

**UN MODELO DE COMUNICACION DE SOPORTE A LA PLANIFICACION  
FLEXIBLE EN UN SISTEMA DE AGENTES COOPERANTES**

**Dr. Fernando Ramos Quintana**  
framos@campus.mor.itesm.mx  
División de Graduados e Investigación  
ITESM, CAMPUS MORELOS.

**M.C. José Luis Liñán García**  
jlinan@campus.mor.itesm.mx  
División de Ingeniería y Ciencias  
ITESM, CAMPUS MORELOS.

**Abstract**

En este artículo se presenta un Modelo de Comunicación, necesario para la cooperación y colaboración en un Sistema Multiagentes Cooperantes, el cual se aplica a una línea de manufactura flexible compuesta por 4 agentes robóticos (Centros de maquinado, sistema de visión, sistema de almacenaje, banda transportadora) y se demuestra como el modelo propuesto facilita la resolución de problemas de manera cooperativa. El modelo de comunicación propuesto se vincula estrechamente a la flexibilidad, tanto de la planificación como de la ejecución de tareas, aspecto frecuentemente descuidado en los sistemas flexibles.

Se realizó una simulación y se demuestra como el esquema de comunicación soporta las actividades de coordinación y colaboración entre los agentes.

**Palabras claves:** Agente inteligente autónomo, Sistemas Multiagentes, esquema de comunicación, cooperación.

## 1. Introducción:

Según [HUHNS87] la Inteligencia Artificial Distribuida (IAD) está preocupada por la “solución cooperativa de problemas globales por un grupo de entidades distribuidas”. Estas entidades conocidas como agentes, van desde simples elementos de procesamiento hasta entidades de gran complejidad que exhiben conducta racional. La palabra “agente” es usada en un amplio sentido para designar una entidad inteligente actuando racionalmente e intencionalmente respecto a sus objetivos y el estado actual de su conocimiento.

La solución de problemas o ejecución de tareas es colaborativa, en el sentido de que es necesario compartir información de forma que permita al grupo de agentes como un todo, producir una solución o exitosamente realizar una tarea global. El grupo de agentes es distribuido, ya que tanto datos como control están lógicamente y geográficamente distribuidos.

En un Sistema Multiagente un grupo de agentes coopera (con sus capacidades de **percepción, comunicación y decisión**) en la solución de problemas globales.

Uno de los problemas más fuertes a los que se enfrenta el planificador en esta clase de universos es la concurrencia ocasionada por la compartición de recursos y las acciones que compiten en el espacio y en el tiempo [RAMOS95]. La ejecución de un plan predefinido el cual sirve para alcanzar un objetivo, exige el **intercambio de información**, lo cual demanda recursos para su comunicación en el sistema en que se encuentra, de tal suerte que se necesita una forma controlada de regular los posibles actos de comunicación que garanticen ese intercambio de información, y provean el medio eficaz para resolver las tareas comunes[LINAN95].

## 2. Antecedentes

Uno de los casos típicos de aplicación de los Sistemas Multiagentes son las celdas flexibles de manufactura.(SFM). En tales sistemas, las características de flexibilidad deben ser entendidas como la capacidad que tiene el sistema para adaptarse a diversas tareas bajo configuraciones y condiciones diferentes. En esencia, el sistema debe satisfacer requerimientos por el lado de las características físicas de los dispositivos que intervienen en la tarea, como por el lado de la programación del sistema. Este último aspecto atiende la planificación y la ejecución. Este trabajo no se ocupa del problema de la planificación flexible en sí, sin embargo está ligado a través de los esquemas de comunicación necesarios para mantener el aspecto de flexibilidad deseado. La capacidad de planificación flexible en nuestro trabajo se ve reflejada en el hecho de obtener planes nuevos a partir de un conjunto de planes de referencia a través de un mecanismo de adaptación [RAMOS95], es decir, un sistema de planificación flexible debe ser respaldado por un sistema flexible de comunicación que en esta clase de sistemas debe reunir las siguientes características: Adaptarse fácilmente a cambios señalados por el plan a ejecutarse, Reconfigurarse, Ser tolerante a fallas y monitorear el plan a ejecutar.

El universo de aplicación donde se ensaya nuestro modelo de comunicación es una línea flexible de manufactura, la cual está configurada de la siguiente manera:

Cuatro agentes con sus mundos locales respectivos (ver figura 1).

Una banda transportadora

Para la simulación es necesario modelar cada uno de los elementos del sistema como un agente con sus capacidades de decisión, autonomía y comunicación necesarios para la solución del problema en un mundo multiagente.

### 3. Objetivo del trabajo

Nuestro problema se centra en la simulación de una red de comunicación entre agentes robóticos que soporte la planificación flexible y la tolerancia a fallas para una línea flexible de manufactura.:

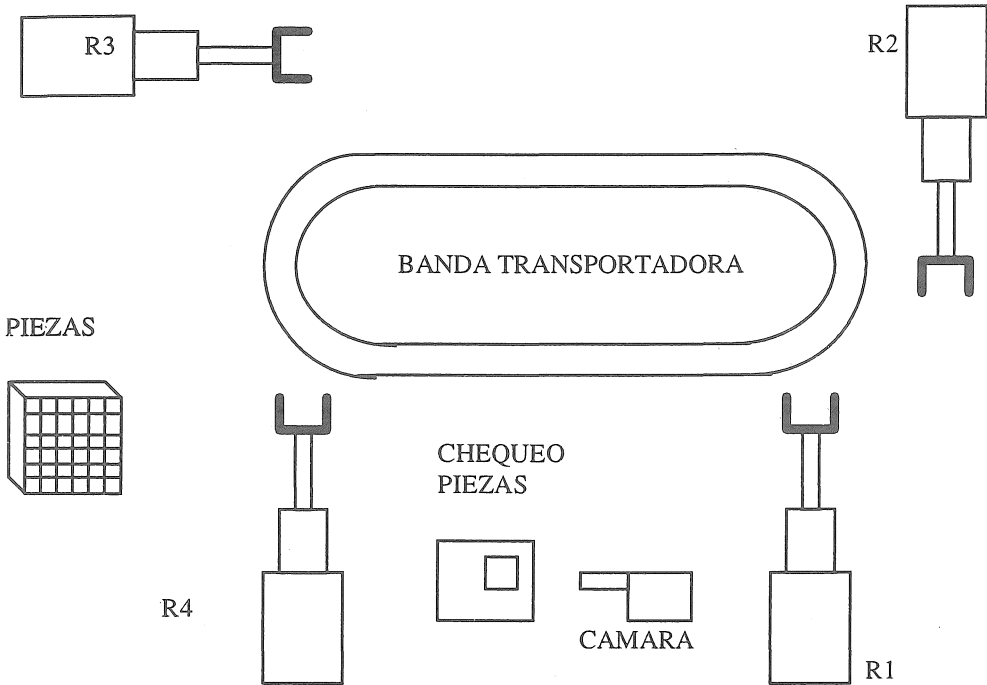


FIGURA 1. LINEA FLEXIBLE DE MANUFACTURA

De una manera global para lograr la implementación se necesitan las siguientes actividades:

- 1.) La planificación de la tarea. En esta parte se asume que el plan ha sido generado.
- 2.) Modelación de un agente autónomo.
- 3.) Modelación de un grupo de agentes.
- 4.) Modelo de la comunicación entre agentes.

#### 4. ARQUITECTURA PROPUESTA.

Dado que estamos interesados en la implementación práctica de los agentes, hemos investigado que la programación orientada a objetos concurrente, debe ser la base para la modelación de agentes autónomos con sus esquemas de comunicación que le permitan la interacción con su mundo [BENVE94]. Esta es la primera aproximación hacia la programación orientada a agentes [ETZIO92] y [CORIA93]. Sin embargo encontramos una limitante, un objeto en su implementación tradicional, no tiene una función para realizar su autonomía. Los objetos cooperantes se consideran que forman un grupo pero la noción de grupo no se da [HADDA93], al menos en la concepción de objetos que se permiten con los lenguajes tradicionales. Entonces un agente autónomo debe ser un objeto concurrente, pero que cuente con su propio intérprete de mensajes para lograr autonomía [LANE91].

Para modelar un mundo multi-agente con la filosofía orientada a objetos, es necesario considerar ante todo características de comunicación, interacción y comportamiento las cuáles están implícitas en las características siguientes:

- **Autonomía:** necesita tener una función para controlar la ejecución de mensajes tomando en cuenta su estado interno o condición de su entorno. **Grupo de agentes:** un agente influye sobre el grupo enviando un mensaje a los otros agentes al mismo tiempo.

##### **El modelo de agente autónomo**

En sistemas multiagentes el orden de los mensajes que llegan de diferentes agentes no se garantiza, por lo tanto, un agente autónomo debe tener la capacidad de recibir los mensajes en cualquier orden. El agente necesita evaluar un mensaje antes de la ejecución, ya que el agente tiene que guardar su consistencia interna para procesar su tarea.

Es en esta característica entre otras, donde se puede señalar la diferencia con los sistemas tradicionales, ya que para realizar la actividad de evaluación se requiere asociarle características de inteligencia o autonomía.

Esquemáticamente la estructura de un agente debe ser como la de la figura 2.

Recordar que el interprete de mensajes debe estar intimamente ligado al plan a ejecutar. Es aquí donde mantiene uno de los puntos de liga con el planificador.

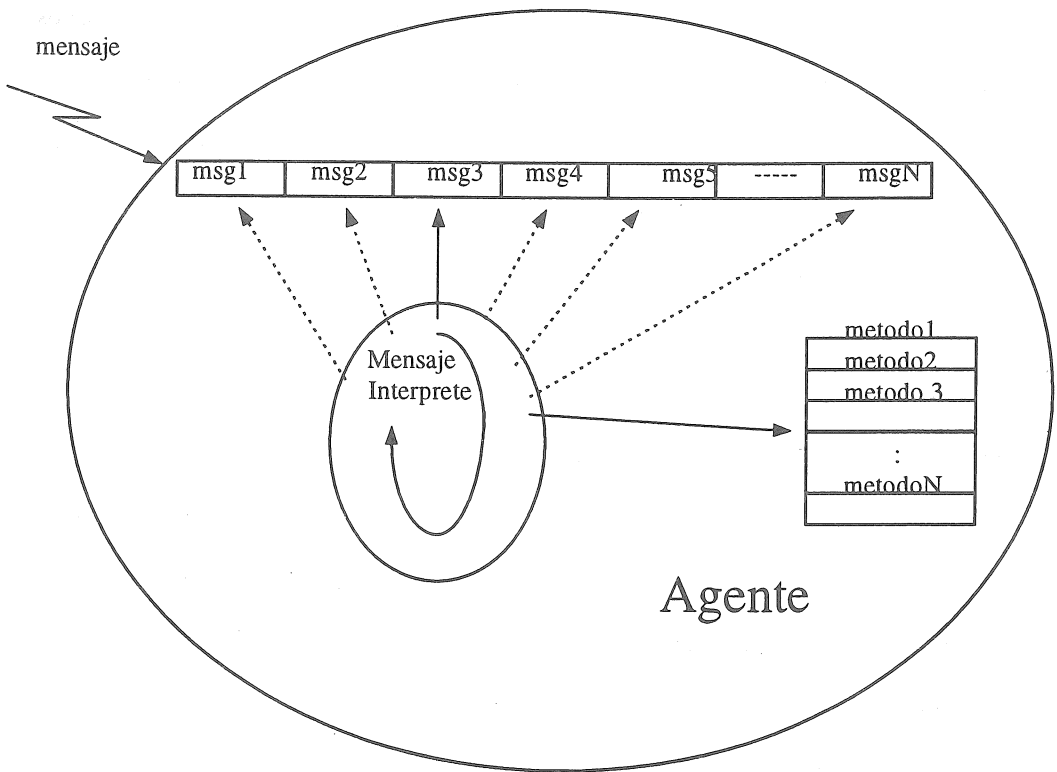


FIGURA 2. ESTRUCTURA DE UN AGENTE.

Un agente puede enviar un mensaje a otro especificando su nombre como sigue  
(send nombre-de-agente mensaje)

### **Intérprete de mensajes**

Cuando un mensaje llega a un agente, el mensaje causa una interrupción que es añadida dentro de la cola de mensajes, luego el intérprete de mensajes busca por un mensaje importante en la cola, lo recupera y le asigna los métodos adecuados, esto es lo que provee la autonomía. Una tarea asignada al buffer del agente es guardar los valores en el buffer interno. El procedimiento del buffer es la aceptación de : put or get mensaje. Debido a que los agentes tienen un tamaño finito de buffer, put o get message causa errores cuando el buffer está lleno o vacío respectivamente. Como resultado, el intérprete de mensajes puede controlar el comportamiento de un agente y mostrar su autonomía.

### **Funciones usadas en el intérprete de mensajes**

El rol básico e importante de un intérprete de mensajes es manejar la cola de mensajes. Para buscar un mensaje en la cola de mensajes es deseable contar con las siguientes primitivas.

receive: recupera el primer mensaje de la cola

receive-block: Esta primitiva es una versión de recibir pero en block.

receive-peek Esta primitiva sólo se asoma al primer mensaje en la cola de mensajes.

message-queue-length Regresa el número actual de mensajes contenidos en la cola de mensajes.

receive-if *conditions actions* Esta primitiva busca en la cola de mensajes y trae el primer mensaje que satisface las condiciones

receive-if-all *conditions actions* Esta primitiva es una versión extendida de receive-if.

Cuando usamos un intérprete de mensajes, múltiples mensajes pueden ser procesados usando las operaciones receive-if or receive-if-all. Entonces, el orden de llegadas de mensajes no es importante debido a que dos mensajes no son procesados hasta satisfacer las *condiciones* en la expresión.

Lo anterior nos permite ligar el plan a realizar con la ejecución, tal como se muestra en [RAMOS95], de donde tomamos parte del plan representado en la figura 3, se puede ver que la ejecución es posible con las primitivas condicionales que toman en cuenta las acciones NA, N1, N2, N3, N4, DPB, TPB, TPC, DPE.

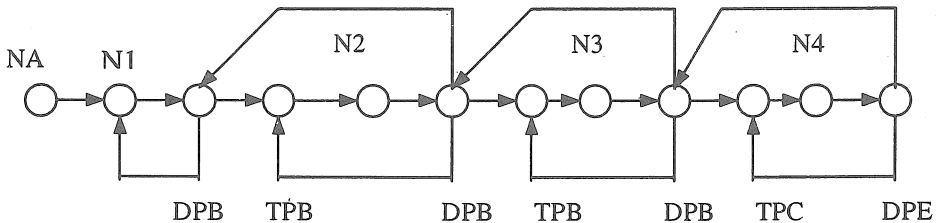


FIGURA 3. PLAN A EJECUTAR.

Donde N1, N2, N3 y N4 son acciones que ejecutan los robots R1, R2, R3, y R4 respectivamente; NA, nodo de arranque; DPB y TPB, deposita y toma la pieza de la banda respectivamente; TPC, toma la pieza de la camara y DPE deposita la pieza en el estante.

Un intérprete de mensajes puede ser pensado como administrador que maneja la ejecución de mensajes y provee autonomía a los agentes. En este modelo, puesto que hay sólo un actividad en un agente, los métodos en el mismo agente no se ejecutan en paralelo.

### El Modelo de Grupo de Agentes Autónomos

En el mundo real, hay muchos casos en donde un agente puede enviar un mensaje no sólo a un solo agente, sino también a múltiples agentes al mismo tiempo. En un grupo de agentes,

transmitir un mensaje, es considerado similar a emitir múltiples veces un mensaje a todos los agentes existentes en el mismo ambiente, que es el lugar donde existen los agentes. Usando la noción de un **ambiente**, un grupo es representado y un mensaje es emitido múltiples veces a todos los agentes del grupo. Por eso, la noción de un grupo de agentes es necesaria para poder representar un mundo multiagente como una compañía o una organización democrática.

### Operaciones de Envío Extendidas

Una operación de envío se usa cuando un agente necesita enviar un mensaje a un objetivo (agente destino), especificando el nombre del objetivo. Aunque una operación de envío apoye sólo la comunicación punto a punto, la operación se extiende para realizar envíos múltiples usando la noción de un ambiente como sigue:

`send-all (ambiente-nombre mensaje &clave :excepto-yo)`

`send-to (ambiente-nombre agente-nombre mensaje)`

Usando un ambiente, la comunicación grupal se realiza por un mensaje indirecto pasando por un ambiente.

Un ambiente puede ser creado o borrado como un agente usando las siguientes operaciones:

`new (ambiente-clase ambiente-nombre)`

`delete (destruir ambiente-nombre)`

`insert (agente ambiente-nombre)`

`exit (agente ambiente-nombre)`

`count (ambiente-nombre &clave :excepto-yo)`

`list (ambiente-nombre &clave :excepto-yo)`

## 5. SIMULACION

En la simulación del Sistema Flexible de Manufactura, se consideraron 5 etapas dentro del proceso de manufactura. La primera etapa consistió en tener una banda inicial en la que se entrega a un robot 1 cubos metálicos para ser moldeados en la etapas posteriores. El robot recibirá una orden de inicio con la cual depositará cada pieza en la banda común. En la segunda etapa, en el momento que exista en la banda común una pieza depositada por el robot 1, el robot-2 será capaz de tomar la pieza y llevarla hasta un torno para ser moldeada y enseguida ser depositada nuevamente en la banda común. En la tercera etapa en el momento que se halla depositado la pieza moldeada, el robot 3 será capaz de tomar esa pieza y llevarla hasta una fresa para ser perforada. Y finalmente en la cuarta etapa, cuando el robot 3 deposite la pieza perforada en la banda común, el robot 4 tendrá la capacidad de tomarla para llevarla a la fase de control de calidad y por último al depósito de producto final o a la basura.

Se consideró como una quinta etapa a la banda común. Cada fase estará dibujada en el monitor de una estación de trabajo mostrando lo que está pasando en ella.

Para el desarrollo de la simulación se aprovecharon las características que el sistema operativo UNIX ofrece y son: herramientas para la implementación de procesos como son los "forks"; las facilidades de memoria compartida para la toma de decisiones en la etapa de concurrencia; los sockets como elementos de comunicación entre procesos de UNIX .

Se usó el lenguaje C++ para la representación de los objetos tal como se describió en su arquitectura

Se utilizaron las herramientas de MOTIF y X11 para el desarrollo de la interfaz gráfica

Se usaron 5 estaciones de trabajo RS/6000 con el sistema operativo AIX versión 3.25

## Comunicaciones

Las comunicaciones fueron implementada usando sockets. Al ejecutar cada proceso, la estación de trabajo (agente) que lo realiza debe conocer el hostname de la estación de trabajo que funciona como maestro o coordinador. El coordinador no necesita esta información ya que aceptará las cuatro primera conexiones de agentes que le soliciten (considerando que son los agentes adecuados en el ambiente para completar el proceso de manufactura).

Se definió un proceso de graficación con el objeto de ir mostrando las acciones que está realizando cada uno de los agentes y la evolución en la ejecución de la tarea global.

Se comprobó que gracias al esquema de comunicación la coordinación de las actividades dentro del problema global se realizó satisfactoriamente.

## 6. Conclusiones y trabajos futuros

Con la ayuda de la simulación se demostró que la solución cooperativa de problemas se facilita con la implementación de agentes (objetos autónomos concurrentes) que estén soportados con un esquema de comunicación .

Se demostró que la autonomía de cada uno de los agentes de un grupo, ayuda a la toma de decisiones al momento de colaborar con actividades locales enfocada hacia una actividad global.

Como trabajos futuros se esta proponiendo al existencia de diferentes tipos de fallas y se está redefiniendo el concepto de los agentes con el propósito de ser lo suficiente robustos y poder enfrentar situaciones anómalas y tolerar las fallas.

Consideramos que las bases que ofrece el esquema de comunicación, se mantendrán aún cuando la solución de las tareas se vuelva más compleja o requiera actividades de negociación.

## 7. Bibliografía

- [BENVE94] "Synchronous Languages Provide Safety in Reactive System Design"  
Albert Benveniste, 1994, Control Engineering sep/1994
- [CORIA93] "Stepwise Development of Correct Agents: A Behavioural approach based on  
Colored Petri nets". Michael Coria 93, Laboratoire MASI, Institut Balise  
Pascal/CNRS-, UA Universite de Paris, France
- [ETZIO92] "Building Softbots for UNIX (Preliminary report)".  
Oren Etzioni, Neal Lesh, R. Segal, 92, Dep. of Comp. Science, Univ of  
Washington.
- [HADDA93] "Knowledge about others agents in heterogeneous dynamics". Haddadi A.  
Sundemeyer K. Proc.of Int. Conf. on Intelligent and Cooperative Information  
System.
- [HUHNS87] "Distributed Artificial Intelligence" Research Notes in AI, Morgan Kaufmann,  
Huhns, M.N. editor. 1987.
- [LANE91] "Distributed Architectures for autonomus inderwater vehicles".  
Lane D.M.; Chantles M.J..IEEE Colloquium on autonomus guided vehicles.
- [LINAN95] "El Estado del Arte en Sistemas Multiagentes". ITESM, Campus Morelos,  
Candidato a Doctor, José Luis Liñán García, nov/1995.
- [NTUEN91] "An experiment in human-robot interaction during task execution".  
Ntuen, C.A.; Park E.H.; IEEE Internation Conference on System Man, and  
Cybernetics
- [RAMOS95] "Adapting Reference Plans to build new plans by using transformation operators in  
flexible manufacturing lines", F. Ramos, E. Aguillon, ITESM, Morelos  
Proceeding of the third IASTED international conference in Robotics and  
Manufacturing, 1995
- [SAMAN93] "Un modelo de planeación con ultiagentes aplicables a Sistemas Flexibles de  
Manufacturas". Samano, J; Ramos F.  
XXIII Reunión de Investigación y Desarrollo del ITESM, enero 1993.
- [SYCAR91] "Distributed constrained heuristic search".  
Sicara K; Roth S.F.; Sadeh N.; Fox M.S. 91 IEEE Transactions on Systems,  
Man and Cybernetics Vol.21 p.1446/61 Nov.-Dec. 1991
- [TUIJN93] "Distributed objects in a federation of autonomous Cooperating agents".  
Tuijnman, F.; Afsarmenesh, H. 93 Proceeding of International Conference on  
Intelligente and Cooperative Information Systems