CRITERIO DE MOVILIDAD PARA AGENTES SOFTWARE EN UNA ARQUITECTURA HÍBRIDA DE CONTROL DE ROBOTS MÓVILES

Juan Luis Posadas Yagüe
Universidad Politécnica de Valencia, Camino de Vera s/n, Valencia 46022, jposadas@disca.upv.es

José Luis Poza Luján
Universidad Politécnica de Valencia, Camino de Vera s/n, Valencia 46022, jopolu@disca.upv.es

José Enrique Simó Ten
Universidad Politécnica de Valencia, Camino de Vera s/n, Valencia 46022, jsimo@disca.upv.es

Resumen

En este artículo se presenta un criterio de movilidad para agentes software móviles que constituyen los componentes de una arquitectura híbrida para el control de robots móviles. En primer lugar, se justifica el uso de dichos agentes en este tipo de arquitecturas, caracterizadas por componentes que deben interactuar en un entorno dinámico accediendo a información que se encuentra distribuida entre los nodos del sistema. En segundo lugar, se establece el criterio para la movilidad de los agentes con el objetivo de cumplir los tiempos de ejecución de los mismos y de reducir las necesidades de comunicación en el acceso a los datos. Se propone un nuevo nivel semántico donde en la especificación de la arquitectura se indican los componentes o agentes a ejecutar pero no su ubicación física, ésta se determina automática y dinámicamente en función de las características del sistema. Cada agente podrá moverse allí donde las condiciones del entorno le sean más favorables.

Palabras Clave: Sistemas multiagente, Robots móviles, Sistemas de control distribuido, Sistemas de tiempo real.

1 INTRODUCCIÓN

Los sistemas multiagente (MAS) [2], caracterizados por la modularidad y ejecución independiente de los agentes, resultan adecuados para la especificación e implementación de arquitecturas híbridas para el control de robots móviles. Pueden establecerse relaciones entre los agentes software, elementos principales de los MAS, y los agentes físicos propios de arquitecturas para robots. De hecho, se observa claramente una implicación directa de las características de los paradigmas existentes en robótica móvil [5] con las características de los agentes móviles. Esto es debido a la autonomía que caracteriza a estos agentes y que permite estudiarlos como si se tratara de verdaderos agentes físicos. En este sentido, existen implementaciones de sistemas donde se asocia cada agente físico o robot a un agente software formando una sola entidad, de manera que el agente software proporciona la inteligencia y el robot la capacidad de interacción con el entorno real. Sin embargo, se puede realizar una abstracción mayor donde varios agentes móviles a través de la red pueden utilizar diferentes agentes físicos para la consecución de sus tareas como si se trataran de nuevos recursos disponibles en el entorno software distribuido. Desde este punto de vista, el diseño de una arquitectura general para la implementación de este tipo de sistemas, requiere la existencia de algún criterio que permita la movilidad, entre los nodos de la arquitectura, de los agentes software que además tendrán que poder acceder a la información distribuida de los sensores e indicar acciones a los actuadores. Si además se necesita reacción en tiempo real, como por ejemplo para poder evitar obstáculos, tendrán que coexistir diferentes tipos de agentes con diferentes restricciones temporales, las cuales deberán contemplarse en el establecimiento de los criterios de movilidad.

2 AGENTES EN ARQUITECTURAS DE CONTROL DE ROBOTS MÓVILES

Los agentes nos ofrecen en un amplio dominio de aplicaciones [2] tanto la habilidad de resolver problemas que hasta entonces no habían podido resolverse (por no disponerse de la tecnología adecuada o porque la utilización de la existente implicaba un coste excesivo a nivel de dificultad, tiempo, consumo, riesgo, etc.) como la habilidad de resolver problemas de una forma significativamente mejor (atendiendo al coste, sencillez, eficiencia, rapidez, etc.) que la forma utilizada hasta el momento.

2.1 JUSTIFICACIÓN DEL USO DE AGENTES

Existen una serie de características que cuando forman parte de la especificación de un sistema se suelen citar para justificar la adopción de una tecnología basada en agentes. Dichas características se describen en las secciones siguientes.

2.1.1 Sistemas reactivos

Cuando el sistema es reactivo existiendo una continua interacción con el entorno [7]. A este tipo de sistemas pertenecen los denominados sistemas complejos (*complex systems*) y los sistemas abiertos (*open systems*).

2.1.1.1 Sistemas complejos

Son sistemas donde, debido a su complejidad, la modularidad y la abstracción constituyen las herramientas más adecuadas para el desarrollo natural de su diseño e implementación.

Los agentes representan una poderosa herramienta para realizar sistemas modulares. Un sistema basado en múltiples agentes facilita la tarea de descomponer un problema en un determinado número de componentes más pequeños y sencillos, los cuales son más fáciles de desarrollar, especializados en resolver subpartes de dicho problema. Esta descomposición permite a cada agente utilizar la técnica más apropiada para resolver su problema particular, en lugar de tener que adoptarse forzosamente una técnica global para todo el sistema que podría no ser la óptima en la resolución de cada uno de los casos particulares. Desde este punto de vista, se aprecia la idoneidad del empleo de los agentes para la descomposición en subtareas de las tareas u objetivos asignados a un robot con arquitectura de control híbrida.

El concepto de agente también proporciona una útil abstracción del sistema. De forma que el programador puede concebir el sistema complejo como una sociedad cooperativa de elementos autónomos especializados en la resolución de problemas concretos y sencillos. Mediante la cooperación de estos elementos autónomos se abordan y se resuelven objetivos y problemas cada vez más complejos. Las arquitecturas híbridas para el control de robots móviles suelen organizarse en diferentes niveles formados por componentes independientes que tienen que cooperar entre ellos. En este sentido, los agentes también resultan idóneos para especificar e implementar dichos componentes.

2.1.1.2 Sistemas abiertos

Son sistemas cuya estructura cambia de forma dinámica. Se caracterizan porque los componentes que lo forman no son conocidos a priori, pueden cambiar a lo largo del tiempo y pueden ser heterogéneos (en el sentido en que pueden implementarse por diferentes personas, en diferentes momentos y usando diferentes técnicas y herramientas de software). En este sentido, los agentes son adecuados para este tipo de sistemas por sus características de autonomía y ejecución independiente que permiten el desarrollo individual de los mismos así como su integración de forma dinámica en el sistema. Las características de los sistemas abiertos se relacionan directamente con las arquitecturas híbridas para el control de robots móviles donde no hubiera que especificarse a priori ni todos los componentes software ni su ubicación física, sino que éstos se adaptaran dinámicamente a un entorno también dinámico.

2.1.2 Sistemas distribuidos

Cuando la información, el control o los recursos se encuentran distribuidos. En este caso, el uso de agentes permite su distribución en función de la ubicación de los datos o recursos. Una arquitectura adecuada para el control de robots móviles debería ser híbrida y distribuida acorde con la propia naturaleza distribuida de la información sensorial y comportamientos.

2.1.3 Sistemas interoperativos

Cuando el sistema posee elementos que deben interactuar con otros elementos de otros sistemas. Los agentes pueden facilitar la interacción entre sistemas mediante el uso de lenguajes comunes de comunicación entre agentes. En una arquitectura híbrida multinivel tiene que existir una interacción entre los componentes de los distintos niveles que la forman.

2.1.4 Conclusiones

En todos estos casos, la utilización de agentes suele proporcionar una forma natural de modelar el sistema e implementarlo, resultando, por lo tanto, idóneos en la especificación e implementación de arquitecturas híbridas para el control de robots móviles. Los agentes constituirán los componentes de tales arquitecturas y dependiendo de la función a realizar o ubicación dentro de la arquitectura (nivel al que pertenezcan) tendrán que poseer determinadas características. Dichas características permiten realizar una clasificación de los distintos tipos de agentes existentes.

2.2 CLASIFICACIÓN DE LOS AGENTES

Los agentes se pueden clasificar [6] atendiendo a diversos factores.

Según su movilidad:

- Estáticos: permanecen donde son creados.
- Móviles: pueden moverse a través de la red.

Según su forma de actuar:

- Deliberativos: derivan del paradigma de pensamiento deliberativo. Los agentes procesan un modelo de razonamiento simbólico interno que les permite planificar sus actuaciones y negociar con otros agentes para alcanzar sus objetivos.
- Reactivos: tienen su origen en las investigaciones llevadas a cabo por Brooks (1986) y Agre y Chapman (1987). Estos agentes no tienen ningún modelo simbólico interno de razonamiento, sino que actúan según el modo de comportamiento estímulo-respuesta.

En general un agente puede clasificarse por la característica que más destaque e influya en su comportamiento e implementación.

2.3 AGENTES MÓVILES

Los agentes móviles son objetos móviles con autonomía flexible que pueden tomar decisiones en cuanto a su movilidad (dónde y cuándo viajar, etc.).

En base al modelo de movimiento empleado, pueden distinguirse dos tipos de agentes móviles:

- Agentes con sólo movimiento de código (*just code*). En el transporte de los agentes móviles sólo se realiza el transporte del código necesario para su ejecución en otro lugar, no se transportan los datos correspondientes al contexto o estado de ejecución. Ejemplos [3]: TCL, JAVA con RMI (*Remote Method Invocation*).
- Agentes con movimiento de código y contexto de ejecución (*not just code*). En este caso puede interrumpirse la ejecución de un agente en un punto concreto y proseguir, tras realizarse el movimiento a otro ordenador, con la ejecución de dicho objeto desde el punto en el que se interrumpió. Ejemplos [3]: TELESCRIPT, OBLIQ.

Los agentes móviles también se diferencian por el modelo de comunicación utilizado para interactuar entre ellos:

- Agentes con sólo comunicación local. Para que un agente se comunique con otro agente en otra máquina tiene que viajar a esa máquina. Ejemplo: TELESCRIPT.
- Agentes con comunicación por red. No es necesario que un agente se desplace al lugar dónde se encuentra el agente con el que tiene que comunicarse. En este caso el transporte de un agente implica, además del movimiento de código y contexto, el movimiento de las conexiones. Ejemplo: OBLIQ.

2.3.1 Ventajas de los agentes móviles

Algunas de las ventajas que pueden obtenerse con los sistemas de agentes móviles [4] se describen a continuación.

Reducir la carga de la red. Permiten que los agentes se comuniquen en el mismo nodo y no a través de la red, reduciéndose de esta forma las transferencias de datos por la red. También, cuando hay que procesar una gran cantidad de información ubicada en nodos remotos, puede transferirse el código necesario allí donde se encuentren los datos y procesarse de forma local. La idea es mover el código en lugar de los datos si éstos ocupan un mayor volumen.

Vencer la latencia de la red. Sistemas de tiempo real estricto, tales como robots, necesitan responder a cambios en su entorno en un tiempo determinado. Realizar un control de los robots a través de la red puede no ser aceptable por las latencias introducidas por ésta. Los sistemas móviles ofrecen una solución al permitir la transmisión al robot, desde un nodo central, del código necesario para controlarlo de forma local.

Adaptación dinámica al entorno. Los agentes móviles pueden percibir su entorno de ejecución y reaccionar autónomamente a cambios. De esta forma, múltiples agentes móviles tienen la habilidad de distribuirse a través de los nodos de la red manteniendo siempre la configuración óptima para la resolución de un determinado problema. Cada agente puede moverse al nodo donde las condiciones del entorno le sean más favorables para su ejecución.

Estas características son adecuadas en una arquitectura híbrida cuyos componentes puedan, por un lado, incorporase al sistema de forma dinámica y, por otro lado, moverse a través de los diferentes nodos con el objetivo de acceder a la información del sistema reduciendo el coste temporal de las

comunicaciones. Además, el uso de agentes móviles permite emplear técnicas de delegación de código para controlar los robots mediante el envío, en primer lugar, del código necesario a los nodos distribuidos y, posteriormente, la ejecución del mismo de forma local, evitando así los problemas de latencia de la red en la ejecución del código de control, y pudiéndose acotar su tiempo de ejecución para el estudio y cumplimiento de las restricciones temporales del sistema.

3 ARQUITECTURA SC-AGENT

La arquitectura basada en agentes propuesta [9, 10, 11] se denomina SC-Agent y está formada por tres bloques: un nivel deliberativo, un nivel reactivo y un sistema de comunicaciones interfaz entre ambos niveles (fig. 1).

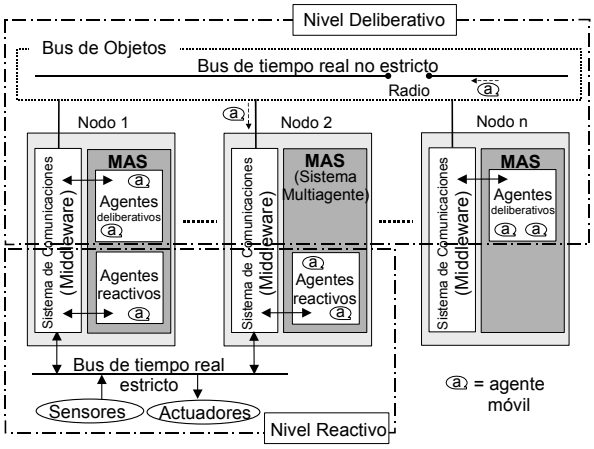

Figura 1: Estructura y conexiones de la arquitectura distribuida propuesta. Buses de tiempo real estricto y no estricto requeridos por el sistema de comunicaciones

El nivel deliberativo es un sistema de tiempo real no estricto que representa un nivel de conocimiento superior y ejecuta procesos de planificación basándose en el estado de un modelo simbólico interno del entorno. Como resultado de la planificación, los objetivos se dividen en patrones de comportamiento [1, 5] que son enviados al nivel reactivo. Cada patrón de comportamiento está implementado a través de uno o varios agentes *software*.

El nivel reactivo es un sistema de tiempo real estricto conectado con los sensores y los actuadores, tiene que reaccionar en un tiempo acotado ante los diferentes valores de los sensores (estímulos). El nivel reactivo recibe del nivel deliberativo los patrones de comportamiento o agentes, que se ejecutan concurrentemente y que usan la información sensorial para el cálculo de las acciones que llevarán a cabo los actuadores.

El sistema de comunicaciones (SC) [8] proporciona la infraestructura *hardware* y *software* necesaria para el acceso a los sensores y actuadores y para la interacción entre los distintos componentes de la arquitectura. Constituye una interfaz entre los niveles reactivo y deliberativo mediante la cual, el nivel deliberativo envía los patrones de comportamiento o agentes al nivel reactivo y accede a la información sensorial para incorporarla en el modelo interno simbólico del entorno. El diseño del sistema de comunicaciones contempla los diferentes requerimientos temporales de los niveles reactivo y deliberativo.

Para permitir las comunicaciones en el nivel reactivo, los nodos reactivos tendrán que estar interconectados mediante un bus de tiempo real estricto. Por otro lado, para permitir las comunicaciones en el nivel deliberativo, los nodos deliberativos tendrán que estar interconectados mediante un bus de tiempo real no estricto.

Ambos buses son necesarios debido a las diferencias existentes entre los requerimientos temporales de los niveles reactivo y deliberativo. El nivel reactivo necesita un bus de comunicaciones [12] que garantice el tiempo de acceso y la transmisión de las tramas de datos para responder en un tiempo acotado a los cambios del entorno. Por su parte, el nivel deliberativo puede usar un bus sin estas restricciones que por el contrario permita comunicar estructuras de datos más complejas (como supondría la propia transmisión de los agentes).

4 CRITERIO DE MOVILIDAD DE LOS AGENTES

En el establecimiento de un criterio de movilidad de los agentes, tienen que considerarse distintas necesidades de control temporal.

Necesidad del control temporal de la ejecución de los agentes. Debe asegurarse la ejecución de aquellos agentes reactivos cuya omisión pondría en peligro al sistema. En un sistema donde no se conoce a priori ni el número ni la ubicación de los procesos o agentes que tienen que ejecutarse, características que dependen de la dinámica del sistema, no pueden calcularse los tiempos correspondientes a los "peores casos" de ejecución necesarios para realizar un estudio de la planificabilidad del sistema que permita asegurar los tiempos periódicos de ejecución de los procesos. Sin embargo, sí pueden medirse los tiempos de ejecución reales, lo cual permite la adaptación dinámica de los agentes en función de sus

restricciones temporales y tiempos reales de ejecución (por ejemplo, si un agente no consigue ejecutarse en un tiempo inferior o igual al requerido en su especificación, podría moverse a otro nodo del sistema distribuido donde se cumplieran dichas características temporales). Para ello, cada agente del sistema debe poseer como restricción temporal un periodo de ejecución que ha de cumplirse con mayor o menor precisión dependiendo de la importancia del agente (reactivo o deliberativo). Por ejemplo, si el sistema no pudiera cumplir los periodos de ejecución de los agentes necesarios para que el robot vaya a una determinada velocidad esquivando los obstáculos que encuentre en su camino, podría optar por reducir la velocidad (con la consiguiente degradación de prestaciones pero asegurando la integridad del sistema mediante el uso de periodos de ejecución mayores) o podría optar por delegar a otros nodos la ejecución de agentes no críticos (en caso de existir) para así asegurar la ejecución de los estrictamente necesarios en ese momento.

Necesidad del control temporal del acceso a los datos. El control temporal del acceso a los datos implica el control del tiempo requerido y empleado en el acceso a los datos del sistema. Los agentes también deberán caracterizarse con los tiempos máximos admisibles en el acceso a los datos que requieren. En un bus de tiempo real estricto puede estimarse el tiempo de transmisión y acceso a los datos correspondiente al "peor caso" mediante un estudio previo de la planificabilidad del acceso al bus. Sin embargo, en un bus de tiempo real no estricto no puede calcularse dicha estimación, aunque sí pueden obtenerse los "retrasos reales" en el acceso a los datos. El "retraso real" de un dato proporciona su antigüedad. Utilizar "retrasos reales" implica, al no disponer a priori del tiempo correspondiente al "peor caso", no poder planificar el momento de recepción de los datos. Sin embargo, utilizar "retrasos reales" permite al agente, en el momento de la recepción de los datos, validar los datos considerando su antigüedad y adaptarse dinámicamente a los tiempos de recepción reales (por ejemplo, moviéndose el agente a otro nodo donde los tiempos de recepción sean menores). Para ello, es necesario que el sistema proporcione la antigüedad de los datos disponibles en el mismo.

Como conclusión, los agentes deberán poder determinar dinámicamente el grado de adecuación al nodo en el que residen mediante el cálculo de algún índice de medida. Para dicho cálculo, es necesario el control temporal tanto de la ejecución de los agentes como del acceso a los datos caracterizándolos con su antigüedad. El índice de medida empleado se ha denominado "Índice de Comodidad" y su formulación se describe en la sección siguiente.

4.1 ÍNDICE DE COMODIDAD

Se ha denominado "Índice de Comodidad" a la medida en la que se basan los agentes para decidir si el nodo donde residen es adecuado para su ejecución o por el contrario deben moverse a otro nodo distinto.

La ubicación de cada agente en un computador u otro depende de varios factores:

En función de las restricciones de tiempo real de su tarea. Si el sistema no puede cumplir el periodo de ejecución del agente, éste podría moverse a otro nodo donde sí se cumpliera su periodo de ejecución. A diferencia de los agentes deliberativos, el no cumplimiento del periodo de ejecución de un agente reactivo podría ser crítico en la integridad del sistema.

En función de la antigüedad de los datos que reciba. Un agente ejecutándose en un computador con acceso directo al bus de tiempo real estricto recibirá los valores transmitidos mucho antes que un agente ejecutándose en otro computador conectado al sistema a través del bus de tiempo real no estricto. Por ello, un agente que forme parte del nivel reactivo probablemente deberá ejecutarse en un nodo con acceso directo al bus de tiempo real estricto, para así minimizar los tiempos tanto de recepción de la información como de actuación y poder realizar un control temporal más preciso evitando las estimaciones realizadas en las transmisiones por buses de tiempo no real. Sin embargo, un agente deliberativo perfectamente podrá ejecutarse en un nodo donde la información temporal de los valores que reciba no sea tan precisa como antes aunque sí con una antigüedad estimada adecuada para la realización de su tarea. Todo ello implica que para cada agente habrá que especificar, como parte de sus restricciones temporales, la máxima antigüedad admisible para cada uno de los datos que utilice en su procesamiento.

La formulación del "Índice de Comodidad" (I_C) viene dada por la siguiente expresión (1):

$$I_C = Min\left(I_{Act}, I_{Dat}\right) \qquad (1)$$

donde I_{Act} denota un "Índice de Actividad" que es igual a (2):

$$I_{Act} = \frac{t_{MAXComp} - \left(t_{Esp} + t_{\Pr oc} + t_{Com}\right)}{t_{MAXComp}} \qquad (2)$$

siendo:

$t_{MAXComp}$: restricción temporal del agente que indica el máximo tiempo de computación o ejecución

permitido para el procesamiento de la tarea asignada. Este valor se corresponderá con el periodo de ejecución del agente.

t_{Esp}: tiempo de espera real correspondiente al retraso producido en la ejecución del agente debido a interrupciones del sistema o expulsiones por tareas más prioritarias del sistema.

t_{Proc}: tiempo real correspondiente al procesamiento de la tarea asignada al agente.

t_{Com}: tiempo real correspondiente al retraso producido en la ejecución del agente debido a las tareas de comunicación (por ejemplo, en su caso, para el acceso a los datos).

e I_{Dat} denota un "Índice de Acceso a los Datos" que es igual a (3,4):

$$I_{Dat} = Min\left(\frac{t_{MAXAnt}(d) - t_{Ant}(d)}{t_{MAXAnt}(d)}\right) \forall d \in L \qquad (3)$$

$$L = \{D_1, D_2, ..., D_n\} \qquad (4)$$

siendo:

n: número de datos que el agente requiere en su ejecución.

L: lista con el conjunto de datos (D_1, D_2, ..., D_n) que el agente requiere en su ejecución.

$t_{MAXAnt}(d)$: tiempo máximo de antigüedad del dato d para que sea válido en el procesamiento del agente.

$t_{Ant}(d)$: tiempo real de antigüedad del dato d.

El "Índice de Actividad" de un agente indica si se están cumpliendo las restricciones de máximo tiempo de cómputo para dicho agente. Para ello, se tiene que contabilizar el tiempo real de ejecución del agente mediante la suma de su tiempo de procesamiento más los retrasos correspondientes a los tiempos de espera por expulsiones e interrupciones y los tiempos de espera por las comunicaciones. Cuando dicha suma supera el valor máximo permitido, el valor del "Índice de Actividad" es negativo indicando el incumplimiento de las restricciones temporales establecidas. Cuanto menor sea el índice mayores serán los retrasos.

El "Índice de Acceso a los Datos" de un agente indica si se están recibiendo los datos con una antigüedad adecuada para el procesamiento del agente. Para ello tiene que calcularse, para cada dato, la diferencia entre la máxima antigüedad permitida y la antigüedad real en ese momento. El valor que establece el "Índice de Acceso a los Datos" viene dado por la menor de las diferencias en proporción calculadas. Un "Índice de Acceso a los Datos" negativo indica que existe al menos un dato que no es válido para el procesamiento del agente. Cuanto menor sea el índice mayor es la antigüedad del dato.

El "Índice de Comodidad" se obtiene tomando el menor de los dos índices ("Índice de Actividad" o "Índice de Acceso a los datos").

Cuanto mayor es el valor del "Índice de Comodidad" ($I_{Comodidad}$), mayor es la adecuación del agente al nodo donde está ejecutándose. El valor máximo para $I_{Comodidad}$ es 1, y se corresponde con el caso ideal donde el coste temporal de ejecución del agente y acceso a los datos es despreciable y, por tanto, igual a 0. El rango de valores entre 0 y 1 indican un $I_{Comodidad}$ adecuado, lo cual significa que el agente cumple con sus restricciones temporales (periodo de ejecución y máxima antigüedad admisible para los datos) en el nodo donde está ejecutándose. Cuanto más cerca de 1 se encuentra el valor del $I_{Comodidad}$, más holgura existe entre los valores reales de los tiempos y los máximos admisibles. Un $I_{Comodidad}$ con valor negativo significa que el agente no está cumpliendo con sus restricciones temporales en el nodo de ejecución. En este caso el agente podría decidir moverse a otro nodo. Cuanto menor sea el valor de $I_{Comodidad}$, mayores serán los retrasos en el cumplimiento del periodo de ejecución del agente o en la antigüedad de los datos recibidos.

Cada agente, en función del valor de su "Índice de Comodidad", podrá decidir si quedarse o no en el nodo donde reside (por ejemplo: si $I_{Comodidad} > 0$ → permanecer en el nodo, si $I_{Comodidad} < 0$ → moverse a otro nodo). En este caso es el propio agente quien decide qué hacer, sin embargo, el sistema también podría realizar tareas de reajuste de ejecución de agentes en cada nodo en función de los mismos "Índices de Comodidad". Podría existir un agente deliberativo que se encargara de esta función.

5 CONCLUSIONES

La movilidad introduce más expresividad en el sistema. El hecho de que los agentes sean móviles para poder viajar entre los nodos les ofrece la posibilidad de encontrar dinámicamente su ubicación óptima. Esta idea introduce un nuevo nivel semántico en la ejecución de código donde se indica qué hay que ejecutar pero no dónde. La ubicación física de los componentes a ejecutar se realizará en el sistema de forma dinámica e inteligente adaptándola en tiempo real a las necesidades y cambios del mismo. Todo ello implica la necesidad de un índice de

medida que indique la adecuación del agente en el nodo donde está en ejecución o si, por el contrario, requiere moverse a otro nodo. Para la especificación y el cálculo de este índice es fundamental la incorporación al sistema de un control temporal en la ejecución de los agentes y en el acceso a los datos.

Referencias

[1] Arkin, R., (1998). Behavior-Based Robotics. MIT Press.

[2] Jennings, N.R., Wooldridge, M.J., (1998). Agent Technology. Springer.

[3] Karnit, N.M., Tripathi, A.R., (1998). Design Issues in Mobile-Agent Programming Systems. IEEE Concurrency July-September 1998, pp. 52-61.

[4] Lange D.B., Oshima, M., (1999). Seven Good Reasons for Mobile Agents. Communications of the Association of Computing Machinery, v. 42, no. 3.

[5] Murphy, R.R., (2000). Introduction to AI Robotics. MIT Press.

[6] Nwana, H., Lee, L., Jennings, N., (1996). Coordination in Software Agent Systems. BT Technology Journal, 14(4), 1996.

[7] Pnueli, A., (1986). Specification and development of reactive systems. Information Processing 86. Elsevier, North-Holland.

[8] Posadas, J.L., Perez, P., Simo, J.E., Benet, G., Blanes, F., (2002). Communications Structure for Sensory Data in Mobile Robots. Engineering Applications of Artificial Intelligence 15 (2002) pp. 341-350.

[9] Posadas, J.L., Simo, J.E., Poza, J.L., Pérez, P., Benet, G., Blanes, F., (2003). Una arquitectura para el control de robots móviles mediante delegación de código y sistemas multiagente. IV Workshop de Agentes Físicos - WAF'03. Alicante.

[10] Posadas Yagüe, Juan Luis (2003). Arquitectura para el control de robots móviles mediante delegación de código y agentes. PhD Thesis. Polytechnic University of Valencia.

[11] Posadas, J.L., Simo, J.E., Blanes, F., Benet, G., Poza, J.L., Albero, M., (2004). An architecture to control mobile robots by means of code delegation and multi-agent systems. 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles - IAV2004. Lisbon, Portugal.

[12] Tindell, K.W., Hansson, H., Wellings, A.J., (1994). Analysing Real-Time Communications: Controller Area Network (CAN). Proceedings of IEEE RealTime Systems Symposium RTSS'94. F. Jahanian and K. Ramaritham, eds., IEEE Computer Society Press, pp. 259-263.