



UNIVERSIDAD DE BELGRANO

Las tesinas de Belgrano

**Facultad de Tecnología Informática
Ingeniería Informática**

AI4U Inteligencia artificial para todos

Nº 261

Matías Gabriel Chernovagura

Tutor: Juan Lestani

Departamento de Investigaciones
Febrero 2009

Resumen

En este trabajo se analizan tanto los agentes inteligentes como los sistemas multiagentes y se propone mediante una minuciosa descripción el diseño de un sistema que aplicando esta tecnología pueda acercar la Inteligencia Artificial a las necesidades e intereses del usuario común.

Palabras clave: inteligencia artificial; agentes inteligentes; sistemas multiagentes.

Índice

1 – Introducción	7
2 – Hipótesis	7
3 - La inteligencia artificial – Marco teórico	7
3.1 – Orígenes	8
3.2 - Actuando humanamente	9
3.3 - Pensando humanamente	9
3.4 - Pensando racionalmente	9
3.5 - Actuar racionalmente.....	10
3.6 - Inteligencia artificial distribuida	10
4 - Los Agentes Inteligentes	10
4.1 – Historia	10
4.2 – Definición	11
4.3 - Los agentes y su entorno	12
4.4 - Propiedades de los ambientes	12
4.5 – Racionalidad.....	13
4.6 - Medidas de <i>performance</i>	14
4.7 - El paradigma de la programación orientada a agentes	14
4.7.1 - Agentes vs. Objetos	14
4.8 - Agentes vs. Sistemas expertos	15
4.9 - Lenguajes de agente	15
4.9.1 - Lenguajes de programación de la estructura del agente	15
4.9.2 - Lenguajes de comunicación de agentes	15
4.9.3 - Lenguajes de programación del comportamiento del agente	16
4.10 - Estructura de los agentes	16
4.10.1 - Los elementos de aprendizaje como parte de la estructura de un agente.....	17
4.11 - Resolución de problemas mediante la búsqueda.....	18
4.12 - Funciones heurísticas	19
4.13 - Agentes característicos	20
4.13.1 - Agentes basados en el conocimiento	20
4.13.2 - Agentes basados en la lógica proposicional	20
4.13.3 - Agentes basados en circuitos.....	20
4.14 - Representación del conocimiento	21
4.15 - La planificación	21
4.16 - Funciones de la utilidad	21
4.17 - Actuando bajo incertidumbre.....	22
4.18 - Redes de decisión.....	22
4.19 - Formas de aprendizaje	22
4.19.1 - Aprendizaje inductivo	23
4.19.2 - Aprendizaje mediante árbol de decisión.....	23
4.20 – Comunicación.....	23
4.21 - Análisis sintáctico	24
4.22 - Interpretación semántica.....	24
4.23 - Interpretación pragmática	24
4.24 - Procesamiento probabilístico del lenguaje.....	24
4.24.1 - Recuperación de información.....	25
4.24.2 - Extracción de información	25
4.24.3 - Traducción de máquina	25
5 - Sistemas multiagentes	26
5.1 – Definición.....	27
5.2 – Ventajas.....	27
5.3 – Arquitectura	27
5.4 - Relaciones entre agentes	27

5.4.1 – Coordinación entre agentes	28
5.5 - Sistemas multiagentes centralizados vs. sistemas multiagentes descentralizados	29
5.6 - DPS vs. MAS	29
5.7 - Modelos de sistemas multiagentes	29
5.7.1 - El Modelo OMG	30
5.7.2 - El Modelo FIPA	30
5.7.3 - El Modelo KAoS	30
5.7.4 - El Modelo General Magic	30
5.8 - Diseño de una Arquitectura de sistemas multiagentes	30
5.9 - Interacción entre agentes	31
5.10 – Comunicación	31
5.11 - ARPA – KQML	31
5.11.1 - KQML	31
5.11.2 - Inconvenientes de KQML	33
5.11.3 – KIF	33
5.11.4 - Ejemplo KQML/KIF	34
5.12 - Mecanismos y servicios de transporte	34
5.13 – Coordinación	35
5.13.1 - Formas de coordinación	35
5.14 - Construcción de un sistema multiagente	36
5.14.1 - Herramientas para la construcción de agentes	36
5.15 – FIPA	36
5.15.1 - FIPA – ACL	36
5.15.2 - FIPA – Arquitectura abstracta	38
5.15.3 - FIPA - Modelo de referencia de agentes	39
5.15.4 - FIPA – Protocolos de interacción	40
5.16 - Sistemas de agentes heterogéneos	40
5.17 - Ejemplo de arquitectura: MAGGIS	40
5.18 - El proyecto MIX: enfoque multiagente para sistemas híbridos	41
5.19 - Agentes híbridos	43
5.20 - Arquitectura modular	43
5.21 - Sistemas de producción	43
5.22 - Código de buena conducta en los agentes	44
5.23 - Aplicaciones de agentes	44
5.23.1 - Agentes asistentes personales	44
5.23.2 - Agentes móviles	45
5.23.3 - Agentes de información	45
5.23.4 - Agentes en internet	45
5.23.5 – Otros casos puntuales	45
5.24 – Crecimiento académico	46
6- AI4U – Inteligencia Artificial para todos	48
6.1 – ¿Por qué un Sistema Multiagente?	48
6.2 – Asistente	49
6.3 – FIPA	50
6.4 - Comunicación con el usuario	50
6.5 – Arquitectura	50
6.6 - Tipos de sistemas multiagentes	52
6.6.1 - Funciones globales	53
6.6.2 - Sistema multiagente que ayude al estudiante	54
6.6.3 - Sistema multiagente vendedor	55
6.6.4 - Sistema multiagente que ayude en el trabajo	56
6.6.5 - Sistema multiagente para ayuda en investigaciones	57
6.6.6 - Sistema multiagente para soporte técnico y asistencia al cliente	57
6.7 - Comunicación interna	58
6.8 - Agente de comunicación	58
6.9 - Reconocimiento de voz	59
6.9.1 – ConceptNet	60

6.10 - Capacidad del habla	60
6.11 – Animación	61
6.12 – Exportación	61
6.13 – Actualizaciones.....	61
6.14 – Prototipo no funcional.....	61
7- Conclusión.....	62
8 – Bibliografía.....	63

1 - Introducción

¿Quién no imaginó alguna vez poder tener en la computadora un pequeño personaje que lo pudiera ayudar a solucionar sus necesidades dentro de su ambiente profesional? ¿Que lo ayude a encarar sus problemas, tanto laborales como de estudio, e incluso que pueda vender sus productos o atender a los clientes por *Messenger*?

Imaginemos además, que ese personaje es tan inteligente que no sólo nos permite interactuar con él (podemos hablarle y lograr incluso que nos conteste) sino que aprende solo, trabaja sin siquiera necesitar nuestra intervención y resuelve solo sus problemas...

Y vamos un poco más allá, porque si además logramos que cualquier persona pueda tenerlo, configurarlo y usarlo, habremos contribuido notoriamente a solucionar algunos de los problemas más comunes que tienen actualmente quienes necesitan de la computación para desarrollar su trabajo dentro de los distintos ámbitos de la vida social moderna .

Éstos son los planteos básicos que se proponen con el desarrollo de este trabajo, ya que no sólo se pretende profundizar en el área de los agentes inteligentes y el significado revolucionario de la Inteligencia Artificial, sino descubrir el mundo de los sistemas multiagentes para conocer cómo se puede diseñar un sistema multiagente sencillo, utilizable por todo tipo de usuario que requiera, para su obtención, sólo seguir un sencillo asistente.

Por otra parte, quisiera aclarar que en esta tesina me pareció apropiado combinar algunos de los aportes tanto de la Tesina Monográfica como de la de Aplicación Práctica ya que, debido a que el tema planteado es algo muy nuevo en el medio, resultó difícil reunir la información y posibilidades existentes sin perder de vista nuestro objetivo de recopilar y concentrar la información sobre un tema específico.

Así la primera parte de este trabajo está basada en una recopilación bibliográfica, comprensible y crítica de esta tecnología, ordenada de forma tal que, desde la definición de un agente, se llegue hasta algunos ejemplos de estructuras de sistemas multiagentes para poder acceder al problema en forma gradual.

Por último, se detallará todo el proyecto que justifica la hipótesis, encarándolo con los esquemas de una tesina de Aplicación Práctica, puesto que este trabajo se complementa con la presentación de un prototipo no funcional siguiendo las pautas del proyecto descrito.

2 - Hipótesis

La Inteligencia Artificial ha llegado a un nivel de madurez tal que permitiría generar sistemas inteligentes y la prueba de todo esto se encuentra en la nueva tecnología de Agentes Inteligentes.

Así, mediante el desarrollo de una arquitectura específica se podría crear un sistema multiagente tal, que mediante el uso de un asistente que lo genere, permita convertir a esta tecnología en una accesible, personalizable y masiva.

Estos sistemas multiagentes creados por los mismos usuarios contarían con todas las facultades de los agentes, por lo que, incluso aprender, adaptarse a los gustos de su usuario y auto perfeccionarse en su funcionamiento por su naturaleza de entidad inteligente.

3 - La inteligencia artificial - Marco Teórico

El campo de la Inteligencia Artificial es una de las ciencias más nuevas e intenta no sólo entender, sino construir entidades inteligentes, de modo que es necesario indicar las teorías que están presentes en las investigaciones anteriores sobre el tema, tanto en forma explícita como implícita.

Si queremos señalar el momento en que surgió el concepto de Inteligencia Artificial, sin dudas debemos remontarnos al período posterior a la Segunda Guerra Mundial. De hecho, el nombre fue acuñado formalmente en 1956 durante la famosa conferencia de Dartmouth, organizada por John McCarthy.

En dicha conferencia, una nueva generación de investigadores (Minsky, McCarthy, Newell, Simon, Samuel, Rochester, Shannon, Solomonoff, Selfridge y More) definieron las directrices y líneas de actuación futuras en el ámbito de la Inteligencia Artificial.

Una primera definición de la Inteligencia Artificial nos lleva a considerarla como una disciplina científico-técnica que trata de introducir sistemas artificiales capaces de realizar conductas que, de ser ejecutadas por seres humanos, se entendería que han requerido del uso de la inteligencia. Además debe ser consi-

derada como el estudio de los dispositivos de la inteligencia y las tecnologías que lo mantienen.

Pero esta definición es insuficiente, de modo que al investigar otras posibilidades se encuentra tal cantidad que encaramos su clasificación para poder entenderlas mejor. Para esto, por un lado ponemos

- 1) Las que se refieren a los procesos de pensamiento y al razonamiento y
- 2) Las que apuntan al comportamiento.

De esta manera se puede desarrollar la siguiente tabla:

<i>Sistemas que piensan como humanos</i>	<i>Sistemas que piensan racionalmente</i>
<p>“El excitante nuevo esfuerzo por hacer a las computadoras pensar... <i>máquinas con mentes</i>, en un sentido completo y literal” (Haugeland, 1985)</p> <p>“]La automatización de[actividades que tenemos asociadas al pensamiento humano, actividades como la toma de decisiones, la resolución de problemas, aprender,...” (Bellman, 1978)</p>	<p>“El estudio de las facultades de la mente a través del uso de modelos computacionales” (Charniak y McDermott, 1985)</p> <p>“El estudio de la computación que hace posible percibir, razonar y actuar” (Winston, 1992)</p>
<i>Sistemas que actúan como humanos</i>	<i>Sistemas que actúan racionalmente</i>
<p>“El arte de crear máquinas que realizan funciones que requieren de la inteligencia al ser realizadas por la gente” (Kurzweil, 1990)</p> <p>“El estudio de cómo lograr que las computadoras hagan cosas que, hasta el momento, son realizadas mejor por personas” (Rich y Knight, 1991)</p>	<p>“Inteligencia Computacional es el estudio del diseño de agentes inteligentes” (Poole y otros, 1998)</p> <p>“IA... es respecto al comportamiento inteligente de artefactos” (Nilsson, 1998)</p>

Algunas definiciones de Inteligencia Artificial, organizada en cuatro categorías¹

3.1 - Orígenes

Haciendo una breve reseña histórica, se puede señalar que desde sus comienzos hasta la actualidad, la Inteligencia Artificial ha tenido que hacer frente a una serie de problemas.

Así, en el año 1955 Simón, Newell y Shaw, propusieron el primer lenguaje de programación encaminado a la resolución de dificultades de la Inteligencia Artificial, el IPL-11. Un año más tarde estos tres científicos desarrollaron el primer programa de Inteligencia Artificial al que llamaron Logic Theorist, el cual resultó decisivo para sus investigaciones.

El programa era apto para demostrar teoremas matemáticos mediante la representación de cada problema como un modelo de árbol en el que se seguían ramas en busca de la solución correcta.

En 1957 McCarthy desarrolló el lenguaje LISP y la empresa IBM designó un equipo para la investigación en esa área.

En 1961 se creó SAINT (Symbolic Automatic INTEgrator) por Slagle el cual se orienta a la demostración simbólica en el área del álgebra y en 1964 Bertrand construyó el sistema SIR (Semantic Information Retrieval) capaz de entender oraciones en inglés.

Estos programas obviamente corrían en un computador y se utilizaron, por ejemplo, en el control robótico, la comprensión de lenguajes naturales, el procesamiento de imágenes basado en conocimientos previos, las estrategias de juegos, etc.

1. Stuart Russell & Peter Norvig, *Artificial Intelligence: A Modern Approach (Second Edition)*, Prentice Hall, 2003.

3.2 - Actuando humanamente

Un momento importante se da en el año 1950, cuando Alan Turing, propuso un reto al que se conoce como el "Test de Turing", y el objetivo de ese test era proveer de una operación que definiera, de forma satisfactoria o no, la existencia de la inteligencia artificial.

Una computadora podría pasar este test si luego de competir con un ser humano una serie de preguntas y respuestas por escrito, no se pudiera distinguir si las respuestas vienen de una computadora o de otro humano.

De esta manera se puede deducir que la computadora debería tener las siguientes capacidades:

- **Procesamiento de lenguaje natural** que le permita comunicarse exitosamente. Cabe destacar que al referirnos a un lenguaje natural estamos considerando el inglés, siguiendo la definición de Turing,
- **Representación de conocimiento** para almacenar lo que conoce o escucha
- **Razonamiento automático** para utilizar la información almacenada y responder las preguntas, además de trazar nuevas conclusiones,
- **Aprendizaje de máquina** para adaptarse a nuevas circunstancias y para detectar y extrapolar patrones.

El Test de Turing indica expresamente que no debe haber ningún tipo de interacción entre el interrogador y la computadora, ya que según Turing, la simulación física de una persona no forma parte de la inteligencia.

A pesar de esto, existe el Test de Turing Total, en el que se incluye una señal de video, así el interrogador suma a su prueba las habilidades de percepción y toda referencia visual física que se presente, como el paso de objetos.

Por esto, una computadora, para pasar el Test de Turing Total necesitaría además de:

- **Visión computarizada** para percibir los objetos,
- **Robótica** para manipular los objetos y moverse con respecto a estos.

Quisiera agregar que a pesar de tener más de 50 años, el test de Turing sigue siendo siempre considerado como un desafío todavía planteado y por eso estas seis disciplinas todavía forman parte de las investigaciones y desarrollos que se realizan dentro del marco de la inteligencia artificial.

3.3 - Pensando humanamente

Por otra parte, está claro que antes de poder determinar si una computadora piensa como los humanos, se debería conocer justamente cómo piensan los humanos.

Así, según lo establecido por la psicología acerca de la forma de estudiar a la mente humana, existen dos maneras de hacerlo: 1) a través de la introspección 2) por medio de experimentos psicológicos.

Por otra parte, existe un campo interdisciplinario de la ciencia, llamado Ciencia Cognitiva, que estudia tanto modelos computarizados desde la Inteligencia Artificial como técnicas experimentales de la Psicología para intentar construir teorías precisas y testeables sobre el funcionamiento de la mente humana.

En sus inicios, se intentó mantener juntas a la Inteligencia Artificial y a la Ciencia Cognitiva, pero esta última necesita basarse en investigaciones y experimentos sobre humanos o animales, mientras que la Inteligencia Artificial sólo tiene acceso a las computadoras. Por esto, ambos campos fueron separados posteriormente lo cual los benefició ya que pudieron evolucionar mucho más rápido aunque constantemente se realimentan de sus propias investigaciones.

3.4 - Pensando racionalmente

En cuanto a la racionalidad del pensamiento, nos podemos remontar hasta Aristóteles para poder hablar del "pensamiento correcto", que es cuando este filósofo se refiere a los procesos de pensamiento irrefutables, ya que sus silogismos y todos los métodos posteriores que dieron origen a la lógica, plantearon una manera de operar sobre pensamientos siguiendo patrones irrefutables de razonamiento.

Ya en la época moderna, alrededor del año 1965 se desarrollaron programas que, mediante todas las notaciones y reglas lógicas desarrolladas por los expertos, podían resolver cualquier problema descrito con notación lógica. De esta manera, los expertos de la lógica esperaban poder construir sistemas inteligentes utilizando estas técnicas.

Pero había dos grandes obstáculos: primero, la dificultad de representar exclusivamente con la notación formal de la lógica el conocimiento informal, particularmente cuando el conocimiento no es 100% cierto, y en segundo lugar, existe una gran diferencia entre resolver un problema "en principio" y realizarlo en la práctica.

3.5 - Actuar racionalmente

Al hablar del actuar racionalmente dentro de la Inteligencia Artificial vamos a adelantarnos un poco hacia el concepto del Agente Inteligente, que se desarrollará profundamente mas adelante, para poder contar con un elemento de la Inteligencia Artificial con capacidad de acción racional.

Siguiendo estos razonamientos, consideraremos que un agente es simplemente algo que actúa, de hecho la palabra viene del latín *agere*, que significa “para hacer”.

Es decir que un agente de computadora es mucho más que un simple programa, porque se espera que opere de forma autónoma, perciba su ambiente, persista durante un periodo prolongado, se adapte a cambios y sea capaz además de tomar otros objetivos. Un agente racional es aquel que actúa buscando el mejor resultado, o, dependiendo de las circunstancias, el mejor logro esperado.

De modo que el estudio y la aplicación de la Inteligencia Artificial para el diseño de agentes racionales presentan por lo menos dos grandes ventajas:

1) Son más generales que el acercamiento a través de las “Leyes del Pensamiento” porque la deducción por lógica es solo uno de los varios mecanismos posibles que existen para lograr la racionalidad.

2) Los desarrollos científicos que se acercan al comportamiento o pensamiento humano son considerados mas “tratables”, porque el estándar de racionalidad se encuentra claramente definido.

3.6 - Inteligencia artificial distribuida

Según la definición utilizada en el Departamento de Sistemas e Informática de la Universidad Nacional de Rosario (República Argentina), se considera Inteligencia Artificial Distribuida a “la disciplina dirigida al desarrollo de métodos y técnicas para la solución de problemas complejos por medio del comportamiento inteligente de un sistema integrado por unidades llamadas agentes”.²

Por otra parte, según Carlos Fernández, de la Universidad Politécnica de Madrid (España), la Inteligencia Artificial busca la resolución distribuida de problemas donde se estudia la solución de problemas de forma colaborativa por un grupo distribuido de entidades o agentes inteligentes, situación que ocurrirá ya que la mútua colaboración es necesaria porque cada agente no es capaz de resolver el problema por sí mismo o porque es más rentable la resolución conjunta.³

4 - Los Agentes Inteligentes

El Dr. Nicholas Jennings, al recibir el premio al mejor investigador novel en el Congreso Internacional de Inteligencia Artificial celebrado en Estocolmo (Suecia) en el año 1999 dejó en su discurso un concepto muy claro en lo que se refiere al potencial futuro de los agentes inteligentes: “*Los agentes constituyen el próximo avance más significativo en el desarrollo de sistemas y pueden ser considerados como la nueva revolución en el software*”.

Al igual que ocurre con la propia definición de la Inteligencia Artificial, se pueden encontrar propuestas de un gran número de definiciones del concepto de agente, sin que ninguna de ellas haya sido plenamente aceptada por la comunidad científica, siendo quizás la más simple la de Russell, que considera un agente como una entidad que percibe y actúa sobre un entorno⁴.

Los agentes además pueden ser vistos como un nuevo paradigma de la Ingeniería de Software porque se deben diseñar e implementar sistemas complejos distribuidos. Una última forma de verlos es como herramientas para entender sociedades humanas ya que permiten una forma de simular este tipo de asociación.

4.1 - Historia

Entendiendo entonces que la Inteligencia Artificial Distribuida busca la solución repartida de los problemas, y sumándose a la capacidad de la distribución del conocimiento y del entendimiento el hecho de que no hay sociedad sin inteligencia, llegaremos al concepto de los agentes inteligentes.

Es decir que el crédito de este nacimiento debe atribuirse a la interacción que se produjo entre la inteligencia artificial, la ingeniería de software, las bases de datos y los sistemas distribuidos, pero también influyeron en su nacimiento algunas disciplinas no informáticas como la sociología, la psicología y la economía las que aportaron muchos años de aprendizaje y conocimiento para el desarrollo de esta nueva tecnología.

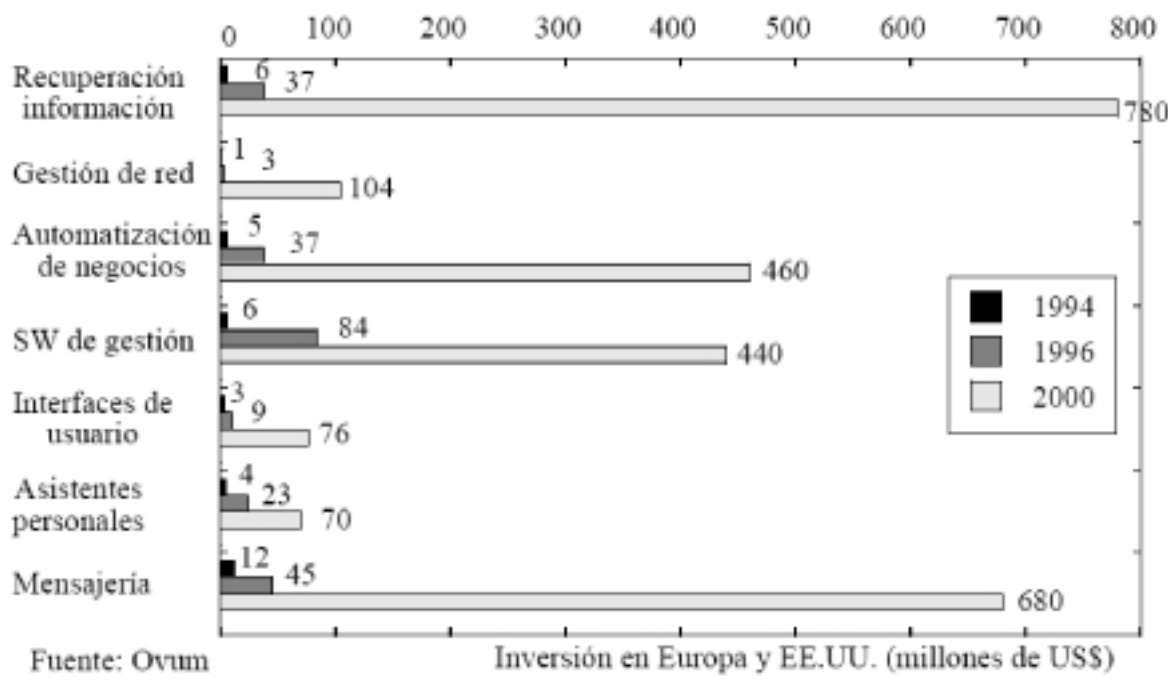
En cuanto al crecimiento del área, algunas de las razones están en el progreso llevado a cabo por las tecnologías complementarias, sobre todo por Internet donde la distribución de la información y las tecnologías asociadas conducen de una forma casi ideal a la utilización de los sistemas basados en agentes.

2. Departamento de Sistemas e Informática, “Agentes Inteligentes”, *Introducción a la Inteligencia Artificial*, Licenciatura en Ciencias de la Computación, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, 2004.

3. Carlos Ángel Iglesias Fernández, “Introducción a la Inteligencia Artificial Distribuida”, *Sistemas Inteligentes*, Grupo de Sistemas Inteligentes, Universidad Politécnica de Madrid, 1999.

4. Michael Wooldridge, *An Introduction to MultiAgent Systems*, John Wiley & Son, LTD, 2002.

Si se quiere conocer en número este crecimiento, en el siguiente cuadro se pueden comparar las inversiones realizadas en los años 1994, 1996 y 2000 en distintos cambios de la informática. Un simple análisis del cuadro nos lleva a entender que existe una marcada proyección en ciertos sectores, los cuales justamente corresponden a funciones en las cuales los Agentes Inteligentes podrían participar e incluso liderar.



Representación de la inversión en áreas informáticas⁵

Por otro lado, los avances en las tecnologías de objetos distribuidos han proporcionado una infraestructura sin la cual el desarrollo de sistemas de agentes a gran escala, llegaría a ser mucho más difícil y menos efectivo.

4.2 - Definición

En este punto, cabe aclarar que un agente no es un programa o al menos que es algo más que un programa ya que al fin y al cabo, todo lo que se ejecuta en una computadora es un programa. Es decir, no se lo puede llamar "un programa" en el mismo sentido en que a un sistema operativo no se lo puede llamar "un programa".

Así, una de las características que lo distinguen de los programas es la **autonomía**, lo cual implica dos cosas:

- 1) Que son proactivos, es decir, que no sólo actúan respondiendo a una acción del usuario, sino que también actúan siguiendo sus propios objetivos,
- 2) Que son persistentes, o sea, que no se pueden "apagar"; incluso aunque el usuario no esté interactuando con ellos, los agentes siguen funcionando, recolectando información, aprendiendo y comunicándose con otros agentes.

Al referirnos a su inteligencia hablamos de que pueden afectar lo que se perciba en el futuro, por lo menos refiriéndonos a la capacidad de aprender de los errores para no volver a cometerlos. También implica un bucle percepción/procesamiento de información/acción, el cual representa un comportamiento similar al de cualquier ser vivo medianamente inteligente.

Basándonos en estos conceptos, se pueden caracterizar distintos agentes de acuerdo a los atributos que posean (y que van a definir su comportamiento) para resolver un determinado problema.

Dentro de las muchas caracterizaciones de agentes que se pueden encontrar, vamos a repasar las características enunciadas por Franklin, Graesser y Nwana⁶, ya que son las que marcan, actualmente, la tendencia en las investigaciones:

- **Continuidad Temporal:** se considera un agente un proceso sin fin, ejecutándose continuamente y desarrollando su función.

5. Carlos Ángel Iglesias Fernández, "Introducción a la Inteligencia Artificial Distribuida", Universidad Politécnica de Madrid, 1999.

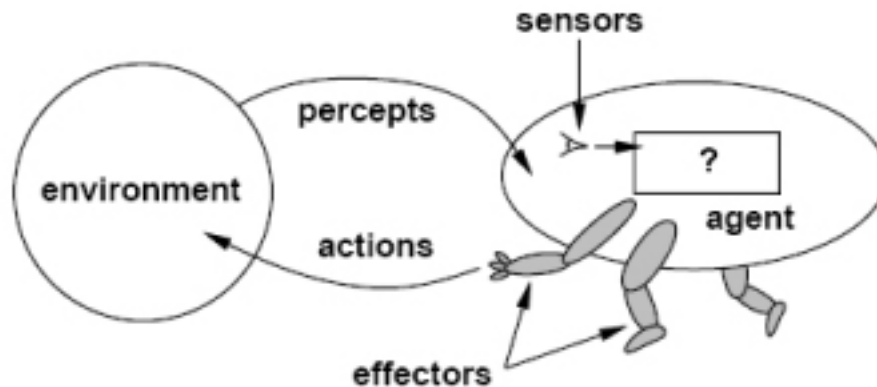
6. Franklin, Stan & Graesser, "Is it an agent or just a program?: A taxonomy for autonomous agents", *Intelligent Agents III: Agent Theories, Architectures, and Languages*, Springer-Verlag, 1997.

- **Autonomía:** un agente es autónomo en la medida en que su comportamiento está determinado por su propia experiencia y no sólo por el conocimiento que le ha sido incorporado. El agente es capaz de adaptarse aunque el entorno cambie severamente.
- **Sociabilidad:** este atributo permite a un agente comunicarse con otros agentes o incluso con otras entidades.
- **Racionalidad:** el agente siempre realiza “lo correcto” a partir de los datos que percibe del entorno.
- **Reactividad:** un agente actúa como resultado de cambios en su entorno. En este caso, un agente percibe el entorno y esos cambios dirigen el comportamiento del agente
- **Pro-actividad:** un agente es pro-activo cuando es capaz de controlar sus propios objetivos a pesar de cambios en el entorno.
- **Adaptatividad:** está relacionado con el aprendizaje que un agente es capaz de realizar y, si puede, cambiar su comportamiento basándose en ese aprendizaje.
- **Movilidad:** capacidad de un agente de trasladarse a través de una red telemática.
- **Veracidad:** asumir que un agente no comunica información falsa a propósito.
- **Benevolencia:** aceptar que un agente está dispuesto a ayudar a otros agentes si esto no entra en conflicto con sus propios objetivos.

4.3 - Los agentes y su entorno

Se denominan “censores” a los elementos que utilizan los agentes para percibir su ambiente y “actuadores” a los que utiliza para intervenir sobre los mismos. Es decir que un agente robótico utilizaría cámaras e infrarrojos como sensores y varios motores como actuadores mientras que un agente de software recibe a través del teclado, archivos o paquetes de red por ejemplo y actúa por medio de su pantalla o tal vez escribiendo archivos.

Por otra parte, se utiliza el término “percepción” para referirse a cualquier entrada de percepción que recibe el agente en cualquier momento, y el de “secuencia de percepción” al historial de todas las percepciones que el agente ha recibido, tal como lo mostramos en el gráfico adjunto:



Interacción de un agente con su ambiente⁷

Ante esta relación entre un agente y su entorno, se puede llegar a entender la interacción como una función matemática donde nacería una tabla que establece la acción de los actuadores en relación a cierta percepción de los sensores.

Por supuesto que hay que tener muy claro que esta tabla sólo es una caracterización externa del agente, ya que internamente las funciones son implementadas por el programa del agente.

4.4 - Propiedades de los ambientes

En cuanto al ambiente sobre el cual se desarrollará la vida del agente, se advierte que influirá de forma constante en éste por lo que es muy importante analizarlo detalladamente para conocer los desafíos que puede enfrentar el agente durante su vida y poder así optimizar su diseño.

Un ambiente puede ser clasificado de las siguientes maneras:

- **Completamente observable o parcialmente observable**

Un ambiente es completamente observable cuando los sensores del agente tienen acceso a toda información relevante para su funcionamiento que exista en el ambiente durante todo el tiempo, mientras que se considera a un ambiente parcialmente observable cuando existe algún obstáculo que impide a alguno de los sensores del agente percibir su información o simplemente si esa información está ausente.

⁷ M. Wooldridge, *An Introduction to Multiagent Systems*, J. Wiley & Son., 2002.

- **Determinista o estocástico**

Se denomina determinista a un ambiente si su próximo estado se encuentra determinado por el actual y la acción que realice el agente. En caso contrario, dicho ambiente es estocástico. Por ejemplo, un agente que conduce un taxi claramente vive en un ambiente estocástico ya que no se puede predecir el futuro comportamiento del taxi.

- **Episódico o secuencial**

En un ambiente episódico la experiencia del agente se encuentra dividida en episodios atómicos, entre los cuales no existe ninguna relación. Cada uno de estos episodios consiste en una percepción del agente y luego por una acción de éste. Por otro lado, en un ambiente secuencial una decisión del agente puede afectar a todas las futuras decisiones, como por ejemplo un agente que juega al ajedrez.

- **Estático o dinámico**

Se denomina a un ambiente dinámico si el mismo puede cambiar mientras el agente se encuentra decidiendo la acción a tomar, mientras que por el contrario, se lo llama estático cuando el ambiente no puede realizar ningún cambio hasta que el agente realice la acción elegida.

- **Discreto o continuo**

Esta clasificación puede ser asignada al estado del ambiente, al tiempo que se maneja, a las percepciones y a las acciones del agente. Por ejemplo, un agente que juega al ajedrez se encuentra sobre un ambiente con estados discretos, mientras que un agente que maneja un taxi se mueve en un ambiente con estados y tiempo continuo.

- **Agente simple o multiagente**

La distinción puede ser relativamente clara: un sistema se considera multiagente cuando se encuentran dos o más agentes teniendo una relación, la cual puede ser de competencia o de cooperación. Por ejemplo, dos agentes que juegan al ajedrez, se encuentran formando un sistema multiagente donde mantienen una relación competitiva ya que cada uno busca maximizar su beneficio en el juego, lo cual lleva a la minimización del otro.

4.5 - Racionalidad

Por otra parte, un agente debe hacer siempre lo correcto de acuerdo a sus percepciones. Con esto se entiende que el agente racional, para ser tal, debe emprender la mejor acción posible en una situación dada.

Se pueden encontrar cuatro puntos de los que depende la racionalidad de un agente:

- Las medidas de performance que definen los criterios del éxito.
- El conocimiento previo del agente sobre su ambiente.
- Las acciones que el agente puede desempeñar.
- La secuencia de percepción del agente hasta el momento.

De acuerdo a lo señalado anteriormente, coincidimos con la siguiente definición propuesta para caracterizar un agente racional:

“Un agente racional debe seleccionar para cada posible secuencia de percepción una acción de la cual se espere que maximice su medida de performance, el cual es determinado por las evidencias provistas por la secuencia de percepción y cualquier cosa incorporada a los conocimientos que el agente posee”⁸

Cabe aclarar que al referirnos a la racionalidad en un agente, es muy importante no confundir esta idea con el concepto de omnisciencia, ya que por ejemplo, puedo estar nadando en un lago conociendo la corriente, la profundidad, mi cansancio y muchísimos factores más pero decido seguir nadando un poco más. De golpe una bala perdida de unos cazadores que estaban cerca me mata. Y entonces cabe la pregunta: ¿fue irracional mi decisión de seguir nadando?

Es decir que un agente va a trabajar siempre utilizando todo el conocimiento al que tiene acceso pero esta situación no lo lleva a ser omnisciente y poder asegurar un 100% de efectividad en sus decisiones racionales.

Ahora que sabemos que el agente busca la racionalidad en sus actos, podemos clasificar tres formas distintas con las que el agente puede contar para razonar:

- **Razonamiento progresivo**, donde se va desde situaciones iniciales hasta situaciones finales.
- **Razonamiento regresivo**, donde se va desde situaciones finales hasta situaciones iniciales.
- **Razonamiento bidireccional**, utilizado en problemas complejos, donde se avanza un poco en forma progresiva desde situaciones iniciales, un poco en forma regresiva desde situaciones finales, y así sucesivamente, hasta que, en algún paso, los caminos de inferencia se encuentren.

8. Stuart Russell & Peter Norvig, *Artificial Intelligence: A Modern Approach (Second Edition)*, Prentice Hall, 2003.

4.6 - Medidas de *performance*

Ya señalamos que un agente racional es aquel que “hace lo correcto”. Ahora bien, “hacer lo correcto” no tiene una definición fija y clara, por lo que es necesario definir las medidas de *performance* que determinen un nivel de adecuación para las posibles acciones y saber así, qué tan correcta es cada opción con la que se cuenta.

De modo que estas medidas de *performance* van a contener los criterios necesarios para determinar el éxito o no del comportamiento del agente, considerando que estas medidas no son adaptables a todos los agentes, sino que cada agente necesita sus propias medidas que lo lleven a determinar si, por ejemplo, cierta secuencia de acciones lo llevan al éxito o no.

Cabe agregar que la selección de las medidas de *performance* no es una tarea sencilla. De hecho, intentar medir el éxito de una persona sobre una acción está totalmente sujeto a los criterios únicos de cada individuo; por ejemplo, medir el éxito de la limpieza del hall de un edificio, seguramente encontrará tantas medidas de *performance* como habitantes tenga el edificio, sin contar el criterio propio del portero que será el que irá realizando las evaluaciones del éxito según sus propios conceptos mientras realiza su trabajo.

4.7 - El paradigma de la programación orientada a agentes

Yoav Shoham, profesor de la Universidad de Stanford, ha propuesto un nuevo paradigma de programación⁹, basándose en una vista social de la computación: el de la programación orientada a agentes.

La idea clave del paradigma es la programación directa de agentes en términos de las nociones mentales e intencionales que las teorías de agente han desarrollado para representar las propiedades de los mismos. El motivo de tal propuesta es que, el ser humano, utiliza la postura intencional como un mecanismo de abstracción para representar las propiedades de sistemas complejos, es decir que de la misma forma que utilizamos la postura intencional para describir al ser humano, podría ser útil su uso para programar máquinas.

La primera implementación de este paradigma fue en el lenguaje de programación AGENT0. En este lenguaje, el agente es especificado en términos de un grupo de capacidades (cosas que el agente puede hacer), un grupo de creencias, uno de propósitos y uno de reglas. El componente clave que determina cómo el agente actúa es el grupo de reglas de compromisos.

4.7.1 - Agentes vs. Objetos

La diferencia entre la programación orientada a objeto y la orientada a agentes puede ser apreciada en la diferencia misma entre un objeto y un agente. Los objetos están programados para hacer lo que está definido cuando se invoca un método mientras que los agentes, al recibir un mensaje, toman la decisión sobre qué hacer, o sea, son autónomos y tienen una mayor flexibilidad en su comportamiento.

En la siguiente tabla se puede encontrar una simple comparación entre los paradigmas de programación orientados a objetos o a agentes donde se distinguen en cada caso las unidades básicas, los parámetros para definir el estado de una unidad básica, el método de procesamiento computacional, el tipo de mensajes y las restricciones en los métodos:

OOP versus AOP

	OOP	AOP
Basic unit	object	agent
Parameters defining state of basic unit	unconstrained	beliefs, commitments, capabilities, choices, ...
Process of computation	message passing and response methods	message passing and response methods
Types of message	unconstrained	inform, request, offer, promise, decline, ...
Constraints on methods	none	honesty, consistency, ...

⁹ Yoav Shoham, *Artificial Intelligence*, Elsevier Science Publishers Ltd., 1993.

Tabla de comparación entre la programación orientada a objetos y la orientada a agentes¹⁰

Por otra parte, los agentes incorporan una noción más fuerte de autonomía que los objetos, en particular deciden si realizar o no una acción. Además son capaces de un comportamiento flexible y los modelos de objetos no incluyen comportamientos.

Así existe una frase utilizada en el medio para recordar esta diferencia: “los objetos lo hacen gratis; los agentes lo hacen porque quieren”.

4.8 - Agentes vs. Sistemas expertos

Al hablar de inteligencia artificial siempre deben ser considerados los sistemas expertos ya que éstos fueron un gran logro de la disciplina.

En efecto, cuando empezaron a surgir los agentes inteligentes hubo quienes los consideraban un versión más pequeña de los sistemas expertos pero está muy claro que hay grandes diferencias entre ambas tecnologías, ya que los sistemas expertos “clásicos” no están acoplados al entorno en el cual actúan y suelen usar un intermediario. Además, no son capaces de un comportamiento reactivo, proactivo y suelen no estar equipados de habilidad social (cooperación, coordinación y negociación).

Sin embargo, existen algunos sistemas expertos, como los que cumplen tareas de control en tiempo real que tienden a comportarse como agentes inteligentes.

En resumen, las principales diferencias entre los agentes y los sistemas expertos, tal como lo define Michael Wooldridge¹¹, son:

- Los sistemas expertos “clásicos” son incorpóreos ya que no se encuentran asociados a ningún ambiente en el que actúan sino que cuentan con el usuario como un “intermediario”.
- Los sistemas expertos, generalmente, no cuentan con las cualidades de reactividad, comportamiento proactivo.
- Los sistemas expertos no son generalmente equipados con una habilidad social en lo que se refiere a cooperación, coordinación y negociación.

4.9 - Lenguajes de agente

Por **lenguaje de agente** entendemos a aquellos que permiten programar sistemas computacionales, en términos de conceptos desarrollados en modelos formales de agentes. Se pueden distinguir dos tipos principales de lenguajes de programación:

- **Lenguajes de agentes de propósito general:** lenguajes destinados a programar agentes genéricos utilizables en cualquier aplicación.
- **Lenguajes de agentes específicos:** lenguajes para un tipo de agente específico, por ejemplo los lenguajes para agentes móviles Telescript o Agent-Tcl.

Además, al referirnos a los lenguajes de programación de agentes se pueden encontrar los siguientes distintos niveles:

- **Lenguajes de programación de la estructura del agente**, los cuales permiten programar las funcionalidades básicas para definir a un agente como las funciones de creación de procesos y funciones de comunicación entre agentes.
- **Lenguajes de comunicación de agentes**, los cuales definen los formatos que serán utilizados por los agentes para el intercambio de mensajes, las primitivas de comunicación y los protocolos disponibles.
- **Lenguajes de programación del comportamiento del agente**, los cuales permiten definir el conocimiento del agente. Esto incluye el conocimiento inicial, las funciones de mantenimiento de dicho conocimiento, las funciones para alcanzar sus objetivos y las funciones para desarrollar habilidades.

4.9.1 - Lenguajes de programación de la estructura del agente

Este nivel de programación normalmente sólo es utilizado por los desarrolladores de una plataforma de desarrollo de agentes y los lenguajes empleados suelen ser lenguajes de propósito general como C, C++, Java, etc. o lenguajes específicos como Prolog/C o CUBL (Concurrent Unit Based Language) dentro del proyecto DAISY.

4.9.2 - Lenguajes de comunicación de agentes

Se pueden encontrar dos tipos de lenguajes de comunicación:

- **Lenguajes procedimentales** que se basan en el intercambio de directivas procedimentales, es decir, un agente recibe un mensaje que implica la ejecución de un procedimiento. Suelen emplear lengua-

10. Yoav Shoham, “Agent-oriented programming”, *Artificial Intelligence 60*, Robotics Laboratory Computer Science Department, University of Stanford, Estados Unidos, 1993.

11. M. Wooldridge, *An Introduction to Multiagent Systems*, J. Wiley & Son., 2002.

jes de intérpretes de órdenes (scripts) como Perl, Tcl, etc., permitiendo un rápido prototipado aunque no suelen ser fácilmente escalables ni reciclables. Son especialmente útiles para la construcción de agentes en aplicaciones finales como agentes de usuario o agentes móviles. Dentro de este enfoque podríamos encontrar a Sodabot o Telescript.

- **Lenguajes declarativos** que se basan en el intercambio de actos comunicativos, es decir, un agente recibe un mensaje con un acto comunicativo que le permite interpretar el contenido del mensaje. El ejemplo más extendido de este enfoque es KQML.

4.9.3 - Lenguajes de programación del comportamiento del agente

Estos lenguajes permiten la programación de los agentes mismos, por lo que en sus facultades se encuentran la posibilidad de definir la estructura del agente, su conocimiento y sus habilidades.

Podemos distinguir los siguientes tipos de lenguajes:

- **De descripción de agente**, los cuales, basándose en que los agentes se derivan de una clase de agente genérica, permiten la definición de los elementos básicos del modelo de agente como la base de conocimiento, grupos de agentes, habilidades del agente, servicios ofrecidos, planes para alcanzar objetivos, etc. La descripción se traduce a un lenguaje ya ejecutable. Como ejemplos de este tipo de lenguaje podemos encontrar a MACE ADL, AgentSpeak y MAST/ADL.
- **De programación orientados a agentes**, los cuales, siguiendo el paradigma la programación orientada a agentes, tienen en cuenta el estado mental del agente para programar las funciones de transición entre estos estados mentales, que consisten en creencias, capacidades y obligaciones. Como ejemplos de este tipo de lenguaje se encuentran AGENT0, PLACA y Agent-K.
- **Basados en reglas de producción**, los cuales son utilizados para la programación de la base de conocimiento de los agentes. Como ejemplos de este tipo de lenguaje se encuentran MAGS, DYNACLIPS (arquitectura de pizarra basada en CLIPS) y RTA, que permite definir las conductas del agente con reglas.
- **De especificación**, los cuales emplean una especificación del agente que se ejecuta directamente para generar su conducta, pudiendo verificar propiedades de la especificación. Podemos decir que, como ejemplos de este tipo de lenguaje, se encuentran METATEM y DESIRE.

4.10 - Estructura de los agentes

Para que un agente pueda existir hacen falta dos cosas: por un lado la arquitectura donde va a funcionar, que puede ser por ejemplo una computadora, además de todos los sensores necesarios, y, por el otro, tenemos el programa agente. Este programa es el verdadero desafío de la Inteligencia Artificial y justamente por esto existen muchos tipos de diseños generales entre los cuales se pueden distinguir cuatro muy característicos:

- **Agentes de reflejo simples**

Este tipo de agente es definitivamente el más sencillo porque simplemente toma una decisión en relación a una percepción, sin importarle el historial de percepciones que pueda tener. Como indica su nombre, este tipo de agente funciona generalmente por reflejos en relación a una llamada "regla de condición-acción". Por ejemplo, el auto de adelante prende la luz de freno; entonces el agente debe frenar.

- **Agentes de reflejo basado en modelo**

Esta clase de agentes es muy efectiva en ambientes parcialmente observables ya que su forma de trabajar es guardando una especie de "estado interno" que depende del historial de percepción. Estos agentes deben conocer cómo "el mundo se desarrolla" de forma independiente del agente y también cómo afectan sus acciones al mundo. Esta forma de conocer "cómo funciona el mundo" es llamado "modelo del mundo" y por esto el nombre de la categoría.

- **Agentes basados en objetivos**

En muchísimos casos a los agentes no les alcanza con conocer el estado actual para poder tomar una decisión. Por ejemplo, un agente que maneja un auto y se encuentra en un cruce, no le es suficiente conocer detalladamente el cruce sino que necesita saber adonde quiere llegar.

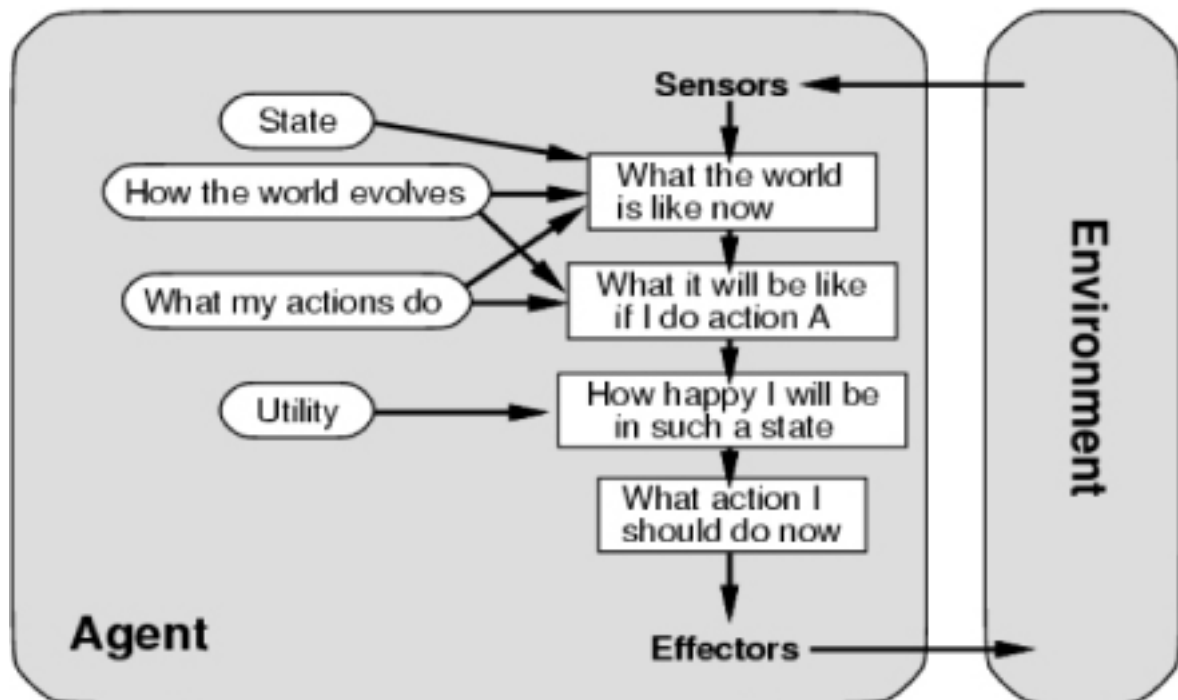
Por este motivo está claro que el agente necesita contar con la información de su objetivo y de esta manera siempre realizar las evaluaciones de las distintas opciones para intentar tomar la decisión correcta que lo lleve hacia su objetivo.

Como el objetivo puede no ser encontrado luego de tomada una sola decisión sino que es necesario recorrer cierto camino de evaluaciones y decisiones, existen dos sub-campos de la Inteligencia Artificial que estudian las técnicas necesarias para saber elegir el mejor camino, una que utiliza la "búsqueda" y la otra mediante la "planificación".

- **Agentes basados en la utilidad**

Tener un objetivo no es suficiente para que un agente cuente con todo lo necesario para tomar la decisión correcta. Volviendo al agente que maneja un auto y tiene como objetivo llegar a un lugar, es casi seguro que este agente cuenta con más de un camino que lo llevará hacia su objetivo. ¿Qué camino debe elegir el agente? Por supuesto el que le represente una mayor utilidad.

Esta función de utilidad tiene el oficio de calificar un estado, o una secuencia de éstos con un número real asociado a un determinado grado de felicidad y es mediante esta calificación que el agente sabrá qué decisión tomar en cada caso.



Modelo de un agente basado en la utilidad¹²

Otra clasificación de arquitecturas de agentes propuesta por Michael Wooldridge¹³ se basa en uno de los aspectos que deben balancear como la percepción/acción y el razonamiento sobre como actuar. En esta categorización encontramos:

- **Deliberativas**, que contienen un modelo simbólico, explícitamente representado del entorno se basa en hipótesis del sistema de símbolos físicos.
- **Reactivas**, que generalmente no incluyen representación simbólica del mundo. Además se considera que la inteligencia real está "situada".
- **Híbridas**, que tratan de combinar las dos aproximaciones.

4.10.1 – Los elementos de aprendizaje como parte de la estructura de un agente

Volviendo a la definición de un agente inteligente, ya aclaramos que una de las características más importantes que distinguen a un programa común de un programa agente es la capacidad de aprendizaje y para que cualquiera de los tipos de agente descritos anteriormente cuenten con esta capacidad, tienen que contar con cuatro componentes: el elemento crítico, el elemento de aprendizaje, el generador de problemas y el elemento de *performance*.

Más adelante nos referiremos con mayor detalle al aprendizaje en los agentes inteligentes, pero sería interesante describir brevemente antes a estos cuatro elementos y a la interacción que se genera entre ellos.

Así, el elemento de aprendizaje es el responsable de realizar las mejoras mientras que el elemento de *performance* es el responsable de tomar las acciones externas. Por otra parte, el elemento de aprendizaje utiliza la retroalimentación desde el elemento crítico acerca de cómo le esta yendo y determina cómo el elemento de *performance* puede ser modificado para realizar mejor la acción la próxima vez.

Es decir que la función del generador de problemas es la de sugerir acciones que llevan a una experiencia nueva e informativa. De esta manera se va expandiendo el campo de experiencia del agente y se va buscando constantemente el aprendizaje y la auto-mejora.

12. Stuart Russell & Peter Norvig, *Artificial Intelligence: A Modern Approach (Second Edition)*, Prentice Hall, 2003.

13. M. Wooldridge, *An Introduction to Multiagent Systems*, J. Wiley & Son., 2002.

Por otra parte, teniendo en cuenta que el generador de problemas se dedica justamente a sugerir acciones exploratorias, el elemento de performance es lo que antes considerábamos el agente completo, ya que es el que realiza las acciones externas y el crítico se dedica a informar como va el agente. Está claro que la clave del aprendizaje se encuentra en el elemento de aprendizaje y es por esto que más adelante se va analizar en más profundidad cómo es que los agentes pueden aprender.

4.11 - Resolución de problemas mediante la búsqueda

A lo largo de su vida, un agente se enfrenta a constantes problemas y una forma de resolverlos es a través de la búsqueda. Para entender este concepto creemos que es importante ver el funcionamiento del “agente revolvedor de problemas”, el cual se encuentra dentro del grupo de los agentes basados en objetivos.

A fin de ir ejemplificando cada concepto, vamos a plantear un problema global como ejemplo: vamos a plantearnos un agente que maneja un auto cuyo problema planteado es el de conducir desde la Ciudad de Buenos Aires hacia Tucumán.

Este agente deberá elegir el mejor camino para realizar el viaje por lo que el primer paso para poder resolver dicho problema será realizar correctamente la formulación del objetivo. Para esto, deberá inicialmente basarse en la situación actual del agente y luego, la formulación del problema, será un proceso en el que se deberá decidir qué acciones y estados serán considerados dependiendo del objetivo.

Esta formulación de problemas está compuesta formalmente por cuatro elementos:

- El estado inicial, que indica donde debe iniciarse la secuencia que lleve a la solución. En nuestro ejemplo, se trata de Buenos Aires.
- La descripción de todas las posibles acciones que pueda tomar el agente. Estas acciones claramente se encuentran ligadas al estado actual del agente. En nuestro ejemplo, una de las acciones siendo el estado actual Córdoba, puede ser Mendoza o Santiago del Estero pero no Jujuy.
- La prueba de objetivo que determina si cierto estado es el objetivo. En nuestro ejemplo, se debe chequear si un estado se trata de Tucumán.
- La función de costo de camino que asignará un costo numérico a cada trayecto. Es aquí donde aparecen las medidas de *performance* para determinar qué costos reflejan los gustos del agente y de esta forma decidir. En nuestro ejemplo podemos considerar como costo la distancia entre distintos nodos (estados) y manejarnos con una medida de *performance* que busca la menor distancia posible. De esta forma, conseguimos que el agente realice el camino más corto entre Buenos Aires y Tucumán.

función AGENTE-RESOLVEDOR-PROBLEMAS-SIMPLES (*percepción*) **retorna** una acción

entrada: *percepción*, una *percepción*

estático: *seq*, una secuencia de acciones, inicialmente vacía
estado, una descripción del estado del mundo actual
objetivo, un objetivo, inicialmente nulo
problema, la formulación del problema

estado ← ACTUALIZACION-ESTADO (*estado*, *percepción*)

si *seq* es vacío **entonces hacer**

objetivo ← FORMULAR-OBJETIVO (*estado*)

problema ← FORMULAR-PROBLEMA (*estado*, *objetivo*)

seq ← BUSCAR (*seq*)

acción ← PRIMERO(*seq*)

seq ← RESTO(*seq*)

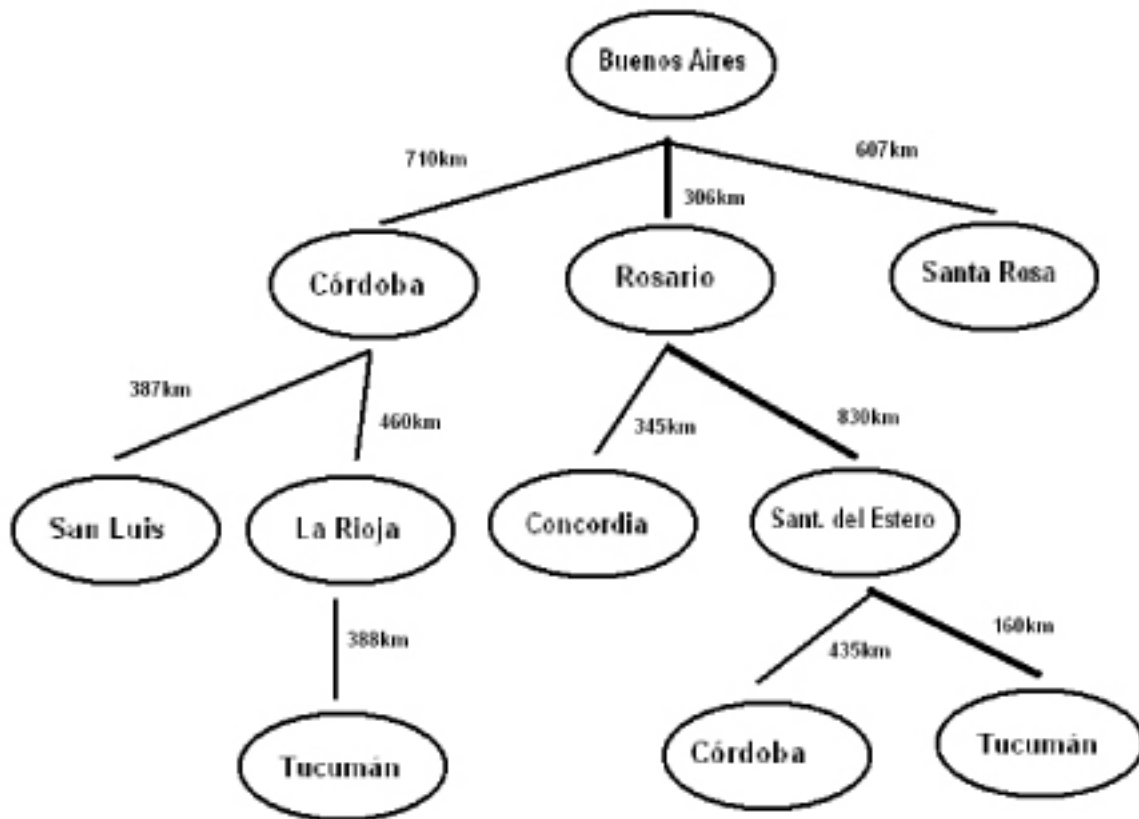
retornar *acción*

Pseudo-código de un agente revolvedor de problemas simples¹⁴

La búsqueda es el proceso que tiene como entrada el problema y como salida retorna una solución en forma de secuencia de acciones, por eso hay que evaluar las distintas alternativas que existen a la hora de realizar una búsqueda.

Volviendo a nuestro ejemplo, si el agente está buscando el camino desde Buenos Aires a Tucumán, se puede crear un grafo que represente todas las opciones con las que cuenta el agente, siendo cada nodo una ciudad y los arcos las rutas entre ellos. Cada arco tendrá como valor la distancia entre las distintas ciudades, que es a la vez el costo de camino que debe realizar el agente.

¹⁴ Stuart Russell y Peter Norvig, *Artificial Intelligence: A Modern Approach (Second Edition)*, Prentice Hall, 2003.



Grafo que representa las opciones de viaje del agente para ir desde Buenos Aires hacia Tucumán. Siguiendo la medida de performance de elegir el camino mas corto, el agente debe elegir Buenos Aires – Rosario – Santiago del Estero - Tucumán

Contando con este grafo descubriremos que un agente debería recorrer todo el grafo para descubrir cual es la ruta óptima, grafo que tomaría milésimas de segundos con las computadoras actuales, pero si pensamos que éste es un simple ejemplo de la estructura de resolución de problemas por búsqueda, sabremos que el agente puede enfrentarse con grafos realmente inmensos.

Por este motivo existen distintas estrategias que buscan evitar la pérdida de tiempo y espacio y encontrar la solución de la manera más rápida. Para calificar a estas estrategias, se utilizan las siguientes medidas de *performance*:

- **Amplitud:** El algoritmo debe garantizar que va a encontrar una solución si ésta existe.
- **Optimización:** La estrategia debe buscar la mejor solución de todas.
- **Complejidad temporal:** Cuanto tiempo demorará el algoritmo en encontrar la solución.
- **Complejidad espacial:** Cuanto espacio de memoria va a necesitar el algoritmo para realizar su búsqueda.

Algunas de las estrategias de búsqueda más utilizadas son la de “*búsqueda por primera amplitud*”, “*búsqueda por costo uniforme*”, “*búsqueda por primera profundidad*”, “*búsqueda de profundidad limitada*” y la “*búsqueda bidireccional*”.

Las investigaciones y búsquedas de distintas estrategias de búsqueda mantienen ocupados a un gran sector de matemáticos que intenta constantemente encontrar la mejor solución para realizar una búsqueda porque las diferencias entre las distintas estrategias se encuentran al plantearse cuándo expandir un nodo y cuándo no, cómo realizar evaluaciones constantes o seleccionadas, cómo ir hasta el final de un nodo y luego regresar, etc.

No vamos a profundizar describiendo el funcionamiento de cada estrategia porque no representa el objetivo de este trabajo pero sí creemos que es necesario mencionar su existencia y saber que cada una tiene sus ventajas y desventajas para elegir una de ellas a fin de crear un agente como parte del diseño general del mismo.

4.12 - Funciones heurísticas

Dentro del grupo de algoritmos de búsquedas dentro de grafos, se encuentra el de “mejor-primera búsqueda” que utiliza una función de evaluación para elegir qué rama del grafo va a ir expandiendo y así guiando su búsqueda. De esta manera, la selección de esta función determinará el éxito o no del algoritmo.

Cabe aclarar que existe toda una familia de algoritmos de “mejor-primera búsqueda”, cada uno con su función de evolución propia.

El componente clave de estos algoritmos es la función heurística, la cual se expresa $h(n)$ y en el ejemplo que venimos manejando, se utilizó como función heurística el cálculo de la distancia entre los distintos nodos.

Las funciones heurísticas son el método por el cual se puede agregar conocimiento sobre el problema al algoritmo de búsqueda. De esta manera, con la alteración y la búsqueda de una función heurística óptima se puede conseguir el mejor algoritmo de búsqueda.

En el ejemplo del viaje desde Buenos Aires a Tucumán que hemos citado anteriormente se puede utilizar como función heurística, la suma del costo de llegar a un nodo desde el nodo inicial, el costo de llegar al objetivo desde dicho nodo. Para verlo más claro, esta función heurística aplicada a Santiago del Estero, nos retornaría la suma desde Buenos Aires hasta Santiago del Estero que es de 1136km con la distancia desde Santiago del Estero a Tucumán que es el objetivo, 160km, con lo cual obtendríamos un total de 1296km.

Así, aplicando esta función heurística, encontraremos otras opciones para llegar a Tucumán desde Buenos Aires que cambiaría la forma de explorar el grafo. De esta manera se entiende que la clave para realizar la búsqueda de una solución de la mejor forma posible se encuentra en la función heurística que se utilice.

4.13 - Agentes característicos

Dentro de los distintos tipos de agentes desarrollados por distintos grupos de investigadores existen algunos que merecen ser destacados y, aunque sea, brevemente detallados. Estos son los agentes basados en el conocimiento, los basados en la lógica proposicional y los basados en circuitos.

4.13.1 - Agentes basados en el conocimiento

Este tipo de agentes cuenta con una característica que los diferencia de las otras grandes familias de agentes, la base de conocimiento (KB en inglés), y se la puede definir como un conjunto de sentencias que se encuentran expresadas en un lenguaje que se llama, justamente, “lenguaje de representación del conocimiento”.

Cabe aclarar que para poder interactuar internamente con su base de conocimiento, el agente cuenta con dos opciones. Por un lado puede agregar una sentencia utilizando la tarea TELL (contar) y por el otro cuenta con ASK (preguntar) para poder hacer una consulta a la base de conocimiento.

De esta manera, al contar con esta base de conocimiento como soporte de fondo, el agente puede tomar decisiones de una forma distinta. Por ejemplo, si ponemos un agente en un laberinto que desconoce, se puede agregar en su base de conocimiento determinadas referencias que le sirvan para encontrar la salida (por ejemplo, una pared húmeda indicaría que del otro lado hay agua, por lo que no se puede pasar). Así el agente podrá ir incorporando a su conocimiento las partes del laberinto que va conociendo.

4.13.2 - Agentes basados en la lógica proposicional

Los agentes basados en la lógica proposicional son uno de los tantos agentes que trabajan utilizando todas las reglas y notaciones de la lógica moderna. Utilizan la base de conocimiento para almacenar todas las sentencias lógicas que van aprendiendo y al igual que los agentes basados en el conocimiento, utilizan la sentencia TELL para agregar el conocimiento.

Para poder manejarse por el mundo, estos agentes utilizan la sentencia ASK ya que necesitarán realizar las consultas a la base de conocimiento, que al combinarla con las herramientas que proporciona la lógica moderna, encuentran un valor de verdadero o de falso a las distintos problemas que se van planteando.

Pero para poder representar el mundo moderno en un lenguaje que utiliza sólo sentencias de lógica, estos agentes utilizan una metodología lógica de representar al mundo, tema que se analizará un poco mas adelante.

4.13.3 - Agentes basados en circuitos

Estos agentes están dentro de la familia de los agentes de reflejo, cuyas percepciones son la entrada a los circuitos secuenciales: una red de puertas que implementan una conexión lógica, y de registros, que almacenan el valor de verdad de cada proposición simple.

De esta manera se genera un flujo de datos que recorren los circuitos realizando las operaciones lógicas, que finalmente indicarán un valor de resultado verdadero o falso lo cual llevará al agente a actuar o no. Pero para que no sea un simple circuito de acción y cumpla con los requisitos mínimos de un agente,

estos agentes tienen un diseño muy complejo, por lo que primero se los diseña de una forma lógica en su totalidad y luego se lleva esto al mundo electrónico de los circuitos.

4.14 - Representación del conocimiento

Por otra parte, para que un agente pueda moverse dentro de un mundo, primero debe entenderlo, de modo que debe contar con alguna técnica de representación del conocimiento.

La ingeniería ontológica ha permitido a los agentes contar con las herramientas y técnicas para poder representar el tiempo, las acciones, los objetos físicos y hasta las creencias a través de conceptos abstractos. Pero obviamente, realizar una representación abstracta de todo lo que forma este mundo es algo más bien utópico, por lo que se apunta a representar lo necesario para cada agente.

Entonces, la representación se basa en toda la teoría de la lógica y aplica los conceptos intentando esquematizar y vincular el conocimiento de una manera práctica y razonable.

Por ejemplo, se pueden relacionar categorías como *Perros* ϵ *EspeciesDomésticas*, o entender la composición física como *ParteDe(Buenos Aires, Argentina)*, o manejar medidas como *Diámetro(PelotadeBasket) = Pulgadas(9.5)*, o manejar el conocimiento de sustancias como $x \epsilon$ *Manteca* \wedge *ParteDe(y,x) \Rightarrow y \epsilon* *Manteca*.

De esta manera, un agente que utiliza una base de conocimiento puede almacenar toda la información que necesite e ir actuando en base a esto. Las sentencias lógicas le permitirán realizar todos los planteos necesarios y las herramientas de la lógica moderna lo llevarán a poder tomar una decisión correcta.

4.15 - La planificación

La planificación es otra metodología con la que un agente completo debe contar para poder resolver problemas. A través de la planificación un agente cuenta con una determinada secuencia de acciones que lo deberían llevar al objetivo deseado.

En muchos casos, si un agente intenta resolver un problema mediante la búsqueda puede encontrarse perdiendo muchísimo tiempo y recursos en general, ya que, por ejemplo, si a un agente le pido que me compre un libro cuyo número de ISBN es 4156215421 desde una tienda virtual y existe un algoritmo de compra por cada número ISBN sobre un total de 10 millones de acciones, el agente se encontrará revisando el estado de cada una de los 10 millones de acciones.

Por el contrario, un agente de planificación debe poder trabajar con un objetivo explícito como *Tener(ISBN4156215421)* directamente, y para realizar esto, el agente solo necesitará del conocimiento general que *Comprar(x)* lleva a *Tener(x)*, por lo que el agente entenderá que la acción correcta es la de *Comprar(ISBN4156215421)*.

Una gran ventaja que tiene el uso de la planificación es la descomposición del problema. Por ejemplo, si un agente tiene que organizar los colectivos que deben tomar cuatro chicos para ir a la escuela, le resultará mucho más sencillo descomponer el problema en sub-problemas, uno por cada chico, y así buscar las paradas más cercanas para cada destino. Si esto se intentara resolver mediante la búsqueda, el agente se encontraría realizando muchísimas más operaciones para encontrar la solución.

Por otra parte, existen dos lenguajes importantes para representar las planificaciones básicas, el Strips y el ADL (Action Description Language, en inglés). Estos lenguajes buscan que el agente pueda representar estados, objetivos y acciones.

Las diferencias entre ambos lenguajes son técnicas y ambos pueden ser utilizados para cualquier tipo de caso; así, por ejemplo, en Strips solo se pueden utilizar literales positivos como la sentencia *Lindo* \wedge *Rico*, mientras que en ADL se pueden utilizar tanto positivos como negativos, como en *Pelado* \wedge \neg *Alto*. Otra diferencia es que en Strips el objetivo es una conjunción como *Blanco* \wedge *Lustrado* mientras que en ADL el objetivo puede contener conjunciones o disyunciones.

4.16 - Funciones de la utilidad

Poder designar mediante una función un valor numérico a un posible estado en el que un agente puede encontrarse rigiéndose exclusivamente de las preferencias, es la forma por la cual un agente puede aplicar los gustos de un usuario en su funcionamiento.

Un agente muy sencillo se manejaría con una función de utilidad ordinaria que tendría a su cargo realizar una especie de ranking de estados al asignarle la utilidad que le corresponda a cada uno. Por otro lado, un agente un poco más complejo se encontrará en situaciones donde pueden existir distintas valoraciones de utilidad e incluso conflictos entre ellas, por lo que dicho agente tendrá que manejarse con la teoría de los multiatributos de utilidad.

Dicha teoría maneja una estructura de preferencias en la que se plantean los atributos y los posibles valores de cada uno, y en el peor de los casos se tendrían d^n , siendo n los atributos y d los distintos valores de cada uno.

Este último caso se presentaría cuando las preferencias del agente no tienen cierta regularidad, ya que de lo contrario se contaría con los teoremas de representación que indican cierta tendencia de preferencia en un agente y que indicará que a cierto tipo de estructura de preferencia le corresponde cierta función de utilidad.

4.17 - Actuando bajo incertidumbre

Desafortunadamente muchos agentes deben funcionar sin poder contar con el conocimiento completo del mundo que los rodea y a pesar de esto deben cumplir con sus objetivos.

Por otra parte, a un agente lógico le sería imposible poder construir una descripción correcta y completa de cada acción que debe ejecutar, ya que pensemos que, por ejemplo, si un agente debe manejar un auto para ir a buscar a una persona a lo largo de un trayecto de 15 minutos, no puede afirmar con seguridad que llegará a tiempo, sino que su planteo es que llegará a tiempo si no se le pincha la rueda, no se le funde el motor, no choca, etc.

Evidentemente, para poder afrontar estas situaciones un agente debe ser capaz de manejarse con preferencias que le permitan controlar desde un comienzo, las alternativas con las que cuenta y buscar evitar encontrarse en un callejón sin salidas, lo cual lo lleva a recurrir a la teoría de la utilidad a fin de representar y razonar con preferencias.

Por otro lado, el agente va a tener que hacer uso de la teoría de la probabilidad, la cual asigna a cada sentencia que se le presenta un grado numérico de creencia entre 0 y 1. De esta manera, la probabilidad le provee al agente una manera para sintetizar la incertidumbre que viene de la pereza (a veces es mucho trabajo listar de forma completa todos los antecedentes y consecuentes necesarios para asegurarse alguna regla excepcional) y la ignorancia del agente mismo.

En síntesis, las preferencias, expresadas por las utilidades, se combinan con las probabilidades en una teoría general de las decisiones racionales llamada teoría de la decisión (teoría de la decisión = teoría de la probabilidad + teoría de la utilidad), cuya idea principal se encuentra en un principio llamado Utilidad Máxima Esperada (MEU, según sus siglas en inglés), según la cual un agente es racional, si y sólo si, elige las acciones que producen la utilidad máxima esperada, promediando sobre todos los posibles resultados de una acción.

4.18 - Redes de decisión

Un mecanismo general para tomar decisiones racionales, es la utilización de las redes de decisión. En este caso cada red representa la información sobre un estado del agente, sus posibles acciones, el estado resultante de cada una de estas acciones y la utilidad de ese estado. Para poder ilustrarla existen tres tipos de elementos: los nodos de oportunidad, los de decisión y los de utilidad.

El algoritmo para realizar la evaluaciones sobre las redes de decisión funciona de la siguiente manera:

- 1) Fijar las variables para el estado actual.
- 2) Establecer para cada posible valor del nodo de decisión:
 - a. Fijar el nodo de decisión a ese valor.
 - b. Calcular la probabilidad posterior para cada nodo padre del nodo de utilidad, utilizando un algoritmo de inferencia probabilística convencional.
 - c. Calcular la utilidad resultante de la acción.
- 3) Retornar la utilidad resultante para esa acción.

4.19 - Formas de aprendizaje

Si revisamos lo ya enunciado, recordaremos que en un agente que cuenta con la capacidad de aprendizaje existe, entre otras cosas, un elemento de *performance* que decide qué acciones ejecuta el agente; y también un elemento de aprendizaje que modifica al elemento de *performance* para que tome mejores decisiones.

Pero el diseño de este elemento de aprendizaje se encuentra afectado por las elecciones de qué componentes del elemento de *performance* serán los que aprenderán, qué retroalimentación se encuentra disponible para aprender y qué representación se utilizará para estos componentes.

Al referirnos a los componentes, se habla del mapeo directo entre las condiciones del estado actual y las acciones, la manera de inferir las propiedades relevantes del mundo desde la secuencia de percepción, la información sobre la manera en que el mundo evoluciona y los posibles resultados que se pueden obtener de acuerdo a las acciones posibles que el agente puede tomar.

Pero también se deben considerar la información de utilidad que indica el atractivo de los estados globales, la información acerca del valor de cada acción (que indique el atractivo de las acciones) y el

objetivo que describe la clase de estados cuyo logro maximiza la utilidad del agente.

El tipo de retroalimentación se encuentra directamente vinculado a la naturaleza del problema de aprendizaje al que el agente se enfrenta y estos problemas pueden ser catalogados en tres grandes grupos:

- 1) Problemas de aprendizaje supervisado, que involucra el aprendizaje de funciones mediante ejemplos de sus salidas y sus entradas.
- 2) Problemas de aprendizaje no supervisado, que involucran patrones de aprendizaje en la entrada cuando no se cuenta con un valor específico de salida. Por ejemplo, un agente que maneja un auto tendrá que desarrollar gradualmente los conceptos de “días de buen tráfico” o “días de mal tráfico”, sin haber recibido ejemplos marcados de cada uno.
- 3) Problemas de aprendizaje reforzado que es el más general de los tres, y es el que explica cómo un agente puede aprender sin contar con alguien o algo que lo guíe.

Esta recompensa o refuerzo viene siendo estudiada en los animales y humanos hace más de 60 años, porque es un proceso muy complejo, ya que si un agente no cuenta con algún retorno, que le indique si lo que hace está bien o mal, ¿cómo puede aprender? Por ejemplo, a primera vista, un agente que juega al ajedrez, solo al final recibiría su recompensa, es decir si es que ganó, por lo que durante el juego no tiene ningún indicio que le diga si va bien o mal.

Este tipo de aprendizaje puede ser encontrado de dos maneras, en su forma pasiva donde la política del agente es fijada y la tarea es aprender con los estados de utilidad, lo cual puede involucrar aprender un modelo del ambiente, mientras que la forma activa se refiere a la situación en la cual el agente además, debe aprender qué hacer.

Por último, hay que destacar que el aprendizaje del agente debe ser “inteligente”, o sea, deberá apoyar sus aprendizajes en el conocimiento que ya cuenta y en su diseño. Para entender esto más fácilmente veamos un ejemplo:

Supongamos que una persona no conoce nada de Francia y llega a ese país, por primera vez, habla con la primera persona que ve y ésta le contesta en francés, así aprenderá que en Francia se habla francés. Pero si esta persona le dice que se llama Jean-Claude, el agente no asume que todos en Francia se llaman así porque esta distinción se apoya en su información previa; en lingüística esto se llama competencia comunicativa que le permite hacer esta diferencia.

4.19.1 - Aprendizaje inductivo

Un algoritmo para el aprendizaje determinístico supervisado es dado por el valor de entrada correcto de una función desconocida para particulares entradas y se debe tratar de obtener la función desconocida o algo cercano. Por ejemplo: siendo el par $(x, f(x))$, donde x es la entrada y $f(x)$ la salida de la función aplicándole la x y teniendo una colección de f ; si quiero obtener una función h que se aproxime a f , esta función h es llamada hipótesis.

Determinar cuál h es particularmente una buena aproximación a f , es la parte difícil. Una buena hipótesis va a generalizar correctamente hasta ejemplos nunca vistos.

4.19.2 - Aprendizaje mediante árbol de decisión

Un árbol de decisión toma como entrada un objeto o situación descrita por un grupo de atributos y retorna una decisión. Para llegar a esto, el árbol va realizando de forma secuencial distintos tests que lo van llevando a la decisión y cada nodo del árbol corresponde a un posible valor del test. La estructura que se utiliza para su armado está basada en el pensamiento humano.

Lo interesante es que la combinación de este proceso con los métodos de aprendizaje le permite al agente aprender que ciertas hipótesis generan cierto resultado y sobre un grupo de atributos pueden crearse cientos de árboles de decisión y clasificarlos para obtener el más indicado.

4.20 - Comunicación

La mayoría de los agentes tienen la necesidad de comunicarse con otros agentes o con los usuarios. En el caso de la comunicación con otros agentes estaríamos hablando de un sistema multiagente, tema que será visto en profundidad más adelante, en el capítulo “Sistemas Multiagente”.

Consideramos que el agente debe ser capaz de poder comunicarse con el usuario intentando simular, en la forma más cercana posible, el lenguaje de un humano y por otra parte puede utilizar su capacidad de aprendizaje para ir amoldando sus comunicaciones a los gustos del usuario.

Un lenguaje es definido como un conjunto finito o infinito de cadenas compuestas a su vez una concatenación de símbolos terminales admitidos por el lenguaje.

Pero a diferencia de un lenguaje lógico donde los símbolos terminales incluyen ejemplos como P o \wedge , los lenguajes como el español, el inglés o cualquier otro se encuentran dentro del grupo de lenguajes

naturales y no cuentan con una definición estricta aunque no por esto no pueden ser analizados y utilizados por un agente.

Así, una comunicación típica donde A quiere informarle a B sobre una propuesta P utilizando las palabras L se encontrará compuesta por siete procesos:

- **Intención:** De alguna manera A tiene la intención de comunicarle P a B.
- **Generación:** A planea cómo convertir la idea de P en las palabras L de manera que le sea mas amigable de escuchar a B.
- **Síntesis:** A produce de una manera física una realización de L, ya sea en un papel, vibraciones en el aire, etc.
- **Percepción:** B recibe la realización física hecha por A y decodifica esto en L. Si se trata por ejemplo de una comunicación por voz, este proceso de decodificación se llama reconocimiento de voz.
- **Análisis:** B deduce que L puede tener varios posibles significados. Esta parte del análisis se divide a su vez en tres: el análisis sintáctico, la interpretación semántica y la interpretación pragmática. El primero se encarga de realizar un árbol de análisis para la cadena de entrada donde los nodos interiores representan frases y las hojas palabras. La interpretación semántica extrae el significado de la declaración como una expresión en alguna representación del lenguaje. Por último, la interpretación pragmática tiene en cuenta que una misma palabra puede tener distintos significados en diferentes situaciones por lo que se encuentra asociada al contexto y mediante esto buscar el significado adecuado de L.
- **Desambigüedad:** B entiende que A intenta transmitir P (idealmente idéntico al P que pensó A). Hay que tener en cuenta que existe la ambigüedad intencional en una comunicación por lo que el agente, al igual que un humano contará con varias interpretaciones, pero él deberá elegir una.
- **Incorporación:** B decide creerle o no a A. Un agente sofisticado puede distinguir esto y entender al P interpretado como una evidencia y no como una confirmación de su significado.

Al parecer, el análisis es el paso más complejo, de modo que cabe considerar un poco de cada una de sus tres etapas.

4.21 - Análisis sintáctico

Como dijimos brevemente, en esta etapa se genera el árbol correspondiente para una cadena de entrada. Para esto, la función de análisis necesitará de la frase y de la gramática correspondiente y retornará un árbol llamado en ingles "parse tree".

El proceso de generación del árbol puede ser mediante una técnica "bottom-up" o al revés. En ambos casos se define como un problema solucionable mediante la búsqueda, por lo que se contará como elementos un estado inicial, una función de sucesión que irá seleccionando como se armará el árbol y por ultimo una prueba de objetivo que determinará cuándo el árbol generado corresponde exactamente a la cadena de entrada.

Para hacer más eficiente este proceso, dentro del agente se cuenta con la idea de una programación dinámica donde una vez que una sub-cadena sea analizada, los resultados de dicho análisis sean guardados para evitar tener que repetir el proceso una y otra vez.

4.22 - Interpretación semántica

En esta etapa se busca la extracción del significado de la expresión. Así se realiza un recorrido a través del árbol, vinculando las palabras para generar el significado. Por ejemplo, si tenemos la frase "María cocina lasaña", se conseguirá una interpretación lógica que indique Cocinar (Maria, lasaña).

Durante este proceso se deberá analizar las palabras que indiquen a un sujeto oculto para no caer en un grave error de interpretación, además de saber manejarse con indicaciones temporales y cuantificadores. Una frase como "Todos los martes la lasaña es cocinada" implicará todos estos conceptos.

4.23 - Interpretación pragmática

Agregarle a la interpretación información del contexto para determinar cuál es la interpretación correcta, es la función principal de esta etapa, lo cual implica identificar qué frases se refieren a la situación actual, frases que en inglés son conocidas como "indexical".

Al respecto, daremos un simple ejemplo: si se estuviera analizando la frase "y después me pude bañar tranquilo", y se cuenta con la frase "estuve todo el día limpiando la casa", el agente podrá contar con ésta como frase "indexical" para obtener el significado correcto de la analizada.

4.24 - Procesamiento probabilístico del lenguaje

Es posible entonces, que un agente entienda la comunicación humana pero al referirnos a textos

mucho más complejos como un libro o tal vez muchas páginas webs, se vuelve necesario considerar un acercamiento conocido como “basado en el cuerpo”.

Este acercamiento implica el uso de las estadísticas y el aprendizaje para tomar ventajas del cuerpo. Así, se hace uso de modelos de lenguaje probabilístico que cuentan con otra capacidad de aprendizaje más simple que las ya señaladas, y que aprenden en base a los datos del cuerpo.

En efecto, el modelo de lenguaje probabilístico define una distribución probabilística sobre todo un conjunto de cadenas, aunque éste sea infinito, y estos diagramas deben ser capaces de representar de una forma clara y precisa los valores que se le asignaron a cada palabra.

Para este acercamiento existen tres tareas importantes: la recuperación de información, la extracción de información y la traducción de máquina.

4.24.1 - Recuperación de información

Es la tarea de buscar los documentos que le son relevantes a las necesidades de información del usuario. El sistema más conocido de búsqueda de información es el de los buscadores de internet, los cuales mediante el uso de palabras claves realizan poderosas búsquedas en sus grandes bases de datos y devuelven los resultados ordenados según valores. Es muy importante poder diferenciar los resultados relevantes de los que no lo son por lo que se busca el valor máximo en las medidas de precisión de las búsquedas.

Para este proceso se pueden identificar cuatro partes:

- **La colección de documentos** donde cada sistema decide qué va a considerar como documento: un párrafo, una página o un texto de muchas páginas.
- **La presentación de la consulta en un lenguaje de consultas** donde se especifique qué se quiere saber.
- **Un conjunto de resultado** que contenga los documentos que fueron considerados relevantes para la consulta.
- **La presentación de resultado** que contenga la lista organizada con los títulos u otro tipo de presentación.

4.24.2 - Extracción de información

Este proceso se refiere a la creación de las entradas en la base de datos examinando un texto y buscando las ocurrencias de una clase particular de objeto o evento y las relaciones entre estos objetos y acciones.

El método más simple para la extracción de información es el sistema basado en el atributo, el cual asume que el texto completo corresponde a un solo objeto por lo que la tarea es extraer atributos de ese objeto.

Extracciones más complejas pueden implicar el uso de transductores en cascada de estados finitos, los cuales consisten en una serie de autómatas finitos que reciben un texto como entrada, translucen el texto a diferentes formatos y lo pasan al siguiente autómata. Se busca en estos sistemas la complementación de todos los autómatas para globalmente extraer la información necesaria.

4.24.3 - Traducción de máquina

Se llama de esta manera al proceso que realiza una traducción automática desde un lenguaje natural hacia uno determinado. Este proceso es útil para varias tareas, incluyendo las siguientes:

- **Traducción bruta**, donde el objetivo es simplemente obtener la esencia de una entrada. En esta traducción se toleran sentencias no elegantes ni gramaticalmente correctas.
- **Traducción de fuente restringida**, en la cual la materia del asunto y el formato de la fuente se encuentran severamente restringidas.
- **Traducción pre-editada**, en la cual el usuario edita previamente la fuente del documento para hacerla conforme a un subconjunto del idioma antes que la máquina haga la traducción.
- **Traducción literal**, en la cual las matrices de la fuente son preservadas.

5 - Sistemas multiagentes

En este punto cabe que nos preguntemos, ¿por qué distribuir la inteligencia artificial? Evidentemente, una respuesta es que así se pueden solucionar problemas que se le plantean a los usuarios en un mundo real, es decir que hay que plantear una estrategia de solución acorde al mundo real.

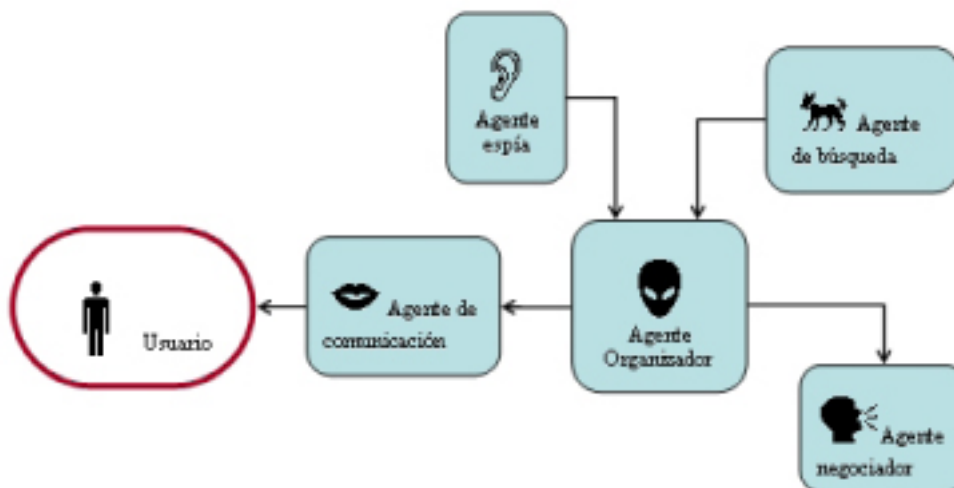
Pero estos problemas están físicamente distribuidos, además el mundo en sí esta compuesto por entidades autónomas que interactúan entre si y con el entorno.

Por otra parte, teniendo en cuenta que los requisitos de los usuarios o del mundo en general siempre son más complejos que una simple acción, es muy raro encontrar a un agente sólo funcionando. Por este motivo, la mayoría de los sistemas inteligentes basados en agentes cuentan con más de uno de éstos funcionando en su interior, por eso, estos sistemas son llamados sistemas multiagentes.

Además, la necesidad de desarrollar aplicaciones complejas compuestas de multitud de sub-sistemas que interactuonen entre si, obliga a distribuir la inteligencia entre diversos agentes y construir sistemas multiagente.

Un sistema multiagente, entonces, permite la gestión inteligente de un sistema complejo, coordinando los distintos subsistemas que lo componen e integrando los objetivos particulares de cada subsistema en un sistema común; y si se dota a cada subsistema de una capacidad de decisión local, el problema de gestión se puede abordar definiendo políticas de cooperación, coordinación y negociación entre agentes.

Lo cierto es que la colaboración entre los componentes es lo que permite la resolución del problema y la consecución de los objetivos pre-definidos, así, y a modo de ejemplo, imaginemos que necesitamos un sistema inteligente que sirva de nexo entre el perro y su amo. Este sistema necesitará de varios agentes para su funcionamiento tal como se advierte en la siguiente figura:



Esquema

En un sistema multiagente, los agentes deben compartir conocimientos sobre el problema y sus posibles soluciones. Además, pueden compartir una meta o no, pero en este último caso no debe haber conflicto.

La idea es que no exista un control global, que los datos puedan encontrarse descentralizados, que se mantenga una computación asincrónica porque así se proveen de más robustez, eficiencia y permiten la interoperabilidad de sistemas existentes.

De modo que para que un conjunto de agentes pueda desarrollar una actividad conjunta en un entorno compartido, deberá existir algún tipo de coordinación. Esto quiere decir que si se trata de un sistema con agentes cooperativos se requiere una planificación entre ellos que permita el intercambio correcto.

Obviamente, en este tipo de interacción los agentes aportan sus habilidades para conseguir el objetivo en común, en muchos casos distribuyéndose las tareas y aplicando especializaciones a los agentes.

En un sistema con agentes competitivos existirá una negociación entre los agentes ya que cada uno buscará cumplir sus propias metas y terminarán compitiendo por los recursos. En este último caso, de darse, haría falta algún sistema de coordinación.

Un tercer caso puede ser el de un sistema con independencia completa o parcial entre los agentes donde los agentes actúan como si estuvieran solos y solamente realizando una visión global del sistema se podrá encontrar el objetivo general de todo el sistema multiagente.

15. Javier de Andrés de la Universidad de Oviedo, Enrique Bonsón de la Universidad de Huelva, Tomas Escobar de la Universidad de Huelva y Carlos Serrano de la Universidad de Zaragoza, "Inteligencia Artificial y Contabilidad", *Publicaciones AECA*, España, 2006.

5.1 - Definición

Tal como ocurrió con los agentes, se han propuesto varias definiciones para el concepto de **sistema multiagente**. Una muy conocida es la enunciada por Durfee según la cual "Un SMA se puede definir como una red débilmente acoplada de resolvedores de problemas que interactúan para resolver problemas que están más allá de las capacidades individuales o de los conocimientos de cada uno de los resolvedores individuales de problemas"¹⁶

Más recientemente, se ha dado al término sistema multiagente un significado más general, y actualmente es usado para definir todos los tipos de sistemas compuestos por múltiples componentes autónomos que poseen las siguientes características:

- Cada agente tiene capacidad para solucionar parcialmente el problema.
- No hay un sistema global de control.
- Los datos no están centralizados.
- La computación es asíncrona.

Uno de los factores actuales que promueve la investigación es la creciente popularidad de Internet que proporciona la base para un entorno o ambiente abierto donde los agentes interactúan para conseguir sus objetivos individuales o colectivos.

Por otra parte, se pueden encontrar dos grandes enfoques en lo que se refiere a la construcción de sistemas multiagentes:

- **El enfoque formal** que consiste en dotar a los agentes de la mayor inteligencia posible, utilizando descripciones formales del problema a resolver y de hacer reposar el funcionamiento del sistema en tales capacidades cognitivas.
- **El enfoque constructivista** que persigue la idea de brindarle inteligencia al conjunto de todos los agentes, para que a través de mecanismos ingeniosamente elaborados de interacción, el sistema mismo genere un comportamiento inteligente que no estaba planeado necesariamente desde un principio o definido dentro de los mismos agentes.

5.2 - Ventajas

Indicaremos a continuación qué ventajas tienen los sistemas multiagentes frente a los agentes individuales o aproximaciones centralizadas:

- Se modelan los problemas en términos de componentes autónomos que interactúan entre sí. Ésta es una forma natural de representar la distribución de tareas, planificación de equipos, etc.
- No existe un único punto de error. Los SMA distribuyen los recursos y las capacidades de ejecución de las tareas a lo largo de una red de agentes interconectados.
- Se recupera, filtra y coordina globalmente información proveniente de diversas fuentes distribuidas.
- Se permite la interconexión e interoperación de múltiples sistemas existentes.
- Se mejora el rendimiento global del sistema.

5.3 - Arquitectura

En cuanto a su diseño, los agentes presentan un desafío muy novedoso para la ingeniería del software, ya que son descritos como agentes resolvedores de problemas individuales persiguiendo objetivos de más alto nivel.

Pero, a pesar de que esta abstracción parece prometedora todavía no ha sido adoptada masivamente por los diseñadores de sistemas. Una razón es que el desarrollo de los sistemas multiagentes es técnicamente difícil. Los esfuerzos no sólo tienen que superar el desafío que representan los problemas de la programación distribuida, sino que también tienen que superar las complejidades asociadas al soporte de la colaboración entre agentes.

En efecto, si el paradigma orientado al agente tiene éxito, se necesitarán metodologías sistemáticas para especificar y estructurar aplicaciones como sistemas multiagentes.

De modo que la organización es fundamental en las sociedades de agentes ya que representarán la base de la producción común. Por este motivo siempre hace falta definir en el diseño el número y el tipo de agentes además del rol que va a jugar cada uno y en este último punto se pueden definir a priori o dejar que los agentes se especialicen. Por último, en el diseño también hace falta definir las relaciones, sean estáticas o dinámicas, existentes entre los agentes.

5.4 - Relaciones entre agentes

Como las estructuras y modelos de sistemas multiagentes son algo relativamente novedoso, a menudo se observan distintos tipos de relaciones entre los agentes lo cual nos lleva a destacar siete características:

16. Durfee, E. H., Lesser, V. R. & Corkill, "Cooperative distributed problem solving", *The Handbook of Artificial Intelligence*, Vol. 4, pp. 83 – 148, 1989.

- **Conocimiento:** cuando un agente A conoce a un agente B, tiene una representación de B y podría incluso comunicarse directamente con él. Además cuenta con soporte al resto de las relaciones.
- **Comunicación:** cuando existe un canal de comunicación entre dos agentes A y B.
- **Subordinación:** cuando un agente B realiza una tarea que un agente A le pide, que puede ocurrir de dos maneras: estática, donde B no puede negarse (relación maestro/esclavo) o dinámica, que es cuando B puede negarse (relación cliente/servidor).
- **Operativa:** cuando para realizar una tarea, un agente A depende de otra que un agente B debe llevar a cabo.
- **De Información:** cuando un agente A cree lo que ha dicho un agente B y partes de las creencias de A dependen de las de B.
- **Conflicto:** cuando varios agentes necesitan los mismos recursos.
- **Competitiva:** cuando varios agentes compiten entre sí por la consecución de sus objetivos.

A la hora de hablar de esta relación entre los agentes, dentro de un sistema multiagente, desde el punto de vista de la solución distribuida de los problemas debemos referirnos a dos puntos muy importantes, la coherencia y la coordinación.

En cuanto a la coherencia, nos referimos de una manera informal como el “que tan bien el sistema multiagente se comporta como una unidad, a lo largo de algunas medidas de evaluación”¹⁷.

Para poder medir este nivel de coherencia dentro de la comunicación entre los agentes se debe controlar la calidad de la solución, la eficiencia en el uso de los recursos, la claridad en la operación o incluso medir qué tan bien responde el sistema ante una falla.

Por el lado de la coordinación, nos referimos al “grado con el que cuenta el sistema para evitar actividades ‘extrañas’ como la sincronización y la alineación de sus actividades”.¹⁸

En un sistema perfectamente coordinado, los agentes no deberían chocar entre ellos en las búsquedas de sus objetivos propios en procura de la solución global. Incluso en este sistema ideal, no deberían requerir de la comunicación explícita para evitar estos ‘golpes’ sino que deberían poder predecir las actitudes de los otros agentes.

5.4.1 – Coordinación entre agentes

Buscando un poco más de profundidad en lo que se refiere a la coordinación entre los agentes, cabe mencionar ciertas técnicas que se utilizan en relación a esto.

Algunas de éstas son:

- **Coordinación a través de una planificación global parcial**

El principal punto de este tipo de coordinación es que los agentes intercambian información con el fin de buscar conclusiones comunes en lo que se refiere el proceso de solución de un problema.

El nombre del método de coordinación define una planificación parcial, porque el sistema no genera un plan sobre el problema entero y lo global se debe a que los agentes generan planes no-locales por medio del intercambio de la información de los planes locales.

- **Coordinación a través de intenciones conjuntas**

Este tipo de coordinación se acerca más al perfil que encuentra a los sistemas multiagentes como un modelo de trabajo en grupo de humanos.

En relación a esto, esta técnica busca que se comparta las intenciones de los agentes con el fin de proveer la estabilidad y la pronosticabilidad necesarias para la interacción social y la flexibilidad y reactividad que es imprescindible para hacer frente a los cambios dentro del ambiente.

- **Coordinación por modelado mutuo**

La idea de esta coordinación es que en una relación entre dos agentes, cada agente realice un modelado del otro, incluyendo creencias, intenciones, etc., y de acuerdo a esto se busque la coordinación entre ellos.

Para entenderlo de una manera más sencilla podemos imaginarnos que dos agentes necesitan acceder a un recurso exclusivo. Ante esto la primera solución que se puede encontrar es que ambos agentes se detengan para no “chocar” en el intento de acceder a este recurso, pero en este caso estarían desperdiciando el uso del recurso ambos agentes.

Ante este problema puede aparecer este tipo de coordinación, en la cual cada agente realizará un modelado del otro y así uno le dejaría acceder al recurso al otro porque tiene prioridades dentro del sistema o su objetivo es más importante, por ejemplo.

- **Coordinación por normas y leyes sociales**

En esta coordinación, al igual que en el quehacer diario de los humanos, los agentes se regirán por normas y leyes sociales para establecer la coordinación entre ellos.

17. Morgan Kaufmann, *Readings in Distributed Artificial Intelligence*, Bond, A. H. y Gasser L., 1988.

18. Morgan Kaufmann, *Readings in Distributed Artificial Intelligence*, Bond, A. H. y Gasser L., 1988.

Todas estas convenciones establecidas dentro del desarrollo del sistema multiagente son las que utilizarán los agentes para poder establecer sus coordinaciones. Estas reglas representan un balance entre las libertades individuales de los agentes y el objetivo global del sistema multiagente.

5.5 - Sistemas multiagentes centralizados vs. sistemas multiagentes descentralizados

En la mayoría de los casos las arquitecturas de sistemas multiagentes han preferido un enfoque centralizado, lo cual implica que el grupo de agentes colabora en forma tal que uno de ellos centraliza el control para asegurar los objetivos del grupo, pero una de las consecuencias inmediatas es que los otros agentes deben aceptar el control del agente central e, incondicionalmente, seguir sus indicaciones.

Por otra parte, uno de los problemas fundamentales en el área de los sistemas multiagentes es el de crear agentes de software con alto grado de autonomía y capaces de interactuar en ambientes distribuidos bajo un esquema altamente descentralizado. Precisamente en el Centro de Inteligencia Artificial del ITESM, Campus Monterrey, Méjico, se ha estado investigando este punto enfocándolo a la programación grupal de actividades.

Así, para abordar el tema de los sistemas multiagentes consideran que es fundamental que los agentes sean capaces de inferir y mantener, a través del tiempo, modelos cognitivos acerca de los otros agentes. Para entender mejor el término de modelos cognitivos, cabe explicar que se refiere a representaciones de agentes como sistemas intencionales; es decir, representaciones computacionales en términos de conceptos cognitivos tales como intenciones, actitudes, deseos y habilidades.

Por otra parte, estos modelos pueden funcionar como herramientas abstractas que proporcionan a cada agente formas convenientes y familiares para describir, explicar y predecir el comportamiento de los otros agentes. Al utilizarlos, cada agente debe ser capaz de solucionar tareas propias y colectivas, interactuando, negociando y adaptándose al grupo de la mejor manera posible, con lo que se podrían programar actividades colectivamente.

Además, durante este proceso de programación, todos los participantes deben tener en cuenta el objetivo grupal y sus propios objetivos individuales, entre ellos, maximizar la satisfacción de sus preferencias.

5.6 - DPS vs. MAS

Es muy importante hablar de las diferencias que hay entre la resolución de problemas distribuidos (DPS, en inglés) y los sistemas multiagentes (MAS, en inglés).

El concepto de DPS es el de un grupo de agentes inteligentes (expertos, resolvedores de problemas) que comparten sus recursos y coordinan sus actividades para resolver un problema.

Algunas de sus características son:

- Las tareas que cada agente ha de realizar están prefijadas.
- Hay un plan centralizado de resolución del problema, y se designa un miembro que ejerce un control global.
 - Los Agentes son de distintos tipos:
 - Benevolentes
 - Objetivos Comunes
 - Homogéneos

Aludir al MAS, como se ha mencionado anteriormente, implica referirnos a agentes autónomos que trabajan juntos para resolver problemas, ya que tienen las siguientes características:

- Los agentes deciden de forma dinámica qué tareas deben llevar a cabo y quién las ejecuta.
- No hay un sistema de control global.
- La Computación es asíncrona.
- Los datos son descentralizados.
 - Los agentes son:
 - No benevolentes
 - Tienen Objetivos Múltiples
 - Son Heterogéneos
 - Son Autónomos

5.7 - Modelos de sistemas multiagentes

Cabe agregar que algunos grupos de investigación e industriales independientes empezaron a perseguir la estandarización de la tecnología multiagente; así se advierten esfuerzos importantes, en el Object Manager Group (OMG), la Foundation for Physical Agents (FIPA), el grupo Knowledge-able Agent-oriented System (KAoS), y el grupo General Magic.

A continuación describiremos un poco más a éstos ya que son algunos de los referentes que se manejan dentro del ámbito de investigación y desarrollo de los sistemas multiagentes.

5.7.1 - El Modelo OMG

El grupo OMG propone un modelo de referencia como pauta para el desarrollo de las tecnologías de agentes, que esboza las características de un entorno de agentes compuesto por agentes y agencias, como entidades que colaboran usando patrones generales y políticas de interacción.

Bajo este modelo, los agentes se caracterizan por sus capacidades, el tipo de interacción, y la movilidad, mientras que las agencias soportan la ejecución concurrente de los agentes y su movilidad entre otras cosas.

5.7.2 - El Modelo FIPA

La Foundation for Intelligent Physical Agents (FIPA) es un grupo multi-disciplinario que persigue la estandarización de las tecnologías de agentes.

Esta organización ha puesto a disposición de todos los desarrolladores e investigadores del tema, una serie de especificaciones para dirigir el desarrollo de sistemas multiagentes, y particularmente importantes son las especificaciones de su Agent Management y su Agent Communication Language.

La aproximación de FIPA al desarrollo de sistemas multiagentes está basada en “un mínimo marco para la gestión de los agentes en un entorno abierto”, como dicen en su sitio web. Este marco de trabajo es descrito usando un modelo de referencia que especifica la normativa del entorno donde los agentes existen y operan, además de una plataforma de agentes que especifica una infraestructura para su despliegue e interacción.

5.7.3 - El Modelo KAoS

Este sistema es descrito por la gente de KAoS como una arquitectura distribuida abierta para los agentes software.

La arquitectura de KAoS describe las implementaciones de los agentes (empezando por la noción de un agente simple y genérico, hasta los agentes con un papel de mediadores y buscadores de pareja) y describe ampliamente las interacciones dinámicas entre agentes a través de la comunicación por mensajes, usando políticas de conversación.

5.7.4 - El Modelo General Magic

Esta tecnología modela un sistema multiagente como un mercado electrónico donde los consumidores y los proveedores de bienes y servicios pueden encontrarse y realizar transacciones comerciales.

Así, este mercado se modela como una red de computadores que soporta una colección de sitios que ofrecen servicios a los agentes móviles, entidades que residen en un lugar particular en un momento concreto, y que tienen las siguientes características:

- Pueden viajar de un lugar a otro.
- Pueden encontrarse con otros agentes, lo que les permite invocar uno a otro sus respectivos procedimientos.
- Pueden crear conexiones para permitir a un agente comunicarse con otro en un lugar diferente.
- Tienen autoridad para representar individuos y organizaciones del mundo real.
- Tienen permisos para indicar las capacidades de los agentes.

5.8 - Diseño de una arquitectura de sistemas multiagente

Los departamentos de Informática y de Ingeniería Mecánica de la Universidad de Calgary en Canadá, han creado un grupo llamado “*Collaborative Agents Group*” que ha adoptado algunas ideas de los modelos enunciados anteriormente para definir una arquitectura para sistemas multiagentes dinámicos y colaboradores. Esta arquitectura será probada revisando arquitecturas de fabricación multi-agentes del más alto nivel en un entorno de agentes colaboradores.

Así, la arquitectura modela un entorno abierto compuesto por áreas de distribución lógicas, donde los agentes existen. Estos agentes son descritos como:

- **Agentes mínimos:** Son el común denominador abstracto de todos los agentes de la arquitectura. Implementan la funcionalidad de la comunicación básica, necesaria en una interacción entre agentes.
- **Coordinadores locales de área:** Hay un coordinador local por área. Los coordinadores locales de área coordinan y representan a los agentes en su área, y les ayudan a iniciar interacciones entre ellos. Además también proporcionan un servicio de directorio de “páginas blancas” para los agentes del área.

- **Servidores de páginas amarillas:** Son responsables de guardar y tener disponible información sobre los servicios ofrecidos por los agentes.
- **Servidores de dominio cooperativos:** Proporcionan un entorno virtual que soporta la comunicación y el intercambio de información multi-agente. Estos entornos, llamados dominios cooperativos, permiten a los agentes suscribir, intercambiar mensajes, y acceder a información compartida.

En resumen, bajo esta idea, un sistema multiagente es un entorno compuesto por áreas, y cada área tiene que tener un coordinador local que ayuda a los demás agentes del área.

Los agentes pueden identificarse como presentes en el área si se han registrado con el coordinador local del área. Así usarán los servicios de los coordinadores locales del área para acceder a otros agentes del sistema.

También pueden notificar sus servicios y descubrir cuáles son los servicios que ofrecen otros agentes a través de los servidores de páginas amarillas mientras que aquellos que necesiten compartir datos con otros agentes, pueden juntarse en entornos virtuales llamados dominios cooperativos, soportados por los agentes servidores de dominios cooperativos.

5.9 - Interacción entre agentes

La capacidad de interacción es una de las más importantes de los agentes. Dicho de otra forma, los agentes interaccionan de forma recurrente para compartir información y realizar las tareas para conseguir sus objetivos. Los investigadores en lenguajes de comunicación entre agentes mencionan tres elementos clave para conseguir la interacción multi-agente:

- Un lenguaje y un protocolo de comunicación común.
- Un formato común del contenido de la comunicación.
- Una ontología compartida.

5.10 - Comunicación

La comunicación, coordinación y cooperación son puntos claves en los sistemas multiagente, ya que los estándares son indispensables para un escenario de inteligencia artificial, ya que deben soportar: la comunicación entre agentes, la sintaxis del mensaje, las ontologías (los agentes deben compartir una representación formal del conocimiento del contexto) y la interoperabilidad e interacción de distintos dispositivos en un entorno heterogéneo.

Así, esta comunicación es la base de la colaboración y la coordinación y sin ella el agente se encuentra aislado. Los agentes se comunican para:

- Solicitar la realización de tareas que no son capaces de llevar a cabo.
- Solicitar/proporcionar información.
- Compartir las creencias (modelos de mundo, de los agentes, etc.).
- Coordinarse para realizar alguna tarea compleja.

Por otra parte, al hablar de un lenguaje de comunicación entre agentes debemos mencionar las dos principales aproximaciones que existen al tema. La primera aproximación es procedural, cuando la comunicación se basa en un contenido ejecutable, un hecho que se hace posible con lenguajes como Java o Tcl. La segunda aproximación es declarativa, ya que la comunicación está basada en sentencias declarativas, como definiciones, asunciones, etc.

Debido a las limitaciones de las aproximaciones procedimentales como el hecho que el contenido ejecutable es difícil de controlar, coordinar y mezclar, los lenguajes declarativos son los preferidos para el diseño de lenguajes de comunicación entre agentes.

Cabe agregar que la mayor parte de las implementaciones de lenguajes declarativos están basadas en actos ilocutivos, como peticiones u órdenes, acciones conocidas como performativas. Uno de los lenguajes declarativos más populares es KQML que describiremos a continuación.

5.11 - ARPA - KQML

Knowledge Query and Manipulation Language fue concebido por ARPA tanto como un formato de mensajes, como un protocolo de manejo de mensajes para el soporte de la comunicación en tiempo real a fin de lograr el intercambio de conocimiento entre agentes.

KQML abarca dos grandes partes:

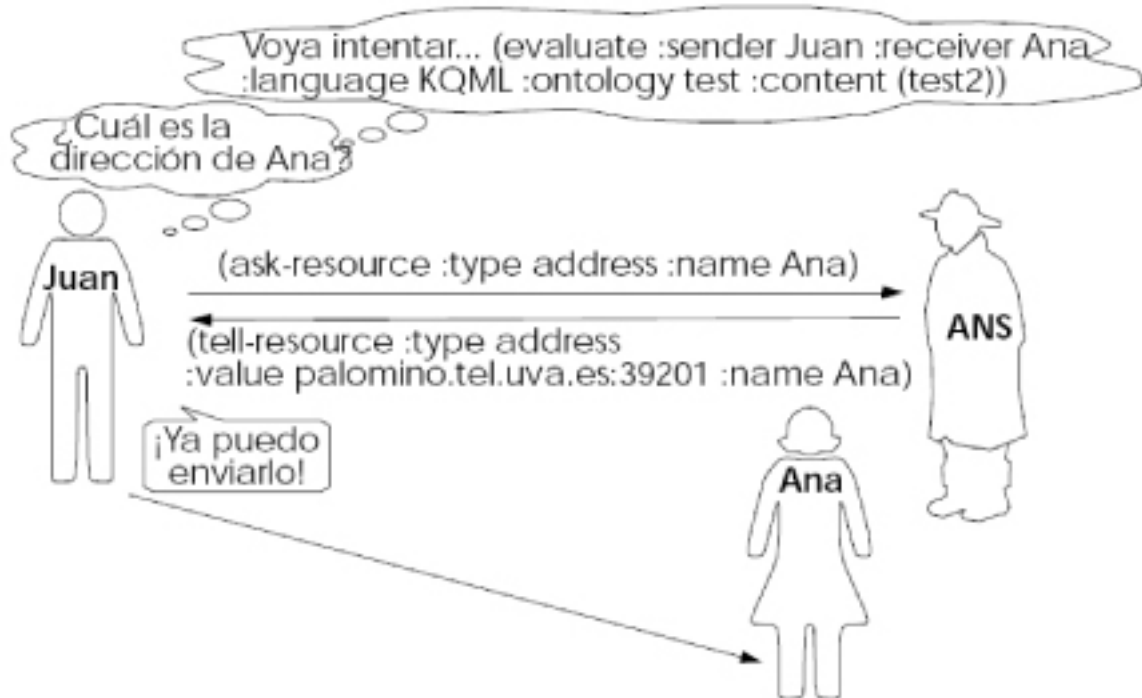
- **KQML**, cuya traducción al español sería “lenguaje de manipulación y consulta de conocimiento”.
- **KIF**, cuya traducción al español sería “formato de intercambio de conocimiento”.

5.11.1 - KQML

KQML es un lenguaje “externo” que define varios “verbos de comunicación” aceptables para poder realizar acciones performativas.

Algunos ejemplos de estas acciones son::

- ask-if ('es cierto que. . .')
- perform ('por favor, realiza la siguiente acción. . .')
- tell ('es verdad que. . .')
- reply ('la respuesta es. . .')



Petición de un favor a otro agente en KQML¹⁹

Este lenguaje KQML puede verse dividido en tres niveles:

- **Nivel de comunicaciones**, que describe los parámetros de la comunicación a un nivel básico, como el emisor, el receptor, y los identificadores de la comunicación.
- **Nivel de mensajes**, que contiene una performativa y que indica el protocolo de la interpretación.
- **Nivel de contenidos**, que contiene información relativa a la performativa enviada.

El formato de un mensaje en KQML empieza con una palabra como "register", que es la acción (performativa) que intenta el mensaje. El resto del mensaje contiene las palabras claves para las capas de mensaje y comunicación.

19. Carlos Ángel Iglesias Fernández, "Introducción a la Inteligencia Artificial Distribuida", Universidad Politécnica de Madrid, 1999.


```
(tell
  :sender agent1
  :receiver agent2
  :in-reply-to id1
  :language KIF
  :ontology blocks-world
  :content
    (and (block A)
         (block B)
         (on A B))
)
```

Ejemplo de mensaje KQML²⁰

Las palabras claves usadas en los mensajes KQML se definen de la siguiente manera:

- **sender**: agente que envía el mensaje.
- **receiver**: agente que recibe el mensaje.
- **from**: emisor original; usado cuando el mensaje es enviado a través de agentes intermediarios.
- **to**: receptor final; usado cuando el mensaje es enviado a través de agentes intermediarios.
- **in-reply-to**: identificador del mensaje que ha disparado el envío de este mensaje.
- **reply-with**: identificador usado por el mensaje que conteste a este mensaje.
- **language**: lenguaje para interpretar la información en el campo de contenido de este mensaje.
- **ontology**: identifica la ontología para interpretar la información en el campo de contenido de este mensaje.
- **content**: Información específica del contexto que permite describir las características específicas del mensaje.

5.11.2 - Inconvenientes de KQML

La acogida de la comunidad de investigadores y desarrolladores de sistemas multiagentes a KQML fue buena y por eso se desarrollaron diversas implementaciones, pero el sistema también tuvo sus críticas debido a los siguientes aspectos:

- La semántica de las performativas de KQML no fue desarrollada de una forma rigurosa (sólo informalmente). Por este motivo si dos agentes hablaban KQML era imposible afirmar si estaban usando el lenguaje de forma adecuada.
- El conjunto de performativas de KQML era demasiado grande y “ad hoc”.
- Los mecanismos de transporte para los mensajes KQML no estaban definidos de forma precisa, lo que impedía a los agentes que hablaban KQML pudieran interoperar.
- Faltaba un tipo entero de performativas de las denominadas de compromiso.

5.11.3 - KIF

Knowledge Interchange Format (KIF) es un lenguaje para el intercambio de conocimiento entre computadoras. En un principio fue diseñado como un intento de constituir un lenguaje común para expresar propiedades de un dominio en particular.

KIF es un lenguaje estrechamente basado en la lógica de primer orden y su notación es un claro ejemplo de esto. Como características del lenguaje podemos señalar:

- **Esenciales**
 - Semántica declarativa
 - Completo

20. Dr. Miguel Rebollo, “Comunicación entre agentes”, *Departamento de Sistemas Informáticos y Computación*, Universidad Politécnica de Valencia, 2002.

- Permite representar meta-conocimiento
- **Optimizaciones**
 - Capacidad de ser traducido
 - Legibilidad
 - Usabilidad

Para la representación del conocimiento, KIF utiliza una base de conocimiento (KB) formada por un conjunto (no secuencia) finito de fórmulas constituido por:

- **Términos**, que se utilizan para representar objetos del mundo (variables, símbolos de objeto, símbolos de función, símbolos de relación, términos funcionales, etc.).
- **Sentencias**, que se utilizan para representar los hechos sobre el mundo (constantes lógicas, ecuaciones, desigualdades, sentencias relacionales, etc.).
- **Reglas**, que se utilizan para los pasos de inferencia (KIF permite la definición de reglas de inferencia hacia adelante y hacia atrás, y cuando las premisas de una regla son sentencias se denominan pre-requisitos, etc.).
- **Definiciones**, que se utilizan para las declaraciones categóricas y axiomas (completas, expresión que define el comportamiento completamente o parciales, restringen el concepto sin proporcionar necesariamente una equivalencia completa).

5.11.4 - Ejemplo KQML/KIF

Para poder establecer una comunicación dentro de esta metodología ambos agentes deben aceptar un conjunto de términos que les permitirá establecer la comunicación. La especificación formal de este conjunto de términos es conocido como la ontología.

Esta situación lleva a generar un gran esfuerzo para lograr una definición de ontologías comunes, y poder así mantener normativas en las comunicaciones entre agentes pero existen incluso herramientas que fueron desarrolladas para este propósito, tal como la Ontolingua.

En efecto, esta Ontolingua fue realizada por el Knowledge Software Laboratory (KSL) de la Universidad de Stanford y sirve para crear, modificar y usar ontologías. La Ontolingua se encuentra disponible en la web y cualquier persona que solicite el correspondiente permiso puede utilizarla de forma virtual.

El componente central de Ontolingua es la biblioteca de ontologías, expresada en un lenguaje especial basado en KIF. El programa del servidor provee acceso a esta biblioteca y permite de esta manera proveer una plataforma común en la cual las ontologías desarrolladas por distintos grupos pueden ser compartidas en la búsqueda de un punto de vista en común, tal como fue definido en el objetivo del proyecto.

Para que veamos un pequeño ejemplo de una simple comunicación realizada entre dos agentes A y B en KQML/KIF. Contamos con las siguientes líneas, en las que A intenta averiguar preguntándole a B si 'chip1' es mayor a 'chip2':

```
A to B:      (ask-if (> (size chip1) (size chip2)))
B to A:      (reply true)
B to A:      (inform (= (size chip1) 20))
B to A:      (inform (= (size chip2) 18))
```

5.12 - Mecanismos y servicios de transporte

Por otra parte, los mensajes deben poder ser:

- Planificables o servidos por eventos
- Síncronos o asíncronos
- Direcciones físicas o por rol
- Unicast / multicast / broadcast

Implementaciones:

- CORBA
- RMI
- DCOM

El servicio de transporte es capaz de enviar un mensaje, codificarlo para su transmisión como una secuencia de bytes. En el caso más habitual, el servicio es

- **De confianza**, los mensajes bien formados llegan a su destino.
- **Fiable**, el mensaje se recibe tal y como se envía.
- **Ordenado**.

Por otra parte, un agente tiene la opción de decidir si usa procesamiento síncrono o asíncrono. Además, los parámetros del acto de envío de mensajes, como un plazo máximo si no hay respuesta, no se codifican

en el nivel de mensaje, pero son parte del interfaz proporcionado por el servicio de envío de mensajes.

Así, el servicio de envío de mensajes detecta condiciones de error, como mensaje mal formado, no-enviable, agente no alcanzable, etc.

Así, un agente tendrá un nombre tal que permita al servicio de envío de mensajes remitir el mensaje a la dirección correcta. El servicio de envío de mensajes debe ser capaz de determinar el mecanismo de transporte correcto (TCP/IP, SMTP, HTTP, etc.) y permitir cambios en la ubicación del agente si es necesario.

5.13 - Coordinación

Las tareas de coordinación son aquellas necesarias para que un sistema multiagente pueda coordinar las acciones y de esta manera conseguir la articulación de las acciones individuales a fin de que el sistema se comporte de forma coherente.

Algunas razones para coordinar las acciones en un sistema multiagente son:

- La necesidad de compartir información o resultados.
- La necesidad de coordinar el uso de recursos limitados.
- La optimización de costes eliminando tareas improductivas o repetidas.
- La existencia de objetivos distintos pero interdependientes.

Esta coordinación dentro del sistema multiagente puede darse de una forma directa cuando todos los agentes se comunican con todos a través de un mediador que coordine todas estas comunicaciones o mediante el uso de sistemas federados donde conviven un mediador y distintos agentes formando conjuntos.

En esta última opción aparecen algunas ventajas como una menor necesidad de coordinación, aunque también algunas desventajas como la existencia de cuellos de botella al encontrarse todas las operaciones dependiendo de un solo elemento.

En una coordinación debe estar claro el objetivo final que es el de evitar los conflictos entre agentes y aquí se deberá tener en cuenta dos tipos de relaciones posibles entre las acciones: las positivas y las negativas.

Las positivas son aquellas en la que una acción favorece la realización de otra. Dentro de esta clasificación se encuentran a) las igualitarias donde no tiene que haber un agente preasignado a la acción, b) las incluidas en las que una acción es parte de otra acción y c) las favorecedoras donde una acción favorece a otra.

Por el lado de las negativas, que son aquellas que impiden que las acciones se desarrollen simultáneamente, encontramos a) las de incompatibilidad de objetivos y b) las de insuficiencia de recursos consumibles o no consumibles.

5.13.1 - Formas de coordinación

Cabe agregar que existen distintas formas de coordinación que se pueden llegar a utilizar. Las más importantes son las siguientes:

- **Sincronización:** representa la forma más elemental ya que se establece la secuencia temporal de acciones, en qué momento temporal se debe realizar cada acción y se evitan las interferencias entre acciones.
- **Planificación:** se crean planes y se ejecutan. En este caso existe la necesidad de coordinación y es adecuada para entornos predecibles, y sistemas poco flexibles y lentos. El proceso de planificación consiste en:
 - Pensar el plan.
 - Distribuir acciones y coordinarlas.
 - Ejecutar el plan.
 - Supervisar la ejecución (re-planificación).

Así, existen tres tipos: la planificación centralizada, la centralizada de planes parciales y la distribuida.

- **Coordinación reactiva**, en la cual no se planifica. Existen dos casos típicos:
 - Con objetivos independientes, donde el movimiento de los agentes se produce en una misma porción del espacio y existe el acceso a recursos ilimitados.
 - Con objetivos dependientes, donde a su vez se utilizan dos técnicas: la de uso de reglas simples y la de uso de marcas de entorno.
- **Coordinación por regulación**, en la cual se establecen reglas de comportamiento que evitan los conflictos.

5.14 - Construcción de un sistema multiagente

Para poder construir un agente no existe una metodología única o fuertemente arraigada en los diseñadores de software pero se pueden definir globalmente cuatro pasos:

- **Definición de número y tipo de agentes**
- **Definición del modelo de agente** (o de agentes si el modelo es heterogéneo)
- **Definición de la organización del sistema multiagente**
- **Implementación del sistema multiagente**

5.14.1 - Herramientas para la construcción de agentes

Acompañando al crecimiento de los agentes inteligentes, fueron apareciendo distintas herramientas que ayudan a su creación. Algunas son comerciales como el AgentBuilder o el Agent Development Kit (ADK) de Tryllian de Estados Unidos, y también existen algunas gratuitas como JADE (Java Agent Development Framework), desarrollado por un grupo de personas que participaron en distintos momentos del proyecto, muchos de ellos italianos.

AgentBuilder es, como la definen sus creadores, una herramienta de software integrada que permite desarrollar rápidamente agentes de software inteligentes y también aplicaciones basadas en agentes.

En el caso de la herramienta de Tryllian, se pueden destacar algunas características como la programación en Java, la utilización de SSL y XML como métodos de comunicación, una orientación al desarrollo de agentes móviles y la proporción de una API gráfica para programar agentes además de un entorno para ejecutarlos.

Pero probablemente la más utilizada en ambientes académicos sea JADE, que es gratuita y para su desarrollo cuenta con el apoyo de varias empresas como Motorola y France Telecom aunque es propiedad de Telecom Italia. Esta herramienta proporciona un middleware compatible con FIPA.

5.15 - FIPA

La Fundación para los Agentes Físicos Inteligente (FIPA, en inglés) es una fundación sin fines de lucro compuesta por diversas compañías y organizaciones con el objetivo de producir especificaciones estándar para la tecnología de agentes.

El 8 de Junio de 2005, la FIPA fue aceptada por la IEEE como la organización de estándares para los agentes y los sistemas multiagentes, lo cual la convierte en la referencia oficial de cualquiera que intente desarrollar esta tecnología en el mundo.

FIPA fue originalmente formada en el año 1996 como una organización suiza para producir especificaciones de software a fin de lograr la interacción entre agentes y sistemas basados en agentes heterogéneos.

Sus primeros pasos fueron impulsados gracias al apoyo de Telecom de Italia y ya en el año 2002 consiguió presentar el primer estándar cuyo objetivo era, como el de cualquier estándar, facilitar la interoperabilidad entre los agentes.

Parte de su política es la de no involucrarse en la arquitectura interna de los agentes, sino sólo en sus comportamientos externos. Para esto definió un modelo de referencia de una plataforma multiagente y un conjunto de servicios que ésta proporciona.

Dentro de las especificaciones FIPA podemos encontrar elementos acerca de la interacción entre agentes y humanos, gestión de seguridad en los agentes, gestión de soporte en la movilidad, servicio de ontología, guía del desarrollador y otros 94 documentos de especificaciones.

5.15.1 - FIPA - ACL

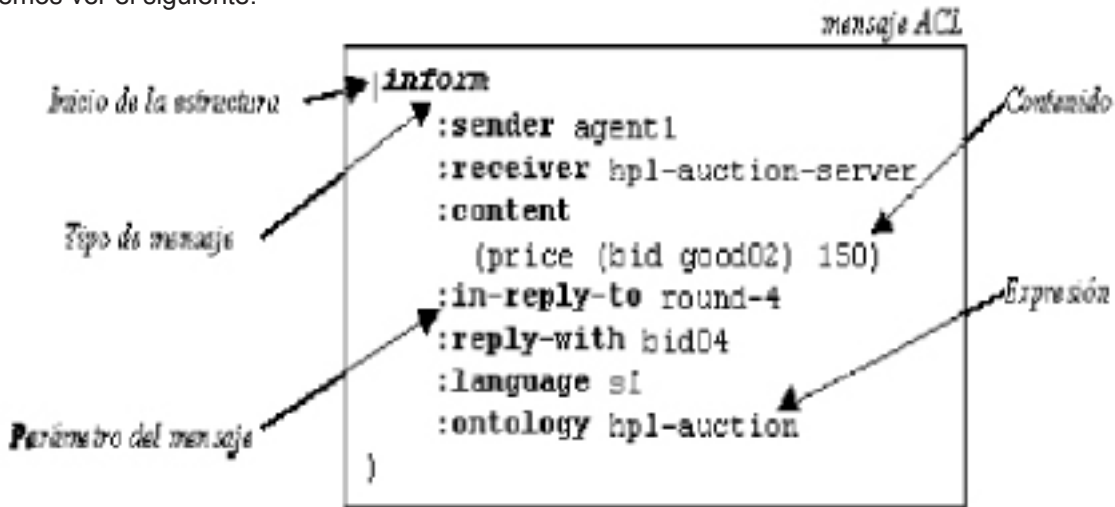
El objetivo principal de esta arquitectura abstracta es conseguir el intercambio de mensajes con contenido semántico (que representan actos de habla) entre agentes, que pueden utilizar diferentes mecanismos de transporte, lenguajes de comunicación (ACL, en inglés) o lenguajes de contenido.

Esta es la pieza central de la búsqueda de la estandarización por parte de FIPA y se plasma en los 22 actos comunicativos distintos que permiten representar las intenciones de la comunicación que se encuentran en el ACL.

Para cada una de las acciones comunicativas se muestra dentro de las especificaciones FIPA una descripción detallada en lenguaje natural de la acción y sus consecuencias, un modelo formal escrito en SL de la semántica de la acción, sus precondiciones formales y sus efectos y por último, algunos ejemplos de uso del acto comunicativo.

Así, estas acciones pueden ser agrupadas en función de la actividad social a la que se encuentran destinadas generando cuatro grandes grupos: las de información, la realización de acciones, la negociación y la intermediación lo cual lleva a la necesidad de incluir mecanismos para el registro, localización y transferencia de mensajes entre agentes.

Un mensaje FIPA-ACL encapsula el contenido de estos mensajes utilizando una serie de atributos definidos y estandarizados por la FIPA. Como ejemplo de un mensaje formateado según estas normas podemos ver el siguiente:

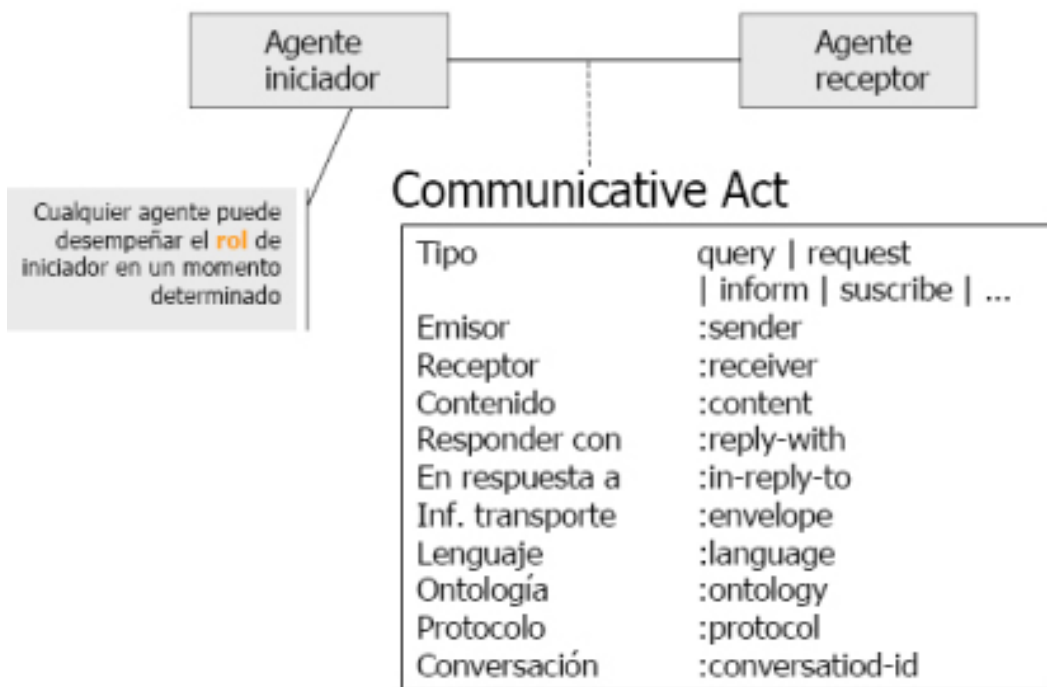


Estructura de un mensaje ACL²¹

En el ejemplo se muestra el contenido de un mensaje ACL de acuerdo con la especificación FIPA y la descripción de sus componentes.

Por otra parte, existen requerimientos muy importantes en ACL:

- Los agentes deben estar preparados para enviar un mensaje “not-understood”, si reciben un mensaje que no reconocen o cuyo contenido no son capaces de procesar. De la misma manera deben ser capaces de recibir y manejar adecuadamente estos mensajes provenientes de otros agentes.
- Un agente ACL puede seleccionar cualquier subconjunto predefinido de tipos de mensajes y protocolos. La implementación de estos mensajes debe ser correcta respecto a la definición semántica de los actos referenciados.
- Un agente ACL que utilice actos comunicativos debe implementarlos correctamente según su definición.



Modelo de comunicación ACL entre agentes²²

21. L. F. Castillo, M. G. Bedia & J. M. Corchado, “Librería para el intercambio de mensajes FIP-ACL entre dispositivos móviles a través de Bluetooth”, *Dep. Ciencias Computacionales*, Universidad Autónoma de Manizales y *Dep. Informática y Automática*, Universidad de Salamanca, 2005.

22. Dr. Miguel Rebollo, “Comunicación entre agentes”, *Departamento de Sistemas Informáticos y Computación*, Universidad Politécnica de Valencia, 2002.

- Los agentes pueden utilizar actos comunicativos con otros nombres pero son los responsables de que el agente receptor comprenda el significado del mismo. Sin embargo, un agente no puede definir un acto comunicativo con otro lenguaje si su significado coincide con algunos de los actos estándares predefinidos.
- Un agente ACL debe ser capaz de generar mensajes sintácticamente bien formados en la forma de transporte que corresponde al mensaje que desea enviar. De la misma manera, debe ser capaz de traducir una secuencia de caracteres que esté bien formada en la sintaxis del mensaje correspondiente.

Parameter	Category of Parameters
performative	Type of communicative acts
sender	Participant in communication
receiver	Participant in communication
reply-to	Participant in communication
content	Content of message
language	Description of Content
encoding	Description of Content
ontology	Description of Content
protocol	Control of conversation
conversation-id	Control of conversation
reply-with	Control of conversation
in-reply-to	Control of conversation
reply-by	Control of conversation

Parámetros de mensajes FIPA ACL²³

5.15.2 - FIPA – Arquitectura abstracta

Para permitir el funcionamiento de los estándares de comunicación de FIPA al utilizar su ACL, existen dos tipos de servicios posibles: el servicio de directorio y el servicio de transporte de mensaje.

El servicio de directorio proporciona un lugar donde los agentes registran información (entradas de directorio). Una información que es utilizada por los agentes para encontrar otros con los que desean interactuar. Por su parte, cada entrada de directorio se compone de:

- **Nombre del agente**, el cual es global y único.
- **Localizador** que contiene un conjunto de descripciones de transporte. Es una estructura que contiene el tipo de transporte, una dirección específica para ese tipo y cero o más propiedades.
- **Atributos**, que son un conjunto de propiedades asociadas a un agente. Los atributos pueden utilizarse para llevar a cabo una búsqueda de agentes que cumplan ciertas características y aquí se puede recoger información sobre las ontologías que entiende el agente, servicios que proporciona, etc.

En el caso de este servicio, un agente puede tener varias descripciones de transporte permitiendo, por lo tanto, diferentes formas de comunicarse con otros agentes.

Por otro lado existe también el servicio de transporte de mensajes, en el cual la especificación también cubre el lenguaje de comunicación entre agentes. Éste se basa en el paso de mensajes que representan actos de habla y los agentes se comunican formulando y enviando los unos a los otros mensajes individuales.

La especificación de FIPA ACL (FIPA ACL Message Structure Specification), especifica un lenguaje de paso de mensajes estándar estableciendo la codificación semántica (elimina cualquier ambigüedad y confusión derivada del uso del lenguaje) y pragmática de los mensajes que se codifican utilizando una codificación y se transmiten sobre un transporte.

El documento (FIPA Message Transport Specification) contiene información detallada sobre el transporte de mensajes. El estándar no especifica un mecanismo concreto para el transporte interno de los mensajes, mientras que FIPA únicamente establece que los mensajes intercambiados entre las diferentes

23. Foundation for Intelligent Physical Agents, SC00061G FIPA ACL Message Structure Specification, 2002.

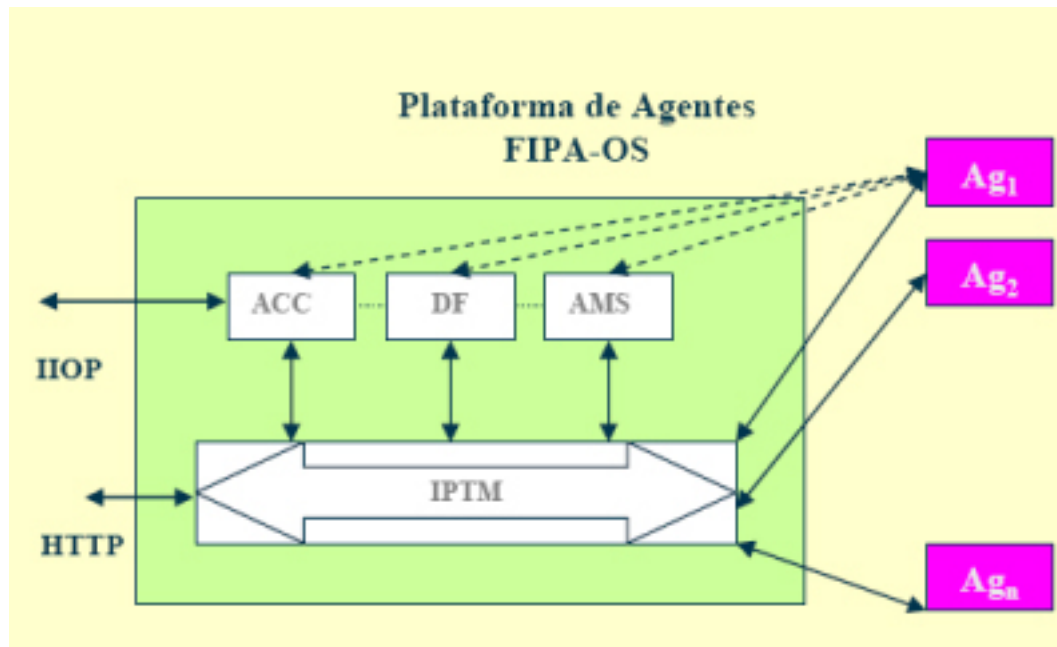
plataformas deben ser codificados en forma textual (XML, Strings FIPA u objetos Java serializados). De esta forma se garantiza la comunicación entre diferentes plataformas que utilicen tecnologías o infraestructuras distintas.

En este tipo de servicio, cuando se envía un mensaje, se transforma utilizando una representación codificada apropiada para su transporte.

5.15.3 - FIPA - Modelo de referencia de agentes

Este modelo proporciona el marco de trabajo normativo donde los agentes existen y operan. Así se identifican los servicios que debe proporcionar toda plataforma de agentes. Cada uno de estos servicios, excepto el de transporte de mensajes, es suministrado por agentes especializados, lo que supone que la comunicación entre ellos se llevará a cabo mediante mensajes ACL que se dividen en:

- **AMS** (Sistema de gestión de Agentes)
 - Es el agente que supervisa y controla el acceso y uso de la plataforma.
 - Conoce en todo momento el estado de la plataforma y los agentes que pertenecen a ella.
 - Su misión consiste en la creación, destrucción, control de cambio de estado de los agentes, autenticación y registro de los agentes, control de movilidad, gestión de los recursos compartidos y gestión del canal de comunicación.
 - Presta un servicio de páginas blancas en el que el nombre de cada agente registrado tiene asociado su dirección de transporte.
- **ACC** (Canal de Comunicaciones para los agentes)
 - Servicio de transporte de mensajes
- Es el encargado de proporcionar, a la plataforma, los mecanismos básicos para en el envío de mensajes, tanto con agentes locales como remotos y pertenecientes a la misma plataforma FIPA o a otra distinta.
- Cuando un agente en una plataforma quiere enviar un mensaje a otro en una localización remota, entrega este mensaje al STM que se encargará de entregárselo al STM de la plataforma donde se encuentre el otro agente.
- El último STM comunicará al agente destino que ha recibido un mensaje.
- El modelo de comunicación entre agentes es asíncrono. Este hecho implica que el STM no queda bloqueado ante el envío o recepción de mensajes, para los que deben definirse políticas de gestión de colas de mensajes.



Modelo de referencias de agentes de FIPA²⁴

- **DF** (Servicio de Directorio)
 - Es el agente encargado de proporcionar un servicio de páginas amarillas a la plataforma, en las que cada agente registra los servicios que es capaz de proveer.

24. María Victoria Belmonte Martínez, "Programación de sistemas multiagente", *Departamento de Lenguajes y Ciencias de la Computación*, E.T.S.I. Informática, Universidad de Málaga, 2006.

- De esta forma es posible buscar un agente por sus capacidades y no sólo por su nombre.
- **IPMT** (Un sistema encargado del transporte de mensajes)

5.15.4 - FIPA – Protocolos de interacción

La estructura de la comunicación o conversación entre agentes a menudo se corresponde con patrones típicos de secuencias de mensajes. A estas secuencias se las denomina Protocolos de Comunicación o Interacción (IP).

FIPA IPL (FIPA Interaction Protocol Library) es una biblioteca compuesta por 11 protocolos de interacción estándar. Para cumplir el estándar FIPA no es necesario que el agente haga uso de uno de estos protocolos, pero si los emplea han de ajustarse a la especificación de FIPA.

Por su parte, un agente puede participar simultáneamente en diálogos múltiples con diferentes agentes utilizando diferentes IP y el término conversación se usa para representar cualquier instancia de este diálogo.

Para la definición de estos protocolos estándares, FIPA definió un lenguaje de representación de protocolos llamado AUML. Este lenguaje está basado en los diagramas de secuencias de UML, y sus diagramas se denominan diagramas de protocolo. Dentro de los diagramas AUML se pueden distinguir los siguientes elementos:

- **Agentes y Roles.** Un rol de agente puede considerarse como el conjunto de agentes que ofrecen determinados servicios, son capaces de cierto tipo de comportamiento, o simplemente satisfacen una determinada interfaz. El rol de un agente se representa gráficamente mediante un rectángulo que incorpora el nombre del rol de agente.
- **Vida y Actividad.** La línea de vida define el tiempo en el que existe un agente. Se representa gráficamente mediante una línea vertical intermitente, mientras que los hilos de interacción indican los períodos dentro de la línea de actividad de un rol, en los que los agentes correspondientes realizan alguna tarea (como reacción a un mensaje recibido). Se representan mediante rectángulos superpuestos a la línea de vida.

El posible paralelismo en las tareas de un rol se modela mediante la bifurcación y/o reunión de la línea de vida correspondiente. Lo más usual es el paralelismo “o-exclusivo”, que da lugar a hilos de interacción alternativos y mutuamente exclusivos. Dicha bifurcación se representa gráficamente mediante una línea horizontal y un rombo que contiene un ×.

- **Mensajes.** Se modelan mediante una flecha que va del rol emisor al rol receptor. Los mensajes que se intercambian entre agentes suelen ser asíncronos, por lo que se representan en los diagramas de protocolos por flechas de cabeza “hueca”. Los PDs permiten la definición de mensajes compuestos. La forma típica de construir dichos mensajes en los IPs de agentes es el ó exclusivo, es decir: el agente selecciona exactamente uno de los distintos mensajes alternativos. Gráficamente se denota por un rombo que contiene una x.

5.16 - Sistemas de agentes heterogéneos

Se llama de esta manera a los sistemas multiagentes que se encuentran formados por al menos dos o más agentes que pertenecen a clases de agentes diferentes. Es aquí donde la ingeniería de software basada en agentes debe actuar para facilitar la interoperación de agentes de software heterogéneos.

Este tipo de sistemas es muy común en la actualidad ya que existe una creciente demanda para que los agentes puedan interoperar y de esta manera proporcionar servicios añadidos pero el requisito clave para permitir esta interoperabilidad es la utilización de un ACL en común.

De esta manera, agentes que tengan arquitecturas distintas, objetivos distintos, ambientes distintos o cualquier otra característica distinta pueden relacionarse entre sí y obtener lo que se necesita.

En el caso que uno de los elementos de este sistema que debe interactuar con un agente no es un agente en sí sino un software existente encontramos tres opciones:

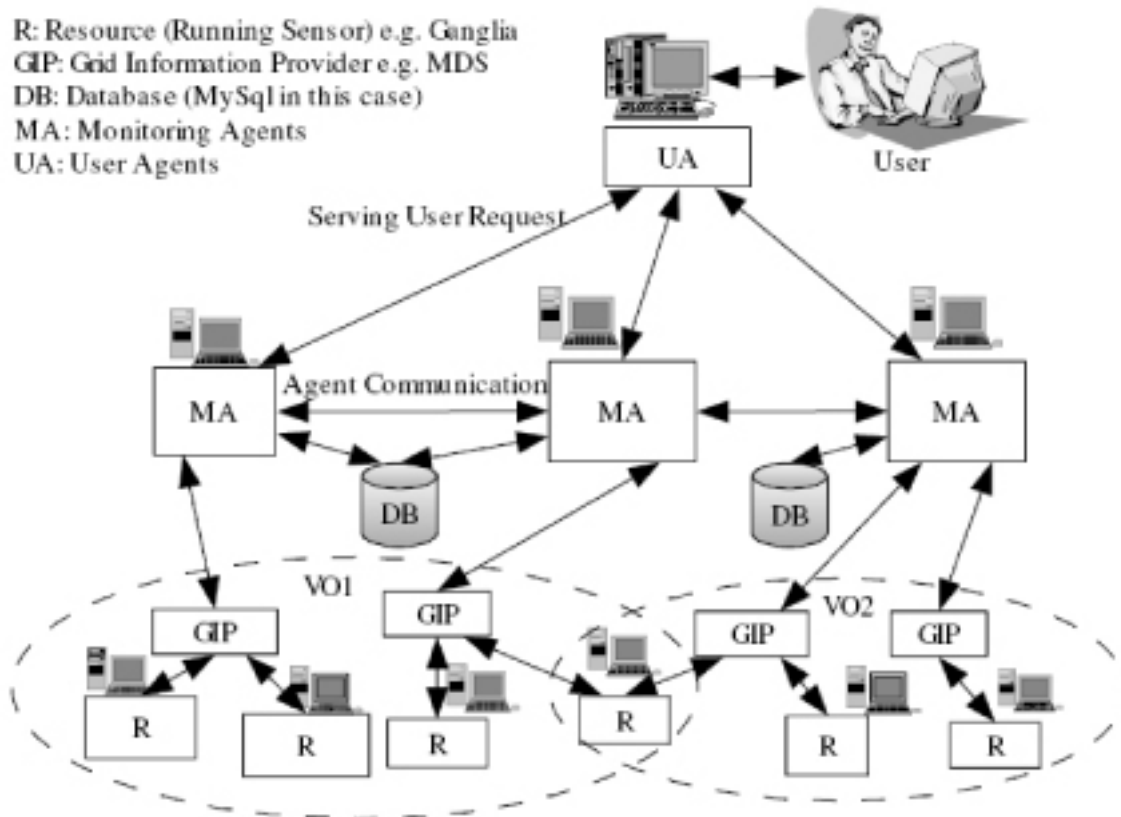
- **Reescribirlo totalmente:** con el fin de cumplir los requisitos de agente.
- **Utilizar un traductor:** Programa que reciba mensajes de otros agentes (en ACL), los traduzca al protocolo de comunicación propio del sistema y los pase al sistema. Se debe implementar el paso inverso (traducir la respuesta del sistema a ACL).
- **Wrapper:** Se inserta en el programa una parte de código para permitir la comunicación en ACL. Es la opción más utilizada por ser la más eficiente.

5.17 - Ejemplo de arquitectura: MAGGIS

MAGGIS es un sistema multiagente que utiliza una arquitectura clásica con dos capas, una de colección de datos que comprende todos los componentes de monitoreo y sintetización, y otra capa que

maneja el conocimiento por lo que incorpora la representación de datos, el modelado, la comunicación y el análisis de componentes.

Para ver estas dos capas de una manera funcional podemos recurrir a la siguiente figura:



Arquitectura del sistema multiagente MAGGIS²⁵

Aquí se puede ver cómo la colección de datos es equivalente a los agentes de monitoreo mientras que la capa de conocimiento es equivalente a los agentes de usuario. De esta manera los agentes de monitoreo tienen a su cargo el monitoreo y sintetización de información así como el modelado de los datos. Los agentes de usuario tienen la responsabilidad de analizar la información y presentar los resultados correspondientes al usuario.

Además, los agentes del usuario tienen la tarea de recibir y administrar las solicitudes de los usuarios así como también la de almacenar los perfiles de cada uno.

En este sistema multiagente la comunicación entre los agentes es implementada mediante el uso de KQLM y el almacenaje de la información que manejan los agentes de monitoreo se basa en un modelo espacio-temporal de datos, el cual a su vez adopta un modelo conceptual de pirámide para el manejo de la información.

Por otra parte, una gran ventaja que se puede encontrar en el diseño realizado en MAGGIS es la capacidad de poder agregar información desde distintas fuentes de datos sin tener que alterar el diseño del sistema.

En ambas capas de agentes se definen funcionamientos claros, de manera que cuando un agente de monitoreo es instanciado, este adquiere información desde la fuente correspondiente implementando un modelo multi-hilos, la cual le permite realizar de forma muy dinámica la extracción de datos. De la misma manera, los agentes de servicios al usuario cuentan con un mecanismo multi-hilo no-bloqueante que le permite manejar solicitudes de distintos usuarios.

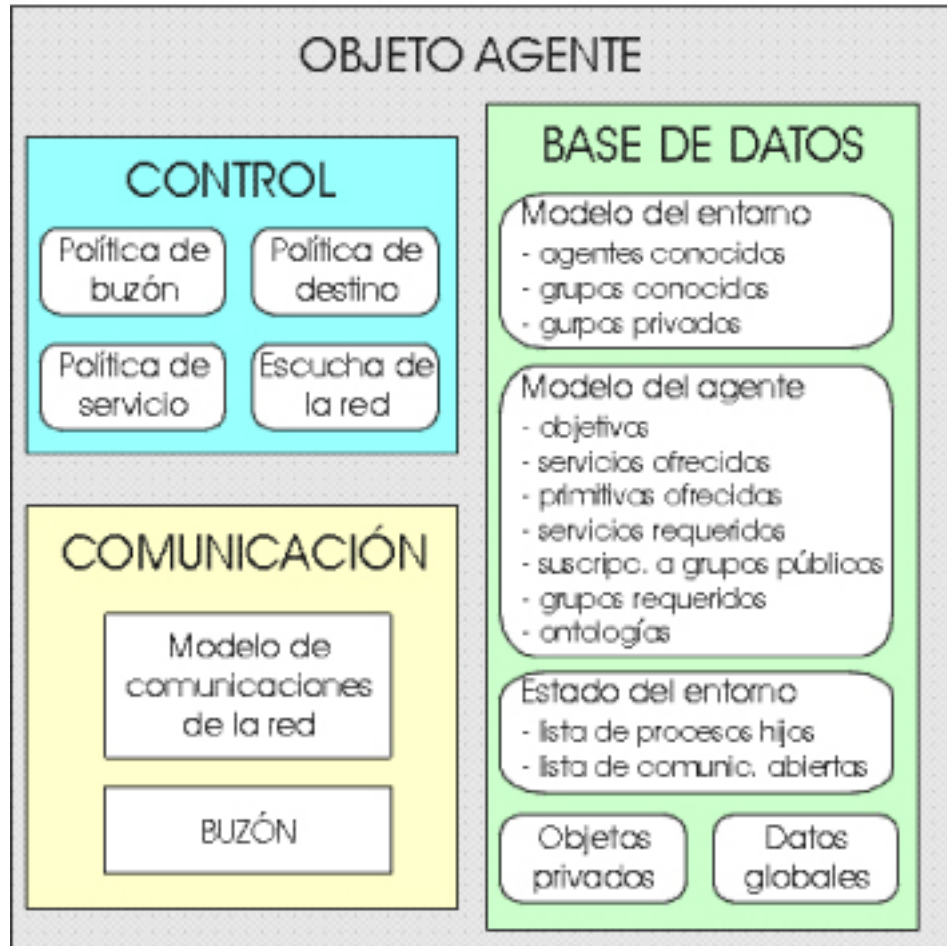
5.18 - El proyecto MIX: enfoque multiagente para sistemas híbridos

Cabe agregar que la complementariedad entre el enfoque simbólico y el conexionista en la resolución de problemas ha despertado un interés creciente por sistemas híbridos que utilicen de forma integrada ambos enfoques. Estos sistemas son el centro de interés del proyecto ESPRIT MIX (Modular Integration

25. Shaowen Wang, Anand Padmanabhan, Yan Liu, Ransom Briggs, Jun Ni, Tao He, Boyd M. Knosp & Yasar Onel, "A Multi-agent System Architecture for End-User Level Grid Monitoring Using Geographic Information Systems (MAGGIS): Architecture and Implementation", *Academic Technologies-Research Services of Information Technology Services*, The University of Iowa, 2003.

of Connectionist and Symbolic Processing in Knowledge-Based Systems), patrocinado por la Unión Europea (Programa ESPRIT, proyecto 9119).

MODELO DE AGENTE MIX



Modelo de agente MIX²⁶

Los objetivos generales del proyecto, propuestos por el equipo investigador (José Carlos González Cristóbal, Juan Ramón Velasco y Carlos A. Iglesias Fernández) son los siguientes:

- Mejorar la versatilidad de los sistemas híbridos simbólico-conexionistas combinando esquemas de representación, inferencia y aprendizaje originarios de los paradigmas simbólico y conexionista.
- Aumentar las capacidades de razonamiento de estos sistemas mitigando su limitación tradicional a la lógica proposicional.
- Clarificar el impacto teórico de la integración simbólico-conexionista sobre aspectos fundamentales del aprendizaje automático, como la elección y la combinación de métodos, asignación de méritos y la asimilación de conocimiento.
- Asegurar la escalabilidad y utilidad real de los modelos híbridos resultantes aplicándolos a problemas de clasificación, predicción y optimización tanto en el ámbito industrial como en otros campos.

Así algunas características del proyecto son:

- La existencia de dos clases fundamentales de agentes:
 - Agente básico (clase BaseAgent)
 - Agente servidor de nombre (clase YPAgent)
- La necesidad de que debe existir un único agente servidor de nombre para cada dominio con una dirección fija y conocida por el resto de los agentes.
- La inclusión de varias políticas de selección de agentes para el envío de mensajes y de respuestas.

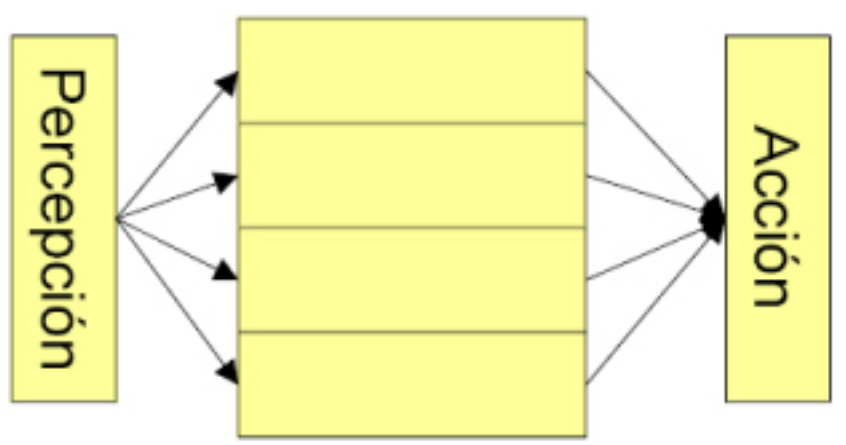
26. Ramón M. Gomez Labrador, "El lenguaje ADL", *Departamento de Lenguajes y Sistemas Informáticos*, Universidad de Sevilla, 1997.

- La existencia de dos tipos de grupos de agentes: públicos y privados.
- Se añade el protocolo de comunicaciones Contract Net.
- Este modelo incluye el Lenguaje para la Definición de Agentes (ADL), que permite la definición de agentes y clases de agentes. Es un lenguaje que no permite definir servicios, éstos deben programarse en C++.

5.19 - Agentes híbridos

Este tipo de agente junta las ventajas de las arquitecturas reactivas y las deliberativas evitando determinados inconvenientes que los llevan a tener una estructura de dos capas. Para esta clase de agente existen dos arquitecturas básicas, una horizontal y una vertical.

La arquitectura horizontal es la que en todas sus capas recibe las percepciones que son capaces de seleccionar las acciones. Además cada capa actúa como un agente individual.



Arquitectura horizontal²⁷

Así, puede existir competencia entre las capas por lo que se puede contar con un mediador, una arquitectura de subsunción, redes neuronales, etc.

En el caso de la arquitectura vertical sólo una o algunas de las capas reciben la percepción y lo mismo ocurre para la acción. De esta manera se obtiene un comportamiento menos flexible aunque más robusto que permite comportamientos más complejos. Pueden ser de uno o dos pasos dependiendo el caso.

5.20 - Arquitectura modular

Es una de las arquitecturas más utilizadas especialmente en los agentes deliberativos y como lo indica su nombre, todo su diseño se basa en la estructura de módulos.

Algunos de los módulos más comunes son:

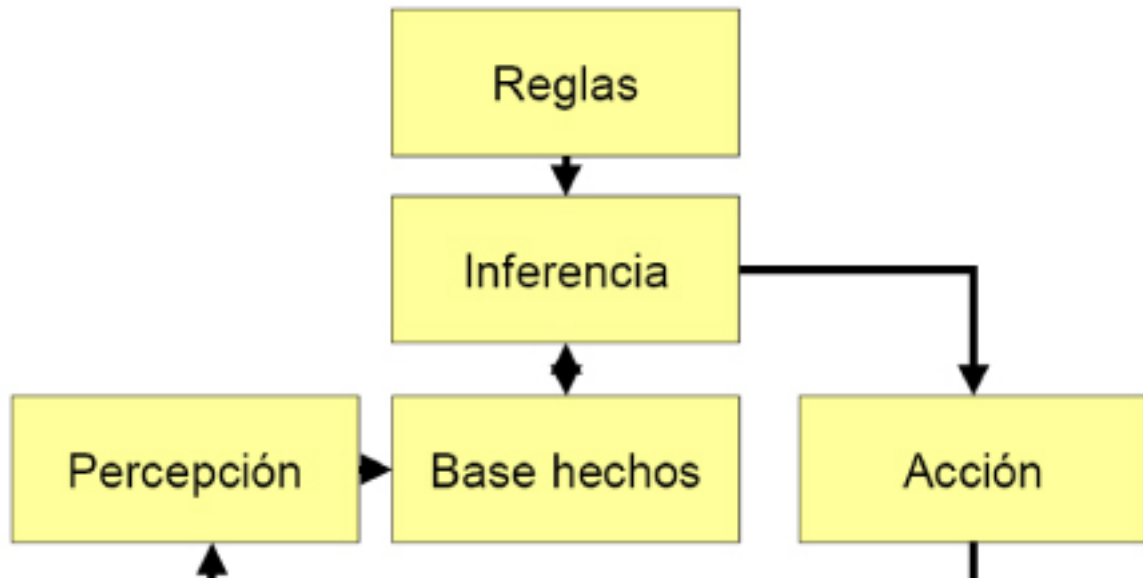
- Percepción y acción
- Comunicación
- Base de creencias
- Gestión de compromisos
- Habilidades relevantes y toma de decisiones
- Planificación de acciones

5.21 - Sistemas de producción

Ésta es una de las arquitecturas más clásicas, y cuenta con un sistema basado en reglas. Como componentes se pueden distinguir una base de hechos, un conjunto de reglas y un mecanismo de inferencia.

Su funcionamiento es tal que la percepción añade datos a la base de hechos y si varias reglas compiten, el mecanismo de inferencia selecciona una (incluso 'aleatoriamente').

27. Diseño de software de agentes inteligentes autónomos, *Agentes Inteligentes*, Universidad Carlos III, 2005/6.



Sistemas de producción²⁸

En esta arquitectura existen dos problemas, la no monotonía o sea que el orden en que se aplican las reglas puede ser relevante y por lo general, las reglas no se pueden combinar.

5.22 - Código de buena conducta en los agentes

Según Eichmann y Etzioni²⁹, los agentes son un elemento más que interactúa en nuestra sociedad y por lo tanto deben cumplir un código de conducta que debería contar con los siguientes puntos:

- Los agentes se deben identificar adecuadamente.
- Deben moderar el consumo de recursos que llevan a cabo de un determinado servidor.
- Deben interactuar solamente con los servidores relevantes.
- Deben compartir la información.
- Deben respetar la 'autoridad' del ambiente en que trabajen.
- Deben dar información veraz y actualizada acerca de su estado y potencialidades.
- No deben alterar destructivamente el entorno.
- Deben dejar el entorno tal y como lo encontraron.
- No deben ejecutar acciones de las que desconozcan las consecuencias.

5.23 - Aplicaciones de agentes

Hay muchas aplicaciones para los sistemas multiagentes como:

- **Resolución cooperativa de problemas**, como la gestión de plantas eléctricas y redes de telecomunicaciones, control de tráfico aéreo, diagnóstico médico, etc.
- **Agentes interfaz**, como los sistemas que emplean técnicas de inteligencia artificial para apoyar a los usuarios en una tarea determinada.
- **Especializados en información**, como los sistemas que acceden a fuentes de información y la procesan para dar respuesta a los usuarios.

5.23.1 - Agentes asistentes personales

Estos agentes son tan flexibles que exhiben capacidades de adaptación al medio, de manera tal que liberan al usuario de tareas repetitivas usuales, basándose en la idea de la delegación.

Este tipo de agente puede actuar por iniciativa propia y no necesita esperar la del usuario. Además cuentan con distintas maneras de aprendizaje como la observación directa al usuario, mediante el uso de refuerzos positivos o negativos por parte del usuario, recurriendo a instrucciones explícitas o simplemente intercambiando información con otros agentes.

Algunos usos de estos agentes son el de gestión de agenda, búsqueda de información o productos en internet, la gestión del correo electrónico, creación de comunidades de interés, etc. Usualmente cuentan

28. Diseño de software de agentes inteligentes autónomos, *Agentes Inteligentes*, Universidad Carlos III, 2005/6.

29. D. Eichmann, "Ethical Web agents", *Second International World-Wide Web Conference*, pp. 3 – 13, 1994.

con una interfaz con gran cantidad de información como la bolsa, salas de control, salas de emergencia, etc.

5.23.2 - Agentes móviles

Estos agentes pueden migrar de una máquina a otra para realizar las tareas que se le han encomendado y por eso deben ser capaces de ejecutarse independientemente de la plataforma.

Así cuentan con dos posibilidades, una de un sólo salto en la que migran de un sitio a otro y la otra mediante varios saltos, donde van saltando de máquina en máquina hasta que completan su trabajo.

5.23.3 - Agentes de información

Estos agentes gestionan, manipulan e integran información recogida de diversas fuentes. Es muy común que sean especialistas en la búsqueda de información en internet y que sepan navegar por la red buscando y ordenando la información.

Además proveen de un servicio de acceso uniforme a bases de datos variadas. También se subscriben en estos términos los agentes de mediación (middleware), que cumplen misiones de facilitación.

5.23.4 - Agentes en internet

Con solo navegar un poco por la red podemos encontrarnos con que los agentes están rápidamente expidiéndose en Internet y brindando los más variados servicios. Algunos agentes que podemos encontrar pueden ser catalogados en rubros, como los asistentes personales que nos ayudan como lo haría un ayudante personal.

Se pueden encontrar que realizan funciones tales como:

- **Agenda inteligente.** Si le digo que estoy por ir a la playa, me dice que lleve toalla.
(<http://agents.media.mit.edu/projects/tasks/calendar.jpg>)
- **Asesor financiero.** Dependiendo si el usuario acepta o no los riesgos, por ejemplo, recomienda algunos valores o no.
(<http://agents.media.mit.edu/projects/investing>)
- **Auditor.** De acuerdo al tipo de cliente presenta el cuestionario de auditoria más apropiado a sus características.
(http://www.cica.ca/index.cfm/ci_id/1606/la_id/1.htm)

También se pueden encontrar agentes que negocian por uno en subastas virtuales (http://www.botspot.com/BOTSPOT/Windows/Shopping_Bots/Auction_Bots) o especies de agentes secretos o espías que monitorean una página identificada previamente por el usuario e informan cuando se producen cambios en esta (<http://www.changedetection.com>).

5.23.5 – Otros casos puntuales

Se pueden encontrar en el mercado mundial muchísimas aplicaciones que funcionan con las tecnologías de los agentes inteligentes y sistemas multiagentes. Los rubros son los más variados y podemos encontrar esta tecnología en aplicaciones de comercio internacional, de reportes financieros, simulación y entrenamiento, administración de redes, administración de interfaces, redes de telecomunicaciones, de planificación y optimización en logística y cadenas de abastecimiento, control de plantas industriales, modelado simulado para guiar la toma de decisiones, transporte, medicina, etc.

A continuación veremos muy brevemente siete casos reales donde se ha aplicado esta tecnología de forma exitosa:

- **Software orientado a agentes y su aplicación militar**

El proyecto HV-CGF (Human Variability in Computer Generated Forces) del Ministerio de Defensa del Reino Unido ha entendido que el uso de los agentes inteligentes para la simulación militar es la clave para una nueva generación de sistemas orientados a esta función.

Para esto, el proyecto trabaja con un sistema llamado "JACK", un ambiente basado en Java para la creación de sistemas multiagentes. Mediante JACK pueden crear los mas variados escenarios militares y dejar en mano de los agentes inteligentes creados el protagonismo de las simulaciones.

- **Eurobios y el modelo de simulación basado en agentes**

Eurobios ha creado para su cliente SCA Packaging, unas de las empresas líderes de producción de cartón corrugado a nivel mundial, un sistema de simulación y optimización basado exclusivamente en la tecnología de los agentes inteligentes.

Este sistema multiagente creado le permitió a la empresa optimizar todo su sistema de producción y el de logística, llevándola a incrementar sus beneficios, tanto económicos como empresariales.

- **Magenta y su sistema para mejorar la logística de distribución**

El sistema multiagente creado por Magenta esta pensado para la organización y optimización de la planificación de las embarcaciones de transporte.

Con este sistema se consiguió solucionar los dos grandes problemas que tenía la industria. Por un lado el manejo de grandes volúmenes de información que debía ser tenida en cuenta para la planificación de los barcos dentro de un ambiente dinámico y por el otro lado, la ineficiencia con la que contaba la compañía al tener los conocimientos específicos de cada restricción y característica de los envíos dependientes de los distintos empleados.

Los agentes que participan del sistema lograron solucionar de forma optima estos problemas y crear un punto de inflexión en los beneficios obtenidos por la compañía de transporte.

- **El sistema de logística de Whitestein**

Whitestein ha creado la “Living Systems Adaptive Transportation Networks”, una aplicación de logística de transporte basada en la búsqueda del bajo costo y optimización de los recursos.

Aunque solo el 20% del sistema cuenta con la tecnología de los agentes inteligentes, dentro del sistema completo los agentes cuentan con un papel primordial en la búsqueda de la optimización de la logística.

Por ejemplo, cuando un conductor le envía al sistema la información de su posición y la ruta propuesta, el sistema determina si el vehículo puede realizar alguna otra operación o si le conviene tal vez cambiar de ruta. Esta negociación es realizada exclusivamente por el sistema multiagente. Cada vehículo se encuentra representado por un agente inteligente y entre todos utilizan un protocolo parecido a las subastas para poder negociar.

- **Nutech Solutions y su sistema para la producción y distribución de la energía**

La empresa Nutech Solutions ha desarrollado un sistema pensado para una gran proveedora de energía que ha tenido que enfrentar, al igual que sus competidoras, a los cambios en el mercado de la energía donde ha aparecido la desregulación y la fluctuación de los precios y costos de distribución.

El sistema ha sido creado para una compañía que cuenta con unas 40 plantas y distribuye oxígeno y nitrógeno líquido a mas de 10 mil posiciones de sus clientes.

El sistema que han hecho busca la optimización en las operaciones de la empresa y para esto combina distintos algoritmos de búsqueda, de optimización y, por supuesto, la tecnología de los agentes inteligentes.

- **La solución de Acklin para el intercambio de información entre las compañías de seguros**

En el caso de Acklin, la aplicación de la tecnología de los agentes inteligentes se ha puesto al servicio de las compañías de seguros. El sistema multiagente desarrollado, llamado KIR, consiste en una red de agentes intercomunicados que funcionan dentro de cada compañía de seguros en el fondo de cada sistema y de esta manera pueden intercambiar valiosas informaciones que sirvan para el beneficio de cada uno de los participantes.

- **El uso de agentes inteligentes en Rockwell Automation para los sistemas de enfriamientos de agua**

Rockwell Automation ha trabajado mucho en el campo de los sistemas multiagentes y de esta manera ha llegado a desarrollar una aplicación inteligente que se utiliza en los barcos de la marina de los Estados Unidos para controlar y gestionar el enfriamiento del agua.

Los agentes desarrollados trabajan mucho con una base lógica y llevan a cabo negociaciones y tomas de decisiones colectivas apuntadas a poder realizar reconfiguraciones dinámicas de los distintos sistemas de enfriamiento.

5.24 – Crecimiento académico

Otra prueba del enorme crecimiento de los Agentes Inteligentes y los Sistemas Multiagentes se pueden encontrar en los marcos académicos más destacados a nivel mundial.

Ya es algo común que en todas las universidades del mundo, dentro del área de tecnología se encuentren trabajos, talleres y materias dedicados exclusivamente a estos temas. De hecho, son los profesores de estas materias, los principales referentes mundiales del desarrollo de la tecnología, quienes impulsan a sus alumnos a participar de este nuevo mundo.

Gracias a la existencia de internet y a las vinculaciones inter universitarias es muy común encontrar que las principales investigaciones y los más destacados desarrollos, sin dejar de lados a las más importantes publicaciones, se encuentran realizadas por profesores de distintas universidades que se encuentran para trabajar y hacer crecer este nuevo marco.

La madurez de estas conexiones académicas ha llegado ya a un punto tan fuerte a nivel mundial que podemos encontrarnos con eventos como el “Fifth International Joint Conference of Autonomous Agents

and Multiagent Systems” realizado en Japón durante Mayo del 2006.

Esta conferencia mundial surgió en 2002, como fruto de la unión de tres grandes eventos:

- AGENTS (International Conference on Autonomous Agents)
- ICMAS (International Conference on Multi-Agent Systems)
- ATAL (International Workshop on Agent Theories, Architectures and Languages)

El tema de los agentes inteligentes y los sistemas multiagentes no sólo ocupa lugar en las conferencias orientadas exclusivamente a esta tecnología, sino que se puede encontrar que día a día ocupan más lugar y tienen gran importancia en otros grandes eventos como el “International Joint Conference on Artificial Intelligence” que en su edición 2006 en la India tuvo todo un protagonismo especial reservado para la nueva tecnología.

Así, para demostrar el nivel mundial que ocupa esta tecnología, se encuentra la AgentLink, una agencia europea creada con el único fin de incentivar la investigación y desarrollo de toda tecnología basada en agentes inteligentes.

Esta agencia realiza cada año su gran evento llamado ATC (Agent Technology Conference). La última edición (2005) se realizó en Estocolmo y contó con la presencia de los más destacados científicos del campo.

Además del encuentro anual, AgentLink organiza cada año una escuela de verano en distintas ciudades de Europa con el fin de poder expandir los conocimientos sobre el tema en los estudiantes universitarios de todo el continente.

Esta gran apuesta, demuestra como Europa encuentra en la tecnología de los agentes inteligentes el futuro inmediato de la inteligencia artificial y apuesta desde ya con todas sus fuerzas para no quedar atrás en la carrera por la investigación y desarrollo de la tecnología.

Una última gran actividad que realizó AgentLink fue el “Third AgentLink III Technical Forum” que se llevó a cabo en Budapest, Hungría en Septiembre de 2005. Se eligió este lugar para poder llevar a cabo casi en simultaneo el “4th International Central and Eastern European Conference on Multi-Agent Systems”, otro de los más grandes eventos del tema a nivel mundial.

La tecnología de los Agentes Inteligentes y los Sistemas Multiagentes está creciendo día a día. Esto lo saben los científicos, los políticos y los empresarios, por eso no es una casualidad que año tras año se realicen más eventos, participen más personas y se invierta más dinero.

En efecto, esta tecnología ha dejado de ser el futuro y ha pasado a ser el presente de la Inteligencia Artificial a nivel mundial y por supuesto deseamos que tenga un buen desarrollo también en nuestro país.

6 - AI4U – Inteligencia Artificial para todos

Tal como ya lo señalamos, el objetivo de este trabajo es demostrar que la inteligencia artificial ha llegado a un nivel de madurez tal que le permitiría crear un sistema utilizando la tecnología de los sistemas multiagentes que pueda ser de uso masivo, sencillo y personalizable. Estos sistemas además podrían ser generados por los mismos usuarios mediante el uso de un sencillo asistente.

A continuación se va a detallar completamente la idea de un producto llamado “AI4U” (iniciales de Inteligencia Artificial para Usted en inglés), el cual se encuadra en el tipo de sistema descrito anteriormente.

Este producto podría ser adquirido por cualquier persona en el mundo y su objetivo sería el de proveer al usuario de un sistema multiagente que cumpla en satisfacer exactamente sus necesidades a la hora de realizar sus trabajos. Cumpliendo con los principios de la tecnología aplicada en esta clase de sistemas, el mismo podrá adaptarse y aprender del usuario para ir personalizándose a medida que busca cumplir sus objetivos.

Así, aunque su estructura y esencia será la de un sistema multiagente, para el usuario final esta organización será invisible, por lo que para él este sistema será interpretado como un “asistente inteligente”.

Cabe aclarar que aunque a lo largo de la descripción de este trabajo llamaremos numerosas veces ‘asistente’ al sistema multiagente creado, se utilizará este término sólo con el fin de acompañar la imagen que tendrá el usuario del sistema al utilizarlo y sentirlo como tal. Pero la realidad es que en muchos casos puede estar incluso muy lejos de ser un asistente, como por ejemplo, el caso del sistema multiagente para ventas que será descrito más adelante en el punto 6.6.3.

De modo que para estructurar de alguna manera el diseño del sistema multiagente desde el punto de vista de las necesidades del usuario, se definirán para cada sistema multiagente, posibles funciones que pueda llegar a realizar. Cada una de estas funciones será llevada a cabo por un agente inteligente diseñado específicamente para esto.

Todas las funciones serán seleccionables y de esta manera el usuario podrá elegir las que realmente desea que formen parte del sistema. Existirán dos tipos de funciones, las globales que pueden ser incorporadas a cualquiera de los posibles sistemas multiagentes y las funciones particulares que forman parte sólo del sistema multiagente correspondiente según su funcionalidad.

Para que se cumplan los puntos de sencillez y el alcance masivo del producto, creemos que es necesario facilitarle a cualquier tipo de usuario el proceso de creación del sistema deseado. Para esto, la creación del sistema multiagente se realizará por medio de un asistente, el cual será simple, claro y de fácil manejo.

Ya hace varios años, los principales productos de software de venta masiva cuentan con distintos asistentes dentro de su funcionamiento que llevan incluso a los usuarios principiantes a poder realizar las tareas que requieren, sin tener que contar con el conocimiento avanzado solicitado en cada caso.

De esta manera, mediante el uso de varios pasos en los cuales el usuario simplemente deberá seleccionar todas las características que tendrá su nuevo sistema multiagente. Al finalizar, automáticamente se creará el asistente que eligió y de la manera seleccionada. Por ejemplo, de haber seleccionado el reconocimiento de voz, el usuario podrá comunicarse a través de ese medio con el asistente como si éste fuera un ser humano. Esto es muy importante para reforzar el concepto de masividad ya que así se conseguirá que el asistente sea utilizado por cualquier tipo de persona, incluso un analfabeto.

La arquitectura del sistema multiusuario generado a medida del usuario, será detallada mas adelante, pero es muy importante destacar que aunque existirá un agente por cada función seleccionada y varios agentes para obtener el funcionamiento global del sistema, ante los ojos del usuario, el sistema será visto como un todo y no reconocerá sus distintas partes ya que éstas estarán totalmente coordinadas entre ellas para comportarse de esta manera.

Por otra parte, al ser muy variado el uso del sistema multiagente que se ha creado, el entorno en el que funcionará también lo será. Es por esto que al finalizar el asistente, el usuario deberá elegir de qué manera quiere “exportar” al sistema creado, lo cual implica que si eligió crear un sistema multiagente que le ayude proveyendo de soporte técnico a clientes, cuenta con la posibilidad de exportarlo para que funcione a través de algún servicio de mensajería instantánea como el Microsoft Messenger y lo utilice para realizar las comunicaciones con los clientes.

6.1 – ¿Por qué un Sistema Multiagente?

Al imaginarnos el funcionamiento interno de AI4U seguramente existirán muchas ideas distintas sobre la estructura del mismo, pero hay ciertos puntos³⁰ que determinan por qué una solución basada en un

30. M. Wooldridge, *An Introduction to Multiagent Systems*, J. Wiley & Son., 2002.

sistema multiagente es la ideal para este producto. Algunas de éstas, que valen la pena destacar, son:

- El ambiente que existe dentro de todo el diseño del sistema es abierto, complejo y altamente dinámico. En este tipo de ambiente, los sistemas capaces de acciones autónomas flexibles son la gran solución.
- Los agentes son una metáfora de la función del sistema. Al imaginarnos el funcionamiento del sistema vemos que existen distintas acciones a llevar a cabo de manera independiente. Esta metáfora representa claramente a los agentes inteligentes. Maes³¹ entiende a los agentes como “asistentes expertos” cooperando con el usuario en la solución de problemas.
- La idea cuenta con distribución de información, control y experiencia. Así se demuestra como una concepción centralizada complicaría todo el diseño y funcionamiento. La solución ideal sería la que cuenta con bases de datos semi-autónomas, como las de los sistemas multiagentes.

6.2 - Asistente

El asistente, por definición del proyecto, debe poder ser comprendido por cualquier clase de usuario y su objetivo final es la creación de un sistema multiagente acorde a las propiedades seleccionadas por el usuario.

Este asistente deberá contar con ocho pantallas que llevarán de una manera clara al usuario a lo largo de los distintos pasos necesarios para la creación del sistema deseado. Los ocho pasos que deberán integrarse al asistente serán:

1. **Bienvenida:** En esta primera pantalla se le da la bienvenida al usuario y se le explica brevemente el objetivo del asistente. El usuario deberá tener claro cuál es el objetivo del asistente para así responder correctamente cada uno de los pasos y terminar creando exactamente el sistema que necesita.
2. **Selección de tipo de sistema multiagente:** Aquí el usuario seleccionará qué tipo de sistema desea crear. El usuario contará con una breve explicación de cada uno de los tipos de sistemas disponibles en donde se expondrá el perfil con el que contaría cada una de las opciones.
3. **Selección de funciones particulares:** De acuerdo al tipo de sistema multiagente seleccionado en la pantalla anterior, aquí se presentará un listado de todas las funciones particulares de esa clase de sistema multiagente. Cada una de estas funciones contará con una breve descripción en la que explicará sus principales características para que sea posible entender su funcionamiento. De esta manera, el usuario podrá seleccionar exactamente las funciones que le interesen para el sistema multiagente que está por crear.
4. **Selección de funciones globales:** En esta pantalla el usuario encontrará el listado de todas las funciones globales disponibles para los sistemas multiagentes. De cada una de ellas podrá obtener un breve resumen que explique sus características y así el usuario seleccionará qué funciones globales tendrá su sistema.
5. **Opción de reconocimiento de voz:** Si el usuario quiere que el sistema multiagente creado cuente con la capacidad de reconocimiento de voz, en esta pantalla podrá activar esta opción. De hacerlo, se habilitarán otras opciones relacionadas con el tema como la opción de seleccionar con qué nombre se desea llamar al sistema por medio de la voz.
6. **Opción del habla:** Si el usuario quiere que el sistema se comunique con él mediante el uso del habla, en esta pantalla podrá activar esta opción. De hacerlo, se habilitarán todas las configuraciones relacionadas con el tema y el usuario podrá seleccionar el tipo de voz que quiera para su sistema. Las opciones estarán ligadas a nombres de personas para facilitar la vinculación de las características de voz para el usuario. Además, de cada tipo de voz podrá escuchar una muestra para facilitar la elección con solo hacer click sobre las opciones.
7. **Opción de animación:** Si el usuario quiere que el sistema cuente con una imagen animada visible, puede agregársela en esta pantalla. Al hacerlo, se habilitarán todas las opciones correspondientes a la animación y al igual que en el caso de la voz, las opciones de animación se encontrarán asociadas a un personaje, por lo que el usuario deberá elegir uno del listado para que interprete a su sistema multiagente. Para ir seleccionando de la lista a los distintos personajes, se podrá ver la imagen de la animación y una información simplemente ilustrativa del personaje.
8. **Exportación:** Al llegar aquí, el usuario ya habrá terminado de definir todas las características del sistema multiagente que desea crear. El objetivo de esta pantalla es que el usuario seleccione la forma en la que quiere que el sistema sea creado. Como se explicará mas adelante, el usuario puede poner el sistema creado en un entorno web o si lo elige, diseñándolo para funcionar sobre los sistemas de mensajería instantánea.

31. Maes P., “Agents that reduce work and information overload”, Communication of the ACM, 1994.

9. Finalización: Esta pantalla informará al usuario que el asistente ha finalizado y el programa está listo para crear el sistema multiagente deseado. Al usuario sólo le restará hacer click en el botón "Crear" para que el sistema multiagente empiece a ser creado. Durante este proceso se visualizará una barra de estado que mostrará el nivel de avance de la creación del sistema. Al finalizar se presentará un mensaje indicando esto y agradeciendo al usuario por haber elegido AI4U.

6.3 - FIPA

Como se dijo anteriormente, FIPA es parte de las organizaciones que forman parte de la IEEE y es la encargada de establecer los estándares de la tecnología que se aplicará en este caso, por lo que este proyecto debe sumarse a esta regulación que seguramente es la base de todo el crecimiento que vendrá en un futuro muy cercano en el área. Por todo esto, en el diseño de los sistemas multiagentes generados por el asistente se utilizarán todas las normativas establecidas por FIPA.

Otro punto importante, que justamente es una de las principales motivaciones de la existencia de FIPA, es que la utilización de los estándares señalados en los documentos de FIPA permitirán a los agentes que forman parte de los sistemas creados poder comunicarse con agentes externos al sistema y realizar el intercambio de información que sea necesario en cada caso.

Aunque recién nos encontramos en los comienzos de la aplicación de las normativas de FIPA, aplicarlo en el diseño de los sistemas creados le darán al producto una visión de futuro y los agentes creados estarán listos para cuando, en muy poco tiempo, el mundo entero se encuentre lleno de agentes inteligentes funcionando bajo las normas FIPA.

6.4 - Comunicación con el usuario

Para que el sistema multiagente generado cumpla con la accesibilidad masiva, que se busca obtener con el diseño de este sistema, el usuario debe contar con la posibilidad de incorporarle al sistema creado la propiedad de comunicarse mediante el uso de la voz y el reconocimiento de la misma.

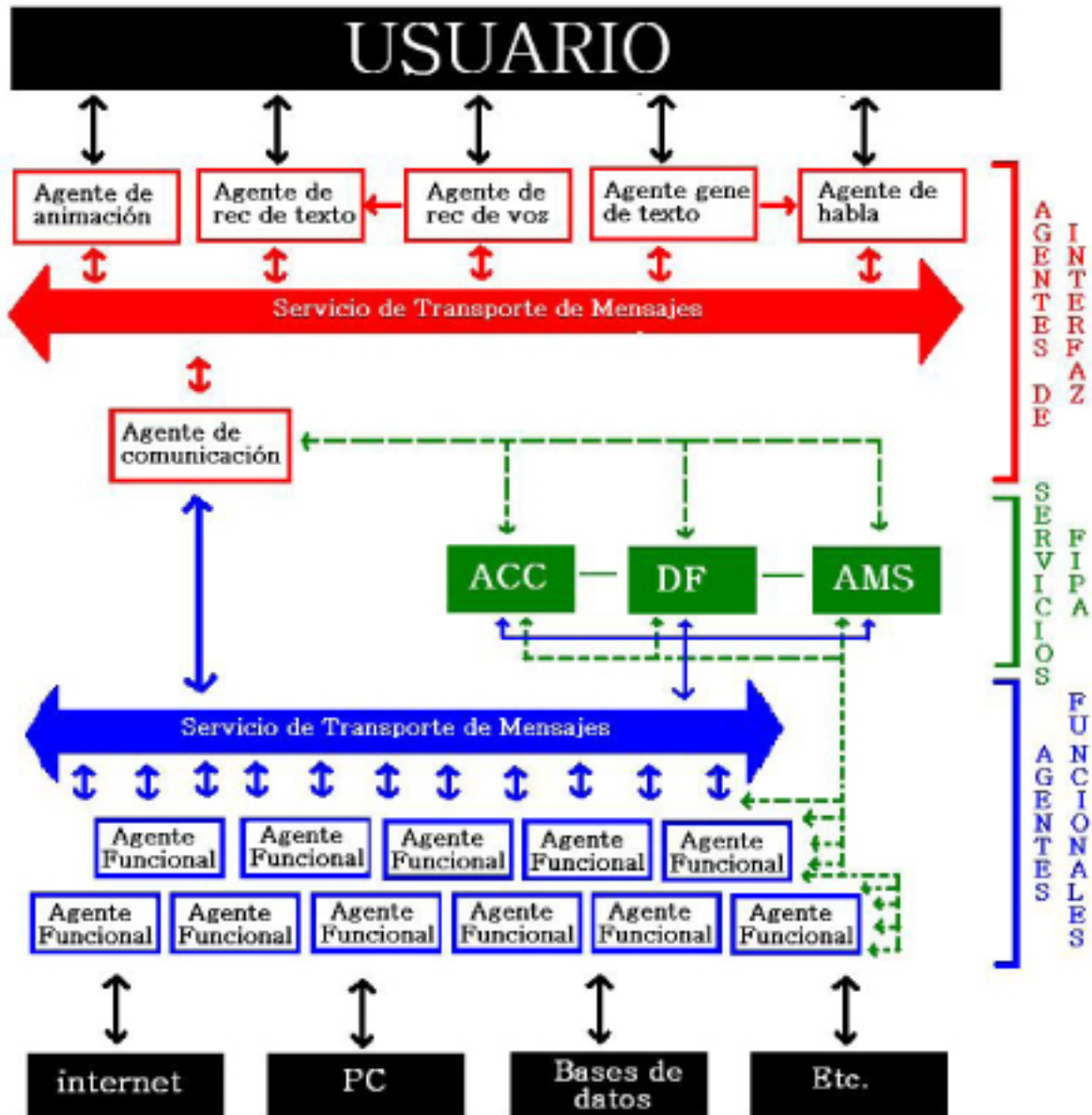
Otra opción que fortalece la accesibilidad del sistema por parte de los usuarios, es la capacidad de dotar al sistema con una animación que represente su funcionamiento y haga al mismo más amigable y ameno. Esta visualización física del sistema puede fortalecer los lazos entre el sistema y los usuarios menos capacitados como los analfabetos y menores de edad entre otros.

Así, el tipo de presencia física del sistema deberá ser seleccionado por el usuario en el asistente, ya que es una característica más de la personalización del mismo. Claramente, las animaciones que genere el sistema deberán estar acordes a lo que éste está diciendo o haciendo, para acompañar el significado de sus palabras en la interpretación que realizará el usuario.

6.5 - Arquitectura

Siguiendo la filosofía de adhesión a los estándares de FIPA, la arquitectura de los sistemas multiagentes creados se encontrarán basadas en el modelo de referencia de agentes definido por FIPA. De esta manera se encontrarán los servicios de gestión de agentes, canal de comunicación, directorio de servicios y sistema encargado de transporte de mensajes dentro de cada sistema multiagente creado.

Realizando una visión global de la arquitectura como se efectúa en la siguiente figura, se pueden distinguir tres áreas dentro de la estructura. Por un lado encontramos la que forman los agentes de interfaz, todos estos los encargados de mantener la comunicación con el usuario. Luego tendremos a los servicios descritos dentro de el modelo de referencia de FIPA y por último a los agentes funcionales que cumplirán con las distintas funciones definidas durante el asistente.



Arquitectura de los sistemas multiagentes creados. En rojo se encuentra todo lo que forma parte de los Agentes de Interfaz y las líneas rojas reflejan sus comunicaciones. Lo que se encuentra en verde forma parte de los servicios FIPA y con líneas verdes entrecortadas se muestran como todos los agentes tienen accesos directos a estos servicios. Por último, en azul se encuentran todos los componentes de los agentes funcionales y las líneas azules representan sus comunicaciones.

Como se puede ver en el gráfico, el único servicio que utilizan los agentes de interfaz es el uso del sistema encargado de transporte de mensajes, el resto de la información necesaria para mantener la comunicación entre ellos es conocida desde un principio ya que las mismas son fijas.

Por otro lado podemos ver cómo los agentes funcionales mantienen sus comunicaciones internas mediante el uso de los servicios establecidos en el modelo de FIPA y también puede mantener sus accesos externos con internet, la computadoras, base de datos y cualquier otro recurso que le sea importante para cumplir sus funciones y buscar su objetivo.

Una pieza clave en esta estructura es el agente de comunicación, el cual sirve como nexo entre todos los agentes funcionales y los de interfaz. Al ser el único agente de interfaz con acceso a los servicios definidos por FIPA, es el único agente de interfaz al que los agentes funcionales tienen acceso.

De la misma manera, este agente es el coordinador de toda la comunicación con el usuario por lo que, como se puede ver en el esquema, la información generada por algún agente funcional para el usuario, pasará primero por el agente de comunicación y éste decidirá si se le comunicará al usuario y además de qué manera, o sea, qué agentes de interfaz participarán de esta comunicación.

6.6 - Tipos de sistemas multiagente

Teniendo en cuenta que la orientación de este sistema es para los usuarios comunes, tanto en casas como en oficinas, sería necesario consultarles a ellos mismos cuáles son sus necesidades, y de acuerdo a esto definir los tipos de sistemas multiagentes que se puede crear y cuáles deberían ser las funciones específicas de cada sistema.

Para poder realizar esta consulta se llevó a cabo un trabajo de encuesta para identificar qué tipo de sistemas y funciones necesitan los usuarios. La encuesta fue realizada sobre 50 personas de distintas edades seleccionadas al azar pero cubriendo la siguiente distribución: 25% son estudiantes, 25% son académicos, 25% son empresarios y 25% son profesionales, todos Argentinos, residentes en el país.

Cabe destacar que la encuesta fue realizada puramente con el fin de buscar ideas de distintos tipos de usuarios para llevar a cabo este diseño de AI4U, contando con funciones que realmente son requeridas por los usuarios. A pesar de esto, hay que tener muy presente que, en caso de llevar a cabo el desarrollo real del sistema, se debería hacer un estudio mucho más complejo y profundo para poder determinar fehacientemente cuales son las funciones que deberían encontrarse dentro de un sistema multiagente generado por AI4U.

Para que la información obtenida de los resultados de la encuesta sea productiva y se puedan definir los requisitos reales de los usuarios, la encuesta utilizada ha sido corta y sobre todo puntual en sus preguntas, tal como se puede ver en el modelo presentado a continuación:

ENCUESTA

Nombre:

Sexo:

Edad:

1 – Si pudiera tener en su computadora un programa inteligente con el que pueda hablar como si fuera una persona, ¿en qué tareas de su vida le gustaría que le ayude?

2 – ¿Que funciones le gustaría que haga?

3 – Fuera del contexto de la tarea que ha elegido, ¿que otras cosas le gustaría que haga?

4 – ¿Qué característica considera imprescindible para este tipo de programa?

5 – De existir y tener un precio accesible, ¿compraría este programa?

Encuesta utilizada

Una vez obtenido los resultados de todas las encuestas, se los ha computado y analizado cuantitativamente. El resultado de este trabajo demostró que con el diseño de seis sistemas multiagentes distintos se pueden capitalizar en seis grupos distintos los requisitos de los usuarios.

La última pregunta se diferencia del resto ya que no busca identificar posibles diseños de sistemas multiagentes para los usuarios, sino saber si éstos accederían a comprar el producto. El resultado fue más que contundente: el 100% de los encuestados respondieron de manera afirmativa.

Teniendo en cuenta los resultados de las encuestas, en AI4U existen seis tipos de sistemas multiagentes que el usuario puede seleccionar. Cada uno de ellos cuenta no sólo con sus funciones características, sino que el diseño general del sistema se encuentra orientado para cumplir su perfil.

Por esto, más allá de que las funciones seleccionadas le den una orientación al sistema multiagente por las características de estas mismas, el sistema en sí, al ser seleccionado en un principio, cuenta con un diseño y funcionamiento exclusivamente orientado a su definición.

Para entender lo que decimos de una manera mas clara imaginemos un ejemplo.

Un sistema multiagente forense, no sólo se comportaría como un investigador por contar con las funciones “análisis”, “rayos X” u otras, sino que todo el sistema en sí cuenta con un perfil y una orientación global en su ser.

Así, para que esto sea posible, al crear el sistema multiagente de acuerdo a la categoría seleccionada por el usuario, el mismo contará con una política de funciones de utilidad que harán funcionar globalmente al sistema con el correspondiente perfil. De esta manera, un sistema multiagente creado para ayudar en una investigación encontrará una utilidad mayor en cuestiones afines a esta característica durante su funcionamiento, de la misma manera que lo hace un ser humano con una orientación profesional.

Volviendo a los resultados obtenidos en la encuesta, los seis sistemas multiagentes definidos según los requerimientos de los usuarios que busquen dar soporte en distintas áreas son:

- **Sistema multiagente que ayude al estudiante**
- **Sistema multiagente vendedor**
- **Sistema multiagente que ayude en el trabajo**
- **Sistema multiagente para ayuda en investigaciones**
- **Sistema multiagente para soporte técnico y asistencia al cliente**

6.6.1 - Funciones globales

Existen funciones que se encontrarán disponibles de incorporar en cualquiera de los distintos sistemas multiagentes y podrán ser seleccionadas en la pantalla correspondiente del asistente. Cabe destacar que, como se indicó anteriormente, la existencia de cada una de estas funciones indican la de un agente inteligente que es el responsable de llevarlas a cabo. Estos agentes, son identificados dentro de la estructura definida de los sistemas multiagentes como agentes funcionales.

Algunos de los agentes que podemos implementar en cualquier sistema multiagente seleccionado en el asistente son:

- **Agente de administración de correo electrónico**

El objetivo de este agente será la administración de las cuentas de correo electrónico del usuario. Esto implica que el agente ayuda al usuario a gestionar todos sus mails, tanto antes de “bajarlos” como cuando ya se encuentran en el programa de correo electrónico. También tendrá soporte para las cuentas de webmail del usuario. Algunas de sus funciones serán la de identificar cuáles son los mails importantes y de ser necesario notificará al usuario, también ayudará a encontrar mails guardados, a gestionarlos y organizarlos según las especificaciones y gustos del usuario, responder mails automáticamente, etc.

- **Agente de ayuda bancaria**

Este agente tendrá el objetivo de dar un soporte bancario simple al usuario. Su nivel de manejo es orientado a la administración bancaria de una persona común y no tiene la complejidad del agente bancario que forma parte de los sistemas multiagentes que ayuda en el trabajo. Dentro de sus funciones se encontrará ayudar a la administración de cuentas bancarias del usuario, recordar el pago de cheques adelantados, informar rápidamente los saldos y movimientos en las distintas cuentas bancarias (permitiendo al usuario ahorrar el tiempo que toma visitar uno a uno los distintos sistemas de home banking), informar sobre cotizaciones de monedas y acciones de solicitarlo el usuario, etc.

- **Agente recordatorio de fechas personales**

El objetivo de este agente, tal como lo indica su nombre, será recordarle las fechas al usuario. De esta manera no olvidará los cumpleaños de amigos, familiares, compañeros de trabajo, además de eventos sociales como casamientos, funciones de algún espectáculo, etc. y contará con opciones como las de recordar qué se regaló en ocasiones anteriores y cuanto se gastó.

- **Agente de administración multimedia**

Hoy en día encontramos en cualquier computadora fotos digitales, canciones e incluso películas completas pero es necesario mantener un inventario de todo los archivos multimedia que se encuentran en la pc. Este agente ayudará rápidamente a encontrar algún archivo y los organizará según el criterio que decida el usuario tal como organizar las fotos por fecha o la música de acuerdo a un determinado compositor o estilo musical.

- **Agente de soporte técnico personal**

Como estos sistemas multiagentes se encuentran orientados a cualquier tipo de usuario, encontraremos que la mayoría de éstos cuentan con un nivel de conocimiento técnico muy bajo en todo lo relacionado a una computadora. Por este motivo, el objetivo de este agente será el de ayudar a estos usuarios a sobrellevar distintos problemas técnicos que le ocurran a la pc.

En base a las soluciones que contará, podrá ir sumando, como se encuentra en su esencia de aprendizaje, los conocimientos de acuerdo a la experiencia que vaya teniendo con el usuario. Por ejemplo, si

al usuario le pasa repetidamente un problema en Word y el agente encontró una solución, la próxima vez evitará hacer tantas preguntas y supondrá que debe tratarse del error usual. Obviamente, en caso que no lo sea, deberá intentar otra solución, incluso fuera del conocimiento ya adquirido, tal como buscar en internet.

- **Agente de juegos**

Este agente contará con una variada lista de juegos simples para jugar con el usuario. De esta manera, el sistema también será útil en los momentos de ocio. Entre los juegos propuestos se encuentran las damas, el ajedrez, ahorcado, reversi, etc. y el sistema contará además con distintos niveles para que el usuario pueda ir avanzando a medida que se adquiera mayor experiencia en los juegos.

- **Agente de traducción**

Este agente contará con la capacidad de realizar traducciones de textos entre distintos idiomas. De esta manera podrá ayudar al usuario a navegar por páginas de internet, leer textos, escribir mails, chatear, etc. en otros idiomas y perfeccionarse profesionalmente.

- **Agente diccionario**

Este agente brindará al usuario un servicio de diccionario sobre el idioma que solicite. Además contará con la posibilidad de proporcionarle al usuario sinónimos o antónimos. Incluso podrá conectarse y realizar búsquedas mas profundas en sitios como el de la Real Academia Española y por último, contará con la posibilidad de realizar corrección ortográfica en cualquier texto solicitado.

- **Agente de búsqueda virtual**

Buscar información en internet se convirtió en una tarea diaria para casi todos los usuarios de la red pero se pierde mucho tiempo todos los días buscando en distintos motores de búsquedas hasta encontrar lo que se quiere. Este agente no sólo buscará textos, imágenes, videos, etc. según los criterios de búsqueda en muchos buscadores, sino también, de acuerdo a los gustos, los calificará e intentará siempre encontrar exactamente lo que quiere el usuario.

- **Agente de acceso rápido**

La función de este agente será la de localizar e identificar cuales son las carpetas y los programas que utiliza el usuario para ciertos trabajos, por lo que cuando éste se disponga a realizarlo, el agente le proveerá de un rápido acceso. Por ejemplo, si el usuario necesita hacer una animación en flash, el agente rápidamente le podrá abrir el programa y las carpetas donde se guardan los sonidos, imágenes y otros archivos para trabajar. De la misma manera, si se está trabajando con un archivo determinado, el agente podrá poner a disposición rápida otras carpetas relacionadas.

6.6.2 - Sistema multiagente que ayude al estudiante

Este sistema multiagente estará pensado para ayudar al estudiante en sus estudios y podrá darle tanto el soporte estratégico como el de contenido en las diferentes tareas que se tengan que llevar a cabo.

Algunos de los agentes particulares de este sistema multiagente serán:

- **Agente de tareas**

Este agente se especializará en las tareas y trabajos del usuario. No sólo le recordará de las tareas que tiene pendiente hacer (tipo de trabajo, descripción, materia, fecha y hora de entrega, si es en grupo con quienes, etc.), sino que también le ayudará a mantener un control del progreso de las mismas. Además será útil para buscar material puntual para resolver los ejercicios de las tareas.

- **Agente de materias**

Este agente, al contar con los programas curriculares, podrá ir guiando al usuario a lo largo de determinada materia anticipándole el material a estudiar, así llegado el momento, este agente ya tendrá reunido direcciones de internet, apuntes, artículos y otro material sobre el nuevo tema.

- **Agente de resumen**

Este agente es clave ya que su función será la de crear resúmenes en forma automática de acuerdo a especificaciones y gustos del usuario. Para crear estos apuntes, puede comunicarse con los agentes de apuntes y de materias y ayudarse a determinar qué es lo más importante y qué no y así optimizar la calidad de los textos creados.

- **Agente de apuntes**

Este agente no sólo le ayudará al agente a administrar sus apuntes sino que contará con funciones como por ejemplo armar una selección de acuerdo a un apunte y a todo el material de la materia. Además podrá procesar distintos apuntes y crear uno más completo. Todos los apuntes creados serán en relación a los gustos del usuario.

- **Agente sintetizador**

Este agente contará con una función particular y a la vez muy práctica, la de crear archivos de audio con el contenido de estudio. De esta manera se podrá pasar de texto a mp3 todo el capítulo de un libro,

por ejemplo. Podrá generar un archivo de audio con el resumen de los temas que entren en un parcial para que luego el usuario los cargue en su reproductor mp3 y los utilice para estudiar. Este agente es además ideal tanto para minusválidos como para ciegos.

- **Agente bibliográfico**

Cuando uno estudia, es muy común que necesite buscar en internet más bibliografía de un determinado tema para poder apoyar el material de la materia. Este agente se dedicará a buscar material específico de estudio al usuario en internet. De esta manera si el usuario tiene duda sobre un tema, le puede consultar al sistema y es aquí donde este agente le sugerirá ciertos links para que visite y se ayude.

- **Agente de evolución académica**

Su objetivo será el de mantener la evolución académica del usuario. Para esto, informará al usuario sobre las materias aprobadas, las que faltan, las notas, las características principales de las materias, y también la dificultad, correlatividades, etc. Toda esta información podrá ser devuelta a pedido del usuario o el agente podría generar pantallas donde se muestre toda la información ordenada.

- **Agente de exámenes**

Su función será la de ayudar al usuario antes de ir a un examen. Para esto, sólo necesitará saber las preguntas que se dieron en los distintos exámenes anteriores y le pedirá al usuario que indique un puntaje de dificultad a cada pregunta. De esta manera el agente buscará analizar la tendencia y el patrón que existe en los exámenes ya que es muy común que se repitan preguntas y que algunas fechas de exámenes sean mas difíciles que otras. Por todo esto, buscará indicarle al usuario qué preguntas le pueden tomar y que fecha de examen le conviene presentarse.

- **Agente administrador de bibliografía**

Ocurre en muchas oportunidades que en distintas materias se repite alguna bibliografía y es aquí donde este agente aparecerá para indicarle al usuario si cuenta con el material y de ser así, dónde lo tiene. Para esto, el agente tendrá un registro de todo el material que utiliza el usuario para las distintas materias y así podrá ayudarle al usuario a reutilizar material y a encontrarlo rápidamente.

- **Agente de compañeros**

En algunas universidades es posible averiguar si un estudiante está cursando una materia o no porque las notas de cada estudiante son públicas. Toda esta información le será útil a este agente. Su función principal será la de mantener una especie de base de datos acerca de los compañeros de estudio. De esta manera, sabiendo qué materias cursa y cursó cada compañero, sus notas en exámenes, referencias personales del usuario, etc. se puede saber si es conveniente buscarlo para formar parte de un trabajo en grupo, pedirles apuntes o no, asociarse para implementar una idea, etc.

6.6.3 - Sistema multiagente vendedor

Este sistema tendrá un objetivo funcional global muy claro, el de vender. Le podrá ser muy útil al usuario, en caso de exportarlo y ponerlo en una página web, y también si la idea es vender distintos productos a través de la mensajería instantánea.

Para imaginarlo mejor, un ejemplo sería el sistema multiagente ideado para un videoclub que se comunicara con los clientes a través de la mensajería instantánea y a través de ésta alquilar películas, informar los estrenos según los gustos del cliente, recordar una película que no devolvió, etc. (este ejemplo forma parte del prototipo no funcional)

Algunos de los agentes particulares de este sistema multiagente serán:

- **Agente de productos y servicios**

Será el encargado de almacenar todo el conocimiento acerca de los productos o servicios que el usuario desea vender. Además de aprender el nombre de éstos, deberá aprender su descripción, características, ventajas, desventajas y las respectivas respuestas a las consultas. En caso de cambios entre los productos y servicios, sólo hará falta que el agente se entere de las modificaciones y así se actualizará la información. Esta centralización de información será clave en el sistema multiagente ya que los otros agentes que lo constituyen le consultarán por cualquier información relacionada con los productos y servicios.

- **Agente de clientes**

Este agente podrá tener identificado a los clientes, almacenando no sólo toda la información acerca de ellos sino también las características de sus operaciones de compras. Así podrá saber cuándo a un cliente le interesa cierto tipo de productos, si tiene un precio diferencial, si debe dinero y no se le puede vender hasta que cancele deudas, etc. Otra información que deberá conocer este agente es cualquier particularidad del cliente en lo que se refiere a los gustos en la comunicación. Obviamente, el usuario tendrá acceso a cambios sobre la información de algún cliente y por ejemplo poder establecer restricciones.

- **Agente de negociación**

En toda operación de venta existe una negociación entre el cliente y el vendedor. En algunos casos

con una simple solicitud alcanza pero en los otros casos que son la mayoría, la habilidad de la negociación en la venta, es la que separa a un buen vendedor de uno malo. Es aquí donde este agente aparecerá ya que deberá ir aprendiendo cada vez más sobre técnicas de negociación en las ventas que haga el sistema, basándose en la experiencia, en la información que pueda recibir de otros agentes como el agente de clientes, o de acuerdo a las características del producto que puede obtener del agente de productos, etc.

- **Agente de marketing**

Tendrá la función de realizar todo lo que se refiera a marketing de los productos y servicios que se intentan vender. Así, será muy útil en los sistemas exportados a mensajería instantánea donde, en base a los gustos de los clientes, sabrá cuando comunicarle sobre una baja de precio o sobre un nuevo producto, como en el ejemplo anterior de un sistema multiagente para un videoclub.

6.6.4 - Sistema multiagente que ayude en el trabajo

Este sistema multiagente se encontrará orientado para ayudar a los usuarios en sus trabajos. De esta manera contará con funciones que ayuden al manejo tanto de una PyME como a un trabajador que siga estos procedimientos.

Algunos de los agentes particulares de este sistema multiagente serán:

- **Agente agenda**

Tendrá la capacidad de almacenar toda la información de clientes, proveedores o contactos que tenga el usuario. Esta información podrá ser provista tanto por el usuario como extraída de alguna otra fuente. De esta manera, al solicitarle información sobre un contacto, el agente podrá contar no sólo con toda la información personal sino también información particular a cada caso como deuda, pedidos pendientes, plazos y formas de pago, memos guardados, etc. Así se le otorgará al usuario un acceso rápido de información valiosa. Un ejemplo para comprender su utilidad se puede ver si llama por teléfono un cliente, ya que se le podrá pedir al sistema que le diga cuanto debe el cliente y al instante se contará con la respuesta.

- **Agente de comunicación**

Este agente, mediante el uso del agente de agenda podrá comunicar al usuario con el contacto, ya sea preparando un mail o marcándole el teléfono al usuario y avisarle cuando le han contestado. Además, de estar conectada la computadora al teléfono (o al celular por bluetooth, por ejemplo) y de serle solicitado al sistema, al recibir una llamada el agente puede identificar quien llama al usuario y presentarle automáticamente la información en pantalla del contacto.

- **Agente de recordatorios empresariales**

Este agente tendrá las funciones de recordarle al usuario todo lo que éste le solicite, como vencimientos de pagos, recordatorio de envió de muestras y/o de prueba de muestras de proveedores, etc. Además de solicitudes puntuales, el agente deberá aprender sobre ciertos trámites que se deben hacer de forma periódica, como la presentación de una declaración jurada semestral, y poder mantener al usuario al tanto de esto. Además de recordar podrá indicarle también al usuario donde deberá hacerlo, horario, contacto, etc. y de ser indicado por el usuario, el agente deberá encontrar la manera de contactarlo en caso que no se encuentre presente, como enviándole un mail o tal vez un mensaje de texto al celular.

- **Agente de administración de documentos**

Ayudará a la administración de todos los documentos laborales que tiene el usuario, de esta manera, rápidamente el usuario podrá encontrar documentos, ya sea por fecha, nombre, tema, etc. El agente también podrá identificar y, luego de la aprobación del usuario, mover los documentos para que se encuentren mejor asociados y organizados. Además, al trabajar con un documento nuevo, el agente puede sugerirle al usuario donde guardarlo: por ejemplo, si se recibió por mail un nuevo manual sobre un sistema X, el mismo deberá ser guardado en la carpeta "x/manual", de ser ésta la estructura elegida.

- **Agente de procedimientos**

Este agente será muy importante para todo aquel que se maneje con guías de procedimientos en su trabajo ya que aprenderá y ayudará al usuario a poder mantener el control de su trabajo. También deberá entender las estructuras de los procedimientos e identificar cuándo se trata de tareas o grupos de subtareas, e ir chequeando si una tarea fue completada o en qué porcentaje se encuentra completada y qué falta realizar. Además, deberá acceder a todas las bases de datos con las que se cuente para poder buscar referencias del procedimiento actual en algunos anteriores, y así no sólo indicar si se va bien sino también presentarle al usuario si ya existe algún procedimiento similar para que lo pueda utilizar como referencia.

- **Agente de guardias**

Será ideal para la gente que trabaja con sistemas de guardias como en un hospital. No sólo recordará guardias activas o pasivas sino también toda la información que lo rodea, como cambios de guardias

posibles de acuerdo a los horarios de otras personas, memos guardados para las guardias, tareas programadas de guardias, etc.

- **Agente de trabajo en grupo**

En muchos casos se hacen trabajos en distintos grupos y recordar todos los avances, reportes, reuniones, papeles, etc. de cada grupo se convierte en un trabajo confuso. Para esto existirá este agente que contará con un soporte totalmente orientado a los trabajos en grupo guardando toda la información anteriormente mencionada, además de todo lo que el usuario le solicite en relación al tema.

6.6.5 - Sistema multiagente para ayuda en investigaciones

El objetivo de este sistema multiagente será el de dar soporte al usuario en todo lo que se refiera a un trabajo de investigación. Para esto, todo el sistema entenderá de forma global que se está trabajando sobre un tema y mantendrá una orientación integral para todas sus funciones.

Al trabajar todos los agentes sobre un mismo trabajo de investigación, el ingreso de información al sistema multiagente llevará a que muchos de los agentes que lo componen tengan la necesidad de acceder, por lo que en este sistema la cooperación será muy importante. Por ejemplo, la presentación de cierto prototipo en un congreso por parte del grupo de trabajo, llevará al agente de curriculum a hacer su registro y también al de informes el suyo, entre otras tareas.

Algunos de los agentes particulares de este sistema multiagente serán:

- **Agente de búsqueda de información**

Este agente ayudará a buscar información sobre el tema de investigación en distintas fuentes, además podrá mantener actualizado al usuario sobre fuentes puntuales. Por ejemplo, si el usuario trabaja en una investigación sobre las ranas, el agente podrá avisarle al usuario acerca de la aparición de artículos relacionados a las ranas en las revistas científicas que él le haya indicado. Además, constantemente buscará en internet la aparición de nuevos libros, artículos y revistas que puedan serle de interés al usuario para su trabajo de investigación.

- **Agente de curriculum**

Éste se especializará en la confección del curriculum vitae, una tarea por demás odiosa para el investigador. De esta manera no sólo ayudará al usuario a mantener actualizado su curriculum sino también podrá generar distintas versiones según la manera en la que sea necesario presentarlo. Además, será capaz de mantener la curricula de todo un equipo de investigación y actualizarlos según corresponda. Por ejemplo, si dos miembros fueron a hacer una presentación en un congreso, uno como director y otro como co-director, el agente automáticamente actualizará los curriculum de esos miembros como corresponda.

- **Agente de informes**

Todo trabajo de investigación lleva en paralelo un trabajo de informes sobre su desarrollo. Este agente, ayudará en la elaboración de los informes y las rendiciones de cuentas que sean necesarias presentar periódicamente. El usuario sólo necesitará indicarle al agente qué tipo de reportes necesitará realizar y qué elementos se deberán registrar y éste automáticamente podrá crearlos. Para esto además, podrá contar con reportes anteriores como referencias y al usuario como guía.

6.6.6 - Sistema multiagente para soporte técnico y asistencia al cliente

Este sistema multiagente se encontrará orientado a cubrir los servicios de asistencia al cliente y de soporte técnico de cualquier empresa. Al igual que el sistema multiagente para ventas, este sistema será útil tanto exportado para mensajería instantánea como para un sitio web.

Al referirnos al soporte técnico apuntamos a la solución de problemas técnicos en servicios y productos mientras que en la asistencia al cliente, se apunta más a aspectos generales de los servicios, productos o a información acerca de la empresa. Con este sistema multiagente se buscará una solución completa para los problemas o dudas de los clientes.

Algunos de los agentes particulares de este sistema multiagente serán:

- **Agente productos y servicios**

Contará con toda la información sobre los productos o servicios y permitirá responder a todas las preguntas de los clientes. Mas allá de conocer toda la información básica del producto o servicio, deberá ser un verdadero experto en ellos, ya que deberá poder responder cualquier tipo de preguntas técnicas de los usuarios.

- **Agente de soluciones avanzadas**

Éste será el encargado de poder resolver los problemas de los usuarios sobre los servicios o productos. Para esto, el agente deberá ser capaz de hacer un diagnóstico del problema con la menor cantidad de preguntas y luego ofrecer una solución al cliente. Además, este agente como tal, podrá aprender nuevos diagnósticos y nuevas soluciones e ir así perfeccionándose y mejorando su funcionamiento día a día.

- **Agente información general**

Será el responsable de poder responder a las preguntas generales que tenga el usuario, es decir, a pedidos de información sobre la empresa como dónde queda ubicada la fábrica, por ejemplo, sus dimensiones, complejidad, etc.

- **Agente de consultas**

Su función será la de recibir todas las quejas, consultas, sugerencias o mensajes que el usuario desea dejarle a la empresa. Deberá poder almacenar estos mensajes junto a toda la referencia que le sea útil al usuario, como la información personal de quien dejó el mensaje. Luego podrá clasificar los textos y presentárselos al usuario de la forma en que éste lo solicite.

6.7 - Comunicación interna

Tal como se indicó en un principio, los sistemas multiagentes generados por el asistente deberán cumplir con todas las normativas de FIPA. Por tal motivo, para la comunicación interna entre los agentes se utilizará todo lo dictaminado para el lenguaje FIPA-ACL y sus 22 actos comunicativos. Bajo estos estándares se diagramarán los mensajes que utilizarán todos los agentes que forman el sistema para cualquier tipo de comunicación.

En cuanto a la arquitectura para la comunicación entre los agentes, tal como se describió en la sección de arquitectura de los sistemas multiagentes, existirán dos estructuras conviviendo. Por un lado encontramos a todos los agentes funcionales comunicándose entre ellos según el modelo de referencia de agentes dictaminado por FIPA y por el otro tenemos los canales de comunicación directa entre los agentes, ya sea el Servicio de Transporte de Mensajes de los agentes de interfaz, así como el de los agentes funcionales.

Volviendo a los agentes de la estructura FIPA, existirán cuatro componentes que cumplirán las funciones de sistema de gestión de agentes (AMS), canal de comunicación (ACC), directorio de servicios (DF) y sistema encargado de transporte de mensajes (IPTM). Dentro de los agentes que se comunicarán dentro de esta estructura se encuentra también el agente de comunicación.

Este agente será, además, el nexo con todos los agentes de interfaz, quienes interpretan la otra estructura de comunicación. Al tratarse de canales de comunicaciones fijas y a la vez dirigidos por el agente de comunicación, estos agentes no necesitan ningún tipo de servicio de directorio y sólo les alcanza con un servicio de transporte de mensaje, como el definido en la arquitectura abstracta de FIPA.

Todos estos agentes, conocerán el nombre, la localización y los atributos, de los agentes con los que deba comunicarse. De esta manera, los canales quedan fijos y las comunicaciones, obviamente utilizando FIPA-ACL, serán tal como se define en el esquema presentado en la sección de arquitectura.

6.8 - Agente de comunicación

Como se pudo observar en la arquitectura presentada para los sistemas multiagentes, existe dentro de ellos un agente de comunicación. Su función es la coordinación de todos los agentes que se encuentran dentro del grupo de agentes de interfaz.

Éstos son los que llevan a cabo todo tipo de comunicación posible entre el sistema y el usuario. Por este motivo, el agente de comunicación puede ser definido como el cerebro de la comunicación sistema-usuario.

Aunque cada agente cuenta con la capacidad de aprendizaje, tal como fue indicado en sus definiciones, el agente de comunicación tiene el papel más importante en lo que se refiere aprendizaje dentro de los agentes de interfaz. Esto se debe a que es su responsabilidad aprender de los gustos del usuario en lo que se refiere a comunicación.

Así, se determinará si una información que surge dentro de algún agente del sistema será presentada al usuario o no, por ejemplo y también la manera en que ésta será presentada, ya sea sólo por el agente generador de texto, o también con la participación del generador de voz siempre, según los gustos del usuario.

Tal vez esto suene como un “cuello de botella” dentro de la comunicación entre el sistema multiagente y el usuario pero creemos que es clave para el éxito de AI4U que el sistema sepa elegir cuándo comunicarse con el usuario y no convertirse en una molestia. Hay que entender entonces que es casi tan importante la información generada por el sistema como la decisión de este de presentar la información o no según el momento y los factores que determine el usuario.

Así como este agente cumple la función descrita anteriormente, también la cumple en la dirección contraria, ya que debe comunicar lo que los agentes de interfaz capturaron desde el usuario a los agentes funcionales que corresponda.

Para esta tarea, el agente contará con los servicios definidos en FIPA a los que tiene acceso según

la estructura diseñada. De esta manera contará con el directorio de servicios, que le ayudará a definir a qué agentes comunicar lo que solicitó el usuario.

Además, entre todas las comunicaciones internas entre los agentes funcionales, las solicitudes de los usuarios se hacen conocer masivamente y cada agente comprenderá su papel en dicha solicitud y respondiendo de forma acorde, como está definido en sus principios.

6.9 - Reconocimiento de voz

Para que el sistema multiagente creado cuente con la capacidad de reconocimiento de voz existirá un agente encargado. El agente reconocedor de voz convertirá las palabras del usuario en un texto que será transmitido al agente reconocedor de texto, tal como se indica en la arquitectura del sistema.

Para realizar la conversión se utilizará la interfaz Microsoft Speech API y para optimizar su funcionamiento, se aprovecharán las ventajas del uso de los agentes realizándose una corrección por aprendizaje y las ventajas obtenidas del desarrollo de ConceptNet, el cual será explicado mas adelante (ver punto 6.9.1).

La elección de Microsoft Speech API se basa en su ya conocido éxito en el mercado, en su constante actualización por parte de Microsoft, su diseño flexible y sobre todo su completa orientación a la posibilidad de explotarlo de las más diversas maneras por los desarrolladores.

La SAPI es una interfaz de reconocimiento de habla y de traducción de texto a voz, para la programación de aplicaciones. Permite incluir en cualquier aplicación reconocimiento de habla (SR) y lectura hablada de texto (TTS), así actúa como una capa de abstracción entre las aplicaciones y los motores sobre tecnología del habla.

Otro punto fuerte en la elección de esta interfaz es que la misma es relativamente cómoda para:

- **Reconocimiento de comandos:** permite al usuario dar órdenes simples a la aplicación o al sistema operativo. También permite rellenar formularios que admiten un número pequeño de entradas.
- **Dictado:** permite al usuario dictar cualquier texto arbitrario para usarlo como texto de entrada en una aplicación.
- **Texto hablado:** permite al usuario transformar cualquier texto escrito en texto hablado. Se emplea una síntesis de voz.

Tanto en el reconocimiento de voz como en el texto hablado aparece un elemento muy importante que es el motor y que no es parte de la SAPI. El motor es quién realmente realiza el trabajo de reconocer el texto a partir de una señal de audio o el trabajo de generar una señal de audio similar a la voz humana a partir del texto escrito.

Dicho motor es un elemento de diseño e implementación compleja que actúa básicamente como una “tubería” por la que entra PCM (código de modulación por pulsos) y sale el texto.

Un motor de reconocimiento de voz fundamentalmente:

- Transforma el PCM en una representación acústica más manejable, dividiendo la señal en intervalos pequeños sobre los que aplica una FFT.
- Aplica una gramática para limitar el conjunto de fonemas que debe esperar escuchar en la siguiente entrada.
- Decide (se imagina) qué fonemas ha oído.
- Convierte los fonemas en palabras. Aquí también intervienen las gramáticas de una determinada lengua.

Cuando las primeras tecnologías de reconocimiento de voz aparecieron en el mercado, todas requerían un primer entrenamiento sobre el usuario en el que lo hacían leer un texto para poder capturar todas sus particularidades del habla.

La nueva generación de motores de reconocimiento de voz no necesita este entrenamiento. A pesar de esto, el agente de reconocimiento, como está definido en su esencia, aprende y es por esto que a medida que va pasando el tiempo va haciendo su propio “entrenamiento” en el que va a ir aprendiendo las características del habla del usuario y de esta manera logrará optimizar el funcionamiento global de su objetivo.

Dentro de la característica de corrección por aprendizaje, podemos encontrar dos técnicas que ya fueron probadas y calificadas como exitosas. Por un lado encontramos la corrección por probabilidad de uso de palabras, en la cual se identifican las palabras más utilizadas por el usuario realizando un ranking probabilístico y de acuerdo a esto se brinda apoyo al sistema de reconocimiento ante la duda entre palabras.

La otra técnica es el uso de palabras que son manejadas por ejemplo en los sistemas multiagentes creados para dar soporte en investigaciones. Estas palabras claves son las mismas utilizadas por otros agentes para cumplir sus objetivos dentro del proyecto y ayudarán al sistema de reconocimiento en caso de dudas entre palabras.

Por ejemplo, si se trata de un proyecto de investigación sobre las ovejas en la Patagonia y el sistema de reconocimiento tiene duda entre “abeja” y “oveja”, contar con estas palabras claves le ayudará en algunas decisiones. Esta técnica se puede ver potenciada en ConceptNet, explicado a continuación.

6.9.1 - ConceptNet

En el MIT se desarrolló ConceptNet, el cual busca utilizar el razonamiento común para mejorar el reconocimiento de voz. Este sistema trabaja como apoyo de Microsoft Speech API por lo que es completamente implementable en nuestro sistema.

ConceptNet busca despejar la ambigüedad que ocurre entre palabras con fonéticas similares, mejorar las interfaces de corrección de errores y mejorar la precisión global de reconocimiento.

Para poder realizar todo esto, a las alternativas de reconocimiento del API de Microsoft se les suma un listado de palabras generadas según el contexto de la frase. A esta combinación se le suma una ordenación basada en el sentido de la palabra dentro del contexto y así consigue sus objetivos.

Para que el funcionamiento quede más claro, veremos un ejemplo. Imaginemos que se está diciendo el siguiente texto: “El otro día fui a la escuela. Cuando llegué me dirigí al aula del ‘director’”. Ahora bien, cuando el usuario pronuncia la palabra ‘director’, el Microsoft Speech API encontrará varias alternativas como las siguientes: “director”, “rector”, “pintor”. Si el sistema reconoció bien la palabra, seleccionará ‘director’.

Ahora es cuando ConceptNet entra en acción. Entendiendo que se está hablando de la escuela, seleccionará de su base de datos, todas las palabras vinculadas a escuela que tenga, como “pizarrón”, “maestro” y también “director”. Al reencontrar la palabra reconocida, se disminuye notablemente el grado de error del sistema.

Por último, ConceptNet ordenará las alternativas en relación al contexto de la frase y a la coincidencia o no de los listados anteriores. De esta manera consigue el objetivo del sistema.

6.10 - Capacidad del habla

Para dotar al sistema multiagente con la capacidad del habla, existirá un agente, que convertirá lo que el agente de generación de texto cree con el correspondiente sonido. El agente se encontrará dentro de la estructura de agentes de interfaz descrita en la sección de arquitectura.

El sonido generado deberá ser acorde a la personalidad que se seleccionó al crear el sistema, por lo que el agente utilizará el sistema Microsoft Agent. Un punto muy fuerte a la hora de la selección de esta tecnología, es que la misma se encuentra puramente orientada para ser utilizada por desarrolladores.

La Microsoft Agent Programming Interface es la que da las libertades necesarias para que se le pueda aportar muchísimas de las ventajas propias de un agente inteligente en su funcionamiento. Además su diseño se encuentra basado en la utilización de personajes con sus características correspondientes del habla.

A través de los servicios que ofrece Microsoft Agent, los desarrolladores pueden incluir personajes animados en las interfaces de usuario de sus aplicaciones o de sus páginas Web. Estos personajes pueden moverse libremente a través de toda la pantalla, hablar, o mostrar viñetas de texto, o incluso esperar órdenes de voz.

Esta interfaz se encuentra basada en COM y de esta manera provee muchas libertades de programación. Entre sus servicios se incluyen la habilidad de cargar un personaje, reproducir cierta animación, hacerlo hablar utilizando el motor de sinterización del habla e incluso la sincronización de los labios en relación a lo que se está diciendo.

Todo este potencial de Microsoft Agent se encuentra dentro del servidor de animación con el que cuenta. La tarea más significativa de este servidor es la presentación y administración en pantalla de la animación del personaje. Para esto el servidor solicita los cuadros desde el componente Agent Data Provider el cual carga el archivo de animaciones correspondiente al carácter, descomprime los datos de animación y sonidos, y se los envía al servidor.

Este último, al recibir la información, presenta la imagen en una región de la pantalla permitiéndole al personaje aparecer en cualquier parte de ella. Esta técnica fue desarrollada no solo por la flexibilidad sino también porque crea la ilusión de que el personaje se encuentra operando dentro del ambiente de trabajo del usuario y se encuentra junto al usuario trabajando con él.

Un motor de texto a voz, fundamentalmente:

- Normaliza el texto, que significa básicamente sustituir los elementos del texto de los que no se puede deducir su pronunciación directamente de su forma escrita (abreviaturas, números, fechas, etc.) por la forma escrita que da su correcta pronunciación.

- Elimina las ambigüedades que pudiera haber sobre la pronunciación. Para esto muchas veces hay que estudiar el contexto y la probabilidad de aparición de cada vocablo.
- Genera una pronunciación para las palabras.
- Aplica una prosodia adecuada.
- Concatena los segmentos de señal para producir una voz continua.

6.11 - Animación

Aprovechando las capacidades de Microsoft Agent que se mencionaron anteriormente, se dotará al sistema multiagente para contar con una animación física que lo represente.

Las animaciones deberán encontrarse asociadas al tipo de personaje, para que el usuario se acostumbre y entienda los movimientos animados del sistema. Así, el usuario vinculará ciertos movimientos a hechos de importancia y prestará más atención al sistema.

La animación dentro del sistema multiagente estará a cargo del agente de animación, el cual formará parte también de la estructura de agentes de interfaz descrita en la sección de arquitectura (ver punto 6.5).

6.12 - Exportación

Como se dijo al describir el asistente, existirán tres tipos de exportaciones para los sistemas multiagentes creados por el usuario: la exportación local, la exportación web y la exportación de mensajería instantánea.

En la opción de exportación local, el sistema será creado en la misma computadora donde se instaló y ejecutó el asistente. Se accederá al sistema mediante un acceso directo y se contará además con la posibilidad de que el mismo se cargue automáticamente al inicio del sistema operativo.

Para poder reconocer que el mismo se encuentra funcionando y además para poder invocarlo, tendrá un icono en la barra de tareas y de haberse incorporado el reconocimiento de voz, el mismo podrá ser simplemente llamado como se le haya aplicado.

En el caso de la exportación web se buscará que el sistema pueda ser utilizado en un sitio web. Por este motivo, al ser creado, contará con un código en JavaScript que al ser incorporado en el sitio web le permitirá implementar su interfaz de comunicación ahí.

Para la última opción, la exportación para mensajería instantánea, el sistema solicitará la información de la cuenta que se ha creado para que pueda acceder al servicio y automáticamente se conectará a él. Una vez conectado, se comunicará como cualquier otro ser humano y cumplirá con su objetivo.

De contar con la opción de comunicación por voz, el sistema podrá aceptar conferencias virtuales y poder comunicarse mediante el uso de la voz. Además, de contar con una animación propia, podrá mostrarla como si esta fuera su cámara web.

6.13 - Actualizaciones

Otra gran ventaja del diseño que se utilizará para los sistemas multiagentes creados por el usuario, es la posibilidad de ser ampliados con nuevas funciones. O sea, por su tipo de arquitectura, será posible agregarle en un futuro nuevos agentes al sistema con nuevas funcionalidades.

De esta manera, se podrá en un futuro distribuir paquetes con nuevas funciones para agregarles a los sistemas creados, o incluso permitir la actualizaciones y ampliaciones online.

6.14 – Prototipo no funcional

Un prototipo no funcional también es conocido como el diseño de la interfaz de usuario ya que no contiene un código con funcionamiento sino que se limita a realizar una presentación visual al usuario del sistema.

En este caso, el objetivo del prototipo no funcional es exclusivamente el de presentar gráficamente como funcionaría el AI4U de ser construido y, principalmente, demostrar la facilidad que encontraría el usuario tanto durante el asistente de creación como durante el funcionamiento del sistema.

Como la mayor parte del sistema funciona de forma oculta para el usuario, lo que se puede demostrar en este prototipo es la interacción del sistema con el usuario, que es la única parte visible.

Es por esto que, el prototipo ha sido desarrollado en Flash porque se ha buscado demostrar de una manera gráfica la simplicidad del programa frente al usuario y sobre todo, mostrar por medio de animaciones, ejemplos de interacciones donde se vea al sistema multiagente en funcionamiento.

A pesar de esto, se ha buscado acercar lo máximo posible al prototipo con una versión final del producto y para esto, en el desarrollo del prototipo se utilizaron todas las herramientas descritas anteriormente, que formarían parte de AI4U.

Por lo tanto, al ver el funcionamiento del producto por medio del prototipo, hay que tener en cuenta que se ha utilizado a Microsoft Agent para el manejo de las animaciones, así como también los sistemas de reconocimiento de voz y de conversión de texto a audio, tal como funcionarían en una versión final.

El prototipo creado cuenta con un menú principal en el que se puede seleccionar qué parte se desea visualizar. Las opciones presentadas son tres animaciones claves donde se podrá, luego de leer este trabajo, ver toda la idea en funcionamiento.

La primera opción corresponde al asistente que se utiliza para crear el sistema multiagente. Será posible visualizar todos los pasos que componen al asistente y principalmente notar cómo cualquier usuario podría, por medio de este asistente, crear su sistema multiagente totalmente personificado.

La primera de las interacciones presenta en funcionamiento a un sistema multiagente creado como "Sistema de ayuda en investigaciones". Dentro de las funciones particulares, se le incorporo sólo la de "Búsqueda de información" y de las funciones globales se seleccionaron "Correo Electrónico", "Multimedia" y "Acceso rápido".

Cabe aclarar que el sistema no tiene capacidad de habla pero si cuenta con reconocimiento de voz y además la capacidad de animación, utilizando el personaje "Merlín". La exportación ha sido local y el sistema se encuentra funcionando desde el comienzo de la investigación del usuario y ha aprendido todo lo que necesitaba saber acerca de la investigación por lo que se encuentra en pleno proceso de poder ayudarlo.

En la interacción se puede ver como un sistema multiagente como éste, una vez "entrenado" por el usuario, puede ser de extrema ayuda en investigaciones o cualquier otro tipo de trabajo relacionado.

La otra interacción entre un usuario y un sistema multiagente que se presenta, trata de un sistema multiagente creado como un "Sistema de ventas". Como funciones particulares se le han incluido las de "Productos y servicios", "Clientes" y "Marketing" y en el caso de las funciones globales se le han incluido sólo la de "Fechas personales". No se le han incluido ni reconocimiento de voz, ni capacidad de hablar, ni animación mientras que a la hora de exportación, se ha seleccionado la exportación hacia "Mensajería instantánea".

Dentro de esta interacción, el sistema se encuentra ya funcionando sobre una cuenta de MSN Messenger, dentro de un dvd club llamado "DVD Movies" y ya lleva tiempo funcionando allí. Por esto ya aprendió no sólo la información personal de los clientes conectado por mensajería instantánea sino también acerca de sus gustos.

DVD Movies cuenta con un sistema muy simple basado en Excel donde se registran los alquileres y toda la información relacionada, como la disponibilidad de las películas, por lo que el sistema puede consultar aquí cuando quiere actualizar su información acerca de las disponibilidades o reservas de películas.

Así, en la interacción se puede ver un pequeño chat donde se ve claramente al sistema funcionando y sobre todo, los grandes beneficios que se pueden obtener de contar con él funcionando en un dvd club, por ejemplo.

7 - Conclusión

A lo largo de este trabajo hemos analizado qué es un Agente Inteligente y estudiado el prometedor mundo de los Sistemas Multiagentes. Además, más allá de los aspectos teóricos, hemos planteado todo el diseño de una posible aplicación que permitiría llevar la tecnología de la Inteligencia Artificial al uso diario de los usuarios comunes en sus distintas tareas.

A pesar de lo prometedor que pueda parecer todo el diseño desarrollado bajo el nombre de "AI4U", hay que ser realistas también y saber que el desarrollo definitivo y real de este producto necesitaría no sólo mucho tiempo y muchos recursos sino también una fuerte inversión por parte de los interesados en su desarrollo.

Por otra parte, aunque durante todo el trabajo se ha insistido en demostrar que ya tenemos la tecnología y el conocimiento suficiente para llevar la Inteligencia Artificial a través de los Sistemas Inteligentes a la producción masiva de productos, todavía las grandes empresas informáticas piensan que por el momento hay que seguir manteniendo a esta tecnología dentro del campo de la investigación y desarrollo.

Veamos que el mundo que nos rodea se parece cada día más a las visiones futuristas más optimistas que tuvieron grandes científicos en distintos momentos de la historia, donde tal vez el sistema multiagente de un taxi se pueda comunicar con el agente del semáforo en una esquina para saber su situación y a la vez estar conectados a un GPS y así decidir el mejor camino, además teniendo en cuenta la preferencia del pasajero. Esperemos que los Sistemas Multiagentes logren ser la llave del gran cambio y en muy poco tiempo.

Creemos que la apuesta es grande y tal vez arriesgada para los ojos de los directivos de las grandes compañías pero consideramos que ver la mitad del vaso lleno y las enormes posibilidades futuras de este tema, cambiaría completamente la perspectiva. De arriesgarse alguna empresa, desarrollar el producto final y tener éxito, no sólo representaría un quiebre histórico en el mundo del desarrollo de software, sino que representaría ganancias millonarias para los precursores.

Al igual que los autores de todos los documentos y bibliografía que estudiamos antes de realizar este trabajo, también les tenemos un enorme agradecimiento por haber apostado por los Sistemas Multiagentes y sus posibilidades a las que quiero agregar mi pequeña contribución mediante este trabajo.

Somos conscientes de que hay muchos años de investigaciones y muchos recursos invertidos... por eso tal vez ya tengamos entre nosotros la clave para implementar la Inteligencia Artificial en un mundo cada vez más informatizado que necesitaría de sus aportes para pasar a un nivel quizás impensado en la actualidad.

8 - Bibliografía

- Bond, A. H. & Gasser L., *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, 1988.
- C. Reynolds, *Flocks, Herds an Schools: A distributed Behavioral Model*, Computer Graphics 21, 1987.
- Carlos Ángel Iglesias Fernández, "Introducción a la Inteligencia Artificial Distribuida", *Sistemas Inteligentes*, Grupo de Sistemas Inteligentes, Universidad Politécnica de Madrid, 1999.
- Carlos Serrano, "La contabilidad en la era del conocimiento", *Sistemas informativos contables*, Universidad de Zaragoza, 2006.
- Computer Science Department, "Agent Communication", *An Introduction to Multiagent Systems*, The Faculty of Science, The University of Liverpool, 2006.
- D. Eichmann, "Ethical Web agents", *Second International World-Wide Web Conference*, pp. 3 – 13, 1994.
- Departamento de Sistemas e Informática, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, "Agentes Inteligentes", *Introducción a la Inteligencia Artificial*, Licenciatura en Ciencias de la Computación, Universidad Nacional de Rosario, 2004.
- Dr. Miguel Rebollo, "Comunicación entre agentes", *Departamento de Sistemas Informáticos y Computación*, Universidad Politécnica de Valencia, 2006.
- Eichmann, D. T., "Ethical Web Agents", *Proceedings of the 2nd WWW Conference*, 1994.
- Ephrati, E., Zlotkin, G. & Rosenschein, J.S., "A non-manipulable Meeting Scheduling System", *13th International Workshop on Distributed Artificial Intelligence*, 1994.
- Foundation for Intelligent Physical Agents, "FIPA ACL Message Structure Specification", *FIPA TC Communication*, 2002.
- Foundation for Intelligent Physical Agents, "Historia de la fundación y todas las especificaciones técnicas dictaminadas", *Sitio web de la FIPA*: <http://www.fipa.org>, 2006.
- Foundation for Intelligent Physical Agents, *SC00061G FIPA ACL Message Structure Specification*, 2002.
- Franklin, Stan & Graesser, "Is it an agent or just a program?: A taxonomy for autonomous agents", *Intelligent Agents III: Agent Theories, Architectures, and Languages*, Springer-Verlag, 1997.
- Garrido, L. and Sycara, K., "Multi-Agent Meeting Scheduling: Preliminary Experimental Results", *Second International Conference on Multi-Agent Systems*, 1996.
- Garrido, L., Brena, R. & Sycara, K., "Cognitive Modeling and Group Adaptation in Intelligent Multi-Agent Meeting Scheduling", *First Iberoamerican Workshop in Distributed Artificial Intelligence and Multi-Agent Systems*, 1996.
- Grupo de Docencia Inteligencia Artificial, Informática Teórica y Compiladores, "Diseño de Software de Agentes Inteligentes Autónomos", *Departamento de Informática*, Universidad Carlos III de Madrid, 2005/6.
- Hyacinth S. Nwana, "Software Agents: An Overview", *Knowledge Engineering Review*, 1996.
- J. J. Merelo, "Agentes Autónomos Inteligentes", Red Científica: Ciencia, Tecnología y Pensamiento: <http://www.redcientifica.com/doc/doc199903310001.html>, 2006.
- Jacques Ferber, *Multi-agent Systems: An introduction to distributed artificial intelligence*, Ed. Addison-Wesley, 1999.

- Javier de Andrés de la Universidad de Oviedo, Enrique Bonsón de la Universidad de Huelva, Tomas Escobar de la Universidad de Huelva y Carlos Serrano de la Universidad de Zaragoza, "Inteligencia Artificial y Contabilidad", *Publicaciones AECA*, 2006.
- Juan José Samper Márquez, "Agentes inteligentes (Curso)", *WikiLearning*: http://www.wikilearning.com/agentes_inteligentes-wkccp-5095-1.htm, 2005.
- Julian V. & Botti V., "Agentes Inteligentes: el siguiente paso de la inteligencia artificial", *Departamento de Sistemas Informáticos y Computación*, Universidad Politécnica de Valencia, 2000 (<http://www.ati.es/novatica/2000/145/vjulia-145.pdf>).
- K. Sycara, "The many faces of Agents", *American Association for Artificial Intelligence*, Estados Unidos, 1998.
- L. F. Castillo, M. G. Bedia & J. M. Corchado, "Librería para el intercambio de Mensajes FIP-ACL entre dispositivos móviles a través de Bluetooth", *Departamento Ciencias Computacionales*, Universidad Autónoma de Manizales y *Departamento Informática y Automática*, Universidad de Salamanca, 2005.
- Leandro, "Inteligencia Artificial", *Monografias.com*: <http://www.monografias.com/trabajos16/inteligencia-artificial/inteligencia-artificial.shtml>, 2006.
- Leonardo Garrido Luna, "Sistemas multiagentes descentralizados", *Centro de Sistemas Inteligentes*, Transferencias, Año 10 (Número 37), Publicación del Campus Monterrey del Tecnológico de Monterrey, 1997.
- Lesser, V. & Gasser, L., "Proceedings of the First International Conference on Multiagent Systems", *MIT Press*, 1995.
- Loreto Susperregi, Iñaki Mautua, Carlos Tubío, Miguel Angel Pérez, Iñigo Segovia & Basilio Sierra, "Una arquitectura multiagente para un Laboratorio de Inteligencia Ambiental en Fabricación", *Fundación TEKNIKER*, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad del País Vasco, 2004.
- M. Marcos, "Transparencias IA", *Inteligencia Artificial*, Universidad Jaime I, 1999.
- M. Wooldridge & J. Whiley & Son., *An Introduction to Multiagent Systems*, 2002
- M. Wooldridge & N. Jennings, "Agent-Oriented Software Engineering. Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering", 1999.
- M. Wooldridge & N. Jennings, "Intelligent Agents: Theory and Practice", 1995.
- M. Wooldridge & N. Jennings, "Software Agents", *IEE Review*, 1996.
- Maes P., "Agents that reduce work and information overload", *Communication of the ACM*, 1994.
- Maes, P., "Agents that Reduce Work and Information Overload", *Communications of the ACM*, 37(7), pp. 30-40, 1994.
- Maes, Pattie, "Artificial Life Meets Entertainment: Life like Autonomous Agents", *Communications of the ACM*, 1995.
- María Victoria Belmonte Martínez, "Programación de Sistemas Multiagente", *Departamento Lenguajes y Ciencias de la Computación*, E.T.S.I. Informática, Lenguajes y Ciencias de la Computación, Universidad de Málaga, 2006.
- O'Hare, G., Jennings, N., John Wiley & Sons, *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons., 1996.
- Philip R. Cohen & C. Raymond Perrault, "Elements of a Plan-Based Theory of Speech Acts", *Cognitive Science* 3, 1979.
- Raifa, H., "The Art and Science of Negotiation", *Harvard University Press*, 1982.
- Ramón M. Gómez Labrador, *El lenguaje ADL*, Departamento de Lenguajes y Sistemas Informáticos, Escuela Técnica Superior de Ingeniería Informática, Universidad de Sevilla, 1997.
- Roberto A. Flores-Mendez, "Towards a Standardization of Multi-Agent System Frameworks", *Computer Science Department*, University of Calgary, 1999.
- Roxana A. Belecheanu, Steve Munroe, Michael Luck, Tim Miller, & Peter McBurney, "Comercial Applications of Agents: Lessons, Experiences and Challenges", *International Conference on Autonomous Agents and Multiagent Systems*, Industry Track, Hakodate, Japan, 2006.
- Sen, S. and Durfee, E. H., "Unsupervised Surrogate Agents and Search Bias Change in Flexible Distributed Scheduling", *First Conference on Multi-Agent Systems*, AI Press, 1995.
- Shaowen Wang, Anand Padmanabhan, Yan Liu, Ransom Briggs, Jun Ni, Tao He, Boyd M. Knosp & Yasar Onel, "A Multi-agent System Architecture for End-User Grid Monitoring Using Geographic Information Systems (MAGGIS): Architecture and Implementation", *Academic Technologies-Research Services of Information Technology Services*, Department of Physics and Astronomy, The University of Iowa, 2003.

- Stan Franklin & Art Graesser, "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents", *Institute for Intelligent Systems*, University of Memphis, 1996.
- Stuart Russell & Peter Norvig, *Artificial Intelligence: A Modern Approach (Second Edition)*, Prentice Hall, 2003.
- Taub, E., "Sharing Schedules", *Mac User*, pp. 155-152, Julio, 1993.
- Waseem Daher, Jose Espinosa, Alex Faaborg & Henry Lieberman, "Using Commonsense Reasoning to Improve Voice Recognition", *Software Agent Group*, MIT Media Laboratory, Massachusetts Institute of Technology, 2005.
- Wikipedia, la enciclopedia libre, *Sistemas Multiagentes*: http://es.wikipedia.org/wiki/Sistemas_multiagentes, 2006.
- Yoav Shoham, "Agent-oriented programming", *Artificial Intelligence 60*, Robotics Laboratory Computer Science Department, University of Stanford, Estados Unidos, 1993.
- Yoav Shoham, *Artificial Intelligence*, Elsevier Science Publishers Ltd., 1993.

