership, team work, motivation, project management**

Yes
Not_much
No

Valors possibles per la clau **Change_residence i Travel** (si es tracta de fitxers per agents Personals)

Yes
Depend
Not_specially
No

Valors possibles per la clau **Sex**

Male
Female

Valors possibles per la clau **Residence , Travel , Car, Driving_licence**(si es tracta de fitxers per agents Empresa) i Married (fitxers agents personal)

0 en cas negatiu 1 en cas afirmatiu

Cal esmentar que els strings que no corresponguin a alguns dels anteriors cal posar-los entre cometes “ ”, ja que Jade té problemes amb strings que tenen espais en blanc inclosos o que tenen segons quins caràcters.

7.3 Els fitxers de resultats

Cada agent Empresa crearà un fitxer de resultats, que també serà de tipus *properties*, on s'emmagatzemaran l'identificador, el nif i el valor dels atributs de cada un dels candidats abans de fer les valoracions de l'adequació a una feina.

```
#Applicant: 0 ("39723145") for work 10
#Tue Sep 17 02:12:02 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Two_years
Pascal=Good
English=Good
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
```

```
Access=Good
Qualification=Good
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 1 ("39722235") for work 10
#Tue Sep 17 02:12:02 CEST 2002
...
#Applicant: 2 ("39843561") for work 10
#Tue Sep 17 02:12:02 CEST 2002
...
#Applicant: 3 ("39767295") for work 10
#Tue Sep 17 02:12:02 CEST 2002
...
#Applicant: 4 ("39822234") for work 10
#Tue Sep 17 02:12:02 CEST 2002
...
```

En finalitzar el procés d'ordenació s'emmagatzemaran els resultats obtinguts amb la valoració per a cada agent.

```
#Results of valuing the applicants for the work 10
#Tue Sep 17 02:13:40 CEST 2002
fairly_good=3 ("39767295") Personal5@ads1\1099/JADE
very_good=4 ("39822234") Personal4@ads1\1099/JADE
recommendable=1 ("39722235") Personal3@ads1\1099/JADE
good=2 ("39843561") Personal2@ads1\1099/JADE
adequate=0 ("39723145") Personal1@ads1\1099/JADE
```

8 JOC DE PROVES

A continuació mostrarem algunes proves realitzades que afecten al comportament general del sistema.

Prova 1

Primerament, s'ha realitzat una prova general del prototipus. Aquesta prova consistirà en engegar un conjunt d'agents Personals, d'empresa i ESPs mitjançant l'agent Simulador. L'agent Simulador s'encarrega d'engegar tots els altres agents que intervindran en la prova. Simula com anirien entrant nous agents al sistema i quins seria el funcionament del prototipus en un cas real. Per a aconseguir això va engegant agents cada un cert.

L'agent Simulador engegarà els següents agents:

- 20 agents Personals, aquests agents intentaran trobar una feina.
- 3 agents d'Empresa, aquests agents intentaran trobar treballadors.
- 3 agents ESP (Empresa de Selecció de Personal), aquests agents faran de mitjancers entre els agents Personals i els agents d'Empreses que cerquen treballadors.
- 1 agent ClusDMA, agent que realitzarà l'ordenació dels candidats de millor a pitjor.

En aquesta prova s'ha aprofitat el fet que es posen en marxa bastants agents Personals per a crear, no sols agents Empresa que ofereixin una feina, sinó que també hi ha un agent Empresa que ofereix més d'una feina.

Farem un seguiment de l'exemple a partir d'una de les empreses: Les dades de les condicions de la feina són les següents:

Empresa: Emp1

```
#Fitxer d'introduccio de dades referents a la feina
#Cada feina te un identificador numeric

#DADES GLOBALS
Number_works=1

#WORK1

Work_id_0=10
Contract_type_0=Permanent
Hours_0=40
Salary_0=1000
City_0="Tarragona"
Studies_0=Enginyeria_tecnica
```

Especiality_0=ETIG
Experience_0=Two_years
Age_0=Between_18_and_25_years
Travel_0=1
Residence_0=0
Driving_licence_0=0
Car_0=0
Access_0=No
Informix_0=No
Oracle_0=Middling
MySQL_0=No
Ada_0=High
C++_0=Middling
Java_0=High
Lisp_0=Low
Pascal_0=No
Prolog_0=No
MS_Dos_0=No
OS/2_0=No
Unix_0=No
Windows_0=No
Typing_0=No
Accountancy_0=No
English_0=Read
French_0=No
German_0=No
Leadership_0=No
Motivation_0=Yes
Team_work_0=No
Project_management_0=No

Un cop feta la pre-selecció per part dels agents ESP les persones adequades per aquesta feina, es veu que els pre-seleccionats són els agents següents, les dades d'entrada aquests agents Personals són les següents:

Els fitxers d'entrada per als agents Personals són:

Personal1.log

Name="Marc"
Surname1="Marti"
Surname2="Molet"
Nif="39723145"
Country="Spain"
Zip_code="49008"
City="Vic"
Street="Castell"
Street_number="39"
Phone="933231554"
Mobile_phone="639234724"
Email="mmarti@terra.es"
Day=23
Month=4
Year=1979
Sex=Male
Married=0
Degree=Enginyeria_tecnica
Especiality=ETIG
Qualification=Good
Experience=Two_years

Salary=1100
Hours=40
Contract_type=Permanent
Access=Middling
Informix=No
Oracle=Middling
MySql=No
Ada=High
C++=High
Java=High
Lisp=Middling
Pascal=No
Prolog=No
MS_Dos=No
OS/2=No
Unix=Middling
Windows=Middling
Typing=No
Accountancy=Middling
English=Spoken
French=No
German=No
Leadership=Not_much
Motivation=Yes
Team_work=No
Project_management=Yes
Change_residence=Depend
Travel=Depend
Destination="Tarragona"
Driving_licence=1
Car=1

Personal2

Name="Carles"
Surname1="Ferrer"
Surname2="Boada"
Nif="39843561"
Country="Spain"
Zip_code="08720"
City="Vilafranca del Penedes"
Street="Major"
Street_number="3"
Phone="934443152"
Mobile_phone="639299724"
Email="cferrer@terra.es"
Day=8
Month=9
Year=1979
Sex=Male
Married=0
Degree=Enginyeria_tecnica
Speciality=ETIG
Qualification=Good
Experience=Three_years
Salary=1150
Hours=40
Contract_type=Permanent
Access=Middling
Informix=No

Oracle=High
MySQL=No
Ada=High
C++=Middling
Java=High
Lisp=Middling
Pascal=No
Prolog=No
MS_Dos=No
OS/2=No
Unix=Middling
Windows=Middling
Typing=No
Accountancy=Middling
English=Spoken
French=No
German=No
Leadership=Not_much
Motivation=Yes
Team_work=No
Project_management=Yes
Change_residence=Depend
Travel=Depend
Destination="Tarragona"
Driving_licence=1
Car=1

Personal3

Name="Maria"
Surname1="Cases"
Surname2="Vidal"
Nif="39722235"
Country="Spain"
Zip_code="43001"
City="Tarragona"
Street="Mallorca"
Street_number="17"
Phone="977237634"
Mobile_phone="639756294"
Email="mcases@hotmail.com"
Day=28
Month=12
Year=1978
Sex=Female
Married=0
Degree=Enginyeria_tecnica
Especiality=ETIG
Qualification=Good
Experience=Two_years
Salary=1100
Hours=40
Contract_type=Permanent
Access=High
Informix=No
Oracle=Middling
MySQL=No
Ada=High
C++=High
Java=High
Lisp=Low

Pascal=No
Prolog=No
MS_Dos=No
OS/2=No
Unix=Middling
Windows=Middling
Typing=No
Accountancy=Middling
English=Spoken
French=No
German=No
Leadership=Not_much
Motivation=Yes
Team_work=No
Project_management=Yes
Change_residence=Depend
Travel=Depend
Destination="Tarragona"
Driving_licence=1
Car=1

Personal 4

Name="Pere"
Surname1="Tobau"
Surname2="Lleo"
Nif="39822234"
Country="Spain"
Zip_code="43764"
City="El Catllar"
Street="Major"
Street_number="1"
Phone="977447634"
Mobile_phone="649226294"
Email="petll@hotmail.com"
Day=1
Month=1
Year=1978
Sex=Male
Married=1
Degree=Enginyeria_tecnica
Especiality=ETIG
Qualification=Good
Experience=Four_years
Salary=1000
Hours=40
Contract_type=Permanent
Access=High
Informix=No
Oracle=High
MySQL=No
Ada=High
C++=High
Java=High
Lisp=Middling
Pascal=No
Prolog=No
MS_Dos=No
OS/2=No
Unix=Middling
Windows=Middling

Typing=No
Accountancy=Middling
English=Spoken
French=No
German=No
Leadership=Yes
Motivation=Yes
Team_work=No
Project_management=Yes
Change_residence=No
Travel=Not_specially
Destination="Tarragona"
Driving_licence=1
Car=1

Personal 5

Name="Sandra"
Surname1="Rius"
Surname2="Perez"
Nif="39767295"
Country="Spain"
Zip_code="43815"
City="Santes Creus"
Street="Monestir"
Street_number="5"
Phone="977223344"
Mobile_phone="685226664"
Email="sandra@tinet.org"
Day=1
Month=2
Year=1980
Sex=Female
Married=0
Degree=Enginyeria_tecnica
Speciality=ETIG
Qualification=Excellent
Experience=Two_years
Salary=1050
Hours=40
Contract_type=Permanent
Access=Low
Informix=No
Oracle=Middling
MySQL=No
Ada=High
C++=Middling
Java=High
Lisp=Middling
Pascal=No
Prolog=No
MS_Dos=No
OS/2=No
Unix=Middling
Windows=Middling
Typing=No
Accountancy=Middling
English=Spoken
French=No
German=No
Leadership=Yes

```
Motivation=Yes
Team_work=No
Project_management=Yes
Change_residence=No
Travel= Yes
Destination="Tarragona"
Driving_licence=1
Car=1
```

A partir d'això podem veure que el candidat més adequat per la feina és el Personal4, amb nif : 39822234. Tot i que les altres persones tenen característiques molt similars. La persona propietària de l'agent Personal4 té més experiència, va obtenir una millor qualificació en els seus estudis, té molta disponibilitat per viatjar i desitja un salari similar a l'ofert.

Quan l'agent Empresa obté els candidats i aconsegueix totes les dades necessàries contacta amb l'agent ClusDMA perquè faci l'ordenació.

A continuació es mostra el fitxer de resultats dels agents Empresa (abans i després de fer-se l'ordenació). En cada un es mostra, per a cada candidat a una feina, els valors qualitius per a cada atribut que intervenen en el procés d'ordenació. Al final del fitxer es reflexen els resultats de l'ordenació amb el grau adequació del candidat amb la feina, el nif del candidat i el nom de l'agent que el representa.

Emp1Result.log

```
#Applicant: 0 ("39723145") for work 10
#Tue Sep 24 21:23:40 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Two_years
Pascal=Good
English=Good
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Good
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
```

Ada=Good
#Applicant: 1 ("39722235") for work 10
#Tue Sep 24 21:23:40 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Two_years
Pascal=Good
English=Good
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Good
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 2 ("39843561") for work 10
#Tue Sep 24 21:23:40 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Three_years
Pascal=Good
English=Good
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Good
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good

Prolog=Good
Ada=Good
#Applicant: 3 ("39767295") for work 10
#Tue Sep 24 21:23:40 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Two_years
Pascal=Good
English=Good
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Borderline
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Good
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 4 ("39822234") for work 10
#Tue Sep 24 21:23:40 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Four_years
Pascal=Good
English=Good
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Good
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Excellent
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good

Salary=Very_Good
Prolog=Good
Ada=Good
#Results of valuing the applicants for the work 10
#Tue Sep 24 21:23:43 CEST 2002
fairly_good=2 ("39843561") Personal2@adsl\1099/JADE
very_good=4 ("39822234") Personal4@adsl\1099/JADE
recommendable=1 ("39722235") Personal3@adsl\1099/JADE
good=3 ("39767295") Personal5@adsl\1099/JADE
adequate=0 ("39723145") Personal1@adsl\1099/JADE

Emp2Result.log

#Applicant: 0 ("39012289") for work 20
#Tue Sep 24 17:49:59 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Between_5_and_10_years
Pascal=Good
English=Good
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Pass
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 1 ("39040302") for work 20
#Tue Sep 24 17:49:59 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Between_5_and_10_years
Pascal=Good
English=Good
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good
Windows=Good

OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Borderline
Qualification=Good
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 2 ("39112287") for work 20
#Tue Sep 24 17:49:59 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Between_5_and_10_years
Pascal=Good
English=Good
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Borderline
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Excellent
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 3 ("39440302") for work 20
#Tue Sep 24 17:49:59 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=More_than_10_years
Pascal=Good
English=Borderline
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Borderline
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good

```
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Borderline
Qualification=Good
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 4 ("39887295") for work 20
#Tue Sep 24 17:49:59 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Between_5_and_10_years
Pascal=Good
English=Good
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Borderline
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Good
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Results of valuing the applicants for the work 20
#Tue Sep 24 17:50:02 CEST 2002
fairly_good=1 ("39040302") Personal9@ads1\ :1099/JADE
very_good=4 ("39887295") Personal6@ads1\ :1099/JADE
excellent=2 ("39112287") Personal7@ads1\ :1099/JADE
recommendable=3 ("39440302") Personal10@ads1\ :1099/JADE
good=0 ("39012289") Personal8@ads1\ :1099/JADE
```

Emp3Result.log

```
#Applicant: 0 ("39770306") for work 30
#Tue Sep 24 17:50:28 CEST 2002
Hours=Very_Good
Project_management=Good
```

Experience=Between_5_and_10_years
Pascal=Good
English=Borderline
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Good
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Excellent
Informix=Good
C++=Borderline
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 1 ("39770301") for work 30
#Tue Sep 24 17:50:28 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=One_year
Pascal=Good
English=Borderline
Lisp=Good
Motivation=Acceptable
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Acceptable
MS_DOS=Good
Change_residence=Good
Windows=Borderline
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Pass
Informix=Good
C++=Borderline
Java=Borderline
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 2 ("39770308") for work 30
#Tue Sep 24 17:50:28 CEST 2002
Hours=Very_Good

Project_management=Good
Experience=One_year
Pascal=Good
English=Borderline
Lisp=Good
Motivation=Acceptable
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Acceptable
MS_DOS=Good
Change_residence=Good
Windows=Borderline
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Good
Informix=Good
C++=Borderline
Java=Borderline
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 3 ("39770304") for work 30
#Tue Sep 24 17:50:28 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Two_years
Pascal=Good
English=Borderline
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Good
German=Good
Leadership=Good
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Borderline
Oracle=Good
Access=Good
Qualification=Excellent
Informix=Good
C++=Borderline
Java=Acceptable
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 4 ("39770300") for work 30
#Tue Sep 24 17:50:28 CEST 2002

```
Hours=Very_Good
Project_management=Good
Experience=No_experience
Pascal=Good
English=Borderline
Lisp=Good
Motivation=Acceptable
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Acceptable
MS_DOS=Good
Change_residence=Good
Windows=Borderline
OS/2=Good
MySQL=Good
French=Good
Unix=Borderline
Oracle=Good
Access=Good
Qualification=Pass
Informix=Good
C++=Borderline
Java=Bad
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 0 ("39060306") for work 31
#Tue Sep 24 17:50:29 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Four_years
Pascal=Good
English=Borderline
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Acceptable
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Borderline
Oracle=Good
Access=Good
Qualification=Good
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Borderline
Ada=Good
#Applicant: 1 ("39070307") for work 31
```

```
#Tue Sep 24 17:50:29 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Four_years
Pascal=Good
English=Bad
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Acceptable
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Borderline
Oracle=Good
Access=Good
Qualification=Pass
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Borderline
Ada=Good
#Applicant: 2 ("39080308") for work 31
#Tue Sep 24 17:50:29 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Between_5_and_10_years
Pascal=Good
English=Good
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Acceptable
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Excellent
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
```

```
#Applicant: 3 ("39090309") for work 31
#Tue Sep 24 17:50:29 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Less_than_a_year
Pascal=Good
English=Borderline
Lisp=Good
Motivation=Bad
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Acceptable
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Borderline
Oracle=Borderline
Access=Good
Qualification=Pass
Informix=Good
C++=Borderline
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Good
Ada=Good
#Applicant: 4 ("39050302") for work 31
#Tue Sep 24 17:50:29 CEST 2002
Hours=Very_Good
Project_management=Good
Experience=Between_5_and_10_years
Pascal=Good
English=Borderline
Lisp=Good
Motivation=Good
Team_work=Good
Travel=Acceptable
German=Good
Leadership=Acceptable
MS_DOS=Good
Change_residence=Good
Windows=Good
OS/2=Good
MySQL=Good
French=Good
Unix=Good
Oracle=Good
Access=Good
Qualification=Good
Informix=Good
C++=Good
Java=Good
Accountancy=Good
Typing=Good
Salary=Very_Good
Prolog=Borderline
```

```
Ada=Good
#Results of valuing the applicants for the work 30
#Tue Sep 24 17:50:34 CEST 2002
fairly_good=2 ("39770308") Personal19@adsl\1099/JADE
very_good=0 ("39770306") Personal18@adsl\1099/JADE
recommendable=1 ("39770301") Personal17@adsl\1099/JADE
good=3 ("39770304") Personal20@adsl\1099/JADE
adequate=4 ("39770300") Personal16@adsl\1099/JADE
#Results of valuing the applicants for the work 31
#Tue Sep 24 17:50:36 CEST 2002
fairly_good=0 ("39060306") Personal12@adsl\1099/JADE
very_good=2 ("39080308") Personal14@adsl\1099/JADE
recommendable=1 ("39070307") Personal13@adsl\1099/JADE
good=4 ("39050302") Personal11@adsl\1099/JADE
adequate=3 ("39090309") Personal15@adsl\1099/JADE
```

Els resultats obtinguts per tots els agents empresa han estat els següents:

- La feina amb identificador 10, ha estat ocupada per la persona amb nif 39822234, l'agent que representa a aquesta persona és l'agent Personal4.
- La feina amb identificador 20, ha estat ocupada per la persona amb nif 39112287, l'agent que representa a aquesta persona és l'agent Personal7.
- La feina amb identificador 30, ha estat ocupada per la persona amb nif 39770306, l'agent que representa a aquesta persona és l'agent Personal18.
- La feina amb identificador 31, ha estat ocupada per la persona amb nif 39080308, l'agent que representa a aquesta persona és l'agent Personal14.

Prova 2

En aquest cas hem realitzat una prova en un cas extrem. Suposem que tenim una empresa que ofereix una feina, l'agent d'aquesta empresa demanarà als agents ESP amb qui ha connectat que li proporcionin una llista de tots els possibles candidats per la feina que tinguin a les seves bases de dades. Pot passar que cap empresa de selecció trobi candidats adequats per la feina i, per tant, l'empresa no rebi cap candidat. En aquest cas, l'agent de l'empresa que ofereix la feina tornarà a demanar que es busquin candidats, ja que mentrestant podrien haver arribat noves persones que cerquen feina que sí que podrien ser candidates a la feina. Si no en troba repetirà aquest procés les vegades que calgui.

Per a realitzar la prova hem utilitzat un agent Empresa que ofereix una feina per la qual no hi ha cap persona educada. El comportament de l'agent empresa és el de demanar una vegada i una altra als agents ESP que busquin candidats.

Podem veure la seqüència de missatges enviats en la Fig. 18

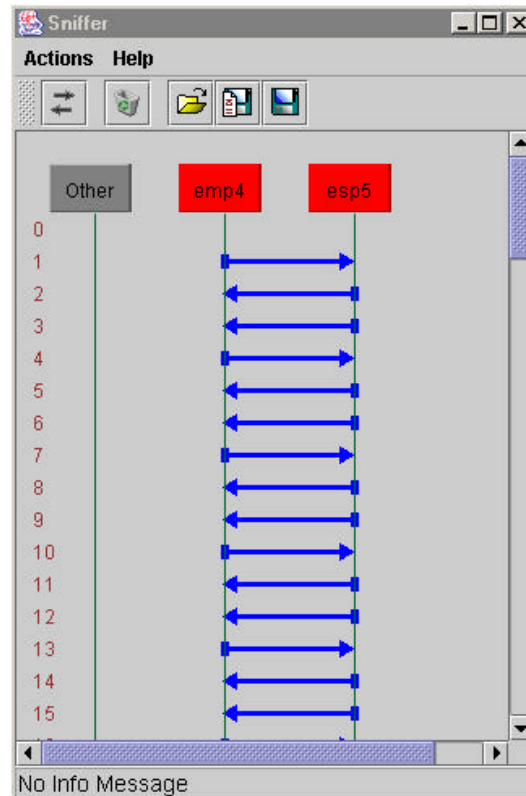


Fig. 18

Prova 3

En aquesta prova es pretén observar un comportament similar a l'anterior, però en un cas diferent.

Quan un agent empresa rep una llista de candidats es comunica amb l'agent ClusDMA perquè aquest li realitzi una ordenació dels candidats, però aquest agent necessita un mínim de cinc candidats per a poder realitzar aquesta ordenació, també es podrien produir fallades o es podria refusar el missatge per algun altre motiu. Una possible solució a aquest problema és fer que l'agent Empresa torni a demanar candidats a les ESP quan per algun motiu l'agent ClusDMA no pot realitzar la seva tasca. Si no es controlés aquest cas l'agent Empresa es quedaria aturat i mai podria trobat un treballador que ocupés el lloc de treball.

Per a realitzar la prova em engegat un conjunt d'agents Personals i un agent Empresa, en aquest cas el nombre de candidats era menor que cinc i l'agent ClusDMA no podia realitzar l'ordenació. El comportament de l'agent empresa ha estat demanar repetidament als agents ESPs que li proporcionessin una llista de candidats, llavors l'agent Empresa es posava en contacte amb l'agent ClusDMA perquè realitzés l'ordenació, com que aquest agent no podia, l'agent Empresa tornava a demanar a les ESP que busquessin candidats.

Podem veure la seqüència de missatges enviats en la Fig. 19

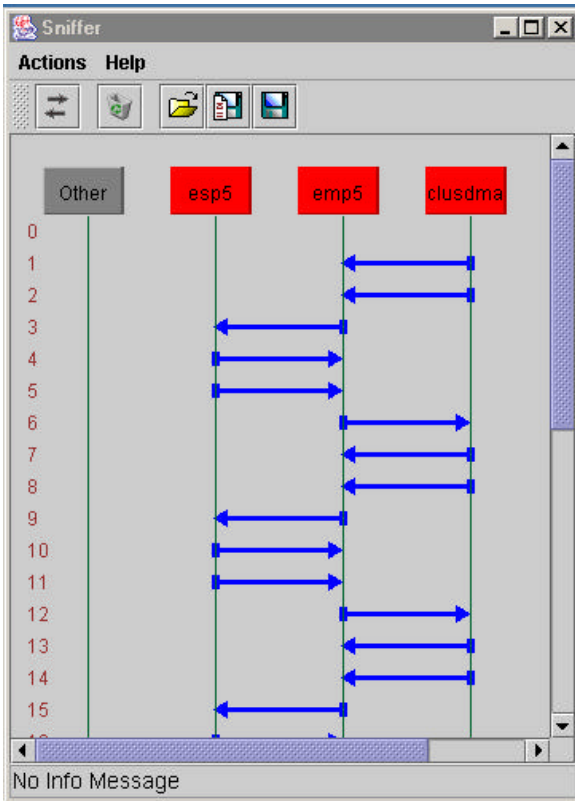


Fig. 19

9 TREBALLS FUTURS

L'objectiu d'aquest projecte era dissenyar un sistema Multi-Agent que servís per a realitzar la interacció entre les empreses de selecció de personal, les empreses que cercaven treballadors i les persones que buscaven feina. L'objectiu es pot dir que s'ha aconseguit, però es podrien realitzar millores.

Com a treballs futurs podríem enumerar:

- Agent ClusDMA necessita un mínim de cinc candidats per a realitzar la valoració de l'adequació de cada candidat amb una feina. Es podria implementar mitjançant tècniques d'intel·ligència artificial, una manera d'ordenar els candidats quan el seu nombre és inferior a cinc.
- Quan una empresa busca personal, hem vist que ha de seleccionar un grup d'ESPs per a fer de mitjanceres. Actualment aquesta selecció es fa de manera aleatòria, però és podria tenir un criteri de selecció pel qual cada empresa guardés, cada cop que utilitza un servei d'una ESP, quants candidats ha rebut i si ha trobat algun candidat idoni o no; així, l'empresa podria anar aprenent quines són les ESPs més adequades a les seves demandes de treball.
- Cada ESP pot recollir dades del nombre de persones apuntades, nombre d'empreses que han utilitzat el servei, nombre de feines ofertes, nombre de persones que han trobat feina, etc. Per a recollir aquesta informació caldria realitzar consultes simples a les bases de dades de les ESP. Amb totes aquestes dades es podrien elaborar estadístiques.

10 VALORACIÓ

Realitzar aquest projecte m'ha suposat adquirir molts coneixements nous d'Agents i Sistemes Multi-Agent. Tot i que, després d'haver cursat l'assignatura d'Intel·ligència Artificial tenia nocions d'aquest tema, s'ha d'agrair el fet que el GruSMA organitzés sessions per a donar una base teòrica a les persones que volien realitzar un projecte basat en Sistemes Multi-Agent.

També, han estat de molta utilitat per a la realització d'aquest projecte els coneixements obtinguts en assignatures com Bases de Dades, pels coneixements de SQL obtinguts, i Tecnologia de la Programació, ja que va proporcionar una bona base en la programació en Java.

Cal dir, que haver d'utilitzar els agents implementats per David Sánchez m'ha donat l'oportunitat de conèixer el seu projecte i d'adquirir una idea general dels MCDM (Mètodes de Decisió Multi-Criteri).

En aquest projecte, mitjançant Sistemes Multi-Agent s'ha pogut desenvolupar una aplicació distribuïda, dotada de molta flexibilitat, on els diversos agents treballen conjuntament per a aconseguir els seus respectius objectius.

Finalment, voldria agrair a la meva tutora Aïda Valls l'interès demostrat en aquest projecte i als meus companys de GruSMA per les seves aportacions amb projectes sobre Sistemes Multi-Agent.

11 BIBLIOGRAFIA

[FIPA]

Foundation for Intelligent Physical Agents, web page: <http://www.fipa.org>

[FIPA, 2001]

Foundation for Intelligent Physical Agents (2001), *FIPA Device Ontology*, FIPA

[FIPA, 2000a]

Foundation for Intelligent Physical Agents (2000), *FIPA ACL Message Structure*, FIPA

[FIPA, 2000b]

Foundation for Intelligent Physical Agents (2000), *FIPA Agent Management Specification*, FIPA

[FIPA, 2000c]

Foundation for Intelligent Physical Agents (2000), *FIPA SL Content Language Specification*, FIPA

[FIPA, 1998a]

Foundation for Intelligent Physical Agents (1998), *Agent Communication Language*, FIPA

[FIPA, 1998b]

Foundation for Intelligent Physical Agents (1998), *Agent Management*, FIPA

[FIPA, 1998c]

Foundation for Intelligent Physical Agents (1998), *FIPA97 Developers Guide*, FIPA

[GruSMA]

GruSMA, Grup de Sistemes Multi-Agent, DEIM, Universitat Rovira i Virgili, plana web: <http://www.etse.urv.es/recerca/banzai/toni/MAS>

[Isern, 2000]

Isern, D. (2000), *Desenvolupament de Sistemes Multiagent: JADE*, apunts GruSMA, Universitat Rovira i Virgili

[Sánchez, 2001]

Sánchez, D. (2001), *Un Sistema Multi-Agent d'ajuda a l'assignació d'òrgans en trasplantaments. El mètode ClusDM amb criteris lingüístics*. Projecte Final de Carrera, Enginyeria Tècnica en Informàtica de Sistemes, Universitat Rovira i Virgili

[Valls et al., 2002]

Valls, A., Moreno, A., Sánchez, D. (2002), A multicriteria decision aid agent applied to the selection of the best receiver in a transplant, *4th International conference on Enterprise Information Systems*, pp. 431-438, Ciudad Real, Spain