

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
Facultad de Ciencias Químicas e Ingeniería  
Maestría y Doctorado en Ciencias e Ingeniería



**Comunicación Entre Agentes: Envío e Interpretación  
de Mensajes Usando Redes Semánticas y Lógica Difusa**

TESIS

PARA OBTENER EL GRADO DE  
MAESTRO EN CIENCIAS

*Presenta:*

**IVÁN ESPINOZA HERNÁNDEZ**

*Director:*

**DRA. DORA LUZ FLORES GUTIÉRREZ**

TIJUANA, BAJA CALIFORNIA, MÉXICO

SEPTIEMBRE 2011

# Índice general

<b>1. Introducción</b>	<b>II</b>
1.1. Descripción de la investigación . . . . .	III
1.2. Objetivo . . . . .	IV
1.3. Contribuciones . . . . .	IV
1.4. Organización del documento . . . . .	V
<b>2. Fundamentos Teóricos</b>	<b>1</b>
2.1. Agentes autónomos inteligentes . . . . .	1
2.1.1. Tipos de agentes . . . . .	4
2.1.2. La inteligencia en agentes . . . . .	5
2.1.3. Interacción entre agentes . . . . .	6
2.1.4. La importancia de los agentes . . . . .	10
2.2. Redes semánticas . . . . .	11
2.3. Lógica difusa . . . . .	14
2.3.1. Lógica difusa tipo 1 . . . . .	15

---

2.3.2. Lógica difusa tipo 2 . . . . .	19
<b>3. Modelo de Mensaje y Desarrollo de Herramientas para su Interpretación</b>	<b>22</b>
3.1. Plataforma MAS . . . . .	23
3.1.1. Wiinik . . . . .	23
3.2. Extensión de lógica difusa tipo 2 . . . . .	31
3.2.1. Extensión para NetLogo . . . . .	34
3.2.2. Extensión para Wiinik . . . . .	39
3.2.3. Editor gráfico para el diseño de sistemas de inferencia difusos tipo 2 .	41
<b>4. Casos de Estudio</b>	<b>44</b>
4.1. Caso Don Juan . . . . .	44
4.2. Caso de Trabajadores . . . . .	47
4.2.1. Descripción general de las interacciones . . . . .	48
4.2.2. Incorporación de lógica difusa . . . . .	50
<b>5. Conclusiones y Trabajo Futuro</b>	<b>54</b>
5.1. Conclusiones . . . . .	54
5.2. Trabajo futuro . . . . .	56

# Índice de figuras

2.1. Clasificación de agentes propuesta por S. Nwana . . . . .	5
2.2. Atributos que contiene un mensaje en FIPA-ACL . . . . .	8
2.3. Ejemplo del una red semántica. . . . .	13
2.4. Muestra de un conjunto difuso representado de manera discreta (a) y continua (b) . . . . .	16
2.5. Operaciones de conjuntos difusos . . . . .	18
2.6. Diagrama a bloques de un sistema de inferencia difuso . . . . .	18
2.7. Conjunto difuso tipo 2 viendo los ejes $x$ y $\mu$ . . . . .	20
2.8. Conjunto difuso tipo 2 con vista de los ejes $x$ , $\mu$ y $f(\mu)$ . . . . .	20
2.9. Conjuntos difusos tipo 2 generalizados y por intervalos . . . . .	21
3.1. Interacción y dependencia entre agentes . . . . .	25
3.2. Muestra de un guión para crear cuatro agentes . . . . .	30
3.3. Muestra de una regla de comportamiento . . . . .	30
3.4. Conjuntos difusos disponibles con sus parámetros . . . . .	32
3.5. Razonamiento difuso tipo 2 por intervalos de Mamdani . . . . .	40

---

3.6. Área de diseño principal del editor IT2Fuzzy . . . . .	41
3.7. Área de edición para conjuntos difusos . . . . .	42
3.8. Área de edición para las reglas difusas . . . . .	43
4.1. Intercambio de mensajes entre un agente “Don Juan” y una “víctima” . . .	46
4.2. Simulación de Don Juan vista con la terminal de Wiinik . . . . .	46
4.3. Simulación de Don Juan vista con las herramientas de Jade . . . . .	47
4.4. Caso de estudio de trabajadores, flujo de mensajes y estructura de la informa- ción intercambiada . . . . .	49
4.5. Código en Jess de la estructura de un FIS de un agente ambiente . . . . .	52
4.6. Código en Jess de la estructura de un FIS de un agente trabajador . . . . .	53

# Índice de tablas

3.1. Métodos disponibles en el objeto <i>FIS</i> . . . . .	33
3.2. Tipo de datos utilizados por la extensión de NetLogo . . . . .	34
3.3. Sintaxis de métodos disponibles de la extensión para NetLogo . . . . .	38
4.1. Valores que contiene el objeto BPNM para los perfiles de “Don Juan” y “víctima”	45
4.2. Componentes de un sistema de inferencia difuso utilizado en la generación de trabajos . . . . .	51

# Capítulo 1

## Introducción

Las interacciones entre humanos pueden ser ambiguas y complejas, igualmente el contenido y método de transmisión de mensajes. Una estrategia para estudiar fenómenos complejos es dividirlo en componentes separables que puedan ser atendidos de manera individual y de esta manera simplificar el problema. Se cuenta con agentes inteligentes que pueden simular características humanas y se usan con éxito en la simulación social [17], redes semánticas que son capaces de expresar ideas complejas y la lógica difusa tipo 2 nos aporta la posibilidad de interpretar valores lingüísticos.

Cada problema se debe resolver con las herramientas correctas, con esto en mente, las herramientas computacionales deben adaptarse al problema que se pretende resolver y no el problema a la herramienta. La computación tiene una gran versatilidad y por esta razón se ha podido aplicar con gran éxito en muchas áreas del conocimiento. Debido a su naturaleza de cálculos exactos es muy útil para procesar datos numéricos, realizar operaciones matemáticas, entre otras funciones de gran interés para las ciencias exactas.

Para las aplicaciones que requieren el estudio o interacción de seres humanos es conveniente elevar el nivel de abstracción con que se trabaja en la herramienta para aproximar a

la forma en que normalmente se interactúa con las cosas [19]. Esto puede requerir de optar por procesos no óptimos que faciliten el entendimiento. En el caso de las ciencias sociales, las herramientas necesarias dependerán mucho de la información que se requiera obtener y la teoría que se utilice para modelar los fenómenos. Un modelo estadístico es suficiente para estudiar tendencias en una población; un modelo cognitivo ayuda a dar detalles sobre decisiones individuales [16].

Debido a la gran cantidad de variables que conforman la transmisión e interpretación de mensajes, este trabajo se enfocará en la simulación de mensajes textuales simulando verbales por medio de agentes autónomos inteligentes combinado con redes semánticas, lógica difusa y modelos cognitivos. Lograr una interacción de este tipo permite estudiar múltiples aspectos como lo es la adquisición y almacenamiento de conocimiento, relaciones interpersonales de humanos, las acciones que pueden alterar el significado de un mensaje y la forma de interpretarse.

## 1.1. Descripción de la investigación

La base de esta investigación es la hipótesis de que es posible transmitir mensajes ambiguos y ser interpretados entre agentes autónomos inteligentes tomando en cuenta características como la imprecisión del lenguaje, percepción del transmisor hacia el receptor, perfiles de comportamiento entre otras variables. Modelar la comunicación humana presenta el problema de ser tan variada en temas como en forma de transmitir y recibir. Cada individuo tiene un perfil único, creencias, valores y objetivos que lo hacen prestar más atención a ciertas cosas en ciertas circunstancias y este perfil evoluciona con cada interacción que se tenga. Hay cosas que no se pueden decir con palabras, para comunicar ideas se usan gestos, miradas e incluso objetos que se poseen pueden decir algo a quien se fije, intencionalmente o no [5, 20, 21].

Con las herramientas computacionales disponibles es posible recrear y simular ciertos aspectos de la comunicación con ciertas restricciones, como lo serían perfiles simples, situaciones específicas y conocimiento del mundo limitado, esto se ha demostrado en trabajos anteriores [13, 3]. Los agentes autónomos inteligentes tienen características que facilitan simular las interacciones y autonomía de las personas, incluyendo sus propias creencias y objetivos. La complejidad de la información transmitida y almacenada puede ser representada por redes semánticas y la ambigüedad de los mensajes se pueden interpretar en parte con lógica difusa.

## 1.2. Objetivo

El objetivo de esta investigación es proponer una estructura de mensaje que permita transmitir e interpretar información generalizada y ambigua entre agentes autónomos inteligentes simulando interacciones entre humanos. Esto con el fin de ayudar al estudio de la comunicación en las ciencias sociales y que se pueda aplicar el conocimiento en aplicaciones que requieran interacción humano computadora. Se usará como base a los agentes autónomos, redes semánticas, lógica difusa y teorías psicológicas para el desarrollo de herramientas y casos de estudio.

## 1.3. Contribuciones

Las contribuciones de esta investigación son:

- la propuesta de una estructura de mensaje que tome en cuenta características del lenguaje natural
- el desarrollo de herramientas que permitan crear e interpretar mensajes ambiguos

- un análisis sobre el uso de este modelo mediante casos de estudio

## 1.4. Organización del documento

En el capítulo 2 se presenta la teoría que proporciona los fundamentos para esta investigación. Se introduce el tema de agentes con un enfoque especial en los métodos de comunicación, las formas de estructurar los mensajes para almacenar y transmitir y su interpretación. Otros temas que se revisan incluyen a las redes semánticas y la lógica difusa tipo 2. Estos temas como áreas de estudio individuales son complejos y tienen aplicaciones variadas. Aplicarlos en conjunto genera grandes posibilidades aplicables al estudio de las ciencias sociales.

En el capítulo 3 se introduce el modelo propuesto que permite transmitir e interpretar mensajes ambiguos. Se discute las capacidades que ofrece, detalles de su utilidad y la forma en que se mezclan diversas técnicas computacionales para simular situaciones específicas y detallar comportamientos individuales.

El capítulo 4 presenta algunos casos de estudio en el que se aplica el modelo propuesto. Se da una explicación de los escenarios, las herramientas utilizadas, se explica la interacción que ocurre entre los agentes y se explica el significado de lo que ocurre.

Por último, en el capítulo 5 se presentan las conclusiones, los resultados de la investigación y se menciona el trabajo futuro.

# Capítulo 2

## Fundamentos Teóricos

### 2.1. Agentes autónomos inteligentes

No existe una definición específica sobre lo que es un agente, solamente hay una lista de características que pueden poseer y no es necesario contar con todas. Como introducción, un agente se puede describir como un programa, ya sea independiente o parte de otro programa, que es capaz de interactuar y reaccionar en un ambiente y también comunicarse con otros agentes [17]. El ambiente se define como cualquier lugar donde pueda existir un agente, ya sea físico o virtual [32]. Algunas características consideradas esenciales para un agente son: autonomía, habilidad social, reactividad y proactividad.

La autonomía significa que un agente debe de ser capaz de tomar decisiones y actuar sin la necesidad de un controlador externo, ya sea otro agente o un humano. De cierta forma un agente nunca hace algo que no sea permitido, todo su comportamiento esta de alguna manera predefinido cuando fue programado. Aunque algunos agentes más avanzados pudieran generar sus propias estrategias para resolver problemas, siempre estarán siguiendo instrucciones predefinidas, ya sea los métodos permitidos o los objetivos definidos.

Otra característica de un agente es poder comunicarse con otros agentes y para esto requiere de una habilidad social. El método estándar de comunicación es mediante el pase de mensajes entre agentes y por lo general lleva un formato comúnmente aceptado que contempla campos como el tipo de mensaje (orden, pregunta, petición, etc.), receptores del mensaje, emisor y el mensaje en sí, entre otros campos. Un agente no necesariamente se tiene que limitar a esta forma de comunicación, en algunos casos se puede utilizar el ambiente para guardar e intercambiar información aunque esta no es una estrategia común por la complejidad que conlleva.

Un agente muchas veces se encontrará en ambientes dinámicos, esto significa que si se desea un agente autónomo en estas condiciones se debe poder ajustar a las nuevas condiciones, tiene que ser reactivo. Muchas simulaciones se realizan en ambientes completamente controlados, aunque sean cambiantes se tiene conocimiento sobre los límites a que se puede llegar y un agente puede ser programado para estar preparado para todas las condiciones posibles. Esta solución no es factible para ambientes reales o virtuales sobre los cuales no se tiene control. Para estas situaciones lo mejor que se puede hacer es hacer un estudio de los límites a que se puede llegar bajo condiciones extremas y normales, esto garantizará un cierto nivel de viabilidad sin considerar escenarios catastróficos, para los cuales ni los seres vivientes están preparados del todo.

Como consecuencia de la autonomía de un agente, se requiere de objetivos que cumplir, lo que le daría un propósito de ser al agente. Dependiendo del objetivo u objetivos a cumplir un agente debe de tomar las acciones apropiadas, tiene que estar trabajando activamente para cumplir sus metas. Un objetivo pudiera llevarlo a ser muy activo en sus intentos de cumplir o pudieran requerir que sea pasivo, solo observando parámetros y tomando medidas preventivas cuando sea requerido.

Un agente debe cumplir en gran medida con estas características, pero estas definiciones

son abstractas y no son buena guía para una implementación en un modelo computacional. Otra lista de características que es más aplicable sería: percepción, capacidades, memoria y políticas [16].

La percepción es la habilidad de poder tener conocimiento del ambiente en que se encuentra un agente, de los objetos que lo rodean, obstáculos, topografía y otros agentes. La complejidad de esta habilidad dependerá en gran medida del tipo de simulación y el detalle que se de en el ambiente pero como requerimiento mínimo se debe poder tener conocimiento de otros agentes, de otra manera sería difícil cumplir con la habilidad social.

Las capacidades de un agente pueden incluir acciones como movimiento, comunicación y algún tipo de interacción con el ambiente, ya sea escribir o tomar algún objeto. La única acción indispensable es la comunicación, esto es enviar y recibir mensajes de otros agentes. Cualquier acción adicional dependerá del tipo de simulación o aplicación que se quiera desarrollar.

Contar con una memoria de eventos o información importante es una capacidad muy útil para un agente autónomo y reactivo. La capacidad de recordar éxitos y fracasos y generar estrategias basándose en esa información le da una ventaja al agente que sepa explotar esa habilidad. Las políticas de acción de un agente pudieran ser influenciadas no solo por las percepciones presentes si no por experiencias previas ya sea del agente mismo o de otros agentes que compartieron información.

De acuerdo a [4] un agente se define formalmente como

$$A_g = \langle Sit, Act, Dat, f_{A_g} \rangle \quad (2.1)$$

donde

1. *Sit* es el conjunto de situaciones en que un agente se puede encontrar;

2. *Act* es el conjunto de acciones que un agente puede realizar;
3. *Dat* es el conjunto de valores que la memoria interna de un agente puede contener;
4.  $f_{Ag}$  es la función de comportamiento que depende de la situación actual y de los valores que se encuentren en la memoria del agente para poder decidir la siguiente acción. Su definición formal es  $f_{Ag} : Sit \times Dat \rightarrow Act$ .

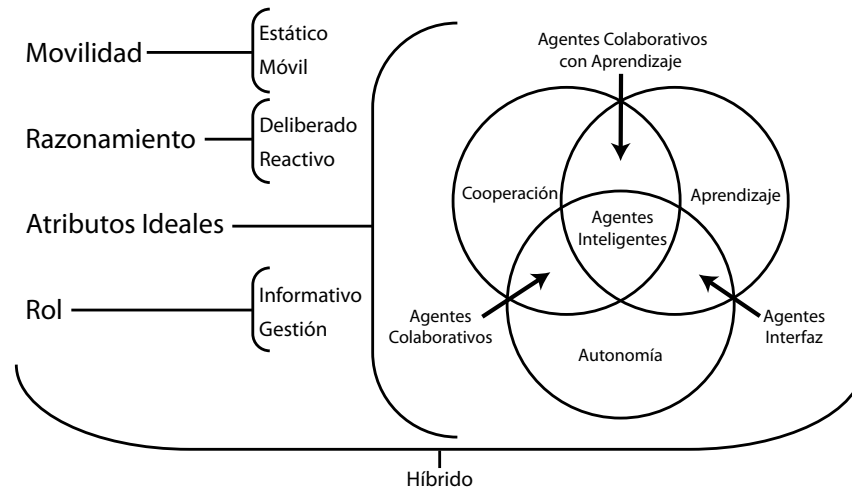
El conjunto *Act* limita a un agente a sólo ser capaz de realizar las acciones que se encuentren definidas dentro del conjunto. Si se alterara este conjunto tendría repercusiones notables en los resultados que arroje la función  $f_{Ag}$  y por ende cambiaría las experiencias que se viven dentro del ambiente. Variaciones de este parámetro sirve para simular diferentes tipos de seres que pueden tener habilidades muy distintas o muy similares pero no iguales.

Los agentes son utilizados en una gran variedad de aplicaciones pero por su naturaleza de ser independientes, moldeables a perfiles específicos y sus capacidades de ser sociables, es factible un uso de estos como herramienta de estudio en las ciencias sociales.

### 2.1.1. Tipos de agentes

Como ya se mencionó, no existe una única definición para *agente* pero sí se cuenta con una serie de características mínimas con las que se puede identificar a un agente como tal. No es necesario contar con todas y algunas pueden ser opuestas, esto lleva a la existencia de distintos tipos de agentes. La figura 2.1 muestra la clasificación de agentes propuesta por S. Nwana [25], esta cuenta con tres atributos principales los cuales pueden ser combinados y de estos se derivan cuatro tipos de agentes. Estos atributos no reflejan la totalidad de lo que es un agente y por lo tanto se cuenta con otra serie de atributos que definen el rol de un agente. Al ser posible la combinación de atributos, también es posible la combinación de tipos de agentes en un mismo sistema, esto es lo que comúnmente se le llama sistema

multiagente [17]. Existen alternativas a la clasificación propuestas por otros autores pero todas se basan en los mismos principios. Unos enfrentan la clasificación de agentes al asignar grados de agencia [15] mientras otros se basan en las tareas principales asignadas [11].



**Figura 2.1.** Clasificación de agentes propuesta por S. Nwana

La aplicación que se le dé será la que determine las características con las que debe de contar un agente. En el caso particular de la simulación social es de especial interés las características de movilidad, razonamiento deliberado y la combinación que forma a los agentes inteligentes. Al igual que los agentes, la inteligencia artificial es un área aun en estudio y con grandes avances, pero aún lejos de lo que se considera inteligencia en humanos y por esto sería irracional esperar que un agente denominado inteligente sea capaz de realizar tareas de la misma manera que una persona. Algunas investigaciones se han enfocado en el estudio de comportamiento complejo y emociones utilizando la simulación basada en agentes [14, 29].

### 2.1.2. La inteligencia en agentes

La inteligencia es una noción difícil de definir y algo controversial. Diversos autores la consideran como la habilidad de aprender de la experiencia, de poder adaptarse a nuevas situaciones y a los cambios en el ambiente o poder llevar a cabo pensamientos abstractos [27].

La enciclopedia de las ciencias cognitivas de MIT (del inglés, *Massachusetts Institute of Technology*) dice: “Un agente inteligente es un dispositivo que interactúa con su ambiente de una forma flexible y orientada a objetivos, reconociendo estados importantes del ambiente y actuando para lograr resultados deseados” [31].

Sin embargo, en la práctica la inteligencia es un atributo relativo y es evaluado en conexión con habilidades humanas. En comparación, algunos animales como una rata normalmente no se consideraría inteligente en comparación con un humano pero sí demuestran atributos que se reconocen estar por encima de una cucaracha.

Tomando estas consideraciones, un agente puede ser visto como inteligente si es capaz de realizar tareas con algún grado de complejidad, de forma intencional y que se adapte a los cambios en el ambiente. No obstante, la evaluación de este comportamiento es evaluado de manera arbitraria por humanos y por lo tanto la inteligencia es una propiedad asignada de manera subjetiva [10].

### 2.1.3. Interacción entre agentes

Una característica esencial de los agentes es la capacidad de comunicar e interactuar, ya sea con otros agentes, el ambiente o algún humano. Para cumplir este objetivo es necesario contar con algunos elementos esenciales como lo es un lenguaje común, un formato común para el contenido de la comunicación y una ontología común [7]. El reto de la comunicación se hereda de los estudios en otras áreas, principalmente en las ciencias sociales y de estas se han encontrado soluciones o guías para el camino a seguir.

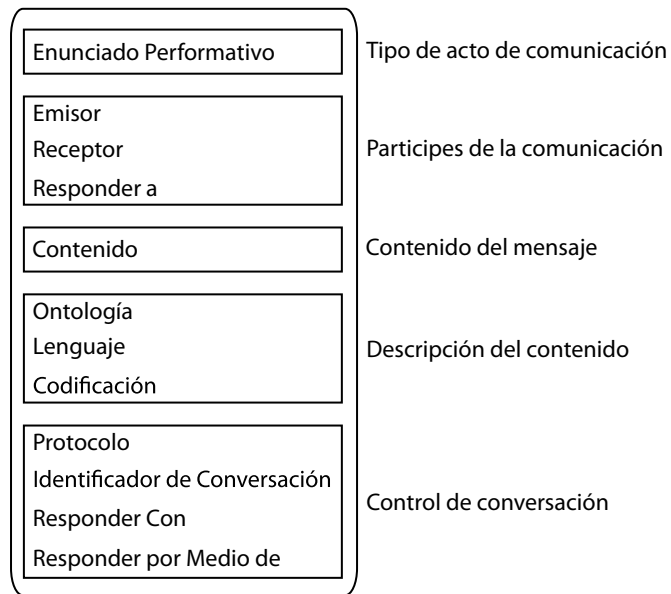
## Lenguajes de comunicación

Dado que es posible crear una variedad de agentes con características y objetivos distintos, se vuelve necesario llegar a un acuerdo en un lenguaje común que facilite la interoperabilidad y el desarrollo de los agentes. Con el tiempo y los avances por diversos grupos se han desarrollado lenguajes de comunicación denominados ACL (por el inglés, *Agent Communication Language*) entre los cuales destacan los lenguajes KQML (por el inglés, *Knowledge Query Manipulation Language*) y FIPA-ACL (por el inglés, *Foundation for Intelligent Physical Agents - Agent Communication Language*) [7, 8]. Algunas características propuestas por [23] que contienen estos lenguajes son: forma, contenido, semántica y ambiente.

La forma se refiere a la sintaxis del mensaje, debe de ser fácil de leer ya sea para una computadora o para un humano. En el contenido se hace la distinción entre la expresión de actos comunicativos y lo que se comunica, la semántica representa las características deseables de la semántica de cualquier otro lenguaje natural y el ambiente hace énfasis en la heterogeneidad que puede existir y el dinamismo. Otras características, con respecto a lo técnico, son: implementación, interconectividad y confiabilidad.

La figura 2.2 muestra los atributos con los que cuenta un mensaje de la especificación FIPA-ACL [8]. En esta se muestra un atributo muy importante que se basa en teorías lingüísticas, el denominado enunciado performativo. Este indica el tipo de mensaje que se envía en el contenido como puede ser una petición, una pregunta, aceptar o rechazar entre otros. Al incluir esta información se le da cierto sentido al mensaje dentro de un contexto, al existir distintos contextos posibles el atributo de ontología se encarga de describir la relación que existe entre los conceptos que se conocen. El resto de los atributos son meramente para formalizar el protocolo y facilitar el envío y recepción.

A pesar de basarse en fundamentos lingüísticos y ser técnicas funcionales para la comunicación entre agentes, los ACL actuales como tal fallan en captar gran parte de la complejidad



**Figura 2.2.** Atributos que contiene un mensaje en FIPA-ACL

que se presenta en descifrar el significado en una conversación natural. Esto se debe en parte al diseño de los mismos, nunca fue la intención tomar en cuenta fenómenos complejos como lo es una broma, sarcasmo o detalles que influyen en el significado como el tono de voz, gestos corporales al hablar, circunstancias en que se dice algo entre otros. Lo que es importante notar es que los ACL pueden ser utilizados como base y extenderse al incluir mecanismos que puedan dar mayor información sobre un mensaje [13], esta información puede provenir del mismo agente emisor o de sensores en el agente receptor.

## Ambientes

El ambiente es cualquier lugar en que puede existir y funcionar un agente, es el medio por donde interactúan y se comunican [17]. Este espacio puede ser físico o virtual dependiendo de la implementación. No existe otra característica esencial pero pueden llegar a ser espacios complejos al permitir interacciones con objetos o limitar el movimiento de los agentes mediante reglas a seguir. Las reglas que rigen un ambiente pueden ser basadas en las que se

conocen del mundo real o pueden describir otro tipo de universo ficticio.

Dependiendo a que entidad se le denomina agente será lo que identifique su ambiente. Como ejemplo si se tiene una red de sensores distribuidos, un agente puede ser una entidad de software que puede moverse entre sensores colectando información y supervisando su buen funcionamiento, en este caso el ambiente del agente es un sensor individual y su interacción con este se limita a coleccionar información captada y la supervisión del estado del sensor. En cambio, si se considera a un sensor individual como un agente, entonces su ambiente es cualquier lugar donde sea puesto en funcionamiento, se podrá comunicar con otros sensores y no será capaz de movimiento autónomo. Cuando un ambiente es virtual, entonces es habitado por agentes de software, cuando es físico es habitado por agentes físicos artificiales [36].

### Sistemas multiagentes

Por cuestiones de simplificar, es deseable tener una solución centralizada que cuente con toda la información y pueda ser optimizada. En muchos casos esto no es posible ya que se puede contar con recursos descentralizados con acceso restringido en el caso de aplicaciones industriales. En el caso de fenómenos sociales, se cuenta con actores distintos con motivaciones y objetivos diversos que no es práctico o deseable modelar en una entidad única. Los agentes al ser entidades de software autónomas con motivación a actuar en base a objetivos preestablecidos son una opción válida para enfrentar este tipo de situaciones.

Un sistema multiagente MAS (por el inglés, *Multi Agent System*) proporciona un ambiente común en el cual distintos agentes pueden existir, interactuar y comunicarse, facilitando su implementación, observación y manejo. Generalmente son ambientes abiertos que no cuentan un diseño centralizado y dan espacio a la existencia de agentes autónomos inteligentes distribuidos [37].

Algunas características mencionadas en [32] con que cuenta un MAS y que afectan a los

agentes que lo habitan son:

1. Conocimiento: que tanta información del ambiente está disponible a los agentes
2. Predecible: que tanto puede ser predicho por un agente
3. Controlable: hasta que punto un agente puede modificar el ambiente
4. Historial: si los estados futuros dependen de la historia completa o únicamente del estado actual
5. Teleológico: si hay partes intencionales
6. Tiempo Real: si el ambiente puede cambiar mientras un agente esta en un estado de deliberación

#### 2.1.4. La importancia de los agentes

En un principio el trabajo de desarrollo en el área fue motivado por deseos de mejorar las técnicas de la inteligencia artificial distribuida [37]. Aunque compartan características con otro tipo de sistemas como objetos o sistemas expertos, es importante notar las diferencias pues así es como resaltan las ventajas que pueden ofrecer.

Un agente puede ser caracterizado como un objeto especializado, ambos contienen una descripción interna de sus cualidades y funciones, ambos son capaces de comunicarse, pueden ser móviles y persistentes. La diferencia clave es la autonomía y el lenguaje con que se comunican. El conocimiento que contiene un agente puede ser visto de forma similar a lo que se tiene en un sistema experto salvo que este último no es capaz de interactuar con un ambiente y está limitado a consultas de información, a diferencia de un agente el cual sí puede tener un efecto sobre su ambiente y no necesariamente cuenta con toda la información

requerida. Una de las características de un sistema multiagente es la descentralización de información.

El campo de estudio y aplicación de los agentes es diverso, se alimenta de distintas áreas como la lingüística, sociología, inteligencia artificial entre otras y aporta resultados al permitir experimentar con teorías de estas mismas áreas o al desarrollo de aplicaciones comerciales. Entre sus diversas aplicaciones se encuentra el desarrollo de sistemas distribuidos, novedosos sistemas para la búsqueda de información y el modelado de sistemas complejos como lo son los fenómenos sociales o problemas de optimización.

## 2.2. Redes semánticas

Una red semántica es una forma de representación que usa grafos de nodos etiquetados y aristas dirigidas etiquetadas para representar el conocimiento. Los nodos representan ya sea objetos, conceptos o eventos y las aristas representan las relaciones entre los nodos [28]. Como una extensión de grafos dirigidos, las redes semánticas pueden ser definidas como

$$G = (V, E) \tag{2.2}$$

donde  $V$  es el conjunto de nodos y  $E$  es el conjunto de aristas denotando las distintas uniones entre los nodos. Distintos tipos de redes semánticas han sido definidas tales como IS-A, grafos conceptuales y marcos [1, 22, 34]. A continuación se presenta una serie de tipos de redes semánticas descritas por [33]:

1. Las *redes de definición* enfatizan el subtipo o la relación es-un(a) entre un tipo de concepto y un subtipo recién definido. La red resultante, también llamada jerarquía generalizada o de subsunción, implementa la regla de herencia para copiar las propiedades

definidas de un supertipo a todos sus subtipos. Ya que las definiciones son verdaderas por definición, a menudo se asume que la información en estas redes es necesariamente verdadera.

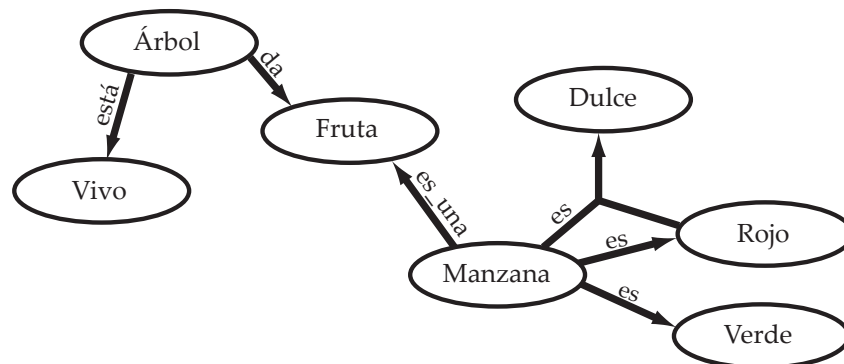
2. Las *redes de aserción* están diseñadas para aseverar proposiciones. A diferencia de las redes de definición, se supone que la información en una red de aserción es contingentemente cierta, a menos que sea explícitamente marcada con un operador modal. Algunas redes de aserción han sido propuestas como modelos para las estructuras conceptuales que subyacen la semántica del lenguaje natural.
3. Las *redes de implicación* usan la implicación como la relación primaria para conectar nodos. Pueden ser utilizadas en la representación de patrones de creencias, causalidad o inferencias.
4. Las *redes ejecutables* incluyen algunos mecanismos, como el pase de marcadores o procedimientos adjuntos, que pueden realizar inferencias, pasar mensajes o buscar patrones y asociaciones.
5. Las *redes de aprendizaje* construyen o extienden sus representaciones al adquirir conocimiento de ejemplos. El nuevo conocimiento puede cambiar la red antigua agregando y/o eliminando nodos y aristas o al modificar valores numéricos, llamados pesos, asociados con los nodos y arcos.
6. Las *redes híbridas* combinan dos o más de las técnicas mencionadas, ya sea en una única red o por separado pero con una interacción muy cercana.

Las implementaciones de redes semánticas para computadoras fueron desarrolladas inicialmente para la inteligencia artificial y para las traducciones por computadora, pero versiones anteriores han sido por mucho tiempo utilizadas en la filosofía, psicología y la lingüística. Lo que es común en todas las redes semánticas es una representación declarativa gráfica que

puede ser utilizada ya sea para representar conocimiento o para apoyar sistemas automatizados a razonar sobre el conocimiento. Algunas versiones son altamente informales, pero otras versiones son sistemas de lógica formalmente definidos [33].

Cuando se trata de lingüística, es importante tener un entendimiento claro de la distinción entre conceptos y términos. Un término es “algo que escuchamos o vemos, puede ser una secuencia de sonidos (en el habla) o una secuencia de letras (en la escritura). En el lenguaje de señas para los sordos, un término es una combinación de gestos”. En cambio, un concepto es “una entidad abstracta, no los sonidos que escuchamos o las letras o los gestos que vemos, si no lo que entendemos al escuchar los sonidos o al ver los términos en escritos o gestos” [6].

El poder entender los términos usados en algún contexto es útil para herramientas autónomas para resolver problemas específicos. El lograr que una herramienta computacional, ya sea agente o sistema experto, demuestre aunque sea en una capacidad limitada el entendimiento de conceptos es un gran logro e implica el logro de una asociación entre el ambiente y el lenguaje usado para describirlo. Desafortunadamente una de las grandes fortalezas de las redes semánticas para la representación del conocimiento también es una de sus grandes debilidades. Casi cualquier idea que pueda ser entendida puede ser expresada en una red semántica pero depende en gran medida del diseñador y no existe una única forma correcta de la representación, la extracción de conocimiento, transmisión e interpretación.



**Figura 2.3.** Ejemplo de una red semántica.

La figura 2.3 muestra una red semántica que describe la relación que existe entre manzanas y algunos elementos con que tiene relación. Esta red representa una porción del conocimiento que se puede adquirir sobre un tema, en este caso, manzanas. Se describe que las manzanas crecen en árboles y que existe por lo menos dos tipos de manzanas que tienen sabor distinto, aunque no se mencionan ambos, sí se condiciona el sabor que se conoce con un tipo de manzana.

### 2.3. Lógica difusa

Las herramientas computacionales son útiles en ayudar a crear modelos que representen aspectos del mundo real y gran parte de estos modelos que pretenden describir y predecir fenómenos se basan en las matemáticas, en conceptos discretos y absolutos. Este enfoque funciona en algunas áreas pero fracasa en otras ya que se basa en conceptos totalmente distintos a la realidad. Una de estas áreas incluye el manejo de incertidumbre que no se basa en variables aleatorias como es el caso del parentesco de un objeto o concepto a otro o en mediciones no exactas (grande, pequeño, mucho).

En la teoría de conjuntos clásica se usan valores absolutos, es decir, un elemento es o no es parte de un conjunto, no existen valores intermedios de pertenencia. Por ejemplo, un conjunto clásico  $A$  de números reales mayores a 6 puede ser representado como

$$A = \{x|x > 6\} \tag{2.3}$$

donde existe el límite claro y sin ambigüedad de 6 tal que si  $x$  es mayor a este número, entonces  $x$  pertenece al conjunto  $A$ , de lo contrario  $x$  no pertenece al conjunto. Aunque los conjuntos clásicos son adecuados para diversas aplicaciones, y han demostrado ser una herramienta importante en matemáticas y las ciencias computacionales, no reflejan la naturaleza de los

conceptos y pensamientos humanos, los cuales tienden a ser abstractos e imprecisos. Para ilustrar este punto, de forma matemática se puede expresar el conjunto de personas altas como el conjunto de personas cuya altura sea mayor a 6 pies; sea este el conjunto descrito por la ecuación 2.3, si  $A =$  “persona alta” y  $x =$  “altura”. Debido a la naturaleza del conjunto clásico, esto definiría a una persona que mide 6.001 pies como alta, pero no así con una persona que mide 5.999 pies. Esta clasificación no es aplicable y la falla viene de la brusca transición entre inclusión o exclusión de un conjunto.

En contraste a los conjuntos clásicos, un conjunto difuso es, como su nombre lo implica, un conjunto sin límites claros. Esto es, la transición de “pertenece al conjunto” a “no pertenece al conjunto” es gradual y es caracterizada por funciones de membrecía que le dan flexibilidad al modelar expresiones lingüísticas comunes, tales como “el agua está caliente” o “la temperatura está alta”. Tales imprecisiones en las definiciones de conjuntos o clases juegan un papel importante en el pensamiento humano, particularmente en el dominio de reconocimiento de patrones, comunicación de información y abstracción. Se debe notar que lo difuso no viene de la aleatoriedad de los elementos del conjunto sino de la naturaleza incierta e imprecisa de pensamientos y conceptos abstractos [41, 18].

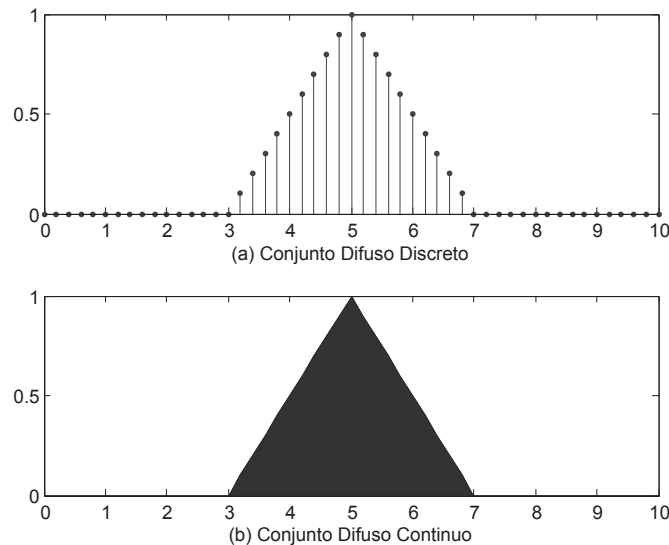
### 2.3.1. Lógica difusa tipo 1

Sea  $X$  un universo de valores válidos y  $A$  un conjunto tal que  $A \subseteq X$  donde cada elemento de  $X$  puede o no pertenecer a  $A$  y esto se puede representar como una serie de pares ordenados  $(x, 0)$  o  $(x, 1)$  que representan ya sea  $x \notin A$  o  $x \in A$ . En un conjunto difuso es válido tener cualquier valor de pertenencia entre 0 y 1 y se representa formalmente como

$$A = \{(x, \mu_A(x)) | x \in X\} \quad (2.4)$$

donde  $\mu_A(x)$  denomina la función de membrecía o MF (del inglés, *Membership Function*). La función de membrecía contiene un mapa de cada valor de  $X$  con su valor de pertenencia correspondiente con valor de entre 0 y 1. Un conjunto difuso puede ser definido de forma discreta o continua. Un conjunto discreto es un conjunto finito que contiene un valor de pertenencia por cada elemento del conjunto, este conjunto abarca un espacio definido con intervalos definidos y los valores intermedios no están definidos. Un conjunto continuo se define por medio de una función y es posible obtener un valor de pertenencia para cualquier elemento del universo.

La figura 2.4 muestra la representación gráfica de un conjunto difuso de forma discreta y continua. Ambos conjuntos son de forma triangular y fueron creadas usando la misma función de membrecía. Se puede trabajar con ambas formas pero por cuestiones técnicas, en una implementación de un sistema difuso es común que se utilice versiones discretas de los conjuntos. Entre mayor sea el número de divisiones del universo, será más aproximado a su versión continua y por lo tanto se podrán obtener mejores resultados.



**Figura 2.4.** Muestra de un conjunto difuso representado de manera discreta (a) y continua (b)

Para ejemplificar, si se quiere clasificar cualquier cosa con un valor numérico del 0 al

10 donde 0 es muy malo y 10 es excelente, entonces se cuenta con un universo  $X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ . Las variables lingüísticas como “muy malo” y “excelente” son subjetivas y se nombran por lo que mejor describa el grupo de valores que abarca un conjunto en el contexto en que se utiliza. Como se muestra en la figura 2.4, el conjunto abarca una serie de valores intermedios en el universo lo cual se pudiera denominar, en el caso de una opinión, “indiferente” o en el caso de rendimiento puede ser “mediocre” o “regular”. En el ejemplo se muestra un conjunto con 51 valores, un conjunto de forma equivalente con 11 valores se puede representar como  $A = \{(0/0), (0/1), (0/2), (0/3), (.5/4), (1/5), (.5/6), (0/7), (0/8), (0/9), (0/10)\}$ .

La creación de un conjunto difuso depende de dos aspectos: la identificación de un universo de valores adecuados y de especificar una función de membresía adecuada. La elección de la función de membresía es un proceso subjetivo, lo que significa que diferentes personas pueden llegar a conclusiones distintas sobre el mismo concepto. Esta subjetividad se deriva de las diferencias individuales en la percepción y expresión de conceptos abstractos y muy poco que ver con aleatoriedad. Por lo tanto, la subjetividad y no aleatoriedad de un conjunto difuso es la diferencia principal entre el estudio de conjuntos difusos y la teoría de la probabilidad [18].

### Razonamiento difuso

Al ser una extensión de conjuntos clásicos, es posible realizar operaciones similares como lo es la unión, disyunción y negación en conjunto difuso [40], la figura 2.5 muestra estas operaciones de forma gráfica. Estas operaciones son utilizadas para transformar conjuntos difusos en lo que se denomina un sistema de inferencia difuso o FIS (por el inglés, *Fuzzy Inference System*). Este funciona a base de reglas lógicas que pueden trabajar con valores numéricos o difusos como entrada, se evalúan las reglas y se juntan los resultados individuales para formar lo que es la salida difusa, si se desea un valor numérico se debe pasar por un

proceso de defusificación. La figura 2.6 muestra un diagrama a bloques de la estructura clásica de un sistema de inferencia difuso.

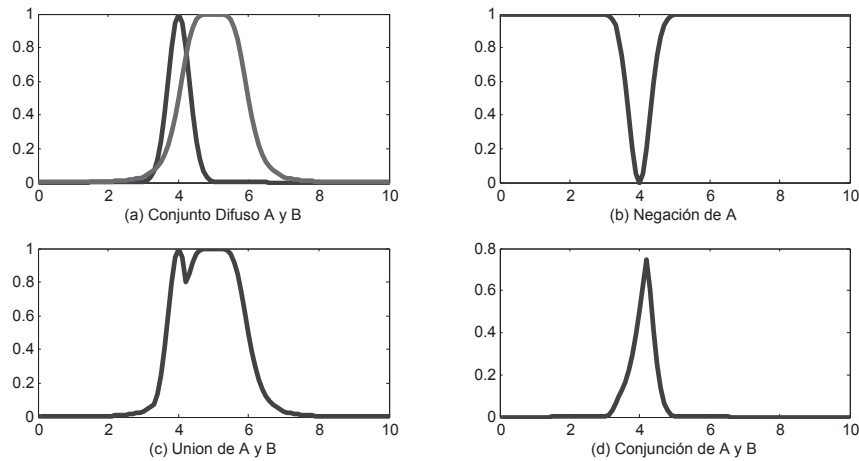


Figura 2.5. Operaciones de conjuntos difusos

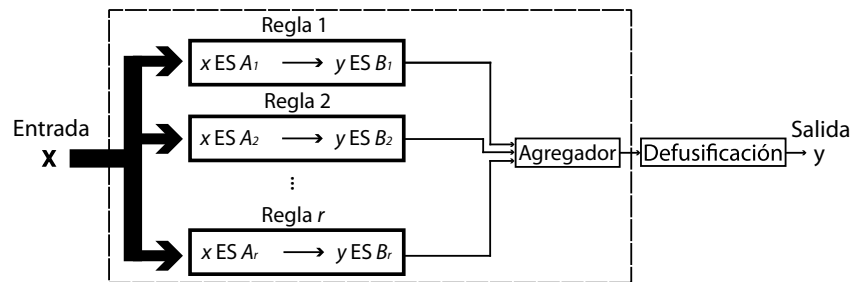


Figura 2.6. Diagrama a bloques de un sistema de inferencia difuso

Con los años se han desarrollado distintos métodos que siguen el modelo general de un sistema difuso y cada uno tiene sus ventajas y desventajas. Los métodos aplican a cada componente de un FIS por lo cual es posible hacer combinaciones. Entre los tipos de razonamiento más comunes se encuentran los métodos de Mamdani, Sugeno y Tsukamoto [18].

### 2.3.2. Lógica difusa tipo 2

El concepto de un conjunto difuso tipo 2 fue introducido por Zadeh [40] como una extensión del concepto de conjuntos difusos ordinario tipo 1. Un conjunto difuso tipo 2 se caracteriza por una función de membresía cuyo valor de pertenencia por cada elemento del universo es una función de membresía en el rango  $[0, 1]$ , a diferencia de los conjuntos difusos tipo 1 donde el valor de membresía es un valor numérico en el rango de  $[0, 1]$ .

Un detalle que tienen los conjuntos difusos tipo 1 es que una vez definida la función de membresía para un concepto, esta se basa en la opinión subjetiva de uno o más individuos y no muestra mas que un valor por cada elemento del universo. Al hacer esto, se pierde parte de la ambigüedad con que se tratan algunos conceptos, en especial en los que personas pueden tener una opinión un poco distinta y todas son consideradas válidas. Los conjuntos difusos tipo 2 permiten manejar incertidumbres lingüísticas así como incertidumbres numéricas.

Un conjunto difuso tipo 2 se puede expresar como

$$\tilde{A} = \{(x, u), \mu_{\tilde{A}}(x, u)\} \quad (2.5)$$

el cual es muy similar a un conjunto tipo 1 mostrado en la ecuación 2.4. Un ejemplo práctico de un conjunto difuso tipo 2 puede ser

$$\tilde{A} = \left\{ \begin{array}{l} \{0/0 + 1/0.1 + 0/0.2\}/0 + \{0/0.4 + 1/0.5 + 0/0.6\}/1 + \\ \{0/0.8 + 1/0.9 + 0/1\}/2 + \{0/0.4 + 1/0.5 + 0/0.6\}/3 \end{array} \right\} \quad (2.6)$$

y de forma gráfica se puede observar en las figuras 2.7 y 2.8

Una gran desventaja de este tipo de conjuntos es la cantidad de cálculos requeridos para obtener resultados confiables. En el ejemplo mostrado se cuenta con un número reducido de

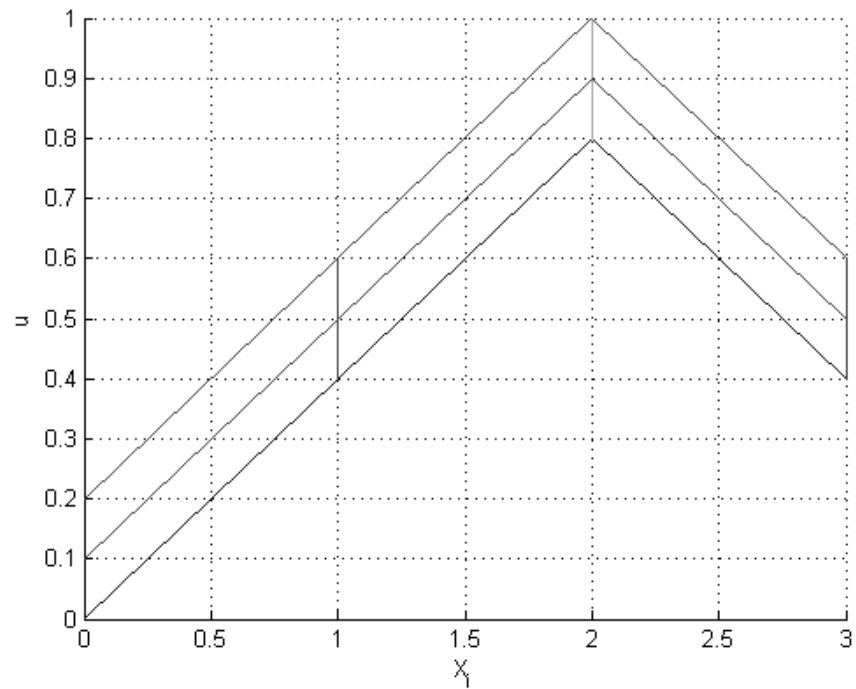


Figura 2.7. Conjunto difuso tipo 2 viendo los ejes  $x$  y  $\mu$

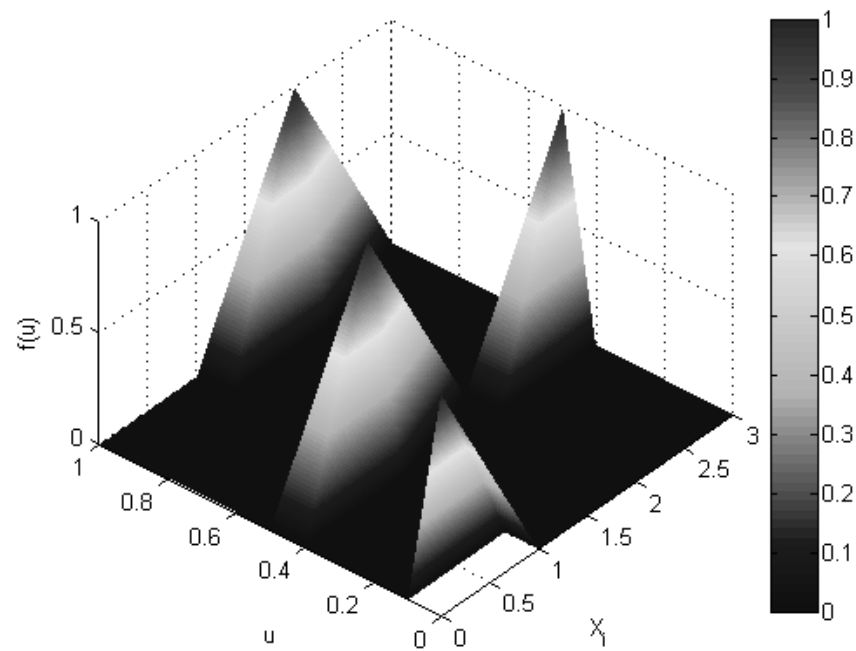
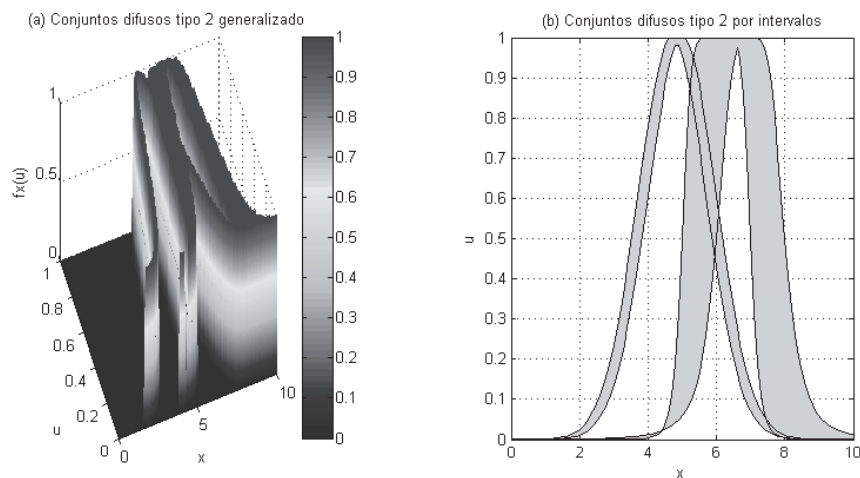


Figura 2.8. Conjunto difuso tipo 2 con vista de los ejes  $x$ ,  $\mu$  y  $f(\mu)$

divisiones, tres divisiones en cada elemento y cuatro elementos en el conjunto. Si se desea hacer una reducción de tipo por el método del centroide se requieren  $3^4$  operaciones lo que sería un total de 81. Elevando un poco la cantidad de divisiones a diez por elemento con 10 elementos se requeriría la cantidad de  $10^{10}$  operaciones lo que es 10,000,000,000 (diez billones). Al considerar que un conjunto difuso comúnmente contará con por lo menos 150 divisiones para obtener buenos resultados, se puede concluir que no es factible el uso de este método en cuestiones prácticas.

Sin sacrificar mucho en cuestión de modelado de incertidumbre, se puede trabajar con una alternativa que es la lógica difusa tipo 2 por intervalos. Esta simplifica las operaciones al únicamente tomar en cuenta las porciones que tengan una pertenencia de 1. La figura 2.9 muestra a dos conjuntos representados en su forma general y reducidos a sus intervalos.



**Figura 2.9.** Conjuntos difusos tipo 2 generalizados y por intervalos

## Capítulo 3

# Modelo de Mensaje y Desarrollo de Herramientas para su Interpretación

Los humanos son seres sociales por naturaleza, se depende de otros para realizar actividades dentro de una comunidad y cada individuo juega un papel. Únicamente aquellos con los conocimientos y habilidades requeridas pueden realizar ciertas tareas de manera adecuada aun cuando la mayoría de los miembros serían físicamente capaces de realizar dicha tarea. La complejidad en las interacciones de cualquier sociedad moderna o primitiva va más allá de estas declaraciones pero proporcionan un buen punto de partida para el desarrollo de modelos de comportamiento y herramientas computacionales que permitan implementar agentes que puedan interactuar en ambientes limitados.

Adquirir y compartir información entre un grupo de personas es una parte vital en el desarrollo de cualquier sociedad. Cada descubrimiento llevará a un grupo un paso más cerca a una meta común y así asegurar su sobrevivencia y bienestar. El poder representar el conocimiento adquirido de un agente de tal manera que sea fácilmente aplicable a distintas áreas con alguna similitud aumentaría en gran medida la habilidad para resolver problemas.

Además, la habilidad de transmitir conocimiento entre individuos requiere de ciertas habilidades y un conocimiento común de conceptos básicos que puedan ser representados por sonidos, escritos o gestos. Simular estas habilidades en un ambiente limitado presenta muchos retos, entre los cuales está el representar cualquier tipo de conocimiento, la extracción de información relevante de ese conocimiento y poder transmitir de tal forma que sea entendido por todos los receptores.

## **3.1. Plataforma MAS**

El entorno determina las opciones disponibles y los objetivos que se pueden cumplir, entre otras cosas. En un entorno virtual se tiene una combinación de conceptos que se pueden basar en elementos de un ambiente físico o en reglas ficticias. En el caso de un entorno social, se representan las reglas que rigen las relaciones que existen entre los objetos y una situación en particular, lo que es apropiado e información similar. Un agente es capaz de tener una abstracción de su ambiente que abarca hasta el límite en que es capaz de interactuar con ese ambiente.

### **3.1.1. Wiinik**

Wiinik es una herramienta computacional desarrollada con el fin de apoyar el modelado de comportamientos humanos, sus interacciones y razonamiento con la intención de imitar las limitaciones que existen en el mundo real lo más cercano a como actualmente se entiende. Al combinar varias técnicas tales como MAS, programación basada en reglas, lógica difusa, entre otras, la herramienta es útil en distintos casos. Esta herramienta sigue un modelo de agentes similar al propuesto por [3] en donde cada agente consiste de un agente tipo *Persona* (del latín que significa máscara) y uno o más agentes cognitivos. Este tipo de arquitectura

permite que se puedan implementar y experimentar con distintas teorías psicológicas en las simulaciones como lo es la teoría de análisis transaccional [39]. Cada agente individual puede tener una composición única dando mayor diversidad al tipo de situaciones y teorías que pueden ser simuladas.

Se hace uso de la plataforma JADE (del inglés, *Java Agent Development*) [2] la cual es de código abierto y se encarga de todas las tareas de gestión de sistemas multiagentes o MAS (del inglés, *Multi Agent System*). Se eligió utilizar a JADE entre otras opciones ya que está implementado completamente en Java, es compatible con los estándares creados por la fundación para agentes físicos inteligentes [8] y su diseño como middle-ware permite un desarrollo rápido y eficiente. Las arquitecturas de agentes actuales no especifican la forma de que un agente debe tratar el conocimiento adquirido de su ambiente o como generar conocimiento de sus experiencias y la arquitectura presentada permite flexibilidad en este aspecto.

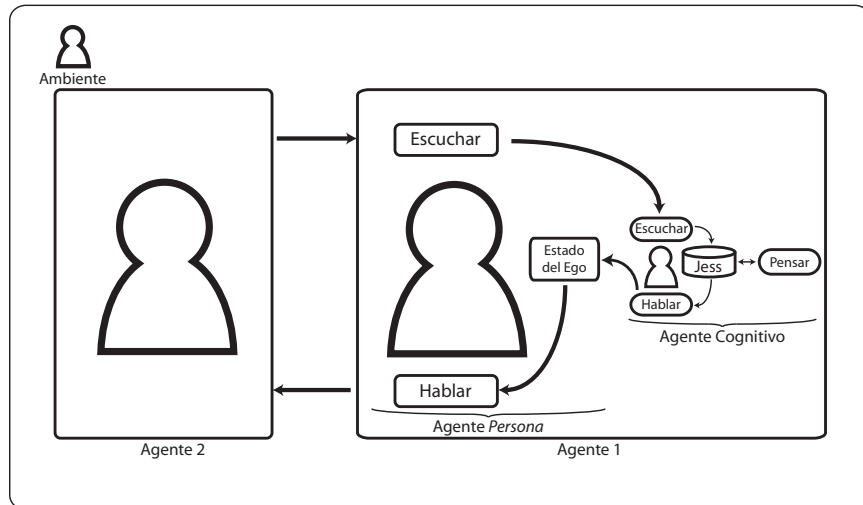
Uno de los motivos por el cual fue desarrollada la plataforma Wiinik es facilitar la experimentación con distintas teorías de comportamiento y técnicas de representación del conocimiento que puedan llevar a la creación de agentes que sean capaces de entender su ambiente. Al experimentar con distintas posibilidades ayuda a contestar preguntas sobre qué métodos funcionan mejor en circunstancias específicas y generales. Poder optimizar métodos generales a situaciones específicas permitiría a los agentes adaptarse a los cambios del ambiente y trabajar con información parcial.

#### **Arquitectura de agentes**

Con la intención de simular las interacciones humanas y razonamiento, el diseño de los agentes trata de imitar las limitaciones que existen en el mundo real y se modeló la comunicación y las mentes de los agentes basándose en distintas teorías pero con flexibilidad para

modificar. El concepto detrás del diseño es tener la mayor flexibilidad posible en el sentido de interacciones con el menor número de configuraciones o tipos de agentes. Esta flexibilidad se logra al diseñar un número limitado de agentes y delegar los detalles del comportamiento, almacenamiento de información y toma de decisiones a archivos de guiones que pueden ser manipulados por los mismos agentes.

Idealmente, cada acción que un agente sea capaz de simular que le permita interactuar con el ambiente, como lo es moverse y hablar, está modelado en distintos bloques de comportamiento (en la implementación en Java serían distintas clases) y únicamente son activados cuando las reglas de comportamiento lo permitan.



**Figura 3.1.** Interacción y dependencia entre agentes

Se cuenta con cinco tipos de agentes, entre los cuales destacan los tipos *Persona* y *Cognitivo*. Los agentes restantes tienen tareas muy específicas y están únicamente de soporte como lo es llevar el tiempo de una simulación, permitir interacción con un ambiente virtual entre otras tareas. La figura 3.1 muestra la dependencia que existe entre los agentes de tipo *Persona* y *Cognitivo* y la existencia de otros agentes como es el caso del ambiente. Los agentes principales siguen la arquitectura similar a la propuesta por [3]. En este modelo, cada agente está compuesto por un agente *Persona* y uno o más agentes *Cognitivos*, cada grupo

### 3. Modelo de Mensaje y Desarrollo de Herramientas para su Interpretación 26

de agentes sólo tiene conocimiento de los agentes tipo *Persona* y los agentes tipo *Cognitivo* de su grupo. Esto permite, entre otras cosas, implementar distintas teorías psicológicas como lo es la teoría de análisis transaccional.

El conjunto de acciones *Act* de la tupla 2.1 se define como

$$Act = \{Say, Listen, Think, Move\} \quad (3.1)$$

que son *decir*, *escuchar*, *pensar* y *mover* respectivamente. Cada elemento es implementado como un comportamiento de un agente en clases del tipo “behavior”.

1. *Say behavior*. Este comportamiento se activa cuando un agente necesita comunicarse. Se encarga de dar una forma apropiada al mensaje y enviarlo al receptor deseado.
2. *Listen behavior*. Se encarga de recibir todo mensaje destinado a este agente y puede reenviar a los agentes internos o agregar a la base de conocimiento. Únicamente los mensajes provenientes de agentes tipo *Persona* son tratados de esta manera.
3. *Think behavior*. Cada que exista nueva información, este comportamiento permitirá que se ejecute la máquina de inferencia y si existe alguna salida de las reglas será procesada.
4. *Move behavior*. Representa la habilidad de un agente para moverse entre distintos ambientes o en uno solo.

Dado que más de un agente cognitivo puede existir en un grupo de agentes y todos pueden tener instrucciones distintas para un evento, se vuelve necesario elegir entre todas una única acción a seguir. Este conflicto se enfrenta implementando el concepto de catexis que aporta la teoría de la psicodinámica y ayuda a determinar el tiempo que un agente invierte en cualquier acción o estado. La catexis se define como los procesos psicodinámicos que determinan la selección de distintas propuestas, posiciones, roles psicológicos o actividades que se ejecutan

en un tiempo determinado. Por medio de tres tipos de energía [35] (potencial, cinética y libre) se define la acción final en cualquier momento de decisión.

Los agentes cognitivos cuentan con un parámetro de interés por cada acción que pueden realizar y cada rol que pueden tomar. Conforme pase el tiempo o suceda algún evento, el interés que mantengan por la acción que se realice o estado en que se encuentren disminuirá mientras que el valor de otras acciones o estados aumenta, este aumento puede ser uniforme o variable. Cuando suceda que alguna acción potencial cuente con un valor mayor a la de la acción actual se da un cambio de acción a la que tenga mayor valor.

## **Mensajes**

Cuando se trata de situaciones sociales, se vuelve muy importante captar el significado completo de un mensaje. Esto requiere de ir más allá del significado literal y empezar a considerar el contexto y percepciones. Se hace uso de un modelo propuesto por [13] para representar datos sobre las percepciones que tienen los actores de un intercambio de información entre otros campos que juntos almacenan datos necesarios para la correcta interpretación de un mensaje en distintos contextos.

Un mensaje  $M$  está formalmente definido como una 7-tupla que contiene información que permite una interpretación más profunda de mensajes más allá del significado literal del contenido.  $M$  contiene información que se basa en percepciones y puede ser integrado a repositorios de conocimiento añadiendo información de contexto a cualquier evento y se define como

$$M = (Id, T, P, C, BPNM, A, Ag_{RS}) \quad (3.2)$$

donde:

1.  $Id$  es un campo de identificación único utilizado para asociar un mensaje con un evento y de esta manera permitir a un agente tratar con eventos concurrentes sin confusión.
2.  $T$  contiene el t3pico de la conversaci3n. Es utilizado para identificar el contexto en el cual un mensaje es utilizado para poder ser interpretado correctamente.
3.  $P$  es lo que se denomina *performative*, es lo que indica la raz3n, prop3sito o intenci3n del mensaje.
4.  $C$  es el contenido del mensaje. Este elemento puede ser un texto simple o tener un formato tal que pueda ser incorporado directamente a la base de conocimiento de un agente y procesado como informaci3n. Al no contar con un formato espec3fico permite tal flexibilidad que incluso puede llegar a contener nuevas reglas de comportamiento o modificar las existentes. En esta investigaci3n se experiment3 con el env3o de fragmentos de una red sem3ntica, la cual representa la informaci3n adquirida por los agentes de su entorno, as3 comunic3ndose de forma significativa en el sentido que se comparten conceptos complejos y enriquecidos por los otros componentes del mensaje.
5. El elemento BPNM (*Behavior Profile Node Message*) propuesto por [13] se define como  $BPNM = (\lambda, ES_i, ES_u, \pi_i, \pi_u)$  donde:
  - a)  $\lambda = [0, 1]$ .  $\lambda$  representa la percepci3n que un agente tiene sobre alg3n evento espec3fico. Consiste en un valor num3rico real con valores v3lidos entre 0 y 1 el cual es calculado usando l3gica difusa.
  - b)  $ES = \{Parent, Adult, Child\}$ .  $ES_i$  y  $ES_u$  representan el estado del ego en el que se encuentra un agente y el estado que percibe respectivamente. La manera en que un agente act3a, qu3 objetivos decide perseguir y qu3 informaci3n ser3 relevante en la toma de decisiones depende en gran medida de este valor.

- c)  $\pi = [0, 1]$ .  $\pi_i$  y  $\pi_u$  representan la postura psicológica que un agente tiene y la que percibe respectivamente. Consiste en un valor numérico real con valores válidos entre 0 y 1 el cual es calculado usando lógica difusa.
6.  $A$  se utiliza para poder identificar qué mensajes ya han sido procesados. Esto es con la intención de identificar mensajes anteriores permitiendo mantener un registro de las conversaciones que han ocurrido.
7.  $Ag_{RS}$  identifica al emisor o al receptor de un mensaje.

## Guiones

Las acciones de un agente están modeladas a través de guiones de comportamiento los cuales están compuestos por reglas lógicas y fragmentos de información. Al obtener las instrucciones de guiones de una máquina de inferencia, los agentes pueden ajustar su comportamiento individual sin afectar ninguna otra parte de su programación y esto se puede hacer mientras estén activos. Se utiliza a Jess [12] (por el inglés, *Java expert system shell*) como máquina de inferencia y es el componente principal en la toma de decisiones. Se eligió Jess ya que está implementado en Java y por lo tanto se puede integrar con la plataforma JADE.

Los guiones de Jess son utilizados en distintos tipos de agentes con propósitos que no se limitan a controlar las acciones durante una simulación. Algunas otras tareas incluyen facilitar la creación de agentes, la figura 3.2 muestra un ejemplo en como se puede especificar la creación de cuatro agentes utilizando guiones de Jess y la figura 3.3 muestra un fragmento de las reglas que definen el comportamiento de un agente.

Es posible especificar más de un guión para cada agente *Persona*. El hacer esto creará a un agente cognitivo por cada guión y cada uno recibirá toda entrada que llegue al agente *Persona* al cual están asignados. Para la toma de decisiones final en los agentes que cuentan

```
(assert (Load (Type TimerAgent)
              (Name "kronos")))
; -----
(assert (Load (Type PersonAgent)
              (Name "agent1")
              (Script ./Scripts/DonJuan/DonJuan.clp)))
; -----
(assert (Load (Type PersonAgent)
              (Name "agent2")
              (Script ./Scripts/DonJuan/Victim.clp)))
; -----
(assert (Load (Type PersonAgent)
              (Name "agent3")
              (Script ./Scripts/DonJuan/Victim.clp)))
```

**Figura 3.2.** Muestra de un gui3n para crear cuatro agentes

```
(defrule receive-job "receive job"
  ?h <-(Hear
          (attended ?a&:(= ?a FALSE))
          (id ?msgId)
          (performative "REQUEST")
          (topic "Work")
          (content ?job)
          (sender ?id))
          (me (aid ?aid) (bpNode ?bp))
  =>
  (bind ?jj (assert-string ?job))
  (modify ?jj (MID ?id))
  (printout t "Asserting job " ?job crlf))
```

**Figura 3.3.** Muestra de una regla de comportamiento

con m3s de un agente cognitivo, cada acci3n resultante en los agentes cognitivos individuales es comparado contra el resultado de los dem3s y por medio de un proceso de selecci3n, la acci3n m3s urgente resulta la ganadora.

#### Extensiones

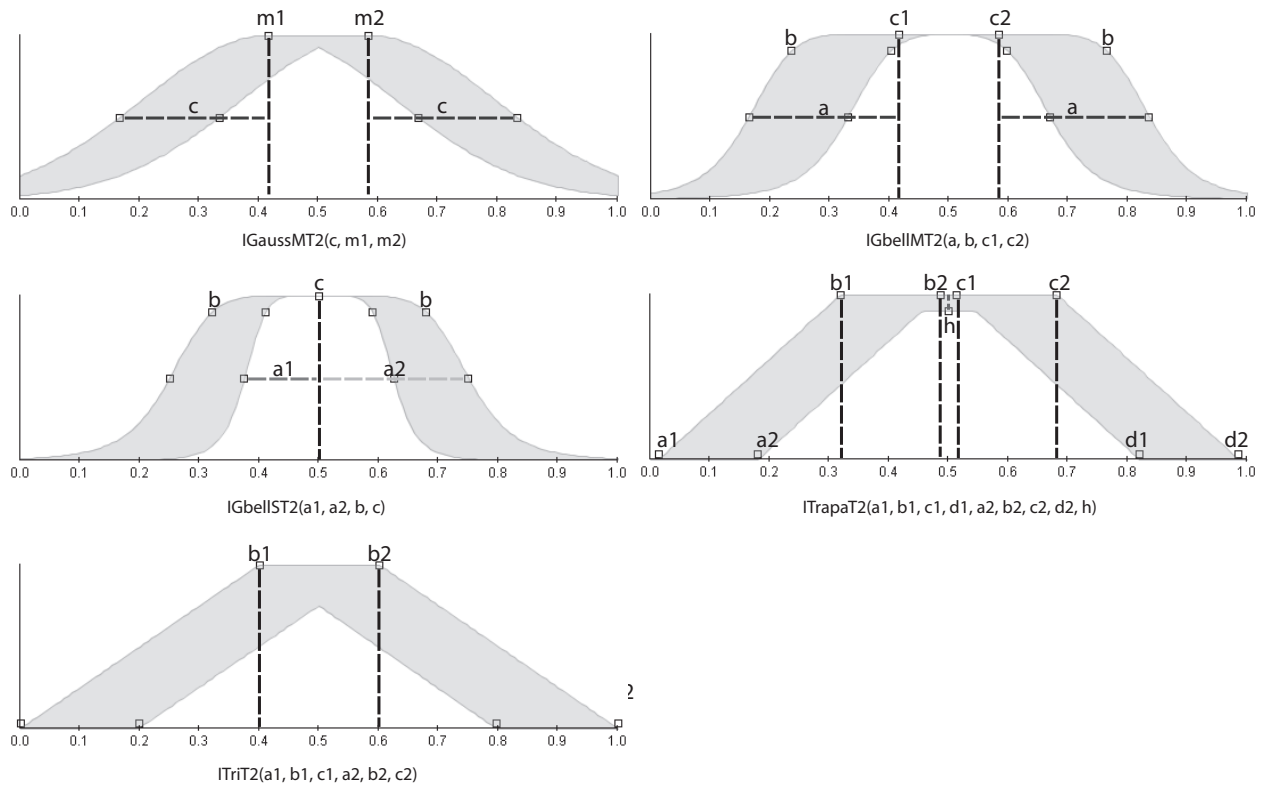
Con el fin de no limitar las herramientas creadas a las necesidades actuales, se implementó un mecanismo que permita expandir las capacidades por medio de módulos independientes de código o *plug-ins*. Estos módulos podrán ser utilizados en los guiones de los agentes y pueden contener nuevas funciones que se integran directamente con Jess u otro tipo de código que se utilice como métodos de Java.

## 3.2. Extensión de lógica difusa tipo 2

Un agente que habite un ambiente abierto o impredecible seguramente tendrá que trabajar con información ambigua o incompleta. Esto se añade a la problemática que emerge al manejar conceptos que pueden variar dependiendo de opiniones individuales y contexto como *mucho*, *poco*, *alto*, *bajo*, *frío* o *caliente*. Al aplicar la lógica difusa se pueden manejar estos conceptos y la incertidumbre se puede modelar, así dando flexibilidad a la toma de decisiones. Al modelar la incertidumbre, se puede ampliar la definición o modelar con mayor precisión el conjunto de lo que pertenece y lo que no pertenece en un concepto.

Cada concepto se modela como un conjunto difuso que contendrá el valor de pertenencia para cada elemento del universo de interés. La figura 3.4 muestra las distintas funciones de membresía disponibles para la extensión junto con los parámetros de cada una. Una vez definidas las entradas y salidas del sistema se definen las reglas las cuales son procesadas utilizando un razonamiento tipo Mamdani. Se utiliza este método por ser el más general en el manejo de las variables lingüísticas y los métodos variantes (Sugeno o Tsukamoto) pueden ser considerados casos especiales de Mamdani bajo ciertas circunstancias.

Las reglas únicamente se pueden definir del tipo *AND* ya que aunque inconveniente, con este tipo se puede expresar cualquier regla del tipo *OR*. No se cuenta con modificadores



**Figura 3.4.** Conjuntos difusos disponibles con sus parámetros

para los conjuntos (*muy*, *no tan*, entre otros) y las entradas son conjuntos difusos tipo 2 a diferencia de sistemas tradicionales que utilizan valores singleton, es decir, valores numéricos no difusos (tipo 0).

La extensión cuenta con algunas funciones que facilitan la creación de un sistema difuso, el cual puede ser creado en por lo menos tres distintas formas. La primera da la mayor flexibilidad y dificultad al requerir que se especifique cada operación de conjunto de forma manual. Las operaciones disponibles en la herramienta son: T-norm tipo 2 por intervalos (*itnormt2*), producto cartesiano por intervalos (*iProdCar*), composición Max-Star (*iMaxStar*) y la unión de conjuntos por intervalos (*IUnionT2*). La segunda alternativa es generar un objeto tipo *FIS* y a este agregar las entradas, salidas y reglas. La tabla 3.1 muestra los métodos con que cuenta el objeto *FIS*.

Método	Descripción
GenFis	Genera un nuevo objeto <i>FIS</i>
addOut	Agrega una nueva salida
addIn	Agrega una nueva entrada
addToIn	Agrega un nuevo conjunto difuso a una entrada
addToOut	Agrega un nuevo conjunto difuso a una salida
addRule	Agrega una nueva regla
inIs	Define el valor a evaluar para una entrada
evalFis	Evalúa las reglas difusas con los parámetros proporcionados
getMf	Arroja la función de membrecía contenida en un objeto <i>FISMF</i>
getLabel	Arroja la etiqueta contenida en un objeto <i>FISMF</i>
getInMf	Arroja la función de membrecía contenida en una entrada
getOutMf	Arroja la función de membrecía contenida en una salida
clearRules	Elimina todas las reglas definidas en un objeto <i>FIS</i>

**Tabla 3.1.** Métodos disponibles en el objeto *FIS*

La tercera opción es utilizar la instrucción *LoadFis* para cargar un archivo xml que contiene la estructura de un FIS (entradas, salidas, funciones de membrecía y reglas).

Una vez evaluado el sistema de inferencia, se cuenta con tres métodos para la reducción de tipo, esto es, pasar de un conjunto difuso tipo 2 a un tipo 1 y de éste a un tipo 0. A esto es lo que se le conoce como defusificación y comúnmente se obtiene un valor numérico aunque también es posible obtener el valor lingüístico. Los métodos disponibles son: centroide, método Karnik–Mendel y un método aproximado. Estos tres métodos arrojan resultados muy similares, la diferencia está en la cantidad de operaciones que se requiere para cada uno y en la precisión que se desea obtener.

Con los métodos existentes, el uso de la lógica difusa tipo 2 viene con un costo considerable en rendimiento. Al ser un concepto relativamente nuevo, se espera que con el tiempo se logren desarrollar algoritmos con mayor eficiencia y equipo computacional con mayor capacidad de procesamiento. Tomando en cuenta las limitaciones, este trabajo se enfoca en los beneficios potenciales del modelado de incertidumbre y no en un uso práctico inmediato.

Tipo	Descripción
MF	Contiene los valores correspondientes a un conjunto difuso. Se divide en tres listas que contienen los valores del universo y los valores de pertenencia superior e inferior
FIS	Contiene toda la información de la estructura del sistema de inferencia difuso
FISMF	Contiene un objeto MF con una etiqueta
IProdCar	Resultado de un producto cartesiano de conjuntos difusos tipo 2 por intervalo
IDefuzzT2	Contiene los datos de los puntos izquierdo, centro y derecho de una reducción de tipo o defucificación

**Tabla 3.2.** Tipo de datos utilizados por la extensión de NetLogo

### 3.2.1. Extensión para NetLogo

La primera versión de la extensión de lógica difusa tipo 2 fue desarrollada para la plataforma NetLogo [38]. Esta fue seleccionada por ser una herramienta conocida, fácil de utilizar, gratuita y con facilidades de desarrollo de extensiones. Para poder utilizar esta extensión se necesita seguir la documentación estándar para cualquier otra extensión en NetLogo y en el código se utiliza la instrucción `extensions[fuzzy]`. La sintaxis de las instrucciones disponibles sigue el patrón `fuzzy:<ins>` donde `<ins>` es la instrucción que se desea usar y sus parámetros. La tabla 3.2 muestra los tipos de datos relacionados con la extensión y la tabla 3.3 muestra la sintaxis de todas las instrucciones disponibles.

Instrucción	Sintaxis
fuzzy:linspace	Genera una lista de tamaño N con valores entre X1 y X2
	<code>fuzzy:linspace X1 X2 N</code>
	X1 valor numérico de inicio
	X2 valor numérico del final
	N tamaño de la lista

Funciones de Membrecía

fuzzy:iGaussmt2	<p>Conjunto difuso del tipo gaussiano</p> <p><code>fuzzy:iGaussmt2 X C M1 M2</code></p> <p>X lista que contiene el universo de interés, obtenida por medio de la instrucción <i>linspace</i></p> <p>consultar la figura 3.4 para descripción de los parámetros</p>
fuzzy:iGbellmt2	<p>Conjunto difuso del tipo campana generalizada con centro variable</p> <p><code>fuzzy:iGbellmt2 X A B C1 C2</code></p>
fuzzy:iGbellst2	<p>Conjunto difuso del tipo campana generalizada con soporte variable</p> <p><code>fuzzy:iGbellst2 X A1 A2 B C</code></p>
fuzzy:iTrapaT2	<p>Conjunto difuso del tipo trapezoidal</p> <p><code>fuzzy:iTrapaT2 X A1 B1 C1 D1 A2 B2 C2 D2 H</code></p>
fuzzy:iTriT2	<p>Conjunto difuso del tipo triangular</p> <p><code>fuzzy:iTriT2 X A1 B1 C1 A2 C2 B2</code></p>
Métodos del objeto FIS	
fuzzy:LoadFis	<p>Genera un nuevo objeto FIS a partir de un archivo xml</p> <p><code>fuzzy:LoadFis &lt;dir&gt;</code></p> <p>&lt;dir&gt; la dirección del archivo xml</p>
fuzzy:GenFis	<p>Genera un nuevo objeto FIS</p> <p><code>fuzzy:GenFis</code></p>
fuzzy:addOut	<p>Agrega una nueva salida al objeto FIS</p> <p><code>fuzzy:addOut &lt;FIS&gt; &lt;OL&gt;</code></p> <p>&lt;FIS&gt; objeto FIS al que se le agregará la salida</p> <p>&lt;OL&gt; etiqueta que se le asigna a la salida</p>
fuzzy:addIn	<p>Agrega una nueva entrada al objeto FIS</p> <p><code>fuzzy:addIn &lt;FIS&gt; &lt;IL&gt;</code></p>

	<p>&lt;FIS&gt; objeto FIS al que se le agregará la entrada</p> <p>&lt;IL&gt; etiqueta que se le asigna a la entrada</p>
fuzzy:addToOut	<p>Agrega un conjunto difuso a una salida</p> <p>fuzzy:addToOut &lt;FIS&gt; &lt;OL&gt; &lt;MFL&gt; &lt;MF&gt;</p> <p>&lt;OL&gt; etiqueta de la salida</p> <p>&lt;MFL&gt; etiqueta que se le asigna al conjunto difuso</p> <p>&lt;MF&gt; conjunto difuso de tipo MF que se agrega</p>
fuzzy:addToIn	<p>Agrega un conjunto difuso a una entrada</p> <p>fuzzy:addToIN &lt;FIS&gt; &lt;IL&gt; &lt;MFL&gt; &lt;MF&gt;</p> <p>&lt;IL&gt; etiqueta de la entrada</p> <p>&lt;MFL&gt; etiqueta que se le asigna al conjunto difuso</p> <p>&lt;MF&gt; conjunto difuso de tipo MF que se agrega</p>
fuzzy:addRule	<p>Agrega una regla difusa a un objeto FIS</p> <p>fuzzy:addRule &lt;FIS&gt; [[&lt;IL&gt;&lt;MFL&gt;] ... [&lt;IL&gt;&lt;MFL&gt;]]</p> <p>[[&lt;OL&gt;&lt;MFL&gt;] ... [&lt;OL&gt;&lt;MFL&gt;]]</p>
fuzzy:inIs	<p>Define el conjunto difuso de entrada a evaluar</p> <p>fuzzy:inIs &lt;FIS&gt; &lt;L&gt; &lt;MF&gt;</p> <p>&lt;L&gt; etiqueta de la entrada a la que se le asigna el valor</p> <p>&lt;MF&gt; conjunto difuso a evaluar</p>
fuzzy:evalFis	<p>Evalua las reglas definidas con los valores de entrada y arroja un objeto tipo FISMF</p> <p>fuzzy:evalfis &lt;FIS&gt;</p>
fuzzy:getMf	<p>Arroja el objeto MF contenido en un objeto FISMF</p> <p>fuzzy:getMf &lt;FISMF&gt;</p>
fuzzy:getLabel	<p>Arroja la etiqueta contenida en un objeto FISMF</p>

	<code>fuzzy:getLabel &lt;FISMF&gt;</code>
<code>fuzzy:getInMf</code>	Arroja el objeto MF de una entrada <code>fuzzy:getInMf &lt;FIS&gt; &lt;IL&gt; &lt;MFL&gt;</code>
<code>fuzzy:getOutMf</code>	Arroja el objeto MF de una salida <code>fuzzy:getOutMf &lt;FIS&gt; &lt;OL&gt; &lt;MFL&gt;</code>
<code>fuzzy:clearRules</code>	Elimina todas las reglas definidas en un objeto FIS <code>fuzzy:clearRules &lt;FIS&gt;</code>
Propiedades de los Conjuntos Difusos	
<code>fuzzy:getMfUpper</code>	Arroja la lista de los valores de pertenencia superiores de un conjunto difuso <code>fuzzy:getMfUpper &lt;MF&gt;</code>
<code>fuzzy:getMfLower</code>	Arroja la lista de los valores de pertenencia inferiores de un conjunto difuso <code>fuzzy:getMfLower &lt;MF&gt;</code>
<code>fuzzy:getMfX</code>	Arroja la lista de los valores del universo <code>fuzzy:getMfX &lt;MF&gt;</code>
<code>fuzzy:LeftPoint</code>	Arroja el punto izquierdo de un conjunto difuso defusificado <code>fuzzy:LeftPoint &lt;IDF&gt;</code> <IDF> objeto del tipo IDefuzzT2
<code>fuzzy:RightPoint</code>	Arroja el punto derecho de un conjunto difuso defusificado <code>fuzzy:RightPoint &lt;IDF&gt;</code>
<code>fuzzy:CenterPoint</code>	Arroja el centro de un conjunto difuso defusificado <code>fuzzy:CenterPoint &lt;IDF&gt;</code>
Operaciones de los Conjuntos Difusos	
<code>fuzzy:itnormt2</code>	Arroja el resultado de la operación t-norm entre dos conjuntos difusos, $\tilde{A} * \tilde{B}$

	<p><code>fuzzy:itnormt2 &lt;MF&gt; &lt;MF&gt; OP</code></p> <p>OP determina qué método se debe utilizar para realizar la operación, puede ser el mínimo (“min”) o el producto (“prod”)</p>
<code>fuzzy:iProdCar</code>	<p>Arroja el resultado del producto cartesiano entre dos conjuntos difusos, <math>\tilde{A} \times \tilde{B}</math></p> <p><code>fuzzy:iProdCar &lt;MF&gt; &lt;MF&gt; OP</code></p> <p>OP determina qué método se debe utilizar para realizar la operación, puede ser el mínimo (“min”) o el producto (“prod”)</p>
<code>fuzzy:iMaxStar</code>	<p>Arroja el resultado de una operación de composición entre un conjunto difuso y un producto cartesiano, <math>\tilde{A} \circ (\tilde{A}_1 \rightarrow \tilde{C}_1)</math></p> <p><code>fuzzy:iMaxStar &lt;MF&gt; &lt;IPC&gt; OP</code></p> <p>&lt;IPC&gt; resultado de un producto cartesiano</p> <p>OP determina qué método se debe utilizar para realizar la operación, puede ser el mínimo (“min”) o el producto (“prod”)</p>
<code>fuzzy:IUnionT2</code>	<p>Arroja el resultado de una operación de unión entre dos conjuntos difusos, <math>\tilde{A} \cup \tilde{B}</math></p> <p><code>fuzzy:IUnionT2 &lt;MF&gt; &lt;MF&gt;</code></p>
<code>fuzzy:idefuzzt2</code>	<p>Realiza la reducción de tipo sobre un conjunto difuso</p> <p><code>fuzzy:idefuzzt2 &lt;MF&gt; OP</code></p> <p>OP determina el método que será utilizado en la reducción de tipo. Los métodos disponibles son: centroide (“centroid”), método Karnik–Mendel (“km”) y un método aproximado (“aprox”)</p>

**Tabla 3.3.** Sintaxis de métodos disponibles de la extensión para NetLogo

#### 3.2.2. Extensión para Wiinik

La extensión difusa de Wiinik está diseñada para su uso con Jess y la combinación de lógica difusa y Jess no es nueva. Extensiones como FuzzyJess [26] proporcionan las capacidades de un sistema difuso tipo 1 tradicional, permitiendo que las reglas de producción se definan como difusas. Incluso proporciona modificadores que se pueden aplicar a los conjuntos difusos y dan una mejor descripción de las variables lingüísticas. La extensión para Wiinik es un sistema de inferencia difuso tipo 2 lo cual permite modelar la incertidumbre de los conjuntos difusos dando mayor flexibilidad a los agentes que se controlan para tolerar pequeñas variantes en la toma de decisiones.

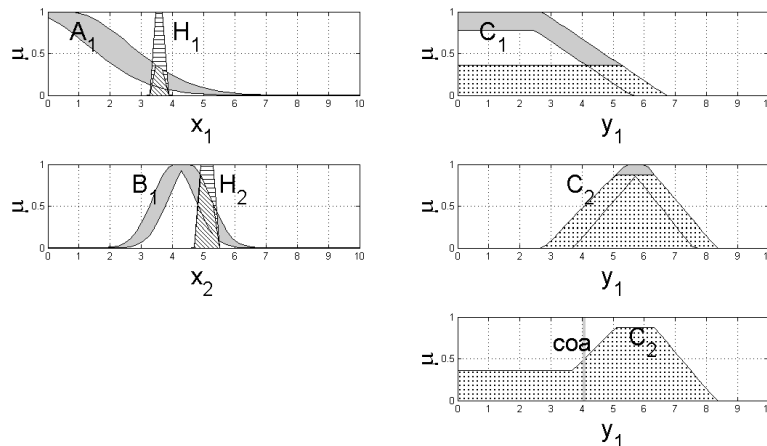
La forma de uso de la extensión consiste en importar un archivo JAR (por su nombre en inglés, Java ARchive) al código de Jess y se tendrá acceso a los objetos necesarios para definir y ejecutar un FIS tipo 2. Se cuenta también con la definición de algunas funciones de usuario que facilitan la creación de conjuntos difusos. Lo que esto significa es que, a diferencia de FuzzyJess, el uso de esta extensión no requiere de una reingeniería de las reglas para adecuarlas a una regla difusa y no requiere modificaciones al código estándar del interpretador de Jess.

Al ser una versión basada en la extensión creada para NetLogo, ambas contienen en gran parte la misma funcionalidad. Las diferencias son mínimas y mayormente de sintaxis. Un cambio notable en funcionalidad es la posibilidad de obtener el valor lingüístico del consecuente con mayor valor. Esta forma de defusificación es similar a lo que se obtiene de un FIS tipo Sugeno de orden cero y los consecuentes constantes son valores lingüísticos. La figura 3.5 muestra dos reglas difusas evaluadas con el método de Mamdani. Las reglas difusas son:

**SI**  $A$  es  $A_1$  **ENTONCES**  $C$  es  $C_1$

**SI**  $B$  es  $B_1$  **ENTONCES**  $C$  es  $C_2$

Si se elige defusificar por el método del centroide la respuesta será el que se denomina *coa* (por el inglés, *Center Of Area*). En cambio, si se desea obtener la variable lingüística con mayor valor de pertenencia entonces la respuesta es  $C_2$ .



**Figura 3.5.** Razonamiento difuso tipo 2 por intervalos de Mamdani

A continuación se presentan en forma resumida el proceso de crear un sistema de inferencia difuso en Jess con muestras de sintaxis.

1. Definir el universo de interés

```
(bind ?ls1 (linspace 0 1 51))
```

2. Crear un objeto tipo *FIS*

```
(bind ?fis (new Fis))
```

3. Definir las entradas y sus variables lingüísticas

```
(call ?fis addInput "A")
```

```
(call ?fis addToInput "A" "A_1" (IGaussMT2 ?ls1 0.15 -0.04 0))
```

4. Definir las salidas y sus variables lingüísticas

```
(call ?fis addOutput "C")
```

```
(call ?fis addToOutput "C" "C_1" (ITriT2 ?ls2 0 3 5 2 4 8))
```

5. Definir las reglas (SI  $A$  es  $A_1$  ENTONCES  $C$  es  $C_1$ )

```
(call ?fis addRule "[A:A_1]" "[C:C1]")
```

### 3.2.3. Editor gráfico para el diseño de sistemas de inferencia difusos tipo 2

Mientras la habilidad de incorporar la lógica difusa a un agente es deseable, no siempre es sencillo. El diseñar e implementar sistemas de inferencia difusos debe ser hecho lo más simple posible para mejor aprovechar el tiempo en el perfeccionamiento de los parámetros y no en detalles técnicos. En respuesta a esta necesidad se desarrolló un editor gráfico para diseñar sistemas de inferencia difusos tipo 2 por intervalos para el uso con las extensiones de Wiinik y NetLogo. El editor llamado “IT2Fuzzy” (*Interval Type 2 Fuzzy*) se asemeja a herramientas más reconocidas como las encontradas en Matlab, así facilitando el uso de esta.

Para los expertos y no expertos por igual, es preferible utilizar herramientas gráficas para el diseño. Dada la incertidumbre que manejan los FIS, requiriendo a expertos determinar la cantidad y tipo de valores en los conjuntos difusos, el poder ajustar el modelo de forma rápida y sencilla reduce el tiempo de desarrollo. La figura 3.6 muestra el ambiente de diseño principal, el cual permite ensamblar un sistema difuso de una forma sencilla de entender.

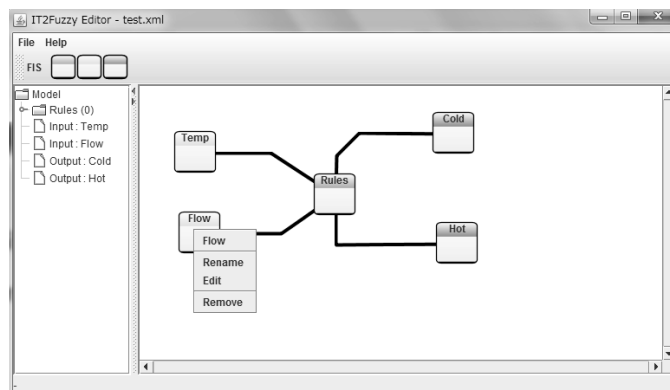
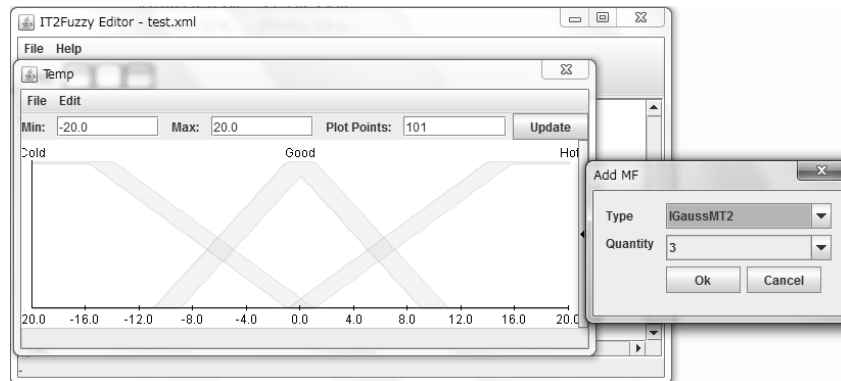


Figura 3.6. Área de diseño principal del editor IT2Fuzzy

### 3. Modelo de Mensaje y Desarrollo de Herramientas para su Interpretación 42

La figura 3.7 muestra el editor de los conjuntos difusos. Cada entrada y salida definida puede contar con varios conjuntos representando las variables lingüísticas. En el diseño de sistemas difusos se recomienda que se cuente con mínimo tres variables lingüísticas abarcando el universo de interés.



**Figura 3.7.** Área de edición para conjuntos difusos

El elemento que une a las entradas y salidas es el de las reglas. En este se define la relación que tendrán todas las variables y por lo tanto el comportamiento del sistema difuso. La figura 3.8 muestra al editor de reglas difusas el cual cuenta con una funcionalidad muy básica de agregar, editar y borrar.

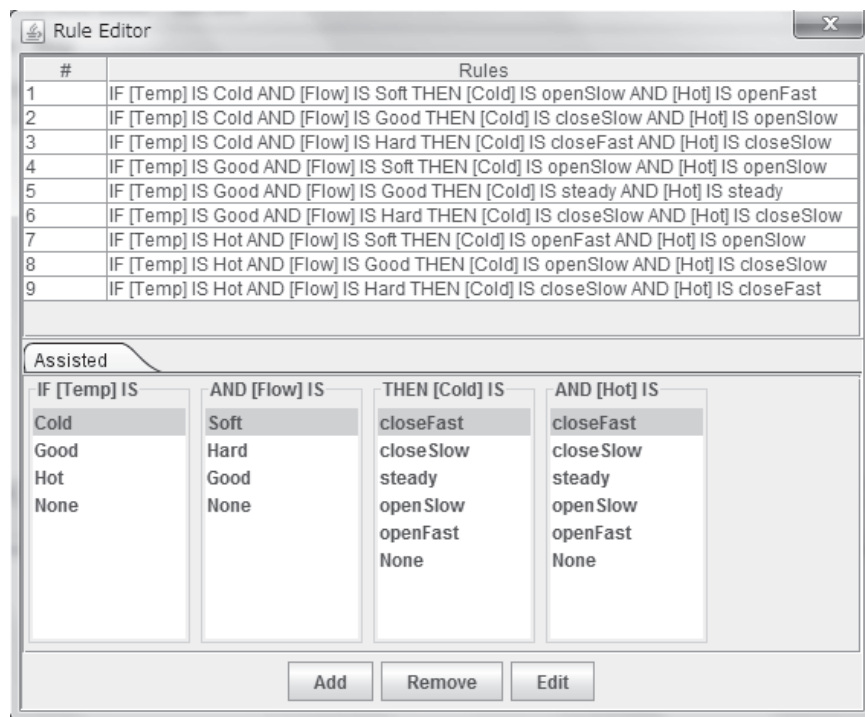


Figura 3.8. Área de edición para las reglas difusas

# Capítulo 4

## Casos de Estudio

### 4.1. Caso Don Juan

Este caso de estudio se basa en la patología descrita en [24] y en trabajos relacionados [9]. Los agentes son modelados de tal forma que sigan el comportamiento típico descrito en el síndrome de Don Juan, únicamente cuando sus perfiles concuerden y se encuentren en el estado del ego correcto. Además, los guiones de comportamiento implementan el concepto de catexis [30] que permite cambiar de actividad conforme a las necesidades de una manera estable.

El primer mensaje intercambiado entre agentes es un saludo, esto permite a cada agente intercambiar información sobre la percepción que tiene uno sobre el otro. Basándose en esta información los agentes decidirán cómo actuar y decidir cómo seguirá la conversación. Si un agente no cumple con el perfil que se busca entonces ningún mensaje le seguirá, de lo contrario, el guión continuará su curso. La información utilizada para decidir este comportamiento se encuentra contenida en el objeto BPNM de la estructura del mensaje definido en la ecuación 3.2.

	<i>Don Juan</i>	<i>Víctima</i>
$E_{s_i}$	P (Adulto)	C (Niño)
$E_{s_u}$	C (Niño)	P (Adulto)
$\pi_i$	1 (Ok)	0 (NotOk)
$\pi_u$	0 (NotOk)	1 (Ok)

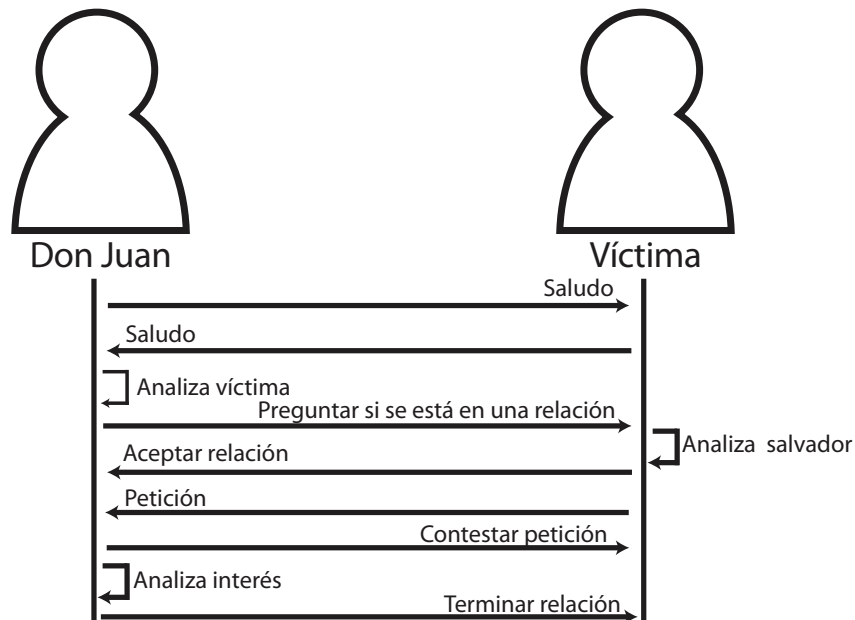
**Tabla 4.1.** Valores que contiene el objeto BPNM para los perfiles de “Don Juan” y “víctima”

Si el agente actuando tiene un perfil de “Don Juan” y decide que un agente con quien acaba de intercambiar un mensaje de saludo cumple con el perfil de víctima y no tiene una relación con otro agente, se le propone una relación. La tabla 4.1 muestra los valores del perfil de comportamiento que definen los roles de “Don Juan” y “víctima”. El segundo agente recibe este mensaje y dependiendo si el agente emisor es percibido de tener el perfil de salvador y si se desea estar en una relación, se decidirá si se acepta la propuesta o no. En esta decisión pueden influir distintos factores dependiendo de la complejidad que se desea tener, para este caso se aisló el caso y se cuentan con una reducida cantidad de factores influyentes.

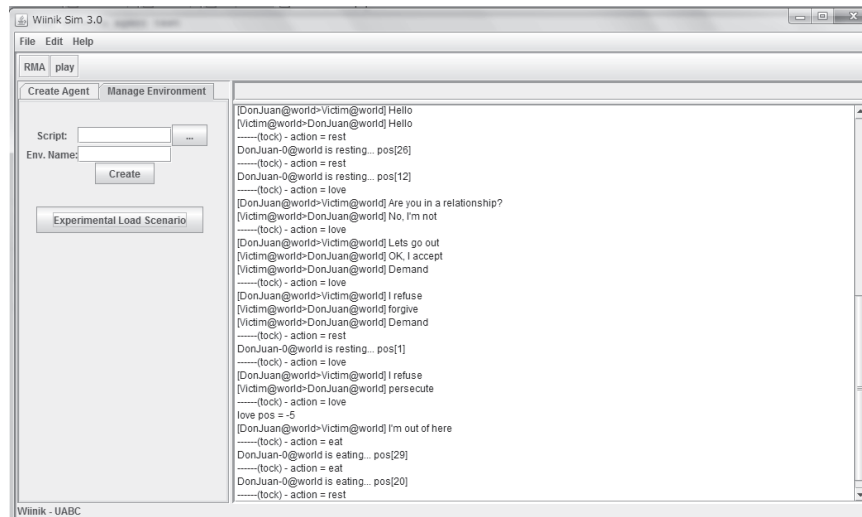
Cuando los agentes acuerdan a estar en una relación, una serie de mensajes son intercambiados en el cual el agente con rol de víctima empezará a demandar cosas del agente salvador (Don Juan). Este incremento de responsabilidad y compromiso reducirá el interés del agente Don Juan hacia la víctima y pronto terminará con la relación. La víctima quedará descontenta con el agente Don Juan y este buscará otra víctima.

Mientras los agentes interactúan, ciertos valores internos han cambiado, principalmente los que indican la percepción que se tiene sobre el otro agente y sobre si mismo lo cual permite que el guión continúe. Una variedad de situaciones pueden alterar el curso y puede resultar en que no se siga el patrón, estas alteraciones pueden venir del exterior (ambiente, otros agentes) o del interior (necesidades del agente). La figura 4.1 muestra un posible escenario en el intercambio de mensajes siguiendo el guión de Don Juan y una víctima. En las figuras 4.2 y 4.3 se muestra la simulación del escenario visualizado en Wiinik y con las herramientas de

Jade respectivamente.



**Figura 4.1.** Intercambio de mensajes entre un agente “Don Juan” y una “víctima”



**Figura 4.2.** Simulación de Don Juan vista con la terminal de Wiinik

El objetivo de este escenario es poder variar las actividades que un agente sigue dependiendo del estado psicológico y la percepción de los eventos que ocurren. Otro factor influyente en las acciones de los agentes es las necesidades individuales, la forma de alternar de forma

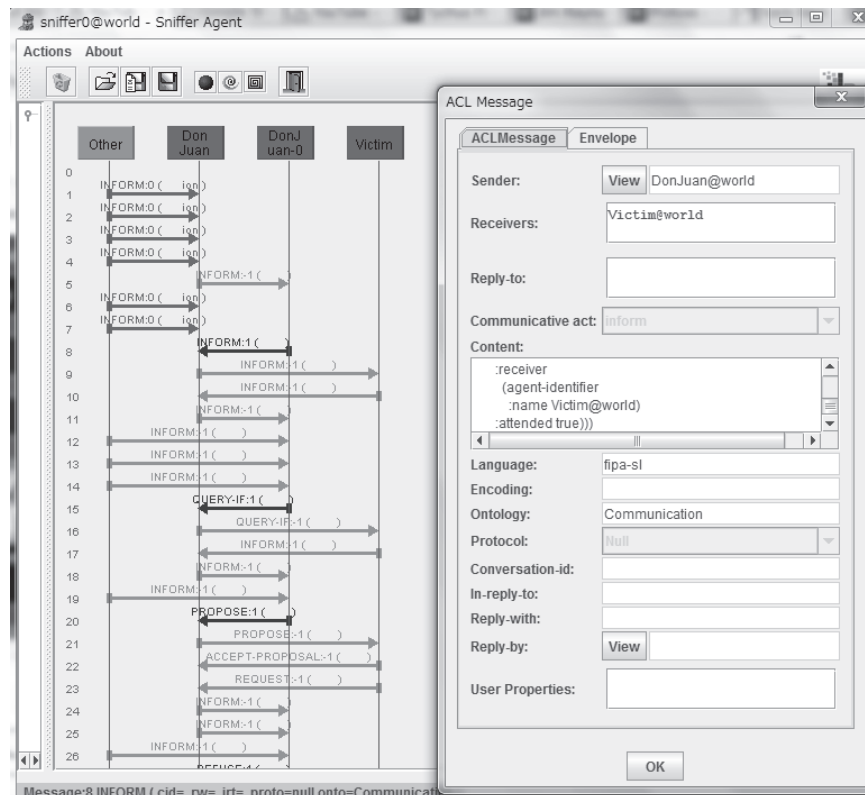


Figura 4.3. Simulación de Don Juan vista con las herramientas de Jade

estable entre acciones es implementando el concepto de catexis. Este concepto permite asignar distintas tendencias de importancia a las acciones, lo cual define una parte del perfil de los agentes.

## 4.2. Caso de Trabajadores

El caso de estudio consiste en simular una compañía que fabrica algún producto determinado y existen tres distintos tipos de agentes que son: gerentes, trabajadores y clientes que son representados por el ambiente. Cuando se trabaja en un grupo para cumplir un objetivo, es importante que cada miembro cumpla su rol y que se comparta información sobre conceptos de interés común. El éxito o fracaso será determinado por el rendimiento del grupo y

no por logros individuales. Este caso de estudio demuestra cómo información clave puede ser estandarizada entre un grupo y se le interpreta de formas distintas o da mayor importancia a distintas partes de la información.

#### 4.2.1. Descripción general de las interacciones

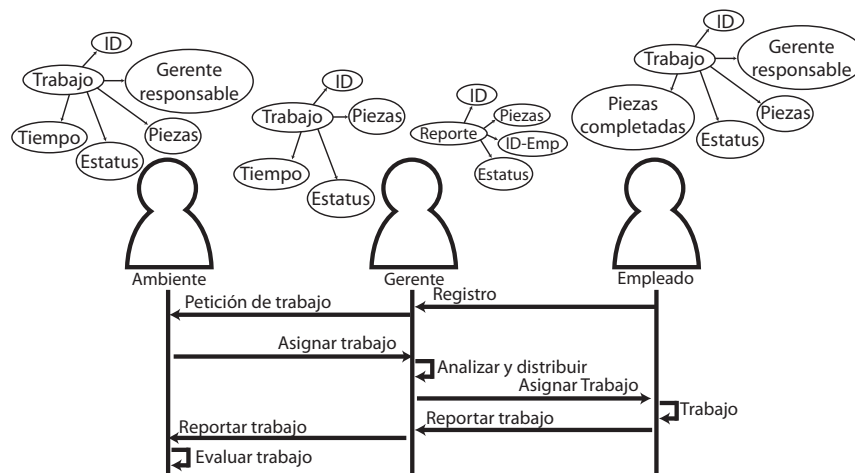
En este ejemplo, un gerente debe primero contar con un registro de cuántos empleados están bajo su cargo listos para trabajar y sus capacidades. La capacidad de los empleados se mide por el nivel de fatiga que puede tener, cada empleado debe comunicar a su gerente para informar esos datos lo antes posible. Una vez que el gerente tiene un registro de los empleados disponibles, se le puede enviar un mensaje al agente que representa al ambiente para solicitar un trabajo. El agente ambiente responderá con especificaciones para un trabajo y esperará una respuesta de aceptación o rechazo.

Dependiendo del tamaño del trabajo y de cuántos empleados hay disponibles, un gerente debe decidir si se cuenta con las capacidades para cumplir con el trabajo a tiempo. Si el trabajo se rechaza, entonces se pierde credibilidad y recibirá trabajos menores. Se debe decidir si esto es mejor que arriesgar incumplir con las especificaciones acordadas. Si un trabajo es aceptado, el gerente dividirá el trabajo entre los empleados tomando en cuenta sus capacidades, esto puede resultar en una mayor carga para los empleados en mejor condición.

Dado que un agente gerente tiene conocimiento sobre las condiciones de cada empleado, puede que tenga consideración sobre que tan cansado se encuentra un agente o puede ignorar esta información. Si se elige asignar trabajo a un agente cansado, esto causará que se cometan una mayor cantidad de errores y se puede demorar el trabajo más de lo planeado. Un agente empleado realizará su tarea asignada lo mejor que sus capacidades se lo permitan pero también cuenta con diversas actividades que pueden llegar a tener mayor prioridad. Dependiendo del perfil de un agente, la asignación de prioridad a las actividades pueden conducir a un com-

portamiento desde adicción al trabajo hasta perezoso. Cuando un agente se encuentra en un estado debilitado, entonces su probabilidad de cometer errores aumenta considerablemente y puede que decidan dejar de trabajar por un tiempo determinado mientras se recuperan.

Una vez que el agente empleado termina lo que se le asignó, mandará un mensaje a su gerente para informar que la tarea fue completada. Cuando el gerente recibe suficientes tareas terminadas para cumplir con la meta establecida, esta información se le comunica al agente ambiente y este califica el trabajo realizado basándose en el tiempo que tardó. Una vez terminada una tarea asignada, el ciclo se repite al hacerse de nuevo una petición de trabajo al ambiente por parte del gerente pero bajo condiciones distintas ya que los trabajadores han tenido desgaste y la percepción del ambiente hacia el gerente ha sido alterada.



**Figura 4.4.** Caso de estudio de trabajadores, flujo de mensajes y estructura de la información intercambiada

La figura 4.4 muestra un caso simplificado que consiste en un agente de cada tipo interactuando. Lo recomendable es siempre contar con más de un empleado ya que un solo trabajador se cansaría muy rápido y se tendría que rechazar muchos trabajos antes de que se pueda aceptar más y estos solo pueden ser aceptados si son pequeños o se da mucho tiempo para completarlos. Lo que se encuentra arriba de cada agente es la estructura del concepto de trabajo que tiene cada uno. Como se puede observar, existen conceptos compartidos lo

cual permite que se entiendan pero se le puede dar distinta interpretación o importancia.

#### 4.2.2. Incorporación de lógica difusa

Existen formas de medir que no son exactas y que pueden depender del contexto y criterio personal de los individuos que lo usan. Tal es el caso de conceptos como grande/pequeño, rápido/lento o mucho/poco. Estos conceptos pueden ser representados como conjuntos difusos, así permitiendo que puedan existir ligeras variaciones en la definición para conceptos comunes. Cuando se trata con un conjunto fijo de acciones pero con criterios variables para decidir qué acción tomar, se puede utilizar la lógica difusa como una opción viable al definir los perfiles de comportamiento. Para ejemplificar, se añaden reglas difusa a dos de los tipos de agentes, las cuales definen las prioridades que tendrán. El tercer tipo de agente se deja sin cambios en su toma de decisiones.

El agente ambiente, el cual representa a los clientes o las condiciones del mercado, tiene dos conceptos que se representan por medio de conjuntos difusos, el tamaño del trabajo y el tiempo en que se requiere que se termine. Estos conceptos se evalúan con cinco reglas difusas que se muestran en la tabla 4.2 y se pueden codificar en Jess utilizando la extensión creada como se muestra en la figura 4.5. El FIS arroja resultados numéricos en ambas salidas (cantidad y tiempo) y estos forman las especificaciones de un trabajo que se envía al agente gerente.

Cabe mencionar que a diferencia de muchos sistemas de inferencia difusos, para este caso se utiliza un conjunto difuso tipo 2 como entrada en vez de un valor numérico. Esto permite trabajar con incertidumbre desde el inicio, reflejando la percepción del agente y no una medición exacta. La granularidad de los conceptos descritos se debe mantener a un mínimo para obtener resultados útiles.

Los agentes del tipo trabajador también cuentan con un FIS el cual se encarga de definir

	Variable Lingüística	Valor Lingüístico
Entradas	Tamaño del trabajo ( <i>JSize</i> )	Pequeño ( <i>Small</i> ) Mediano ( <i>Medium</i> ) Grande ( <i>Large</i> )
	Tiempo ( <i>Time</i> )	Urgente ( <i>urgent</i> ) No urgente ( <i>notUrgent</i> )
Salidas	Tamaño del trabajo	1 2 3
	Tiempo	Rápido ( <i>Fast</i> ) Lento ( <i>Slow</i> )
Reglas		
<b>SI</b> <i>Tamaño del trabajo</i> <b>ES</b> <i>pequeño</i> <b>ENTONCES</b> <i>Tamaño del trabajo</i> <b>ES</b> <i>1</i> <b>SI</b> <i>Tamaño del trabajo</i> <b>ES</b> <i>mediano</i> <b>ENTONCES</b> <i>Tamaño del trabajo</i> <b>ES</b> <i>2</i> <b>SI</b> <i>Tamaño del trabajo</i> <b>ES</b> <i>grande</i> <b>ENTONCES</b> <i>Tamaño del trabajo</i> <b>ES</b> <i>3</i> <b>SI</b> <i>Tiempo</i> <b>ES</b> <i>urgente</i> <b>ENTONCES</b> <i>Tiempo</i> <b>ES</b> <i>Rápido</i> <b>SI</b> <i>Tiempo</i> <b>ES</b> <i>No urgente</i> <b>ENTONCES</b> <i>Tiempo</i> <b>ES</b> <i>Lento</i>		

**Tabla 4.2.** Componentes de un sistema de inferencia difuso utilizado en la generación de trabajos

la acción a tomar basándose en el nivel de salud que se tiene. De forma similar al agente ambiente, se pueden manipular los parámetros de las variables difusas con el fin de crear distintos perfiles de comportamiento. La figura 4.6 muestra la estructura del FIS utilizado en los agentes trabajadores. Su uso varía del utilizado en el agente ambiente al arrojar un resultado lingüístico en vez de numérico.

Al agente gerente no se le agrega un FIS, esto con la intención que las decisiones tomadas por este agente sean basadas en mediciones exactas. Dado que este agente decide si los trabajos se pueden completar con los trabajadores disponibles, este tipo de decisiones tiene que ver más con aritmética y probabilidad que con conceptos ambiguos.

En conclusión, en este caso de estudio existen tres tipos de agentes y cada uno cuenta con su propia interpretación de lo que es un trabajo y su proceso de decisión varía entre cálculos numéricos y razonamientos difusos. Los mensajes intercambiados pueden contener

```
(bind ?fis (new Fis))
(call ?fis addInput "JSize")
  (call ?fis addToInput "JSize" "Small" (IGaussMT2 ?l1 0.15 -0.04 0))
  (call ?fis addToInput "JSize" "Medium" (IGaussMT2 ?l1 0.15 0.48 0.52))
  (call ?fis addToInput "JSize" "Big" (IGaussMT2 ?l1 0.15 1.04 1))
(call ?fis addInput "Time")
  (call ?fis addToInput "Time" "urgent" (IGaussMT2 ?l1 0.21 -0.08 0.08))
  (call ?fis addToInput "Time" "notUrgent" (IGaussMT2 ?l1 0.21 0.92 1.08))
(call ?fis addOutput "JSize")
  (call ?fis addToOutput "JSize" "1" (ITriT2 ?l2 0 3 6 2 5 8))
  (call ?fis addToOutput "JSize" "2" (ITriT2 ?l2 11 14 17 13 16 19))
  (call ?fis addToOutput "JSize" "3" (ITriT2 ?l2 22 25 28 24 27 30))
(call ?fis addOutput "Time")
  (call ?fis addToOutput "Time" "Fast" (IGaussMT2 ?l3 7.85 36.92 43.08))
  (call ?fis addToOutput "Time" "Slow" (IGaussMT2 ?l3 7.85 -0.08 6.08))
(call ?fis addRule "[JSize:Small]" "[JSize:1]")
(call ?fis addRule "[JSize:Medium]" "[JSize:2]")
(call ?fis addRule "[JSize:Big]" "[JSize:3]")
(call ?fis addRule "[Time:urgent]" "[Time:Fast]")
(call ?fis addRule "[Time:notUrgent]" "[Time:Slow]")
```

**Figura 4.5.** Código en Jess de la estructura de un FIS de un agente ambiente

información como fragmentos de conocimiento (llamados *facts* en Jess), texto simple o incluso conjuntos difusos.

```
(bind ?fis (new Fis))
(call ?fis addInput "Fatigue")
  (call ?fis addToInput "Fatigue" "None"
    (ITrapaT2 ?l1 -0.07 -0.05 0.11 0.2 -0.03 -0.01 0.21 0.3 0.9))
  (call ?fis addToInput "Fatigue" "Little"
    (ITrapaT2 ?l1 0.31 0.44 0.54 0.61 0.41 0.47 0.57 0.71 0.9))
  (call ?fis addToInput "Fatigue" "Alot"
    (ITrapaT2 ?l1 0.71 0.86 1.19 1.19 0.80 0.95 1.26 1.26 0.9))
(call ?fis addOutput "Action")
  (call ?fis addToOutput "Action" "Work"
    (ITriT2 ?l1 0 0.08 0.16 0.05 0.11 0.22))
  (call ?fis addToOutput "Action" "WorkSlow"
    (ITriT2 ?l1 0.4 0.49 0.56 0.45 0.52 0.62))
  (call ?fis addToOutput "Action" "Rest"
    (ITriT2 ?l1 0.8 0.85 0.94 0.86 0.93 1.0))
(call ?fis addRule "[Fatigue:None]" "[Action:Work]")
(call ?fis addRule "[Fatigue:Little]" "[Action:WorkSlow]")
(call ?fis addRule "[Fatigue:Alot]" "[Action:Rest]")
```

**Figura 4.6.** Código en Jess de la estructura de un FIS de un agente trabajador

# Capítulo 5

## Conclusiones y Trabajo Futuro

### 5.1. Conclusiones

El mundo es más complejo de lo que se puede expresar en texto, no existen límites bien definidos para clasificar, el significado de acciones varía entre culturas y la visión del ambiente depende de percepciones y conocimiento previo. Aun limitando el campo de estudio, es necesario trabajar con modelos reducidos y con suposiciones que no reflejan un comportamiento real del fenómeno. En el caso de la comunicación, se debe abordar de tal forma que concuerde con las distintas reglas sociales, actitudes y mensajes no verbales que enriquecen y modifican lo que los interlocutores transmiten y reciben.

La comunicación estandarizada entre agentes fue diseñada con el propósito de garantizar interoperabilidad y reducir ambigüedad en la comunicación. Estas características no son siempre deseables, especialmente cuando se trata de simular interacciones humanas donde la ambigüedad no solamente es inevitable, sino esencial si se desea entender fenómenos complejos como bromas, sarcasmo o comentarios capciosos. Únicamente tomando un modelo reducido de transmisor y receptor, se puede contar con complicaciones en la comunicación, lo que se

dice, lo que se quiere decir y lo que se cree que se dice no siempre concuerda y es causa de conflicto.

En este trabajo se presentó un modelo de mensaje que permite agregar información de contexto. Junto con una arquitectura de agente flexible, un modelo de estructurar conocimiento y mecanismos para su interpretación, se forma un conjunto de herramientas que cumplen con los objetivos planteados por esta investigación. Con respecto al modelo y lo desarrollado, se puede resaltar lo siguiente:

- La habilidad de implementar distintos escenarios, modelos psicológicos y modelos de conocimiento sin cambiar la arquitectura básica de los agentes disponibles.
- El comportamiento de los agentes se controla por medio de guiones, lo cual permite que los agentes no sean recompilados al cambiar instrucciones y las reglas pueden variar durante la ejecución de un escenario. Cada agente cuenta con su propia máquina de inferencia para procesar sus reglas y almacenar el conocimiento adquirido, así formando agentes únicos ya sea de guiones de inicio idénticos o distintos.
- Los mensajes intercambiados pueden contener información sobre el contexto en que se utilizan y la percepción que se tiene sobre el evento y los interlocutores. La información adicional puede llegar a alterar de forma significativa un mensaje, ya sea que se identifique como broma, sarcasmo o mentira.
- Al intercambiar mensajes que contengan valores lingüísticos, estos se pueden traducir en acciones o valores por medio de un sistema de inferencia difuso capaz de interpretar los valores recibidos. Los valores procesados pueden venir de distintas fuentes, ya sea en el contenido de un mensaje, la percepción de ese mensaje o de las condiciones percibidas del ambiente. Toda la información se puede tomar en cuenta para producir nuevos conocimientos o motivar a realizar ciertas actividades.

- Al considerar que no todo usuario potencial es un experto en los conceptos que se manejan, en específico de la lógica difusa, se facilita el diseño de un FIS al utilizar herramientas gráficas. En el caso del diseño de un FIS tipo 2, existe la alternativa de utilizar Matlab o el editor IT2Fuzzy, creado para su uso específico con las extensiones de NetLogo y Wiinik.
- A pesar de el potencial del modelo y las herramientas que lo implementan, cabe la necesidad de agregar aun más. Por esta razón se deja abierta la posibilidad de aumentar las capacidades utilizando extensiones. Este es el caso de la incorporación de lógica difusa tipo 2, la cual se implementa como una extensión en las herramientas de Wiinik y NetLogo.

## 5.2. Trabajo futuro

Siguiendo con la investigación y dado que se conforma de distintas áreas, se puede seguir desarrollando en varios aspectos. Con respecto al modelo de mensajes, de agentes y la herramienta Wiinik, se puede mencionar lo siguiente:

- Agregar la capacidad de controlar con guiones todas las actividades de los agentes, incluyendo a los agentes tipo *persona* y *timer*.
- Implementar mecanismos que permitan ver los valores internos de los agentes en tiempo de ejecución. Las herramientas proporcionadas por Jade no son suficiente para visualizar las reglas internas y conocimiento contenido en Jess.
- Proporcionar alternativas para la visualización de las interacciones entre agentes. Una conversación en texto es suficiente para un mundo donde únicamente existen los interlocutores, pero al agregar capacidades de interactuar con ambientes complejos se

requiere visualizar lo que ocurre en el mundo de una forma detallada.

- Mejorar el control que se tiene sobre una simulación, incluyendo los parámetros que se manejen y poder pausar, continuar, guardar y cargar datos.
- Facilitar el descubrimiento de nuevo conocimiento al implementar técnicas de minería de datos. En este aspecto se encuentra la posibilidad de estudiar lo que sería la creatividad, imaginación y astucia. Que un agente pueda observar su ambiente e idear soluciones a problemas, soñar con un mejor futuro o crear fantasías increíbles y nuevas formas de transmitir. Tal es el caso del arte y la música que son formas de transmitir ideas o emociones.
- Incorporar más técnicas del área de inteligencia artificial. De interés en particular se encuentran las distintas arquitecturas de redes neuronales y los algoritmos genéticos. Estas técnicas mejorarán la autonomía de los agentes y al experimentar, se puede adaptar las arquitecturas al campo de lo social y ambiguo ya que muchas de estas técnicas son diseñadas con aplicaciones de control como su objetivo.
- La arquitectura de los agentes que se implementó no es más que una aproximación al modelo teórico. Se debe mejorar a un modelo basado en agentes nodos que puedan ser independientes, semi-independientes y dependientes. Las relaciones de dependencia serían dinámicas y las redes de agentes formarían comunidades, corporaciones, gobiernos y cualquier otro tipo de unión.
- En la transmisión de los mensajes con los parámetros del objeto BPNM, la percepción de uno mismo no debe ser transmitida, si no dada otra interpretación en la forma de transmitir. Todo agente solo podrá conocer al mundo por su percepción, no podrá conocer lo que el mundo piensa de si mismo.

En el caso de la implementación de la lógica difusa en los agentes, se puede trabajar en las siguientes áreas:

- Agregar modificadores lingüísticos
- Implementar nuevas técnicas que permitan mejorar la defusificación lingüística. La habilidad de arrojar más de un valor por salida e interpretar las funciones agregadas de tal forma que se pueda crear nuevos conceptos.
- Agregar la opción de visualizar una gráfica del comportamiento del FIS en el editor.
- Implementar distintos tipos de razonamiento difuso como el de Sugeno.

# Bibliografía

- [1] R. J. Brachman. What is-a is and isn't: An analysis of taxonomic links in semantic networks. *Computer*, 16:30–36, October 1983.
- [2] G. Caire, F.L. Bellifemine, and D. Greenwood. *Developing Multi-Agent Systems with JADE*. Wiley, 2007.
- [3] M. Castañón Puga. *Epistemología de la Simulación Social: El problema de modelar a la persona*. PhD thesis, Universidad Autónoma de Baja California, 2008.
- [4] J. Denzinger and J. Hamdan. Improving observation-based modeling of other agents using tentative stereotyping and compactification through kd-tree structuring. *Web Intelligence and Agent Systems*, 4(3):255–270, 2006.
- [5] Umberto Eco. *A Theory of Semiotics*. Indiana University Press, 1979.
- [6] Rolf Theil Endresen, Hanne Gram Simonsen, and Andreas Sveen. *Innføring i lingvistikk*. Universitetsforl, Oslo, Norway, 2nd edition, 2000.
- [7] T. Finin, R. Fritzson, D. McKay, and McEntire. Kqml as an agent communication language. In *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, pages 456–463. ACM Press, 1994.
- [8] FIPA. Fipa abstract architecture specification. Technical report, FIPA, 2002.

- [9] Dora-Luz Flores, Antonio Rodríguez Díaz, Juan R. Castro, and C. Carelia-Gaxiola. Ta-fuzzy semantic networks for interaction representation in social simulation. In *Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control*, volume 257, pages 213–225. SpringerLink, 2009.
- [10] R. V. Florian. Autonomous artificial intelligent agents. Technical Report Coneural-03-01, Center for cognitive and neural studies, Romania, 2003.
- [11] Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Institute for Intelligent Systems, University of Memphis, Springer-Verlag, 1996.
- [12] E. Friedman-Hill. Jess, the java expert system shell. Technical Report SAND98-8206, Sandia National Laboratories, 1997.
- [13] Dora Luz Flores G. *Modelado de Contexto usando Redes Semánticas para la Detección de Conflicto Social*. PhD thesis, Universidad Autónoma de Baja California, 2009.
- [14] N. Ghasem-Aghae, M. Kaedi, and T.I. Ören. Effects of cognitive complexity in agent simulation: Fuzzy rules and an implementation. In *International Mediterranean Modeling Multiconference*, 2005.
- [15] D. Gilbert, M. Aparicio, B. Atkinson, S. Brady, J. Ciccarino, B. Grosz, P. O'Connor, D. Osisek, S. Pritko, R. Spagna, and L. Wilson. IBM Intelligent Agent Strategy. *IBM Corporation*, 1995.
- [16] Nigel Gilbert. *When Does Social Simulation Need Cognitive Models?* Cambridge University Press, 2006.

- [17] Nigel Gilbert. *Agent-Based Models Series: Quantitative Applications in the Social Sciences*. Sage, 2008.
- [18] J.-S. Roger Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. MATLAB Curriculum Series. Prentice Hall, Upper Saddle River, NJ, 1997.
- [19] John McCarthy. Ascribing mental qualities to machines. In *Formalizing Common Sense*. Ablex, 1990. Originally published in 1979.
- [20] David McNeil. *Hand and Mind: What Gestures Reveal about Thought*. The University of Chicago Press, 1992.
- [21] David McNeill. *Gesture and Thought*. University of Chicago Press, 2005.
- [22] M. Minsky. A framework for representing knowledge. In P. Winston, editor, *The Psychology of Computer Vision*, pages 211–277. Mcgraw-Hill, New York, 1974.
- [23] R.Ñeches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W.R. Swartout. Enabling technology for knowledge sharing. *AIMag*, 12(3):36–56, 1991.
- [24] M.Ñovellino. The don juan syndrome: The script of the great losing lover. *Transactional Analysis Journal*, 36(1):33–43, 2006.
- [25] Hyacinth S. Nwana. Software agents: An overview. *Knowledge Engineering Review*, 11(3):205–244, 1996.
- [26] R. Orchard. Fuzzy reasoning in jess: The fuzzy j toolkit and fuzzy jess. In *Proceedings of the Third International Conference on Enterprise Information Systems, ICEIS*, pages 533–542, 2001.
- [27] R. Pfeifer and C. Scheier. *Understanding Intelligence*. The MIT Press, Cambridge, Massachusetts, 2001.

- [28] M.R. Quillian. Semantic memory. In M. Minsky, editor, *Semantic Information Processing*, pages 227–270. MIT Press, 1968.
- [29] Marco Remondino. Multi-agent based modelling: From social simulation to real time strategy games. In *Game-On*, 2003.
- [30] A. Rodríguez Díaz, A. Cristóbal Salas, M. Castañón Puga, Jáuregui C., and González C. Personality and behaviour modelling based on cathexis flux. In *Proceedings of the Joint Symposium on Virtual Social Agents*, pages 130–136. AISB, The Society for the Study of Artificial Intelligence and the Simulation of Behaviour, 2005.
- [31] S.J. Rosenschein. Intelligent agent architecture. In Rob Wilson and Frank Keil, editors, *The MIT Encyclopedia of Cognitive Sciences*, pages 411–412. The MIT Press, Cambridge, Massachusetts, 1999.
- [32] Stuart Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 1995.
- [33] Stuart C. Shapiro. *Encyclopedia of Artificial Intelligence*. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition, 1992.
- [34] J. F. Sowa. *Conceptual structures: information processing in mind and machine*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.
- [35] Juan D. Velásquez and Pattie Maes. Cathexis: a computational model of emotions. In *Proceedings of the first international conference on Autonomous agents*, AGENTS '97, pages 518–519, New York, NY, USA, 1997. ACM.
- [36] Gerd Wagner. The aois glossary. In Gerd Wagner, Yves Lesperance, and Eric Yu, editors, *Proceedings of the Agent-Oriented Information Systems Workshop (AOIS) at the 17th National Conference on Artificial Intelligence*, pages 165–181, Berlin, 2000. iCue.

- 
- [37] Gerhard Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.
- [38] U. Wilensky and IL. Evanston. Netlogo. center for connected learning and computer based modeling. Technical report, Northwestern University, 1999. <http://ccl.northwestern.edu/netlogo>.
- [39] S. Woollams and M. Brown. *TA: the total handbook of transactional analysis*. Prentice Hall, London, 1979.
- [40] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning, i–ii. *Information Science*, 8:199–249, 301–357, 1975.
- [41] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

**Universidad Autónoma de Baja California**  
**FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA**  
**COORDINACIÓN DE POSGRADO E INVESTIGACIÓN**

FOLIO No. 064

Tijuana, B. C., a 25 de agosto de 2011

**C. IVÁN ESPINOZA HERNÁNDEZ**  
Pasante de: Maestro en Ciencias  
Presente

El tema de trabajo y/o tesis para su examen profesional, en la  
Opción TESIS

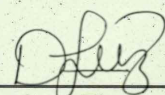
Es propuesto, por los C. Dres. Dora Luz Flores Gutiérrez y Manuel Castañon Puga  
Quienes serán los responsables de la calidad de trabajo que usted presente,  
referido al tema Comunicación Entre Agentes: Envío e interpretación de Mensajes  
Usando Redes Semánticas y Lógica Difusa.

el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

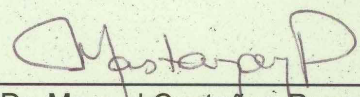
- I.- INTRODUCCIÓN
- II.- FUNDAMENTOS TEÓRICOS
- III.- MODELO DE MENSAJE Y DESARROLLO DE HERRAMIENTAS PARA SU INTERPRETACIÓN
- IV.- CASOS DE ESTUDIO
- V.- CONCLUSIONES Y TRABAJO FUTURO

  
\_\_\_\_\_  
Q. Noemí Hernández Hernández

**Sub-Director Secretario**

  
\_\_\_\_\_  
Dr. Dora Luz Flores Gutiérrez

**Director de Tesis**

  
\_\_\_\_\_  
Dr. Manuel Castañon Puga

**Co-Director de Tesis**

  
\_\_\_\_\_  
Dr. Luis Enrique Palafox Maestre

**Director**