



INSTITUTO POLITÉCNICO NACIONAL CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

No. 67 Serie: ROJA Fecha: Octubre 99

Colaboración Dirigida entre Agentes con Propósito

Jesús M. Olivares Ceja¹
Ma. Del Carmen Domínguez¹
Araceli Demetrio A¹
Roberto Zavala Fernández¹
Adolfo Guzmán A¹

RESUMEN

Presentamos en este trabajo un modelo en el que se pueden estudiar las interacciones entre agentes que tienen propósitos y se pueden dar comportamientos emergentes de colaboración. En este caso cuando mi agente participa en alguna interacción, adquiere un papel que es el que desempeña. La participación del agente es para alcanzar sus propósitos. Las acciones de un agente son siempre compatibles y congruentes, cuando algo no va bien, termina la ejecución de la hebra con falla. Cada papel activo genera una hebra de ejecución (thread), por lo tanto, cuando el agente alcanza algún propósito algunas hebras pueden quedar incompletamente terminadas o terminar antes como consecuencia de haber satisfecho el propósito que las mantenía activas.

Palabras clave: Agentes, colaboración dirigida, comparador de ontologías mixtas, inteligencia artificial, máquina de eventos inesperados, threads.

¹Centro de Investigación en Computación del Instituto Politécnico Nacional
Este artículo fue publicado en el Foro "Computación, de la Teoría a la Práctica"
Méx., DF, 26-28 may/99 pp. 210-219

COLABORACIÓN DIRIGIDA ENTRE AGENTES CON PROPÓSITO

Jesús M. Olivares Ceja, Ma. Del Carmen Domínguez, Araceli Demetrio A.,
Roberto Zavala Fernández, Adolfo Guzmán A.
e-mail: jesuso@acm.org

Centro de Investigación en Computación del Instituto Politécnico Nacional
Av. Juan de Dios Bátiz, casi esquina con Miguel Othón de Mendizábal Ote.
Unidad Profesional "Adolfo López Mateos", del I.P.N. C.P. 07700 México, D.F.

Resumen

Presentamos en este trabajo un modelo en el que se pueden estudiar las interacciones entre agentes que tienen propósitos y se pueden dar comportamientos emergentes de colaboración. En este caso cuando mi agente participa en alguna interacción, adquiere un papel que es el que desempeña. La participación del agente es para alcanzar sus propósitos. Las acciones de un agente son siempre compatibles y congruentes, cuando algo no va bien, termina la ejecución de la hebra con falla. Cada papel activo genera una hebra de ejecución (thread), por lo tanto, cuando el agente alcanza algún propósito algunas hebras pueden quedar incompletamente terminadas o terminar antes como consecuencia de haber satisfecho el propósito que las mantenía activas.

Las interacciones dirigen de manera emergente la colaboración entre agentes.

Durante la ejecución de las hebras de un agente pueden ocurrir eventos inesperados que también alteran la ejecución de las mismas, esto se genera mediante lo que hemos llamado Máquina de Eventos Inesperados (MEI).

El intercambio de información entre los agentes participantes se realiza en dos etapas, en la primera, consideramos que los agentes manejen una ontología común, es decir, entienden los mismos conceptos; en la segunda, puede suceder que los agentes manejen ontologías mixtas y por lo tanto hemos desarrollado el Comparador de Ontologías Mixtas (COM).

INTRODUCCIÓN

Actualmente, con el aumento en la complejidad de los sistemas de información se hace necesario contar con elementos de software a manera de aliados que apoyen al usuario en el manejo de su información. Sobre esta motivación se han desarrollado algunos sistemas basados en agentes (Maes 1994) que heredan del usuario su autoridad para actuar en su nombre y entonces tomar cierto tipo de decisiones de manera autónoma y productiva (sin intervención explícita del usuario).

Las aplicaciones de dichos agentes son diversas: las encontramos en prototipos para el entretenimiento (Grant 97), diseño emergente (Edmonds 94), manufactura (Zita 97), enseñanza (Selker 1994) (Sanchez 97) (Ayala 98), comercio electrónico (Chavez 97) (Noriegra 97), milicia (Amori 92), filtros de información (Baclace 92), extensiones de los sistemas basados en agentes (Amandi 97) (Conrad 97), programación de agentes, lenguajes de programación (Canfield 94), aplicaciones en Internet (Etzioni 94) (Barret 97), ambientes virtuales (Tu 94) (Maes 95), entre otras.

Los agentes tienen sus antecedentes en los trabajos de Minsky (Minsky 85) y otros investigadores (Erceau), en los intentos de obtener comportamientos inteligentes a partir de componentes no inteligentes y por otra parte como evolución normal del aumento en la complejidad de los sistemas de información existentes y su necesidad de cooperar (Schneiderman 97). Actualmente se han intentado generar algunos estándares sobre el área de agentes, un ejemplo de los logros alcanzados se describen en (Virdhagriswaran 95).

En algunos casos a los agentes de software se les intenta mostrar con características antropomórficas, como emociones (Baes 94), personalidad (Moon 96) (Lester 97) (King 96), creatividad (Boden 94), con esto los investigadores pretenden acercar a los usuarios una forma amigable y de ayuda en los sistemas de información.

En los trabajos mencionados encontramos modelos de los agentes y algunos aspectos de sus interacciones (Riecken 94). En nuestro grupo las investigaciones se refieren principalmente sobre las interacciones entre agentes y los requerimientos que surgen como consecuencia de atender eventos inesperados y manejar ontologías mixtas entre agentes.

Las interacciones contienen papeles que toman los agentes y son los que se ejecutan; para que un agente tome un papel debe cumplir los requisitos de que no esté ocupado y cumplir los requisitos para desempeñarlos. Una vez que algún agente toma un papel, entonces adquiere una hebra de ejecución asociada al mismo.

Como consecuencia de la participación del agente en diversas interacciones, puede ocurrir que en un momento dado tenga que ejecutar instrucciones contradictorias, que pueden causar que una de las hebras en conflicto se termine antes de tiempo, o también puede ocurrir que otras hebras se activen.

Otro aspecto que altera la ejecución de las hebras del agente son eventos inesperados que pueden ocurrir en su ambiente y a los que tiene capacidad de responder, esto lo manejamos mediante una Máquina de Eventos Inesperados.

La comunicación entre agentes se ha estudiado siguiendo dos enfoques, primero, lenguajes imperativos, en los que el usuario le especifica al agente lo que debe hacer (Rus 97); segunda lenguajes declarativos, que se utiliza para intercambio de información, en este sentido se ha desarrollado principalmente el lenguaje KQML (Finin 94). En una primera etapa del modelo hemos considerado que los agentes se comunican utilizando los mismos conceptos (ontología

común), en una segunda, atacamos el problema que se presenta cuando dos o más agentes manejan conceptos diferentes (Guha 94), esto lo resolvemos mediante un Comparador de Ontologías Mixtas el cual toma las ontologías específicas y las mapea contra una ontología común.

1. MODELO DE INTERACCIÓN ENTRE AGENTES

En nuestro modelo de agentes consideramos el mundo cerrado, esto significa que los agentes van a interactuar con agentes. Los agentes cuentan con propósitos que procuran alcanzar mediante la interacción con los otros agentes, pueden participar en interacciones en donde cuentan con elementos para participar, por lo tanto en situaciones en las que el agente no cuenta con elementos necesarios, entonces, él mismo decide no participar.

El usuario externo es quien crea los ambientes de interacción indicando las propiedades tanto de las interacciones como de los agentes, hecho esto, las interacciones se llevan a cabo sin su intervención.

Tanto los agentes como las interacciones en que participan se describen mediante propiedades establecidas mediante variables. En nuestro caso contamos con diferentes tipos de variables:

- 1) Globales, que pueden visualizar cualquier agente, por ejemplo: el reloj de la vida, la cola de eventos, el pizarrón de comunicaciones.
- 2) Regionales, se conocen en un grupo de agentes.
- 3) Privadas, se utilizan para intercambiar información entre dos agentes.
- 4) Internas, son aquellas propias de un mismo agente y que son accesibles para todas sus hebras.
- 5) Locales, son visibles únicamente dentro de una hebra.

En el modelo cada interacción y agente tiene un conjunto de parámetros que los describen, por lo que para generar instancias de ellos se debe indicar los valores de los parámetros, por ejemplo en una interacción 'restaurante', algunos de esos parámetros corresponden con el número de mesas, el costo de cada platillo, etc. Por lo tanto, cada interacción y agente pueden tener varias instancias. Las interacciones reciben externamente al modelo los eventos generados pro la MEI. Los agentes cuando requieren de hacer equivalentes algunos conceptos recurren al COM.

1.1 AGENTES

Un agente tiene varios propósitos que intenta alcanzar llevando a cabo los papeles que toma de las interacciones en las que puede participar. Los propósitos del agente únicamente él los puede cambiar, las interacciones en las que participa únicamente hacen que el agente logre o no sus propósitos de acuerdo con los valores internos de cada agente.

Los agentes son atómicos y se componen de varias hebras de ejecución que corresponden a los papeles que toman de las interacciones. Cada papel que va a tomar un agente debe estar libre dentro de la interacción y una vez ocupado no se le puede quitar.

La información que representa el conocimiento del agente se comparte entre sus diferentes hebras. Por lo que, si una hebra de un agente se entera que es viernes entonces las demás hebras del agente lo saben inmediatamente y no hay forma de que para otra hebra sea, digamos jueves.

Durante la ejecución, puede ocurrir que se generan excepciones que pueden originarse porque no se tiene acceso a algún recurso, se agotó algún recurso, o bien porque ha ocurrido un evento inesperado.

En el modelo se generan instancias de agentes mediante un comando de creación y la especificación de sus propiedades. Cada instancia es un elemento único dentro del sistema.

Un agente percibe los eventos que ocurren en el ambiente mediante la lectura de variables por lo que existen eventos que solamente pueden percibir algunos agentes dado que tienen capacidad para leer las variables que los representan.

Cuando se satisface un propósito del agente entonces él mismo ya no busca interacciones, sino que se desactiva.

1.2 INTERACCIONES

Una interacción tiene varios papeles que se toman los agentes para ejecutarlos y propiciar el alcance de sus propósitos. Para que un agente tome un papel, debe de satisfacer los requerimientos que se especifican para cada uno, heredando las variables que ayudarán en la comunicación entre agentes.

Al igual que los agentes, se generan instancias de interacciones mediante un comando de creación y la especificación de sus propiedades, por ejemplo, un restaurante tiene platillos, precios, cantidad de personal, número de mesas, etc.

Cada hebra descrita consta de operaciones que pueden modificar las variables tanto globales, regionales, privadas, internas y locales.

2. MÁQUINA DE EVENTOS INESPERADOS

Dentro de la dinámica de los sistemas existe un conjunto de eventos inesperados, los cuales deben manejarse de algún modo, en nuestro modelo lo hacemos mediante una Máquina de Eventos Inesperados (MEI), ésta consiste de un programa que administra la forma en que se producen los diferentes eventos inesperados que rodean al ambiente de interacciones.

La MEI se basa en la teoría de máquina de interacción estudiada por Wegner (Wegner 95, 96, 98 a, 98b) y otros autores, quienes opinan que muchos problemas de interacción, no pueden modelarse con máquinas de Turing (MT) ni con lógica de primer orden, pues estos son formalismos o sistemas cerrados que transforman cadenas de entrada en cadenas de salida en forma aislada exterior, de tal forma que no pueden aceptar entradas externas mientras realizan sus cálculos. Estos dispositivos conceptuales se desconectan del mundo al iniciar sus cálculos y, por lo tanto, no pueden considerar el paso del tiempo externo ni los cambios en el ambiente.

Las máquinas de Turing extendidas para incluir acciones de entrada y salida que soporten interacción dinámica con ambientes externos, se llaman Máquinas Interactivas (MIT). El hecho de que las máquinas de Turing no pueden manejar los eventos interactivos que ocurren dentro de un cálculo, apunta a que las máquinas interactivas es un modelo más general que el de las de Turing. Una evidencia más formal de esto, es el hecho de que los flujos de entrada de la máquinas Interactivas no son expresables mediante cadenas finitas, pues cualquier representación finita puede extenderse dinámicamente por una infinidad de circunstancias o eventos.

La afirmación radical, de que las máquinas de Turing no constituye el modelo de cómputo más general, fue señalado por el mismo Turing quién en 1939 mostró que una máquina de Turing con oráculos capaces de predecir el futuro (como el oráculo de Delphi) sería más poderosa que una máquina de Turing ordinaria. Adicionalmente, en 1975, Milner hizo notar que los procesos recurrentes no pueden expresarse como algoritmos secuenciales, mientras que Manna y Pnueli mostraron que procesos reactivos sin fin, como los sistemas operativos, no pueden modelarse por algoritmos.

El descubrimiento de Gödel, de que los números naturales no es un sistema completo, demostró las limitaciones del método axiomático, y puede extenderse para mostrar que las máquinas interactivas tampoco pueden ser descritas completamente mediante dicho enfoque.

El comportamiento observable de las máquinas interactivas puede especificarse mediante historias de interacción. Las historias de interacción de sistema distribuidos son eventos no secuenciales, con duración y con interferencias entre ellos.

2.1 ELEMENTOS DE MEI

En nuestro modelo la MEI es una tabla que contiene una lista ordenada de los eventos inesperados que pueden ocurrir dentro de las interacciones y que afectan o facilitan los propósitos de los agentes. Algunos pueden hacer que un agente pierda o adquiera recursos, algunos eventos pueden generar o eliminar barreras físicas o morales. Cada evento tiene un nombre, tiempo y duración en que ocurrirá, una intensidad, y una función de probabilidad para generar el siguiente del mismo tipo.

Funcionalmente se cuenta con un intérprete de MEI, que se encarga de controlar los eventos calendarizándolos de acuerdo con su posibilidad de ocurrencia.

Para generar la probabilidad de ocurrencia de un evento (es decir, tiempo en que ocurrirá) se usa una función que genera el evento y utiliza conocimiento relacionado con el mismo para generar el siguiente de su tipo, así por ejemplo, si se tiene tiempo de tormenta eléctrica, el evento rayo será muy frecuente, pero si se tiene tiempo despejado, la probabilidad del evento rayo es prácticamente cero, lo que causa que el evento busque unas condiciones favorables para su ocurrencia.

Se cuenta con dos métodos con el que MEI participa con el ambiente de interacciones, el primero es cuando se inician las interacciones en el ambiente y MEI genera una lista de eventos a producir en un cierto intervalo de tiempo; el segundo, permite actualizar los eventos que ocurren en el ambiente de interacciones, el cual se invoca cuando ya se ha efectuado un evento inesperado y se propagó su acción.

3. COMPARACIÓN Y MAPEO DE ONTOLOGÍAS MIXTAS

Un conjunto de conocimiento formalmente representado se basa en una conceptualización: los objetos y otras entidades que se asume existen en un área de interés y las relaciones entre ellas.

Una conceptualización es una vista abstracta, simplificada del mundo que se quiere representar de acuerdo con un propósito. Cada base de conocimiento, sistema basado en conocimiento o agente, está sujeto explícita o implícitamente a una conceptualización.

Una ontología es una especificación explícita de (o una parte de) una conceptualización. Lo que incluye es un vocabulario de términos y especificación de su sentido. Un agente se asocia a una ontología; esto garantiza consistencia.

El término ontología, en el ámbito del intercambio de conocimiento se usa para significar una especificación de una conceptualización. Esto es, una ontología es la descripción de los conceptos y relaciones que pueden existir para un agente o para una comunidad de agentes.

Un compromiso ontológico es un acuerdo para usar un vocabulario, que es consistente, aunque no completo, respecto a la teoría especificada en la ontología. Se trata entonces de definir un vocabulario común en donde se representa el conocimiento compartido.

Podemos construir agentes comprometidos con ontologías, y diseñamos ontologías con las que los agentes pueden compartir conocimiento.

Si algún agente entiende la ontología puede usar la información. Las ontologías en sí mismas también pueden ser reutilizadas.

La reutilización de algún pedazo de conocimiento requiere de una descripción explícita de los puntos de vista que están inherentemente presentes en el conocimiento. De esta forma, no hay forma de saber si el pedazo de conocimiento es aplicable a una nueva aplicación y porqué.

Una ontología especifica una conceptualización, una forma de ver al mundo. Por lo que cada ontología incorpora un punto de vista sobre el mundo considerado.

Una ontología contiene definiciones que nos proveen del vocabulario para referirse a un dominio, las definiciones dependen del lenguaje que usamos para describirlas.

En una ontología podemos identificar niveles de abstracción. Estos niveles de generalización nos dan una topología de ontologías. La idea es caracterizar una red de ontologías usando multiplicidad y abstracción. Como no podemos aspirar a tener una descripción completa del mundo, podemos pensar en una estrategia de construcción gradual de abajo hacia arriba.

Un concepto puede ser representado de muchas formas, por lo que pueden coexistir múltiples representaciones de un mismo concepto.

Es posible establecer un mapeo entre ontologías, mediante el establecimiento de relaciones entre los elementos de una o más ontologías.

Para describir una ontología, se requiere de un lenguaje, en principio se puede usar cualquier lenguaje de programación, pero a veces carecen de expresividad para escribir lo que se quiere decir.

Para describir una ontología se requiere de un lenguaje, en principio se puede usar cualquier lenguaje de programación, pero a veces carecen de expresividad para escribir lo que se quiere decir.

3.1 LENGUAJES DE DEFINICIÓN DE ONTOLOGÍAS

Los lenguajes para definición de ontologías buscan expresividad y facilidad de uso. Algunos de los que se han definido incluyen primitivas para construir agregados, jerarquías múltiples de clase-subclase, reglas y axiomas; otros también incluyen formas de modularización para poder escribir diferentes ontologías y sus interrelaciones y la posibilidad de tomar una visión a un meta-nivel.

Aún no se tiene un estándar general, algunos lenguajes propuestos y utilizados son:

- 1) Workflow Management Coalition (WFMC) y su lenguaje de especificación Express, el cual es una interlingua para definir y especificar productos, asociado a su ciclo de vida: diseño, manufactura, uso, mantenimiento y desecho.
- 2) CORBA (Common Object Request Broker Architecture), está surgiendo como estándar para recuperar objetos y para invocar operaciones en objetos a través de la red, provee un mecanismo en donde los objetos pueden hacer peticiones y recibir respuestas de forma transparente, para esto utiliza el lenguaje IDL (Interface Definition Language) en el que se especifican los objetos y las operaciones para aplicaciones remotas/distribuidas.

3) KIF (Knowledge Interchange Format) y los gráficos conceptuales son lenguajes para representar ontologías basadas en lógica de primer orden. KIF pretende ser un lenguaje capaz de representar la mayoría de los conceptos y distinciones actuales de los lenguajes más recientes de representación de conocimiento. KIF está basado en lógica de predicados con extensiones para definir términos, metaconocimiento, conjuntos, razonamiento no-monotónico, etc.

3.2 INTEGRACIÓN DE ONTOLOGÍAS

Cuando los participantes del intercambio de información y conocimiento no pueden llegar a un acuerdo para establecer una ontología común, como se ha supuesto para los lenguajes de definición de ontologías, surge el problema de establecer las equivalencias conceptuales para lograr la comunicación, es por esto que se han desarrollado proyectos como:

- 1) CYC, el cual es un proyecto de MCC (Microelectronics and Computer Technology Corporation) que establece algunos fundamentos para el razonamiento de sentido común mediante el desarrollo de ontologías para una gran variedad de aplicaciones específicas. El conocimiento está representado declarativamente en una variante de lógica de primer orden en un lenguaje llamado CYL (CYL Language). Para encontrar las equivalencias conceptuales tiene mecanismo de inferencia y de control. Las ontologías están organizadas en conjuntos modulares llamados microteorías, cada una captura el conocimiento y razonamiento requerido para un dominio específico, tales como espacio, tiempo, casualidad o agentes. Pueden existir múltiples microteorías para un solo dominio reflejando diferentes perspectivas. Se puede ver la ontología monolítica, como una red de microteorías.
- 2) TOVE (Toronto Virtual Enterprise), el cual desarrolla una ontología para empresas, usa definiciones basadas en lógica de primer orden y permite deducir respuestas a preguntas de sentido común (usando Prolog).
- 3) Enterprise, es un proyecto parecido a TOVE pero en inglés. Aquí el énfasis es sobre todo en proveer un ambiente de integración de herramientas y métodos usados en los negocios.
- 4) KACTUS, es un proyecto ESPRIT para el desarrollo de una metodología de reutilización de conocimiento técnico, usa el CML (Conceptual Modelling Language), desarrollado como una parte del proyecto de CommonKADS. Hace distinciones explícitas entre conocimiento del dominio, de inferencia, de tareas y de resolución de problemas, una parte central es la biblioteca de ontologías, organizadas por niveles de abstracción.
- 5) Plinius, es un proyecto con el objetivo de extraer semiautomáticamente conocimiento a partir de textos en lenguaje natural enfocado a materiales cerámicos, usa un léxico para mapear tokens de lenguaje natural a expresiones formales en el lenguaje de

representación de conocimiento, la ontología define el lenguaje en que se expresa la parte semántica.

3.3 COMPARADOR DE ONTOLOGÍAS MIXTAS

El modelo de colaboración entre agentes requiere de mecanismos que pueden ser útiles en el caso cuando los agentes utilizan ontologías mixtas. Esto ha motivado la integración de un módulo Comparador de Ontologías Mixtas (COM) el cual permita encontrar equivalencias conceptuales entre agentes que utilizan ontologías que no tuvieron un acuerdo previo, como lo es en la mayoría de las interacciones que se pueden visualizar cuando dos o más agentes interactúan, esto es, que la persona que es especialista en zapatos, no lo es en frutas y verduras (Guzmán 98), es por esto que para establecer la equivalencia entre ontologías, a cada concepto le asociamos un conjunto de propiedades que lo identifican. De tal forma que cuando se realiza el careo entre ontologías se aprovechan las propiedades conceptuales para llegar a establecer una equivalencia conceptual. Cuando dos o más agentes quieren intercambiar información y cada uno maneja una ontología diferente, se recurre al COM especificándole la oración a traducir y la ontología en la que se encuentra y nos devuelve la equivalencia en la ontología indicada, un llamado puede ser:

Resultado: = com(“mocasines”, ontología zapatero, X, ontología frutero);

En el ejemplo el comparador de ontologías, además de los atributos, los conceptos se encuentran organizados en una jerarquía de conceptos, en donde hay conceptos más generales que otros y que representan los diferentes grados de abstracción. Cabe mencionar que existe un nodo común en la cima de los conceptos, pero se procura evitar llegar a este punto, dado que no hay mucho que decir como equivalencia.

Cuando COM ha encontrado una equivalencia, puede integrar el concepto aprendido en la ontología que no contaba explícitamente con éste, por lo que aprender el concepto, significa agregar un nodo o varios al árbol conceptual de quien lo asimila. Cabe mencionar que este proceso no es una copia de un pedazo de un árbol a otro, sino un proceso de negociación utilizando las propiedades de los conceptos.

CONCLUSIONES

Hemos presentado un modelo sobre la colaboración indirecta entre agentes que buscan alcanzar un conjunto de propósitos. Mediante un comando de creación, el usuario establece los parámetros de las interacciones que se desean considerar en la ejecución del modelo, y también de los agentes. Una vez lanzado a ejecución el modelo, los agentes revisan sus propósitos y buscan las interacciones en donde pueden satisfacerlos, como contraparte, las interacciones revisan que los agentes participantes cumplan con los requisitos establecidos en las mismas, cuando ocurre esto, se le da el papel de agente, que ocasiona que se establezca una hebra de ejecución adicional a las que tenga en proceso.

En el ambiente multihebras algunos agentes van a alcanzar sus propósitos, mientras que otros no; por lo que el ambiente que hemos desarrollado nos permite estudiar el comportamiento de las interacciones y de los agentes que llevan a cabo un trabajo colaborativo emergente.

La MEI permite introducir al modelo circunstancias imprevisibles presentes en la realidad, lo que da al modelo los elementos que típicamente se encuentran en la realidad en cualquier sistema en evolución.

Mediante la COM se puede enfrentar el problema que se presenta cuando los agentes no fueron explícitamente implementados para que interactúen, por lo que, aunque sus ontologías sean diferentes, los agentes pueden llegar a establecer comunicación, intercambio y negociación de información y conocimientos; esto último representado por una familia de variables dentro del modelo.

REFERENCIAS

(Amandi 97) Analia, Price Ana, Towards Object-Oriented Agent Programming: The Brainstorming Meta-Level Architecture, in Proceedings of Autonomous Agents 97, Marina del Rey, CA USA.

(amori 92) Amori Richard D., An Adversarial Plan Recognition System for Multi-agent AirboneThreats, Computer Science Department, East Stroudsburg University, 1992

(Ayala 98) Ayala Gerardo. Yano Yoneo, A collaborative Learning Enviroment Based on Intelligent Agents, in Expert Systems with Applications ISSN 0957-4174, vol. 14, Number ½ January/February 1998.

(Baclace 92) Baclace Paul E., Competitive Agents for Information Filtering, Communications of the ACM, Dec 1992, No. 32.

(Barret 97) Barret Rob, Maglio Paul P., Kellem Daniel C., WBI: A Confederation of Agents that Personalize the Web, in Proceedings of Autonomopus Agents 97, Marina del Rey, CA, USA.

(Bates 94) Bates Joshep. The Role of Emotion in Believable Agents, in Communications of the ACM, July 1994, Vol. 37 No. 7.

(Boden 94) Boden Margaret A., Agents and Creativity, in Communications of the ACM, July 1994, Vol. 37 No. 7.

(Canfield 94) Canfield Smith David, et. Al., KIDSIM: Programming Agents without a Programming Language in Communications of the ACM, July 1994 Vol. 37 No. 7

(Chavez 97) Chavez Anthony, Maes Pattie, Kasbah: An Agent Marketplace for Buying and Selling Goods, MIT Media Lab, Cambridge, MA, USA, 1997.

(Conrad 97) Conrad Stefan, et. Al., Towards Agent-Oriented Specificaiton of Information Systems, in Proceedings of Autonomous Agents 97, Marina del Rey, CA, USA.

(Erceau) Erceau Jean, Ferber Jacques, La Inteligencia Artificial Distribuida, in Mundi Cientifico 116 May Volume 11.

(Etzioni 94) Etzioni Oren, Weld Daniel, A Softbot-Based Interface to the Internet, in Communications of the ACM, July 1994, Vol. 37 No. 7.

(Finin 93) Finin Tim et. Al., Specification of the KQML Agent Communication Language, DARPA Knowledge Sharing Initiative, June 15, 1993.

(Finin 94) Finin Tim et. Al., KQML as an Agent Communication Language, CIKM 94 november 1994, Gaitherburg, MD USA.

(Grand 97) Grand Stephen, et. Al., Creatures: Artificial Life Autonomous Software Agents for Home Entertainment, in Proceeding of Autonomous Agents 97, Marina del Rey, CA.USA.

(Gray 97) Gray Robert S., Agent Tcl, in Dr. Dobb's Journal, March 1997.

(Guha 94) Guha R.V., Lenat Douglas B., Enabling Agents to Work Together, in Communications of the ACM, July 1994, Vol. 37 No. 7.

(Guzmán 98) Guzmán Adolfo, Finding the Main Themes in a Spanish Document, in Expert Systems with Applications ISSN 0957-4174, vol. 14, Number ½ Jaunary/February 1998.

(King 96) King Williams Josseph, Ohya Jun. The Representation of Agents: Antropomorphism, Agency and Intelligence, CHI'96 Companion, Vancouver, BC. Canada.

(lester 97) Lester James C., et. Al., The Persona Efficct: Affective Impact of Animated Pedagogical Agents, CHI 97, Atlanta, GA, USA.

(Maes 94) Maes Pattie, Agents that Reduce Work and Information Overload, in Communications of the ACM, July 1994, Vol. 37 No. 7.

(Maes 95) Maes Pattie, Artificial Life Metts Entertainment: Lifelike Autonomous Agents, in Communications of the ACM, November 1995, Vol 38 No. 11.

(Minsky 85) Minsky Marvin, The Society of Mind, Simon & Schuster Inc. 1985.

(Moon 96) Moon Youngme, Nass Clifford, Adaptive Agents and Personality Change: Complementary versus Similarity as Forms of Adaption, CHI'96 Companion, Vancouver, BC, Canada.

(Noriega 97) Noriega Blanco V.Pablo C., Agent Mediated Auctions: The Fishmarket Metaphor, Memory to obtain his PH.D., Universitat Autonoma de Barcelona, Bellaterra, December 12th, 1997.

(Riecken 94) Riecken Doug. M: An Architecture of Integrated Agents, in Communications of the ACM, July 1994, Vol. 37 No. 7

(Rus 97) Rus Daniela. Gray Robert Kotz David. Transportable Information Agents, in Proceedings of Autonomous Agents 97, Marina del Rey, CA.USA.

(Sanchez 97) Sanchez J. Alfredo, Leggett John J., Schnase John L., AGS: Introducing Agents as Services Provided by Digital Libraries, DL 97 Philadelphia PA, USA.

(Selker 94) Selker Ted, COACH: A Teaching Agent that Lerrns, in Communications of the ACM, July 1994, Vol. 37 No. 7.

(Shneiderman 97) Shneiderman Ben, Maes, Pattie, Direct Manipulation VS Interface Agents, Interactions November-December 1997.

(Tu 94) Tu Xiaoyuan, Terzopoulos Demetri, Artificial Fishes: Physics, Locomotion, Perception Behavior.

(Virdhagriswaran 95) Virdhagriswaran Sankar, et. Al., Standardizing Agent Technology, StandadView Vo. 3 No. 3 September 1995.

(Wegner 95) Wegner Peter, Tutorial Notes: Models and Paradigms of Interaction, Departament of Computer Science, Brown University, USA, September 1995.

(Wegner 96) Wegner Peter, The Paradigm Shift from Algorithms to Interaction, Departament of Computer Science, Brown University, USA, October 14th, 1996.

(Wegner 98^a) Wegner Peter, Goldin Dina, Mathematical Models of Interactive Computing, Draft on Observability and Empiricism.

(Wegner 98^b) Wegner Peter, Towards Empirical Computer Science, Brown University, USA.

(Zita 97) Zita Haigh Karen, Veloso Manuela M., High-Level Palnning and Low-Level Execution: Towards a Complete Robotic Agent, in Proceedings of Autonomous Agents 97, Marina del Rey, CA, USA.