

Sistema multiagent difús amb persistència de resultats

José Antonio Caballero Pacheco

Grau d'Enginyeria Informàtica

Intel·ligència artificial

David Isern Alarcón

Carles Ventura Royo

02/01/2018



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Sistema multiagent difús amb persistència de resultats</i>
Nom de l'autor:	<i>José Antonio Caballero Pacheco</i>
Nom del consultor/a:	<i>David Isern Alarcón</i>
Nom del PRA:	<i>Carles Ventura Royo</i>
Data de lliurament (mm/aaaa):	<i>01/2018</i>
Titulació o programa:	<i>Grau d'Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Intel·ligència artificial</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>JADE, Lògica difusa, parametritzable</i>

Resum del Treball (màxim 250 paraules): *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

Treball orientat a l'anàlisi, disseny i implementació d'un sistema multiagent amb agents genèrics que presenten capacitat de raonament: Concretament, raonament basat en lògica difusa. La intencionalitat del projecte és oferir una eina configurable amb l'objectiu d'ajudar a l'usuari a agilitzar el desenvolupament de nous sistemes d'inferència, provar els existents, simular sistemes difusos jeràrquics, executar diferents escenaris alhora, comparar resultats, etc. Tot plegat, d'una manera autònoma, configurable, genèrica i amb persistència de la informació generada a la sessió de treball.

Respecte a la metodologia aplicada al projecte, a partir d'una llista d'objectius extensa, es va seleccionar, prioritzar i planificar els punts més interessants. Incrementalment el treball s'ha desenvolupat fins complir amb tots els objectius principals i per tant, és factible considerar que els resultats obtinguts són satisfactoris. Així doncs, s'ha desenvolupat un SMA on a partir d'un descriptor de configuració, s'inicia un entorn multiagent JADE amb múltiples agents que realitzen raonaments difusos i poden interactuar entre ells, amb usuaris i altres SMA. Els agents genèrics adapten la seva morfologia: Variables d'entrada, Bloc de regles i altres a partir dels sistemes d'inferència associats.

Les conclusions resultants d'aquest projecte van més enllà d'uns resultats concrets, ja que es tracta d'un projecte de naturalesa genèrica. Dit això es pot destacar com es demostra la genuïnitat d'un enfocament genèric multiagent, ja que, d'una banda

permet desenvolupar o provar sistemes d'inferència de qualsevol domini d'aplicació i de l'altre, ajuda a l'usuari a treballar de manera més efectiva, maximitzant l'obtenció de resultats.

Abstract (in English, 250 words or less):

The purpose of this work is analyze, design and develop a multiagent system based on generic agents able to perform Fuzzy inferences. So the target of this proposal is deliver a multi agent system tool in order to ease the development of new inference systems, serve as test platform, help simulate hierarchical fuzzy systems, compare results and so on. All together in a generic and parametrical fashion and persisting all the generated data into a relational database.

In regards of the methodology, a huge list of targets was born but only some of these were considered as main points, consequently, the focus of the project was oriented in order to develop all these. The obtained application is working based on those main targets, so we can assume that the results are good. Besides a parametric and generic MAS have been developed: Starting with an XML descriptor file, the tool is able to setup and start multiple generic agents and make them interact among them or with the user. In addition to that, agents are having the ability to morph based on the FCL files associated.

As a result some conclusions can be made from all this. In fact, it is not a matter of providing some specific results based in a domain, because of the generic nature of the project. However can be noted the usefulness of this application in terms of provide a valuable ease to the users about testing, simulate and develop new fuzzy inference systems, in a parametric, generic and autonomous manner and make the mission of getting useful outcomes, be bit easier.

Índex

1	Introducció	1
1.1	Context i justificació del Treball.....	1
1.1.1	Temàtica escollida	1
1.1.2	Problemàtica a resoldre	1
1.2	Objectius del Treball: Pre-planificació	3
1.3	Objectius del Treball: Post-planificació	5
1.4	Enfocament i mètode seguit	7
1.4.1	Planificació del treball tipus waterfall: Agilització.....	7
1.4.2	Repositori de codi font i control de versions.....	8
1.4.3	Desenvolupament del software: Best Practices	9
1.5	Planificació del Treball	9
1.5.1	Calendari	9
1.5.2	Planificació	10
1.5.3	Costos estimats del projecte	11
1.5.4	Riscos potencials i accions de mitigació	12
1.5.5	Legal	12
1.6	Breu sumari de productes obtinguts	13
1.7	Breu descripció dels altres capítols de la memòria	13
2	Materialització del projecte	14
2.1	Contextualització respecte la Intel·ligència artificial	14
2.1.1	Introducció als agents intel·ligents	14
2.1.2	Raonament aproximat amb Sistemes difusos	15
2.2	Possibilitats del projecte	20
2.3	Entorn de treball pel disseny i implementació del SMA: JADE	22
2.3.1	Introducció.....	22

2.3.2	Perquè JADE?	22
2.3.3	Java Agent Development Framework: Perspectiva general	23
2.3.4	Agents JADE	24
2.3.5	Comportaments dels agents JADE	25
2.3.6	Particularitats dels agents JADE	26
2.3.7	Comunicació dels agents JADE	27
2.4	Implementacions teoria conjunts difusos i lògica difusa	28
2.4.1	Representació dels sistemes d'inferència	28
2.4.2	Implementacions	30
2.5	Desenvolupament del sistema Multi agent.....	32
2.5.1	Perspectiva de l'arquitectura del sistema multi agent.....	32
2.5.2	Descriptor del Sistema multi agent	33
2.5.3	Plataforma multiagent amb JADE	36
2.5.4	Implementació JFuzzyLogic.....	49
2.5.5	Persistència d'interaccions i resultats	50
2.6	Obstacles principals al llarg de la vida del projecte	51
3	Resultats obtinguts	52
3.1	Primer escenari: Agent genèric únic	52
3.1.1	Descriptor del sistema multiagent	53
3.1.2	Sistema d'inferència difús	53
3.1.3	Inicialització del sistema multi agent	54
3.1.4	Interaccions del sistema multi agent	55
3.2	Segon escenari: Múltiples agents genèrics	56
3.2.1	Descriptor del sistema multi agent	57
3.2.2	Sistema d'inferència difús	57
3.2.3	Inicialització del sistema multi agent	58
3.2.4	Interaccions del sistema multi agent	59
3.3	Tercer escenari: Múltiples agents genèrics (Variació)	61
3.3.1	Descriptor del sistema multi agent	61

3.3.2	Sistema d'inferència difús	62
3.3.3	Inicialització del sistema multi agent	62
3.3.4	Interaccions del sistema multi agent	62
3.4	Quart escenari: Fine-tunning d'un sistema d'inferència amb agents	62
3.4.1	Descriptor del sistema multiagent	62
3.4.2	Sistema d'inferència difús	63
3.4.3	Inicialització del sistema multiagent	64
3.4.4	Interaccions del sistema multi agent	64
4	Millores i treball posterior	65
5	Anàlisi dels resultats i conclusions	66
6	Bibliografia.....	67
7	Annexos	68
7.1	Aspectes transversals.....	68
7.1.1	Preparació del model de dades de la Base de dades	68
7.1.2	Logging.....	69
7.2	Especificació model: Persistència de informació.....	69
7.3	Manual d'instal·lació del sistema multiagent	70
7.3.1	Requisits de maquinari	70
7.3.2	Requisits de programari.....	70
7.3.3	Instal·lació	70
7.4	Descripció del projecte Eclipse i creació de noves versions.....	71
7.4.1	Perspectiva del projecte	71
7.4.2	Repositori de codi.....	72
7.4.3	Nova versió del sistema multi agent	72
7.5	Especificació completa model Descriptor.....	73

Llista de figures

Figura 1: Gràfica tasques generals/cost.....	11
Figura 2: Planificació de les tasques superiors del projecte	11
Figura 3: Variables lingüístiques amb els valors lingüístics i els conjunts que els representen	17
Figura 4: Funcions de pertinença més habituals	17
Figura 5: Arquitectura típica d'un Sistema multiagent JADE	23
Figura 6: Arquitectura del sistema multiagent desenvolupat	33
Figura 7: Esquema descriptor del sistema multiagent.....	34
Figura 8: Perspectiva de les interaccions del sistema multi agent.....	37
Figura 9: Inicialització de l'agent Hive	38
Figura 10: Comportament ParallelBehaviour de l'agent Hive	39
Figura 11: Inicialització de l'agent genèric, BaseAgent	40
Figura 12: Comportament SetupBehaviour de l'agent genèric	42
Figura 13: Comportament ParallelBehaviour de l'agent genèric.....	43
Figura 14: Comportament PrimeBehaviour de l'agent genèric	45
Figura 15: Interfície principal del sistema multiagent.....	47
Figura 16: Interfície dels agents genèrics.....	48
Figura 17: Disseny conceptual de la base de dades	51
Figura 18: Descriptor SMA del primer escenari del manual.....	53
Figura 19: Fitxer FCL del primer escenari del manual.....	54
Figura 20: Interfície principal del primer escenari del manual.....	55
Figura 21: Interfície de l'agent del primer exemple, sense valors.	56
Figura 22: Intefície de l'agent del primer exemple, amb valors.....	56
Figura 23: Descriptor SMA del segon escenari del manual	57
Figura 24: Fitxer FCL de l'agent A1 del segon escenari del manual.....	58
Figura 25: Fitxer FCL de l'agent A2 del segon escenari del manual.....	58
Figura 26: Interfície principal del SMA del segon escenari del manual.....	59
Figura 27: Interfície de l'agent A1 del segon escenari del manual.....	59
Figura 28: Interfície de l'agent A2 del segon escenari del manual.....	59
Figura 29: Interfície de l'agent A1 del segon escenari del manual, amb valors.....	60
Figura 30: Interfície de l'agent A2 del segon escenari del manual, amb valors.....	60
Figura 31: Interffície de l'agent A1 del segon escenari del manual, responent.....	60

Figura 32: Interfície de l'agent A2 del segon escenari del manual, sol·licitant	60
Figura 33: Interfície de l'agent A1 del segon escenari del manual, responent	61
Figura 34: Interfície de l'agent A2 del segon escenari del manual sol·licitant	61
Figura 35: Interfície de l'agent A2 del segon escenari del manual, cas particular.....	61
Figura 36: Interfície principal del SMA, segons el segon escenari del manual	61
Figura 37: Petita modificació del descriptor, tercer escenari del manual	61
Figura 38: Interfície de l'agent A1 del tercer escenari del manual	62
Figura 39: Interfície de l'agent A2 del tercer escenari del manual	62
Figura 40: Descriptor del SMA del quart escenari del manual	63
Figura 41: Interfície principal del SMA al quart escenari del manual	64
Figura 42: Interfícies dels agents genèrics del quart escenari del manual.....	64
Figura 43: Interfície dels agents genèrics del quart escenari del manual, amb valors .	65
Figura 44: Disseny lògic de la base de dades	69
Figura 45: Directoris de treball del sistema multiagent	71
Figura 46: Estructura del projecte Eclipse	71

1 Introducció

La introducció pretén il·lustrar els aspectes fonamentals de la proposta en termes: Motivació, objectius, planificació entre altres.

1.1 Context i justificació del Treball

1.1.1 Temàtica escollida

Vaig decidir començar a estudiar a la universitat per ampliar els meus coneixements i poder aspirar a camps de la informàtica tan vius com el de la intel·ligència artificial, amb la idea de trobar-me sempre motivat i poder aspirar a canvis professionals. Per tant la principal motivació d'aquest treball és establir una immersió més íntima al món de la IA: Concretament en el paradigma dels sistemes multiagent i els sistemes experts.

D'una banda, es pot considerar que els sistemes multiagent representen força bé la definició de la IA: Una sèrie d'entitats autònomes amb comportaments específics que poden col·laborar per aconseguir objectius i metes concretes. D'altra banda els sistemes experts difusos mitjançant els fonaments de la lògica difusa, pretén lidiar amb l'ambigüitat [1] i establir graus de pertinença.

Fruit de les moltíssimes possibilitats que ofereixen les dues disciplines anteriors, aquest treball planteja la integració d'un SMA, sistema multiagent, amb un sistema d'inferència basat en conjunts difusos. S'espera aconseguir comportaments intel·ligents a partir de la interacció d'agents i el raonament aproximat d'aquests. Donada la naturalesa genèrica de la proposta, aquest projecte es planteja com una solució integradora altament paramètrica, flexible i adaptable.

Profunditzant lleugerament al punt anterior, bàsicament es proposa crear una infraestructura que permeti parametritzar molts dels aspectes de la inferència que ha de realitzar un agent: Base de regles, Variables d'entrada/sortida, etc. A més a més es pretén que la interacció dels agents, en termes d'inferència, comportament i missatgeria siguin variables. Per això serà necessari el disseny d'una Base de dades relacional que modeli i relacioni tots aquests paràmetres.

1.1.2 Problemàtica a resoldre

Tot i que existeixen nombrosos treballs que tracten els camps dels sistemes multiagent i sistemes experts, és cert que molts d'aquests s'apliquen en dominis d'aplicació molt

concrets i en comptades ocasions es troben sistemes que generalitzin o integrin diversos camps de la intel·ligència artificial en un de sol.

La problemàtica que aquest projecte aspira a resoldre és la d'integrar el camp dels sistemes multiagent i el dels sistemes experts, concretament els basats en conjunts difusos, per aconseguir comportaments intel·ligents: De manera altament configurable i flexible, establir escenaris on diversos tipus d'agents utilitzen el raonament aproximat, col·laboren, negocien, es comuniquen i publiquen resultats.

Utilitzant un contracte verbal determinat, ontologia, i aprofitant la integració del raonament aproximat, cada agent pot inferir o consultar a un altre, comparar raonament de diversos agents, etc. Així doncs es permet establir un ecosistema de col·laboració on tots els agents treballin per complir un objectiu comú.

Simplicitat i temps també són aspectes a resoldre: Es pretén simplificar i accelerar la posada en marxa d'escenaris complexos i dissenyar un model de dades intuïtiu que permeti l'explotació de la informació d'una manera senzilla i integrable amb altres sistemes.

Les problemàtiques anteriors es fonamenten en una sèrie de necessitats generalment imprescindibles en qualsevol sistema o plataforma:

- Flexibilitat: Habilitat per poder plantejar situacions on interactuïn agents que permetin resoldre problemes en diferents dominis d'aplicació.
- SimPLICITAT i temps: Permetre la preparació de situacions complexes en el menor temps possible, obtenir resultats i fer-los disponibles per a ser explotats.

Finalment per materialitzar la proposta, es planteja utilitzar les següents eines i tecnologies:

- Oracle Database 11g Express Edition : Sistema gestor de Bases de dades on materialitzar tant els aspectes paramètrics de la integració com els resultats produïts pels agents.
- jFuzzyLogic o JFML: Llibreries Java per implementar sistemes difusos.
- JADE: Entorn de desenvolupament de sistemes multiagent basat en Java.
- GIT: Sistema de control per versionar l'evolució del codi font del projecte.
- Eclipse: Entorn de desenvolupament integrat.
- Java JDK 1.8: JVM i Kit de desenvolupament per executar i desenvolupar amb Java.

- Altres tecnologies que puguin esdevenir necessàries durant el transcurs del projecte: FreeMarker, Jersey REST Api, etc.

1.2 Objectius del Treball: Pre-planificació

A continuació s'exposen els objectius que persegueix la proposta, abans d'haver realitzat la planificació:

1. Aprofundir en l'estudi de les eines i tecnologies indicades, principalment JADE i jFuzzyLogic/JFML: Concretament a partir de PACs, desenvolupar alguns escenaris per separat (Sense integrar SMA amb Sistema difós) i avaluar els resultats, principalment:
 - a. Inicialment es proposa plantejar un entorn multi-agent on un agent demani un raonament concret amb JFML a un altre agent, i a partir d'aquest realitzar un altre raonament.
2. Estudiar i desenvolupar proves de concepte per garantir la viabilitat de la proposta, en termes de:
 - a. Proves de concepte JFML/jFuzzyLogic
 - i. A part de l'estudi del punt anterior, avaluar fins a quin punt es pot parametritzar un FCL/FML i plantejar un model de dades determinat.
 - ii. Plantejar un mètode de càrrega per construir un FML determinat i utilitzar-lo.
 1. Els fitxers FML/FCL representen el sistema d'inferència
 - iii. Integració amb base de dades
 - b. Proves de concepte JADE:
 - i. Comunicació, comportament i jerarquia d'agents.
 - ii. Que es pot i que no es pot modelitzar amb l'objectiu de generalitzar-los al màxim i plantejar un model de dades adient: Comportaments, configuracions, interaccions...
 - iii. Integració amb jFuzzyLogic/JFML, Protégé, Base de dades.
 - iv. Representació dels agents mitjançant una interfície web.
 - v. Generalitzar els resultats de l'acció d'un agent, solitària o conjunta i plantejar un model de dades general.
 - vi. D'acord amb el punt anterior, plantejar possibles mètodes de càrrega per iniciar el *setup* d'un tipus d'agent determinat.

- c. Freemarker: Creació d'un formulari per mantenir les tuples dels models de dades plantejats en les proves de concepte anterior.
3. Un cop superada la fase de proves de concepte i amb una millor perspectiva, definir un model on poder establir tots els aspectes paramètrics/configurable de la integració
 - a. A més cal definir com i segons que produeix resultats:
 - i. Resultat obtingut fruit d'una inferència
 - ii. Resultat fruit d'una interacció amb altres agents
 - iii. Altres escenaris.
 - b. Aquest model de dades serà inicialment modelat en forma de Esquema XML.
4. Amb el model definit, repassar, analitzar i implementar els processos de càrrega de paràmetres i configuració per jFuzzyLogic/JFML i JADE.
 - a. A més revisar i desenvolupar els processos d'escriptura de resultats fruit de la interacció entre agents:
 - b. Formularis amb FreeMarker per poder afegir o editar tuples mitjançant una interfície web.
5. Preparar configuracions i tipus d'agents a la base de dades per establir diversos comportaments intel·ligents:
 - a. En primer lloc, establir un agent amb un sol bloc de regles.
 - b. Seguidament diversos agents amb diversos blocs de regles i plantejar l'escenari multiagent.
6. Iniciar a JADE els escenaris anteriors i iniciar les interaccions.
7. Emmagatzemar els resultats generats pels agents a la base de dades.
8. Explotació de resultats, comprovació i validació mitjançant consultes SQL.
9. Representació de la integració amb una interfície Web i permetre les següents interaccions:
 - a. Afegir nous agents i carregar configuració específica
 - b. Eliminar agents d'una interacció
 - c. Interacció amb agents: Modificació en temps real
10. Facilitar l'explotació de les dades mitjançant serveis REST

Finalment, s'exposen alguns riscos identificats durant la redacció de la proposta:

- Configuració/parametrització dels element esmentats pugui esdevenir limitada per les pròpies restriccions de disseny d'aquestes eines.
- Interacció de JADE amb l'exterior: Es desconeixen els medis d'integració que JADE ofereix, per exemple, per exposar els agents a l'exterior i que aquests puguin ser representats o interactuats per terceres aplicacions.

- Possiblement hi ha riscos que encara no s'han contemplat.

1.3 Objectius del Treball: Post-planificació

A partir dels objectius anteriors i del vistiplau del professor, s'ha procedit a planificar el treball considerant els següents aspectes:

- Descomposició dels objectius en tasques
- Estimació de la duració de cadascuna a partir d'un criteri de complexitat objectiu: Tasques on intervenen elements amb menys coneixement, s'hi assigna més temps.
- Dedicació dels recursos
 - En aquest cas, l'únic recurs disponible sóc jo mateix, amb una dedicació diària variable.
- Hores disponibles del projecte

Durant el desenvolupament d'aquesta planificació s'ha detectat que la realització de tots els objectius proposats al punt 1.2 pot comprometre les restriccions temporals del projecte i dedicació, per tant, s'han planificat tal que, en cas que hi hagi desviacions positives en la durada de les tasques, es poguï continuar desenvolupant els objectius de la proposta.

La llista d'objectius anteriors es distribueix, segons la planificació, de la següent manera:

Objectius principals:

1. Aprofundir en l'estudi de les eines i tecnologies indicades, principalment JADE i jFuzzyLogic/JFML: Concretament a partir de PACs, desenvolupar alguns escenaris per separat (Sense integrar SMA amb Sistema difús) i avaluar els resultats, principalment:
 - a. Inicialment es proposa plantejar un entorn multiagent on un agent demani un raonament concret amb jFuzzyLogic/JFML a un altre agent, i a partir d'aquest realitzar un altre raonament.
2. Estudiar i desenvolupar proves de concepte per garantir la viabilitat de la proposta, en termes de:
 - a. Proves de concepte jfuzzyLogic/JFML
 - i. A part de l'estudi del punt anterior, avaluar fins a quin punt es pot parametritzar un FCL/FML i plantejar un model de dades determinat.
 - ii. Plantejar un mètode de càrrega d'un fitxer FCL/FML determinat i utilitzar-lo.

1. Els fitxers FCL/FML representen el sistema d'inferència
- b. Proves de concepte JADE:
 - i. Comunicació, comportament i jerarquia d'agents.
 - ii. Que es pot i que no es pot modelitzar amb l'objectiu de generalitzar-los al màxim i plantejar un model de dades adient: Comportaments, configuracions, interaccions...
 - iii. Integració amb jFuzzyLogic, JFML, Protégé, Base de dades.
 - iv. Generalitzar els resultats de l'acció d'un agent, solitària o conjunta i plantejar un model de dades general.
 - v. D'acord amb el punt anterior, plantejar possibles mètodes de càrrega per iniciar el *setup* d'un tipus d'agent determinat.
3. Un cop superada la fase de proves de concepte i amb una millor perspectiva, definir un model on poder establir tots els aspectes paramètrics/configurable de la integració
 - a. A més cal definir com i segons que produeix resultats:
 - i. Resultat obtingut fruit d'una inferència
 - ii. Resultat fruit d'una interacció amb altres agents
 - iii. Altres escenaris.
 - b. Aquest model de dades serà inicialment modelat en forma de Esquema XML.
4. Amb el model definit, analitzar i implementar els processos de càrrega de paràmetres i configuració per jFuzzyLogic/JFML i JADE.
 - a. A més revisar i desenvolupar els processos d'escriptura de resultats fruit de la interacció entre agents:
5. Preparar configuracions i tipus d'agents a la base de dades per establir diversos comportaments intel·ligents:
 - a. En primer lloc, establir un agent amb un sol bloc de regles.
 - b. Seguidament diversos agents amb diversos blocs de regles i plantejar l'escenari multiagent.
6. Iniciar a JADE els escenaris anteriors i iniciar les interaccions.
7. Emmagatzemar els resultats generats pels agents a la base de dades.
8. Explotació de resultats, comprovació i validació mitjançant consultes SQL.

Objectius addicionals

1. jFuzzyLogic/JFML - Integració amb base de dades dels sistemes d'inferència
2. Proves de concepte JADE:

- a. Representació dels agents mitjançant una interfície web.
3. Freemarker: Creació d'un formulari per mantenir les tuples dels models de dades plantejats en les proves de concepte anterior.
4. Representació de la integració amb una interfície Web i permetre les següents interaccions:
 - a. Afegir nous agents i carregar configuració específica
 - b. Eliminar agents d'una interacció
 - c. Interacció amb agents: Modificació en temps real
5. Facilitar l'explotació de les dades mitjançant serveis REST

1.4 Enfocament i mètode seguit

L'enfocament que planteja la proposta es fonamenta en la utilització de les tecnologies esmentades anteriorment, on bàsicament tot gira entorn de JADE. JADE és un *middleware* que permet desenvolupar entorns SMA i ofereix flexibilitat per integrar-se amb altres llibreries i eines.

A partir d'aquí ha de ser possible la integració amb la resta de tecnologies a partir del desenvolupament de mecanismes d'integració específics o bé incrustant les llibreries a les classes de JADE natives. Per exemple, per integrar jFuzzyLogic amb JADE, pot ser suficient amb incorporar la llibreria a JADE i que l'agent genèric la invoqui. Contràriament per integrar-se amb la base de dades pot requerir la implementació de classes que facin pont amb JADE.

Convé destacar quina metodologia de treball s'ha utilitzat durant el transcurs del projecte, i quins elements l'han fet possible, principalment amb relació als punts següents:

- Planificació del treball tipus *waterfall*: Agilització
- Repositori de codi font i control de versions
- *Best Practices*

1.4.1 Planificació del treball tipus *waterfall*: Agilització

Com es pot apreciar a la planificació general del projecte, les fases d'anàlisi, disseny i implementació es troben expressades amb un model *waterfall*, on teòricament, el disseny no comença fins que acaba tot la fase d'anàlisi i el desenvolupament no comença fins que acaben les fases d'anàlisi i disseny.

Tradicionalment aquest és l'enfocament de planificació habitual, tot i que a la pràctica és un model amb riscos importants, ja que:

- Al moment de dissenyar els recursos es poden adonar d'aspectes erronis a la fase d'anàlisi, però ja no hi ha *budget* d'hores per la fase d'anàlisi.
- Al mateix al moment d'implementar es poden identificar errors de disseny greus que necessitin anàlisi i disseny, ara bé, en aquest moment del projecte ja no hi ha hores ni temps.

Considerant tots aquests riscos durant l'execució del projecte s'ha optat per aplicar, sota el marc de la planificació donada, un enfocament "agilitzador". Concretament s'ha treballat les tasques d'anàlisi, disseny i implementació iterativament.

Amb la idea anterior, l'execució del projecte s'ha realitzat a través de múltiples cicles Anàlisi – Disseny – Implementació, amb els avantatges que es llisten a continuació:

- S'obté abans un *outcome* del projecte.
- Hi ha marge de reacció si l'*outcome* actual presenta algun problema greu no contemplat.
 - Ja que encara que queda *budget* en hores per tasques d'anàlisi i disseny.
- Respecte a les dates, s'ha de tenir present que la data final és la de fi de la fase de desenvolupament definida a la planificació.

En definitiva, s'ha considerat interessant comentar aquest aspecte de l'execució del projecte perquè es pot apreciar, com fins i tot una planificació *waterfall* pot ser enfocada d'una manera més àgil i resilient.

1.4.2 Repositori de codi font i control de versions

Tant l'execució del projecte com el desenvolupament d'aquest ha estat un procés iteratiu i incremental, on a mesura que es progressa en les fases d'anàlisi i disseny, s'implementen funcionalitats o elements que materialitzen una funcionalitat determinada.

D'aquesta manera, durant el transcurs del projecte i com s'havia anunciat prèviament, s'ha utilitzat un repositori de codi font, tant per temes de seguretat com per poder apreciar l'evolució del projecte a partir de les noves versions.

S'ha optat per utilitzar un repositori GIT i cada *push* al repositori s'ha acompanyat amb un comentari que descriu els *highlights* principals de la versió. D'aquesta manera hi ha cert grau de traçabilitat de l'evolució del projecte.

1.4.3 Desenvolupament del software: *Best Practices*

El treball ha perseguit des del primer moment desenvolupar un producte de qualitat on s'apliquin les millors praxis possibles, respecte al disseny i desenvolupament del software corresponent.

A totes les àrees on el recurs ha disposat de coneixement professional per realitzar una part del software o component, s'ha realitzat de manera correcta i intuïtiva, mesura interessant si aquest projecte és continuat per una a altre recurs o es decideix continuar amb fases posteriors del seu desenvolupament.

1.5 Planificació del Treball

Tal com s'ha esmentat anteriorment, la planificació real del projecte ha forçat a distribuir els objectius amb el propòsit d'assegurar-ne els principals, i la resta, en funció del transcurs del projecte i la completesa de les tasques.

La inexperiència en la majoria d'eines de la proposta pot produir que la planificació de la proposta no tingui la millor precisió i, per tant, és possible que pugui patir modificacions durant el transcurs del projecte.

La planificació del treball ha tingut en compte els següents aspectes crítics:

- a. Dedicació de l'estudiant i estimació de les tasques en hores
 - i. El pla de treball té un calendari adaptat a la disponibilitat i **dedicació diària real en hores**.
- b. Descomposició dels objectius en tasques específiques per poder controlar l'evolució al llarg del temps.
- c. Dimensionament i agrupació de les tasques: s'han agrupat les tasques en fases i, segons com evolucioni, es podran acaparar més o menys tasques addicionals.
- d. Complexitat de les tasques: S'ha destinat un temps important d'anàlisi i disseny, ja que es treballa amb eines pràcticament desconegudes.

1.5.1 Calendari

A la planificació s'ha creat un calendari anomenat *RealCalendar* que defineix la dedicació del recurs i es troba assignada a totes les tasques:

- a. **Work weeks:** Establim el temps dedicat al projecte cada dia de la setmana:
 - a. Dilluns: 19:00 a 21:00, 22:00 a 22:30 i 22:45 a 23:45
 - b. Resta de dies: 19:00 a 21:00, 22:00 a 22:30

- b. **Excepcions:** Es defineixen les excepcions a la dedicació exposada al punt anterior:
- a. *MoreTime:* Període del 3/10/2017 Fins al 16/10/2017
 - i. Cada dia de 19:00 a 20:00 i 21:00 a 23:30
 - b. *Holidays:* Període del 21/10/2017 al 23/10/2017
 - i. Cada dia *Nonworking*.
 - c. Montornès (Un event): Període del 28/10/2017 al 28/10/2017
 - i. Cada dia *Nonworking*.

1.5.2 Planificació

Per realitzar la planificació, sempre dins del marc de les 300 hores del projecte, s'ha comptat amb certa flexibilitat al moment de distribuir les hores i les tasques superiors:

- a. **Definició dels continguts del treball: 29 h**
 - a. Disseny de la proposta i acord amb el professor.
- b. **Pla de Treball: 45 h**
 - a. Planificació del treball
- c. **Desenvolupament del treball -Fase I: 80 h**
 - a. Anàlisi corresponent als objectius principals.
 - b. Redacció de part de la memòria I
- d. **Desenvolupament del treball -Fase II: 101 h**
 - a. Disseny i implementació dels objectius principals.
 - b. Redacció de part de la memòria II
- e. **Redacció de la memòria – Fase III: 15 h**
 - a. Fi redacció de la memòria
- f. **Elaboració de la presentació: 15 h**
- g. **Defensa publica: 15h**

Addicionalment s'ha introduït una fase anomenada: **Desenvolupament del Treball – Fase III:** Anàlisi, disseny i implementació dels objectius addicionals.

Les principals fites del projecte són les següents:

Data objectiu	Fita	Fase
14/11/2017	Fase prèvia i Anàlisi	Desenvolupament del treball – Fase I
14/11/2017	Plantejament si la fase III és possible	Desenvolupament del treball – Fase I
13/12/2017	Disseny i implementació	Desenvolupament del treball – Fase II

02/01/2018	Fi de la memòria	Redacció de la memòria – Fase III
22/01/2018	Defensa pública	Defensa pública

1.5.3 Costos estimats del projecte

1.5.3.1 Tasques i recursos

A continuació s'exposen els costos estimats del projecte, a partir de la planificació presentada:

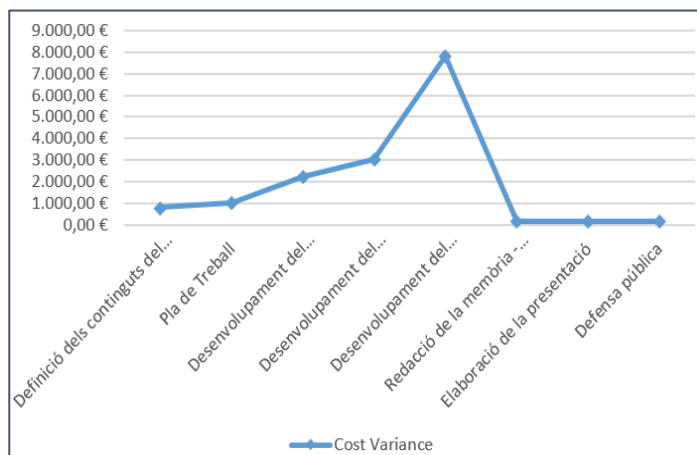


Figura 1: Gràfica tasques generals/cost

Preu de l'hora del recurs: 17 € / h

La proposta de la Fase III finalment no serà realitzada.

El cost del projecte actual és de: **7219,83 €**

El cost estimat de tot el projecte és de: **17370,07 €**

Es presenta la perspectiva general de les tasques:

✓	▶ Definició dels continguts del treball	12,89 days	29	Wed 20/09/17	Mon 02/10/17
✓	▶ Pla de Treball	18,08 days	45	Tue 03/10/17	Sun 15/10/17
✓	▶ Desenvolupament del treball - Fase I	26 days	80	Tue 17/10/17	Tue 14/11/17
✓	▶ Preparació prèvia	6 days	15	Tue 17/10/17	Wed 25/10/17
✓	▶ Anàlisi	20 days	50	Thu 26/10/17	Tue 14/11/17
✓	▶ Redacció de la memòria - Fase I	15 hrs	15	Wed 08/11/17	Mon 13/11/17
✓	▶ Desenvolupament del treball - Fase II	35 days	101	Thu 16/11/17	Mon 18/12/17
✓	▶ Disseny	14,01 days	35,02	Thu 16/11/17	Wed 29/11/17
✓	▶ Implementació	15,4 days	38,5	Wed 29/11/17	Wed 13/12/17
✓	▶ Proves	4,99 days	12,48	Wed 13/12/17	Mon 18/12/17
✓	▶ Redacció de la memòria - Fase II	15 hrs	15	Wed 13/12/17	Mon 18/12/17
✓	▶ Desenvolupament del treball - Fase III	59,8 days	149,5	Mon 18/12/17	Mon 12/02/18
✓	▶ Redacció de la memòria - Fase III	15 hrs	15	Thu 28/12/17	Tue 02/01/18
✓	▶ Elaboració de la presentació	15 hrs	15	Fri 05/01/18	Wed 10/01/18
✓	▶ Defensa pública	15 hrs	15	Wed 17/01/18	Mon 22/01/18

Figura 2: Planificació de les tasques superiors del projecte

Fins ara s'han consumit 270 hores.

Finalment no es realitzarà cap de les tasques planifiques a la Fase III.

Les tasques pendents són:
Objectius addicionals de la Fase III, Elaboració de la presentació i la Defensa pública

1.5.3.2 Maquinari i software

Es deprecien costos de maquinari i accés a la xarxa, ja que s'utilitza l'equip personal del recurs per conduir el projecte. Els costos de les llicències de software dels programes de Microsoft es troben cobertes pel *partnership* de la universitat amb la corporació.

Finalment, la resta de software pertany al domini del software lliure i per tant no genera despeses addicionals.

1.5.4 Riscos potencials i accions de mitigació

A continuació s'exposen els riscos identificats fins al moment que poden afectar negativament al projecte i derivar en desviacions temporals / costos:

Descripció	Àmbit	Acció de mitigació
Compromís /Perduda del codi font desenvolupat.	Programari	Utilitzar un sistema de control de codi, GIT, per versionar el codi font del projecte.
Implementació de codi erroni que compromet el treball desenvolupat fins ara	Programari	Tornar a una versió anterior amb el GIT.
Error de maquinari	Maquinari	El recurs disposa d'altres computadors per continuar la feina, tot i que la posada a punt pot generar desviació temporal en altres tasques.
Sobreestimació d'una tasca	Gestió	El recurs alliberat per la sobreestimació es pot dedicar a la pròxima tasca
Subestimació d'una tasca	Gestió	El recurs ha de continuar amb la tasca subestimada i, la propera tasca es desvia temporalment. El recurs també pot compensar la subestimació amb <i>overtime</i> .
Impossibilitat de complir una tasca per limitacions del software	Requisits	El recurs ha d'acordar amb el professor una solució o <i>workaround</i> .
Recurs indisponible per problemes de salut	Personal	La tasca actual pot patir desviació temporal negativa i com que només hi ha un recurs, aquest haurà de realitzar <i>overtime</i> per a compensar-ho.

1.5.5 Legal

Després d'estudiar la proposta amb deteniment no s'han detectat aspectes que puguin violar cap aspecte legal ni ètic. Respecte aspectes com el copyright i la bibliografia, la font de tot el material utilitzat en aquest treball serà degudament informada a la bibliografia corresponent.

1.6 Breu sumari de productes obtinguts

El producte obtingut després de la realització del projecte és una primera iteració d'un sistema multiagent on, a partir d'un fitxer descriptor amb format XML, posa en marxa un conjunt d'agents que poden interactuar entre ells o amb un actor humà.

Aquests agents són tots iguals, però es caracteritzen en funció del bloc funcional difús que s'indica al fitxer descriptor: Així doncs, les propietats d'entrada, de sortida i regles de raonament poden ser diferents, o iguals, segons es configuri al descriptor.

Les interaccions entre els agents consisteixen en l'intercanvi de valors de sortida i entrada: Principalment els valors de sortida d'un agent, poden servir com a valors d'entrada per altres. Els valors de sortida sempre són fruit del raonament difús de l'agent.

Totes les interaccions són emmagatzemades en un fitxer de *log* corresponent i si s'ha configurat una Base de dades al fitxer descriptor, les dades de la sessió de treball es persisteixen. Finalment el producte presenta una interfície d'usuari per governar tota la funcionalitat del sistema.

1.7 Breu descripció dels altres capítols de la memòria

Els capítols incorporen la següent informació rellevant:

Capítol	Subcapítol	Descripció
2. Materialització del projecte	Contextualització del treball amb la intel·ligència artificial	Pinzellada als aspectes fonamentals relacionats amb la IA.
	Possibilitats del projecte	Expectatives del projecte.
	Entorn de treball pel disseny i implementació del SMA: JADE	Visió general del <i>framework</i> JADE.
	Implementació teoria conjunts difusos i lògica difusa	Visió de diverses implementacions de lògica difusa.
	Desenvolupament del sistema Multi agent	Explicació detallada del treball realitzat de la proposta.
	Obstacles principals al llarg de la vida del projecte	Entrebancs importants durant el transcurs del projecte.
3. Manual exemplificat	Manual aplicat del sistema multi agent. També presenta pinzellades del comportament general del sistema.	

4. Millores	Millores identificades per realitzar a posteriori.
5. Conclusions	Conclusions generals fruit de l'execució del projecte.
6. Bibliografia	Recursos bibliogràfics utilitzats en aquest treball.
7. Annexos	Col·lecció d'annexos que complementen el text principal.

2 Materialització del projecte

Aquesta secció dedica les seves línies a agrupar, per temàtiques, tots aquells aspectes que tenen un paper fonamental tant al projecte com al producte final obtingut.

2.1 Contextualització respecte la Intel·ligència artificial

Es considera oportú contextualitzar el projecte dins de l'univers de la intel·ligència artificial, ja que s'utilitzen dues rames d'aquesta: Agents intel·ligents i raonament aproximat.

2.1.1 Introducció als agents intel·ligents

De manera molt breu, un agent intel·ligent es pot definir com una entitat que compta amb les propietats generals següents: autonomia, reactivitat, habilitat per comunicar-se i a més és capaç de complir amb un objectiu molt específic i concret.

Generalment es poden distingir dues classes d'agents intel·ligents envers la seva complexitat: **Agents reactius i agents deliberatius**. (Moreno i Ribas)

- **Agents reactius:** L'agent més simple, on bàsicament, el comportament intel·ligent pot esdevenir a partir de les interaccions simples que es produeixen amb altres agents reactius.
- **Agents deliberatius:** Agents més complexos que poden comptar amb capacitat de raonament, així doncs, el seu fer individual ja compta amb cert grau d'intel·ligència. A més també poden reaccionar a altres agents o actors humans.

Tanmateix, existeix un altra classificació d'agents, tal que: (Moreno i Ribas)

- **Agents d'interfície:** Agents que col·laboren amb l'usuari per millorar l'eficiència en l'ús de l'aplicació o programa on resideix.
- **Agents d'informació:** Agents que ajuden a gestionar diverses fonts d'informació i ajuden a l'usuari a determinar o escollir quines d'aquestes poden tenir més valor.

- **Agents col·laboratius:** Agents que treballen amb altres agents, mitjançant actes comunicatius. L'acció d'un agent pot venir bé per desenvolupar la feina d'un altre agent.
 - La col·laboració sorgeix partir de la comunicació entre els agents i la feina que realitza cadascun.

2.1.2 Raonament aproximat amb Sistemes difusos

El raonament aproximat és un mètode específic per realitzar raonaments a situacions on la informació presenta incertesa. Els sistemes difusos és un dels enfocaments més habituals per abordar aquesta situació (Wikipedia, 2017).

2.1.2.1 Introducció a la imprecisió i la incertesa

Abans de continuar i per donar una mica de context, resulta convenient posar un exemple d'informació precisa, imprecisa amb conclusions incertes (Torra i Reventós & Godo i Lacasa):

“Escalfar l'aigua a **altes temperatures**”: El concepte alta temperatura, mitja no tenen límits definits, tot i que es podrien representar amb el conjunt següent:

- Temperatura
 - Alta: de 60 a 100 graus
 - Mitja: de 25 a 60 graus
 - Baixa: de 0 a 25 graus

Amb els enfocaments tradicionals, la temperatura a 59 graus es considera mitja i a partir de 60 és alta, Encara que 60 graus potser es troba més a prop de ser temperatura mitjana que no pas alta. Com un sistema pot considerar aquesta subtileza?

2.1.2.2 Els sistemes difusos

Els conceptes d'imprecisió i incertesa introduïts al punt anterior permeten introduir els sistemes difusos: Sistemes basats en la **teoria de conjunts difusos** que permeten obtenir conclusions i modelar el coneixement expert tenint en compte la imprecisió innata que a vegades presenta el llenguatge humà.

El repte, per tant, consisteix bàsicament a:

- Com representar la informació imprecisa: A partir de la teoria de **conjunts difusos**, es defineixen **les variables lingüístiques** del sistema difús.

- Com s'infereix nova informació a partir de dades imprecises: A partir de la teoria de **conjunts difusos**, es defineixen **regles difuses** que representen fets coneguts.

A continuació es dediquen algunes línies a introduir els fonaments de la teoria de **conjunts difusos** i com es relacionen tots plegats.

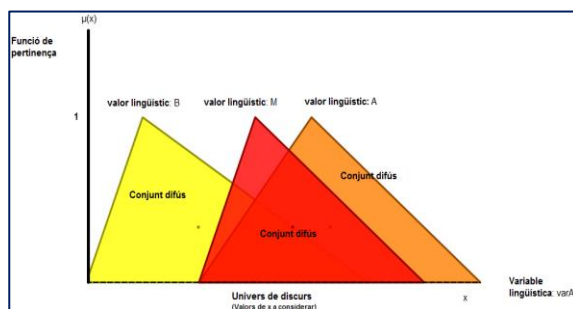
2.1.2.2.1 Variables lingüístiques

Els conjunts difusos permeten modelitzar conceptes o variables lingüístiques on no és possible establir de manera precisa els límits entre els seus possibles valors:

A partir d'aquí sorgeixen una sèrie de conceptes que cal esmentar (Millán Valldeperas):

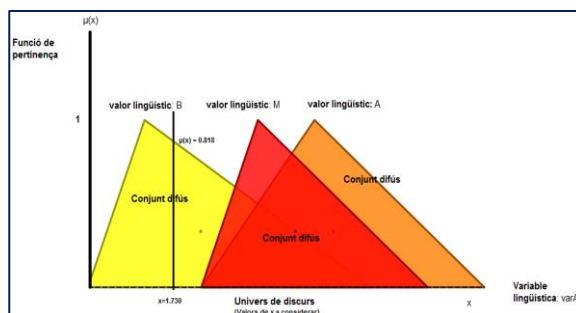
- **Variable lingüística:** Concepte que volem descriure de forma difusa: Per exemple, *varA*.
- **Univers de discurs:** Rang de valors que pot prendre la propietat definida per la variable lingüística: Per exemple, el rang de valors $[0,7]$
- **Valor lingüístic:** Classificacions realitzades sobre la variable lingüística i divideixen l'univers de discurs: Per exemple *B, M, A*.

A partir dels conceptes anteriors, es poden formar els diversos **conjunts difusos** associats a la **variable lingüística**: Cadascun dels valors lingüístics s'associen a un conjunt que té una **funció de pertinença determinada**, sobre l'**univers de discurs** establert.



Gràficament s'il·lustra l'explicació anterior:

La variable lingüística **varA**, pot prendre els valors lingüístics en funció del seu valor numèric *x* i el grau de pertinença ve donat per la **funció de pertinença**.



Per tant, quan **varA** té un valor $x=1.730$, la pertinença a **B és de 0.818** i 0 per la resta de valors lingüístics.

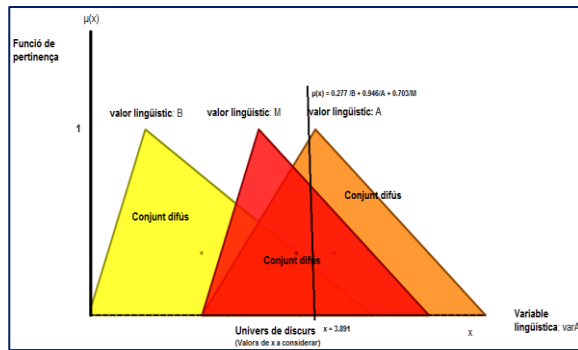


Figura 3: Variables lingüístiques amb els valors lingüístics i els conjunts que els representen

Ara bé, pel valor de $x=3.891$, la pertinença als valors lingüístics es distribueix entre els termes lingüístics tal que:

$$B = 0.277$$

$$M = 0.706$$

$$A = 0.946$$

El procés d'obtenir a partir d'un valor nítid d'entrada, els valors de pertinença corresponents s'anomena **fuzzyficació**.

2.1.2.2 Funcions de pertinença

Tota funció pot ser utilitzada per establir conjunts difusos associats als valors lingüístics, tot i que n'hi ha algunes que resulten més idònies que altres: Tant per la seva simplicitat en termes de computació com aquelles **que permeten modelar una transició de pertinença entre diversos valors lingüístics de manera progressiva** i no pas sobtada: (Millán Valldeperas)

Típicament, s'utilitzen les següents funcions de pertinença:

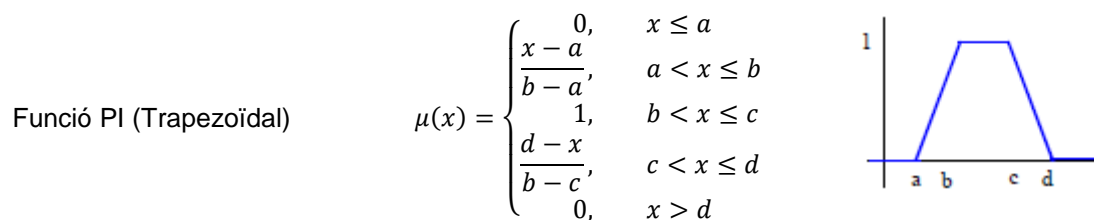
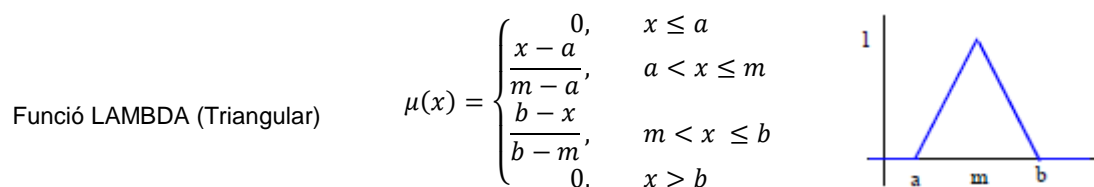
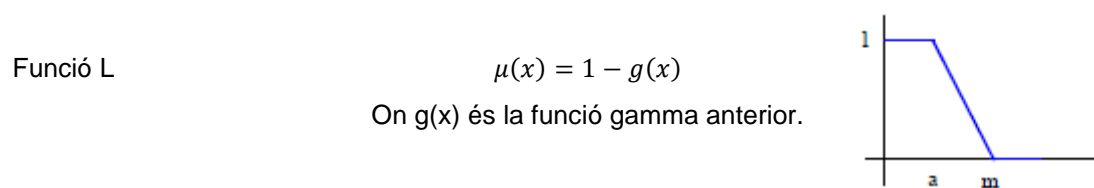
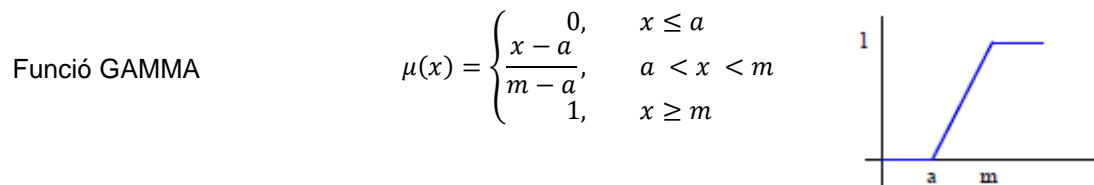


Figura 4: Funcions de pertinença més habituals

Tot i que les funcions de pertinença contemplades són les següents:

Funcions de pertinença	
Continues	Triangular
	Trapezoïdal
	Gauss
	Campana normalitzada
	Sigmoïdal
	Definida a trossos lineal
Discretas	Singletó

2.1.2.2.3 Operacions amb conjunts difusos

Anàlogament als conjunts tradicionals, els conjunts difusos també disposen de les clàssiques operacions d'unió, intersecció, complementari, entre altres (González Morcillo):

- **Unió:** Coneguda com a **T-conorma**, es tracta d'una funció que ha de complir una sèrie de propietats. Sense entrar en detalls, les més habituals són:
 - **Màxim:** $\max(a, b)$
 - **Producte:** $(a + b) - (a \times b)$
 - **Suma limitada:** $\min(a + b, 1)$
- **Intersecció:** Coneguda com a **T-norma**, es tracta d'una funció que compleix una sèrie de propietats. Sense entrar en detalls, les més habituals són:
 - **Mínim:** $\min(a, b)$
 - **Producte algebraic:** ab
 - **Diferència limitada:** $\max(0, a + b - 1)$
- **Complement:** Operació definida per una funció que ha de complir una sèrie de propietats. Igual que amb les anteriors, existeixen una sèrie de funcions a l'abast, com el complement Sugeno.

És interessant llistar breument les operacions amb conjunts difusos, perquè aquestes mateixes seran utilitzades al moment d'avaluar les diverses regles per seleccionar regles i antecedents que compondran el conjunt difús de les diverses variables lingüístiques de sortida.

Per acabar amb les operacions, es molt important remarcar que l'ús de les funcions de t-conorma i t-norma ha de ser **aparellat de la següent manera:** (INTERNATIONAL ELECTROTECHNICAL COMMISSION (IEC), 1997)

T-conorma	T-norma
Màxim	Mínim
Producte	Producte algebraic
Suma limitada	Diferència limitada

El propòsit d'aquest aparellament establert és la de complir les lleis de Morgan, fonamentals en les operacions amb conjunts.

2.1.2.2.4 Raonament difús i regles

A partir de l'establiment de proposicions difuses, simples o complexes, es podrà realitzar el raonament difús, és a dir, realitzar inferències amb el conjunt de variables lingüístiques i el valor numèric x que prengui cadascuna en un instant determinat.

Les proposicions difuses que s'esmenten al paràgraf anterior, són bàsicament **les regles basades en fets reals i coneguts** (Millán Valdeperas):

- **Proposicions difuses simples:** Són les regles o construccions que assignen un valor lingüístic a una variable lingüística: Per exemple varA és B
- **Proposicions difuses compostes:** Totes les regles que combinen una sèrie de proposicions difuses simples: Per exemple varA és B i varB és B

S'intueix a partir de la proposició difusa composta anterior, que existeix una sèrie d'**operadors lògics difusos:** Són aquests operadors els que utilitzaran les funcions amb conjunt difusos definides a [2.1.3.2.3 Operacions amb conjunts difusos](#)

Llavors els operadors lògics difusos són els següents:

- **NOT** (Complement): Aquest operador defineix una funció de pertinença, a partir de la funció escollida pel complement, aplicada a les funcions de pertinença corresponents als valors lingüístic que pren la variable lingüística corresponent.
- **AND** (Intersecció): Aquest operador defineix una funció de pertinença, a partir de la funció escollida per la Intersecció, aplicada a les funcions de pertinença corresponents als valors lingüístics que prenguin les dues variables lingüístiques.
- **OR** (Unió): Aquest operador defineix una funció de pertinença, a partir de la funció escollida per la Unió, aplicada a les funcions de pertinença corresponents als valors lingüístics que prenguin les dues variables lingüístiques

Finalment és molt important especificar **la implicació**, és a dir: varA és B i varB és B llavors out és B. Hi ha moltes maneres d'interpretar o especificar la implicació, però la utilitzada en aquest treball és la **implicació MANDAMI**.

2.1.2.2.5 Implicació MANDAMI

Sense entrar en detalls, la implicació MANDAMI considera que una condició es certa quan l'antecedent i el conseqüent ho són (Millán Valldeperas):

$$p \rightarrow q \equiv p \wedge q$$

2.1.2.2.6 Inferència Difusa

Amb els elements introduïts es poden representar fets i regles difuses i, amb aquests, es poden realitzar inferències. Aquest treball utilitza fonamentalment la inferència amb la implicació de mandami anterior i els operadors lògics segons com s'especifiqui al fitxer FCL corresponent.

En conseqüència, el procés d'inferència es basa en els passos següents (González Morcillo):

Pas	Breu descripció
Procés de Fuzzyficació	A partir dels valors nítids assignats a cada variable lingüística, obtenir els valors de pertinença als diversos valors lingüístics definits.
Procés d'avaluació de regles	A partir dels valors de pertinença anterior, les funcions típiques (t-norma) establertes pels operadors (AND, OR), el criteri d'acumulació (t-conorma) i els blocs de regles: Es realitza l'avaluació per seleccionar les regles més representatives i posteriorment poder obtenir el conjunt difús de les variables de sortida
Procés d'agregació	A partir de les regles seleccionades, es combinen les funcions de pertinença d'aquestes per obtenir un conjunt difús determinat per cada variable de sortida.
Procés de Nitidificació	Amb els conjunts difusos obtinguts per les variables de sortida és moment d'obtenir un valor nítid, ja que normalment convé donar un valor concret i no pas el conjunt difús obtingut, a partir del mètode escollit. Habitualment s'utilitza el centre de masses.

2.2 Possibilitats del projecte

Abans d'aprofundir amb la lectura del material, pot resultar interessant exposar algunes de les possibilitats que ofereix aquest treball:

- Varietat i flexibilitat d'escenaris multi agent

- L'usuari pot tenir diverses versions del fitxer descriptor on especificar, per cada cas, quins agents genèrics i inferències utilitzar.
- Interfície gràfica general
 - La interfície permet localitzar fàcilment tots els agents del sistema multi agent amb un camp de cerca.
 - Un cop localitzat, es possible accedir-hi per treballar amb l'agent.
- Interacció amb els agents genèrics
 - Cada agent disposa d'una interfície que permet la interacció directa per part de l'usuari
 - Informar les seves variables d'entrada
 - Inferència
 - Traçabilitat de l'activitat de l'agent
 - Accés a eines de monitoratge i control JADE: Agent RMA.
- Resiliència als canvis i adaptabilitat
 - Un dels problemes més habituals és els canvis que succeeixen a la realitat.
 - Els resultats fruits d'una inferència poden tenir sentit avui, però no tenir-ne al futur (Moreno i Ribas)
 - Amb diferents versions del descriptor es pot mantenir un sistema multi agent on s'infereixi informació d'avui i un altre, on s'infereixi informació amb un altre criteri.
- Dins del context actual, d'adaptabilitat i flexibilitat, també cal destacar la capacitat de l'aplicació per permetre a l'usuari fer proves, tal que:
 - Al descriptor es poden especificar tants agents com es desitgi, per tant és possible definir agents de prova que només interactuïn amb l'usuari utilitzant uns determinats paràmetres i comparar els resultats amb els d'un altre agent que infereixi amb un altre criteri.
 - És més, resulta fàcilment comprovable com variacions als conjunts difusos associats als valors lingüístics d'una variable influeixen als resultats finals d'una inferència.
- La informació generada a totes les sessions de treball es pot marcar com simulada
 - Un fitxer descriptor amb agents i inferències de prova es pot establir com generador de dades de simulació. Així doncs les dades es persisteixen a la base de dades com simulades i es poden distingir d'altres dades obtingudes al passat.
 - Llavors es poden comparar els resultats simulats i no simulats.

- Interacció amb altres sistemes multi agent
 - Aquesta aplicació pot treballar amb altres sistemes multi agent, sempre que els agents externs utilitzin la missatgeria ACL de la mateixa manera que els agents genèrics d'aquest treball.
 - Algunes interaccions poden ser no persistents a la base de dades.
- Altres possibilitats no contemplades
 - Els punts anteriors són només algunes possibilitats inicials, tot i que altres possibilitats d'ús, un cop conegudes les funcionalitats, poden venir donades per l'usuari en qüestió.

2.3 Entorn de treball pel disseny i implementació del SMA: JADE

2.3.1 Introducció

Amb el propòsit de materialitzar el treball com un sistema multiagent, es va decidir utilitzar JADE com a element central. Moltes de les decisions de disseny preses al transcurs del projecte han estat supeditades a les particularitats d'aquesta eina.

Les pròximes línies pretenen donar una pinzellada al framework JADE i a contextualitzar molts dels aspectes del treball dins d'aquest framework, entre altres:

- Perspectiva JADE
- Agents JADE: Agents nadius i agents estesos.
- Comportaments JADE
- Particularitat dels agents JADE: Programador de les tasques
- Comunicació dels agents
- Ontologia

2.3.2 Perquè JADE?

A l'inici del treball el més semblant a una eina o utilitat coneguda que permetés dissenyar o implementar un sistema multiagent era SeSAM (SeSAm Multi agent simulation, sense data), un entorn de treball que va ser introduït a l'assignatura d'aprenentatge computacional i va culminar en la realització d'una activitat on se simulava una colònia de formigues.

Un cop va finalitzar l'assignatura, no es va tractar més l'assumpte dels sistemes multiagent agent fins al moment d'escollir una proposta de treball final de grau mitjanament acordada amb el consultor. En aquest punt és on JADE va començar a

adquirir protagonisme, ja que algunes de les propostes suggerien utilitzar aquest marc de treball.

Sense gaires més vacil·lacions i en acceptar la proposta, l'estudi de JADE va començar. El fet indiscutible de ser un entorn implementat amb llenguatge Java ha facilitat l'adquisició del coneixement i el desenvolupament del prototipus de forma excepcional. Tot i que al final, el més rellevant és conèixer les peculiaritats pròpies de JADE, més enllà del seu llenguatge d'implementació.

2.3.3 Java Agent Development Framework: Perspectiva general

JADE es considera un entorn de treball basat en la programació orientada a l'agent – *Agent-Oriented Programming*– i aquest paradigma fa que en moltes ocasions l'estratègia per abordar alguns problemes sigui diferent de l'habitual. (Bellifemine, Caire, & Greenwood, 2007)

Dit això JADE és un *middleware*, és a dir, al seu si poden conuiu diversos agents, els quals realitzen determinades tasques i tenen la capacitat per comunicar-se amb altres agents i amb sistemes exteriors; Tot i que aquest treball no s'explota completament la comunicació amb sistemes exteriors.

A continuació s'exposa el clàssic diagrama d'arquitectura JADE i es facilita la contextualització oportuna (Bellifemine, Caire, & Greenwood, 2007):

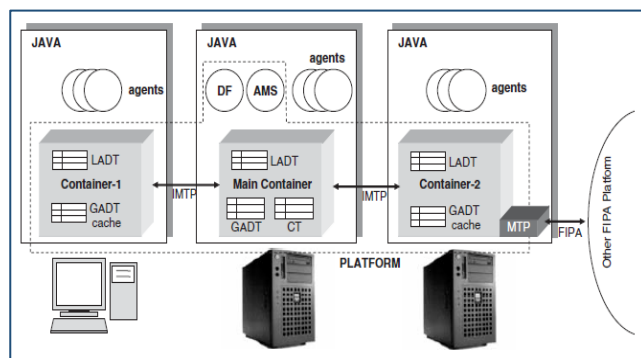


Figura 5: Arquitectura típica d'un Sistema multiagent JADE

El diagrama mostra tres entorns d'execució JADE, distribuïts a 3 màquines. Aquests tres elements componen un sistema multiagent distribuït.

Es pot considerar que l'element central d'aquest SMA és l'entorn d'execució amb el contenidor Principal, *Main Container*, ja que aquest disposa de tots elements fonamentals de qualsevol SMA implementat amb aquest framework.

Molt breument es descriuen els elements més rellevants de l'entorn d'execució principal:

Element	Descripció
Contenidor	Espai dins de l'entorn d'execució JADE que agrupa un conjunt d'agents.

Agents	JADE compta amb una sèrie d'agent nadius que s'encarreguen de tasques fonamentals del SMA: El registre i descobriment d'altres agents, comunicació entre agents, etc.	
	Agent DF	Agent AMS
	Agent que registra la resta d'agents a partir de les seves característiques (<i>ServiceDescription</i>). Actua com agent cercador d'altres agents, a partir de les característiques establertes.	Orquestra la plataforma i també el registre d'agents: Gestiona el cicle de vida de cadascun, proveeix d'identificadors AID als agents, entre altres.
GADT	Taula on s'emmagatzemen les característiques registrades, de cada agent, a través de l'agent DF: Estat de l'agent, localització, etc.	
LADT	Taula que descriu els agents locals de cada plataforma, utilitzada al moment de cercar un agent, abans de contactar amb l'agent DF	
CT	Taula amb el registre de contenidors que conté les adreces i possibilita la comunicació entre contenidors.	
MTP	Protocol de transport utilitzat durant les comunicacions entre entorns d'execució. Tot i que dins d'una plataforma, també es pot utilitzar.	
FIPA	Organisme que vetlla per l'estandardització tecnològica dels sistemes intel·ligents basats en agents	
Elements opcionals rellevants disponibles al treball		
RMA	Agent JADE natiu que compta amb una interfície d'usuari i permet el govern de l'entorn d'execució.	
Sniffer	Agent JADE natiu que permet interceptar les comunicacions entre agents i observar totes les propietats de la missatgeria generada al si d'una interacció.	

És important destacar que aquesta taula no conté tots els elements d'un entorn d'execució i es llisten els fonamentals que aporten context al treball realitzat.

2.3.4 Agents JADE

Tots els agents JADE estenen d'una classe específica anomenada *jade.core.Agent* i a partir d'aquesta, la resta dels agents del sistema multi agent, siguin nadius o no, presenten la seva morfologia específica.

Els agents JADE en si consisteixen en un fil, *thread*, Java

Dins d'aquesta varietat morfològica, cal dir però, que tots els agents tenen un conjunt de fases idèntiques i comuns: Es parla de les fases d'inicialització i finalització dels agents.

2.3.4.1 Fase d'inicialització

Com el nom de la fase deixa entreveure, la fase de setup té un propòsit molt específic (Isern Alarcón):

- Registre de l'agent inicialitzant a l'agent DF.
 - A partir de les propietats específiques de l'agent, *ServiceDescription*
 - Implícitament es registra també a l'agent AMS, i aquest darrer, li assigna un identificador (AID)
- Assignació d'un comportament mínim, per defecte.

Aquests dos punts són **obligatoris** per tots els agents, ja que: D'una banda, fem visible a l'agent per si un agent posterior l'ha de localitzar i de l'altre, s'assigna un comportament (Tasca a realitzar) perquè l'agent pugui realitzar alguna acció inicial.

Per complir amb els objectius anteriors, JADE proveeix el mètode *setup* comú a tots els agents. Aquest mètode *setup* s'ha d'implementar en funció de les necessitats de l'agent.

2.3.4.2 Fase de finalització

És possible que un agent concret, en acabar la seva tasca, segons convingui i de forma controlada, estigui dissenyat per desaparèixer de la plataforma (Isern Alarcón).

En tot cas, es parla del moment de finalitzar l'agent, on bàsicament es realitzen les següents accions:

- Eliminació de l'agent de la plataforma a través de l'agent DF.

JADE proporciona el mètode *takeDown*, per implementar la finalització segons convingui.

2.3.4.3 Cua de recepció de missatges

Els agents, addicionalment també compten amb una cua de recepció de missatges, on s'acumulen els missatges pendents de tractar.

2.3.5 Comportaments dels agents JADE

Tota l'activitat posterior a la inicialització i prèvia a la finalització d'un agent JADE es fonamenta en l'ús dels comportaments, *behaviours*, on principalment (Isern Alarcón):

- Els *behaviours* es poden entendre com petits fragments de funcionalitat, que combinant-los apropiadament, es pot aconseguir un comportament global efectiu.
- Obligatòriament un agent, al moment de ser inicialitzat, ha de comptar amb almenys un comportament. La resta poden ser adquirits durant el transcurs de la seva activitat.

A priori els comportaments s'executen de forma concurrent, tot i que realment s'ha de considerar les peculiaritats esmentades al punt **2.3.6 Particularitats dels agents JADE**

Tots els comportaments tenen en comú els mètodes següents:

Mètodes comuns a tots els agents	
action	Mètode principal del comportament.
reset	El comportament torna a l'estat inicial d'execució
block/restart	Atura temporalment l'execució de l'agent. block sincronitzat amb la cua de recepció de missatges: En rebre un missatge surt del bloqueig.
done	Per comprovar si un comportament ha acabat.

Tot i que JADE incorpora una extensa col·lecció de comportaments per reproduir les situacions típiques, tots els comportaments parteixen de:

SimpleBehaviour – Comportaments Simples	
OneShotBehaviour	Mètode action s'executa un cop.
CyclicBehaviour	Mètode action s'executa cíclicament.
TickerBehaviour	Mètode action s'executa cada cert temps periòdic
WakerBehaviour	Mètode action un cop passa determinat període de temps
MsgReceiver	Mètode action un cop es rep un missatge amb certes característiques
SimpleAchieveREInitiator	Versió simple del patró de comunicació FIPARquest

CompositeBehaviour – Comportaments compostos per comportaments simples subjacents	
ParallelBehaviour	Execució de comportaments subjacents de forma paral·lela
SerialBehaviour	Execució de comportaments amb ordre, no paral·lels
FSMBehaviour	Executa els comportaments subjacents imitant a una màquina d'estats
SequentialBehaviour	Els comportaments subjacents s'executen de manera seqüencial

Com es veurà posteriorment al llarg del treball, els comportaments s'implementen a partir d'estendre un comportament general, tot i que sempre es manté l'essència del comportament primitiu.

2.3.6 Particularitats dels agents JADE

Després d'introduir els comportaments al punt anterior, es considera fonamental començar a descriure algunes de les particularitats de l'entorn JADE:

2.3.6.1 Execució de comportaments i la seva planificació

Tot i que un agent pot executar de manera concurrent diversos comportaments, és fonamental esmentar que la **planificació d'aquests és cooperativa**, (Bellifemine, Caire, & Greenwood, 2007) per tant:

Quan el mètode action d'un comportament s'ha planificat i s'executa, aquest s'executa fins que acaba, en conseqüència **només pot haver un comportament actiu a l'hora**.

Si la planificació de tasques no fos cooperativa però basada en la preempció, un behaviours determinat podria interrompre a un altre en mig de l'execució del seu mètode action.

2.3.7 Comunicació dels agents JADE

La comunicació dels agents es basa en l'intercanvi de missatgeria que compleix amb l'estandardització dictada per la FIPA i gràcies a això, agents que es troben a l'entorn d'execució JADE es poden comunicar entre ells i amb altres sistemes, de forma normalitzada. Es parla dels missatges ACL, *ACL Message*.

Sense entrar en gaires detalls, bàsicament, l'estructura d'un missatge ACL és la següent:

Aspectes fonamentals d'un missatge ACL	
<i>Performative</i>	Intencionalitat bàsica del missatge.
Sender	Emissor.
Receiver	Receptor.
Ontology	Contracte verbal o llenguatge: Permet la normalització del contingut, <i>Content</i> , d'un missatge ACL.
Content	Informació específica del missatge: El contingut s'expressa utilitzant l'ontologia especificada al aspecte anterior.
ConversationId	Identificació d'un intercanvi de missatgeria determinat.

Hi ha altres aspectes que componen un missatge ACL, però els anteriors són els més representatius i utilitzats en aquest projecte.

2.3.7.1 Ontologia

A l'apartat anterior s'ha fet un incís força descriptiu del paper de l'ontologia: Normalització del contingut de missatges intercanviats per dos agents.

Certament, quan dos agents que intercanvien missatges amb la mateixa ontologia, s'estableix un acord verbal per reduir l'espai semàntic fins a un punt on tots dos agents

podran interpretar, amb major o menor grau, el contingut dels missatges i actuar en conseqüència.

Aquest treball utilitza una ontologia simple a l'hora d'intercanviar la informació que un agent computa per un altre. Més endavant es realitzarà l'incís corresponent.

2.4 Implementacions teoria conjunts difusos i lògica difusa

Un cop introduïda molt breument la teoria de conjunts difusos que fonamenta el raonament aproximat realitzat pels agents d'aquest treball, és moment d'exposar una mostra de les implementacions disponibles al mercat i són compatibles amb JADE.

El criteri per seleccionar una implementació o un altre, ha estat el següent:

Criteris	Justificació
Implementació realitzada amb llenguatge JAVA	Per assegurar la interoperabilitat i plena compatibilitat amb JADE i evitar l'ús d'un <i>wrapper</i> .
Codi font de la implementació obert	En cas de problemes o errors, sempre hi ha la possibilitat de poder depurar, <i>debugging</i> , la part de codi corresponent per a intentar entendre el problema.
Implementació basada en software lliure amb llicència GNU	Per evitar augmentar els costos del projecte amb llicències addicionals.

Els punts següents tracten:

- Com representar els sistemes d'inferència pel que fa a els formats de representació.
- Implementacions considerades per aquest treball.

2.4.1 Representació dels sistemes d'inferència

La representació dels sistemes d'inferències és un aspecte molt important, ja que pot influir considerablement en com l'usuari final ha de representar un sistema d'inferència que vulgui utilitzar ja sigui en un agent d'aquest treball o en altres punts.

S'han identificat dues formes de representar sistemes d'inferència disponibles avui dia, ambdues suportades per organismes normalitzadors, cosa que assegura la estandardització i la compatibilitat.

2.4.1.1 FCL (Fuzzy Control Language)

Aquest sistema de representació de sistemes d'inferència es troba sota el beneplàcit de l'IEC 61131-7 i bàsicament especifica l'estructura dels fitxers que implementen els sistemes de raonament sota el marc d'aquesta especificació (INTERNATIONAL ELECTROTECHNICAL COMMISSION (IEC), 1997).

L'estructura dels fitxers no segueix cap format basat en XML ni JSON, però compta amb una sèrie de paraules claus i blocs que permeten informar tots els elements d'un sistema d'inferències: Variables lingüístiques, valors lingüístics, paràmetres d'activació de regles, operacions, mètode d'obtenció dels valors nítids, Regles, etc.

És molt important destacar que l'abast d'aquest estàndard es troba limitat a inferències que utilitzen la **implicació Mandami**, es a dir, amb la **especificació FCL no podem representar** altres esquemes d'inferència. (INTERNATIONAL ELECTROTECHNICAL COMMISSION (IEC), 1997)

Per acabar, el **manual aplicat del capítol 3** hi ha mostres de fitxers amb format FCL.

2.4.1.2 FML (Fuzzy Markup Language)

El sistema de representació FML ve definit per l'estàndard IEEE 1855-2016 del IEEE, *Institute of Electrical and Electronics Engineers*, on bàsicament es defineix un format de representació de sistemes d'inferència basat en XML (Contributors, 2017).

Per tant, els sistemes representats amb FML són fitxers XML amb una estructura de nodes determinada, on es poden informar tots els aspectes que defineixen un sistema d'inferència: Variables, blocs de regles, entre altres.

L'abast d'aquest format de representació va més enllà d'inferències que utilitzen la implicació Mandami, contemplant també altres inferències, com la Tsukamoto. (Soto-Hidalgo, Alonso, & Alcalà-Fdez, sense data)

2.4.1.3 Comparació dels sistemes de representació

Un cop introduïts els dos sistemes de representació es procedeix a realitzar una breu comparació dels sistemes de representació, per mostrar les fortaleeses observades de cadascun:

- **FCL (Fuzzy Control Language)**
 - Tot i que només contempla inferència amb implicació mandami, es tracta d'un sistema de representació anterior a FML i al mercat hi ha moltes

implementacions que la utilitzen: De fet, **les tres implementacions esmentades al treball.**

- El format especificat és prou comprensible i es troba tipificat per l'estàndard corresponent, per tant no hi ha gaire inconvenient a què no segueixi les últimes tendències quant a representació de la informació: JSON, XML, etc.
- **FML (Fuzzy Markup Language)**
 - Aporta valor el fet de poder utilitzar inferències més enllà de la mandami, ja que per dominis d'aplicació molt concrets és interessant disposar de més opcions.
 - El format d'especificació segueix les tendències actuals de representació de la informació, tot i que a escala semàntica, no aporta més del que pot aportar la representació FCL.

2.4.2 Implementacions

Fins ara s'ha introduït les diverses formes de representar textualment els sistemes d'inferència difusos, per tant, és moment d'introduir algunes de les implementacions que es troben al mercat, a partir de la taula següent:

Implementacions estudiades			
	JFML (Java Fuzzy Markup Language)	jFuzzyLogic	jfuzzylite
Característiques principals			
FCL	Si	Si	SI
FML	Si	No	No
Sistema Mandami	Si	Si	Si
Altres sistemes	Si (Tsukamoto, Anya, TSK, etc.)	No	Si (Takagi-Sugeno, Larsen, Tsukamoto, etc.)
Regles programàtica ment en temps d'execució	Si	Si	Si

Altres aspectes			
URL	http://www.uco.es/JFML/	http://fuzzylogic.sourceforge.net/html/index.html	https://www.fuzzylite.com/java/
Llicenciamnt	LGPLv3	LGPLv3	GPLv3

2.4.2.1 JFML (Java Fuzzy Markup Language)

Es tracta de l'única implementació esmentada en aquest treball que utilitza la representació **FML** i per tant, compta amb la capacitat de realitzar inferències de diversos tipus: Mandami, Tsukamoto, etc. (Soto-Hidalgo, Alonso, & Alcalà-Fdez, sense data)

Una característica interessant d'aquesta implementació és la importació de fitxers amb representació FCL i la possibilitat de convertir-los al format FML Funcionalment parlant, l'estudi no ha anat més enllà de la inferència a partir de fitxers FC i exposició de les variables de sortida.

JFML també es caracteritza per poder crear noves variables de manera programàtica i enriquir el sistema d'inferència donat.

Inicialment aquesta implementació era la candidata a ser utilitzada al treball, tot i que finalment no va ser així: Els detalls d'aquest canvi d'implementació es poden trobar a la secció corresponent **2.6 Obstacles principals al llarg de la vida del projecte**.

2.4.2.2 JfuzzyLogic

Aquesta implementació utilitza exclusivament el sistema de representació **FCL** i a priori només contempla la realització d'inferències amb implicació Mandami. (Cingolani & Alcalá-Fdez)

Resulta interessant la superposició d'expressions que aquesta llibreria contempla partint de l'especificació FCL bàsica, ja que simplifiquen la generació del fitxer FCL que representa un sistema difús. Per exemple la definició de les funcions de pertinença dels valors lingüístics es poden expressar o bé com especifica FCL o bé amb una sintaxi pròpia molt intuïtiva.

Finalment, aquesta implementació aporta una sèrie de plugins per l'entorn de treball Eclipse on es representen gràficament els aspectes d'un sistema d'inferència determinat.

Aquest treball utilitza aquesta implementació, pels motius especificats a la secció **2.6 Obstacles principals al llarg de la vida del projecte**.

2.4.2.3 jfuzzylite

La implementació jfuzzylite utilitza tant FCL com un sistema de representació propi. Contempla la possibilitat de realitzar inferències de múltiples tipus: Mandami, Tsukamoto, Takagi- Sugeno, entre altres. (Rada-Vilela, sense data)

Més enllà d'això l'estudi d'aquesta implementació ha estat mínim, ja que el fet de dedicar temps a aquesta implementació té relació amb un dels esdeveniments donats a la secció

2.6 Obstacles principals al llarg de la vida del projecte.

2.5 Desenvolupament del sistema Multi agent

Les properes seccions centren la seva literatura a descriure amb cert nivell de detall les parts fonamentals que formen el sistema multiagent realitzat en aquest treball. A continuació es proporciona una pinzellada a cadascuna de les seccions per ubicar al lector:

Seccions del capítol	Observació
Perspectiva de l'arquitectura del sistema multi agent	S'introdueix l'arquitectura final del sistema multi agent.
Descriptor del Sistema multi agent	Fa referència a tot el relacionat amb la caracterització del sistema multiagent a partir del fitxer Descriptor que ofereix el projecte.
Plataforma multiagent amb JADE	Aquest apartat ofereix detalls quan a com s'ha utilitzat JADE per definir els diversos tipus d'agent, els seus comportaments, les comunicacions entre aquests, l'ontologia entre altres.
Inferència amb JfuzzyLogic	En aquest punt s'explica com s'explota la implementació JfuzzyLogic i com es combina amb els comportaments d'un agent JADE.
Persistència d'interaccions i resultats	Apartat que dedica les seves línies a donar detalls de com es persisteixen tant les interaccions dels agents com els resultats fruit d'aquestes interaccions.
Aspectes satèl·lit i transversals	Es descriuen aspectes del projecte que no pertanyen a un àrea principal del projecte, com pot ser la infraestructura de <i>logging</i> .

Per facilitar la comprensió de tots els aspectes del treball s'han dissenyat figures per facilitar la comprensió i intentar millorar l'experiència del lector.

2.5.1 Perspectiva de l'arquitectura del sistema multi agent

A continuació s'ofereix una figura que mostra l'arquitectura del sistema multi agent desenvolupat en aquest treball:

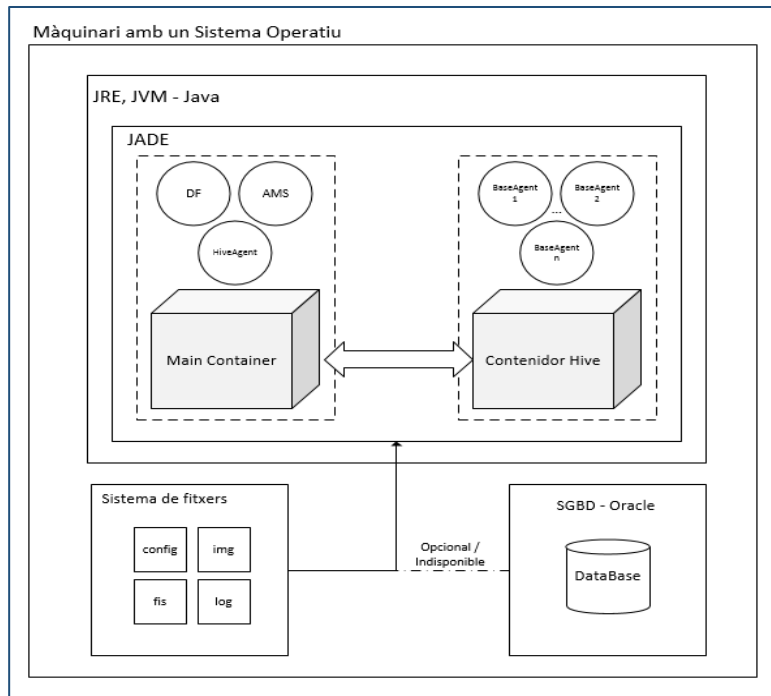


Figura 6: Arquitectura del sistema multiagent desenvolupat

Concretament l'arquitectura presenta els elements següents:

Elements d'arquitectura	Breu detall
Entorn d'execució JADE	Un entorn d'execució JADE amb dos contenidors, Main Container i Hive, un agent propi de tipus HiveAgent i una col·lecció determinada d'agents genèrics <i>BaseAgent</i>
Sistema de fitxers, Directoris	config: Aquí es troba el fitxer Descriptor.xml i el DescriptorSchema.xsd
	img: Directori auxiliar que conté algunes icones
	fis: Directori per emmagatzemar els fitxers FCL que utilitzen els agents del SMA.
	log: Conté el fitxer SMA.log que registra tots els esdeveniments del SMA.
SGBD, Base de dades (Opcional)	Es persisteixen les interaccions del sistema multi agent a la Base de dades. Els paràmetres de connexió s'especifiquen al fitxer Descriptor.xml.

Un cop esmentats per sobre els elements principals de l'arquitectura, es procedeix amb el detall a través de la resta de capítols.

2.5.2 Descriptor del Sistema multi agent

El descriptor del sistema multi agent, mitjançant un fitxer XML, modelitza tota la configuració del sistema multi agent. Llavors a partir dels elements complexos d'aquest,

es representen els agents i les inferències emprades. El fitxer Descripor.xml i DescriptorSchema.xsd es troben al **directori config**.

La decisió d'utilitzar el parell XML/XSD per dissenyar el model de configuració de l'aplicació, respon a la necessitat de proporcionar cert rigor a la part encarregada de configurar el sistema.

Aquesta especificació XML/XSD és exclusiva i pròpia d'aquest treball i presenta algunes característiques especials que cal conèixer. Intuïtivament, es poden modelitzar els següents aspectes del sistema:

Aspecte configurable	Descripció
Mode de treball	Si el sistema genera informació simulada o no.
Agents	Cadascun dels agents del SMA s'han de modelitzar al descriptor.
Inferències	Cadascun dels sistemes d'inferència que poden utilitzar els agents s'han d'informar al descriptor.
Base de dades	Informació de connexió a la base de dades Oracle.

Les pròximes seccions proporcionen informació de tots els aspectes anteriors.

2.5.2.1 Perspectiva general del descriptor

Es mostra una representació visual del model del descriptor:

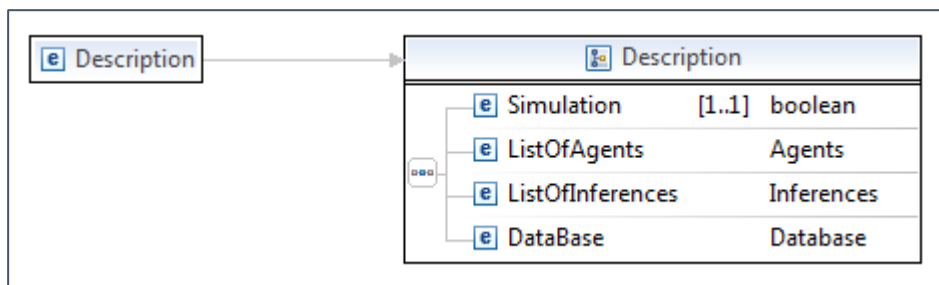


Figura 7: Esquema descriptor del sistema multiagent

Abans de s'informa de l'existència de **l'annex 7.5 Especificació completa model Descriptor** on es pot trobar el detall tècnic de tot el model descriptor.

2.5.2.2 Simulation: Mode de treball

El mode de treball permet diferenciar les dades entre dades de simulació i dades de producció. Per tant, en funció del valor d'aquest camp, les dades que es persisteixen a la base de dades poden ser diferenciades.

Cal advertir que quan les dades no es persisteixen a la base de dades i només són al log, SMA.log, aquesta distinció no es contempla.

2.5.2.3 ListOfAgents: Llista d'agents del sistema multi agent

Com es pot apreciar a la **Figura 7**, els agents del sistema es configuren a partir dels elements ListOfAgents i Agents. Bàsicament consisteix en una col·lecció d'agents on per cadascun, cal establir una sèrie de propietats. Tots aquests agents modelats poden arribar a ser un agent genèric, *BaseAgent*, al SMA.

2.5.2.4 ListOfInferences: Llista de sistemes d'inferència del SMA

Les inferències disponibles que poden utilitzar els agents genèrics han de trobar-se modelades al fitxer descriptor respectant l'especificació del model. Certament, es tracta d'una col·lecció d'inferències, que els agents definits al punt anterior poden utilitzar, si així es configura.

Pot haver inferències informades al descriptor que no siguin utilitzades. És al punt anterior on es defineix per cada agent, les inferències a utilitzar.

2.5.2.5 Database: Base de dades

Hi ha una secció al model que permet configurar la connexió amb la base de dades. Tot i que sembla que hi hagi molta flexibilitat en l'àmbit de la configuració, ara per ara només es contempla l'ús de bases de dades Oracle i per tant, qualsevol intent de configurar un *driver* o URL diferent provoca que s'ignori la persistència de les dades/ interaccions a la base de dades.

2.5.2.6 Validació

Per concloure un dels aspectes més importants del model xml/xsd que descriu el sistema multi agent és l'aspecte de la validació. No superar la validació del descriptor.xml provoca que l'aplicació no comenci a funcionar, tot i que deixa un registre al log SMA.log.

Cal destacar que aquest pas de validació no comprova que els fitxers FCL referenciats es trobin al directori corresponent ni siguin vàlids: Aquesta validació la realitza l'agent genèric.

2.5.3 Plataforma multiagent amb JADE

A continuació s'especifiquen els diversos aspectes dins de l'àmbit de JADE que es troben involucrats en aquest treball. Particularment, en termes de:

Aspecte	Descripció
Agents propis	Agents propis desenvolupats al treball, com són l'agent genèric, <i>BaseAgent</i> i l'agent <i>Hive</i> , <i>HiveAgent</i> .
Comportaments propis	Tots els comportaments desenvolupats per aconseguir que l'agent genèric tingui un funcionament genèric i global.
Comunicació dels agents	Com es comuniquen els agents genèrics entre si i quin acord verbal, Ontologia, utilitzen.
Interfície gràfica dels agents	L'aspecte visual dels agents desenvolupats a la proposta.

Les properes seccions presenten diagrames de mida considerable, per tant es deixa aquest espai en blanc per poder continuar en una pàgina nova.

2.5.3.1 Perspectiva de les interaccions del sistema multiagent

Prèviament al detall, es facilita una perspectiva de les interaccions més habituals que es donen al sistema multi agent. Les interaccions de la figura, en cap cas, bloquejarien interaccions d'altres agents genèrics.

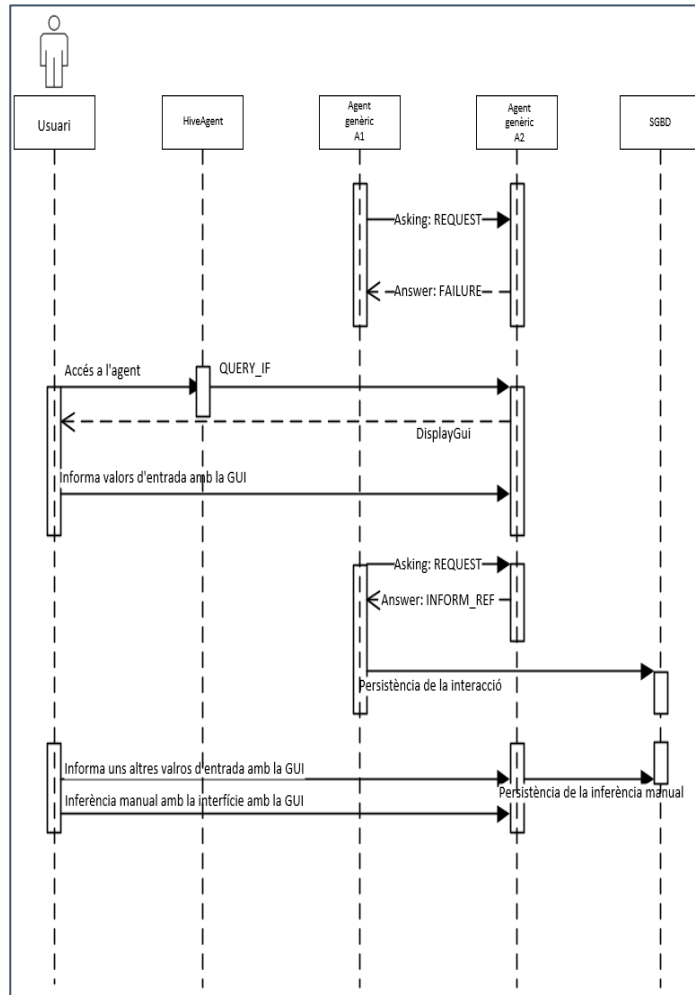


Figura 8: Perspectiva de les interaccions del sistema multi agent

Per un costat, les **interaccions entre agents genèrics, BaseAgent**, es basen en un paradigma de tipus petició/resposta.

Per l'altre, les **interaccions usuari-agent genèric**, encara que similars, es queden únicament en l'àmbit de l'agent interactuat.

Les **interaccions entre agents genèrics, esdevenen segons aquest principi**: Les variables de sortida d'un agent genèric, **resultants d'una inferència**, poden servir per informar les variables d'entrada d'un altre agent genèric.

Ara bé, per **realitzar una inferència, un agent necessita tenir informades les seves variables d'entrada**, i en aquest punt és on participa l'usuari.

2.5.3.2 Dualitat funcional dels agents genèrics

Al punt anterior s'han descrit les diverses interaccions que es donen al sistema multi agent. Dit això és molt important parar atenció a la següent subtileta:

- Un agent genèric a vegades ofereix, i a vegades pot demanar.

Efectivament els agents genèrics presenten aquesta dualitat funcional, llavors en un moment donat, un agent A1 pot demanar a un altre agent A2 les variables de sortida, però un altre agent A3 demanar a l'A1 les variables de sortida del darrer.

Aquesta dualitat funcional és contemplada per **comportament global i genèric** de tots els agents genèrics.

2.5.3.3 Agent Hive

El primer agent a detallar és l'agent Hive, agent especial que compleix amb un dels comesos més fonamental: **Iniciar el Sistema multi agent.**

La posada en marxa del sistema multiagent depèn principalment del fitxer descriptor, que com s'ha esmentat a l'apartat corresponent, modela els agents i les inferències que participen en el SMA.

El diagrama següent mostra gràficament les accions que l'agent *Hive* realitza en ser inicialitzat. La figura s'expressa de manera que els passos realitzats que pertanyen a un **comportament propi** es troben dins d'una àrea de color:

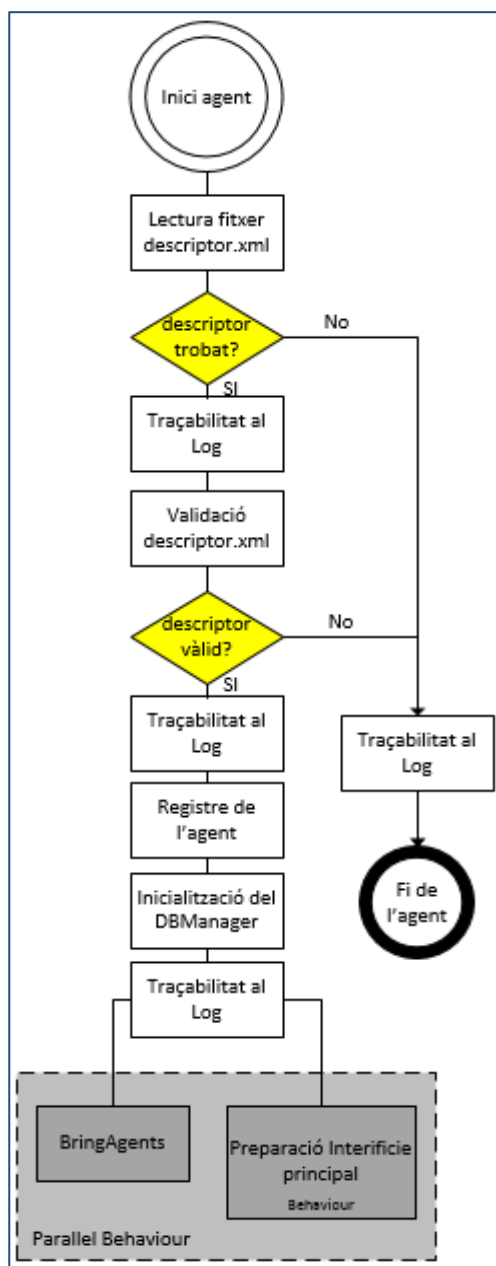


Figura 9: Inicialització de l'agent Hive

Els passos que no es troben dins d'una àrea de color no corresponen a cap comportament particular.

Als punts de decisió grocs, en cas de prendre **No**, s'efectua traçabilitat i l'agent Hive finalitza: **Per tant, no s'inicia el SMA.**

A partir del **fitxer descriptor processat**, s'obté: Llista d'agents *BaseAgent*, dades de la DDBB, etc.

DBManager s'encarrega de **persistir les interaccions dels agents.**

Paral·lelment, amb *ParallelBehaviour*, els **agents genèrics**, *BaseAgent* són **invocats** a través del comportament **BringAgents**.

També es mostra la interfície principal de l'aplicació, a través d'un comportament **Behaviour**.

La missió principal ha finalitzat, però l'agent conviu a l'entorn d'execució JADE.

2.5.3.4 Comportaments de l'agent Hive

A continuació es descriuen els comportaments associats a l'agent Hive i vetllen per la inicialització del sistema multi agent:

Nota: Cada comportament té amb parèntesi el tipus del qual estén.

Comportaments de l'agent <i>Hive</i>		
ParallelBehaviour (Natiu JADE)	Comportament natiu JADE que executa els comportaments subjacents de manera "paral·lela".	
	Comportament Subjacent	
	BringAgents (Behaviour)	Comportament propi encarregat d'invocar la creació de tants agents genèrics com agents especificats al descriptor hi hagi. A més també s'encarrega de donar nom als agents a partir de: Identificador de l'agent, nom de l'agent i la descripció de la inferència que realitzen. Tot agent genèric invocat s'inicialitza per si sol i l'agent Hive no intervé.
	Behaviour (Natiu JADE)	Comportament propi encarregat de propagar a la interfície d'usuari la llista d'agents, un cop s'han inicialitzat.

La figura següent escenifica el comportament **ParallelBehaviour** anterior:

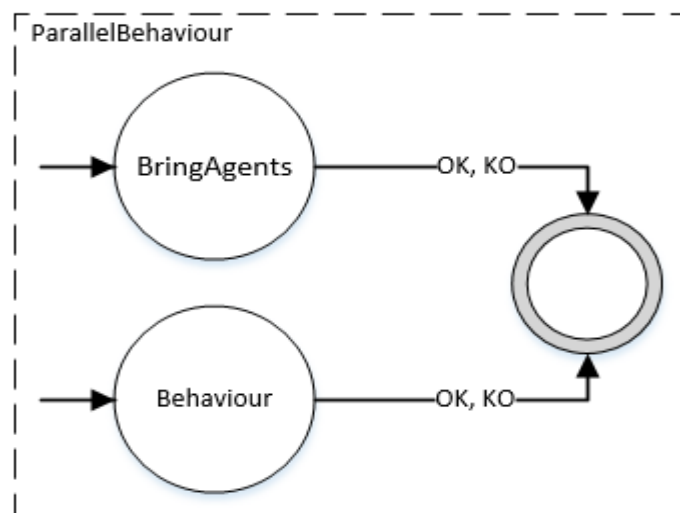


Figura 10: Comportament ParallelBehaviour de l'agent Hive

2.5.3.5 Agent genèric

Els agents genèrics, *BaseAgent*, són el fonament funcional del Sistema Multi agent. **Tots els agents tenen un comportament global comú.** Les diferències principals entre un

conjunt d'agents genèrics són les propietats adquirides: Variables i inferència, obtingudes a través del FCL associat.

Tot agent genèric, *BaseAgent*, és candidat a ser un agent que convisqui a l'entorn d'execució JADE, encara que hi ha un pas més de validació que pot fer que un agent genèric, malgrat trobar-se aparentment ben informat al descriptor, no s'inicialitzi amb èxit: **La validesa del fitxer FCL associat.**

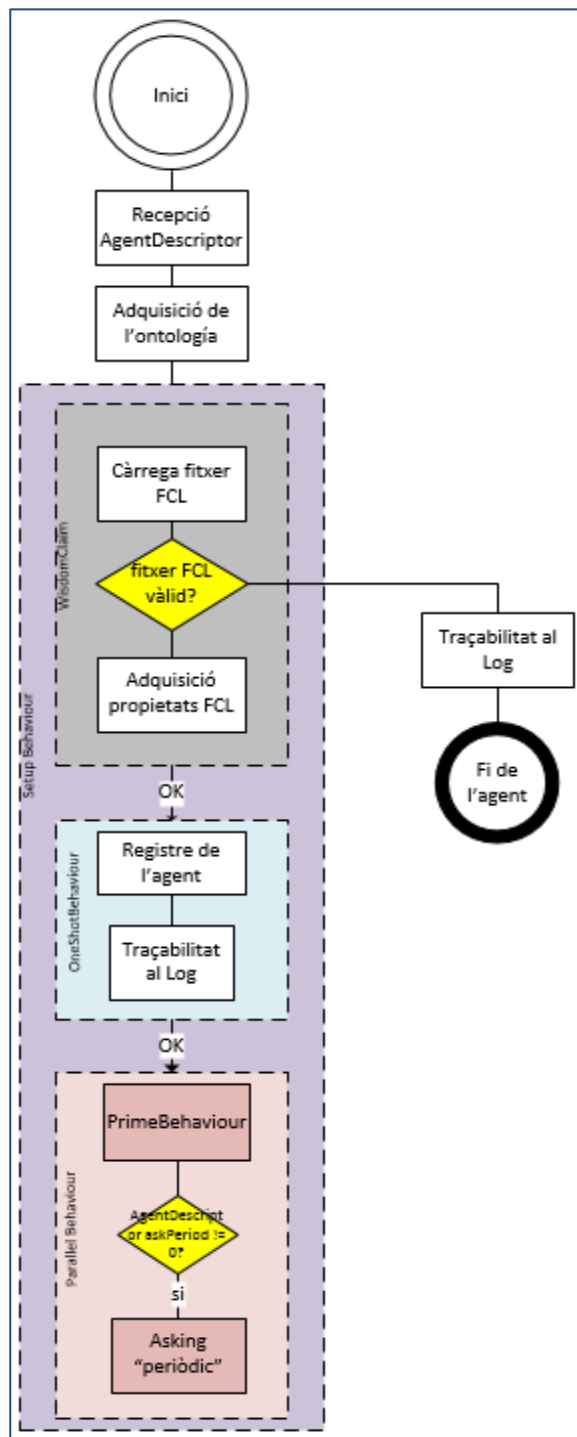


Figura 11: Inicialització de l'agent genèric, *BaseAgent*

Tot agent genèric, *BaseAgent*, a partir de les dades del descriptor rebudes a través de l'agent *Hive*, és responsable de la seva inicialització.

El comportament **SetupBehaviour** governa els comportaments subjacents encarregats d'inicialitzar l'agent. En acabar, dona pas al darrer comportament, **Parallel Behaviour**

WisdomClaim és un comportament encarregat d'intentar processar el fitxer FCL associat.

Si tot va bé, inicialitzen les propietats d'aquest agent genèric: Variables d'entrada, sortida, sistema d'inferència, etc. Altrament l'agent no s'inicialitza i per tant, no es trobarà disponible a l'entorn d'execució JADE.

El registre de l'agent genèric al DF, en cas que *WisdomClaim* acabi correctament, es realitza mitjançant un comportament basic **OneShotBehaviour**.

Per acabar, un comportament complex JADE, **ParallelBehaviour**, governa de manera paral·lela els comportaments principals de l'agent genèric: **PrimeBehaviour** i si escau, **Asking** periòdic.

El detall de *PrimeBehaviour* i *Asking* es donen a la secció [Comportaments de l'agent genèric](#)

Resulta interessant mencionar que un agent incapaç de ser inicialitzat, no compromet la inicialització d'altres agents genèrics especificats al descriptor ni la posada en marxa del SMA.

2.5.3.6 Comportaments de l'agent genèric

A continuació es descriuen els comportaments associats a l'agent genèric, *BaseAgent*, els quals vetllen per la inicialització de l'agent i pel seu comportament genèric global:

Nota: Cada comportament té amb parèntesi el tipus del qual estén.

Comportaments agent genèric, <i>BaseAgent</i>		
SetupBehaviour (FSMBehaviour)	Comportament propi encarregat de dur a terme la inicialització de l'agent genèric.	
	Comportament Subjacent	Descripció
	WisdomClaim (Behaviour)	En aquest comportament entra en joc el camp OutputInferenceld del descriptor , ja que aquest indica quina inferència del descriptor l'agent utilitzarà.
		Lectura, validació i processament del fitxer FCL associat. A partir d'aquest processament l'agent obté les seves propietats principals: Variables i sistema d'inferència. En cas d'excepció, l'agent no acaba d'inicialitzar i no serà pas a l'entorn d'execució JADE.
	OneShotBehaviour (natiu JADE)	Comportament propi encarregat de registrar l'agent, amb les seves propietats , les quals el faran cercable per altres agents genèrics : Mitjançant el comportament subjacent Asking (Executat al comportament ParallelBehaviour) Les propietats són les variables de sortida obtingudes a WisdomClaim i l' outputInferenceld del descriptor .
ParallelBehaviour (natiu JADE)	Comportament genèric de l'agent. La pròxima taula es dedica a aquest comportament.	

La següent figura permet visualitzar gràficament les transicions del comportament **SetupBehaviour**.

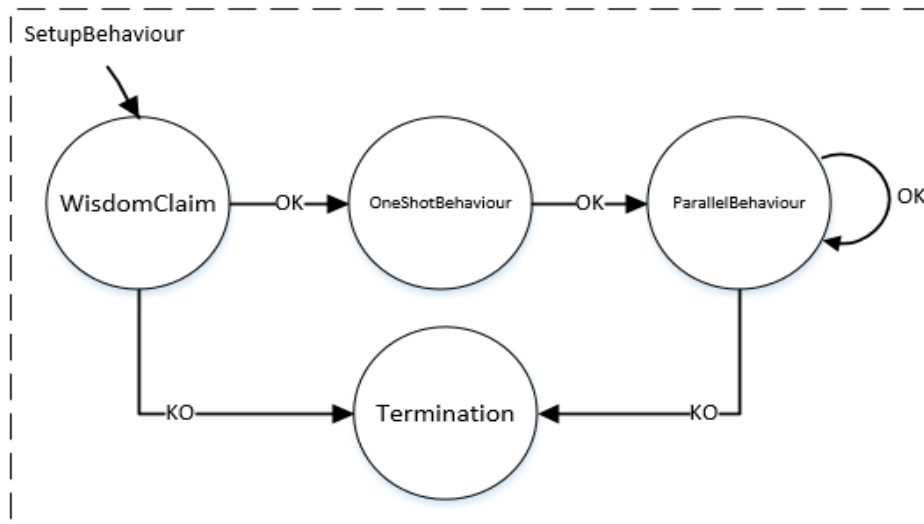


Figura 12: Comportament SetupBehaviour de l'agent genèric

ParallelBehaviour (natiu JADE)	Comportament propi encarregat d'executar, paral·lelament, els comportaments principals dels agents genèrics: PrimeBehaviour i Asking "periòdic"		
	Comportament Subjacent	Descripció	
	PrimeBehaviour (FSMBehaviour)	La pròxima taula es dedica a aquest comportament.	
	Asking (WakerBehaviour)	Comportament executat periòdicament, si s'ha informat al descriptor, askPeriod , per un agent concret.	
		Un agent A1 que executi aquest comportament cerca, mitjançant l'agent DF, altres agents que a l'inferir, generin variables de sortida utilitzables (Per l'agent A1)	
		Per fer aquesta cerca, s'utilitzen les variables d'entrada de l'agent A1. Amb aquestes, es busca coincidència amb les propietats registrades d'altres agents (Els agents sempre es registren amb les seves variables de sortida). A partir d'aquí, hi ha dues variants.	
		<table border="1"> <tr> <td>Amb el camp inputInferenceld del descriptor</td> <td>També es comprova si l' inputInferenceld del agent A1 coincideix amb la propietat registrada outputInferenceld dels agents.</td> </tr> </table>	Amb el camp inputInferenceld del descriptor
Amb el camp inputInferenceld del descriptor	També es comprova si l' inputInferenceld del agent A1 coincideix amb la propietat registrada outputInferenceld dels agents.		
	Si la modalitat anterior dona 0 resultats, s'ignora la restricció de coincidència inputInferenceld anterior.		

		Per acabar, aquest comportament genera una comunicació entre els dos agents.
--	--	--

La següent figura permet visualitzar gràficament les transicions del comportament **ParallelBehaviour**.

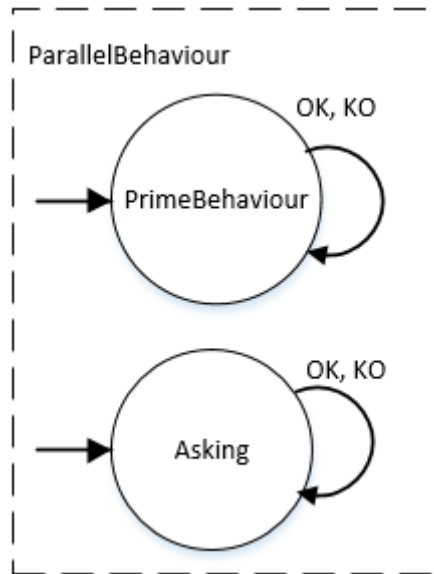


Figura 13: Comportament ParallelBehaviour de l'agent genèric

PrimeBehaviour (FSMBehaviour)	Comportament propi que materialitza el funcionament genèric de tots els agents <i>BaseAgent</i> . La dualitat funcional de l'agent genèric es troba aquí.		
	Comportament Subjacent	Descripció	
	Receiver (Behaviour)	Comportament subjacent inicial que provoca que l'agent genèric esperi a rebre qualsevol missatge.	
	Interaction (Behaviour)	Aquest comportament provoca que, a partir de la Performative del missatge rebut, l'agent executi un comportament concret:	
		<i>Performative</i> del missatge: REQUEST	Sol·licitud d'inferència d'un altre <u>agent sol·licitant</u> . Evoluciona l'agent a Inference.
		<i>Performative</i> del missatge: INFORM_REF, FAILURE	Recepció de <u>resposta d'un altre agent</u> . Evoluciona l'agent Answered.
<i>Performative</i> del missatge: QUERY_IF		Usuari sol·licita que aquest agent mostri la seva interfície gràfica. Evoluciona l'agent a DisplayGui	
Els missatges rebuts amb altres Performatives , provoquen que l'agent evolucioni a Receiver .			

	Inference (Behaviour)	Comportament que s'encarrega d'executar la inferència de l'agent, amb els valors de les variables d'entrada actuals. Efectivament, és aquest el comportament que utilitza la implementació lògica difusa JfuzzyLogic.	
		L'agent té valors a les seves variables d'entrada.	Es realitza la inferència
		L'agent no té valors a les seves variables d'entrada	No es realitza la inferència
		Es faci la inferència o no, l'agent envia una resposta. L'Agent evoluciona cap a Answer. Les inferències manuals fruit d'una interacció agent-usuari, són persistides a la base de dades per aquest comportament.	
	Answer (Behaviour)	Aquest comportament s'encarrega de preparar la resposta a l'agent genèric sol·licitant:	
		Inferència realitzada al comportament Inference.	Prepara missatge de resposta amb <i>Performative</i> : INFORM_REF
		Inferència no realitzada al comportament Inference	Prepara missatge de resposta amb <i>Performative</i> : FAILURE
		Sigui com sigui, l'agent evoluciona cap a Receiver. Finalment, aquest comportament genera una comunicació entre els dos agents.	
	Answered	Aquest comportament s'executa quan l'agent, és l'agent sol·licitant i un altre agent respon:	
		<i>Performative</i> del missatge: INFORM_REF	Processament del missatge, propagació a les variables d'entrada i persistència de resultats a la Base de dades.
		<i>Performative</i> del missatge: FAILURE	Processament del missatge sense cap impacte.
		L'agent evoluciona a Receiver.	
	DisplayGui	El comportament provoca que l'agent mostri a l'usuari la seva interfície gràfica, facilitant la interacció directa amb l'agent. Requereix la recepció d'un missatge ACL, enviat sempre per l'agent Hive a través de la interfície d'aquest.	
<i>Performative</i> del missatge: QUERY_IF		Preparació de la Interfície de l'agent.	

La següent figura permet visualitzar gràficament les transicions del comportament **PrimeBehaviour**:

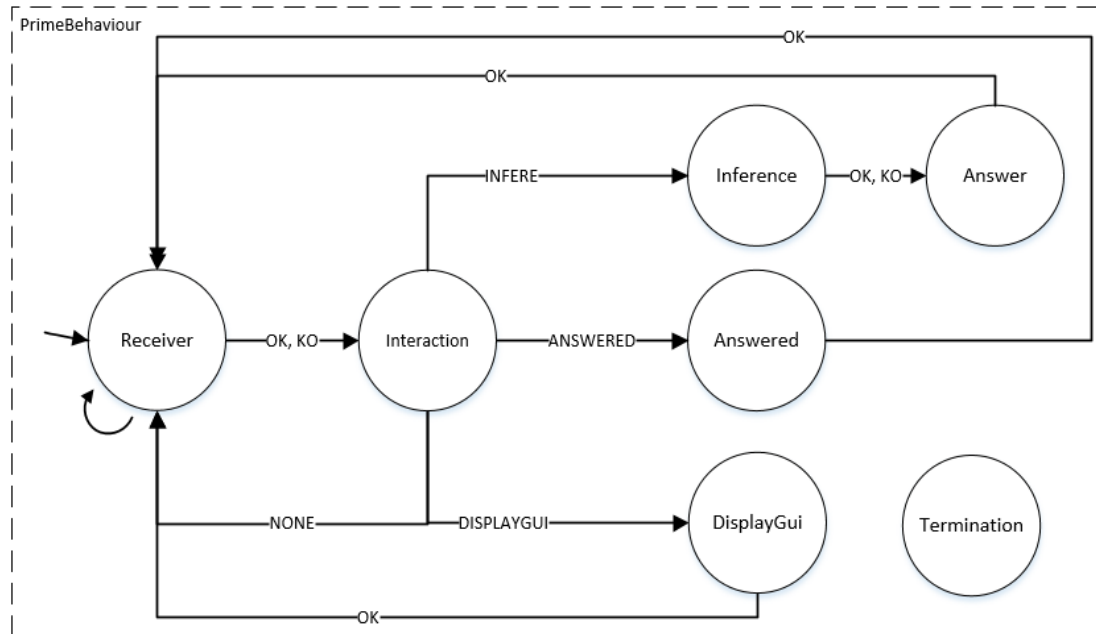


Figura 14: Comportament PrimeBehaviour de l'agent genèric

2.5.3.7 Comunicació entre agents genèrics

La comunicació entre agents genèrics, *BaseAgent*, possibilita les interaccions esmentades al començament:

- Interaccions entre agents
- Interaccions usuari-agent

Les comunicacions es donen als comportaments següents:

Comportament	Descripció	Performative del missatge ACL
Asking	L'agent demana a un altre, si n'hi ha cap, que infereixi i retorni els seus valors de sortida.	REQUEST
Answer	L'agent respon amb els seus valors de sortida a un agent sol·licitant.	INFORM_REF, FAILURE
DisplayGui	L'agent mostra la seva interfície a un usuari.	QUERY_IF

Com sap un agent que respon a qui respondre? Bàsicament es fa a partir del missatge rebut, de *Performative* REQUEST, que informa l'emissor i el receptor. Llavors és qüestió d'invertir l'emissor i receptor del nou missatge generat al comportament Answer.

De les comunicacions anteriors, l'únic missatge ACL que transporta informació funcional és el generat pel comportament Answer, sempre que es tracti de la *Performative*

INFORM_REF: És a dir, prèviament a aquesta resposta l'agent que respon ha pogut realitzar la inferència amb èxit.

La informació rebuda per un agent sol·licitant, per part de l'agent que respon és la següent:

- Variables tant de sortida com d'entrada de l'agent que respon.
- Propietats bàsiques de l'agent que respon: Nom de l'agent, nom del fitxer FCL associat i l'OutputInferenceld.

Aquesta informació serà utilitzada al moment de mostrar els valors a la interfície i **persistir la informació a la base de dades.**

2.5.3.7.1 Ontologia

Lles úniques interaccions que es troben acompanyades amb missatgeria que utilitza ontologia són les vinculades al comportament Answer amb la *Performative* INFORM_REF.

La informació normalitzada amb aquesta Ontologia enriqueix a l'agent sol·licitant i s'utilitza per persistir la interacció a la base de dades. L'ontologia desenvolupada en aquest treball consisteix a:

TfgSmaOntology		
Elements generals	Nom funcional	Descripció
Conceptes	Variable	Utilitzat per representar les variables d'un agent : Tant les de sortida com les d'entrada.
	Propietat	Utilitzat per representar algunes propietats de l'agent que respon : Nom del fitxer FCL, Nom de l'agent i OuputInferenceld.
Predicats	Variables de l'agent	Col·lecció de múltiples Variable.
	Propietats de l'agent	Col·lecció de múltiples Propietat.

Finalment es facilita el contingut, *Content*, d'un missatge de resposta amb informació funcional a partir de les normes de l'ontologia desenvolupada:

Contingut del missatge ACL	Elements rellevants missatge ACL
<pre>((variables (sequence (variable :type input :name varD :value 0.5) (variable :type input :name varC :value 0.3) (variable :type input :name varB :value 5.0) (variable :type input :name varA :value 3.0) (variable :type output :name OutB1 :value 2.5000000000000006) (variable :type output :name OutB2 :value 7.277155351341726))) (agentProperties (sequence (property :propertyName fclFileName :value "2016_1_PAC4_A1.fcl") (property :propertyName agentName :value A1) (property :propertyName inferencId :value "2")))))</pre>	Sender: Agent A1 (Respon a la sol·licitud)
	Receiver: Agent A3 (Sol·licitant, rep la resposta)
	Performative: INFORM_REF
	Language: fipa-sl
	Ontology: tfg-sma-ontology

2.5.3.8 Interfície gràfica dels agents del sistema multi agent

Per acabar amb tota la part del projecte centrada a JADE, val la pena mencionar els aspectes més rellevants de la interfície gràfica de l'agent *Hive* i els agents genèrics, *BaseAgent*.

Totes les interfícies gràfiques disponibles permeten dur a terme algunes accions, com per exemple invocar la interfície d'un agent genèric, buscar agents al SMA a partir del seu nom o descripció, entre altres.

2.5.3.8.1 Interfície gràfica de l'agent Hive

La interfície d'aquest agent són la cara i els ulls de l'aplicació com a tal:

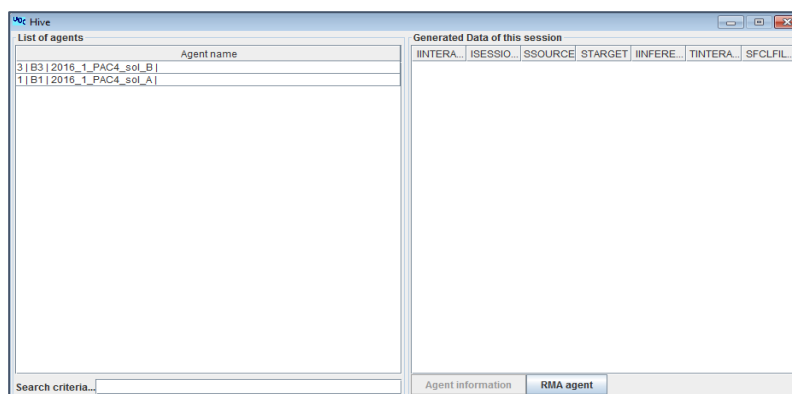


Figura 15: Interfície principal del sistema multiagent

La interfície es compon per dues seccions, ben diferenciades:

Secció Esquerra	Secció dreta	
Llistat on es pot escollir un dels agents genèrics.	Interaccions funcionals del sistema persistides a la base de dades. Destacar que aquesta versió, no permet accedir als valors de les variables involucrades mitjançant la interfície. Un workaround per consultar els valors, és consultar amb SQL la base de dades.	
Cercador d'agents a partir d'una cadena de text, Search Criteria .	Botons d'acció	Agent Information: S'invoca la interfície de l'agent genèric seleccionat.
		RMA Agent : S'invoca a l'agent RMA, Remote Administration Agent, natiu de JADE que permet accedir a un conjunt d'eines interessants.

Per acabar amb la sessió de treball, és suficient amb tancar el programa amb tancar la finestra.

2.5.3.8.2 Interfície gràfica de l'agent genèric

La interfície dels agents genèric és similar a tots els casos, però hi ha lleugeres diferències:

- El formulari per informar les variables d'entrada depèn del fitxer FCL associat a l'agent.
- El formulari o es mostren els resultats de les variables de sortida, també depèn del fitxer FCL associat a l'agent.

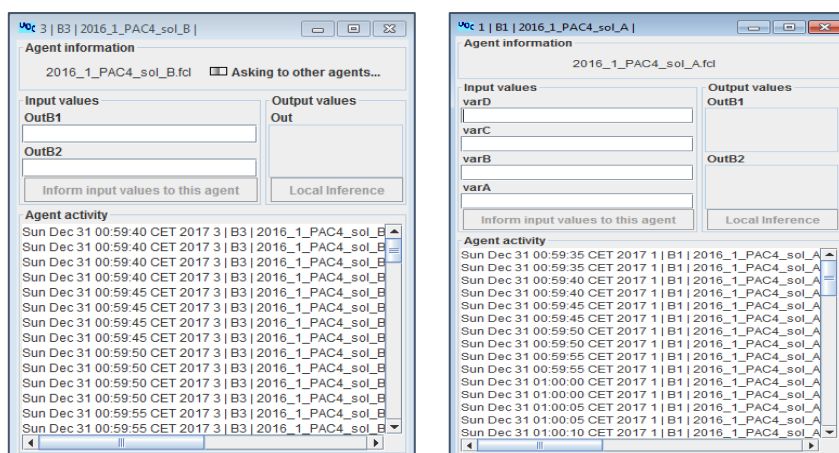


Figura 16: Interfície dels agents genèrics

Deixant a un costat la subtilesa anterior, les interfícies dels agents genèrics presenten:

Títol de la finestra	
El títol de la finestra mostra el nom atribuït per l'agent Hive al moment d'invocar-lo. (Comportament BringAgents): id + nom de l'agent + descripció de la inferència utilitzada.	
Agent Information	
Nom del fitxer FCL associat.	Opcionalment, s'indica si l'agent té el comportament Asking actiu .
Input Values	
Variables d'entrada de l'agent genèric, editables manualment.	Botó per propagar els valors introduïts a l'agent.
Output Values	
Variables de sortida de l'agent genèric, no editables manualment.	Botó per computar la inferència i obtenir els valors nítids de les variables de sortida. No editable manualment.
Agent activity	
Activitat de l'agent, són les mateixes entrades que les emmagatzemades al log, SMA.log, però només apareixen les relacionades amb les interaccions de l'agent propietari de la interfície.	

2.5.4 Implementació JFuzzyLogic

Aquest treball utilitza la implementació basada en JfuzzyLogic (Cingolani & Alcalá-Fdez) per dur a terme el raonament aproximat, basat en la teoria de conjunts difusos. La implementació com a tal és utilitzada en determinats comportaments de l'agent genèric:

Comportament	Descripció
WisdomClaim	S'utilitza JfuzzyLogic per importar el fitxer FCL, validar-lo i processar-lo. D'aquesta manera l'agent genèric adquireix les seves propietats.
Inference	Aquest comportament emprà JfuzzyLogic per realitzar les inferències a partir de les valors d'entrada actuals.

2.5.4.1 Representació del model FCL: Extensions pròpies

Pel que fa a la representació de models FCL per utilitzar amb JfuzzyLogic, anteriorment s'ha mencionat que jFuzzylogic ofereix una sintaxi pròpia per definir les funcions de pertinença dels valors lingüístics d'una variable:

Àmbit de l'extensió	Observació	
Funcions de pertinença dels valors lingüístics	Permet definir les funcions de pertinença clàssiques: Triangulars, Trapezoidals i altres de manera simplificada.	
Exemple		
Funció de pertinença triangular	FCL natiu	Extensió JfuzzyLogic

	TERM B := (0,0) (1,1) (5,0);	TERM B:= trian 0 1 5;
--	------------------------------	-----------------------

Tot i que aquesta extensió simplifica la generació de models FCL, cal advertir que el fet d'utilitzar aquestes extensions provoca que el FCL no segueixi completament l'estàndard i pot comprometre la compatibilitat: Així doncs, els FCL generats amb aquesta extensió poden ser no compatibles.

2.5.5 Persistència d'interaccions i resultats

Al text s'ha citat que les interaccions entre els diversos agents genèrics es persisteixen a una Base de dades Oracle, si es configura de manera apropiada al fitxer descriptor. Les interaccions que es persisteixen a la base de dades són les següents:

Interacció	Comportament principal involucrat	Descripció
Entre agents	Answered	Un agent sol·licitant rep un missatge d'un altre agent, amb Performative INFORM_REF
Amb usuari	Inference	L'usuari realitza una inferència manual mitjançant un agent concret.

Nota respecte al comportament Answered: A la base de dades es guarden els resultats de les variables de sortida i d'entrada de l'agent que respon, però és l'agent sol·licitant a través del comportament Answered qui fa la persistència a la base de dades.

El pròxim apartat pretén especificar el model de dades dissenyat per complir aquest comés.

2.5.5.1 Model conceptual

El model conceptual analitzat i dissenyat parteix de les següents bases funcionals:

Base funcional	Descripció	Entitat conceptual
Sessions de treball	Cada cop que el sistema s'inicia, es genera una sessió de treball.	Sessió
Interaccions de la sessió de treball	A una sessió de treball es poden generar una quantitat variable d'interaccions, sigui entre dos agents o amb l'usuari.	Interacció

Variables de les interaccions d'una sessió de treball	A tota interacció es troben involucrades una sèrie de variables d'entrada i sortida amb uns valors determinats.	Variable
---	---	----------

A partir de les bases funcionals anteriors, es planteja el següent disseny conceptual:

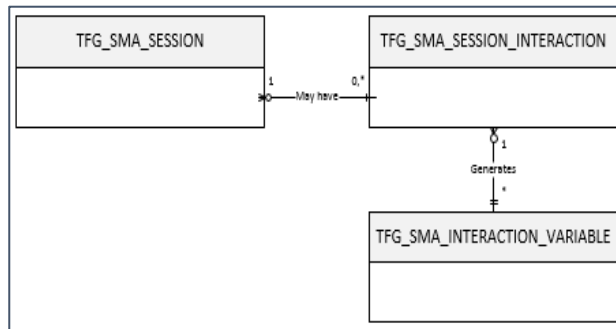


Figura 17: Disseny conceptual de la base de dades

Una Sessió, TFG_SMA_SESSION, pot tenir o no diverses Interaccions, TFG_SMA_SESSION_INTERACTION

Cada Interacció, TFG_SMA_SESSION_INTERACTION, pertany exclusivament a una Sessió i genera múltiples Variable, TFG_SMA_INTERACTION_VARIABLE

Les variables, TFG_SMA_INTERACTION_VARIABLE són resultat únicament d'una sola interacció concreta.

A l'annex 7.2 **Especificació model: Persistència de informació** es pot trobar tant el model lògic resultant.

2.6 Obstacles principals al llarg de la vida del projecte

L'execució del projecte ha estat tot un repte i com tal, sempre apareixen entrebancs al moment més inesperat. A continuació es descriuen els més rellevants al llarg de la vida del projecte:

- Valors negatius a les variables
 - Àmbit d'afectació: Inferència
 - Detall: A la fase de proves la implementació JFML va donar problemes quant a valors negatius
 - Mitigació: Estudi de les implementacions jFuzzyLogic i Jfuzzylite. Ús de la implementació JfuzzyLogic en comptes de JFML
- Resultats a inferències amb més d'un bloc de regles
 - Àmbit d'afectació: Inferència
 - Les variables de sortida d'un agent genèric A1, *BaseAgent*, són sempre nítides.

- Aquest agent A1, té associat un FCL compost FCL1 amb diverses variables de sortida (Algunes intermèdies) i blocs de regles.
 - Si l'agent A1 realitza la inferència FCL1, per si sol, tot bé.
 - En canvi, si per un motiu donat, es decideix repartir el bloc compost FCL1 en diversos FCL1a, FCL1b... FCL1n i assignar cadascun a diversos agents genèrics
 - La sortida de l'agent que incorpora el FCL1n, amb les variables de sortida final (A partir d'inferir amb les variables de sortida intermèdies d'un altre agent) tindrà un resultat diferent del resultat de les variables de sortida finals de l'agent A1.
 - Això succeeix pel fet que els valors de sortida dels agents sempre són nítids i per tant, pel primer cas, els resultats intermedis no és nitidifiquen, però si es donen nitidificats quan es fa amb múltiples agents.
- Mitigació: Proposta de millora informada al **capítol 4 Millores**.

3 Resultats obtinguts

La secció actual té la finalitat de descriure com utilitzar el sistema multi agent, a partir d'escenaris amb explicacions incrementals. Tanmateix s'aprofita per descriure breument el funcionament del sistema. S'espera que el manual, a banda d'il·lustrar el funcionament del sistema, també exposi algunes de les seves possibilitats.

Element a tractar	Descripció
Descriptor del sistema multiagent	Revisió dels aspectes del descriptor a partir d'un exemple
Sistema d'inferència difús	Revisió d'un sistema d'inferència representat a partir de l'especificació FCL
Inicialització del sistema multiagent	Posada en marxa del sistema multiagent
Interaccions del sistema multiagent	Interaccions principals del sistema.

3.1 Primer escenari: Agent genèric únic

A continuació es presenta l'escenari més senzill, on un agent interactua amb un usuari.

3.1.1 Descriptor del sistema multiagent

Fins ara s'ha especificat el descriptor del sistema multiagent, però no s'ha donat cap exemple. La idea d'aquest apartat és mostrar com és efectivament un descriptor vàlid per configurar el sistema multi agent:

```
<?xml version="1.0"?>
<d:Description xmlns:d="urn:Description"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:Description DescriptorSchema.xsd">
  <Simulation>true</Simulation>
  <ListOfAgents>
    <Agent id="1">
      <name>B1</name>
      <askPeriod>5</askPeriod>
      <outputInferenceId>1</outputInferenceId>
      <inputInferenceId>1</inputInferenceId>
    </Agent>
  </ListOfAgents>
  <ListOfInferences>
    <Inference id = "1">
      <name>PAC4</name>
      <fileName>PAC4.fcl</fileName>
      <description>Pràctica IA:PAC4</description>
    </Inference>
  </ListOfInferences>
  <DataBase>
    <driver>oracle.jdbc.driver.OracleDriver</driver>
    <connection>jdbc:oracle:thin:@localhost:1521:XE</connection>
    <user>TFG-SMA</user>
    <password>un5c413r0ns</password>
    <locale>en</locale>
  </DataBase>
</d:Description>
```

Figura 18: Descriptor SMA del primer escenari del manual

Simulation: Indicador per generar dades de simulació

ListOfAgents i **Agent:** Representa el conjunt d'agents genèrics, *BaseAgent*, del sistema.

ListOfInferences i **Inference:** Representa el conjunt d'inferències que els agents de ListOfAgents poden utilitzar

DataBase: Paràmetres de connexió de la base de dades.

D'aquesta manera a l'iniciar el sistema multi agent, hi haurà un agent que utilitzarà la inferència amb nom PAC4 i fitxer FCL PAC4.fcl.

3.1.2 Sistema d'inferència difús

La intenció d'aquest punt és realitzar una breu revisió de l'especificació FCL que utilitzarà l'agent d'aquest exemple:

VAR_INPUT: Variables lingüístiques d'entrada.

VAR_OUTPUT: Variables lingüístiques de sortida.

FUZZYFY <variableInput>: Definició dels valors lingüístics i funcions de pertinença de cadascun.

DEFUZZIFY<variableOutput>: Definició dels valors lingüístics i funcions de pertinença de cadascun. També s'especifica el mètode de nitidificació (Centre de masses) i un valor per

```

FUNCTION_BLOCK PAC4
VAR_INPUT
  Antiguitat: REAL;
  Nitidesa : REAL;
END_VAR
VAR_OUTPUT
  Estat : REAL;
END_VAR
FUZZIFY Antiguitat
  TERM Baixa := (0, 1) (3, 0) ;
  TERM Mitja := (2,0) (6,1) (8,0);
  TERM Alta := (7, 0) (9, 1) (10, 1);
END_FUZZIFY
FUZZIFY Nitidesa
  TERM Baixa := (0, 0) (60, 1) (70, 0);
  TERM Mitja := (65, 0) (100,1) (180, 0);
  TERM Alta := (170, 0) (200,1) (300,0);
END_FUZZIFY
DEFUZZIFY Estat
  TERM Noespotvendre := (0,1) (1,0);
  TERM Espotregalar := (0,0) (1,1) (2,0);
  TERM vendremoltbarat := (1,0) (2,1) (3,0);
  TERM espotvendre := (2,0) (4,1) (6,0);
  TERM bonescondicions := (5,0) (6,1) (7,0);
  TERM seminou := (7,0) (8,1) (9,0);
  TERM moltbonescondicions := (9,0) (10,1);
  METHOD : COG;
  DEFAULT := 0;
END_DEFUZZIFY
RULEBLOCK No1
  AND : MIN;
  ACCU : MAX;
  RULE 1 :IF Antiguitat IS Baixa AND Nitidesa IS Baixa THEN Estat IS vendremoltbarat;
  RULE 2 :IF Antiguitat IS Baixa AND Nitidesa IS Alta THEN Estat IS moltbonescondicions;
  RULE 3 :IF Antiguitat IS Baixa AND Nitidesa IS Mitja THEN Estat IS seminou;
  RULE 4 :IF Antiguitat IS Mitja AND Nitidesa IS Baixa THEN Estat IS Espotregalar;
  RULE 5 :IF Antiguitat IS Mitja AND Nitidesa IS Mitja THEN Estat IS espotvendre;
  RULE 6 :IF Antiguitat IS Mitja AND Nitidesa IS Alta THEN Estat IS seminou;
  RULE 7 :IF Antiguitat IS Alta AND Nitidesa IS Baixa THEN Estat IS Noespotvendre;
  RULE 8 :IF Antiguitat IS Alta AND Nitidesa IS Mitja THEN Estat IS Espotregalar;
  RULE 9 :IF Antiguitat IS Alta AND Nitidesa IS Alta THEN Estat IS bonescondicions;
END_RULEBLOCK
END_FUNCTION_BLOCK

```

Figura 19: Fitxer FCL del primer escenari del manual

defecte en cas que no s'activi cap regla.

RULEBLOCK<Nom del bloc de regles>: Conjunt de regles que relacionen les variables lingüístiques. També es dóna la definició de l'operador AND i el criteri d'agregació ACCU

Es considera oportú proporcionar una pinzellada a alguns punts del bloc **RULEBLOCK**:

Concepte	Descripció
AND: MIN	Comportament de l'operador AND basat en la clàssica funció MIN.
T-norma	L'antecedent que representa a la regla és el de valor inferior .
ACCU: MAX	Quan la variable de sortida té el mateix valor lingüístic a diverses regles, la regla considerada serà la de l'antecedent amb el valor superior .

A partir de la definició dels operadors de les regles, el criteri d'agregació i els valors d'entrada nítids: S'obté la funció de pertinença de la variable de sortida i s'obté el valor nítid.

3.1.3 Inicialització del sistema multi agent

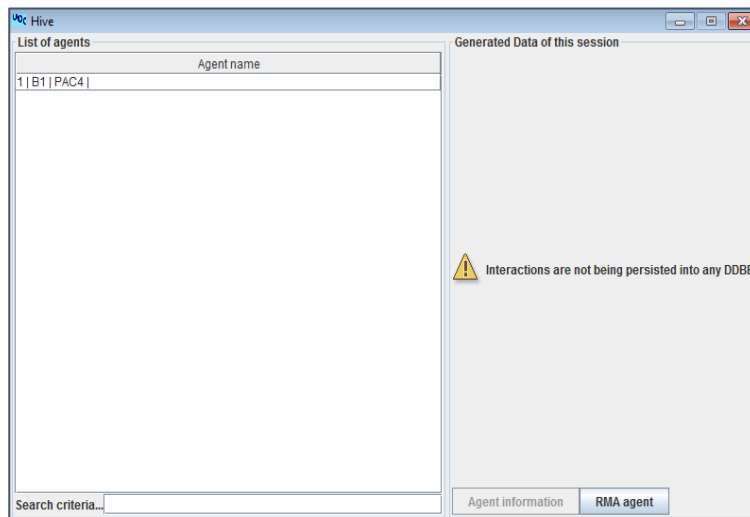
En executar l'aplicació el sistema multiagent s'inicia amb l'agent que s'ha especificat al descriptor. Els passos més rellevants són els següents:

Pas	Detall
-----	--------

Lectura, validació del descriptor	L'agent <i>Hive</i> intenta recuperar un fitxer descriptor.xml i aquest és validat contra l'esquema.
Creació de l'agent genèric	Es crea un agent genèric, BaseAgent, per l'agent informat al descriptor i ell mateix intenta inicialitzar-se a partir del comportament SetupBehaviour i la resta de comportaments subjacents. Finalment acaba executant el comportament ParallelBehaviour: PrimeBehaviour i si escau, askPeriod diferent de 0, l'Asking .
Preparació de la interfície principal	Mentre s'inicialitza l'agent, es prepara la interfície de l'agent Hive, és a dir, la cara i ulls de l'aplicació.

3.1.4 Interaccions del sistema multi agent

Un cop iniciat el sistema, aquest agent només pot interactuar amb l'usuari a través de les interfícies de l'agent Hive i l'agent genèric:



Apareix l'agent a la llista d'agents de la secció esquerra.

Per **accedir a l'agent** és suficient amb **seleccionar-lo** i prémer "**Agent information**" (Acció situada a la part inferior de la secció dreta)

Figura 20: Interfície principal del primer escenari del manual

L'agent presenta la seva interfície. A partir d'aquí:

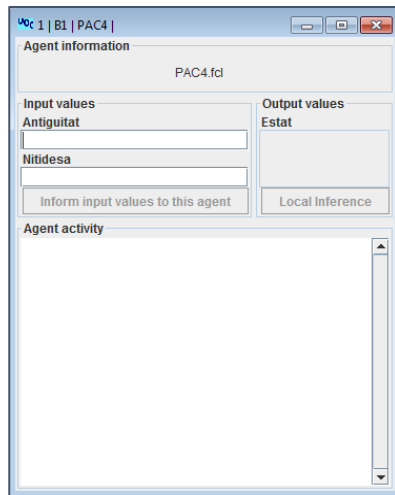


Figura 21: Interfície de l'agent del primer exemple, sense valors.

Introduir valors d'entrada
Prémer "Inform input values to this agent"

Prémer "Local Inference"

L'agent realitza la inferència amb el **comportament Inference** i ho presenta a la variable de sortida.

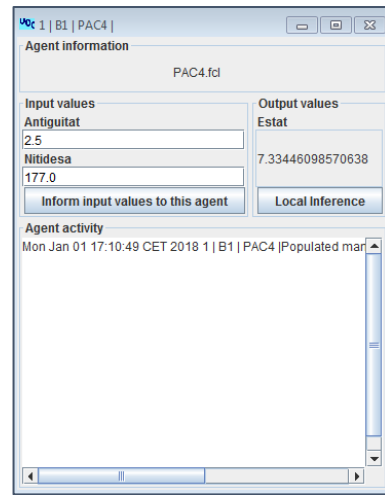


Figura 22: Interfície de l'agent del primer exemple, amb valors

Cada cop que es vulgui informar nous valors d'entrada a l'agent, cal informar-los al camp de text corresponent i prémer **Inform input values to this agent**. La nova inferència amb **Local Inference** utilitzarà els nous valors.

Aquest primer escenari serveix per assentar la base del proper i on, evidentment hi haurà coses en comú.

3.2 Segon escenari: Múltiples agents genèrics

Aquest escenari representa una situació on succeeixen totes les interaccions del sistema. Per motius de claredat, s'exposen les diferències més rellevants respecte a l'exemplificació anterior:

Diferència	Descripció
Més d'un agent	En aquest exemple hi haurà tres agents i dos d'ells es trobaran relacionats (A1 i A2). L'altre serà l'agent B1 del primer exemple, que anirà per lliure i no participarà en les interaccions dels altres dos.
Disponibilitat de la base de dades	En aquest exemple hi haurà una base de dades segons s'ha informat al descriptor.xml i per tant, les interaccions dels agents es persistiran a la Base de dades.
Interaccions autònomes dels agents	L'agent A1 tindrà informat l'element askPeriod diferent de 0, per tant, tindrà el comportament Asking periòdic i sol·licitarà a altres agents.

Ús de múltiples FCL	Cada agent A1 i A2 tindran un fcl associat diferent.
---------------------	--

3.2.1 Descriptor del sistema multi agent

Es parteix del descriptor de la primera exemplificació, però es modifica per adequar-lo a les peculiaritats d'aquesta exemplificació:

Nota: Entre parèntesis els comportaments de *PrimeBehaviour* que participen.

```
<d:Description xmlns:d="urn:Description"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:Description DescriptorSchema.xsd">
  <Simulation>true</Simulation>
  <ListOfAgents>
    <Agent id="1">
      <name>B1</name>
      <askPeriod>0</askPeriod>
      <outputInferenceId>1</outputInferenceId>
      <inputInferenceId>1</inputInferenceId>
    </Agent>
    <Agent id="2">
      <name>A1</name>
      <askPeriod>0</askPeriod>
      <outputInferenceId>2</outputInferenceId>
      <inputInferenceId>2</inputInferenceId>
    </Agent>
    <Agent id="3">
      <name>A2</name>
      <askPeriod>15</askPeriod>
      <outputInferenceId>3</outputInferenceId>
      <inputInferenceId>2</inputInferenceId>
    </Agent>
  </ListOfAgents>
  <ListOfInferences>
    <Inference id = "1">
      <name>PAC4</name>
      <fileName>PAC4.fcl</fileName>
      <description>Pràctica IA:PAC4</description>
    </Inference>
    <Inference id = "2">
      <name>2016_1_PAC4_A1</name>
      <fileName>2016_1_PAC4_A1.fcl</fileName>
      <description>Pràctica IA:PAC4_2016_1</description>
    </Inference>
    <Inference id = "3">
      <name>2016_1_PAC4_A2</name>
      <fileName>2016_1_PAC4_A2.fcl</fileName>
      <description>Pràctica IA:PAC4_2016_2</description>
    </Inference>
  </ListOfInferences>
  <DataBase>
</d:Description>
```

Figura 23: Descriptor SMA del segon escenari del manual

Relació entre agent A1 i A2:

A1 utilitzarà la inferència amb id=2 (outputInferenceId).

A2 utilitzarà la corresponent amb id=3. S'estableix que la inferència amb id =2 pot satisfer les entrades (inputInferenceId)

name: A2	name: A1
Sol·licita les variables de sortida a l'agent A1 (Asking)	Podrà satisfer a l'agent A2 quant tingui les variables d'entrada (Informades per l'usuari)
Un cop l'agent A1 disposi dels seus valors d'entrada, quan A2 li demani, realitzarà la inferència (Inference) i respondrà (Answer) amb els valors de sortida.	
Rep resposta de l'A1 (Answered) i informa les seves variables d'entrada.	

La parametrització de la base de dades és la mateixa que l'anterior, però per fer aquest exemple s'ha iniciat el SGBD a la màquina configurada.

3.2.2 Sistema d'inferència difús

Es mostren la representació FCL del sistema d'inferència dels agents A1 i A2:

```

FUNCTION_BLOCK A1
VAR_INPUT
  varA: REAL;
  varB: REAL;
  varC: REAL;
  varD: REAL;
END_VAR
VAR_OUTPUT
  OutB1: REAL;
  OutB2: REAL;
END_VAR
FUZZIFY varA
  TERM B := (0,0) (1,1) (5,0);
  TERM M := (2,0) (3,1) (6,0);
  TERM A := (2,0) (4,1) (7,0);
END_FUZZIFY
FUZZIFY varB
  TERM B := (0,0) (2,1) (6,0);
  TERM M := (2,0) (4,1) (6,0);
  TERM A := (5,0) (7,1) (9,0);
END_FUZZIFY
FUZZIFY varC
  TERM B := (0,0) (0.2,1) (0.5,0);
  TERM M := (0.2,0) (0.4,1) (0.7,0);
  TERM A := (0.5,0) (0.8,1) (1,0);
END_FUZZIFY
FUZZIFY varD
  TERM B := (0,0) (0.1,1) (0.5,0);
  TERM M := (0.4,0) (0.5,1) (1,0);
  TERM A := (0.7,0) (0.8,1) (1,0);
END_FUZZIFY
DEFUZZIFY OutB1
  TERM B := (0,0) (1,1) (5,0);
  TERM M := (2,0) (3,1) (5,0);
  TERM A := (3,0) (4,1) (5,0);
  METHOD : COG;
  DEFAULT := 0;
END_DEFUZZIFY
DEFUZZIFY OutB2
  TERM B := (5,0) (6,1) (10,0);
  TERM M := (7,0) (8,1) (10,0);
  TERM A := (8,0) (9,1) (10,0);
  METHOD : COG;
  DEFAULT := 0;
END_DEFUZZIFY

```

```

RULEBLOCK rb_OutB1
  AND : MIN;
  ACCU : MAX;
  RULE 1 :IF varA IS B AND varB IS B THEN OutB1 IS B;
  RULE 2 :IF varA IS B AND varB IS M THEN OutB1 IS B;
  RULE 3 :IF varA IS B AND varB IS A THEN OutB1 IS M;
  RULE 4 :IF varA IS M AND varB IS B THEN OutB1 IS B;
  RULE 5 :IF varA IS M AND varB IS M THEN OutB1 IS M;
  RULE 6 :IF varA IS M AND varB IS A THEN OutB1 IS A;
  RULE 7 :IF varA IS A AND varB IS B THEN OutB1 IS M;
  RULE 8 :IF varA IS A AND varB IS M THEN OutB1 IS A;
  RULE 9 :IF varA IS A AND varB IS A THEN OutB1 IS A;
END_RULEBLOCK
RULEBLOCK rb_OutB2
  AND : MIN;
  ACCU : MAX;
  RULE 1 :IF varC IS B AND varD IS B THEN OutB2 IS B;
  RULE 2 :IF varC IS B AND varD IS M THEN OutB2 IS B;
  RULE 3 :IF varC IS B AND varD IS A THEN OutB2 IS M;
  RULE 4 :IF varC IS M AND varD IS B THEN OutB2 IS B;
  RULE 5 :IF varC IS M AND varD IS M THEN OutB2 IS M;
  RULE 6 :IF varC IS M AND varD IS A THEN OutB2 IS A;
  RULE 7 :IF varC IS A AND varD IS B THEN OutB2 IS M;
  RULE 8 :IF varC IS A AND varD IS M THEN OutB2 IS A;
  RULE 9 :IF varC IS A AND varD IS A THEN OutB2 IS A;
END_RULEBLOCK
END_FUNCTION_BLOCK

```

Sistema d'inferència de l'agent A1, amb quatre variables d'entrada i dues de sortida.

Cada variable de sortida té el seu propi bloc de regles.

Figura 24: Fitxer FCL de l'agent A1 del segon escenari del manual

```

FUNCTION_BLOCK A2
VAR_INPUT
  OutB1: REAL;
  OutB2: REAL;
END_VAR
VAR_OUTPUT
  Out: REAL;
END_VAR
FUZZIFY OutB1
  TERM B := (0,0) (1,1) (5,0);
  TERM M := (2,0) (3,1) (5,0);
  TERM A := (3,0) (4,1) (5,0);
END_FUZZIFY
FUZZIFY OutB2
  TERM B := (5,0) (6,1) (10,0);
  TERM M := (7,0) (8,1) (10,0);
  TERM A := (8,0) (9,1) (10,0);
END_FUZZIFY
DEFUZZIFY Out
  TERM MB := (0,1) (0.25,0);
  TERM B := (0,0) (0.25,1) (0.5,0);
  TERM M := (0.25,0) (0.5,1) (0.75,0);
  TERM A := (0.5,0) (0.75,1) (1,0);
  TERM MA := (0.75,0) (1,1) (1,0);
  METHOD : COG;
  DEFAULT := 0;
END_DEFUZZIFY
RULEBLOCK rb_Out
  AND : MIN;
  ACCU : MAX;
  RULE 1 :IF OutB1 IS B AND OutB2 IS B THEN Out IS MB;
  RULE 2 :IF OutB1 IS B AND OutB2 IS M THEN Out IS B;
  RULE 3 :IF OutB1 IS B AND OutB2 IS A THEN Out IS M;
  RULE 4 :IF OutB1 IS M AND OutB2 IS B THEN Out IS B;
  RULE 5 :IF OutB1 IS M AND OutB2 IS M THEN Out IS M;
  RULE 6 :IF OutB1 IS M AND OutB2 IS A THEN Out IS A;
  RULE 7 :IF OutB1 IS A AND OutB2 IS B THEN Out IS M;
  RULE 8 :IF OutB1 IS A AND OutB2 IS M THEN Out IS A;
  RULE 9 :IF OutB1 IS A AND OutB2 IS A THEN Out IS MA;
END_RULEBLOCK
END_FUNCTION_BLOCK

```

Sistema d'inferència de l'agent A2, amb dues variables d'entrada i una de sortida.

Cada variable de sortida té el seu propi bloc de regles.

Figura 25: Fitxer FCL de l'agent A2 del segon escenari del manual

3.2.3 Inicialització del sistema multi agent

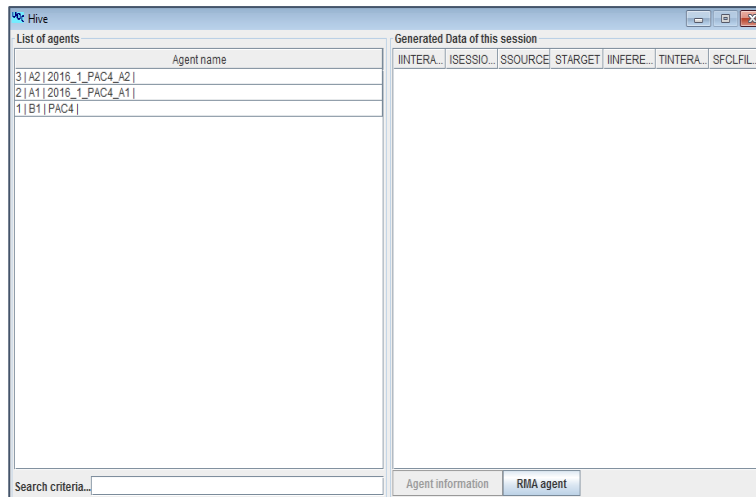
Molt similar a la inicialització de la primera exemplificació, amb les següents diferències

Nota: Entre parèntesis els comportaments de **PrimeBehaviour** que participen.

Pas	Detall
Lectura, validació del descriptor	L'agent <i>Hive</i> intenta recuperar un fitxer descriptor.xml i l'intenta validar.
Creació de l'agent genèric	Es creen tres agents genèrics amb la mateixa filosofia.
Base de dades	Aquesta exemplificació compta amb la Base de dades disponible, segons la configuració del descriptor.

3.2.4 Interaccions del sistema multi agent

De la mateixa manera que la primera exemplificació, es prepara la interfície amb els tres agents:



Apareixen els tres agents a la llista d'agents de la secció esquerra.

Si la Base de dades es troba disponible, es mostra un *grid* a la secció dreta amb les interaccions que es persisteixen al sistema

Figura 26: Interfície principal del SMA del segon escenari del manual

Agent A1

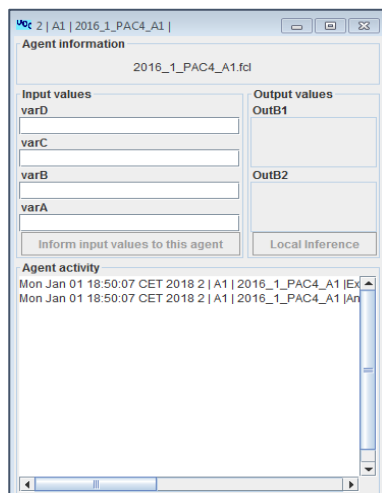


Figura 27: Interfície de l'agent A1 del segon escenari del manual

Agent A2

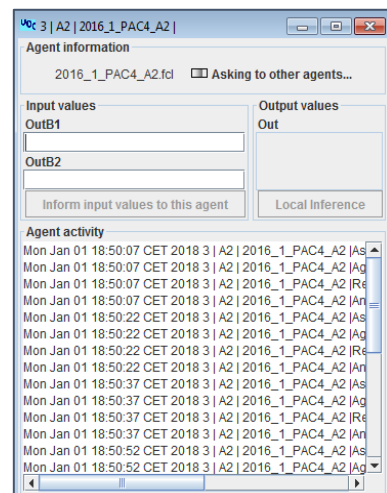


Figura 28: Interfície de l'agent A2 del segon escenari del manual

D'una banda, l'agent A2 indica que té el comportament **Asking** actiu "Asking to other agents" (Asking).

De l'altre, l'agent A1 rep la sol·licitud (Receiver) de l'A2 i respon (Inference i Answer) però la resposta encara no té cap valor.

A continuació s'informaran les variables de l'agent A1:

Agent A1

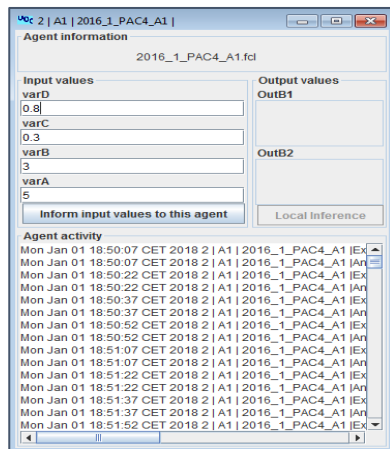


Figura 29: Interfície de l'agent A1 del segon escenari del manual, amb valors

Un cop s'informen les variables d'A1, **Local inferència** s'habilita. L'agent A1 està preparat.

Agent A2

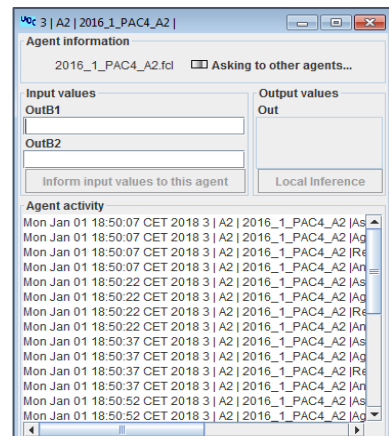


Figura 30: Interfície de l'agent A2 del segon escenari del manual, amb valors

Un cop l'agent A2 torna a sol·licitar a l'agent A1

Agent A1

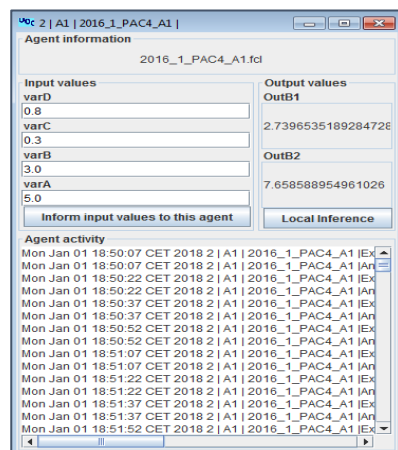


Figura 31: Interfície de l'agent A1 del segon escenari del manual, responent

L'agent A1 rep la sol·licitud de l'A2 (Receiver), realitza la inferència amb els valors informats (Inference) i respon a l'agent A2 (Answer).

L'agent A2 rep (Receiver) la resposta de l'agent A1 i la utilitza per informar les seves variables d'entrada.

Agent A2

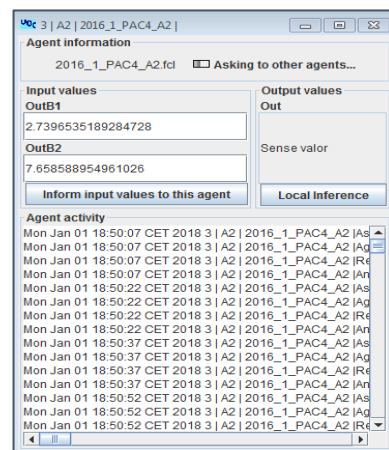
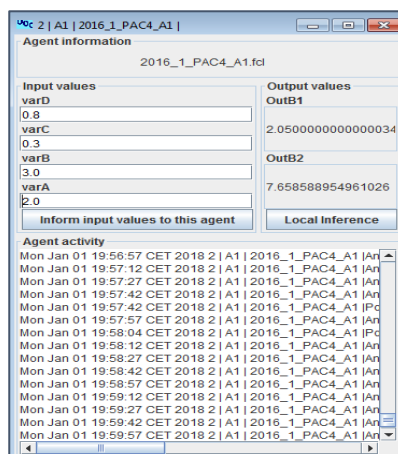


Figura 32: Interfície de l'agent A2 del segon escenari del manual, sol·licitant

Si es modifiquen els valors de l'agent A1 (**No oblidar Inform input values to this agent**)

Agent A1



El pròxim cop que l'agent A1 sol·licita a l'agent A2, s'utilitzaran els nous valors i hi haurà variacions de resultats.

Agent A2

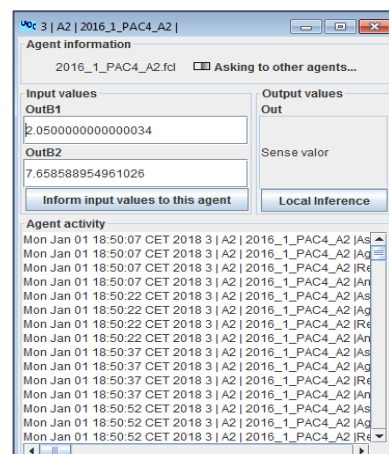


Figura 33: Interfície de l'agent A1 del segon escenari del manual, responent

Figura 34: Interfície de l'agent A2 del segon escenari del manual sol·licitant

Si l'usuari ho desitja, pot fer una inferència manual amb l'agent A2, o a l'agent A1...o al B1:

Un possible agent A3, configurat al descriptor i amb un FCL que tingues únicament com variable d'entrada la variable "Out", sol·licitaria a l'agent A2 aquest valor

Per acabar, les interaccions que s'han persistir a la Base de dades es mostren a la interfície principal, a la secció dreta amb el nom de "Generated Data of this session".

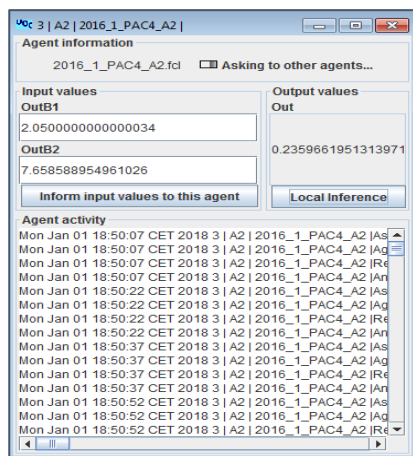


Figura 35: Interfície de l'agent A2 del segon escenari del manual, cas particular

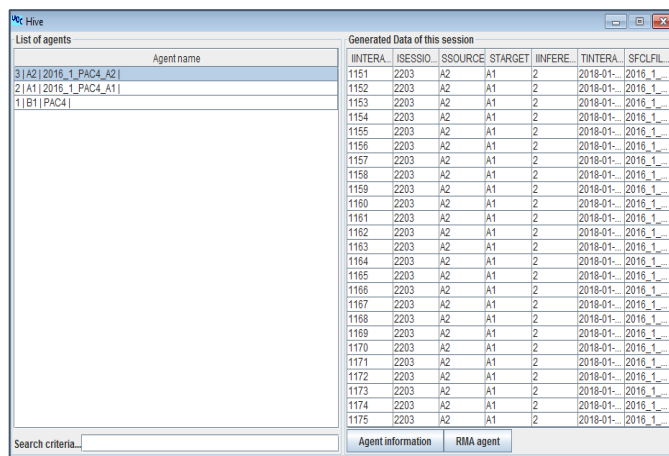


Figura 36: Interfície principal del SMA, segons el segon escenari del manual

Aquesta exemplificació mostra de manera significativa com es donen les interaccions entre agents genèrics.

3.3 Tercer escenari: Múltiples agents genèrics (Variació)

Si es parteix del segon escenari, però s'informa l'inputInferenceld de l'agent A2 amb un valor arbitrari. La idea amb aquesta exemplificació és mostrar el mode de cerca d'agents alternatiu del comportament Asking.

3.3.1 Descriptor del sistema multi agent

Es realitza una petita modificació al descriptor.xml

```
<Agent id="3">
  <name>A2</name>
  <askPeriod>15</askPeriod>
  <outputInferenceId>3</outputInferenceId>
  <inputInferenceId>3</inputInferenceId>
</Agent>
```

S'informa el camp inputInferenceld amb el mateix valor que utilitzarà l'agent A2 per fer inferències.

Figura 37: Petita modificació del descriptor, tercer escenari del manual

3.3.2 Sistema d'inferència difús

Idèntics.

3.3.3 Inicialització del sistema multi agent

Inicialització idèntica.

3.3.4 Interaccions del sistema multi agent

Les interaccions són idèntiques i l'agent A2 és capaç de trobar a l'agent A1, tot i que no s'ha informat a l'A2 quina inferència "pot satisfer" les entrades d'aquest:

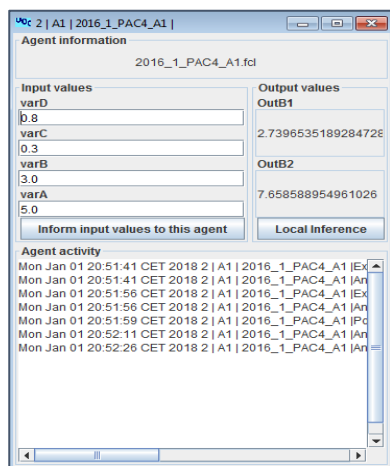


Figura 38: Interfície de l'agent A1 del tercer escenari del manual

L'agent A2 a l'executar el comportament **Asking**, intenta localitzar un agent qualsevol que satisfaci el següent: variables d'entrada i que utilitzi l'inputInferenceld informat.

Si no hi ha coincidències (Com és el cas) Asking ho torna a fer però només a partir de les variables d'entrada.

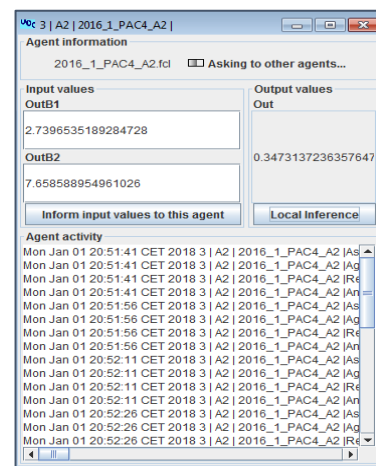


Figura 39: Interfície de l'agent A2 del tercer escenari del manual

3.4 Quart escenari: Fine-tuning d'un sistema d'inferència amb agents

Aquest sistema multiagent també pot resultar útil per optimitzar o adaptar un sistema d'inferència determinat. Per exemple, es poden configurar una sèrie d'agents que utilitzin un sistema d'inferència FCL similar però amb certes diferències respecte a: Funcions de pertinença dels valors lingüístics, regles diferents, etcètera. D'aquesta manera l'usuari pot provar variacions i prendre decisions:

3.4.1 Descriptor del sistema multiagent

Per aquest escenari, s'utilitzaran tres agents més, a part dels agents anteriors. La idea és mostrar com poden conviure agents destinats a finalitats lleugerament diferents. Els nous agents es configuren amb askPeriod 0, per no habilitar el comportament Asking i que només interactuïn amb l'usuari. Finalment, per economitzar l'espai, es facilita una captura del descriptor on només apareixen els nous agents afegits i les inferències associades:


```

<ListOfAgents>
  <Agent id="1">
  <Agent id="2">
  <Agent id="3">
  <Agent id="4">
    <name>TunningA</name>
    <askPeriod>0</askPeriod>
    <outputInferenceId>4</outputInferenceId>
    <inputInferenceId>4</inputInferenceId>
  </Agent>
  <Agent id="5">
    <name>TunningB</name>
    <askPeriod>0</askPeriod>
    <outputInferenceId>5</outputInferenceId>
    <inputInferenceId>5</inputInferenceId>
  </Agent>
  <Agent id="6">
    <name>TunningC</name>
    <askPeriod>0</askPeriod>
    <outputInferenceId>6</outputInferenceId>
    <inputInferenceId>6</inputInferenceId>
  </Agent>
</ListOfAgents>
<ListOfInferences>
  <Inference id = "1">
  <Inference id = "2">
  <Inference id = "3">
  <Inference id = "4">
    <name>PAC4_a_fineTunning</name>
    <fileName>PAC4_a.fcl</fileName>
    <description>Fine-tunning:PAC4_a</description>
  </Inference>
  <Inference id = "5">
    <name>PAC4_b_fineTunning</name>
    <fileName>PAC4_b.fcl</fileName>
    <description>Fine-tunning:PAC4_b</description>
  </Inference>
  <Inference id = "6">
    <name>PAC4_c_fineTunning</name>
    <fileName>PAC4_c.fcl</fileName>
    <description>Fine-tunning:PAC4_c</description>
  </Inference>
</ListOfInferences>

```

S'informen els agents amb la mateixa filosofia.

Anàlogament, es defineixen les noves inferències.

Figura 40: Descriptor del SMA del quart escenari del manual

3.4.2 Sistema d'inferència difús

Amb el propòsit d'ajustar un sistema d'inferència determinat a partir de proves, es parteix del fcl PAC4.fcl i es generen tres variacions: PAC4_a.fcl, PAC4_b.fcl i PAC4_c.fcl. A continuació es faciliten els canvis que pateix cada sistema d'inferència respecte a l'original, PAC4.fcl:

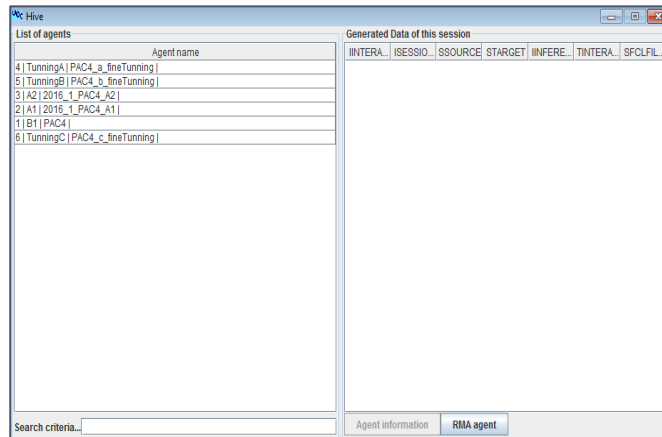
Sistema d'inferència afectat	Modificació FCL
PAC4_a.fcl	Variable: Antiguitat TERM Mitja := (1,0) (5,1) (7,0);
PAC4_b.fcl	Variable Nitidesa TERM Mitja := (65, 0) (100,1) (150, 0); TERM Alta := (195, 0) (200,1) (300,0);
PAC4_c.fcl	Variable Antiguitat TERM Alta := (3, 0) (6, 1) (10, 1);

3.4.3 Inicialització del sistema multiagent

La inicialització del SMA és similar però s'afegeixen tres agents més al sistema multi agent.

3.4.4 Interaccions del sistema multi agent

A continuació s'exposa la explotació d'aquest escenari:



S'aprecia com s'han inicialitzat els agents de nova incorporació al descriptor.

Així mateix, com la llista d'agents comença a créixer...Pot ser interessant utilitzar el cercador d'agents: **Search criteria**.

Com sempre, cal seleccionar els agents d'un en un i prémer Agent Information.

Figura 41: Interfície principal del SMA al quart escenari del manual

Per aquest escenari, cal seleccionar els agents TunningA, TunningB, TunningC i l'agent B1. A continuació, es mostren les interfícies dels 4 agents objectius. Recordar que aquests agents únicament interactuen amb l'usuari. La resta d'agents del sistema van per lliure i poden estar realitzant les seves activitats corresponents.

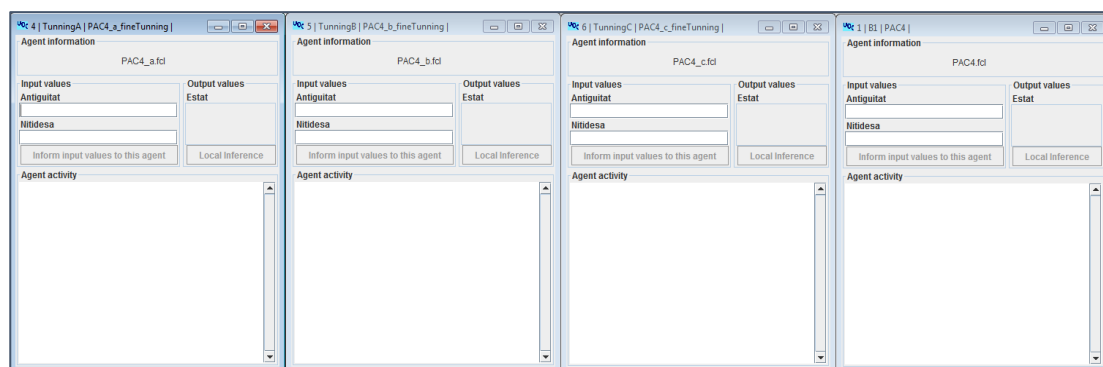


Figura 42: Interfícies dels agents genèrics del quart escenari del manual

Per cada agent, es poden informar valors, comprovar els resultats i apreciar les diferències:

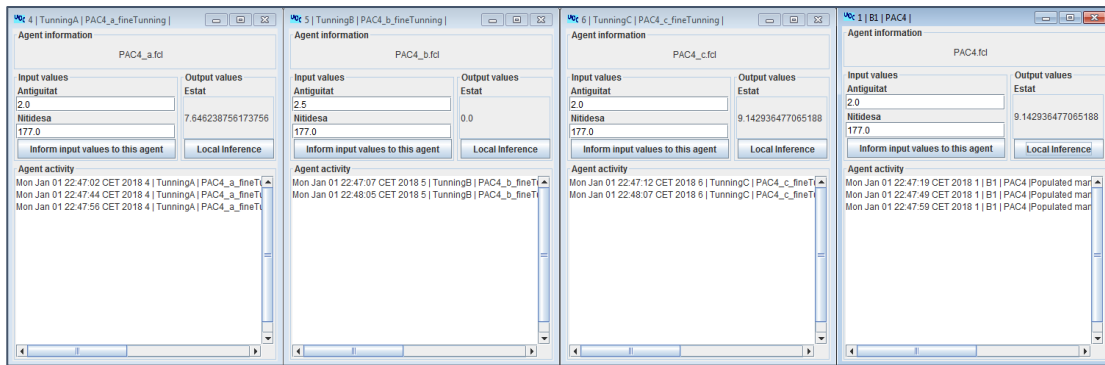


Figura 43: Interfície dels agents genèrics del quart escenari del manual, amb valors

Finalment destacar que totes aquestes interaccions s'han persistit a la Base de dades.

4 Millores i treball posterior

A continuació s'exposen totes aquelles millores identificades fins avui, que poden aportar valor al conjunt del treball:

- Primerament, avaluar si desenvolupar o no els **objectius addicionals** tipificats a la planificació del projecte.
- Revisió de l'obstacle: Nitidificació de resultats a inferències amb més d'un bloc de regles informat al punt **2.6 Obstacles principals al llarg de la vida del projecte** d'aquest mateix document
- Incorporar a la interfície dels agents genèrics traçabilitat gràfica de les interaccions on participa
 - De manera similar a com es mostra amb l'agent RMA de JADE
- Mostrar els valors de les variables involucrades a cada interacció mitjançant a través de la interfície principal.
- Solució de possibles *bugs* que puguin aparèixer a les primeres *releases* del producte.
- Possibilitat d'afegir agents i inferències a partir d'una interfície i emmagatzemar-ho al descriptor.
- Revisió del producte per executar múltiples descriptors alhora, o múltiples instàncies del sistema.
- Preparar *script* ANT per facilitar la creació de noves versions del sistema multi agent.
- Altres millores proposades pels usuaris de l'aplicació.

5 Anàlisi dels resultats i conclusions

Un cop finalitzada la primera iteració del sistema i després d'oferir una visió tant tècnica com funcional de què pot fer el sistema i com ho fa, és moment d'exposar les conclusions derivades del desenvolupament del projecte.

Els resultats obtinguts fins ara i exposats al llarg d'aquest text, posen en manifest que en general, es poden considerar satisfactoris. D'una banda el text exposa com els objectius principals definits a l'inici del projecte s'han fet realitat en tots els sentits tècnics i de l'altre, les diverses exemplificacions funcionals aplicades mostren com el context dels problemes i necessitats exposats a l'inici del projecte poden ser solucionats a través d'aquest treball.

Dins dels objectius, podem distingir entre aquells objectius que formen més part de l'enriquiment de l'estudiant i d'altres, que contribueixen directament al resultat final del producte desenvolupat. Respecte als primers, puc assegurar que he adquirit un coneixement respectable de JADE i les implementacions de lògica difusa que hi ha al mercat. Tanmateix aquest treball ha servit per recordar i enriquir tot el saber relacionat amb el raonament aproximat adquirit al passat, per tant, considero que els objectius d'aquesta naturalesa han estat assolits.

Pel que fa als objectius més orientats als resultats del projecte, cal dir que donada la naturalesa genèrica del projecte, extreure uns resultats o conclusions molt concrets basats en una aplicació pràctica del projecte no es poden donar directament: Tot i que les diferents proves realitzades si donen resultats satisfactoris en termes del funcionament del sistema plantejat, interaccions, resultats de les inferències, etc.

Continuant amb la mateixa direcció, el manual exemplificador desenvolupat en aquest treball, estableix els fonaments de les conclusions funcionals que es donen a continuació: S'ha desenvolupat un producte amb potencial, que pot ser utilitzat àmpliament en un número molt extens de dominis d'aplicació i aporta valor a usuaris que es trobin immersos al món del disseny, desenvolupament i proves de sistemes basats en lògica difusa.

Aportar valor en el sentit d'automatitzar, accelerar o posar en marxa, a través d'aquest sistema, processos funcionals repetitius, simulacions i proves d'una manera àgil i accessible, ja que, un cop es configura el descriptor/FCL d'una escena multiagent determinada, aquest pot ser reutilitzat com a punt de partida per altres escenaris o proves. Sense oblidar que tota la informació generada a les diferents sessions de treball es troba de manera persistent a la base de dades.

Finalment, és cert que l'aplicació encara presenta algunes mancances, com les exposades al capítol de millores o als obstacles. Tot i això considero que l'aplicació és un bon punt de partida per involucrar a usuaris o gent d'aquest univers de treball i col·laborar per acabar un producte que encara, pot ser molt millor.

6 Bibliografia

A continuació es presenta la bibliografia on es basen els continguts d'aquest treball:

(sense data). Recollit de SeSAm Multi agent simulation: <http://www.simsesam.de/>

Bellifemine, F., Caire, G., & Greenwood, D. (2007). *Developing multi-agent systems with JADE*. WILEY.

Cingolani, P., & Alcalá-Fdez, J. (sense data). *jFuzzyLogic*. Recollit de jFuzzyLogic: http://jfuzzylogic.sourceforge.net/html/pdf/Cingolani_Alcala-Fdez_jFuzzyLogic_2013_IJCIS.pdf

Contributors, W. (14 / Novembre / 2017). *Fuzzy markup language*. Recollit de Wikipedia: https://en.wikipedia.org/w/index.php?title=Fuzzy_markup_language&oldid=810365907

González Morcillo, C. (sense data). *2011_Softcomputing*. Recollit de Escuela Superior de informàtica:
http://www.esi.uclm.es/www/cglez/downloads/docencia/2011_Softcomputing/LogicaDifusa.pdf

INTERNATIONAL ELECTROTECHNICAL COMMISSION (IEC). (1997). *IEC 1131 - PROGRAMMABLE CONTROLLERS, Part 7 - Fuzzy Control Programming (Draft)*.

Isern Alarcón, D. (sense data). Recollit de Departament d'Enginyeria Informàtica i Matemàtiques: deim.urv.cat/descarrega/report/DEIM-RT-05-001

Millán Valdeperas, E. (sense data). Recollit de Lenguajes y ciencias de la computación. Universidad de Málaga: <http://www.lcc.uma.es/~eva/aic/apuntes/fuzzy.pdf>

Moreno i Ribas, A. (sense data). Aprenentatge Computacional. *A Agents i sistemes multiagent*. Universitat Oberta de Catalunya.

Oracle Corporation. (sense data). *Oracle Database 11g Release 2 JDBC Drivers*. Recollit de <http://www.oracle.com/technetwork/apps-tech/jdbc-112010-090769.html>

Oracle Corporation. (sense data). *Oracle Database Express Edition 11g Release 2*. Recollit de <http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>

Rada-Vilela, J. (sense data). *jFuzzyLite*. Recollit de FuzzyLite: <https://www.fuzzylite.com/java/>

Soto-Hidalgo, J., Alonso, J. M., & Alcalà-Fdez, J. (sense data). Recollit de Java Fuzzy Markup Language: <http://www.uco.es/JFML/documentation>

Torra i Reventós, V., & Godo i Lacasa, L. (sense data). Intel·ligència artificial. A *Incertesa i raonament aproximat: Sistemes difusos i xarxes bayesianes*. Universitat Oberta de Catalunya.

Wikipedia, C. d. (9 / Febrero / 2017). *Razonamiento aproximado*. Recollit de Wikipedia: https://es.wikipedia.org/w/index.php?title=Razonamiento_aproximado&oldid=96804343

7 Annexos

7.1 Aspectes transversals

Aquesta secció dedica text a assenyalar aspectes interessants del sistema multiagent:

7.1.1 Preparació del model de dades de la Base de dades

El sistema multiagent està dissenyat per preparar el model de dades de la base de dades, si aquest no existeix, al sistema gestor configurat al descriptor. Amb preparació del model de dades, s'entén el següent:

Tasca DDL	Descripció
Creació de les taules del sistema	Si no existeixen a la base de dades destí actual, es creen totes les entitats del disseny proposat.
Creació de les seqüències	Si no existeixen a la base de dades destí actual, es creen totes les seqüències involucrades en la generació de claus

	primàries per les entitats del model del sistema.
--	---

Així doncs, un cop configurada la Base de dades al descriptor, no cal cap acció per part de l'usuari / instal·lador.

7.1.2 Logging

La implementació del *logging* del sistema multiagent s'ha dut a terme a partir de la classe Java: `java.util.logging.Logger`.

Els nivells de Logging utilitzats al llarg de l'aplicació són:

Nivell de logging	Justificació
SEVERE	Quan el sistema multi agent no pot iniciar per diverses causes: No troba el fitxer Descriptor, el descriptor és invàlid, etc.
	Quan no es troba la Base de dades o no es pot accedir.
WARNING	Excepcions produïdes per qualsevol dels comportaments dels agents:
INFO	Traçabilitat informativa: Per exemple, quan s'inicia el SMA, o es registren amb èxit els agents.

7.2 Especificació model: Persistència de informació

A continuació s'exposa el model lògic dissenyat per persistir la informació resultant de les interaccions que es donen al sistema multiagent:

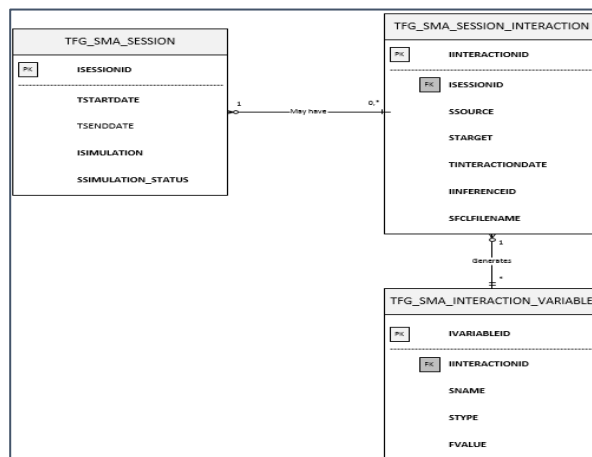


Figura 44: Disseny lògic de la base de dades

El disseny físic del model es troba al codi del sistema multiagent, a la classe: `tfg.sma.ddbb.DbConstants`

7.3 Manual d'instal·lació del sistema multiagent

En aquest annex es descriuen els aspectes més rellevants a l'hora d'instal·lar el sistema multiagent d'aquest treball.

7.3.1 Requisits de maquinari

Els requeriments de maquinari corresponen als mateixos que es tipifiquen per JADE. (Java Agent Development Framework, sense data)

7.3.2 Requisits de programari

La taula següent especifica els requisits de software per executar el sistema multi agent:

Requisit de software	Descripció
S.O	Qualsevol S.O amb Java amb arquitectura x64. Es recomana la versió de Windows 7 o posterior.
Java	Versió per arquitectura x64. Compatibilitat validada amb la versió 1.8.0_151
Base de dades	Oracle Database 11g

A continuació es llisten les llibreries utilitzades al sistema multiagent:

Llibreria	Versió, nom al projecte	Descripció
JADE	4.5.0, jade.jar	Plataforma JADE.
JFuzzyLogic	jFuzzyLogic.jar	Implementació de la teoria de conjunts difusos i lògica difusa.
Oracle Database 11g Release 2 JDBC Drivers	11.2.0.4, ojdbc6.jar	Llibreria per poder interactuar amb la base de dades Oracle.

Totes aquestes llibreries es troben incrustades al JAR de l'aplicació.

7.3.3 Instal·lació

El sistema multiagent s'entrega compilat en forma de JAR executable. A partir cal assegurar els punts següents:

- Compliment dels requisits de software
- Existència dels directoris de treball i fitxers mínims

7.3.3.1 Existència dels directoris de treball i fitxers mínims de treball

Al mateix nivell que el jar executable TFG-SMA.jar han d'existir els següents directoris:

config: Correspon al directori del descriptor.xml i DescriptorSchema.xsd

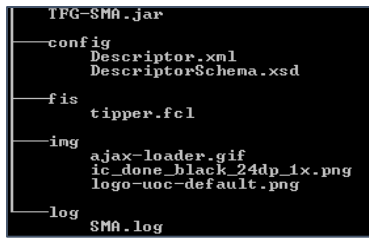


Figura 45: Directoris de treball del sistema multiagent

fis: Correspon al directori dels sistemes d'inferència FCL.

Nota: No es fa cerca recursiva de fitxers fcl, per tant els fitxers fcl a directoris interiors no seran localitzables.

Img: Directori d'icones de les interfícies.

log: Directori del log del sistema multiagent SMA.log.

Tot i que la *release* del projecte ja facilita l'aplicació compilada i preparada per ser utilitzada, pot ser interessant consultar l'annex **7.4 Descripció del projecte Eclipse i creació de noves versions** per poder preparar una nova versió del sistema multiagent.

7.4 Descripció del projecte Eclipse i creació de noves versions

A continuació es descriu l'estructura del projecte Eclipse del sistema multiagent d'aquest treball:

7.4.1 Perspectiva del projecte

El projecte presenta la següent estructura:

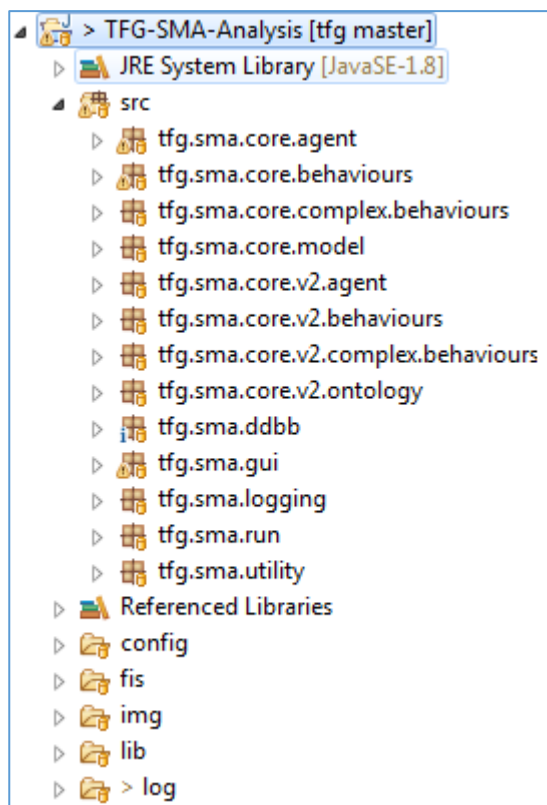


Figura 46: Estructura del projecte Eclipse

Paquets *legacy* del prototip: Desenvolupats al principi del projecte per assentar els fonaments

tfg.sma.core.agent

Agents desenvolupats primitius.

tfg.sma.core.behaviours

Iteracions inicials dels comportaments.

tfg.sma.core.complex.behaviours

Iteracions inicials dels comportaments complexos.

Paquets actuals: Darrera iteració de les classes del projecte

tfg.sma.core.v2.agent:

Agent *Hive* i agent genèric, *BaseAgent*

tfg.sma.core.v2.behaviours

Comportaments simples dels agents

tfg.sma.core.v2.complex.behaviours

Comportaments composts dels agents

tfg.sma.core.v2.ontology

Ontologia que utilitzen els agents durant la comunicació de resultats.

tfg.sma.ddbb

Es pot veure com el projecte també conté els directoris descrits a la part de configuració de l'agent. Si s'executa el projecte a Eclipse, s'utilitzen aquests.

El directori **lib** conté les **llibreries utilitzades al projecte**. La versió compilada del producte les incrusta al JAR executable.

Classes que gestionen la persistència a la Base de dades

tfg.sma.gui

Classes per generar les interfícies dels agents

tfg.sma.logging

Classes per realitzar la tasca de logging

tfg.sma.run

Classe executora, necessària per compilar el projecte com a JAR executable.

tfg.sma.utility

Classe amb utilitats de diversa naturalesa:

Funcions generals d'agents, entre altres.

7.4.2 Repositori de codi

L'evolució del projecte es troba traçada pels diferents push/comentaris realitzats al repositori GIT del projecte. L'adreça del repositori GIT és la següent:

<https://jcaballeropacheco@devca.git.cloudforge.com/tfg.git>

Per accedir al repositori cal disposar de les credencials necessàries. Qualsevol interessat pot contactar amb l'autor per aconseguir accés al repositori.

7.4.3 Nova versió del sistema multi agent

Les causes que poden motivar la necessitat de crear una nova versió del sistema multi agent són les següents:

- Resolució de Bugs
- Actualitzacions de les llibreries subjacents
- Canvis o actualitzacions en parts del codi

Per tant, per generar una nova versió del sistema multi agent, cal:

Pas	Descripció
Fer els canvis al projecte	Per exemple, actualitzar JADE a una versió posterior futura.
Fer proves	Validar que els canvis realitzats no tenen impactes negatius.
Crear el jar executable	Amb Eclipse es pot fer gràficament.
Un cop obtingut el JAR, crear/copiar els directoris de treball i fitxers esmentats a l'annex:	
7.3.3.1 Existència dels directoris i fitxers de treball	

7.5 Especificació completa model Descriptor

A continuació es presenta el model de dades del descriptor del sistema multiagent. Convé advertir que **tots els elements del model XSD són obligatoris**. El XSD mínim correspon amb la darrera entrega del projecte:

7.5.1.1 Mode de treball

Element XSD	Tipus de dada
Simulation	simple, Booleà

Descripció de cada atribut/element

Nom	Descripció
Simulation	Indica si les dades generades de la sessió i persistents a la Base de dades són simulació o no.

7.5.1.2 Agents

Elements XSD	Tipus de dada	
ListOfAgents	Complex, tipus Agents	
Agent	Complex, tipus AgentFeatures	
AgentFeatures	Atributs AgentFeatures	
	Nom atribut	Tipus
	id	enter
	Elements AgentFeatures	
	Nom element	Tipus
	name	Cadena de caràcters
	askPeriod	enter
	outputInferenceld	enter
	inputInferenceld	enter

Descripció de cada atribut/element

Nom	Descripció
id	Identificador únic d'aquest agent
name	Nom de l'agent
askPeriod	Informat a 0 s'ignora, contràriament estableix la periodicitat d'aquest agent per sol·licitar automàticament inferències a altres agents amb el comportament Asking.
outputInferenceld	Identificador de la inferència que caracteritzarà a l'agent (Comportament WisdomClain). Obligatòriament ha de ser un id definit a la secció d'inferències.

inputInferenceld	Identificador de la inferència utilitzat a l'hora de cercar altres agents (Comportament Asking). Obligatòriament ha de ser un id definit a la secció d'inferències
------------------	---

7.5.1.3 Inferències

Elements XSD	Tipus de dada	
ListOfInferences	Complex, tipus Inferences	
Inference	Complex, tipus InferenceFeatures	
InferenceFeatures	Atributs InferenceFeatures	
	Nom atribut	Tipus
	id	enter
	Elements InferenceFeatures	
	Nom element	Tipus
	name	Cadena de caràcters
	fileName	Cadena de caràcters
	description	Cadena de caràcters

Descripció de cada atribut/element

Nom	Descripció
id	Identificador únic d'aquesta inferència
name	Nom de la inferència
fileName	Nom del fitxer FCL associat
description	Descripció informativa de la inferència

7.5.1.4 Base de dades

Elements XSD	Tipus de dada	
DataBase	Complex, tipus Database	
Database	Elements InferenceFeatures	
	Nom element	Tipus
	driver	Cadena de caràcters
	connection	Cadena de caràcters
	user	Cadena de caràcters
	password	Cadena de caràcters
	locale	Cadena de caràcters

Descripció de cada atribut/element

Nom	Descripció
driver	Driver JDBC oracle a utilitzar
connection	URL de connexió a la base de dades oracle
user	Usuari de connexió

password	Paraula de pas de connexió
locale	Localització de la Base de dades.

És obligatori informar aquest node per superar la validació del model. Si no convé, es poden informar els camps amb valors *dummy*.

7.5.1.5 Validació i restriccions

Les següents validacions i restriccions asseguren que el fitxer descriptor disposi d'un mínim d'integritat i sentit:

Validació	Descripció
Document ben format	Es comprova que el descriptor contingui tots els elements obligatoris.
Unicitat dels atributs id	Els atributs id dels agents i les inferències han de ser únics. No pot haver dos agents amb el mateix id ni dues inferències amb el mateix id. En canvi, si pot haver-hi un agent amb un id que coincideixi amb el id d'una inferència.
Integritat referencial	Es comprova que els agents facin referència a sistemes d'inferència que es trobin modelats al descriptor. No pot haver cap agent que referenciï a inferències amb id inexistent.

S'ha de tenir en compte que aquestes validacions, no asseguren que el fitxer FCL sigui vàlid. La validació de cada fitxer FCL és realitzada per cada agent genèric al moment de la seva inicialització.