



UNIVERSIDAD DE JAÉN

Desarrollo de un Sistema Multi-Agente para
automatizar Procesos de Consenso en Problemas de
Toma de Decisión en Grupo

Trabajo Tutelado de Iniciación a la Investigación

Alumno: Iván Palomares Carrascosa

Tutor: Dr. Luis Martínez López

Curso 2009/2010

Jaén, Diciembre 2010

Índice general

Índice de figuras	v
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	4
1.3. Estructura	5
2. Modelos de Consenso en TDG	7
2.1. Problemas de Toma de Decisión en Grupo. Conceptos Básicos	7
2.2. El Consenso en Problemas de TDG	11
2.2.1. Procesos de Consenso	13
2.2.2. Problemas y Retos Futuros en los Procesos de Consenso	15
2.3. Algunos Modelos de Consenso empleados en Problemas de TDG	18
2.3.1. Modelo Teórico del Proceso de Consenso	18
2.3.2. Modelo de Zadrozny con Información Difusa	20

2.3.3.	Modelo de Bryson con Vectores de Preferencia	22
2.3.4.	Modelo de Kim para Problemas de TDG Multicriterio	24
2.3.5.	Modelo de Herrera-Viedma con diferentes Estructuras de Preferencia	25
2.4.	Modelo de Consenso propuesto	27
3.	Sistemas Multi-Agente	33
3.1.	Agentes Software	33
3.1.1.	Tipos de agentes	35
3.2.	Arquitecturas de Diseño de Agentes	37
3.2.1.	Arquitecturas Deliberativas. La Arquitectura BDI	37
3.2.2.	Arquitecturas Reactivas	39
3.2.3.	Arquitecturas Híbridas	40
3.3.	Arquitecturas Multi-Agente: El Estándar FIPA	41
3.3.1.	Arquitectura FIPA	43
3.3.2.	Ciclo de vida de los Agentes en FIPA	44
3.3.3.	Comunicación entre Agentes en FIPA	46
3.4.	La Plataforma Multi-Agente JADE	49
3.5.	Ontologías en Sistemas Multi-Agente	51
3.5.1.	Ejemplos de uso de ontologías en SMA	52
3.5.2.	Estructura de una ontología en JADE	53

4. Desarrollo del sistema (COMAS)	57
4.1. COMAS (<i>Consensus Multi-Agent based System</i>)	57
4.2. Arquitectura Multi-Agente	58
4.2.1. Agente Moderador	61
4.2.2. Agente Experto	63
4.2.3. Agente Evaluador de Consenso	64
4.2.4. Agente Identificador de Cambios	64
4.3. Comunicación entre Agentes	65
4.3.1. Agente Moderador - Agente Experto	66
4.3.2. Moderador-Evaluador de Consenso	67
4.3.3. Moderador-Identificador de Cambios	68
4.3.4. Proceso General de Comunicación entre Agentes	70
4.4. Ontología en COMAS	72
5. Simulación de Uso de la Plataforma en COMAS	83
5.1. Características del Estudio	84
5.1.1. Descripción del Problema	84
5.1.2. Medidas de Consenso	86
5.1.3. Medidas de Similitud	88
5.1.3.1. Similitud basada en Distancia Euclídea	89

5.1.3.2.	Similitud Cuadrática	89
5.1.3.3.	Similitud Cuadrática Mejorada	90
5.1.4.	Operadores de Agregación	91
5.1.4.1.	Media Aritmética	93
5.1.4.2.	Media Ponderada	93
5.1.4.3.	Media Geométrica	95
5.1.4.4.	Media Armónica	96
5.1.4.5.	Operadores OWA (Ordered Weighted Averaging)	96
5.1.4.6.	Operador OR (Máximo)	98
5.1.4.7.	Operador AND (Mínimo)	99
5.1.4.8.	Operador MAX-MIN	99
5.1.4.9.	Operador de Hurwicz	100
5.1.4.10.	Operador de Laplace	101
5.1.5.	Pruebas realizadas	102
5.2.	Resultados Obtenidos	105
5.3.	Conclusión	115
6.	Conclusiones y Trabajos Futuros	119
	Bibliografía	123

Índice de figuras

2.1. Fases del proceso de selección en problemas de TDG	9
2.2. Esquema general del proceso de consenso para TDG	14
2.3. Modelo de consenso propuesto por Saint y Lawson	19
2.4. Modelo de consenso propuesto por Zadrozny	21
2.5. Modelo de consenso propuesto por Bryson	23
2.6. Representación de los rangos de utilidad	25
2.7. Modelo de consenso con diferentes estructuras de preferencia	26
2.8. Modelo de consenso de COMAS	28
3.1. Esquema básico de la arquitectura de subsunción	40
3.2. Diferentes esquemas de arquitectura híbrida	41
3.3. Modelo de referencia del estándar FIPA	43
3.4. Esquema del ciclo de vida de un agente en FIPA	46
3.5. Protocolo de Interacción FIPA Contract Net	47

3.6. Protocolo de Interacción FIPA Request	48
3.7. Arquitectura de JADE	50
3.8. Modelo de contenido para ontologías en JADE	54
4.1. Arquitectura de COMAS	61
4.2. Comunicación entre agente moderador y agente experto	67
4.3. Comunicación entre agente moderador y agente evaluador de consenso	69
4.4. Comunicación entre agente moderador y agente identificador de cambios	70
4.5. Proceso general de comunicación entre agentes	71
4.6. Ontología de COMAS	73
5.1. Similitud cuadrática	90
5.2. Similitud cuadrática mejorada	91
5.3. Resultados obtenidos con similitud Euclídea	114
5.4. Resultados obtenidos con similitud Cuadrática	114
5.5. Resultados obtenidos con similitud Cuadrática mejorada	115

Introducción

La presente documentación constituye la memoria del trabajo de investigación titulado *Desarrollo de un Sistema Multi-Agente para automatizar Procesos de Consenso en Problemas de Toma de Decisión en Grupo* y que sirve como Trabajo Tutelado de Iniciación a la Investigación en la etapa formativa del *Programa de Doctorado en Ingeniería y Arquitectura*, impartido en la Universidad de Jaén. La memoria comienza con un primer capítulo en el que exponemos la motivación del trabajo de investigación elegido, haciendo un breve repaso general al estado del arte en Toma de Decisiones y Sistemas Multiagente, para seguidamente fijar los objetivos de dicho trabajo y definir la estructura que se seguirá en el resto de la memoria.

1.1. Motivación

La Toma de Decisiones es una de las actividades cotidianas de los seres humanos, ya que constantemente nos enfrentamos a situaciones en las que existen varias alternativas, y en ocasiones debemos decidir cuál de ellas es mejor o cuál se adapta mejor a nuestras necesidades.

El proceso de Toma de Decisión es un proceso complejo, ya que es necesario realizar previamente un análisis detallado de las ventajas e inconvenientes asociados a cada alternativa.

Según la *Teoría Clásica de la Decisión* [62, 74], existen diferentes tipos de problemas que se clasificarán atendiendo a distintos puntos de vista: criterios, expertos y contexto de definición. Según el número de expertos que participen en un problema de Decisión, podemos clasificar dichos problemas en: *Toma de Decisión Individual* y *Toma de Decisión en Grupo*.

Un problema de Toma de Decisión en Grupo (TDG) puede definirse como un problema de decisión en el que dos o más decisores o expertos intentan lograr una solución común para dicho problema, teniendo en cuenta las opiniones o preferencias de todos ellos. Tradicionalmente, los problemas de TDG se han resuelto llevando a cabo un proceso de selección en el que la mejor alternativa es elegida, sin tener en cuenta un acuerdo previo entre los expertos [40]. Así, este proceso de selección de alternativas puede dar lugar en ocasiones a soluciones que no sean aceptadas como buenas por parte de algunos expertos [77], debido a que puedan considerar que sus preferencias individuales no se han tenido en cuenta para obtener dicha solución.

Para evitar tales situaciones, es aconsejable llevar a cabo un proceso de consenso, en el que los expertos discuten y modifican sus preferencias de cara a alcanzar un nivel de acuerdo suficiente antes de tomar una decisión. El estudio de los procesos de consenso se ha convertido en un campo de investigación de gran importancia dentro de la Toma de Decisiones.

Debido a su importancia en TDG, se han propuesto en la literatura varios enfoques de *modelos de consenso* [41, 42, 43, 45, 46, 50, 51, 55, 67]. El *consenso* se ha definido clásicamente como el acuerdo total y unánime de todos los expertos participantes en el problema, algo que resulta inusual en problemas de TDG reales. Por esta razón, las medidas de consenso absolutas seguidas en el enfoque clásico fueron evolucionando hacia medidas más flexibles [77], donde el objetivo es alcanzar un estado de acuerdo mutuo en el grupo de expertos, habiendo expresado cada uno de ellos sus preferencias y habiendo sido tenidas en cuenta, de modo que todos los expertos acepten la solución del problema. Esta interpretación flexible de consenso, ha sido tomada en cuenta en la propuesta basada en mayoría difusa denominada *soft consensus* [49, 52, 53], y que será básica en el presente trabajo.

El consenso es un proceso dinámico e iterativo, consistente en una serie de rondas, donde los expertos expresan y discuten sus preferencias acerca de las alternativas del problema. Este proceso es tradicionalmente supervisado y coordinado por un *Moderador* que ayuda a los expertos participantes en el problema a acercar sus opiniones [3, 56]. Se han propuesto algunos modelos basados en Arquitecturas Software clásicas para automatizar el proceso de alcance de consenso [46, 67].

Los procesos de consenso presentan actualmente algunas debilidades y problemas adicionales, como son el consumo excesivo de tiempo, el manejo de incertidumbre y el coste derivado de la necesidad de reuniones y dedicación por parte de los expertos, entre otros. Para superar algunas de estas dificultades, una de las principales líneas de actuación y mejora que se han propuesto sobre los modelos de consenso es la de ampliar el grado de automatización de los mismos, de manera que la necesidad de gestión humana del problema sea mínima o incluso nula. Una opción para conseguir esta automatización es mediante el paradigma de Agentes Software y Sistemas Multi-Agente [66], que supondrá el modelo que se propone seguir en este trabajo de investigación.

Un Agente Software es una entidad capaz de percibir su entorno, procesar tales percepciones y responder o actuar en su entorno de manera racional y correcta, tendiendo a maximizar un resultado esperado. Una de las características más atractivas de los Agentes Software es su carácter autónomo, es decir, la capacidad para realizar las tareas que le permitan alcanzar un objetivo sin supervisión humana. Un Sistema Multi-Agente es un sistema distribuido compuesto por Agentes Software, donde la acción combinada de cada uno de ellos persigue la consecución de un objetivo común, o bien cada uno de ellos persigue su propio objetivo individual.

1.2. Objetivos

El objetivo principal de este trabajo es el desarrollo de un Sistema de Apoyo al Consenso (SAC) para problemas de TDG definidos en contextos con incertidumbre, planteado como un Sistema Multi-Agente cooperativo en el que actúen diferentes tipos de agentes que asumen distintos roles, entre otros, los de experto y moderador, habituales en este tipo de problema, y que intenten alcanzar el máximo nivel de acuerdo con un cierto grado de autonomía. Gracias al carácter autónomo de los Agentes Software, se consigue que estos actúen sin la supervisión continua y directa de los expertos humanos, evitando así que estos tengan que estar presentes en todas las sesiones de consenso. Este sistema nos permitirá también simular procesos de consenso para optimizar parámetros de los distintos modelos de consenso que apliquemos para resolver los problemas de TDG.

Para conseguir este propósito global, nos planteamos alcanzar los siguientes objetivos:

- **Revisión y Análisis de los modelos de Consenso más conocidos.** En la literatura podemos encontrar diferentes modelos teóricos de consenso capaces de llevar a cabo procesos de TDG bajo incertidumbre, esto es, con información vaga o imprecisa. En ellos se describen tanto las estructuras y dominios empleados para la expresión de preferencias de los expertos como los operadores utilizados para trabajar con diferentes tipos de información.
 - **Revisión del Estado de Arte en el área de los Sistemas Multi-Agente.** Dado que el sistema que pretendemos desarrollar se apoyará sobre una tecnología Multi-Agente, será necesario realizar un estudio previo de las arquitecturas y modelos de Sistemas Multi-Agente más comunes en la literatura, de cara a abordar los aspectos relativos a modelos de Consenso previamente revisados desde una perspectiva del diseño y funcionamiento de los Agentes Software.
 - **Diseño de un Sistema Multi-Agente para la búsqueda del consenso.** Para
-

diseñar un Sistema Multi-Agente es necesario llevar a cabo un análisis exhaustivo del contexto en el que se implantará, con el propósito de identificar los diferentes agentes que participarán, lo que supone además dotarlos de una personalidad que dependerá de las funciones que queramos que lleven a cabo dentro del proceso de Consenso.

- **Diseño de una Ontología para facilitar la comunicación entre Agentes.** Dado que una de las principales características de los Sistemas Multi-Agente es la capacidad de interacción y comunicación con otros agentes, es importante definir una ontología donde se especifique claramente el vocabulario y semántica de los términos utilizados en dicha comunicación.
- **Implementación del Sistema Multi-Agente para dar soporte a procesos de Consenso.** La fase de implementación se llevará a cabo mediante la plataforma de desarrollo de Sistemas Multi-Agente JADE¹.
- **Caso de estudio:** Se realizará una simulación del uso de la plataforma para la resolución de diferentes problemas de TDG bajo incertidumbre.

1.3. Estructura

Para alcanzar los objetivos que perseguimos, esta memoria se estructura en los siguientes capítulos:

- **Capítulo 2:** Presenta una revisión de los modelos de Consenso que consideramos a lo largo de esta memoria haciendo especial hincapié en aquellos basados en el concepto de *Soft Consensus*. Para ello, en primer lugar se hace un breve repaso a los problemas de TDG, para a continuación centrarnos en los problemas de Consenso y los conceptos relacionados con los mismos.

¹<http://jade.tilab.com>

- Capítulo 3: En este capítulo se realiza un estudio general de los Sistemas Multi-Agente, abordando los conceptos básicos, arquitecturas, tecnologías, estándares de desarrollo y protocolos de comunicación más utilizados, además de presentar una visión general del empleo de Ontologías para la comunicación entre agentes.
 - Capítulo 4: Presenta nuestra propuesta de Sistema de Apoyo al Consenso Basado en un Sistema Multi-Agente (*COMAS*), indicando los principales detalles relativos al desarrollo del mismo, así como el diseño de la Ontología empleada.
 - Capítulo 5: En este capítulo, se realiza una simulación para llevar a cabo procesos de consenso en problemas de TDG mediante el uso de COMAS, para así analizar y estudiar el rendimiento del sistema bajo distintos parámetros de consenso utilizados en el mismo.
 - Capítulo 6: Este capítulo concluye la memoria de investigación, presentando las conclusiones más relevantes de la investigación realizada, e indicando las líneas de actuación futuras a llevar a cabo.
-

Modelos de Consenso en TDG

Este capítulo realiza un estudio y visión general del consenso dentro del ámbito de la Toma de Decisión en Grupo (TDG). Para ello, comenzamos sentando las bases y conceptos básicos de los problemas de TDG y, más concretamente, aquellos basados en la idea de consenso. A continuación se hace una revisión de algunos de los modelos de consenso más relevantes presentes en la literatura, indicando los aspectos fundamentales de cada uno de ellos y, finalmente, fijaremos el modelo de consenso a utilizar en este trabajo de investigación.

2.1. Problemas de Toma de Decisión en Grupo. Conceptos Básicos

Tomar una decisión consiste en elegir la mejor opción de entre un conjunto de alternativas posibles [9, 49, 74, 82]. A menudo nos enfrentamos a situaciones en las que debemos decidir qué alternativa tomar en función del entorno en el que nos encontramos.

Los problemas clásicos de decisión presentan los siguientes elementos básicos:

1. Uno o varios objetivos por resolver.
2. Un conjunto de alternativas o decisiones posibles para alcanzar dichos objetivos.

3. Un conjunto de factores o estados de la naturaleza que definen el contexto en el que se plantea el problema de decisión.
4. Un conjunto de valores de utilidad o consecuencias asociados a los pares formados por cada alternativa y estado de la naturaleza.

Dado que en la vida real se puede presentar una enorme variedad de problemas de decisión, la Teoría de la Decisión ha establecido una serie de criterios para clasificarlos bajo diferentes puntos de vista:

1. Según el número de criterios o atributos que se han de valorar para cada alternativa. Tenemos problemas de toma de decisión de un solo criterio y multi-criterio [15, 82].
2. Según el ambiente de decisión en el que se han de tomar las decisiones. Se definen problemas de toma de decisión en ambiente de certidumbre, riesgo e incertidumbre [62].
3. Según el número de expertos. Tenemos problemas de toma de decisión individual y toma de decisión en grupo (TDG) [57, 60].

En este trabajo de investigación nos centramos en problemas de decisión bajo incertidumbre donde participan varios expertos, más concretamente, en *problemas de TDG*, cuyas características exponemos a continuación.

Tomar decisiones en grupo implica la participación de varios decisores que han de tomar decisiones de forma colectiva, de cara a alcanzar una solución común a un problema. Un proceso de toma de decisión en el que participen varios individuos o expertos, donde cada uno de ellos aporta sus propios conocimientos y experiencia, dará como resultado, en ciertos ambientes, una decisión de mayor calidad que aquella aportada por un único experto.

La solución a un problema de TDG se puede obtener aplicando un *enfoque directo* o bien un *enfoque indirecto* [39, 40]. En un enfoque directo, la solución se obtiene a partir de las



Figura 2.1: Fases del proceso de selección en problemas de TDG

preferencias individuales de los expertos (sin obtener una opinión social o general antes de la resolución del problema), mientras que en un enfoque indirecto dicha solución se consigue determinando *a priori* una opinión social, y empleando dicha opinión para la obtención de la solución. Tal y como se observa en la Figura 2.1, en ambos enfoques el proceso general para alcanzar una solución al problema de TDG se compone de dos fases [76]:

- (1) *Fase de Agregación*: Se combinan las preferencias de los expertos.
- (2) *Fase de Explotación*: Consiste en obtener una alternativa o un subconjunto de alternativas que den solución al problema de decisión.

Formalmente, un problema de TDG se caracteriza por:

- La existencia de un problema o cuestión común a resolver.
- Un conjunto de posibles alternativas entre las que escoger.

$$X = \{x_1, x_2, \dots, x_n\} (n \geq 2) \quad (2.1)$$

- Un conjunto de individuos (expertos) que expresan sus juicios, opiniones o preferencias sobre el conjunto de alternativas y que tienen la intención de alcanzar una solución en

común al problema planteado.

$$E = \{e_1, e_2, \dots, e_m\} (m \geq 2) \quad (2.2)$$

Cada experto debe utilizar una estructura de preferencia para representar su opinión sobre un conjunto de alternativas. Una de las estructuras más habituales en problemas de TDG es la *relación de preferencia* [44], que será la que utilizaremos en nuestro modelo de consenso. Una relación de preferencia P_i asociada al experto e_i es una matriz cuadrada de dimensión n :

$$P_i = \begin{pmatrix} p_i^{11} & \dots & p_i^{1n} \\ \vdots & \ddots & \vdots \\ p_i^{n1} & \dots & p_i^{nn} \end{pmatrix}$$

donde cada elemento p_i^{lk} representa la preferencia de la alternativa x_l sobre la alternativa x_k del experto e_i . Los elementos situados en la diagonal de esta matriz carecen de significado y se representarán mediante un guión “-”, ya que no es adecuado hablar de la preferencia de una alternativa sobre sí misma.

Dependiendo del problema al que nos enfrentemos, existen situaciones en las que todos los individuos participan para tomar la decisión, y otras en las que el proceso de decisión sólo debe concernir a un decisor o un pequeño grupo de estos, de ahí que puedan surgir inconvenientes cuando un grupo de individuos participa en un proceso de decisión. Así, podemos encontrarnos en situaciones de colaboración entre expertos, de competitividad entre expertos, propuestas compatibles e incompatibles con uno o más expertos, e incluso propuestas que involucren a diferentes entornos (por ejemplo, entre compañías, gobierno, etc.). Por esta razón, existen diferentes criterios que ayudan a resolver problemas de toma de decisión en grupo, basados en diferentes reglas para obtener la solución [16]:

- *Regla de la Mayoría*: Se toma la decisión teniendo en cuenta la opinión de la mayoría de individuos que componen el grupo envuelto en el problema de decisión. Una vez adoptada la decisión de la mayoría, ésta debe ser respetada por las minorías del grupo, por lo que éstas no deben oponerse a la misma, ya que se asume que todos aceptan el
-

uso de la regla. La noción de mayoría admite dos grandes modalidades de aplicación de la regla:

- 1) *Mayoría absoluta*, cuando la opinión mayoritaria ha sido tenida en cuenta por más de la mitad del total de expertos.
 - 2) *Mayoría relativa o simple*, cuando solamente se requiere que la opinión mayoritaria haya sido la más numerosa en cuanto a expertos se refiere, aunque la suma del resto de expertos la supere.
- *Regla de la Minoría*: Se delega la toma de la decisión en un subgrupo de personas, ya que el problema requiere un nivel de experiencia que solamente presentan dichas personas. Es necesario que todos los expertos participantes acepten la regla y, por consiguiente, estén de acuerdo con delegar la toma de la decisión al subgrupo acordado.
 - *Individual*: Esta situación se presenta cuando el grupo recurre a un experto para tomar la decisión o cuando existe un líder en el grupo.
 - *Unanimidad*: Todos los miembros deben estar de acuerdo con la decisión tomada.

En la mayoría de estas situaciones se presenta el problema de que algunos expertos consideren que sus opiniones no han sido tenidas en cuenta suficientemente [2]. Además, existen situaciones en las que es necesario un alto nivel de acuerdo entre los expertos participantes. Por esta razón, surge la necesidad de emplear enfoques basados en *consenso*, que añaden una nueva fase al proceso de TDG con el objetivo de alcanzar un acuerdo global entre todos los expertos antes de tomar la decisión. En la siguiente sección abordaremos el concepto de consenso, dando una visión general del mismo.

2.2. El Consenso en Problemas de TDG

Según la Real Academia de la Lengua Española [28], se define el término consenso como *el acuerdo producido por consentimiento mutuo entre todos los miembros de un grupo o entre*

varios grupos. Esta definición supone la idea de un proceso de TDG en el que ningún experto está en desacuerdo sobre las decisiones tomadas, aunque algunos expertos pueden seguir opinando que su solución individual fuera mejor que la finalmente tomada. Para conseguir el acuerdo es necesario, pues, que *todos* los expertos cambien sus opiniones iniciales, tendiendo a aproximarlas hacia una opinión colectiva que consideren satisfactoria.

A pesar de que la fase de consenso se introduce en los procesos de TDG para incrementar el nivel de acuerdo entre los expertos que participan en el problema de decisión, en ocasiones el concepto de consenso causa cierta controversia, ya que puede ser interpretado de distintas formas, desde un total acuerdo (unanimidad) a una interpretación más flexible.

El enfoque tradicional de consenso supone una visión *rígida* del mismo, donde se considera que existe consenso solamente cuando el acuerdo entre los expertos es total y unánime [17]. Este enfoque presenta el inconveniente de que el acuerdo total suele ser difícil o imposible de alcanzar en la práctica. Además, el consenso por unanimidad a veces puede haberse alcanzado mediante intimidación u otras circunstancias en las que el acuerdo alcanzado no es verdadero (*consenso normativo*) [65], razón por la cual se ha mostrado a menudo como un método poco realista para tomar decisiones reales. El consenso normativo implica en la práctica una imposición *a priori* desde el exterior del grupo que los miembros del mismo asumen sin discutir.

El consenso no debería entenderse como un acuerdo unánime, sino como un proceso en el que la decisión finalmente tomada posiblemente no coincida totalmente con las posiciones iniciales de los expertos. Esta concepción del consenso se conoce como *Consenso Cognitivo* [65], e implica que los expertos modifican sus opiniones iniciales tras una serie de rondas de discusión y negociación. En la literatura podemos encontrar diferentes ideas de *consenso* que intentan “suavizar” la versión rígida de consenso como unanimidad. Algunas de ellas son [49, 50, 77]:

- Todos los miembros del grupo apoyan la decisión tomada.
- Ningún miembro del grupo se opone a la decisión tomada.
- Todos los miembros del grupo asumen la decisión tomada, aunque no la apoyen.
- Estado de mutuo acuerdo entre los miembros del grupo, y donde las opiniones cruciales de los individuos han sido consideradas para satisfacción de estos.

Uno de los enfoques más aceptados para “suavizar” la noción rígida de consenso como unanimidad es el de *soft consensus*. Este enfoque, propuesto por Kacprzyk [49, 50, 51, 52], profundiza en el estudio del concepto de mayoría en problemas de TDG, proponiendo suavizarlo mediante otro más flexible, denominado “mayoría difusa”, que introduce el uso de cuantificadores lingüísticos difusos [49, 55], y donde el concepto de *soft consensus* se entiende por: *La mayor parte de los expertos están de acuerdo en las alternativas importantes*. Nótese que la definición de *soft consensus* está basada en la teoría de conjuntos difusos [58]. Este enfoque ha proporcionado resultados satisfactorios en diferentes problemas de TDG [29, 41, 86], y supondrá uno de los enfoques suavizados de medición del consenso considerados en este trabajo de investigación.

En definitiva, el consenso es un área de investigación de gran importancia en el campo de la toma de decisión en grupo, y son muchos los autores que han propuesto modelos que justifican la necesidad de llevar a cabo un proceso de consenso previo a la selección de alternativas en este tipo de problemas [7, 8, 10, 29, 41, 46, 54, 67, 77].

2.2.1. Procesos de Consenso

El consenso es un objetivo en sí, y para alcanzar dicho objetivo es fundamental realizar un proceso que nos conduzca a él. En esta sección estudiaremos en qué consisten los procesos de consenso, indicando las fases y esquema de actuación que caracterizan a dichos procesos. El principal propósito de los procesos de consenso consiste en alcanzar un nivel de acuerdo

mínimo antes de iniciar el proceso de selección de alternativas, mediante discusión de preferencias, durante una o varias rondas [77]. Este proceso suele estar coordinado o dirigido por una figura humana: el *moderador*. El moderador es una figura clave en procesos de consenso, y sus funciones fundamentales son:

- Evaluar el nivel de acuerdo alcanzado en cada ronda de consenso.
- Identificar las alternativas que impiden alcanzar el consenso deseado.
- Informar a los expertos sobre los cambios que estos deben considerar sobre las preferencias en dichas alternativas.

A continuación, pasamos a describir el esquema general de actuación para procesos de consenso que vamos a considerar en este trabajo, en el cuál se considera un enfoque cognitivo de consenso, así como una visión suavizada del mismo. El esquema de este proceso se muestra en la figura 2.2, y las etapas que lo componen son las siguientes:

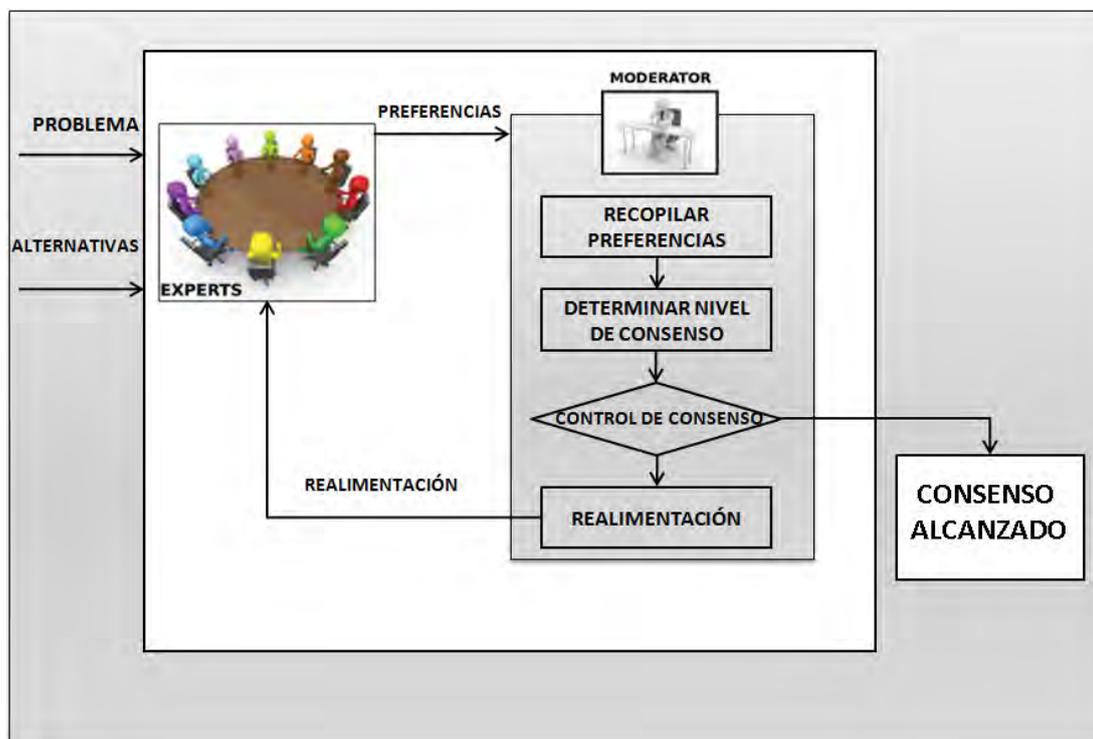


Figura 2.2: Esquema general del proceso de consenso para TDG

- (1) Descripción del problema de toma de decisión.
- (2) Identificación del formato empleado para representar las preferencias, junto con las medidas para calcular el consenso a partir de dichas preferencias [10, 22, 25, 38, 41, 68, 83].
- (3) Cada experto proporciona sus preferencias individuales.
- (4) Comprobación del nivel de consenso actual, de acuerdo a las medidas de consenso elegidas. Diferentes modelos se han propuesto para obtener el grado de consenso empleando diversas medidas y tipos de información [7, 34]. Si el nivel de consenso es suficiente, el proceso finaliza; de lo contrario, continuar con el paso (5).
- (5) Realimentación por parte del moderador a los expertos. Cuando existe una cierta discrepancia entre las preferencias de los expertos, el moderador identifica a los expertos y/o preferencias que impiden alcanzar el nivel de consenso deseado. A continuación, el moderador aconsejará a los expertos que modifiquen dichas preferencias, con el objeto de acercar sus opiniones y mejorar el grado de consenso en la siguiente ronda. Como puede verse, el moderador juega un papel clave en esta etapa, siendo responsable de guiar, controlar y finalizar el proceso de consenso.
- (6) Volver al paso (3). Dado que el número de rondas de consenso debe estar limitado, en los casos en que este número se haya sobrepasado sin llegar a un acuerdo, deben buscarse estrategias alternativas o finalizar el proceso sin éxito [77].

2.2.2. Problemas y Retos Futuros en los Procesos de Consenso

Una vez estudiados los procesos de consenso, podemos pasar a puntualizar las principales debilidades y problemas que con frecuencia han sido encontradas por grupos y organizaciones a la hora de su puesta en práctica [65].

- *Consumo de tiempo excesivo*: A menudo, directivos y expertos se quejan de que los procesos de consenso requieren mucho más tiempo que los procesos de toma de decisión
-

individual o TDG basada en mayoría, ya que la discusión y negociación implican un período de tiempo considerable.

- *Coste*: Las reuniones y necesidad de dedicación por parte de los expertos incrementan el coste del proceso.
- *Incertidumbre*: Dado que el consenso es un concepto inherentemente vago e impreciso, implica incertidumbre, que sin la adecuada experiencia podría suponer un mal uso de la información.
- *Mayoría*: Debido a la complejidad del proceso, existe una continua tentación por parte de los participantes de interrumpirlo y tomar una decisión final basada en mayoría, la cual no necesariamente implica consenso en muchos casos.
- *Conocimiento y modelado de preferencias*: En ocasiones, la representación del conocimiento y modelado de preferencias empleados no son lo suficientemente flexibles para cubrir diversas áreas y tópicos en los que los miembros trabajan.
- *Sin garantías de éxito*: El principal problema existente cuando se toma una decisión en grupo bajo consenso es que dicho consenso puede no llegar a alcanzarse, y algunos participantes pueden considerar el proceso como una pérdida de tiempo.

A continuación, proponemos algunos retos que los procesos de consenso deben abordar para superar las dificultades que hemos expuesto:

- (1) **Automatización**: El elevado coste de los procesos de consenso hace que sea necesario para su reducción un cierto grado de automatización en los mismos. Una forma de conseguirlo es mediante modelos basados en Inteligencia Artificial. Además, se pretende también la automatización de las tareas propias del moderador, expertos u otros roles presentes en algunos modelos de consenso [77], siendo las figuras humanas reemplazadas también por herramientas automáticas basadas en Inteligencia Artificial. Actualmente
-

son conocidos algunos modelos basados en lógica difusa para conseguir esta automatización [46]. En este trabajo se pretende conseguir un cierto grado de automatización en el proceso de consenso, especialmente en las tareas realizadas por el moderador, mediante un Sistema Multi-Agente [66].

- (2) **Compromiso:** La búsqueda de consenso implica aceptación global de la decisión tomada, sin embargo, en la práctica pueden aparecer subgrupos en los que no se considere que sus intereses han sido tenidos en cuenta. Esta situación puede provocar cambios forzados en las opiniones y, en consecuencia, un falso consenso, que puede tener repercusiones negativas en el futuro. Este tipo de comportamientos no tiene actualmente una solución clara, y se suele explicar mediante la Teoría de Juegos [62, 72].

Para evitar esta situación, se han desarrollado herramientas de detección de expertos con opiniones inamovibles durante el proceso, o cambios forzados por terceros, con el objeto de penalizar a estos expertos disminuyendo su importancia o peso en el problema [84]. Por otra parte, la solución más natural consiste en imponer un contrato de compromiso y colaboración que debe ser firmado por los participantes al comienzo del proceso.

- (3) **Manejo de la Incertidumbre:** Los enfoques basados en *soft consensus* suponen la mejor apuesta para trabajar con incertidumbre, habiendo dado muy buenos resultados [54, 56]. No obstante, es importante destacar que, dado que el consenso no debe confundirse con un proceso de TDG basado en mayoría, es necesario redefinir el concepto de mayoría visto en 2.2 para trabajar con estos modelos.

Los procesos futuros de alcance de consenso deben abordar estos y otros retos y proponer posibles soluciones. Si se consigue este objetivo, se logrará una mayor aceptación de la toma de decisión mediante consenso.

2.3. Algunos Modelos de Consenso empleados en Problemas de TDG

En esta sección revisamos brevemente algunos modelos de consenso cognitivo presentes en la literatura [14, 45, 57, 77, 86]. La estructura de estos modelos es similar a la mostrada en 2.2, aunque cada uno de ellos se adapta al contexto específico para el que fue diseñado.

2.3.1. Modelo Teórico del Proceso de Consenso

Este primer modelo propuesto por S. Saint y J.R. Lawson [77] describe en detalle las diferentes fases llevadas a cabo en los procesos de consenso, tal y como se desarrollan en situaciones reales dentro de una empresa u organización. En la Figura 2.3 se muestra una representación gráfica de este modelo, el cual se divide en tres fases principales:

1. *Asimilación de la propuesta.* Se lleva a cabo la presentación y aclaración de las posibles dudas iniciales acerca de la propuesta que se pretende aprobar por consenso. También se consulta si existe algún inconveniente a que la propuesta inicial sea considerada como la solución ideal al problema planteado.
 2. *Resolución de desacuerdos.* Se compone de varias rondas de consenso donde los expertos expresan sus preferencias y discuten para resolver los inconvenientes individuales. Al final de cada ronda se comprueba si se ha alcanzado el acuerdo deseado.
 3. *Cierre del proceso.* Se llega a esta fase si no se ha podido alcanzar un consenso entre los expertos en el período de tiempo inicialmente fijado. Las posibles medidas a tomar en el cierre de la sesión incluyen: retirar la propuesta, prorrogar el tiempo establecido, tomar como decisión final la opinión de la mayoría y/o excluir a los expertos que impiden alcanzar el consenso.
-

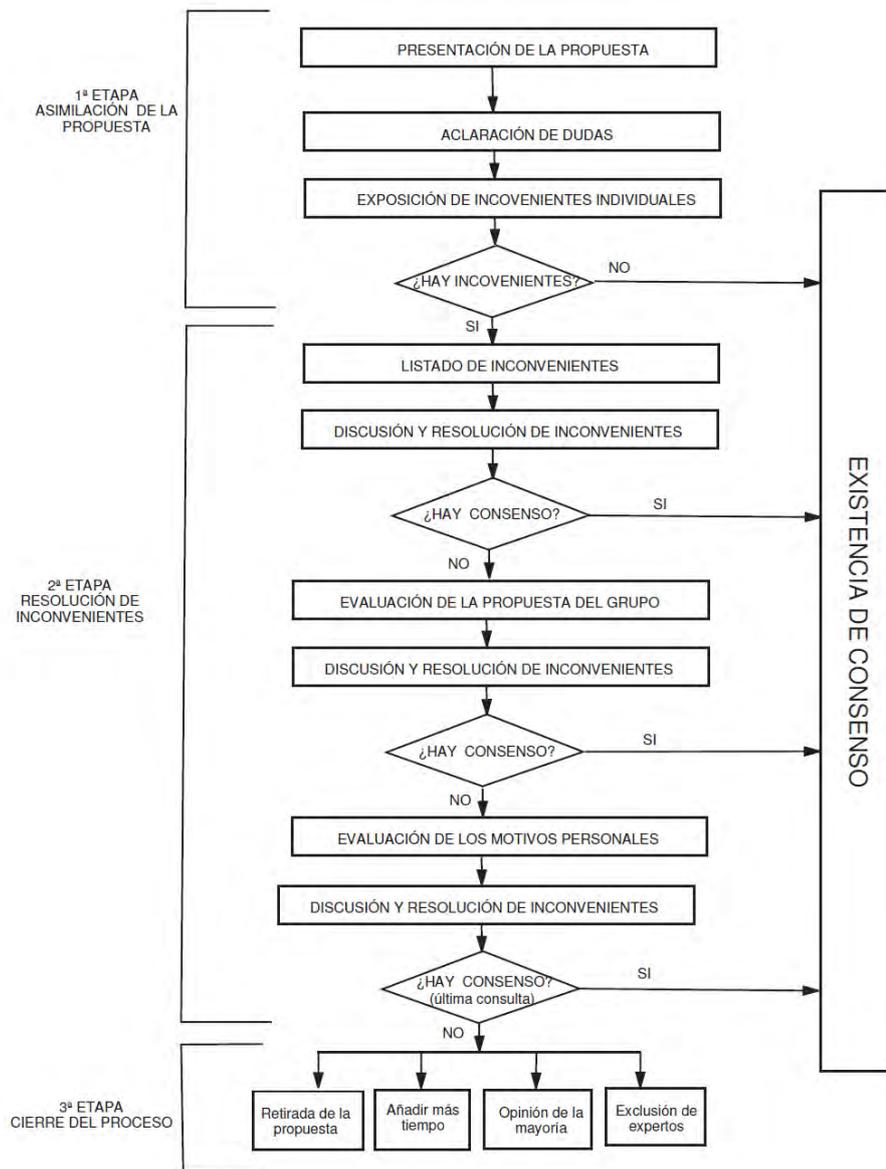


Figura 2.3: Modelo de consenso propuesto por Saint y Lawson

Además, este modelo propone una serie de roles o figuras participantes en el proceso, además del propio moderador.

- a) **Moderador.** Como ya hemos comentado, es el encargado de poner en marcha todo el proceso y de la correcta ejecución del mismo. Es responsable de aconsejar a los expertos participantes para acercar su opinión a la de la mayoría.
- b) **Redactor del acta.** Es el encargado de redactar el acta de cada sesión de consenso, donde se recogen por escrito las conclusiones y/o decisiones alcanzadas. Resulta aconsejable la lectura del acta antes de abandonar la reunión para evitar que algún experto tenga ideas diferentes sobre las decisiones tomadas.
- c) **Controlador del tiempo.** El controlador de tiempo trabaja junto al moderador para asegurar que no se exceda el tiempo asignado a cada una de las fases. Entre sus funciones está la de notificar periódicamente a los participantes el tiempo restante de cada fase.

2.3.2. Modelo de Zadrozny con Información Difusa

Zadrozny et al. [86] propusieron un modelo de consenso en el que por medio de la Lógica Difusa se pretende resolver el inconveniente del manejo de incertidumbre por parte de los expertos, presente en los problemas de decisión. Este modelo, representado en la Figura 2.4, sigue el enfoque de *soft consensus* propuesto por Kacprzyk en [51] y sus principales características son:

- a) Cada experto expresa sus preferencias mediante relaciones de preferencias difusas.
 - b) Llevar a cabo un “análisis de la estructura del grupo de expertos”, en el que se agrupa el conjunto de expertos en diferentes subgrupos en función de la coincidencia en sus preferencias.
 - c) Utiliza el concepto de consenso como mayoría difusa, con el cálculo de proposiciones cualificadas lingüísticamente propuesto por Zadeh [85].
-

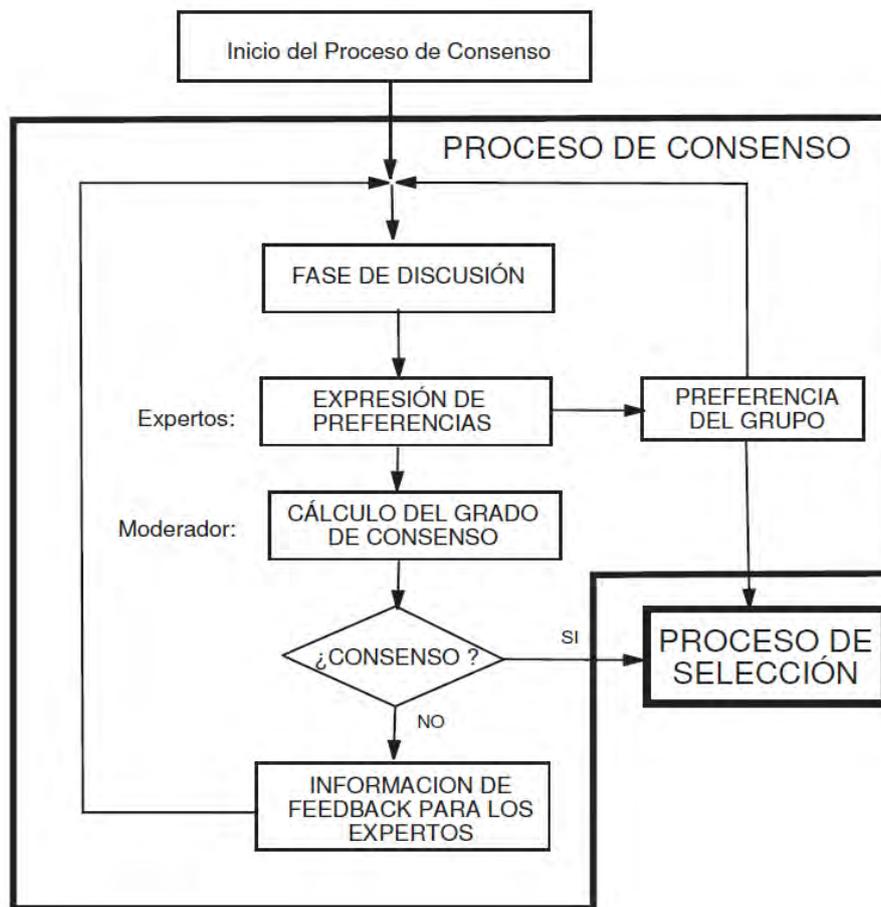


Figura 2.4: Modelo de consenso propuesto por Zadrozny

- d) Además de medir el grado de consenso entre los expertos, introduce nuevos indicadores de consenso. Los indicadores de consenso individuales permiten conocer la situación de las preferencias de cada experto dentro del espacio de preferencias del grupo, e incluyen indicadores de contribución al consenso y grado de consenso personal. Los indicadores de consenso a nivel de grupo proporcionan al moderador información adicional sobre el estado actual del consenso dentro del grupo.

2.3.3. Modelo de Bryson con Vectores de Preferencia

El modelo de consenso propuesto por Bryson en [14] propone las siguientes ideas:

- Presenta un algoritmo detallado con los pasos a seguir para consensuar la opinión del grupo.
- Contempla la opción de ejecutarse de forma cíclica hasta alcanzar el grado de consenso deseado o bien un número máximo de ciclos o rondas previamente fijado.
- Emplea las medidas de similitud para medir la semejanza entre las preferencias de los expertos, para obtener indicadores de consenso a nivel individual y a nivel de grupo.
- También emplea las medidas de similitud para determinar las preferencias y expertos más alejados, para así recomendar las preferencias que estos han de cambiar.

La característica más notable de este modelo es el modo de expresar las preferencias. Dado un conjunto de expertos y un conjunto de alternativas, cada experto e_i expresa su opinión mediante un vector de preferencia,

$$w_i = (w_i^1, \dots, w_i^n) \quad (2.3)$$

que cumple que $\sum_L w_i^l = 1$ y que $w_i^l > 0$ para $l = 1, \dots, n$, siendo n el número de alternativas. Además, la preferencia del experto i sobre la idoneidad de la alternativa l frente a la alternativa k se determina mediante el cociente (w_i^l/w_i^k) . Para obtener w_i cada experto debe definir una matriz o relación de preferencia con la importancia relativa de cada par de preferencias. Para evaluar el grado de consenso entre dos vectores de preferencia (expertos) w_i y w_j , se emplea la siguiente medida de similitud:

$$s(w_i, w_j) = 1 - \text{sen}(w_i, w_j) \quad (2.4)$$

Este modelo presenta además dos umbrales de consenso, α y δ , de forma que si $s(w_i, w_j) \geq \alpha$ existe un gran acuerdo entre los expertos i, j , y si $s(w_i, w_j) \leq \delta$ existe un gran desacuerdo



Figura 2.5: Modelo de consenso propuesto por Bryson

entre estos. El esquema de las fases de las que se compone este modelo es el mostrado en la Figura 2.5. Las principales tareas llevadas a cabo en estas fases son las que citamos a continuación:

1. **Preparación:** Se especifica el máximo número de rondas o ciclos que se llevarán a cabo, MAXCICLOS, los umbrales de consenso α y δ , y se inicializa el contador de ciclos.
2. **Discusión en grupo:** Los expertos se reúnen, discuten y argumentan sus opiniones según sus propios puntos de vista.
3. **Determinación de las preferencias individuales:** Cada experto i expresa sus preferencias.

4. **Cálculo de los indicadores de consenso:** Se calcula una serie de indicadores de consenso [14], tanto a nivel individual como a nivel de grupo de expertos.
5. **Finalización del procedimiento:** Si el indicador de consenso es superior al umbral o se alcanza el número máximo de ciclos, el procedimiento termina, obteniendo el vector de preferencia colectivo $w_{GM} = (w_1, \dots, w_n)$. En caso contrario, los miembros discuten y cambian sus opiniones teniendo en cuenta los valores de los indicadores del consenso, y ayudados por la figura de un moderador.

2.3.4. Modelo de Kim para Problemas de TDG Multicriterio

S.H. Kim et al. han propuesto en [57] el llamado “Procedimiento Interactivo” para problemas de TDG multicriterio con información incompleta. A diferencia de otros modelos vistos en esta memoria, este modelo se basa en representar las preferencias de los expertos mediante intervalos valorados entre 0 y 1 denominados *rangos de utilidad*, cuya amplitud y límites se comparan para determinar el grado de consenso y detectar rápidamente a los expertos que menos contribuyen al consenso. De esta manera se descarta el empleo de medidas de similitud.

En esencia, dada una alternativa del problema x_m y un atributo i , el rango de utilidad del experto k para esta alternativa y atributo se calcula tal y como indica la ecuación (2.5):

$$u_i^k(x_m) = (\text{mín } u_i^k(x_m), \text{máx } u_i^k(x_m)) \quad (2.5)$$

Tal y como se representa en la Figura 2.6, usando los rangos de utilidad individuales se obtiene el rango total del grupo y el rango de consenso del grupo, a partir de la unión e intersección de los rangos de utilidad individuales, respectivamente.

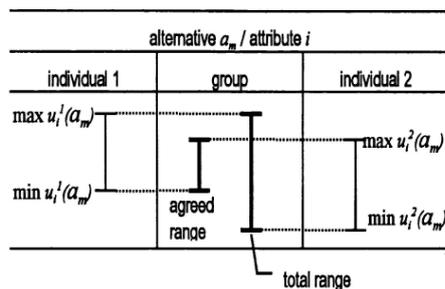


Figura 2.6: Representación de los rangos de utilidad

2.3.5. Modelo de Herrera-Viedma con diferentes Estructuras de Preferencia

Este modelo, propuesto por E. Herrera-Viedma et al. [45], se basa en una visión suavizada de consenso, y se caracteriza por el uso de diferentes estructuras de preferencia por parte de los expertos. El proceso seguido en este modelo se muestra en la figura 2.7.

La principal característica de este modelo, consiste en que cada experto $e_i \in E$ expresa sus opiniones sobre un conjunto de alternativas X mediante una de las cuatro posibles estructuras de preferencia que describimos a continuación:

1. *Orden de preferencia*: El experto proporciona un orden de las preferencias $O_i = \{o_i(1), \dots, o_i(n)\}$, donde $o_i(j) \in \{1, \dots, n\}$ es el índice de una de las alternativas existentes. Esta estructura consiste, por tanto, es un vector de alternativas ordenadas de mejor a peor, según la opinión individual del experto.
2. *Relación de preferencia difusa*: Las opiniones se representan con relaciones de preferencia P_i , donde cada $p_i^{lk} \in [0, 1]$ denota el grado de preferencia de x_l sobre x_k .
3. *Relación de preferencia multiplicativa*: Se trata de relaciones de preferencia donde cada valor p_i^{lk} se interpreta como: “la alternativa x_l es p_i^{lk} veces tan buena como lo es x_k ”. Cada valoración se realiza dentro de una escala numérica, siendo común emplear una escala de 1 a 9, donde un valor de 1 indica indiferencia entre alternativas y un valor de

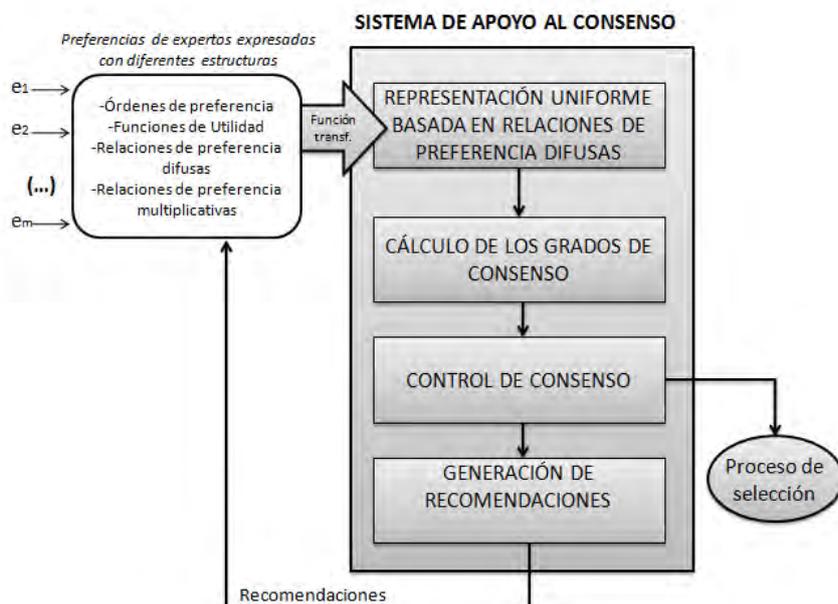


Figura 2.7: Modelo de consenso con diferentes estructuras de preferencia

9 indica total preferencia de x_l sobre x_k . Se asumen además valoraciones recíprocas, tales que $p_i^{lk} \cdot p_i^{kl} = 1$.

4. *Función de utilidad*: El experto expresa sus preferencias mediante un vector de valoraciones numéricas dentro del intervalo unitario, tal y como se muestra en el modelo visto en la sección 2.3.3.

Otras características importantes de este modelo son:

- a) Representación uniforme de las preferencias: Para poder trabajar con diferentes estructuras de preferencia de forma conjunta, se aplica sobre ellas una función de transformación. Herrera-Viedma et al., proponen en [45] el uso de *relaciones de preferencia difusas* como la estructura para representar las opiniones de manera uniforme, ya que han demostrado ser de gran utilidad en problemas de TDG en la práctica, especialmente para la obtención de la opinión colectiva mediante agregación.
- b) El modelo se basa en dos criterios de consenso: *medidas de consenso* flexibles y *medidas*

de proximidad entre las opiniones de cada experto y la opinión colectiva.

- c) Un mecanismo de realimentación, basado en sugerencias de cambio sobre las opiniones, permite reemplazar al moderador humano en esta tarea.

2.4. Modelo de Consenso propuesto

En esta sección vamos a presentar un esquema detallado del modelo de sistema de apoyo al consenso para problemas de TDG que se utilizará en la propuesta de este trabajo de investigación.

Dado que en este trabajo de investigación nos planteamos desarrollar un *sistema automatizado* de apoyo al consenso (abordando así uno de los retos futuros presentes en los procesos de consenso), es fundamental adoptar un modelo de consenso que nos permita conseguir cierto grado de automatización. Muchos de los modelos anteriormente descritos y, en general, presentes en la literatura, no están diseñados para aplicar una automatización directa sobre los mismos. El modelo de consenso que hemos considerado, el cual se basa en ideas expuestas en [45, 46, 86], sí nos permite un cierto grado de automatización, especialmente en los procesos llevados a cabo por el moderador.

Para estudiar las fases de las que se compone nuestro modelo de consenso, es necesario conocer en primer lugar la definición y contexto en el que se situarán los problemas de TDG a abordar. Formalmente, el problema de TDG estará definido de esta forma: dado un conjunto de expertos, cada e_i debe expresar sus preferencias sobre un conjunto de alternativas mediante una relación de preferencia P_{e_i} , donde $p_i^{lk} \in [0, 1]$.

A continuación, pasamos a describir en detalle las etapas seguidas en nuestro modelo, representado en la Figura 2.8.

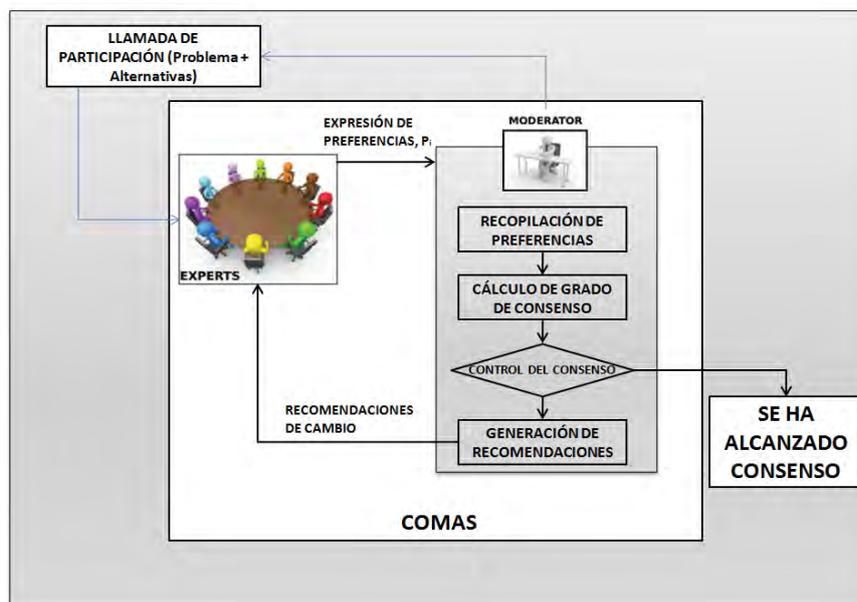


Figura 2.8: Modelo de consenso de COMAS

Llamada de participación en el Proceso de Consenso

En esta primera etapa, el moderador informa al conjunto E de expertos sobre el problema, proporcionándoles el conjunto X de alternativas que componen el problema e invitándoles a participar en el mismo. Los expertos deciden si participan o no en el problema. Una vez transcurrido un período de tiempo previamente fijado por el moderador, dará comienzo el problema, (siempre que al menos hayan aceptado participar 2 expertos en él). Antes de comenzar el proceso de consenso, el moderador fija los parámetros del problema, tales como el umbral de consenso μ y el número máximo de rondas permitido.

Expresión de Preferencias de los Expertos

Los expertos proporcionan al moderador sus opiniones sobre las alternativas por medio de relaciones de preferencia difusas, cuyos elementos notamos como p_i^{lk} , siendo valoraciones numéricas cuyo valor está comprendido entre 0 y 1. Así, $p_i^{lk} = 1$ implica que el experto i tiene absoluta preferencia de la alternativa l sobre la alternativa k , y $p_i^{lk} = 0$ implica que dicho experto muestra un total rechazo de la alternativa l frente a la alternativa k .

Es deseable que las valoraciones cumplan la propiedad de reciprocidad, de manera que si $p_i^{lk} = x$ ($x \in [0, 1]$), entonces $p_i^{kl} = 1 - x$. De esta forma se mantendrá la consistencia entre valoraciones sobre pares de alternativas recíprocos. Además, no tiene sentido valorar p_i^{ll} , por lo que los elementos en la diagonal de la matriz de preferencia resultarán irrelevantes durante el problema.

Cálculo del Grado de Consenso

Una vez recibidas las preferencias de los expertos, el moderador procede al cálculo del nivel de acuerdo entre estos. Para ello, se realizan los siguientes pasos:

- En primer lugar, para cada par de expertos e_i, e_j ($i < j$), se calcula una matriz de similitud $SM_{ij} = (sm_{ij}^{lk})$ utilizando una medida de similitud $s(p_i^{lk}, p_j^{lk})$ para calcular cada elemento de dicha matriz,

$$sm_{ij}^{lk} = s(p_i^{lk}, p_j^{lk}) \quad (2.6)$$

donde sm_{ij}^{lk} es la similitud entre los expertos e_i y e_j sobre el par de alternativas (x_l, x_k) .

- Se calcula una matriz de consenso, $CM = (cm^{lk})$, agregando a nivel de pares todas las matrices de similitud previamente calculadas. Cada elemento de esta matriz se calcula como,

$$cm^{lk} = \phi(sm_{12}^{lk}, sm_{13}^{lk}, \dots, sm_{1m}^{lk}, sm_{23}^{lk}, \dots, sm_{2m}^{lk}, \dots, sm_{(m-1)m}^{lk}) \quad (2.7)$$

donde ϕ es el operador de agregación empleado, y $l, k \in \{1, \dots, n\}$.

Como veremos en el capítulo 5, en el que revisaremos una serie de operadores de agregación diferentes utilizados en nuestro sistema, se hace posible el cálculo del grado de consenso según un enfoque de consenso suavizado. En particular, algunos de estos operadores nos permiten aplicar el enfoque de *soft consensus* durante la agregación de valores para el cálculo del grado de consenso, el cual mostramos a continuación.

- Por último, se procede a calcular el nivel de acuerdo. Esto se hace a tres niveles:

1. Consenso a nivel de pares de alternativas. Se obtiene como,

$$cp^{lk} = cm^{lk}, \forall l, k = 1, \dots, n \wedge l \neq k, \quad (2.8)$$

donde cp^{lk} representa el acuerdo alcanzado sobre el par de alternativas (x_l, x_k) , obtenido directamente a partir de la matriz de consenso CM .

2. Consenso a nivel de alternativas,

$$ca^l = \phi(cp^{l1}, \dots, cp^{l(l-1)}, cp^{l(l+1)}, \dots, cp^{ln}), \quad (2.9)$$

donde ca^l representa el acuerdo sobre la alternativa x_l .

3. Consenso a nivel de relación de preferencia,

$$cr = \phi(ca^1, \dots, ca^n), \quad (2.10)$$

donde cr representa el grado de consenso global alcanzado entre los expertos en la ronda actual.

La obtención de los tres niveles de grado de consenso se puede realizar, como hemos comentado, siguiendo la noción de *soft consensus* de Kacprzyk si se utiliza el operador de agregación adecuado. La posterior etapa dedicada a la generación de recomendaciones, incluye también algunas operaciones de agregación, en las cuales conviene destacar que no consideramos dicha noción, al no encontrarnos en la etapa de medición del grado de consenso.

Control del Consenso

Se comprueba el nivel de consenso cr obtenido en la anterior fase. Si cr es mayor o igual que el umbral de consenso inicialmente fijado, μ , se entiende que se ha alcanzado el nivel de acuerdo deseado y el proceso finaliza; de lo contrario, el proceso requiere mayor discusión y debe continuar, a menos que se haya alcanzado el número máximo de rondas permitido, en cuyo caso el proceso finaliza sin haber llegado a un acuerdo.

Generación de Recomendaciones

Esta fase se da si en la etapa de control de consenso el grado obtenido era menor que el umbral μ , y en ella se sugiere a los expertos la manera en que deberían modificar sus preferencias de cara a incrementar el nivel de acuerdo en rondas siguientes. Para obtener estas *recomendaciones de cambio*, se llevan a cabo tres tareas:

- *Calcular la preferencia colectiva y las matrices de proximidad de los expertos.* La preferencia colectiva $\mathbf{P}_{ec} = (p_c^{lk})$ se calcula agregando todas las preferencias de los expertos $\{\mathbf{P}_{e_1}, \dots, \mathbf{P}_{e_m}\}$ a nivel de pares de alternativas:

$$p_c^{lk} = \phi(p_1^{lk}, \dots, p_m^{lk}) \quad (2.11)$$

A continuación, se determinan las matrices de proximidad, que indican cuán próximas están las preferencias de cada experto a la preferencia colectiva. Cada experto e_i tiene una matriz de proximidad, y cada elemento de dicha matriz se calcula midiendo la similitud entre la preferencia de cada experto y la preferencia colectiva,

$$pp_i^{lk} = s(p_i^{lk}, p_c^{lk}) \quad (2.12)$$

Con estos valores de proximidad, es posible identificar las preferencias más alejadas de la preferencia colectiva, y por consiguiente, los expertos que deberían cambiar sus opiniones.

- *Identificación de las preferencias que deben ser cambiadas.* Existen diferentes criterios para elegir las preferencias que deben modificarse, considerando tanto el grado de consenso como los valores de proximidad [18, 67]. Nuestro modelo propone hacer los cambios en aquellos pares de alternativas cuyo grado de consenso a nivel de alternativas, ca^l no sea suficiente, obteniendo el conjunto de cambios, CC ,

$$CC = \{(x_l, x_k) | ca^l < cr \wedge cp^{lk} < cr\} \forall l, k \in \{1, \dots, n\} \quad (2.13)$$

Una vez identificados los pares a cambiar, nuestro modelo debe identificar a los expertos que deberían hacer cambios en cada uno de estos pares. De este modo, los expertos

cuya preferencia esté más alejada de la preferencia colectiva para el par $(x_l, x_k) \in CC$, deberán de modificar su valoración a dicho par. Para poder identificar a estos expertos, definimos un umbral de proximidad o *proximidad media* \overline{pp}^{lk} para cada par, que se calcula agregando todos los valores o matrices de proximidad de los expertos a nivel de pares:

$$\overline{pp}^{lk} = \phi(pp_1^{lk}, \dots, pp_m^{lk}) \quad (2.14)$$

Nuestro modelo aconsejará cambiar el par $(x_l, x_k) \in CC$ a todo experto e_i cuyo $pp_i^{lk} < \overline{pp}^{lk}$.

- *Establecer las direcciones de cambio.* El modelo utiliza una serie de reglas de dirección para sugerir la dirección correcta en las recomendaciones de cambio, y mejorar el acuerdo en siguientes rondas. Para cada recomendación de cambio $((x_l, x_k), e_i)$ asociada a un experto y par de alternativas determinados, se sugerirá incrementar o decrementar la valoración del experto sobre dicho par. Para ello se tendrá en cuenta además la preferencia colectiva \mathbf{P}_{e_c} . La forma de realizar los cambios consiste en considerar la recomendación, y aumentar o disminuir el valor anterior, siempre dentro del intervalo $[0,1]$.
 - DIR.1: Si $(p_i^{lk} - p_c^{lk}) < 0$, entonces se recomienda al experto e_i incrementar la valoración asociada al par (x_l, x_k) .
 - DIR.2: Si $(p_i^{lk} - p_c^{lk}) > 0$, entonces se recomienda al experto e_i decrementar la valoración asociada al par (x_l, x_k) .
 - DIR.3: Si $(p_i^{lk} - p_c^{lk}) = 0$, entonces el experto e_i no tiene que modificar la valoración asociada al par (x_l, x_k) .

Una vez obtenidas todas las recomendaciones de cambio, el moderador se las proporciona a los expertos. La siguiente ronda comienza con una nueva expresión de preferencias por parte de los expertos, basándose en las recomendaciones de cambio recibidas.

Sistemas Multi-Agente

Este capítulo de la memoria de investigación consiste en una introducción y estudio de los Sistemas Multi-Agente. A lo largo de las siguientes secciones, mostraremos los principales conceptos relativos a Agentes Software y daremos una clasificación de Agentes bajo diferentes puntos de vista, veremos en qué consisten los Sistemas Multi-Agente (SMA) y las arquitecturas existentes para su desarrollo, haciendo hincapié en el estándar FIPA. Por último, presentaremos la plataforma Multi-Agente JADE utilizada en este trabajo e introduciremos el empleo de Ontologías en SMA.

3.1. Agentes Software

No existe una definición precisa para el término agente pero, debido a que estamos situados en un contexto de Ingeniería, es posible distinguir lo que es un agente de lo que no lo es. Así, podemos decir que el concepto de “agente” caracteriza a una entidad software con una arquitectura robusta y adaptable capaz de funcionar en diversos entornos o plataformas y capaz de cumplir objetivos de forma autónoma e “inteligente”, intercambiando información con el entorno o con otros agentes humanos o software [36].

Las principales características que definen el comportamiento de un agente son [66]:

- Funcionamiento continuo y autónomo.
- Comunicación en el entorno y con otros agentes, a través de un lenguaje o formalismo de comunicación.
- Robustez.
- Adaptabilidad, entendida como la capacidad de cumplir objetivos y tareas en diferentes contextos de forma flexible.

Además, algunos agentes cumplen también las siguientes características:

- Razonamiento y Aprendizaje, requisito indispensable en agentes inteligentes.
- Movilidad: Un agente móvil es aquel capaz de desplazarse entre nodos de una red y ejecutarse en distintas plataformas.

A partir de las principales propiedades de los agentes, podemos dar una primera clasificación de los mismos en agentes reactivos, proactivos y sociales.

- **Agente reactivo:** Se caracteriza por ser capaz de interactuar con el entorno que le rodea de forma dinámica, y de responder ante eventos no esperados (comportamiento no determinista).
 - **Agente proactivo:** Su comportamiento va un paso más allá respecto al del agente reactivo. Un agente proactivo genera y trata de alcanzar sus propios objetivos, es capaz de reconocer diferentes oportunidades o alternativas de actuación y, sobretodo, se caracteriza por tomar la iniciativa al tratar con una tarea particular.
 - **Agente social:** Es el resultado de añadir al agente proactivo la habilidad de comunicarse y colaborar con otros agentes.
-

Esta clasificación es una visión tradicional de los tipos de agentes inteligentes existentes. No obstante, posteriormente han surgido nuevas formas de clasificarlos. A continuación, exponemos brevemente los tipos de agentes existentes según diferentes criterios.

3.1.1. Tipos de agentes

Se puede hacer una clasificación de los agentes desde varias perspectivas: según sus características individuales, según el entorno en el que trabajan, según su modo de interacción, según el modo de organización y según su utilidad [27].

Tipos de agentes según sus características individuales

Se distingue entre los agentes reactivos y agentes cognitivos.

- **Agentes reactivos:** Son agentes que realizan tareas sencillas, bajo un ciclo de *percepción/acción* [71]. El agente reactivo recibe una percepción de su entorno, y en base a ella realiza una acción, que puede consistir en cambiar su estado interno o modificar el entorno. Los agentes reactivos no realizan procesos de razonamiento, y carecen de mecanismos de representación de conocimiento.
- **Agentes Cognitivos:** Estos agentes realizan tareas más complejas, utilizando algún tipo de representación simbólica del conocimiento [20, 87]. Para cumplir su objetivo, deben realizar procesos de razonamiento, planificación y aprendizaje. Su modelo computacional se basa en un ciclo de *percepción, asimilación, razonamiento y actuación* [47].

Tipos de agentes según el entorno en el que funcionan

Se entiende por entorno del agente toda la infraestructura computacional que le rodea, y que le proporciona los medios necesarios para desarrollar su actividad. Bajo este punto de vista, podemos considerar dos tipos de agentes:

- **Agentes que requieren un entorno especial:** Precisan de una plataforma software específica para su funcionamiento. Ejemplos de ello son los agentes móviles, ya que cada plataforma en la que pueden encontrarse proporciona los mecanismos para gestionar su ciclo de vida y para facilitar su desplazamiento entre nodos de una red; y los agentes propios del estándar FIPA (véase sección 3.3.1).
- **Agentes que se ejecutan en las plataformas computacionales existentes:** Son agentes que se crean mediante los recursos del sistema operativo y siguen el ciclo de vida de cualquier aplicación. Se suelen implementar en lenguajes independientes de la plataforma empleada, como Java.

Tipos de agentes según el modo de interacción

La interacción se entiende como la comunicación e intercambio de información entre un agente y otras entidades. Estas interacciones son de tres tipos: *agente-agente*, mediante lenguajes estándar de comunicación entre agentes como ACL; *agente-persona*, utilizando los medios adecuados para que las personas puedan comunicarse con los agentes; y *agente-entorno*, que consiste en intercambiar información con elementos como bases de datos, sistema operativo, etc.

Tipos de agentes según el modo de organización

Los agentes se pueden clasificar también según el tipo de estructura organizacional y sus capacidades. La clasificación más común desde este punto de vista distingue entre agentes individuales y agentes que cooperan [27, 87]:

- **Agentes individuales:** Son agentes sin capacidad de cooperación, que realizan sus tareas de forma individual y sin precisar de la colaboración de otros agentes.
 - **Agentes cooperativos:** Pueden realizar tareas individualmente o en colaboración con otros agentes. Suelen formar parte de organizaciones de agentes.
-

- **Agentes competitivos:** Son agentes que para alcanzar su objetivo normalmente deben competir con otros agentes de su entorno por el uso un recurso común limitado.

Tipos de agentes según su utilidad

Los agentes se clasifican bajo este criterio según la finalidad o propósito con el que han sido creados. Las áreas donde se han aplicado las Tecnologías de Agentes han ido creciendo de forma progresiva. Actualmente existen Sistemas Multi-Agente (SMA) en áreas como el comercio electrónico, telecomunicaciones, economía, administración, procesos industriales, ocio y entretenimiento, etc. [1, 4, 5, 26, 75]. Las tareas que llevan a cabo los agentes comprenden actividades como la monitorización, diagnóstico, control de sistemas, búsqueda y recuperación de información, clasificación, tareas de mediación, etc.

3.2. Arquitecturas de Diseño de Agentes

Tras revisar brevemente el concepto de agente y dar una clasificación de agentes bajo diferentes criterios, en esta sección presentamos las arquitecturas existentes más comunes para construir agentes, que se clasifican como: reactivas, deliberativas e híbridas.

3.2.1. Arquitecturas Deliberativas. La Arquitectura BDI

Las arquitecturas deliberativas son aquellas que utilizan modelos de representación simbólica del conocimiento. Los agentes que siguen esta arquitectura parten de un estado inicial y son capaces de deliberar, es decir, generar planes para alcanzar sus propios objetivos [63].

Un agente deliberativo debe disponer de un modelo simbólico del mundo, representado explícitamente, para realizar un razonamiento lógico a partir de él y tomar las decisiones oportunas. La principal ventaja de esta arquitectura es que el conocimiento es más fácil de entender y codificar por el ser humano. Las desventajas que presenta son la dificultad

para traducir el mundo real a un modelo simbólico adecuado y preciso, y el elevado coste temporal, necesario a veces para trabajar con este modelo [6].

Sin duda, la arquitectura deliberativa BDI (*Belief, Desire, Intention*) es la más estudiada y posiblemente la arquitectura deliberativa más extendida [73].

Arquitectura BDI

La arquitectura BDI, cuyos acrónimos en español significan *Creencia, Deseo e Intención*, combina un sólido modelo filosófico del razonamiento humano y una semántica abstracta, lógica y elegante [11, 12, 21, 23, 37]. Desde su establecimiento en la década de los 80, prácticamente ha permanecido inalterable. Los tres componentes básicos de esta arquitectura son las creencias, los deseos u objetivos, y las intenciones o planes.

Las **creencias** representan conocimiento de los agentes sobre el mundo. En términos computacionales, son una forma de representar el estado del mundo, ya sea mediante el valor de una variable, una BD relacional o expresiones simbólicas. Las creencias son esenciales debido al dinamismo y la visión local del mundo. Dado que las creencias pueden representar información imperfecta del mundo, su semántica subyacente debe obedecer a una lógica de creencias, aunque la representación computacional no necesite ser puramente lógica o simbólica.

Un **deseo** (objetivo) representa un estado final deseado. En términos informáticos, un objetivo para un agente será alcanzar cierto valor en una variable, un registro, o cumplir una expresión simbólica representada en alguna formalización lógica.

Con las creencias y los objetivos no es suficiente para poder modelar un sistema capaz de operar en entornos dinámicos e inciertos: si hemos decidido sobre un curso de acción (plan), y el mundo cambia ligeramente, deberíamos cuestionarnos si seguir con el plan o replantearlo. La respuesta es que el sistema necesita acometer los planes y sub-objetivos que planifica, pero además debe ser capaz de reconsiderarlos en momentos cruciales. Estos planes acometidos o formas de proceder se llaman **intenciones**, y representan el tercer componente

la arquitectura BDI. Son un subconjunto de los deseos: aquellos que el agente se ha propuesto alcanzar en un momento dado, y que caracterizan el “estado mental” actual de dicho agente.

Para determinar el grado de persistencia de una intención (su resistencia a ser cambiada ante percepciones de cambio en el mundo), se han definido tres estrategias de dedicación:

- Dedicación total (a ciegas): El agente seguirá con su intención actual hasta que crea haberla alcanzado.
- Dedicación firme: El agente seguirá con su intención actual hasta que crea haberla alcanzado, o bien crea que no puede ser lograda.
- Dedicación abierta: Se mantendrá la intención solamente mientras el agente la considere viable.

3.2.2. Arquitecturas Reactivas

Las arquitecturas reactivas implementan la forma de actuar de los agentes como un mecanismo de correspondencia directa *percepción-acción*, y se basan en mecanismos de respuesta ante estímulos. A diferencia de las arquitecturas deliberativas, carecen de modelo simbólico y por tanto no emplean ningún mecanismo de razonamiento. Su principal ventaja es la mayor efectividad, al prescindir de un mecanismo de razonamiento complejo. Por contra, presentan el inconveniente de que el comportamiento del agente es fijo, y este no puede aprender y razonar a partir de la información que recibe. La arquitectura reactiva más conocida es la llamada *Arquitectura de Subsunción* [13]. Esta arquitectura define una serie de capas conectadas a sensores que transmiten información en tiempo real, de forma que las capas componen una jerarquía de tareas en la que los niveles inferiores tienen menos control sobre los niveles superiores. La figura 3.1 muestra el esquema básico de la arquitectura de subsunción.

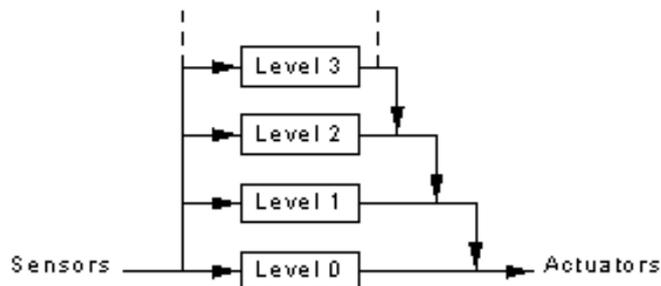


Figura 3.1: Esquema básico de la arquitectura de subsunción

3.2.3. Arquitecturas Híbridas

Para intentar solventar las limitaciones de las arquitecturas deliberativas y reactivas, se han propuesto arquitecturas híbridas, que combinan aspectos de ambos modelos. Los agentes en arquitecturas híbridas presentan dos subsistemas: uno deliberativo, que utiliza un modelo simbólico para generar planes, y otro reactivo, centrado en reaccionar ante eventos en el entorno.

La Figura 3.2 muestra el esquema básico de las arquitecturas híbridas, las cuales se suelen estructurar por capas. Así, la forma de estructurarse puede ser [30, 70]:

- *Vertical*: Una única capa tiene acceso a los sensores y actuadores.
- *Horizontal*: Todas las capas tienen acceso a los sensores y actuadores.

Al igual que en las arquitecturas de subsunción, las capas se disponen jerárquicamente, brindando información sobre el entorno a diferentes niveles de abstracción. Los tres niveles básicos presentes en la mayoría de arquitecturas son:

- *Nivel reactivo*: Es el nivel más bajo, y en él se toman decisiones acerca de cómo responder ante los estímulos recibidos en tiempo real. Normalmente se basa en arquitecturas de subsunción.

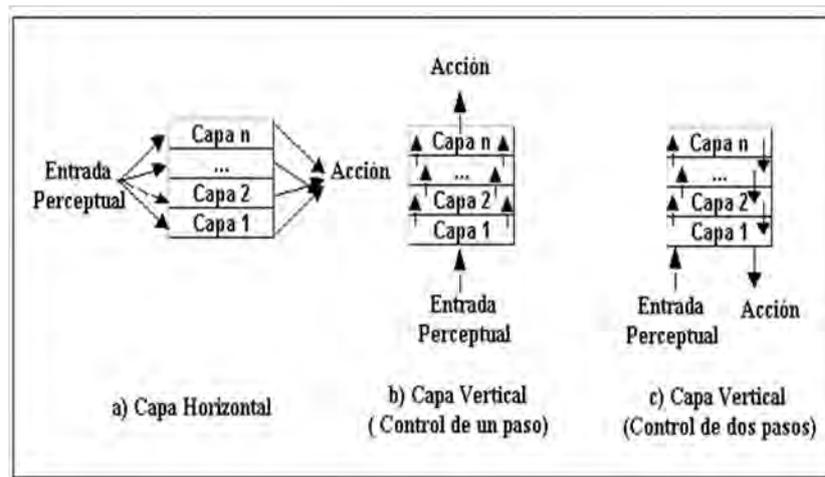


Figura 3.2: Diferentes esquemas de arquitectura híbrida

- *Nivel de conocimiento:* Basado en el conocimiento que el agente posee sobre el medio. En este nivel suele emplearse una representación simbólica del mundo.
- *Nivel social:* Es la capa de más alto nivel, y se encarga de aspectos sociales del entorno, incluyendo información de otros agentes, deseos, intenciones, etc.

3.3. Arquitecturas Multi-Agente: El Estándar FIPA

La necesidad de desarrollar aplicaciones complejas, compuestas de multitud de subsistemas que interactúen entre sí hace necesaria la utilización de *Sistemas Multi-Agente* (SMA), que son sistemas compuestos por un número más o menos grande de agentes que trabajan de forma organizada y coordinada para la gestión inteligente de un sistema complejo, integrando los objetivos particulares de cada uno de los subsistemas que lo componen en un objetivo común.

Los SMA son apropiados en muchos casos: en problemas físicamente distribuidos, cuando la complejidad de la solución requiere de experiencia heterogénea o cuando el problema está definido sobre redes de computadores. En los últimos años, la creciente complejidad

de los problemas hace que cada vez sea más necesario el uso de arquitecturas basadas en SMA [66], y son muchas las aplicaciones basadas en arquitecturas multi-agente que diferentes autores han desarrollado y propuesto en la literatura [4, 5, 20, 26, 35, 75, 87]

Como ocurre con el uso creciente de toda nueva tecnología, surgen dos cuestiones a resolver en la aplicabilidad de SMA: la *interoperabilidad*, que es la facilidad de conexión e integración de SMA, y la *apertura*, o posibilidad de extensión. Por esta razón es importante disponer de estándares de desarrollo, y en el ámbito de los SMA el estándar adoptado por la mayoría de entornos de desarrollo en la actualidad es **FIPA** (*Foundation for Intelligent Physical Agents*) [32].

FIPA nació en 1996 como una asociación para el desarrollo de estándares relativos a tecnología de agentes software. Su principal característica, y la que quizá lo ha convertido en el estándar más extendido, es que únicamente define el comportamiento externo (interfaz) del sistema, dejando a cargo del equipo de desarrollo toda decisión de diseño.

Algunos de los principales logros alcanzados con FIPA son los siguientes [6]:

- Un conjunto de especificaciones estándares que apoyan la comunicación entre agentes, y una serie de servicios intermedios clave (*middleware*).
 - Una arquitectura abstracta con una vista completa de los estándares FIPA2000.
 - Un lenguaje de comunicación entre agentes (FIPA-ACL), además de una selección de lenguajes de contenido, como FIPA-SL.
 - Un conjunto de protocolos de interacción, que abarcan desde un simple paso de mensajes hasta transacciones más complejas.
 - Varias herramientas y librerías para el desarrollo de SMA bajo este estándar, siendo JADE una de las más utilizadas y la elegida para el desarrollo de nuestro sistema.
 - Especificación UML extendida, específica para el desarrollo de agentes: AUML.
-

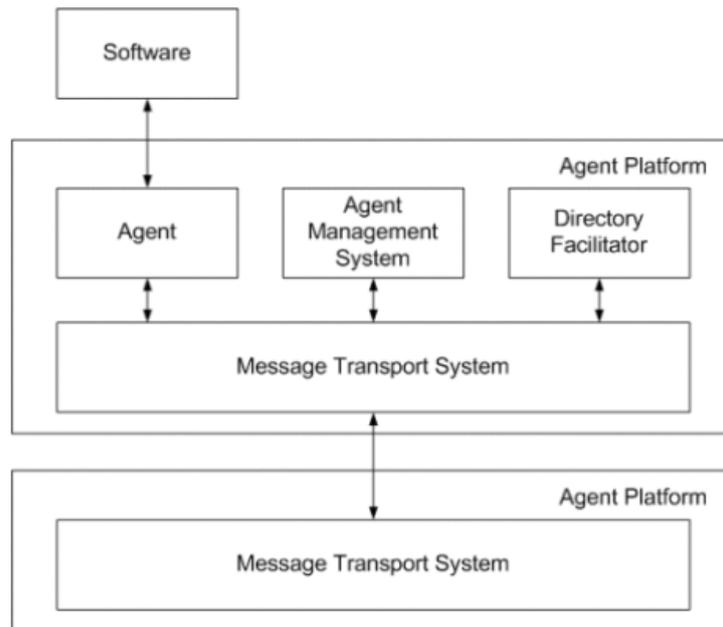


Figura 3.3: Modelo de referencia del estándar FIPA

En las siguientes subsecciones estudiaremos brevemente tres de los aspectos fundamentales del estándar FIPA: el modelo o arquitectura FIPA, que es el modelo de referencia de la arquitectura seguida en FIPA, el ciclo de vida de los agentes en FIPA y la Comunicación entre Agentes en FIPA.

3.3.1. Arquitectura FIPA

La arquitectura seguida en FIPA define un modelo para la administración de agentes en un contexto en el que los agentes FIPA pueden existir, operar y ser gestionados, estableciendo así una referencia lógica para la creación, registro, localización, comunicación, migración y operaciones entre agentes. El modelo de referencia FIPA consta de una serie de componentes, mostrados en la Figura 3.3, y que pasamos a describir a continuación:

- **Plataforma de Agentes:** Es la infraestructura en la que se establecen y utilizan los agentes. El diseño interno de esta plataforma se deja en manos de los desarrolladores y

no forma parte del estándar FIPA más allá de lo aquí expuesto. Dado que una misma plataforma de agentes puede encontrarse distribuida en varios ordenadores, los agentes que residen en la misma no tienen por qué encontrarse en el mismo equipo.

- **Agente:** Un agente en FIPA se considera como un proceso computacional que reside en la plataforma de agentes y normalmente ofrece uno o más servicios, cada uno de los cuales se publica junto a una descripción del mismo.
- **Facilitador de Directorio (DF):** Este componente opcional de la plataforma de agentes proporciona un servicio de “páginas amarillas” al resto de agentes. Este componente mantiene una lista completa y precisa de los agentes existentes y proporciona información actualizada sobre los servicios que estos ofrecen.
- **Sistema de Gestión de Agentes (AMS):** Es el elemento de gestión principal, que conoce en todo momento el estado de la plataforma y de los agentes que pertenecen a ella. Algunos de los servicios que ofrece son la creación, eliminación y control de los cambios de estado entre agentes (véase sección 3.3.2), supervisión para el registro de nuevos agentes en la plataforma y gestión de los recursos y canales de comunicación.
- **Sistema de Transporte de Mensajes (MTS):** Servicio proporcionado por la plataforma de agentes para transportar mensajes FIPA-ACL entre agentes, ya sea dentro de la misma plataforma o entre plataformas diferentes.

3.3.2. Ciclo de vida de los Agentes en FIPA

El ciclo de vida de un agente define los estados en los cuales se puede encontrar en cada momento, así como las transiciones que este puede realizar entre dichos estados. El modelo de ciclo de vida propuesto por FIPA consta de los siguientes estados:

- *Iniciado:* El agente ha sido creado, pero aún no se ha registrado en el AMS, no tiene nombre ni dirección y tampoco se puede comunicar con otros agentes.
-

-
- *Activo*: El agente está registrado en el AMS, tiene un nombre, una dirección y puede acceder a los diferentes servicios ofrecidos.
 - *Suspendido*: El agente está parado, su hilo de ejecución está detenido y no ejecuta ninguna tarea.
 - *En espera*: El agente está bloqueado, a la espera de un mensaje, recurso u otro tipo de evento. Se desbloqueará cuando se cumpla una determinada condición.
 - *Desconocido*: El agente ha sido eliminado, su hilo de ejecución ha terminado y ha sido borrado del AMS.
 - *En tránsito*: Un agente móvil entra en este estado cuando está migrando de una localización a otra. Vuelve a estar activo cuando llega a su destino.

Un agente puede cambiar de un estado a otro a través de transiciones. Las diferentes acciones que puede llevar a cabo un agente para ello son:

- *Crear*: Creación o instalación de un nuevo agente.
 - *Invocar*: Invocación de un nuevo agente. El agente pasa a estar activo.
 - *Suspender*: Pone a un agente en estado suspendido. Puede ser iniciado por el propio agente o por el AMS.
 - *Reanudar*: Continúa con la ejecución de un agente que se encontraba en estado suspendido. Sólo puede ser iniciado por el AMS.
 - *Esperar*: Pone a un agente en estado de espera. Sólo puede ser iniciado por el propio agente.
 - *Despertar*: Continúa con la ejecución de un agente que se encontraba en estado de espera. Sólo puede ser iniciado por el AMS.
-

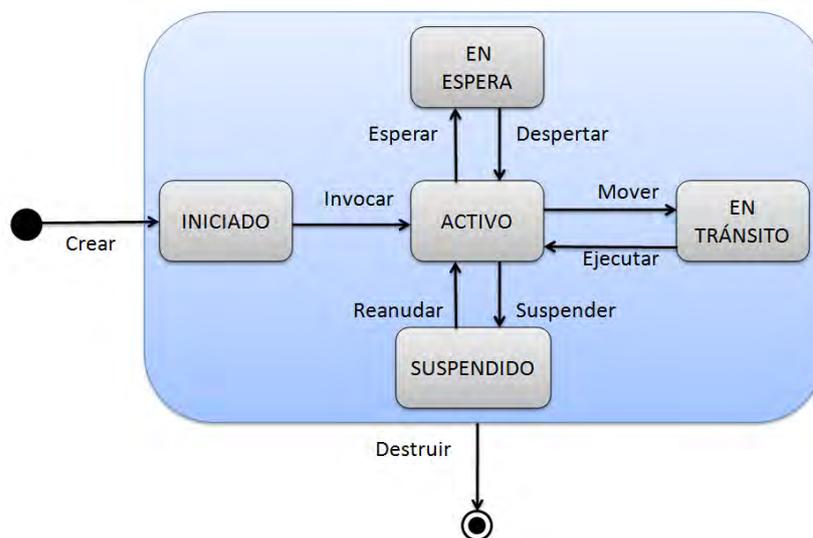


Figura 3.4: Esquema del ciclo de vida de un agente en FIPA

- *Mover*: Envía al agente a otra plataforma, asignándole el estado de tránsito. Sólo puede ser iniciado por el propio agente.
- *Ejecutar*: Continúa con la ejecución de un agente que se encontraba en estado de tránsito. Sólo puede ser iniciado por el AMS.
- *Destruir*: Terminación normal o forzosa de un agente. Sólo puede ser iniciado por el AMS y no puede ser ignorado por el agente.

La figura 3.4 muestra de forma esquemática los estados y transiciones que componen el ciclo de vida estudiado.

3.3.3. Comunicación entre Agentes en FIPA

Una de las características clave en los agentes software es, tal y como hemos visto, la capacidad de estos para comunicarse entre sí. En el caso de FIPA, esta comunicación se lleva a cabo mediante el empleo de mensajes FIPA-ACL. El formato de mensajes FIPA-ACL define la comunicación en términos de una función o acción, llamada *acto comunicativo* o

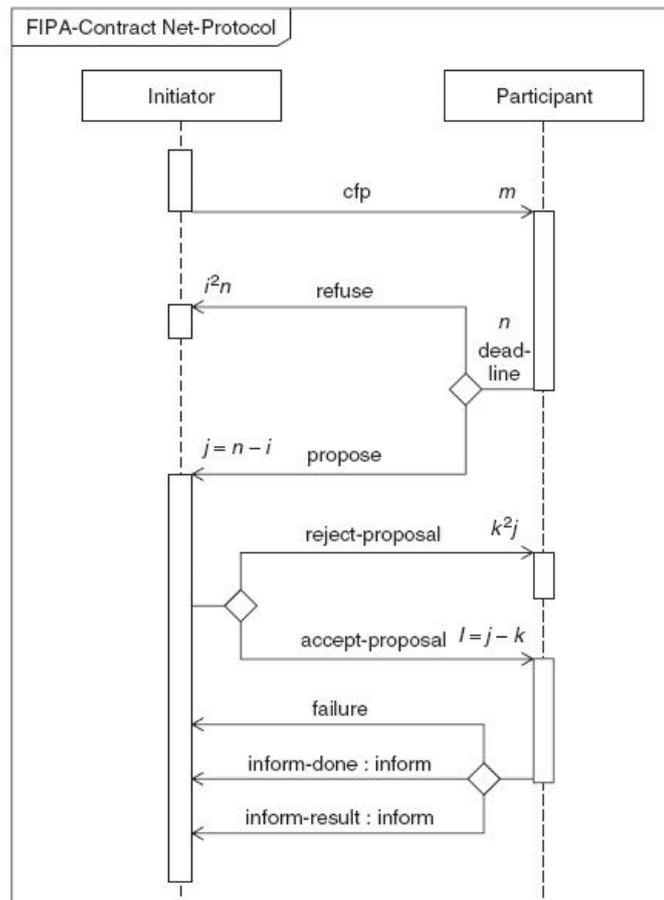


Figura 3.5: Protocolo de Interacción FIPA Contract Net

protocolo de comunicación. Existen 22 actos comunicativos diferentes, basados por lo general en la Teoría del Discurso de J. Searle [78]. Estos actos comunicativos aparecen en detalle en las especificaciones de FIPA [32], siendo algunos de los más empleados los siguientes:

- *Call for Proposal (cfp)*: Forma parte de un protocolo de comunicación completo llamado *Contract Net*, cuya estructura se muestra en el diagrama de secuencia UML de la Figura 3.5. El agente envía un mensaje (cfp), en el que propone a uno o varios receptores participar o llevar a cabo una acción. La respuesta por parte de un agente receptor puede ser un mensaje de tipo *Accept Proposal* si este acepta la propuesta, o de tipo *Reject Proposal* si este decide rechazarla.

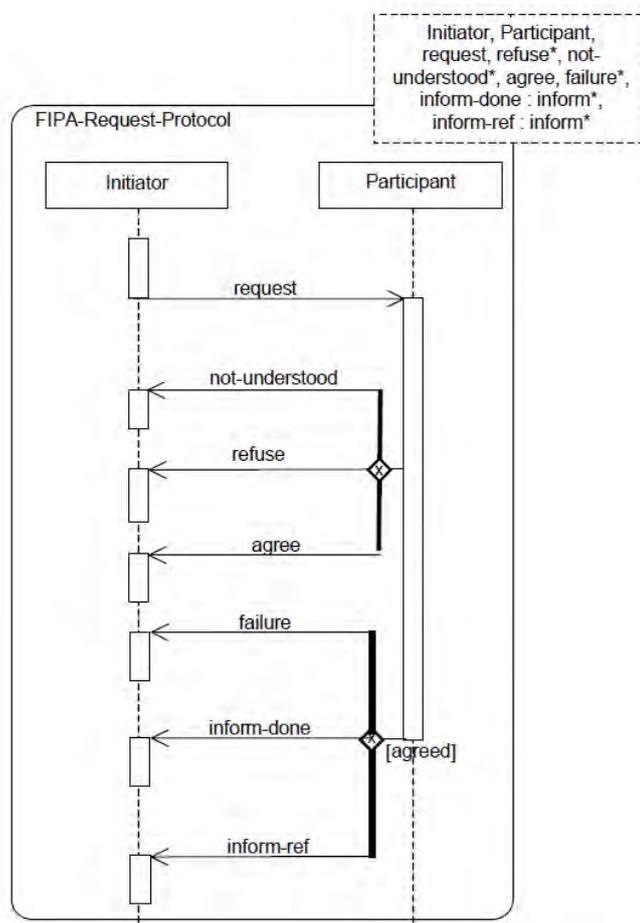


Figura 3.6: Protocolo de Interacción FIPA Request

- *Request*: El emisor pide a el/los receptor/es la realización de una acción. Se trata de un protocolo de comunicación muy utilizado, en el que el receptor responde en primer lugar indicando si acepta la petición (*Agree*) o la rechaza (*Refuse*). En caso de haberla aceptado, enviará además el resultado de la acción que se le ha pedido realizar mediante un mensaje *Inform*, o bien un mensaje *Failure* en caso de haber ocurrido algún error. En la Figura 3.6 aparece un diagrama de secuencia UML para representar el protocolo de interacción *Request*.
- *Query*: Consiste en la consulta de un agente a otro. Puede ser de dos tipos: consulta sobre la veracidad o falsedad de una proposición (*Query-If*), o bien sobre el objeto al que se refiere una expresión referencia dada (*Query-Ref*).

3.4. La Plataforma Multi-Agente JADE

JADE¹ es una conocida plataforma de desarrollo de agentes y SMA, basada en el estándar FIPA, y ha sido la tecnología elegida para el desarrollo de nuestro sistema. Por esta razón, en esta sección veremos en qué consiste y sus principales características.

La plataforma JADE (*Java Agent DEvelopment Framework*) es un entorno software, implementado en Java, para simplificar la implementación de SMA a través de un *middleware* acorde con las especificaciones FIPA y dotado de un conjunto de herramientas gráficas para ayudar al desarrollador con tareas de depuración y puesta en marcha de la aplicación [6].

Veamos algunas de las funcionalidades más atractivas que JADE proporciona al programador:

- Un sistema en el que cada agente se ejecuta como un hilo separado de los demás, capaz de ejecutarse en máquinas remotas y de comunicarse con otros agentes de forma totalmente transparente.
- Transporte eficiente de mensajes asíncronos, con una API transparente.
- Implementación de servicios de *Páginas amarillas* y *Páginas blancas*.
- Administración sencilla y eficiente del ciclo de vida de los agentes. Los agentes reciben un identificador único (*AID, Agent Identifier*) y una dirección automáticamente durante su creación. Se proporcionan APIs para crear, suspender, reanudar, bloquear, despertar, migrar, clonar y destruir agentes.
- Soporte para movilidad de agentes, entre procesos y entre distintas máquinas, de forma transparente.
- Herramientas para interceptar y analizar mensajes de comunicación entre agentes.

¹<http://jade.tilab.com>

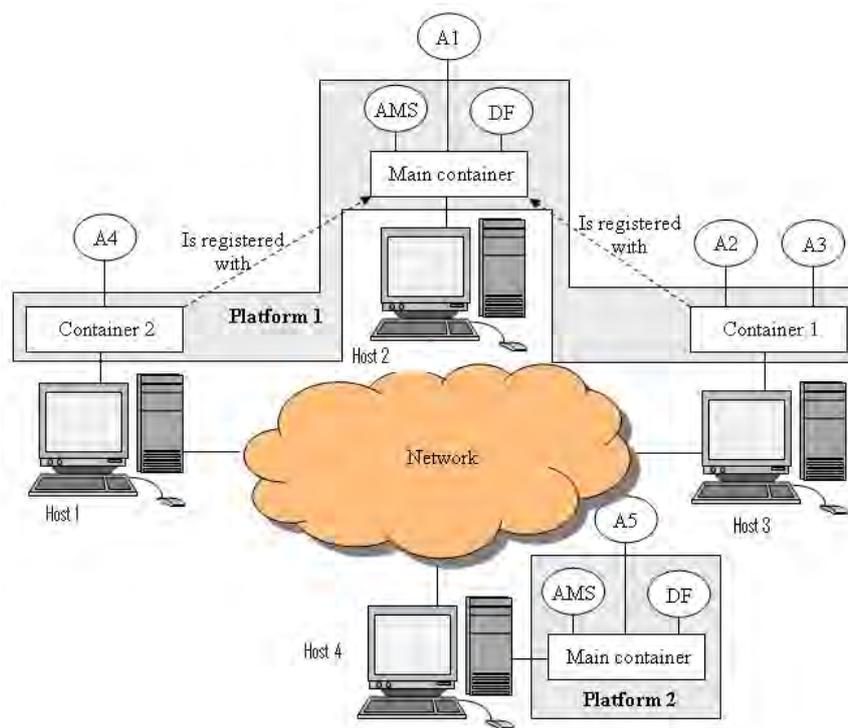


Figura 3.7: Arquitectura de JADE

- Soporte para ontologías y lenguajes de contenido. Dada la importancia de este aspecto en el desarrollo de nuestro SMA, trataremos sobre esta cuestión con mayor detalle en la siguiente sección.
- Integración con tecnologías Web como JSP, servlets, applets y servicios Web.
- Soporte para la plataforma J2ME para dispositivos móviles.

La arquitectura de JADE está basada en los componentes de la arquitectura FIPA estudiada, tal y como podemos apreciar en la Figura 3.7, ya que como hemos dicho JADE sigue las especificaciones de dicho estándar. A continuación explicamos los principales aspectos presentes en dicha arquitectura.

La principal particularidad añadida de esta arquitectura es que una plataforma de agentes se compone de uno o varios contenedores de agentes (*containers*), donde residen los agentes.

Estos contenedores, al igual que la plataforma que los contiene, pueden encontrarse en un mismo equipo o estar distribuidos en una red. Existe un contenedor especial, el contenedor principal (*main container*), que es el núcleo de la plataforma y el primer contenedor en ponerse en funcionamiento, además de ser el responsable de albergar el AMS y DF. El resto de contenedores debe unirse al contenedor principal, registrándose en él por medio de un registro o tabla de contenedores (CT), para formar parte del sistema.

Una vez vistas las características básicas de JADE, dedicaremos la última sección de este capítulo a un aspecto de los SMA de vital importancia en este trabajo: el empleo de ontologías.

3.5. Ontologías en Sistemas Multi-Agente

El desarrollo de aplicaciones multi-agente reales requiere, en muchas ocasiones, el manejo de expresiones de contenido complejas por parte de los agentes [35, 56]. Para ayudar a realizar esta tarea, entornos como JADE brindan la posibilidad de emplear ontologías y lenguajes de contenido. En esta sección revisaremos de forma breve en qué consiste el empleo de ontologías en los SMA y, en particular, en la plataforma JADE, pues tal y como veremos en el capítulo 4, serán aspectos clave en el desarrollo de nuestro sistema, que contará con una ontología utilizada por los agentes para el intercambio de información sobre el proceso de consenso en el que participan.

Por defecto, un mensaje ACL de comunicación entre agentes consiste en una serie de campos, siendo uno de ellos el contenido del mismo. Este contenido es, por defecto, una simple cadena o secuencia de bytes. Sin embargo, en aplicaciones reales, los agentes necesitan a menudo transmitir o recibir información más compleja. Para ello, el contenido del mensaje ACL debe obedecer a una cierta sintaxis, conocida como *lenguaje de contenido*. FIPA no obliga a emplear ningún lenguaje de contenido en particular, pero recomienda el lenguaje SL (*Semantic Language*). Para que la comunicación entre agentes sea exitosa, todos ellos deben

“conocer” el lenguaje de contenido utilizado para la codificación de los mensajes.

Para crear mensajes complejos bajo un lenguaje de contenido, es necesario haber definido previamente un vocabulario o conjunto de términos, y una semántica asociada a dicho vocabulario, de manera que el contenido de dichos mensajes tenga un significado claro para cualquiera de los agentes participantes en el acto comunicativo y no sean meros datos sin significado. Esto se consigue mediante el diseño y establecimiento de una **ontología** para el sistema [79]. En términos informáticos, una ontología es una entidad computacional y artificial, que se crea con el objeto de constituir una forma común y compartida de conocimiento de un dominio [64].

Una ontología en SMA supone la especificación de una conceptualización, la descripción de los conceptos y relaciones entre ellos, que pueden formar parte del conocimiento de un agente o una sociedad de agentes. En la actualidad, son muchas las propuestas y aplicaciones realizadas sobre SMA en las que se utilizan ontologías. En el siguiente apartado revisaremos brevemente algunas de estas aplicaciones.

3.5.1. Ejemplos de uso de ontologías en SMA

En la actualidad existen en la literatura diferentes aplicaciones basadas en SMA en las que se emplean ontologías para el manejo de información y razonamiento por parte de los agentes. En este apartado citaremos algunas de estas aplicaciones.

García-Sánchez et al. [35] han desarrollado en la Universidad de Murcia el entorno SEMMAS, un sistema basado en ontologías capaz de integrar agentes inteligentes y servicios Web semánticos. Gracias al empleo de la ontología de SEMMAS, se consigue explotar el potencial de ambas tecnologías, así como superar las limitaciones de comunicación existentes en ambas. La ontología desarrollada permite una comunicación transparente entre agente y servicio Web, de manera que ninguno de estos componentes precisa de cambios en su implementación y especificación originales. Algunos de los objetos representables por la ontología de SEM-

MAS son las tareas y responsabilidades asumidas por un agente, así como la descripción y prestaciones que ofrece un servicio Web concreto.

El sistema desarrollado por Middleton et al. [69], consiste en un sistema de recomendación híbrido de artículos académicos on-line, provisto de una ontología para modelar perfiles de usuario. Un conjunto de agentes Web utiliza dicha ontología para determinar el perfil de un usuario en función de sus preferencias al utilizar el sistema, así como para generar las recomendaciones en base a dichas preferencias.

Jung et al. [48] proponen en su trabajo un SMA compuesto por agentes mediadores, para estimar la semántica asociada a espacios Web desconocidos, aprendiendo a partir de los fragmentos leídos por los usuarios durante sus búsquedas. Este sistema de recuperación de información emplea una ontología para recopilar información de sitios Web nuevos así como de otros ya conocidos, y reorganizar dicha información de cara a proporcionar información semántica relativa a los nuevos sitios visitados.

Una vez revisadas algunas de las aplicaciones de SMA provistos de ontologías, pasamos a estudiar la estructura de las mismas bajo el punto de vista de la plataforma JADE, ya que será la plataforma empleada en este trabajo de investigación.

3.5.2. Estructura de una ontología en JADE

Las ontologías constan de una serie de componentes, que forman un *modelo de contenido*. En el caso de JADE, el modelo de contenido proporcionado se compone de los siguientes elementos [6]:

- **Conceptos:** Expresiones que representan objetos, cuya información se estructura en varios atributos. No aparecen aislados en los mensajes sino incluidos en predicados y acciones de agentes. Sus atributos pueden ser de tipos de datos simples (primitivas) o pueden ser instancias de otros conceptos.
-

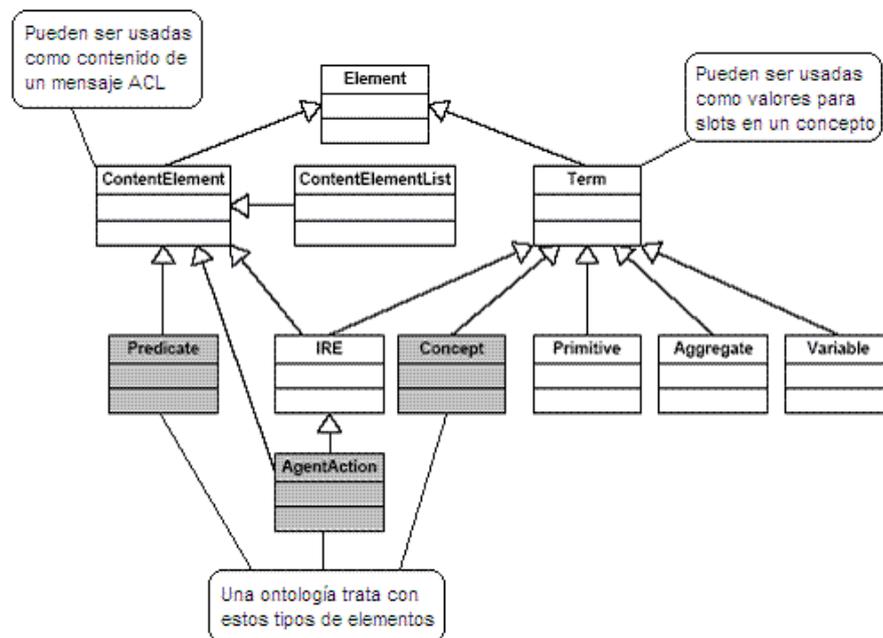


Figura 3.8: Modelo de contenido para ontologías en JADE

- **Predicados:** Son expresiones sobre el estado del mundo, que pueden ser verdaderas o falsas. Se suelen emplear en mensajes *Query-If* y en mensajes de respuesta *Inform*.
- **Acciones de los agentes:** Son expresiones que indican acciones que pueden realizar los agentes. Normalmente constituyen el contenido de mensajes *Request*.
- **Otros elementos:** primitivas (elementos atómicos como números o cadenas de caracteres), agregaciones (atributos compuestos de múltiples elementos), expresiones (identifican las entidades para las que se cumple un predicado), variables.

El modelo de contenido de JADE [6] aparece representado en la Figura 3.8. Como podemos observar, este modelo consta de un amplio conjunto de componentes estructurados de forma jerárquica. Como veremos en el capítulo 4, la ontología de nuestro sistema (y en general la mayoría de ontologías desarrolladas en JADE) utiliza únicamente predicados y acciones de agente de forma directa. Estos a su vez se componen de uno o varios *términos*, los cuales pueden ser conceptos (explícitamente definidos por el desarrollador de la ontología),

primitivas o agregaciones, entre otros. Además, un agente debe encapsular todo predicado o acción de agente como un objeto de la clase *ContentElement* para poder enviarlo a otros agentes como el contenido de un mensaje ACL.

En el siguiente capítulo, dedicado al desarrollo del trabajo de investigación, se llevará a cabo también el diseño de los conceptos, predicados y acciones de agentes específicos para la ontología utilizada en nuestro sistema multi-agente.

Desarrollo del sistema (COMAS)

En este capítulo presentamos el Sistema de Apoyo al Consenso basado en un Sistema Multi-Agente desarrollado para el presente trabajo de investigación (COMAS). Para ello, comenzamos dando una breve introducción al sistema desarrollado para, a continuación, estudiar los principales aspectos relativos a la arquitectura multi-agente diseñada y el mecanismo de comunicación entre agentes. Finalmente, concluiremos el capítulo mostrando la ontología para el manejo de la información intercambiada entre los agentes.

4.1. COMAS (*Consensus Multi-Agent based System*)

COMAS es un Sistema de Apoyo al Consenso basado en un Sistema Multi-Agente (SMA) desarrollado con el propósito de facilitar, guiar y automatizar procesos de búsqueda de consenso en problemas de TDG, cada vez más frecuentes en la mayoría de organizaciones y entornos sociales y empresariales. El sistema se compone de un conjunto de agentes inteligentes, cada uno de ellos con un rol y responsabilidades determinados, encargados de guiar, supervisar y controlar los procesos de consenso. Esto nos permite evaluar diferentes modelos de consenso, lo cual servirá como base para futuros desarrollos en este campo de investigación, así como para llevar a cabo una primera simulación de procesos de consenso en la plataforma bajo el uso de diferentes medidas de consenso, como veremos en el capítulo 5.

El modelo de consenso utilizado por COMAS en el momento de redactar esta memoria de investigación es el expuesto en la sección 2.4, en el cual consideramos la noción de *soft consensus* como mayoría difusa para determinar la existencia o no de consenso. Por otra parte, todo SMA, como es el caso de COMAS, implica durante su concepción el diseño de la arquitectura del mismo, es decir, la especificación del conjunto de agentes necesarios para llevar a cabo el objetivo que se persigue. Así mismo, resulta necesario definir los componentes de información que se utilizarán para la comunicación entre agentes. Estos aspectos relativos al SMA implementado en COMAS serán estudiados en la siguiente sección.

4.2. Arquitectura Multi-Agente

En esta sección presentamos la arquitectura diseñada para el sistema multi-agente sobre el que se sustenta nuestro sistema. Dicha arquitectura está basada en el estándar FIPA¹ (*Foundation for Intelligent and Physical Agents*) que estudiamos en el capítulo 3, e implementada con la ayuda de la plataforma de desarrollo de agentes JADE.

COMAS es un sistema multi-agente cooperativo, en el que los agentes deben de colaborar entre sí para alcanzar un objetivo común: lograr el consenso. Dado que en todo modelo de consenso, y en particular en el modelo implementado en nuestro sistema, se distinguen diferentes roles participantes en el problema de TDG, en nuestro sistema multi-agente existirán diferentes agentes con roles determinados, definiéndose para cada uno de estos roles una serie de responsabilidades y líneas de actuación claras.

Así, los primeros tipos o roles de agentes en COMAS surgieron directamente del modelo de consenso considerado, obteniendo así la figura del *Agente Moderador* y *Agente Experto*.

- *Agente Moderador*: Este agente asume el rol del moderador humano en el proceso de consenso, y es el principal responsable de garantizar un correcto desarrollo en dicho

¹<http://www.fipa.org>

proceso de forma global. Existirá un agente moderador por cada problema que se esté abordando en un momento dado.

- *Agente Experto*: Un agente experto representa a un experto humano en el sistema, actuando como tal de forma autónoma. El número de agentes expertos existentes en el sistema es variable, y depende del número de expertos que participen en el problema de TDG en un momento dado.

Debido a que las responsabilidades inherentes a un moderador humano resultaban demasiado complejas para ser asumidas únicamente por el agente moderador, decidimos añadir una serie de agentes específicos y propios de COMAS, con el objetivo de dar soporte al moderador con algunas de las tareas de las que se componen el proceso completo de consenso. Estos agentes suponen diferentes especializaciones del moderador y son los siguientes:

- *Agente Evaluador de Consenso*: Este agente se encarga de la obtención del grado de consenso alcanzado en cada ronda, así como de notificar al agente moderador sobre ello.
- *Agente Identificador de Cambios*: Su responsabilidad se centra en llevar a cabo las operaciones necesarias para realizar la fase de *Generación de Recomendaciones*, estudiada en la sección 2.4.
- *Agente Analista*: Este agente complementario al conjunto de agentes dedicados al proceso de consenso asume la labor de almacenar la información relativa a cada proceso de forma persistente en una base de datos.

Otros agentes y componentes fundamentales en la arquitectura del sistema son los siguientes:

- *Agente Interfaz*: Es un agente intermediario entre los agentes de COMAS y la interfaz de usuario. Proporciona a los agentes los parámetros de entrada introducidos por medio
-

de la interfaz y recoge los resultados de la ejecución del proceso de consenso para mostrarlos en dicha interfaz. Existe una instancia de este agente para cada usuario o experto participando en el problema.

- *Agentes y componentes FIPA*: La plataforma JADE, al estar basada en el estándar FIPA, proporciona una serie de agentes y componentes de utilidad para nuestra arquitectura, como son el agente *DF (Directory Facilitator)* que registra los servicios ofrecidos por cada agente de COMAS, el *AMS (Agent Management System)* para administrar y controlar a todos los agentes del sistema durante su ciclo de vida, y el *MTS (Message Transport System)*, que permite una ejecución distribuida de los procesos de consenso, con agentes provenientes de diferentes plataformas.
- *Ontología y mensajes ACL*: Nuestros agentes se comunican mutuamente por intercambio de mensajes FIPA ACL. Dado que el sistema cuenta con una ontología diseñada para que los agentes compartan el mismo vocabulario y semántica asociados a los problemas a tratar, los mensajes que estos intercambien tendrán como contenido expresiones con predicados o acciones de agente pertenecientes a esta ontología. La ontología diseñada se mostrará con mayor detalle en la sección 4.4.
- *Sistema de Apoyo al Consenso*: Contiene todo el software necesario para realizar todos los cálculos y operaciones propios del modelo de consenso explicado en el Capítulo 2. Destacar que este sistema ha sido realizado con el objeto de utilizar medidas “suavizadas” del consenso, ajustándose así a enfoques como el de *soft consensus* de Kacprzyk.
- *Base de Datos*: Contiene información sobre los procesos de alcance de consenso llevados a cabo por los agentes.

La Figura 4.1 muestra de forma gráfica la arquitectura multi-agente diseñada, con cada uno de los componentes anteriormente mencionados y las formas de comunicación entre estos.

Dada la gran importancia de los agentes específicos de COMAS diseñados en este trabajo

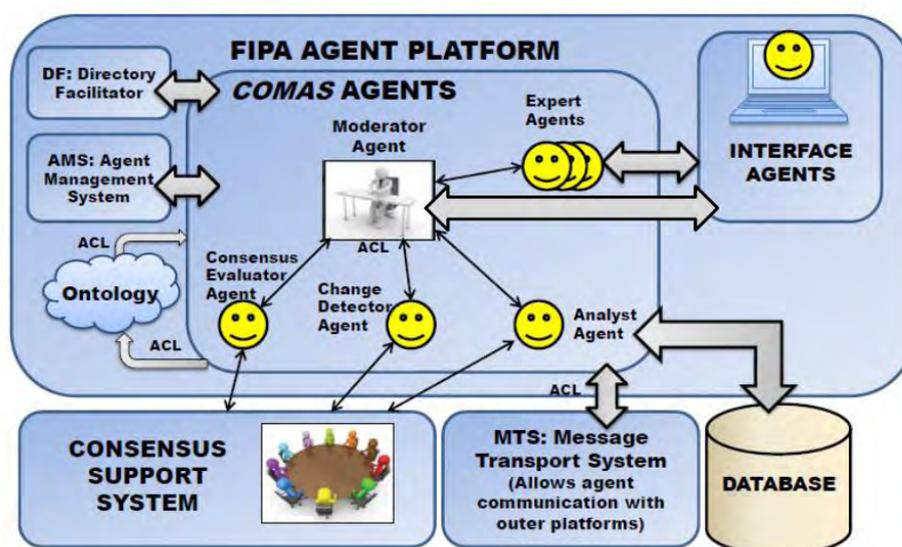


Figura 4.1: Arquitectura de COMAS

investigador, en los siguientes sub-apartados analizaremos en mayor detalle aquellos agentes dedicados a realizar tareas relacionadas con el propio proceso de alcance de consenso: agente moderador, agente experto, agente evaluador de consenso y agente identificador de cambios.

4.2.1. Agente Moderador

El agente moderador supone el eje central de nuestro sistema multi-agente, ya que además de emular a la figura del moderador humano en procesos de consenso reales, es el encargado de mediar todos los actos comunicativos entre el resto de agentes que componen el sistema. Al igual que en los procesos de consenso reales, solamente existe un agente moderador en un proceso de alcance de consenso en COMAS. Sus principales funciones son las siguientes:

- *Recibir la información relativa a un nuevo problema a resolver.*
- *Localización de agentes para participar en un proceso de consenso:* El agente moderador accede al servicio de páginas amarillas proporcionado por el agente *DF* para encontrar

a todos los agentes expertos, evaluadores de consenso e identificadores de cambios existentes en el sistema.

- *Llamada a la Participación en el Consenso*: Una vez detectados todos los agentes disponibles para iniciar un proceso de consenso, el agente moderador envía una propuesta de participación a cada uno de ellos, indicándoles el problema a tratar y las alternativas disponibles. Es imprescindible la participación de un agente evaluador de consenso, un agente identificador de cambios y al menos tres agentes expertos para participar. Cada agente experto es responsable de decidir si participa o no en el problema.

 - *Petición de preferencias*: Al inicio de cada ronda, el agente moderador debe de solicitar a cada uno de los agentes expertos las preferencias sobre el conjunto de alternativas. En la segunda y sucesivas rondas (en su caso), esta petición viene acompañada por las recomendaciones de cambio sugeridas para cada agente experto.

 - *Solicitar el cálculo del grado de consenso*: El agente moderador recopila todas las preferencias recibidas por los agentes expertos junto con todos los datos y parámetros necesarios para calcular el nivel de acuerdo y se los proporciona al agente evaluador de consenso, que se encargará de realizar las operaciones necesarias sobre el modelo de consenso implementado.

 - *Control del consenso*: En función del nivel de consenso actual y del umbral mínimo exigido, el agente moderador decide si se ha alcanzado un acuerdo o no.

 - *Petición de recomendaciones de cambio*: En caso de no haber alcanzado un acuerdo, el agente moderador comunica al agente identificador de cambios la intención de obtener una serie de sugerencias de cambio para los expertos cuyas opiniones están más alejadas del consenso global.
-

4.2.2. Agente Experto

La figura del agente experto supone la automatización de algunas de las tareas llevadas por un experto humano en un problema real de TDG, como son la expresión de preferencias y la aceptación o rechazo de las sugerencias de cambio sobre las mismas. Existen tantos agentes expertos como expertos queramos que participen en un problema, aunque es posible que no todos participen finalmente en él, debido a su capacidad de decidir si aceptar o no la propuesta de participación del agente moderador.

A continuación explicamos las principales responsabilidades asumidas por un agente experto:

- *Decidir participación en el problema:* Tras recibir una propuesta del agente moderador con la descripción de un problema y las alternativas consideradas, el agente experto decide si participa o no en el problema. Esta acción es usualmente llevada a cabo por el experto humano a través de la interfaz proporcionada por el agente interfaz.
 - *Expresión de Preferencias:* Al inicio de cada ronda, tras recibir la petición por parte del agente moderador, el agente experto recopila su opinión sobre las alternativas en forma de una relación de preferencia, y se las proporciona al agente moderador. La interfaz del usuario experto proporciona los medios necesarios para que este pueda introducir sus preferencias iniciales. Destacar que antes de que el problema dé comienzo, es el propio experto humano quien debe de expresar las preferencias a través de la interfaz, siendo esta tarea responsabilidad del agente experto a partir de ese momento.
 - *Realizar cambios en las valoraciones:* Ocasionalmente, un agente experto puede recibir una o varias sugerencias de cambio sobre alguna de sus valoraciones, indicándole cuál de ellas debe modificar, así como la dirección de cambio (incrementar o decrementar). Por defecto, todos los agentes expertos aceptan los cambios sugeridos, pero también es posible modelar el comportamiento de cada uno de ellos [24], de forma que unos agentes posean una mayor probabilidad de obedecer a los cambios que otros, así como
-

un grado diferente de incremento o decremento, actuando así de forma más cercana a como lo harían diferentes expertos humanos en la vida real.

4.2.3. Agente Evaluador de Consenso

Este agente supone una especialización del moderador humano en procesos reales, ya que se encarga de parte de las tareas que en principio debería asumir el agente moderador, más concretamente, aquellas relativas a la obtención del grado de consenso en cada ronda. El agente evaluador de consenso interactúa con el sistema de apoyo al consenso implementado, realizando las operaciones conducentes a la obtención del nivel de acuerdo alcanzado por los expertos, a partir de las opiniones proporcionadas por estos a través del agente moderador. Los principales pasos a seguir por este agente para cumplir su objetivo son los siguientes:

- Obtener las similitudes entre cada par de expertos a partir de las preferencias de estos.
- Calcular la matriz de consenso en base a las similitudes obtenidas.
- Determinar el grado de consenso a nivel de alternativa y el grado de consenso global en cada ronda, y enviarlos al agente moderador.

4.2.4. Agente Identificador de Cambios

El agente identificador de cambios también asume parte de las responsabilidades de las que un moderador humano tendría que encargarse en procesos reales. En este caso, su labor consiste en obtener un conjunto de recomendaciones o sugerencias de cambio para los expertos. Por ello, el agente identificador de cambios únicamente debe llevar a cabo su labor cuando reciba una petición del agente moderador tras haber obtenido un grado de consenso insuficiente. Al igual que el agente evaluador de consenso, este agente también tiene acceso al modelo de consenso implementado para realizar todas las operaciones necesarias sobre él.

Para cumplir su objetivo, el agente identificador de cambios realiza las siguientes operaciones sobre el modelo:

- Identificar las valoraciones (pares de alternativas) en las que no existe un nivel de acuerdo suficiente.
- Para cada valoración identificada en el paso anterior, determinar aquellos agentes expertos más alejados de la opinión de la mayoría, es decir, los agentes expertos que contribuyen en menor grado a lograr un consenso sobre la valoración.
- Generar una recomendación de cambio para cada experto y par de alternativas identificados, y asignarle una dirección de cambio (incrementar valoración o decrementar valoración).
- Devolver todas las recomendaciones generadas al agente moderador.

Como hemos podido ver al estudiar la arquitectura multi-agente propuesta, un aspecto fundamental en nuestro sistema es el de hacer posible una comunicación efectiva entre los diferentes agentes diseñados. Este aspecto será tratado con mayor profundidad en la siguiente sección.

4.3. Comunicación entre Agentes

Los agentes en COMAS emplean los protocolos de comunicación de FIPA para intercambiar mensajes ACL con información relevante sobre el problema. En particular, nuestros agentes utilizan los protocolos *Propose* y *Request*, cuyo funcionamiento fue estudiado en la Sección 3.3.3.

En los siguientes sub-apartados conoceremos los principales actos comunicativos entre los agentes desarrollados, prestando especial atención a los comportamientos (*behaviours*), llevados a cabo para modelar dichos actos de comunicación. Un comportamiento o *behaviour*

es el nombre que recibe cada uno de los procesos o tareas que puede llevar a cabo un agente particular. La implementación de estos procesos es responsabilidad del programador, no obstante JADE proporciona modelos genéricos o tipos de comportamiento, a partir de los cuales definiremos normalmente nuestros comportamientos particulares.

4.3.1. Agente Moderador - Agente Experto

1. El agente moderador comienza su interacción con un agente experto enviándole un mensaje *ACL-Propose* invitándole a participar en el problema de consenso. Esto se hace mediante un comportamiento de tipo *ProposeInitiator*, llamado *CallForConsensus*.
 2. A través del comportamiento *CallForConsensusAnswer*, de tipo *ProposeResponder*, el agente experto recibe y analiza la propuesta enviada por el agente moderador, y le responde aceptando (mensaje *ACL-AcceptProposal*) o rechazando (mensaje *ACL-RejectProposal*) la propuesta.
 3. Si el agente experto aceptó la propuesta, el agente moderador le envía un mensaje *ACL-Request* para solicitarle las preferencias sobre el problema. La implementación necesaria para esta petición se realiza mediante los comportamientos *AssessmentRequest* y *AssessmentRequestHandler*, de tipo *AchieveREInitiator*.
 4. El agente experto debe atender la petición de valoraciones y, en primer lugar, responder al agente moderador indicándole si acepta (mensaje *ACL-Agree*), rechaza (mensaje *ACL-Refuse*) o no entiende la petición (mensaje *ACL-NotUnderstood*). En caso de haber aceptado, el agente experto envía un segundo mensaje de respuesta, de tipo *ACL-Inform*, que contiene sus preferencias sobre el problema.
 5. Se repiten los pasos 3 y 4 para cada ronda de consenso. En la segunda y siguientes rondas, el mensaje de petición de valoraciones por parte del agente moderador incluirá las recomendaciones de cambio sugeridas al agente experto.
-

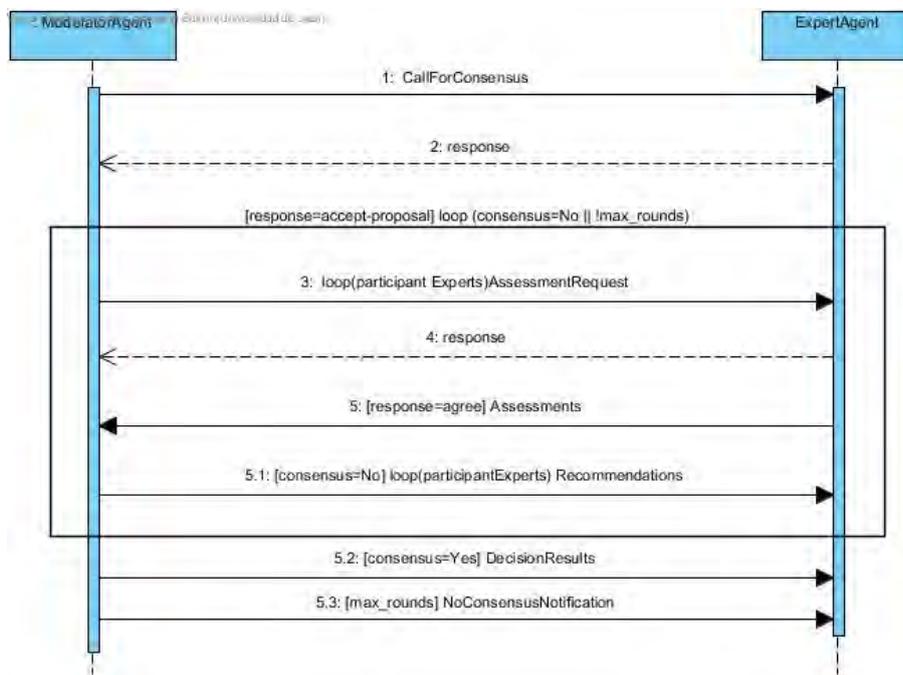


Figura 4.2: Comunicación entre agente moderador y agente experto

6. Una vez se ha llegado al final del proceso, el agente moderador notifica de haber alcanzado consenso, o bien de haberse superado el límite de rondas, en su caso.

La Figura 4.2 muestra un diagrama de secuencia con las principales etapas del proceso comunicativo entre estos dos agentes.

4.3.2. Moderador-Evaluador de Consenso

1. El agente moderador comienza su interacción con el agente evaluador de consenso enviándole un mensaje *ACL-Propose* invitándole a participar en el problema de consenso, mediante un comportamiento de tipo *ProposeInitiator*, llamado *CallForConsensus*.
2. A través del comportamiento *CallForConsensusAnswer*, de tipo *ProposeResponder*, el agente evaluador de consenso recibe y analiza la propuesta enviada por el agente moderador, y le responde aceptando (*ACL-AcceptProposal*) o rechazando (*ACL-*

RejectProposal) la propuesta.

3. El agente moderador envía un mensaje *ACL-Request* para solicitar al agente evaluador de consenso el nivel de acuerdo en la ronda actual según las preferencias de los expertos. La implementación necesaria para esta petición se realiza mediante los comportamientos *ConsensusCalcRequest* y *ConsensusCalcRequestHandler*, de tipo *AchieveREInitiator*.
4. El agente evaluador de consenso debe atender la petición de valoraciones y, en primer lugar, responder al agente moderador indicándole si acepta (mensaje *ACL-Agree*), rechaza (mensaje *ACL-Refuse*) o no entiende la petición (mensaje *ACL-NotUnderstood*). En caso de haber aceptado, el agente evaluador de consenso envía un segundo mensaje de respuesta, de tipo *ACL-Inform*, con los resultados de determinar el nivel de acuerdo en la ronda actual.
5. Se repiten los pasos 3 y 4 para cada ronda de consenso.
6. Una vez se ha llegado al final del proceso, el agente moderador notifica de haber alcanzado consenso, o bien de haberse superado el límite de rondas, en su caso.

En la Figura 4.3 podemos apreciar los pasos del proceso comunicativo llevado a cabo entre estos dos agentes.

4.3.3. Moderador-Identificador de Cambios

1. El agente moderador comienza su interacción con el agente identificador de cambios enviándole un mensaje *ACL-Propose* invitándole a participar en el problema de consenso, mediante un comportamiento de tipo *ProposeInitiator*, llamado *CallForConsensus*.
 2. A través del comportamiento *CallForConsensusAnswer*, de tipo *ProposeResponder*, el agente identificador de cambios recibe y analiza la propuesta enviada por el agen-
-

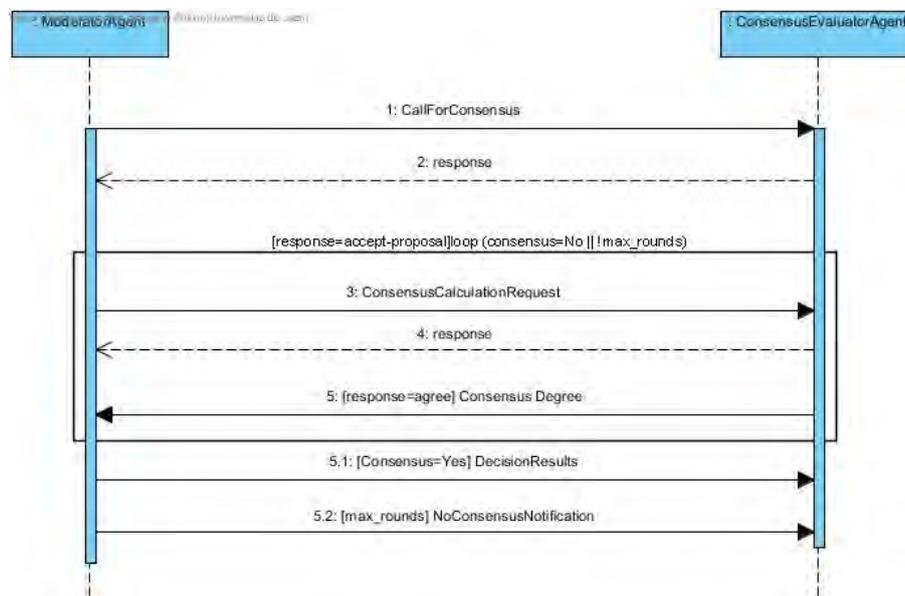


Figura 4.3: Comunicación entre agente moderador y agente evaluador de consenso

te moderador, y le responde aceptando (*ACL-AcceptProposal*) o rechazando (*ACL-RejectProposal*) la propuesta.

3. Si en la ronda actual no se ha alcanzado consenso, el agente moderador envía un mensaje *ACL-Request* para solicitar al agente identificador de cambios una serie de recomendaciones de cambio para los expertos y pares de alternativas más alejados del consenso. La implementación necesaria para esta petición se realiza mediante los comportamientos *RecommendationRequest* y *RecommendationRequestHandler*, de tipo *AchieveREInitiator*.
4. El agente identificador de cambios debe atender la petición de valoraciones y, en primer lugar, responder al agente moderador indicándole si acepta (mensaje *ACL-Agree*), rechaza (mensaje *ACL-Refuse*) o no entiende la petición (mensaje *ACL-NotUnderstood*). En caso de haber aceptado, el agente identificador de cambios envía un segundo mensaje de respuesta, de tipo *ACL-Inform*, que contiene las recomendaciones de cambio sugeridas.
5. Se repiten los pasos 3 y 4 para cada ronda de consenso.

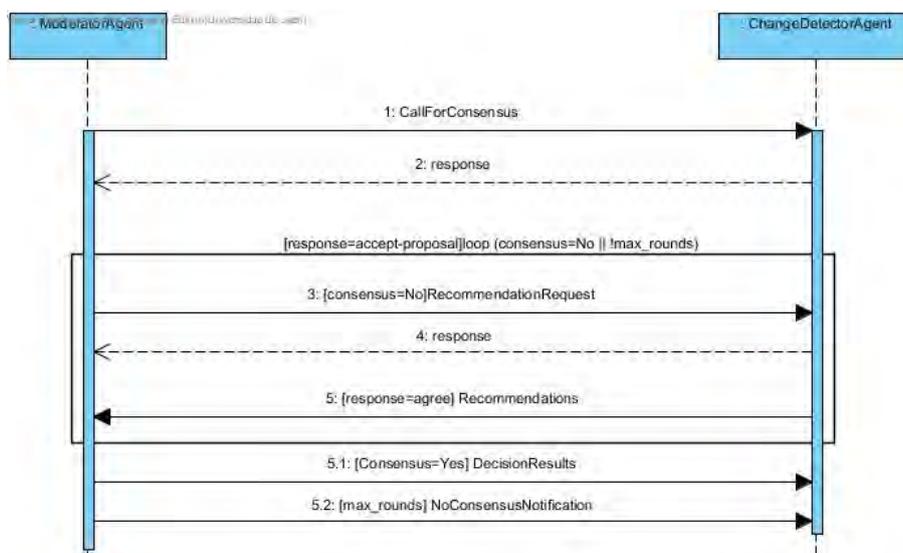


Figura 4.4: Comunicación entre agente moderador y agente identificador de cambios

6. Una vez se ha llegado al final del proceso, el agente moderador notifica de haber alcanzado consenso, o bien de haberse superado el límite de rondas, en su caso.

En la Figura 4.4 podemos apreciar las etapas del acto comunicativo llevado a cabo entre estos dos agentes.

4.3.4. Proceso General de Comunicación entre Agentes

El siguiente diagrama de secuencia representado en la Figura 4.5 muestra, de forma resumida, el proceso completo de comunicación entre todos los agentes propios de COMAS durante un proceso de consenso. En este diagrama se puede apreciar además cómo un agente interfaz se encarga de instanciar cada uno de los agentes de nuestro sistema, además del uso del *DF* por parte del agente moderador para localizar al resto de agentes en la plataforma.

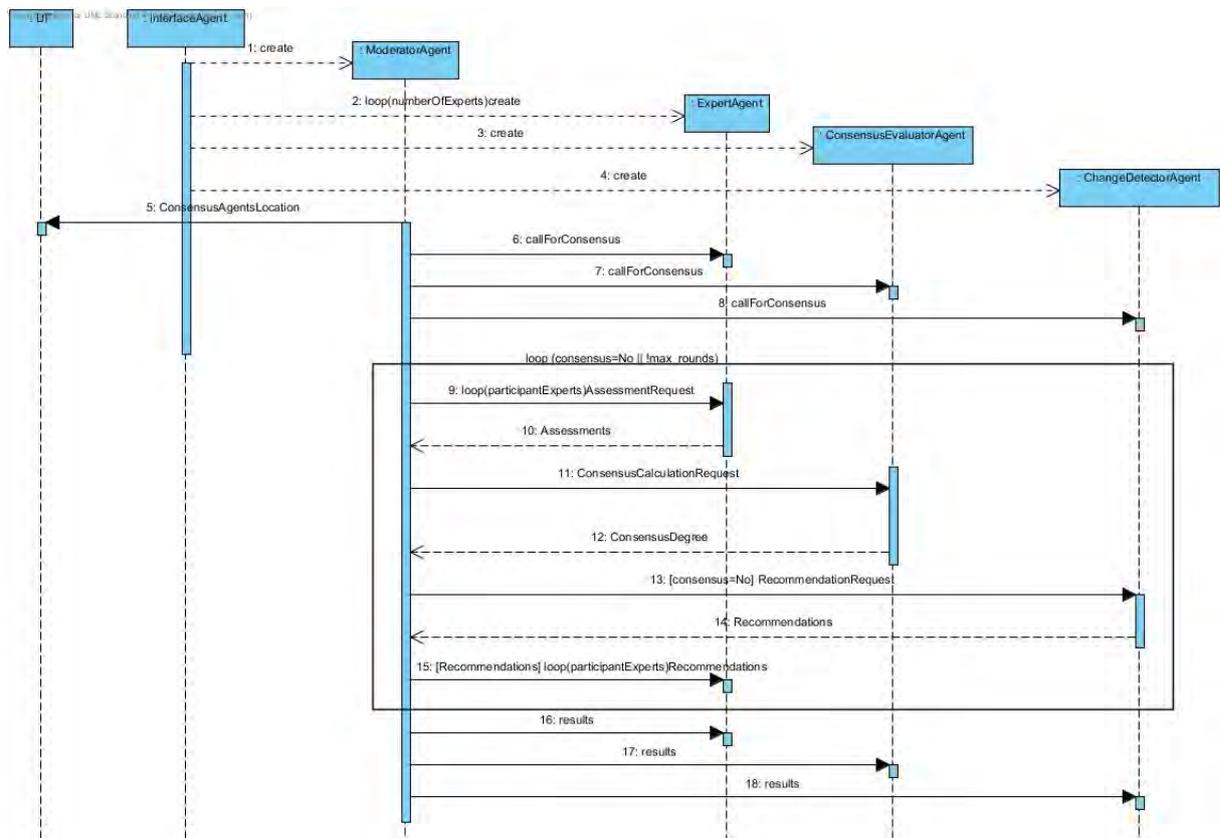


Figura 4.5: Proceso general de comunicación entre agentes

4.4. Ontología en COMAS

En el capítulo 3 sentamos las bases del empleo de ontologías en SMA, estudiando su estructura y algunas aplicaciones reales de las mismas. En esta sección conoceremos la ontología diseñada para nuestro sistema, así como la forma en que los agentes la utilizan.

Desde un punto de vista ontológico, COMAS está basado en la idea propuesta por Kacprzyk y Zadrozny en [56], donde se definen dos ontologías diferentes:

- (I) Una *Ontología basada en el dominio*, en la que son representados los conceptos generales para llevar a cabo procesos de consenso.
- (II) Una *Ontología basada en la aplicación*, en la que se definen conceptos, predicados y acciones de agente relativos al problema específico considerado, es decir, se trata de una ontología dependiente de cada problema a abordar.

La ontología finalmente diseñada en nuestro trabajo es una única ontología inspirada en gran parte de los componentes encontrados en estas dos ontologías, ya que considera tanto los conceptos necesarios para poder llevar a cabo un proceso de alcance de consenso completo como aquellos que los agentes deben utilizar para razonar sobre el conocimiento relativo al problema.

La estructura y componentes de nuestra ontología aparecen representados en la Figura 4.6. A continuación explicaremos con mayor detalle cada uno de estos componentes.

Conceptos

- **AID**: Concepto abstracto que contiene la información básica para identificar a un agente. Contiene entre otros elementos de información el nombre de un agente y su dirección. Actualmente se utiliza como un atributo del concepto *ExpertAgent*, aunque el diseño de la ontología queda abierto para su uso por otros tipos de agente.
-

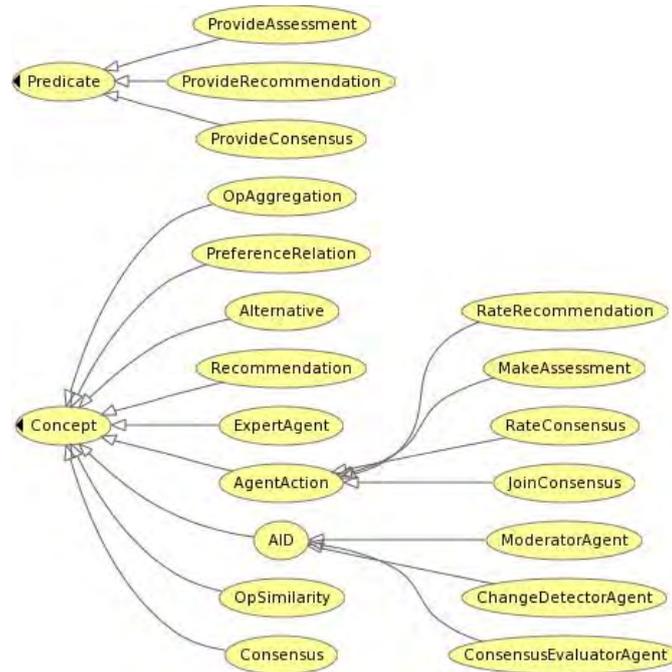


Figura 4.6: Ontología de COMAS

- **Alternative:** Representación de una de las alternativas para el problema de TDG. Estos elementos forman parte de la descripción general del problema que el agente moderador debe proponer al resto de agentes. Se compone de los siguientes atributos:
 - *description*: Atributo de tipo cadena. Contiene la descripción de la alternativa.
 - **Consensus:** Contiene información sobre el nivel de consenso alcanzado en un momento dado. Consta de los atributos siguientes:
 - *currentConsensus*: Atributo de tipo real, que contiene el grado de consenso actual.
 - *consensusMatrix*: Atributo múltiple (agregado) de tipo real. Contiene los elementos de la matriz de consenso obtenida en la ronda actual.
 - *consensusThreshold*: Atributo de tipo real. Contiene el umbral de consenso.
 - **ExpertAgent:** Este concepto se refiere a uno de los agentes expertos participantes en el consenso. Se compone del siguiente atributo:
-

- *aid*: Atributo de tipo concepto *AID*. Identificador del agente experto.
 - **OpAggregation**: Operador de agregación que se utilizará durante el proceso de consenso. Se compone del siguiente atributo:
 - *aggOperator*: Atributo de tipo cadena, que indica el nombre del operador de agregación a utilizar.
 - **OpSimilarity**: Especifica el operador de similitud con el que se trabajará durante el proceso de consenso. Se compone de un atributo:
 - *simOperator*: Atributo de tipo cadena. Contiene el nombre del operador de similitud.
 - **PreferenceRelation**: Este concepto tiene como objetivo representar una relación de preferencia completa, que como sabemos es la estructura empleada por un experto para representar sus opiniones sobre un conjunto de alternativas. Contiene el siguiente atributo:
 - *assessments*: Atributo agregado de tipo real. Contiene el conjunto de valoraciones dadas por el experto a cada par de alternativas.
 - **Recommendation**: Sirve para representar una recomendación de cambio individual sugerida a un experto para que este modifique sus preferencias en vista a alcanzar un acuerdo. Consta de los siguientes atributos:
 - *aid*: Atributo de tipo concepto *AID*. Identificador del agente experto a quien es destinada la recomendación.
 - *numAlt1*: Atributo de tipo entero. Número de la primera alternativa existente en el par a modificar.
 - *numAlt2*: Atributo de tipo entero. Número de la segunda alternativa existente en el par a modificar.
-

- *direction*: Atributo de tipo cadena. Dirección en la que el experto debe modificar la valoración, puede tomar los valores “Increase” o “Decrease”.

Acciones de Agente

- **JoinConsensus**: Contiene toda la información necesaria para elaborar una propuesta de participación en un problema de TDG a resolver mediante consenso. Los atributos de que se compone son:
 - *maxRounds*: Atributo de tipo entero. Indicador del número máximo de rondas permitido en el proceso, para dar una idea aproximada de la duración del mismo.
 - *setOfAlternatives*: Atributo agregado de tipo concepto *Alternative*. Conjunto de alternativas existentes en el problema.
 - *problemDescription*: Atributo de tipo cadena. Descripción del problema propuesto.
 - **MakeAssessment**: Predicado utilizado para solicitar las preferencias a los expertos. Se compone de los atributos siguientes:
 - *setOfRecommendations*: Atributo agregado de tipo concepto *Recommendation*. Conjunto de recomendaciones de cambio para los expertos. Este atributo se utiliza únicamente a partir de la segunda ronda, ya que en la primera se solicitan las valoraciones iniciales.
 - *round*: Atributo de tipo entero. Número de ronda actual.
 - **RateConsensus**: Predicado de vital importancia para el buen funcionamiento del sistema. Contiene toda la información de entrada necesaria para determinar el grado de consenso actual, mediante el sistema de apoyo al consenso desarrollado. A continuación describimos cada uno de los atributos que contiene:
 - *maxRounds*: Atributo de tipo entero. Indica el número máximo de rondas permitido.
-

- *numAlt*: Atributo de tipo entero. Indica el número de alternativas existentes en el problema.
 - *opSim*: Atributo de tipo concepto *OpSimilarity*. Operador de similitud empleado.
 - *consensus*: Atributo de tipo concepto *consensus*. Contiene la estructura necesaria para incluir el grado de consenso obtenido, una vez calculado.
 - *opAgg*: Atributo de tipo concepto *opAggregation*. Operador de agregación empleado.
 - *setOfExperts*: Atributo agregado de tipo concepto *ExpertAgent*. Conjunto de expertos que participan en el problema, identificados a través de su agente experto correspondiente.
 - *round*: Atributo de tipo entero. Número de ronda actual.
 - *setOfPreferences*: Atributo agregado de tipo concepto *PreferenceRelation*. Contiene el conjunto de relaciones de preferencia, con las opiniones actuales de todos los expertos.
- **RateRecommendation**: Este predicado contiene la información necesaria para solicitar el cálculo de las recomendaciones de cambio tras un control de consenso con resultado negativo. Contiene los siguientes atributos:
- *numAlt*: Atributo de tipo entero. Número de alternativas existentes.
 - *opSim*: Atributo de tipo concepto *OpSimilarity*. Operador de similitud empleado, de utilidad también en el cálculo de las matrices de proximidad.
 - *consensus*: Atributo de tipo concepto *consensus*. Contiene el grado de consenso a diferentes niveles.
 - *setOfExperts*: Atributo agregado de tipo concepto *ExpertAgent*. Conjunto de expertos que participan en el problema, identificados a través de su agente experto correspondiente.
-

- *setOfPreferences*: Atributo agregado de tipo concepto *PreferenceRelation*. Contiene el conjunto de relaciones de preferencia, con las opiniones actuales de todos los expertos.

Predicados

- **ProvideAssessment**: Este predicado es utilizado por un agente experto como respuesta al recibir una acción de agente del tipo *MakeAssessment*. Consta de un atributo:
 - *preference*: Atributo de tipo concepto *PreferenceRelation*. Consiste en la relación de preferencia actual proporcionada por el experto.
 - **ProvideConsensus**: Predicado que proporciona el grado de consenso recién obtenido en la ronda actual. Incluye el siguiente atributo:
 - *consensus*: Atributo de tipo concepto *Consensus*. Grado de consenso obtenido en la ronda actual.
 - **ProvideRecommendation**: Este predicado proporciona el conjunto de recomendaciones de cambio generadas en cada ronda. Los atributos de que se compone son:
 - *setOfExperts*: Atributo agregado de tipo concepto *ExpertAgent*. Contiene el conjunto de expertos a los que se les debe sugerir recomendaciones.
 - *setOfRecommendations*: Atributo agregado de tipo concepto *Recommendation*. Contiene las recomendaciones generadas en la actual ronda, ordenadas según el experto a quien van dirigidas.
 - *numRecommendationsByExpert*: Atributo agregado de tipo entero. Indica cuántas de las recomendaciones en el atributo anterior pertenecen a cada experto. Dado que tanto este atributo como *setOfExperts* mantienen una relación de orden entre sí, se consigue de esta manera enviar a cada experto sus recomendaciones correspondientes.
-

Por último, tras analizar en profundidad la estructura de nuestra ontología, veremos en qué etapas del proceso son utilizados cada predicado y acción de agente.

1. Acción *JoinConsensus*: El agente moderador utiliza esta acción como el contenido de un mensaje ACL-Propose para invitar al resto de agentes a participar en el problema, en el comportamiento *CallForConsensus*. Un ejemplo de instancia de esta acción es el siguiente:

```
(JoinConsensus :maxRounds 10 :setOfAlternatives (sequence
(Alternative :description Marqués de Cáceres)
(Alternative :description Los Molinos)
(Alternative :description Viña Mayor)
(Alternative :description René Barbier))
:problemDescription "Spanish Wine Selection Problem")
```

2. Acción *MakeAssessment*: Esta acción es utilizada como contenido de un mensaje ACL-Request en el comportamiento *AssessmentRequest*, del agente moderador. Sus destinatarios son una serie de agentes expertos que deben decidir si aceptar o no una petición de sus preferencias para el agente moderador. Recordemos que esta acción de agente es utilizada en la segunda y siguientes rondas para proporcionar a los agentes expertos las recomendaciones de cambio, por lo que en la primera ronda utilizamos una recomendación “vacía” (referida a un par de alternativas ficticio, (0,0)) para indicar que no existen aún recomendaciones. A continuación mostramos un ejemplo de instancia de este predicado:

```
(MakeAssessment :setOfRecommendations (sequence
(Recommendation :numAt12 0 :numAlt1 0 :aid (agent-identifier :name "")
:direction Increase)) :round 1)
```

3. Predicado *ProvideAssessment*: Este predicado supone la respuesta de un agente experto a la acción *MakeAssessment*, cuando este acepta proporcionar sus preferencias al

moderador. Se encapsula como el contenido de un mensaje ACL-*Inform* en el comportamiento *MakeAssessmentHandler*. Su estructura es similar a la del siguiente ejemplo, en el que un agente experto asociado al experto e_i expresa sus preferencias p_i^{lk} sobre 4 alternativas (nótese que el predicado consta de 12 elementos en lugar de 16, al excluir los elementos vacíos de la diagonal principal en la correspondiente relación de preferencia):

```
((ProvideAssessment (PreferenceRelation :assessments (sequence
0.6725 0.1725 0.9375
0.32875 0.5 0.32875
0.6725 0.5 0.32875
0.0625 0.82875 0.5))))
```

4. Acción *RateConsensus*: Tras recibir las preferencias por parte de los agentes expertos, el agente moderador envía un mensaje ACL-*Request* al agente evaluador de consenso, cuyo contenido es una instancia de esta acción de agente. Esto se hace en el comportamiento *ConsensusCalcRequest* del agente moderador. En el siguiente ejemplo mostramos parte del contenido de esta acción.

```
(RateConsensus :maxRounds 10 :numAlt 4 :opSim (OpSimilarity
:simOperator Euclidean) :consensus (Consensus :currentConsensus
0.7089 :consensusMatrix (sequence 0.0) :consensusThreshold 0.75)
:opAgg (OpAggregation :aggOperator WeightedMean) :setOfExperts
(sequence (ExpertAgent :aid <IDENTIFICADORES DE LOS AGENTES EXPERTOS>
)) :round 3 :setOfPreferences (sequence (PreferenceRelation
:assessments (sequence 0.18375000000000002 0.18375000000000002 0.5025
0.7175 0.5525 0.7175 0.7525 0.28375000000000006 0.2525
0.5025 0.35250000000000004 0.6525))
<RESTO DE PREFERENCIAS DE EXPERTOS>))
```

5. Predicado *ProvideConsensus*: Se utiliza como contenido del mensaje ACL-*Inform* con el
-

que el agente evaluador de consenso responde al agente moderador, una vez calculado el grado de consenso en la actual ronda. Este predicado es creado en el comportamiento *ConsensusCalcRequestHandler*, y su contenido sigue la estructura mostrada a continuación:

```
((ProvideConsensus (Consensus :currentConsensus 0.7631399
:consensusMatrix (sequence
-1.0 0.728705357142857 0.7678571428571429 0.758169642857143
0.7615624999999999 -1.0 0.7011160714285712 0.8450892857142857
0.7582589285714286 0.7386160714285717 -1.0 0.7760267857142856
0.7270982142857145 0.8029017857142858 0.7922767857142857 -1.0)
:consensusThreshold 0.75)))
```

6. Acción *RateRecommendation*: Si durante la etapa de control del grado de consenso, el agente moderador determina que este no es suficiente, deberá preparar un mensaje ACL-Request destinado al agente identificador de cambios con la información necesaria para que este le devuelva un conjunto de sugerencias de cambio. El contenido de dicho mensaje es una instancia de esta acción de agente, la cual se crea y prepara para su envío en el comportamiento *RecommendationRequest*. Veamos un ejemplo resumido de esta acción de agente:

```
(RateRecommendation :numAlt 4 :opSim (OpSimilarity
:simOperator EuclideanDistance) :consensus (Consensus
:currentConsensus 0.7089658 :consensusMatrix (sequence
<VALORES DE LA MATRIZ DE CONSENSO> )
:consensusThreshold 0.75) :setOfExperts
<IDENTIFICADORES DE LOS AGENTES EXPERTOS>
:setOfPreferences <PREFERENCIAS DE LOS EXPERTOS> )
```

7. Predicado *ProvideRecommendation*: Este último predicado supone el contenido del mensaje ACL-Inform que el agente identificador de cambios da como respuesta al
-

agente moderador, e incluye las recomendaciones de cambio sugeridas a los expertos. Es creado en el comportamiento *RecommendationRequestHandler* y su estructura es análoga a la del siguiente ejemplo, el cual aparece resumido debido a que este predicado puede llegar a tener una gran extensión:

```
((ProvideRecommendation (sequence 7 4 3 5 6 4 2 3)
(sequence (Recommendation :numAt12 1 :numAlt1 0
:aid <IDENTIFICADOR DEL AGENTE EXPERTO>
:direction Increase) <RESTO DE RECOMENDACIONES>)
(sequence (ExpertAgent <...>) <AGENTES EXPERTOS>)))
```


Simulación de Uso de la Plataforma en COMAS

Una vez estudiada la arquitectura multi-agente y las principales características de nuestro sistema, en este quinto capítulo de la memoria de investigación pretendemos realizar una simulación de uso de la plataforma multi-agente desarrollada, mediante la resolución de problemas de consenso en los que utilizaremos distintas *medidas de consenso*, en las cuales usaremos una serie de *medidas de similitud* y *operadores de agregación* diferentes [7, 17, 19], para llevar a cabo la resolución del problema planteado.

Por esta razón, el capítulo plantea en primer lugar el problema a resolver mediante un proceso de consenso, para a continuación mostrar una revisión de las distintas medidas, operadores y demás parámetros que utilizaremos en nuestra simulación. Seguidamente, mostramos el resultado de la resolución del problema de consenso bajo las diferentes medidas consideradas y, para finalizar, emitimos una serie de guías de actuación a tener en cuenta al utilizar COMAS en un problema determinado, según el objetivo del mismo.

Destacar que la alta escalabilidad de la plataforma multi-agente en COMAS nos permite simular procesos de consenso con un elevado número de expertos de forma rápida y realista, lo cual ha hecho posible llevar a cabo el objetivo que nos planteamos en este capítulo.

5.1. Características del Estudio

El principal objetivo de la simulación a realizar es el de determinar el funcionamiento de la plataforma bajo la resolución de diferentes problemas de consenso, en los que se emplearán distintas medidas de consenso.

En esta sección introducimos las principales características de las pruebas que realizaremos sobre nuestra plataforma mediante simulación. Para ello, describiremos en primer lugar las características del problema que proponemos resolver mediante consenso. Seguidamente, revisaremos las diferentes medidas de consenso que se considerarán para la resolución del problema y explicaremos en qué consistirán las pruebas a realizar.

5.1.1. Descripción del Problema

En este apartado, planteamos el problema de consenso cuyo estudio y simulación supone el principal objetivo de este capítulo. Como veremos, el problema planteado es un ejemplo de problema frecuente en situaciones reales en todo tipo de comunidades, sociedades y organizaciones, normalmente resuelto mediante criterios tradicionales de resolución de problemas de TDG, como es el de tomar la decisión por *mayoría*. Debido a que este modo de resolver este tipo de problemas suele causar cierta controversia y la no aceptación de la solución por parte de algunos individuos, proponemos el uso de COMAS para llevar a cabo un proceso de consenso, para lograr un acuerdo mutuo a través de la discusión y modificación de las preferencias de los expertos, antes de tomar la decisión final.

El enunciado del problema es el siguiente:

En una comunidad de vecinos, se pretende llevar a cabo un cambio en las cuotas de la misma. Ante esto, el conjunto de posibles alternativas $X = \{x_1, \dots, x_4\}$, se compone de las cuatro propuestas siguientes:

- x_1 : Incrementar los gastos corrientes de la comunidad en un 10 %.
- x_2 : Aumentar en un 20 % la parte destinada a gastos de mantenimiento.
- x_3 : Rebajar cuotas en empleados que prestan servicios comunitarios.
- x_4 : Instalar un nuevo ascensor y mantener el resto de servicios tal y como están.

La comunidad se compone de cien vecinos, $E = \{e_1, \dots, e_{100}\}$, los cuales deben llegar a un acuerdo sobre las opiniones que cada uno de ellos tiene sobre estas cuatro propuestas, antes de tomar una decisión posterior en conjunto.

Como podemos ver, este problema y cualquier otro problema similar de índole real requiere de un número de expertos bastante elevado. Las hipótesis de las que partimos para la aplicación de este problema de consenso en nuestro sistema son las siguientes:

- Los vecinos deben de expresar sus opiniones sobre un conjunto de 4 alternativas posibles, $X = \{x_1, \dots, x_4\}$. Para ello, cada experto e_i utilizará una relación de preferencia difusa,

$$P_i = \begin{pmatrix} - & \dots & p_i^{14} \\ \vdots & \ddots & \vdots \\ p_i^{41} & \dots & - \end{pmatrix}$$

donde, mediante valoraciones p_i^{lk} en el intervalo $[0,1]$, deben expresar el grado de preferencia de la alternativa x_l respecto a la alternativa x_k .

- Tenemos un conjunto formado por 100 expertos participantes, $E = \{e_1, \dots, e_{100}\}$, para cada uno de los cuales establecemos una relación de preferencia consistente en *valoraciones aleatorias* al comienzo del proceso, siempre manteniendo la relación de reciprocidad $p_i^{lk} = 1 - p_i^{kl}$, $k \neq l$ en la relación de preferencia correspondiente.

Gracias a la alta escalabilidad y capacidad de cómputo de nuestra plataforma multi-agente, se hace posible la simulación de procesos de consenso con un número de expertos elevado, en contraposición a los ejemplos habituales de resolución de problemas de consenso mostrados en la literatura, donde se suele emplear un número de expertos reducido (entre 4 y 8 expertos).

- Los expertos aceptan y siguen todas las recomendaciones de cambio sugeridas por el moderador a lo largo del proceso. Para ello, si se sugiere al experto e_i incrementar su valoración p_i^{lk} sobre el par de alternativas (x_l, x_k) , este aumentará dicho valor en 0.1. Por contra, si se le sugiere decrementar dicha valoración, disminuirá su valor en 0.1.
- Todos los expertos se consideran igualmente importantes en lo que a sus opiniones se refiere.
- Consideramos que existe acuerdo suficiente cuando el grado de consenso es igual o superior a un umbral $\mu = 0.85$.
- El número máximo de rondas de consenso permitidas en todas las simulaciones es 10.
- El cálculo del grado de consenso se realiza según una visión *suavizada* del mismo, como se explicó en la sección 2.4.

Como veremos en la sección 5.1.5, las pruebas que realizaremos en este estudio consistirán en resolver el problema que acabamos de plantear bajo el uso de diferentes medidas de consenso, para lo cual se emplearán diferentes combinaciones de medidas de similitud y operadores de agregación en el proceso. Por esta razón, en los siguientes apartados realizamos una revisión teórica de dichas medidas.

5.1.2. Medidas de Consenso

En los procesos de consenso, surge de forma natural la cuestión de cómo medir cuán cercanas se encuentran las opiniones de los expertos entre sí, de cara a obtener el grado de consenso o acuerdo entre ellos. Las medidas de consenso se definen con este objetivo, y suponen un indicador para evaluar cómo de lejos se encuentra un grupo de expertos de un acuerdo unánime [59].

Tradicionalmente, en los procesos de consenso se han utilizado medidas de consenso rígidas [17]. Estas medidas pueden tomar únicamente los valores $\{0,1\}$, valiendo 0 cuando

el acuerdo alcanzado es nulo o parcial, y 1 cuando cuando existe un acuerdo total. Estas medidas presentan la clara desventaja de que el acuerdo total o unánime es difícil o imposible de conseguir en situaciones reales y, en muchos de los casos en que pudiese conseguirse, supondría un coste inaceptable.

Por otra parte, la idea de *consenso suavizado* dio lugar en la práctica a medidas de consenso más flexibles, las cuales se utilizan para evaluar el nivel de acuerdo considerando un rango de posibles niveles de acuerdo parciales entre expertos, en el intervalo unitario $[0,1]$. Dentro de esta nueva visión de consenso, el enfoque basado en mayoría lingüística difusa propuesto por Kacprzyk en [49] permitió introducir el concepto de *soft consensus*, que establece que existe consenso cuando la mayoría de expertos participantes en el problema está satisfecha con las alternativas más importantes del mismo. Las medidas de consenso implementadas en COMAS son medidas suavizadas y, como tales, evalúan el grado de consenso alcanzado en cada ronda de discusión mediante valores pertenecientes al intervalo $[0,1]$.

Para conseguir su objetivo, las medidas de consenso deben de determinar el nivel de coincidencia entre las opiniones de los expertos. Esto se consigue gracias a las *medidas de similitud*, las cuales revisaremos a continuación por ser un aspecto fundamental en este estudio. Por otra parte, como pudimos estudiar en el modelo propuesto en el capítulo 2, nuestro sistema mide el grado de acuerdo alcanzado a tres niveles diferentes: par de alternativas, alternativa y relación de preferencia. Estos tres niveles de medidas de consenso hacen necesario el empleo de *operadores de agregación* que, en primer lugar, determinen el grado de consenso a nivel de par de alternativas a partir de los valores de similitud anteriormente obtenidos y, en segundo lugar, permitan obtener a partir de éste el grado de consenso a nivel de alternativa y a nivel de relación de preferencia. Por ello, tras revisar las medidas de similitud consideradas en este estudio, haremos además una breve revisión de los operadores de agregación implementados.

5.1.3. Medidas de Similitud

Una medida de similitud es, de forma general, un indicador del parecido entre dos elementos. Matemáticamente, la similitud viene normalmente condicionada por la distancia entre dos datos o valores. Su utilización en medidas de consenso se debe a que una de las principales actividades a tener en cuenta para la obtención del mismo consiste en medir el parecido entre las opiniones de diferentes expertos. Además, se ajustan al concepto de *consenso suavizado*, debido a que presentan la ventaja de asignar valores comprendidos en el intervalo $[0,1]$ para la similitud entre expertos [17], en lugar de utilizar un enfoque binario, en el que se considera únicamente el hecho de que las opiniones entre expertos sean iguales o distintas entre sí.

Las medidas de similitud se emplean en procesos de consenso para:

- Determinar cómo de alejadas están las opiniones de los expertos entre sí.
- Determinar cómo de alejadas están las opiniones individuales de cada experto de la opinión colectiva.

Las medidas de similitud están estrechamente relacionadas con las medidas de distancia, ya que la similitud entre dos elementos viene determinada por la distancia entre estos. Notaremos a la distancia entre dos elementos x, y pertenecientes a un conjunto X como $d(x, y)$. Asimismo, notaremos a la similitud entre dos elementos x, y pertenecientes a X como $s(x, y)$.

En este apartado vamos a introducir las medidas de similitud implementadas en nuestro sistema y empleadas en este estudio. En general, todas estas medidas de similitud se basan en la comparación de dos valores reales $x, y \in [0, 1]$, ya que se aplican sobre un mismo elemento p_i^{lk} en dos matrices de preferencias de diferentes expertos.

5.1.3.1. Similitud basada en Distancia Euclídea

De forma general, se denomina *Distancia Euclídea* entre dos puntos $A(x_1, y_1)$ y $B(x_2, y_2)$ del plano a la longitud del segmento de recta que une dichos puntos. Se calcula como:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (5.1)$$

En nuestro caso particular, pretendemos calcular la distancia entre dos valoraciones reales, por lo que trabajamos con puntos con una sola coordenada, pertenecientes a la recta real, luego la distancia a aplicar es:

$$d(x_1, x_2) = \sqrt{(x_2 - x_1)^2} \quad (5.2)$$

siendo x_1 y $x_2 \in [0, 1]$ los dos valores cuya similitud queremos calcular. En este caso, la expresión coincide con la diferencia en valor absoluto entre dos números, luego puede simplificarse como:

$$d(x_1, x_2) = |x_2 - x_1| \quad (5.3)$$

En adelante consideraremos la expresión (5.3) como la forma de obtener la distancia $d(x_1, x_2)$.

Finalmente, el valor de similitud se obtendrá mediante la siguiente expresión:

$$s(x_1, x_2) = 1 - d(x_1, x_2) = 1 - |x_2 - x_1| \quad (5.4)$$

Además de esta medida de similitud, proponemos a continuación el empleo de dos nuevas medidas (también relacionadas con la distancia euclídea) para estudiar el comportamiento de nuestro sistema bajo el uso de cada una de ellas. Llamaremos a estas medidas *similitud cuadrática* y *similitud cuadrática mejorada*.

5.1.3.2. Similitud Cuadrática

Esta forma de calcular la similitud guarda una gran relación con la medida basada en distancia euclídea estudiada en el apartado anterior. En este caso, la similitud viene determinada por el cuadrado de la distancia $d(x_1, x_2)$:

$$s(x_1, x_2) = 1 - d(x_1, x_2)^2 \quad (5.5)$$

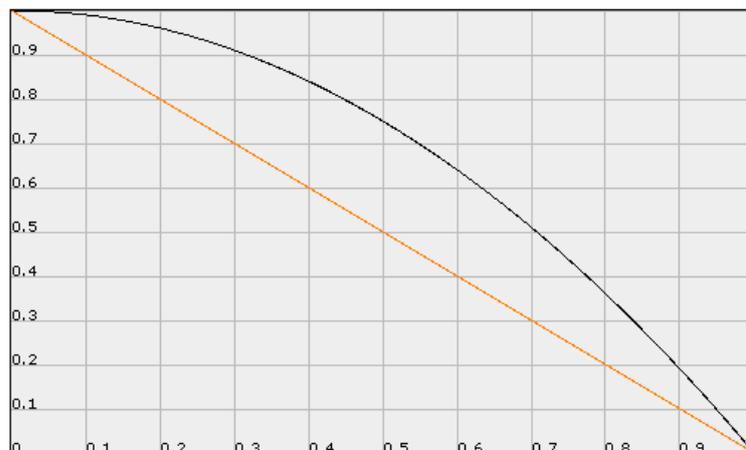


Figura 5.1: Similitud cuadrática

Esta medida presenta la ventaja de propiciar una mayor convergencia al alcance de consenso, al proporcionar mayores valores que la medida anterior. Sin embargo, presenta el inconveniente de ser una medida poco realista, ya que siempre aplica un exceso sobre la similitud real entre dos valores, sin importar cómo de cercanos o alejados estén entre sí.

La figura 5.1 muestra una representación gráfica de $s(x_1, x_2)$ en función de la distancia o diferencia entre los dos valores a comparar, $d(x_1, x_2)$, además de su comparación con la similitud basada en distancia euclídea representada como una recta descendente.

5.1.3.3. Similitud Cuadrática Mejorada

Con el objeto de intentar una mayor convergencia hacia el consenso manteniendo el uso de una medida de similitud realista, propusimos el empleo de una medida cuadrática que solamente aplicase un exceso o “bonificación” sobre aquellos pares de valores cercanos entre sí, así como una “penalización” sobre la similitud entre pares de valores más alejados.

De esta forma surgió la siguiente medida, basada en la función cuadrática por partes

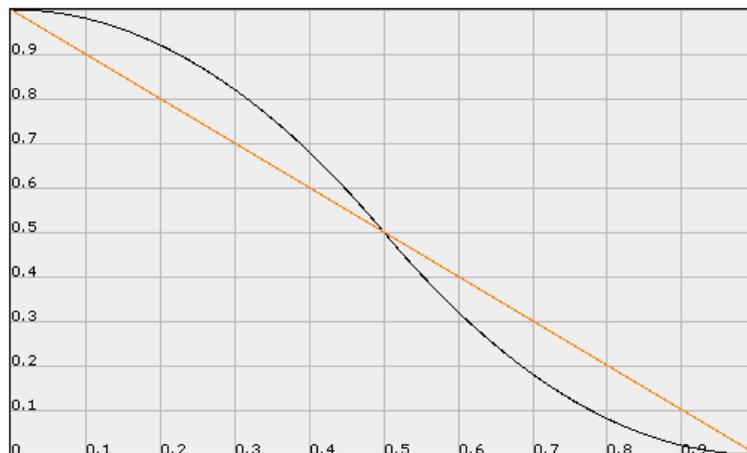


Figura 5.2: Similitud cuadrática mejorada

(para simplificar la expresión, hemos notado $d(x_1, x_2)$ simplemente como d):

$$s(x_1, x_2) = \begin{cases} 1 - 2d^2 & d \leq 0.5 \\ 2(d - 1)^2 & d > 0.5 \end{cases}$$

En la representación gráfica mostrada en la figura 5.2, podemos comparar el comportamiento de esta función con el de la función de similitud basada en distancia euclídea.

5.1.4. Operadores de Agregación

La agregación de información es clave en procesos de toma de decisión [19]. Así, agregar consiste en fusionar un conjunto de elementos de información, obteniendo así un elemento representativo de dicho conjunto.

Existe una gran cantidad de operadores de agregación, de los cuales aquellos comprendidos dentro del rango definido entre el operador máximo (OR) y el operador mínimo (AND), pasando por los operadores basados en medias [61], son de uso común en numerosas disciplinas debido principalmente a su facilidad de uso.

Los operadores de agregación son importantes en los procesos de consenso por varias razones: (i) se utilizan para determinar el grado de consenso a nivel global (relación de

preferencia) alcanzado en cada momento a partir del consenso obtenido para cada par de alternativas, (ii) permiten determinar la opinión colectiva del grupo, y (iii) sirven para medir a nivel colectivo cuan alejados están los expertos de la opinión colectiva. En el primer caso, debemos de tener presente que la mayoría de operadores de agregación que estudiaremos nos permite obtener un valor representativo y flexible del acuerdo alcanzado, dentro del intervalo $[0,1]$. Por ello, suponen una medida acertada de *consenso suavizado*.

En este apartado revisamos algunos de los operadores de agregación existentes en la literatura e implementados por COMAS. De forma general, notaremos la agregación de un conjunto de valores $X = \{x_1, \dots, x_n\}$ como $\phi(x_1, \dots, x_n)$.

Las principales propiedades de los operadores de agregación, presentes cada una de ellas en algunos de los operadores que estudiaremos, son las siguientes:

1. Continuidad: ϕ es una función continua en cada una de sus variables.
2. Conmutatividad: $\phi(x, y) = \phi(y, x), \forall x, y \in X$.
3. Asociatividad: $\phi(x, \phi(y, z)) = \phi(\phi(x, y), z) \forall x, y, z \in X$.
4. Monotonía: Si $x \leq z$, entonces $\phi(x, y) \leq \phi(z, y)$
5. Idempotencia: $\phi(x, \dots, x) = x \forall x \in X$.

Durante la revisión de los diferentes operadores considerados se destacarán cuáles de estas propiedades cumple cada uno de ellos. Los diferentes operadores finalmente implementados se aplicarán durante la simulación solamente en la fase de obtención del grado de consenso a los tres niveles (pares de alternativas, alternativas y relación). Para otras operaciones que precisen de una agregación de valores (como por ejemplo obtener una matriz de proximidad media) empleamos por defecto la media aritmética.

5.1.4.1. Media Aritmética

También conocida como promedio, la media aritmética \bar{x} de un conjunto de valores $X = \{x_1, \dots, x_n\}$ es igual a la suma de todos ellos dividida por el tamaño o cardinalidad del conjunto:

$$\phi(x_1, \dots, x_n) = \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (5.6)$$

Una de las limitaciones de esta medida es que es un operador poco robusto, esto es, se ve fácilmente afectada por valores extremos, ya que valores muy altos tienden a aumentarla, mientras que valores muy bajos tienden a reducirla, razón por la cual en algunos casos puede ser una medida poco representativa del conjunto de valores considerado.

Algunas propiedades particulares cumplidas por este operador son:

- Desviación nula: $\sum_i (x_i - \bar{x}) = 0$
- Dado un valor a , $\sum_i (x_i - a)^2$ es mínimo cuando $a = \bar{x}$.
- Si a todos los valores x_i se les suma una misma cantidad, la media aritmética queda aumentada en esa misma cantidad.
- Si todos los valores x_i se multiplican por un mismo número, la media aritmética queda multiplicada por dicho número.

Además, la media aritmética cumple las propiedades de *continuidad*, *conmutatividad*, *monotonía* e *idempotencia*. Sin embargo, no cumple la propiedad de *asociatividad*.

5.1.4.2. Media Ponderada

La media ponderada de un conjunto de números es el resultado de multiplicar cada uno de ellos por un valor no negativo llamado *peso*, estando cada peso w_i asociado a un valor x_i ,

y dividir la suma de estos productos por la suma de todos los pesos.

$$\phi(x_1, \dots, x_n) = \bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i} \quad (5.7)$$

Este operador se utiliza cuando no todos los componentes que se desean agregar tienen la misma importancia, asignando de esta forma mayores pesos a los valores más importantes y pesos más pequeños a los valores de menor importancia. Un caso particular y muy utilizado de media ponderada consiste en utilizar un conjunto de pesos $W = \{w_1, \dots, w_n\}$ de forma que todos los pesos w_i estén comprendidos entre 0 y 1, y la suma de todos ellos sea 1, en cuyo caso la expresión 5.7 quedaría como:

$$\phi(x_1, \dots, x_n) = \bar{x} = \sum_{i=1}^n w_i x_i \quad (5.8)$$

Al igual que la media aritmética, este operador cumple las propiedades de *conmutatividad*, *monotonía* e *idempotencia*, dado un conjunto de pesos W . Sin embargo, debido a la flexibilidad de los valores en el propio conjunto de pesos, no se puede asegurar la propiedad de *continuidad*.

La media ponderada cumple además las siguientes propiedades:

1. Si todos los pesos son iguales, $w_i = w$, tenemos la media aritmética.
2. *Dictadura*: Dados $w_j = 1$ y $w_i = 0$, con $i \neq j$, la media ponderada será igual al valor x_j .

En nuestra versión actual del sistema de apoyo al consenso implementado en COMAS, aún no se asigna diferente grado de importancia a cada alternativa, experto o valoración, por lo cual la media ponderada emplea un peso de 1 para cada elemento, actuando así como lo haría la media aritmética. No obstante, hemos decidido tener en cuenta la implementación de esta medida para futuras versiones de la aplicación.

5.1.4.3. Media Geométrica

En estadística, la media geométrica de un conjunto de valores X es igual a la raíz n -ésima del producto de todos estos números.

$$\phi(x_1, \dots, x_n) = \bar{x} = \sqrt[n]{\prod_{i=1}^n x_i} = \sqrt[n]{x_1 \dots x_n} \quad (5.9)$$

Su principal atractivo es que resulta menos sensible que la media aritmética a valores extremos en el conjunto. Sin embargo, presenta una clara desventaja cuando uno de los valores x_i es igual a 0, ya que el resultado será 0, independientemente de cómo esté distribuido el resto de valores.

La media geométrica cumple la siguiente propiedad,

$$\log(\sqrt[n]{x_1 \dots x_n}) = \frac{1}{n} \sum_{i=1}^n \log(x_i) \quad (5.10)$$

que nos dice que el logaritmo de la media geométrica es igual a la media aritmética de los logaritmos de los valores x_i .

Otra propiedad interesante que cumple este operador es que está acotado superiormente por la media aritmética, es decir:

$$\sqrt[n]{x_1 \dots x_n} \leq \frac{1}{n} \sum_{i=1}^n x_i \quad (5.11)$$

Lo que significa que la media geométrica siempre devolverá un valor menor o igual que la media aritmética, para un mismo conjunto de datos X . Ambos operadores darán el mismo resultado en casos de unanimidad o idempotencia: $x_i = x$, $i = 1, \dots, n$.

La media geométrica cumple además las propiedades de *idempotencia* y *conmutatividad*.

5.1.4.4. Media Armónica

La media armónica H de una cantidad finita de valores se calcula como la inversa de la media aritmética obtenida a partir de dichos valores a la inversa:

$$\phi(x_1, \dots, x_n) = H = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}} \quad (5.12)$$

Pese a resultar ser una medida bastante representativa en general, y considerar siempre todos los valores de la distribución, encontramos el gran inconveniente de que en nuestro trabajo aparece el 0 con frecuencia como uno de los valores a agregar, en cuyo caso esta medida no puede ser aplicada, al tener lugar una división por 0.

Además, al no estar definida cuando alguno de los valores es 0, no se cumple ninguna de las propiedades deseables en operadores de agregación en estos casos.

Por estas razones, decidimos finalmente no tener en cuenta esta medida en el estudio.

5.1.4.5. Operadores OWA (Ordered Weighted Averaging)

Yager introdujo en [80] el concepto de los operadores OWA, así como su aplicación en problemas de toma de decisión multi-criterio. Debido a la flexibilidad que proporcionan, los operadores OWA se ajustan a medidas de *soft consensus* basadas en mayoría difusa, al ser operadores basados en un conjunto de pesos que pueden ser cuantificados lingüísticamente. Además, su generalidad permite establecer diferentes configuraciones específicas durante su utilización, restringiendo o relajando así el grado en que queramos considerar la existencia o no de consenso mediante diferentes grados de optimismo/pesimismo.

De forma general, dado un conjunto de elementos $X = \{x_1, \dots, x_n\}$ y un conjunto de pesos $W = \{w_1, \dots, w_n\}$, tales que $w_i \in [0, 1]$ y $\sum_i w_i = 1$, el operador OWA asociado, el cual denotamos como F , se define como:

$$F(x_1, \dots, x_n) = w_1 b_1 + w_2 b_2 + \dots + w_n b_n \quad (5.13)$$

Donde b_i es el i -ésimo mayor elemento de X , luego la principal característica en estos operadores de agregación es que los valores a agregar se ordenan de mayor a menor.

Como podemos ver, los operadores OWA precisan en general de un conjunto de pesos w_i para cada elemento, cada uno de los cuales queda ahora asociado al i -ésimo mayor elemento b_i (el cual no será necesariamente x_i). La obtención de los pesos en estos operadores se puede realizar mediante el empleo de *cuantificadores lingüísticos* asociados a conjuntos difusos como *muchos*, *más del 75 %*, etc [49, 55].

Los operadores OWA permiten definir una medida de optimismo (*orness*) en función del conjunto de pesos empleado, W [31]:

$$orness(W) = \frac{1}{n-1} \sum_{i=1}^n ((n-i)w_i) \quad (5.14)$$

Esta medida está comprendida en el intervalo $[0,1]$, y determina el grado en el que la agregación está proxima al operador OR (máximo), siendo 1 cuando se emplea el propio operador OR. Si el valor $orness(W)=0$, esto significa que estamos empleando el operador AND. Por otra parte, al utilizar la media aritmética se obtiene un valor de $orness(W)=0.5$ (grado de optimismo neutral, intermedio entre AND y OR).

De forma análoga se define una medida de pesimismo del operador OWA utilizado, como:

$$andness(W) = 1 - orness(W) = 1 - \frac{1}{n-1} \sum_{i=1}^n ((n-i)w_i) \quad (5.15)$$

Los operadores OWA cumplen de forma general las propiedades de *continuidad*, *monotonidad* e *idempotencia*. Además, no incumplen la propiedad de *conmutatividad* debido a que no tiene sentido hablar de ésta al establecer un orden en los valores x_i : decimos por ello que OWA establece una familia de *operadores simétricos*. Además, son operadores *robustos*, en el sentido de que son capaces de utilizar todos los datos a agregar, minimizando la influencia de valores extremos.

Sin embargo, aunque los operadores OWA son por lo general *no asociativos* [33], presentan un tipo particular de asociatividad conocido como *ordered linkage*, que describimos a continuación: si consideramos un conjunto de valores ordenados $x_1 \leq x_2 \leq \dots \leq x_{2m}$, notamos con $F^{(m)}$ a un operador OWA sobre m elementos y con $F^{(m+1)}$ a un operador OWA sobre $m + 1$ elementos, se cumple lo siguiente:

$$F^{(m+1)}(F^{(m)}(x_1, \dots, x_m), F^{(m)}(x_2, \dots, x_{m+1}), \dots, F^{(m)}(x_{m+1}, \dots, x_{2m})) = \quad (5.16)$$

$$F^{(m)}(F^{(m+1)}(x_1, \dots, x_{m+1}), F^{(m+1)}(x_2, \dots, x_{m+2}), \dots, F^{(m+1)}(x_m, \dots, x_{2m}))$$

Otras propiedades cumplidas por los operadores OWA son (recordar que los pesos w_i están asociados a los valores ya ordenados b_i):

1. $F(x_1, \dots, x_n) = \min(x_1, \dots, x_n)$ cuando $W = \{0, 0, \dots, 0, 1\}$.
2. $F(x_1, \dots, x_n) = \max(x_1, \dots, x_n)$ cuando $W = \{1, 0, \dots, 0, 0\}$.

Existen varios casos particulares de uso de operadores OWA [81], los cuales tendremos en cuenta en nuestro estudio y conoceremos en los siguientes apartados. De entre los operadores anteriormente estudiados, destacar que la *media aritmética* equivale al operador OWA con $w_i = \frac{1}{n}$, con $i = 1, \dots, n$.

5.1.4.6. Operador OR (Máximo)

Consiste simplemente en considerar como la agregación de un conjunto de valores el máximo de dicho conjunto. Este operador supone un caso particular de operador OWA, donde consideramos $w_1 = 1$ y $w_i = 0$ para $i \neq 1$:

$$\phi(x_1, \dots, x_n) = F^*(x_1, \dots, x_n) = \max_i[x_i] \quad (5.17)$$

La noción de este operador como un caso concreto de la familia de operadores OWA nos permite hacer una extensión del mismo, dando lugar al llamado operador *k-máximo*, donde

tomando $w_k = 1$, ($1 \leq k \leq n$) y $w_i = 0$, ($i \neq k$) se obtiene como valor de agregación el k -ésimo mayor valor del conjunto.

Pese a ser una forma sumamente sencilla de agregar una serie de valores, el operador OR resulta poco representativo cuando existe una gran dispersión o variabilidad entre dichos valores.

5.1.4.7. Operador AND (Mínimo)

De forma análoga al operador OR, en este caso consideramos el mínimo valor como agregación del conjunto. El operador AND equivale al caso particular de OWA, donde $w_i = 0$ para $i \neq n$ y $w_n = 1$.

$$\phi(x_1, \dots, x_n) = F_*(x_1, \dots, x_n) = \min_i[x_i] \quad (5.18)$$

Al igual que el operador OR, esta forma de agregación puede ser poco representativa cuando tenemos valores muy diferentes entre sí. Asimismo, la noción de este operador como un caso concreto de la familia de operadores OWA nos permite extenderlo, dando lugar al llamado operador k -mínimo, donde tomando $w_{n-k+1} = 1$, ($1 \leq k \leq n$) y $w_i = 0$, ($i \neq n - k + 1$) se obtiene como valor de agregación el k -ésimo menor valor del conjunto.

5.1.4.8. Operador MAX-MIN

Cuando deseamos agregar un conjunto de valores que forman una matriz de dimensión $p \times q$, es posible aplicar, además de los operadores de agregación expuestos hasta ahora (a partir de todos los valores de la matriz), el siguiente y sucesivos operadores, basados en la idea general de operador OWA. Así, dada una matriz de la forma:

$$C = \begin{pmatrix} c_{11} & \dots & c_{1q} \\ \vdots & \ddots & \vdots \\ c_{p1} & \dots & c_{pq} \end{pmatrix}$$

Podemos considerar c_i como la i -ésima fila de la matriz, es decir, $c_i = \{c_{i1}, \dots, c_{iq}\}$. Partiendo de esta premisa, el operador MAX-MIN consiste en determinar, en primer lugar, el valor mínimo en cada vector c_i , el cual notaremos como $G_p(c_i)$. Es decir:

$$G_p(c_i) = \min_j c_{ij} \quad (5.19)$$

A continuación, se agregan cada uno de estos “mínimos”, eligiendo el mayor de todos ellos, obteniendo así el valor agregado para la matriz, $G_p(C)$:

$$G_p(C) = \max_i G_p(c_i) \quad (5.20)$$

Luego,

$$\phi(c_1, \dots, c_n) = G_p(C) = \max_i \min_j c_{ij} \quad (5.21)$$

Como hemos podido ver, MAX-MIN combina el uso del operador OR, con $orness=1$, con el uso del operador AND, de $orness=0$. Dado que este operador de agregación se aplica en dos fases, empleamos la primera de ellas (mínimo) para obtener los grados de consenso a nivel de alternativa a partir de los grados de consenso a nivel de pares de alternativas; y utilizamos la segunda de ellas (máximo) para calcular el grado de consenso a nivel de relación a partir de los grados de consenso a nivel de alternativa.

5.1.4.9. Operador de Hurwicz

Una estrategia para aplicar un operador intermedio entre el máximo (OR) y mínimo (AND), consiste en aplicar el operador de Hurwicz, donde al igual que en el operador anterior partimos de un conjunto de vectores o filas de una matriz, c_i .

En este caso, se calcula el valor representativo de cada vector c_i como:

$$G_H(c_i) = \alpha G_p(c_i) + (1 - \alpha) G_O(c_i) \quad (5.22)$$

Donde $G_p(c_i) = \min_j c_{ij}$ y $G_O(c_i) = \max_j c_{ij}$. El coeficiente $\alpha \in [0, 1]$ será uno de los parámetros modificables en el estudio, e indica el grado de optimismo u $orness$ del operador.

Un valor de $\alpha=0$ equivaldrá a un optimismo igual a 1, o lo que es lo mismo, utilizar el operador máximo (OR). Por el contrario, un valor de $\alpha=1$ supondrá un grado de optimismo nulo, es decir, la utilización del operador mínimo (AND). Por último y un valor de 0.5 implica el cálculo de la media aritmética entre el valor mínimo y el valor máximo en c_i , la cual, al igual que una media aritmética sobre todos los elementos de c_i , presenta un *orness*=0.5.

Una vez obtenidos los valores $G_H(c_i)$ de nuevo obtenemos la agregación total como el máximo de todos ellos:

$$\phi(c_1, \dots, c_n) = G_H(C) = \max_i G_H(c_i) \quad (5.23)$$

El operador de Hurwicz es en realidad una particularidad de operador OWA, donde $w_1 = 1 - \alpha$, $w_n = \alpha$, y $w_i = 0$ ($i = 2, \dots, n - 1$).

5.1.4.10. Operador de Laplace

Este último enfoque, también conocido como enfoque *normativo*, es un operador de decisión clásico ampliamente utilizado tanto en toma de decisiones como en procesos estadísticos. Consiste en calcular el valor representativo de cada vector c_i de la siguiente forma:

$$G_N(c_i) = \frac{1}{q} \sum_{j=1}^q c_{ij} \quad (5.24)$$

Al igual que en los dos enfoques anteriores, el valor final agregado se obtiene como:

$$\phi(c_1, \dots, c_n) = G_N(C) = \max_i G_N(c_i) \quad (5.25)$$

Por consiguiente, podemos ver cómo el operador de Laplace es en realidad una combinación de la media aritmética con el operador OR. El grado de optimismo (*orness*) del operador de Laplace es de 0.5.

5.1.5. Pruebas realizadas

En este apartado explicaremos brevemente en qué han consistido los procesos de simulación llevados a cabo sobre el sistema. Como hemos dicho al comenzar el capítulo, la principal pregunta de investigación que nos planteamos consiste en determinar de qué forma afecta al proceso de consenso el uso de la plataforma bajo unos operadores/medidas u otros, para posteriormente extraer una serie de conclusiones o guías de actuación para COMAS, en función de las necesidades del equipo de decisión en cada momento.

Antes de llevar a cabo la fase de simulación, es necesario fijar una hipótesis acerca de los resultados de la misma, es decir, una afirmación que sirva como nexo de unión entre los fundamentos teóricos expuestos y dicha experimentación. Nuestra hipótesis sostiene que el empleo de diferentes combinaciones de medidas y operadores afectará de forma directa sobre el desarrollo del proceso de consenso llevado a cabo en nuestro sistema.

Las pruebas consisten en la realización de una serie de ejecuciones para resolver el problema propuesto, en las cuales se han tenido en cuenta las características que hemos planteado para resolver dicho problema en la sección 5.1.1. Cada prueba consiste en combinar una medida de similitud y un operador de agregación diferentes en la resolución del problema, de cara a conocer de qué forma afecta el uso de unas medidas u otras en el proceso de consenso. La medida de similitud utilizada en cada prueba se aplicará en el cálculo de las matrices de similitud entre los expertos, así como en la obtención de las matrices de proximidad de cada uno de ellos con la preferencia colectiva. Por otro lado, el operador de agregación considerado en cada prueba se aplicará exclusivamente en el cálculo del grado de consenso global cr a partir de la matriz de consenso CM . Para realizar la agregación de varios valores en otras etapas del proceso diferentes a esta, como por ejemplo en la posterior obtención de la matriz de proximidad media a partir de las matrices de proximidad, se utiliza por defecto la *media aritmética*.

Tras la anterior revisión teórica, hemos decidido realizar las pruebas con *nueve* de los

operadores de agregación expuestos, debido a las propiedades que estos presentan y a que permiten el cálculo del grado de consenso de forma suavizada.

Las pruebas que se han llevado a cabo son las siguientes:

- **Prueba 1:** Agregación mediante *media aritmética* para la obtención del grado de consenso, y cálculo de la *similitud basada en distancia euclídea*.
 - **Prueba 2:** Agregación mediante *media aritmética* para la obtención del grado de consenso, y cálculo de la *similitud cuadrática*.
 - **Prueba 3:** Agregación mediante *media aritmética* para la obtención del grado de consenso, y cálculo de la *similitud cuadrática mejorada*.
 - **Prueba 4:** Agregación mediante *media geométrica* para la obtención del grado de consenso, y cálculo de la *similitud basada en distancia euclídea*.
 - **Prueba 5:** Agregación mediante *media geométrica* para la obtención del grado de consenso, y cálculo de la *similitud cuadrática*.
 - **Prueba 6:** Agregación mediante *media geométrica* para la obtención del grado de consenso, y cálculo de la *similitud cuadrática mejorada*.
 - **Prueba 7:** Agregación mediante selección del valor máximo (*OR*) para la obtención del grado de consenso, y cálculo de la *similitud basada en distancia euclídea*.
 - **Prueba 8:** Agregación mediante selección del valor máximo (*OR*) para la obtención del grado de consenso, y cálculo de la *similitud cuadrática*.
 - **Prueba 9:** Agregación mediante selección del valor máximo (*OR*) para la obtención del grado de consenso, y cálculo de la *similitud cuadrática mejorada*.
 - **Prueba 10:** Agregación mediante selección del valor mínimo (*AND*) para la obtención del grado de consenso, y cálculo de la *similitud basada en distancia euclídea*.
-

- **Prueba 11:** Agregación mediante selección del valor mínimo (*AND*) para la obtención del grado de consenso, y cálculo de la *similitud cuadrática*.
 - **Prueba 12:** Agregación mediante selección del valor mínimo (*AND*) para la obtención del grado de consenso, y cálculo de la *similitud cuadrática mejorada*.
 - **Prueba 13:** Agregación mediante uso del operador *MAX-MIN* para la obtención del grado de consenso, y cálculo de la *similitud basada en distancia euclídea*.
 - **Prueba 14:** Agregación mediante uso del operador *MAX-MIN* para la obtención del grado de consenso, y cálculo de la *similitud cuadrática*.
 - **Prueba 15:** Agregación mediante uso del operador *MAX-MIN* para la obtención del grado de consenso, y cálculo de la *similitud cuadrática mejorada*.
 - **Prueba 16:** Empleo del operador de *Hurwicz*, con un grado de optimismo alto ($\alpha = 0.25$), para la obtención del grado de consenso, y cálculo de la *similitud basada en distancia euclídea*.
 - **Prueba 17:** Empleo del operador de *Hurwicz*, con un grado de optimismo alto ($\alpha = 0.25$), para la obtención del grado de consenso, y cálculo de la *similitud cuadrática*.
 - **Prueba 18:** Empleo del operador de *Hurwicz*, con un grado de optimismo alto ($\alpha = 0.25$), para la obtención del grado de consenso, y cálculo de la *similitud cuadrática mejorada*.
 - **Prueba 19:** Empleo del operador de *Hurwicz*, con un grado de optimismo intermedio ($\alpha = 0.5$), para la obtención del grado de consenso, y cálculo de la *similitud basada en distancia euclídea*.
 - **Prueba 20:** Empleo del operador de *Hurwicz*, con un grado de optimismo intermedio ($\alpha = 0.5$), para la obtención del grado de consenso, y cálculo de la *similitud cuadrática*.
-

- **Prueba 21:** Empleo del operador de *Hurwicz*, con un grado de optimismo intermedio ($\alpha = 0.5$), para la obtención del grado de consenso, y cálculo de la *similitud cuadrática mejorada*.
- **Prueba 22:** Empleo del operador de *Hurwicz*, con un grado de optimismo bajo ($\alpha = 0.75$), para la obtención del grado de consenso, y cálculo de la *similitud basada en distancia euclídea*.
- **Prueba 23:** Empleo del operador de *Hurwicz*, con un grado de optimismo bajo ($\alpha = 0.75$), para la obtención del grado de consenso, y cálculo de la *similitud cuadrática*.
- **Prueba 24:** Empleo del operador de *Hurwicz*, con un grado de optimismo bajo ($\alpha = 0.75$), para la obtención del grado de consenso, y cálculo de la *similitud cuadrática mejorada*.
- **Prueba 25:** Agregación mediante empleo del operador de *Laplace* para la obtención del grado de consenso, y cálculo de la *similitud basada en distancia euclídea*.
- **Prueba 26:** Agregación mediante empleo del operador de *Laplace* para la obtención del grado de consenso, y cálculo de la *similitud cuadrática*.
- **Prueba 27:** Agregación mediante empleo del operador de *Laplace* para la obtención del grado de consenso, y cálculo de la *similitud cuadrática mejorada*.

La siguiente sección recoge los resultados obtenidos a partir de cada una de las pruebas que hemos llevado a cabo.

5.2. Resultados Obtenidos

Esta última sección del capítulo recoge los resultados obtenidos al llevar a cabo los diferentes experimentos propuestos, con vistas a analizar la evolución y convergencia del consenso alcanzado a lo largo del proceso. Los resultados que mostraremos a lo largo de las siguientes

páginas se centran únicamente en el proceso de alcance de consenso, concretamente en la *evolución del grado de consenso global* (a nivel de relación de preferencia) durante las sucesivas rondas de discusión. Estos resultados se muestran ordenados según la combinación de medida de similitud y operador de agregación utilizados en la resolución del problema. En particular, utilizaremos una serie de tablas para mostrar en cada una de ellas los resultados obtenidos en el grado de consenso al resolver el problema mediante un operador de agregación determinado, combinado con cada una de las tres medidas de similitud propuestas.

Agregación mediante Media Aritmética.

Prueba 1. *Media Aritmética y Similitud basada en Distancia Euclídea.*

Prueba 2. *Media Aritmética y Similitud Cuadrática.*

Prueba 3. *Media Aritmética y Similitud Cuadrática Mejorada.*

	M. Aritmética		
	Euclídea	Cuadrática	Cuadrática Mejorada
<i>Ronda 1</i>	0,655802218	0,823736071	0,711521564
<i>Ronda 2</i>	0,732618477	0,872194013	0,792565862
<i>Ronda 3</i>	0,804385901	-	0,872199773
<i>Ronda 4</i>	0,870955326	-	-
<i>Nº de rondas necesarias</i>	4	2	3

Agregación mediante Media Geométrica.

Prueba 4. *Media Geométrica y Similitud basada en Distancia Euclídea.*

Prueba 5. *Media Geométrica y Similitud Cuadrática.*

Prueba 6. *Media Geométrica y Similitud Cuadrática Mejorada.*

	M. Geométrica		
	Euclídea	Cuadrática	Cuadrática Mejorada
<i>Ronda 1</i>	0,446128848	0,596898868	0,425129194
<i>Ronda 2</i>	0,683090875	0,862076653	0,702775115
<i>Ronda 3</i>	0,764617729	-	0,811505871
<i>Ronda 4</i>	0,837491761	-	0,862970318
<i>Ronda 5</i>	0,870373219	-	-
<i>Nº de rondas necesarias</i>	5	2	4

Agregación mediante Máximo (OR).

Prueba 7. *OR y Similitud basada en Distancia Euclídea.*

Prueba 8. *OR y Similitud Cuadrática.*

Prueba 9. *OR y Similitud Cuadrática Mejorada.*

	OR		
	Euclídea	Cuadrática	Cuadrática Mejorada
<i>Ronda 1</i>	1,0	1,0	1,0
<i>Nº de rondas necesarias</i>	1	1	1

Agregación mediante Mínimo (AND).

Prueba 10. *AND y Similitud basada en Distancia Euclídea.*

Prueba 11. *AND y Similitud Cuadrática.*

Prueba 12. *AND y Similitud Cuadrática Mejorada.*

	AND		
	Euclídea	Cuadrática	Cuadrática Mejorada
<i>Ronda 1</i>	0,020126056	0,035825819	0,001027007
<i>Ronda 2</i>	0,120126056	0,222188828	0,027602888
<i>Ronda 3</i>	0,220126056	0,388551837	0,09417877
<i>Ronda 4</i>	0,320126056	0,534914846	0,200754652
<i>Ronda 5</i>	0,420126056	0,661277856	0,347330534
<i>Ronda 6</i>	0,520126056	0,767640865	0,531852403
<i>Ronda 7</i>	0,620126056	0,854003874	0,705276521
<i>Ronda 8</i>	0,720126056	-	0,838700639
<i>Ronda 9</i>	0,819877749	-	0,867019371
<i>Ronda 10</i>	0,865766204	-	-
<i>Nº de rondas necesarias</i>	10	7	9

Agregación mediante Operador MAX-MIN.

Prueba 13. *MAX-MIN y Similitud basada en Distancia Euclídea.*

Prueba 14. *MAX-MIN y Similitud Cuadrática.*

Prueba 15. *MAX-MIN y Similitud Cuadrática Mejorada.*

	MAX-MIN		
	Euclídea	Cuadrática	Cuadrática Mejorada
<i>Ronda 1</i>	0,071735527	0,173796347	0,008644638
<i>Ronda 2</i>	0,171735527	0,345516944	0,053859414
<i>Ronda 3</i>	0,271735527	0,497237541	0,13907419
<i>Ronda 4</i>	0,371735527	0,628958138	0,264288966
<i>Ronda 5</i>	0,471735527	0,740678735	0,429503742
<i>Ronda 6</i>	0,571735527	0,832399332	0,617429243
<i>Ronda 7</i>	0,671735527	0,887956489	0,772214467
<i>Ronda 8</i>	0,771735527	-	0,886999691
<i>Ronda 9</i>	0,863739696	-	-
<i>Nº de rondas necesarias</i>	9	7	8

Agregación mediante Operador de Hurwicz ($\alpha = 0.25$).

Prueba 16. *Hurwicz(0.25) y Similitud basada en Distancia Euclídea.*

Prueba 17. *Hurwicz(0.25) y Similitud Cuadrática.*

Prueba 18. *Hurwicz(0.25) y Similitud Cuadrática Mejorada.*

	Hurwicz (0.25)		
	Euclídea	Cuadrática	Cuadrática Mejorada
<i>Ronda 1</i>	0,764745743	0,794383301	0,760950512
<i>Ronda 2</i>	0,790745743	0,837185244	0,786950512
<i>Ronda 3</i>	0,816827437	0,86638673	0,813260185
<i>Ronda 4</i>	0,842827437	-	0,840543386
<i>Ronda 5</i>	0,855827437	-	0,860062327
<i>Nº de rondas necesarias</i>	5	3	5

Agregación mediante Operador de Hurwicz ($\alpha = 0.5$).

Prueba 19. *Hurwicz(0.5) y Similitud basada en Distancia Euclídea.*

Prueba 20. *Hurwicz(0.5) y Similitud Cuadrática.*

Prueba 21. *Hurwicz(0.5) y Similitud Cuadrática Mejorada.*

	Hurwicz (0.5)		
	Euclídea	Cuadrática	Cuadrática Mejorada
<i>Ronda 1</i>	0,53818778	0,577164654	0,503575904
<i>Ronda 2</i>	0,638197957	0,740937293	0,565852531
<i>Ronda 3</i>	0,738197957	0,864709924	0,708129158
<i>Ronda 4</i>	0,837174671	-	0,878782882
<i>Ronda 5</i>	0,90070096	-	-
<i>Nº de rondas necesarias</i>	5	3	4

Agregación mediante Operador de Hurwicz ($\alpha = 0.75$).

Prueba 22. *Hurwicz(0.75) y Similitud basada en Distancia Euclídea.*

Prueba 23. *Hurwicz(0.75) y Similitud Cuadrática.*

Prueba 24. *Hurwicz(0.75) y Similitud Cuadrática Mejorada.*

	Hurwicz (0.75)		
	Euclídea	Cuadrática	Cuadrática Mejorada
<i>Ronda 1</i>	0,266898734	0,378276168	0,259427275
<i>Ronda 2</i>	0,346898734	0,50652147	0,295238759
<i>Ronda 3</i>	0,426898734	0,619966772	0,361050242
<i>Ronda 4</i>	0,506918914	0,718612075	0,456861726
<i>Ronda 5</i>	0,58693151	0,802457377	0,582515074
<i>Ronda 6</i>	0,666963138	0,871502687	0,719630143
<i>Ronda 7</i>	0,746963138	-	0,833818659
<i>Ronda 8</i>	0,824248867	-	0,879572152
<i>Ronda 9</i>	0,877475158	-	-
<i>Nº de rondas necesarias</i>	9	6	8

Agregación mediante Operador de Laplace.

Prueba 25. *Laplace y Similitud basada en Distancia Euclídea.*

Prueba 26. *Laplace y Similitud Cuadrática.*

Prueba 27. *Laplace y Similitud Cuadrática Mejorada.*

	Laplace		
	Euclídea	Cuadrática	Cuadrática Mejorada
<i>Ronda 1</i>	0,67521157	0,857200266	0,716919511
<i>Ronda 2</i>	0,750253472	-	0,806460718
<i>Ronda 3</i>	0,817882273	-	0,881873183
<i>Ronda 4</i>	0,883922317	-	-
<i>Nº de rondas necesarias</i>	4	1	3

Resultados según el Operador de Agregación Empleado

A continuación mostramos el número de rondas de discusión que han sido necesarias para alcanzar el consenso en cada una de las pruebas realizadas de forma gráfica. Cada gráfico incluye el resultado obtenido al aplicar los diferentes operadores de agregación bajo una misma medida de similitud.

La Figura 5.3 muestra el número de rondas de consenso obtenido en las simulaciones realizadas bajo el uso de la similitud basada en distancia euclídea. La Figura 5.4 muestra el número de rondas de consenso obtenido en las simulaciones realizadas bajo el uso de la similitud cuadrática. Por último, la Figura 5.5 muestra el número de rondas de consenso obtenido en las simulaciones realizadas bajo el uso de la similitud cuadrática mejorada.

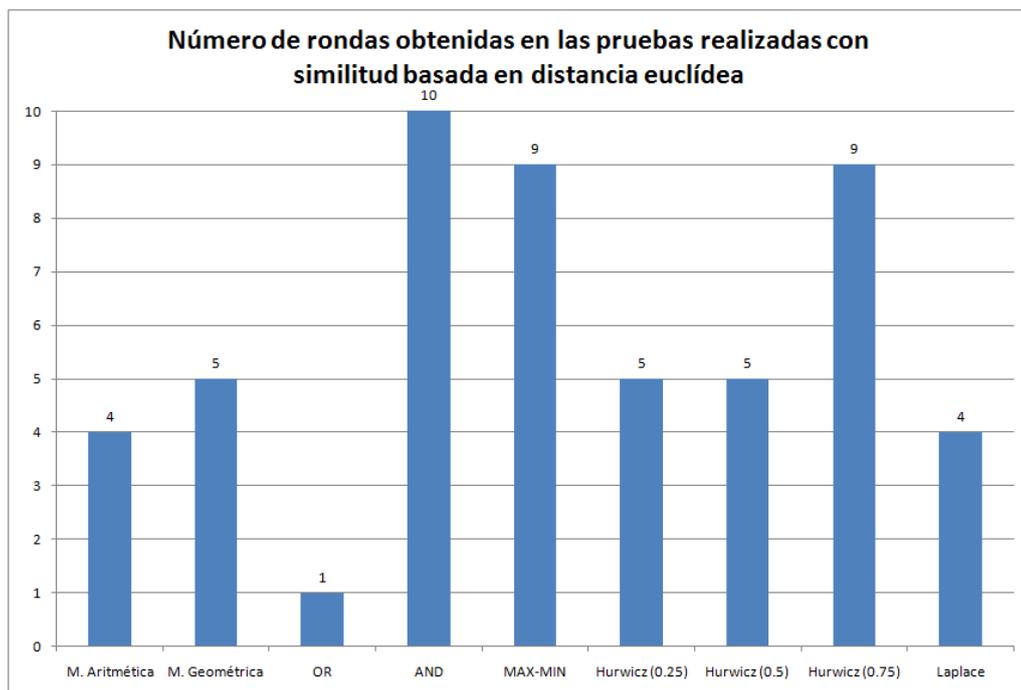


Figura 5.3: Resultados obtenidos con similitud Euclídea

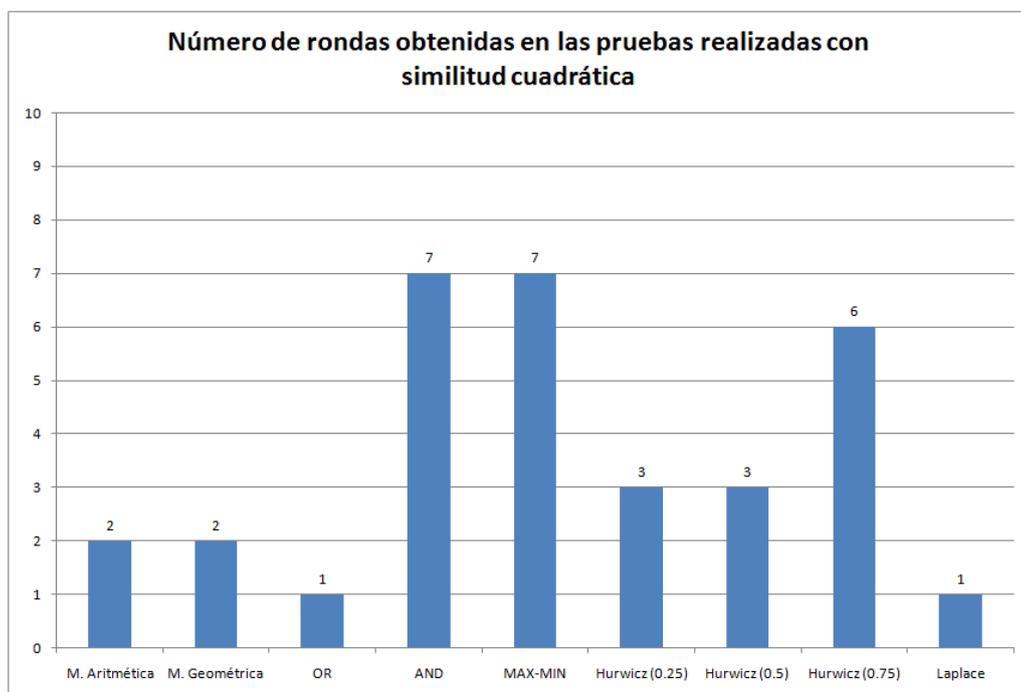


Figura 5.4: Resultados obtenidos con similitud Cuadrática

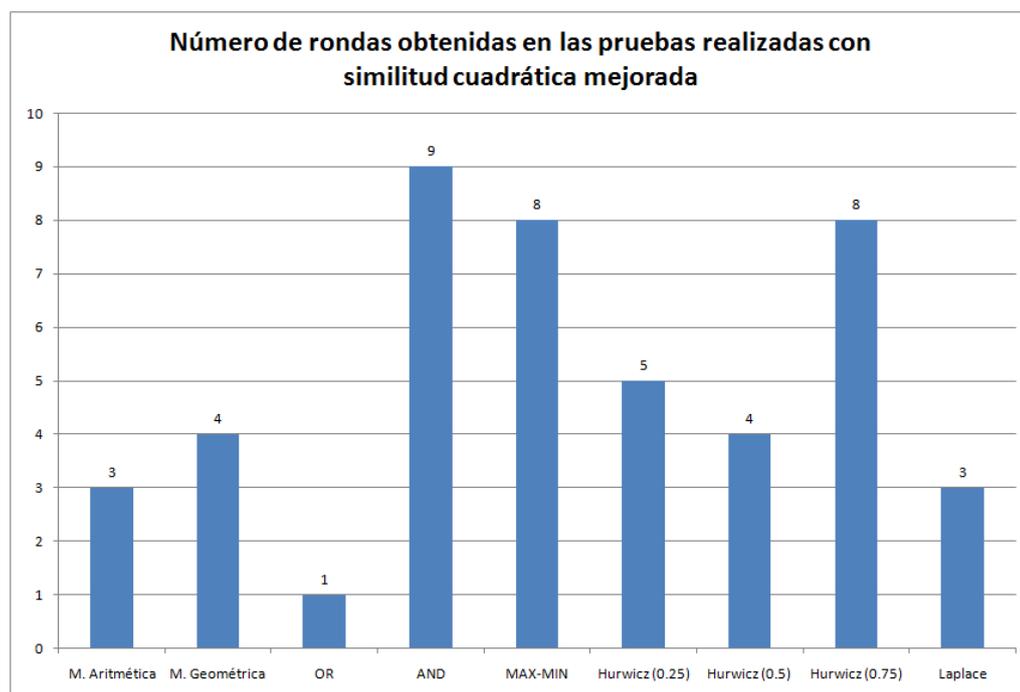


Figura 5.5: Resultados obtenidos con similitud Cuadrática mejorada

5.3. Conclusión

A partir de los resultados mostrados en la sección, resultaría precipitado dar unas conclusiones finales directas sobre la simulación de uso de nuestra plataforma. En su lugar, hemos creído más conveniente mostrar en primer lugar un breve análisis en función de los diversos criterios tenidos en cuenta en el estudio.

Medidas de Similitud

Bajo el punto de vista de las *medidas de similitud*, la similitud cuadrática muestra mayor velocidad de convergencia hacia el consenso que el resto de medidas, con independencia del operador de agregación utilizado. Ello es debido a que esta medida tiene la particularidad de que el valor de similitud es siempre mayor que en el resto de medidas, para una misma distancia $d(x, y)$, tal y como se puede apreciar en la Figura 5.1.

Por otra parte, la similitud basada en distancia euclídea y cuadrática mejorada permiten una mayor discusión, a lo largo de un número de rondas mayor.

Por ello, proponemos el uso de la similitud cuadrática cuando queremos resolver con COMAS un problema en cual no dispongamos de mucho tiempo para llevar a cabo el proceso de consenso en general. En cualquier otro caso, recomendamos el empleo de la similitud basada en distancia euclídea, así como la similitud cuadrática mejorada, para resolver problemas de TDG bajo consenso cuando deseemos una mayor discusión entre los expertos.

Operadores de Agregación

Centrando nuestra atención en los *operadores de agregación*, daremos una serie de recomendaciones de uso de los mismos, además de una breve interpretación de los resultados obtenidos con cada uno de ellos:

- *Media Aritmética*: Presenta una alta convergencia hacia el consenso, aunque menor que en el uso de otros operadores de agregación con los cuales el proceso de discusión antes de alcanzar el consenso es mínimo.
 - *Media Geométrica*: Es una medida que inicialmente asegura una rápida convergencia. Sin embargo, el hecho de no cumplir algunas propiedades importantes al trabajar con valores nulos hace que sea una medida poco aconsejable.
 - *Máximo (OR)*: Las simulaciones realizadas con este operador dejaron claro que este operador resulta inadecuado, ya que proporciona un resultado demasiado optimista y poco realista en comparación con otros operadores.
 - *Mínimo (AND)*: Debido a que es el operador con mayor grado de pesimismo, presenta la mayor lentitud de convergencia, por lo que su uso es desaconsejable.
 - *MAXMIN*: Pese a suponer una pequeña mejora sobre el anterior operador, sigue siendo bastante lento e inadecuado en cualquier proceso, a pesar de estar basado en la familia de operadores OWA.
-

- *Operador de Hurwicz*: El uso de este operador como normal general, y con diferentes valores del parámetro α según las circunstancias, es una elección interesante. Se elegirán valores inferiores a 0.5 para procesos que requieran de una rápida decisión, así como valores superiores a 0.5 cuando sea necesaria una mayor discusión en el proceso. En ningún caso se aconseja utilizar valores superiores a 0.75 (excesivo pesimismo y tendencia a operador AND) y inferiores a 0.25 (excesivo optimismo y tendencia a operador OR). Como sabemos, este es un operador derivado de OWA y, como tal, nos asegura el cumplimiento de las principales propiedades deseables en operadores de agregación.
- *Operador de Laplace*: Es en general una buena opción a utilizar, especialmente si el umbral de consenso es alto (0.85 o superior) y se emplea la similitud euclídea o cuadrática mejorada, en cuyo caso obtendremos un consenso rápido.

Como hemos podido ver, los resultados son totalmente acordes con las propiedades expuestas al estudiar estos operadores.

De forma general, podemos concluir que el uso de diferentes medidas de consenso en nuestro sistema queda determinado por las necesidades del equipo de toma de decisiones correspondiente, ya que habrá situaciones críticas en las que sea necesario alcanzar un acuerdo rápido aunque no todos los individuos estén realmente de acuerdo con la decisión tomada, así como situaciones en las que lo más importante sea conseguir un acuerdo real en todo el equipo a través de un proceso de discusión más elaborado (siempre tomando la noción de *acuerdo* según el concepto de *consenso suavizado*), independientemente del coste invertido en ello. Cabe destacar que el uso de operadores basados en OWA, con diferentes pesos y grado de optimismo, permite obtener un sistema flexible y adaptable a las circunstancias de cada problema y equipo de decisión.

Conclusiones y Trabajos Futuros

Este capítulo cierra la memoria de investigación con las principales conclusiones extraídas a partir de la misma, revisando la principal propuesta del trabajo y los resultados obtenidos. Además, finalizaremos presentando las líneas de investigación y trabajos futuros que nos hemos planteado abordar a partir de estos resultados, algunos de los cuales ya estamos llevando a cabo en la actualidad.

Propuesta y resultados obtenidos

Los procesos de consenso reales en entornos organizativos y/o empresariales implican la participación de múltiples decisores o expertos, así como de una persona encargada de guiarlos, llamada moderador.

Dado que llevar a cabo un proceso de consenso real no suele resultar sencillo, debido al elevado coste temporal en muchas ocasiones, así como la necesidad de llevar a cabo una reunión y la asistencia a la misma por parte de todos los expertos, al comienzo de esta memoria nos planteamos el objetivo de desarrollar un sistema destinado a la automatización de estos procesos. Esta automatización implica un coste computacional tanto mayor cuantos más expertos participen en el problema, razón por la cual decidimos utilizar la tecnología de *Sistemas Multi-Agente*, capaz de llevar a cabo procesos de consenso mediante un elevado número de agentes inteligentes de forma distribuida.

Teniendo en cuenta la elección de la tecnología de sistemas multi-agente para llevar a cabo esta automatización, nuestra meta ha sido la de construir un SAC (Sistema de Apoyo al Consenso) combinado con un Sistema Multi-Agente (SMA) en el que un conjunto de agentes inteligentes con roles determinados se encarguen de realizar un proceso de consenso completo, desde la expresión de las preferencias iniciales por parte del conjunto de expertos hasta el momento en que se alcance un acuerdo entre estos. Estos agentes se comunican, expresan y analizan la información utilizada durante el proceso con la ayuda de una *ontología*, diseñada para el manejo de información en procesos de TDG mediante consenso. Todo ello ha sido realizado con vistas a utilizar nuestro sistema basado en agentes en problemas de consenso con gran demanda computacional. La versión actual es un prototipo inicial para estudiar la viabilidad en este tipo de sistemas.

Los resultados obtenidos a partir del objetivo planteado se pueden resumir en los siguientes puntos:

- El sistema propuesto consigue automatizar los procesos de consenso, reemplazando así la figura del moderador humano. La única responsabilidad de los expertos es la de proporcionar sus preferencias al sistema.
 - La arquitectura multi-agente permite la existencia de un entorno, centralizado o distribuido, en el que un grupo de agentes cooperan y se comunican para lograr un fin común: alcanzar un acuerdo para todos los expertos.
 - La ontología diseñada permite además una comunicación y mecanismo de tratamiento de la información común para todos los agentes, compartiendo así todos ellos el mismo lenguaje y semántica durante todos los actos comunicativos que tienen lugar en el sistema.
 - El estudio e implementación de diversas medidas de consenso, así como una interfaz inicial de usuario para determinar la configuración del proceso antes del comienzo del mismo, han permitido llevar a cabo un estudio en el que se ha podido determinar de
-

qué manera influyen diferentes configuraciones del modelo de consenso considerado en los resultados obtenidos.

Trabajos Futuros

Este trabajo ha presentado una primera versión de COMAS, un Sistema de Apoyo al Consenso basado en un Sistema Multi-Agente. Teniendo en cuenta la importancia actual del consenso en el área de la Toma de Decisiones en Grupo, así como las múltiples líneas de ampliación y mejora de nuestro sistema, nuestros trabajos futuros pretenden abordar las siguientes líneas de investigación (algunas de las cuáles actualmente ya se encuentran en desarrollo):

1. De cara al uso on-line del sistema por parte de diferentes expertos humanos, pretendemos desarrollar una Interfaz Web distribuida, basada en la tecnología de Servicios Web, de manera que diferentes expertos humanos situados en distintas localizaciones en un momento dado pueda expresar sus preferencias iniciales a través de Internet, y así poder llevarse a cabo procesos automáticos de consenso “a distancia”.
 2. Profundizar en los actuales desafíos y debilidades encontrados en los procesos de consenso, especialmente aquellos problemas relacionados con el comportamiento social de algunos expertos en el grupo. Así, a pesar de la hipótesis de partida vista en el capítulo 5 de que en la versión actual de COMAS todos los expertos siguen las recomendaciones sugeridas por el moderador, pretendemos incorporar en nuestro modelo un mecanismo para la detección y control de comportamientos sociales contrarios al consenso por parte de dichos expertos.
 3. Incorporar diferentes modelados de preferencias de los expertos, en problemas definidos en entornos de incertidumbre, así como nuevas estructuras de preferencia que faciliten la expresión de opiniones entre los expertos de un mismo grupo.
 4. De forma general, nuestra principal meta de cara al futuro consiste en aplicar COMAS a entornos de elevada demanda computacional, donde el consenso sobre información o
-

toma de decisiones sea un factor crítico. Ejemplos posibles de aplicación son procesos de *e-Democracia*, así como los entornos de Inteligencia Ambiental o *SmartHomes*, donde una serie de agentes asociados a un conjunto de sensores deben decidir conjuntamente sobre el estado o acción a realizar sobre un entorno inteligente en un momento dado.

Bibliografía

- [1] M. Adellinde and D.W. Uhrmacher. *Multi-Agent Systems: Simulations and Applications*. CRC Press, 2009.
- [2] S. Alonso, E. Herrera-Viedma, F. Chiclana, and F. Herrera. Individual and social strategies to deal with ignorance situations in multi-person decision making. *International Journal of Information Technology and Decision Making*, 8(2):313–333, 2009.
- [3] S. Alonso, I.J. Pérez, F.J. Cabrerizo, and E. Herrera-Viedma. A fuzzy group decision making model for large groups of individuals. *Proceedings of the IEEE International Conference on Fuzzy Systems (Fuzz-IEEE09)*, pages 643–648, 2009.
- [4] J. Bajo, J.M. Corchado, C. Pinzón, Y. Paz, and B. Pérez-Lancho. Scmas: A distributed hierarchical multi-agent architecture for blocking attacks to databases. *International Journal of Innovative Computing, Information and Control*, 6(9):3787–3817, 2010.
- [5] J. Bajo, J.A. Fraile, B. Pérez-Lancho, and J.M. Corchado. The thomas architecture in home care scenarios: A case study. *Expert Systems with Applications*, 37(5):3986–3999, 2010.
- [6] F. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-agent systems with JADE*. Wiley, 2007.

-
- [7] D. Ben-Arieh and Z. Chen. Linguistic labels aggregation and consensus measure for automatic decision-making using group recommendations. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 36(1):558–568, 2006.
- [8] D. Ben-Arieh, T. Easton, and B. Evans. Minimum cost consensus with quadratic cost functions. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 39(1):210–217, 2009.
- [9] J.C. Bezdek, B. Spillman, and R. Spillman. A fuzzy relation space for group decision theory. *Fuzzy sets and Systems*, 1(4):255–268, 1978.
- [10] G. Bordogna, M. Fedrizzi, and G. Pasi. A linguistic modeling of consensus in group decision making based on owa operators. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 27(1):126–133, 1997.
- [11] M.E. Bratman, D.J. Israel, and M.E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(4):1–22, 1988.
- [12] M.E. Bratman, D.J. Israel, and M.E. Pollack. Intention, practical rationality and self-governance. *Ethics*, 119(1):411–443, 2009.
- [13] R. Brooks. Intelligence without representation. *Artificial Intelligence*, 47(1):139–159, 2001.
- [14] N. Bryson. Group decision-making and the analytic hierarchy process. exploring the consensus-relevant information content. *Computers and Operational Research*, 1(23):9, 1996.
- [15] T.X. Bui. *A group decision support system for cooperative multiple criteria group decision making*. Springer-Verlag, 1987.
- [16] C.T.L. Butler and A. Rothstein. *On Conflict and Consensus: A Handbook on Formal Consensus Decision Making*. Takoma Park, 2006.
-

-
- [17] F.J. Cabrerizo, S. Alonso, I.J. Pérez, and E. Herrera-Viedma. On consensus measures in fuzzy group decision making. *Lecture Notes in Computer Science*, 5285:86–97, 2008.
- [18] F.J. Cabrerizo, I.J. Pérez, and E. Herrera-Viedma. Managing the consensus in group decision making in an unbalanced fuzzy linguistic context incomplete information. *Knowledge-Based Systems*, 23(2):169–181, 2010.
- [19] L. Canós and V. Liern. La agregación de información para la toma de decisiones en la empresa. *XIV Jornadas de la Asociación Española de Profesores Universitarios de Matemáticas para la Economía y la Empresa*, 2006.
- [20] F.J. Cantú and H.G. Ceballos. A multiagent knowledge and information network approach for managing research assets. *Expert Systems with Applications*, 37(7):5272–5284, 2010.
- [21] J. Carbó, J.M. Molina, and J. Davila. A bdi agent architecture for reasoning about reputation. *IEEE Systems, Man and Cybernetics Conference*, 2(1):817–822, 2001.
- [22] C. Carlsson, D. Ehrenberg, P. Eklund, M. Fedrizzi, P. Gustafsson, P. Lindholm, G. Merkurueva, T. Riissanen, and A.G.S. Ventre. Consensus in distributed soft environments. *European Journal of Operational Research*, 61(1-2):165–185, 1992.
- [23] F. Castanedo, J.M. Molina, J. García, and J.M. Molina. Extending surveillance systems capabilities using bdi cooperative sensor agents. *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, 1(1):131–138, 2006.
- [24] A. Chávez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. *Proceedings of the 1st International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, pages 75–90, 1996.
- [25] W. Cook and L. Seiford. Priority ranking and consensus formation. *Management Science*, 24(16):1721–1732, 1978.
-

-
- [26] J.M. Corchado, J.A. Fraile, J. Bajo, and B. Pérez-Lancho. Hoca: Multiagent architecture for developing intelligent home care environments. *International Symposium on Distributed Computing and Artificial Intelligence*, 50(1):52–61, 2009.
- [27] Y. Demazeau. *La méthode VOYELLES dans Systèmes Multi-Agents: Des Theories Organisationnelles aux Applications Industrielles*. Hermès, Oslo, 2001.
- [28] Real Academia Española. Diccionario de la lengua española. vigésimo segunda edición, 2001.
- [29] M. Fedrizzi, M. Fedrizzi, and R.A.M. Pereira. Soft consensus and network dynamics in group decision making. *International Journal of Intelligent Systems*, 14(1):63–77, 1999.
- [30] I.A. Ferguson. Towards an architecture for adaptive, rational, mobile agents. *Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent World*, pages 249–262, 1991.
- [31] D.P. Filev. Operations on fuzzy numbers via fuzzy reasoning. *Fuzzy Sets and Systems*, 91(2):137–142, 1997.
- [32] FIPA. Foundation for intelligent physical agents, website: <http://www.fipa.org>.
- [33] J. Fodor, Jean-Luc Marichal, and M. Roubens. Characterization of the ordered weighted averaging operators. *IEEE Transactions on Fuzzy Systems*, 3(2):236–240, 1995.
- [34] J.L. García-Lapresta. Some consensus measures and their applications in group decision making. *FLINS 2008. The 8th International FLINS Conference on Computational Intelligence in Decision and Control*, pages 611–616, 2008.
- [35] F. García-Sánchez, R. Valencia, R. Martínez, and J.T. Fernández. An ontology, intelligent agent-based framework for the provision of semantic web services. *Expert Systems with Applications*, 36(2):3167–3187, 2009.
- [36] F.J. Garijo. Tecnología de agentes: Experiencias y perspectivas para el desarrollo de nuevos servicios y aplicaciones. *Boletín*, 24:1–9, 2002.
-

-
- [37] M. Georgeff, B. Pell, M.E. Pollack, M. Tambe, and M. Wooldridge. The belief-desire-intention model of agency. *Lecture Notes in Computer Science*, 1555:1–10, 1998.
- [38] L. Hamer, Y. Hemeryck, G. Herweyers, M. Janssen, H. Keters, R. Rousseau, and A. Vanhoutte. Similarity measures in scientometric research: The jaccard index versus salton’s cosine formula. *Information Processing and Management*, 25(3):315–318, 1989.
- [39] F. Herrera and E. Herrera-Viedma. Linguistic decision analysis: Steps for solving decision problems under linguistic information. *Fuzzy Sets and Systems*, 115(2000):67–82, 2000.
- [40] F. Herrera, E. Herrera-Viedma, and J. Verdegay. A sequential selection process in group decision making with linguistic assessments. *Information Sciences*, 85(1995):223–239, 1995.
- [41] F. Herrera, E. Herrera-Viedma, and J. Verdegay. A model of consensus in group decision making under linguistic assessments. *Fuzzy sets and Systems*, 78(1):73–87, 1996.
- [42] F. Herrera, E. Herrera-Viedma, and J.L. Verdegay. A rational consensus model in group decision making using linguistic assessments. *Fuzzy Sets and Systems*, 88(1):31–49, 1997.
- [43] E. Herrera-Viedma, S. Alonso, F. Chiclana, and F. Herrera. A consensus model for group decision making with incomplete fuzzy preference relations. *IEEE Transactions on Fuzzy Systems*, 15(5):863–877, 2007.
- [44] E. Herrera-Viedma, F. Chiclana, F. Herrera, and S. Alonso. Group decision-making model with incomplete fuzzy preference relations based on additive consistency. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 37(1):176–189, 2007.
- [45] E. Herrera-Viedma, F. Herrera, and F. Chiclana. A consensus model for multiperson decision making with different preference structures. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 32(3):394–402, 2002.
-

-
- [46] E. Herrera-Viedma, L. Martínez, F. Mata, and F. Chiclana. A consensus support system model for group decision making problems with multigranular linguistic preference relations. *IEEE Transactions on Fuzzy Systems*, 13(5):644–658, 2005.
- [47] M.N. Huhns and M.P. Singh. Cognitive agents. *Internet Computing*, 2(6):87–89, 1998.
- [48] J. Jung. Ontological framework based on contextual mediation for collaborative information retrieval. *Information Retrieval*, 10(1):85–109, 2006.
- [49] J. Kacprzyk. Group decision making with a fuzzy linguistic majority. *Fuzzy Sets and Systems*, 18(2):105–118, 1986.
- [50] J. Kacprzyk. On some fuzzy cores and 'soft' consensus measures in group decision making. *The Analysis of Fuzzy Information*, pages 119–130, J.C. Bezdek, 1987.
- [51] J. Kacprzyk and M. Fedrizzi. A “soft” measure of consensus in the setting of partial (fuzzy) preferences. *European Journal on Operational Research*, 34(1):316–325, 1988.
- [52] J. Kacprzyk, M. Fedrizzi, and H. Nurmi. Group decision making and consensus under fuzzy preferences and fuzzy majority. *Fuzzy Sets and Systems*, 49(1):21–31, 1992.
- [53] J. Kacprzyk, M. Fedrizzi, and H. Nurmi. *“Soft” Degrees of Consensus Under Fuzzy Preferences and Fuzzy Majorities*. Kluwer Academic Publishers, 1997.
- [54] J. Kacprzyk and S. Zadrozny. On a concept of a consensus reaching process support system based on the use of soft computing and web techniques. *Proceedings of the 8th International FLINS Conference*, pages 859–864, 2008.
- [55] J. Kacprzyk and S. Zadrozny. Towards a general and unified characterization of individual and collective choice functions under fuzzy and nonfuzzy preferences and majority via the ordered weighted average operators. *International Journal of Intelligent Systems*, 24(1):4–26, January 2009.
- [56] J. Kacprzyk and S. Zadrozny. Soft computing and web intelligence for supporting consensus reaching. *Soft Computing*, 14(8):833–846, 2010.
-

-
- [57] J.K. Kim, S.H. Choi, C.H. Han, and S.H. Kim. An interactive procedure for multiple criteria group decision making with incomplete information. *Computers and Industrial Engineering*, 35(1-2):295–298, 1998.
- [58] G.J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, 1995.
- [59] L.I. Kuncheva and R. Krishnapuram. A fuzzy consensus aggregation operator. *Fuzzy Sets and Systems*, 79(3):347–356, 1995.
- [60] H.M. Lee. Generalization of the group decision making using fuzzy sets theory for evaluating the rate of aggregative risk in software development. *Information Sciences*, 113(3-4):301–311, FEB 1999.
- [61] H. Legind. Efficient importance weighted aggregation between min and max. *9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002)*, 2002.
- [62] R.D. Luce and H. Raiffa. *Games and Decisions: Introduction and Critical Survey*. Dover Publications, 1989.
- [63] P. Maes. Situated agents can have goals. *Robotics and Autonomous Systems*, 6(1-2):49–70, 1990.
- [64] K. Mahesh and S. Nirenburg. Principles of ontological engineering for natural language processing. *European Conference on Artificial Intelligence, ECAI-96*, 1996.
- [65] L. Martínez and J. Montero. Challenges for improving consensus reaching process in collective decisions. *New Mathematics and Natural Computation*, 3(2):203–217, 2007.
- [66] A. Mas. *Agentes Software y sistemas multi-agente: Conceptos, arquitecturas y aplicaciones*. Pearson Education, 2004.
-

-
- [67] F. Mata, L. Martínez, and E. Herrera-Viedma. An adaptive consensus support model for group decision-making problems in a multigranular fuzzy linguistic context. *IEEE Transactions on Fuzzy Systems*, 17(2):279–290, 2009.
- [68] M. Delgado, J.L. Verdegay, and M.A. Vila. A model for linguistic partial information in decision making problem. *International Journal of Intelligent Systems*, 9(5):365–378, 1994.
- [69] S.E. Middleton, N.R. Shadbolt, and D.C. de Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, 22(1):54–88, 2004.
- [70] J.P. Muller, M. Pischel, and M. Thiel. Modelling reactive behaviour in vertically layered agent architectures. *Intelligent Agents: Theories, Architectures and Languages (LNAI 890)*, pages 261–276, 1995.
- [71] S. Nolfi. Power and the limits of reactive agents. *Neurocomputing*, 42(1):119–145, 2002.
- [72] J. Pérez, J.L. Gimeno, and E. Cerda. *Teoría de Juegos*. Pearson Educación, 2003.
- [73] A.S. Rao and M.P. Georgeff. Bdi agents: From theory to practice. *Proceedings of the First International Conference on Multi-Agent Systems, ICMAS-95*, 1995.
- [74] S. Rios, C. Bielza, and A. Mateos. *Fundamentos de los sistemas de ayuda a la decisión*. RA-MA, 2001.
- [75] D. Rosaci and G.M.L. Sarné. Masha: A multi-agent system handling user and device adaptivity of web sites. *User Model User-Adap Inter*, 16(5):435–462, 2006.
- [76] M. Roubens. Fuzzy sets and decision analysis. *Fuzzy Sets and Systems*, 90(2):199–206, 1997.
- [77] S. Saint and J. R. Lawson. *Rules for Reaching Consensus. A Modern Approach to Decision Making*. Jossey-Bass, 1994.
- [78] J. Searle. *Speech Acts*. Cambridge University Press, 1969.
-

-
- [79] S. Staab and R. Studer. *Handbook on Ontologies (International Handbooks on Information Systems)*. Springer, 2009.
- [80] R.R. Yager. On orderer weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man a Cybernetics*, 18(1):183–190, 1988.
- [81] R.R. Yager. Applications and extensions of owa aggregations. *Int. J. Man-Machine Studies*, 27(1):103–132, 1992.
- [82] R.R. Yager. Non-numeric multi-criteria multi-person decision making. *Group Decision and Negotiation*, 2(1):81–93, 1993.
- [83] R.R. Yager. Fusion of ordinal information using weighted mean aggregation. *International Journal of Approximate Reasoning*, 12(1-2):35–52, 1998.
- [84] R.R. Yager. Penalizing strategic preference manipulation in multi-agent decision making. *IEEE Transactions on Fuzzy Systems*, 9(3):393–403, 2001.
- [85] L.A. Zadeh. Fuzzy logic equals computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2):103–111, 1996.
- [86] S. Zadrozny and J. Kacprzyk. An internet-based group decision and consensus reaching support system. *Applied Decision Support with Soft Computing (Studies in Fuzziness and Soft Computing)*, 124:263–275, Springer, 2003.
- [87] X.F. Zha, S.Y.E. Lim, and W.F. Lu. A knowledge intensive multi-agent framework for cooperative/colaborative design modeling and decision support of assemblies. *Journal of Integrated Desifn and Process Science*, 7(1):99–122, 2002.
-