

# Sistemas multi-agente robóticos: Revisión de metodologías

## Multi-agent robotic systems: Methodologies review

Marcela Vallejo V., Ing.<sup>1</sup>; John F. Ochoa G., Ing.<sup>1</sup>; Jovani A. Jiménez B., PhD.<sup>2</sup>

1. Universidad de Antioquia, Colombia

2. Grupo de Investigación: Inteligencia Artificial en Educación, Universidad Nacional de Colombia  
 [{mvallejov,jfochoaster}@gmail.com](mailto:{mvallejov,jfochoaster}@gmail.com), [jajimen1@unal.edu.co](mailto:jajimen1@unal.edu.co)

Recibido para revisión 30 de Septiembre de 2009, aceptado 23 de Octubre de 2009, versión final 15 de Diciembre de 2009

**Resumen**—El presente artículo brinda una visión general del estado del arte de los sistemas multi-agente robóticos. Inicialmente se hace un recuento del desarrollo del área, los problemas que enfrenta, para finalmente describir algunas de las metodologías que se han utilizado para su diseño.

**Palabras Clave**—Sistemas Multi-Agente, Metodologías de Diseño de MAS, Entornos Cooperativos Robóticos, Enjambres Robóticos.

**Abstract**—This article presents a general view of the state of the art of the robotic multi-agent systems. First of all, it is done an overview of the development in this matter, the problems that faced and finally it is described some methodologies that have been used for its design.

**Keywords**—Systems, Multi-agent Systems, BESA, JME, SmartPhones.

### I. INTRODUCCIÓN

Los sistemas multi-agente (MAS) están siendo usados ampliamente para resolver problemas relacionados con la solución colectiva de tareas: Trabajo cooperativo apoyado en computador, administración de redes, procesos de control distribuido, coordinación de colmenas de insectos, entre otros [1], [2]. Todos estos problemas comparten la misma dificultad de diseño: como llegar de la especificación global del problema a los comportamientos individuales que deben llevar a cabo los agentes en el sistema para realizar una tarea.

El objetivo de cualquier metodología orientada a agentes es brindar asistencia en todas las fases del ciclo de vida de una aplicación basada en agentes, incluyendo su mantenimiento.

En el desarrollo de metodologías para el diseño de sistemas multi-agente, los investigadores han enfocado sus esfuerzos en extender las metodologías existentes. Estas extensiones se han realizado principalmente sobre dos áreas: sobre las metodologías orientadas a objetos (Object Oriented) y sobre las de ingeniería de conocimiento (Knowledge Engineering) [3].

Recientemente los sistemas de múltiples robots se han propuesto como una alternativa para el desarrollo de sistemas robustos, adaptables y flexibles. Se han visionado muchas aplicaciones que incluyen: mantenimiento de plantas nucleares, rescate de desastres, administración de reservas forestales, sistemas de vigilancia, entre otros (Figura 1) [4].



Figura 1. Enjambre de robots Khepera.

El artículo está distribuido de la siguiente manera: el capítulo dos resume el desarrollo de la robótica cooperativa hasta el nacimiento del RoboCup, el capítulo tres define algunos términos clave en la literatura, el capítulo cuatro muestra varios de los problemas del área y finalmente el capítulo cinco describe algunas de las metodologías que existen para el desarrollo de sistemas multi-agente.

## II. BREVE HISTORIA DE ROBÓTICA COOPERATIVA

Los inicios del trabajo en el área los podemos encontrar en el trabajo de Fukada, Nadagawaka, Kawauchi & Buss [5] en el cual describen un acercamiento a la coordinación de múltiples unidades robóticas homogéneas y heterogéneas. Caloud et al. [6] y Norelis [7] aplican una arquitectura de control tradicional basada en planeadores a una tarea de movimiento de bloques implementada con dos robots. Kube et al. [8], [9] describen una serie de simulaciones de robots ejecutando una colección simple de comportamientos que son incrementalmente transferidos a robots físicos. Barman et al. [10] reportan un banco de pruebas preliminar para el estudio del control de múltiples robots en una tarea de jugar fútbol. Parker [11], [12] describe una arquitectura para compartir tareas basadas en comportamientos para el control de grupos de robots heterogéneos, y la demuestra en un grupo de cuatro robots físicos que ejecutan tareas de limpieza de desechos tóxicos y empuje de bloques. Donald et al. [13] reportan la base teórica para la implementación de manipulación de tareas cooperativa con un par de robots móviles. Altenburg [14], describe una variante de la tarea de búsqueda usando un grupo de robots LEGO controlados en un estilo reactivo y distribuido, y Beckers et al. [15], mostrando un grupo de cuatro robots agrupando discos inicialmente distribuidos aleatoriamente a través de comunicación puramente indirecta (stigmergetic communication).



Figura 2. Agentes Robóticos en la RoboCup.

En términos de cooperación y comunicación, el trabajo arriba citado cae en uno de los dos extremos del espectro: O usa extensivamente comunicación y cooperación explícita o no la usa en absoluto.

Dada la gran relevancia que ha adquirido esta área, se ha propuesto, desde el año de 1998, un torneo de fútbol robótico -RoboCup- el cual ha servido como un gran laboratorio para la validación de los modelos de robótica cooperativa [16] tanto a nivel de simulación como a nivel físico (figuras 2 y 3). A partir de este punto la mayoría de trabajos en el área caen dentro del ámbito de investigación que rodea la RoboCup.

## III. ASPECTOS CLAVES EN ROBÓTICA COOPERATIVA

Se pueden encontrar tres características esenciales en la investigación de la inteligencia colectiva: algoritmos para la coordinación y arbitraje de agentes, métodos de comunicación y arquitecturas que permitan la evolución dinámica de la estructura del sistema [4].

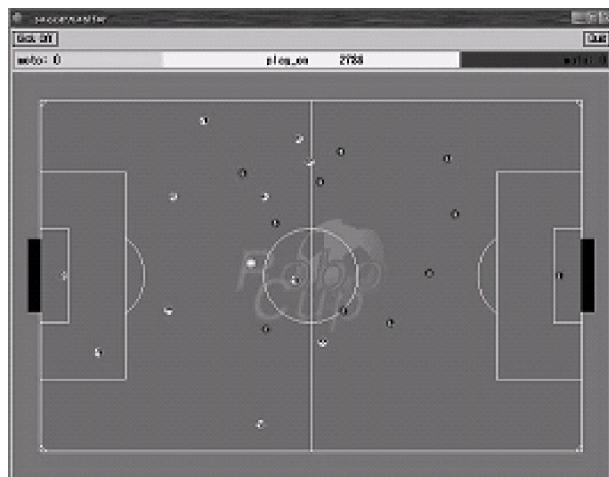


Figura 3. Simulador utilizado para la modalidad de agentes software de la RoboCup.

### A. Arbitraje entre agentes

En el trabajo con agentes, se debe de tener un mecanismo para la distribución de cargas y la coordinación del movimiento entre varios de ellos; debe considerarse que los mecanismos comunes de planificación para robots autónomos no son efectivos en esta área de investigación.

### B. Comunicación

La comunicación entre agentes es un aspecto muy importante para el alcance efectivo de acciones en comunidades de agentes, pero se hace difícil a medida que el número de agentes crece.

Se pueden identificar dos tipos de comunicación en estos sistemas: comunicación directa y comunicación mediante el entorno.

Aún se encuentran bajo investigación la cantidad óptima de agentes para que un sistema garantice la distribución efectiva de la información.

### C. Arquitectura dinámica configurable

El objetivo de las arquitecturas dinámicas es el brindar a las comunidades robóticas más flexibilidad y capacidad de adaptación al medio.

En cualquier área en la que se esté trabajando, es importante tener una idea clara del significado de los términos más importantes que se encuentran dentro de la literatura. Algunas definiciones comunes en el área de sistemas multi-agente son [17]:

- Agente: Se concibe como unidad funcional de los sistemas multi-agente al agente, el cual es una entidad física o abstracta que puede percibir su ambiente mediante sensores, es capaz de evaluar tales percepciones y tomar decisiones por medio de mecanismos de razonamiento simples o complejos, comunicarse con otros agentes para obtener información y actuar sobre el medio en el que se desenvuelven a través de ejecutores [18]. A partir de esto se espera que un agente actué de manera racional, intencional y autónoma en el momento preciso.

- Comportamientos y metas: El término "comportamiento" se ha popularizada como uno de los bloques constitutivos fundamentales en las comunidades de inteligencia artificial, control y aprendizaje. Puede definirse comportamiento como una ley de control para alcanzar y/o mantener una meta particular. Esta definición especifica que un comportamiento es un tipo de operador que garantiza una meta particular, cualquiera sea su tipo.

Un agente puede tener múltiples metas concurrentes, incluyendo al menos una meta a objetivo (meta que al ser alcanzada finaliza un comportamiento particular en un agente) y una o mas metas de mantenimiento (metas de diagnóstico y funcionamiento corriente del agente).

De los innumerables posibles comportamientos para un dominio dado, solo un pequeño subconjunto es relevante y deseable para alcanzar las metas del agente.

- Comunicación y cooperación: La comunicación es la forma más común de interacción entre agentes inteligentes.

Hay dos tipos de comunicación: Comunicación directa: Es un acto puramente comunicativo, con el único propósito de transmitir información, como un acto de habla, o la transmisión de un mensaje de radio. Este tipo de comunicación puede ser uno-a-uno o uno-a-varios, en cualquier caso es dirigida a receptores identificados.

Comunicación indirecta: Es aquella basada en el comportamiento observado de otros agentes y no en comunicación con ellos, puede consistir en dejar elementos en el entorno de manera que cuando otro agente los vea les asigne

algún contenido comunicativo por parte de otro agente.

La cooperación es una forma de interacción, usualmente basada en alguna forma de comunicación y que tiene el objetivo de alterar el estado de los agentes de manera que trabajen en función de un objetivo relevante para el sistema.

Hay dos modos de cooperación:

Cooperación explícita: se define como un conjunto de interacciones que implican intercambio de información o realización de acciones en beneficio de otro agente.

Cooperación implícita: Consiste en acciones que son parte del repertorio de comportamientos propio del agente, pero que tienen efectos en el mundo tales que ayudan a otros agentes a cumplir sus tareas.

- Interferencia y Conflicto: Interferencia hace alusión a alguna influencia que no permite que un agente alcance un objetivo. En las sociedades con iguales objetivos, la interferencia se manifiesta como una competencia por recursos compartidos. En sociedades con objetivos heterogéneos se presentan problemas mayores como puntos muertos y oscilaciones.

Identificamos dos tipos de interferencia: interferencia causada por multiplicidad: llamada competencia por recursos, resulta cuando múltiples agentes compiten por recursos como espacio, información u objetos. A medida que el número de agentes crece, aumenta este tipo de interferencia causando que el desempeño del equipo disminuya.

Interferencia causada por conflictos por objetivos relacionados: Se da cuando se distribuyen objetivos globales en subequipos. Estos objetivos pueden ser diferentes, pero pueden coincidir a nivel de subobjetivos, de manera que se sigue presentando la competencia por recursos iguales, esta vez para alcanzar objetivos globales diferentes.

Una manera de disminuir las interferencias es mediante la definición de reglas sociales que permitan optimizar la distribución de recursos.

## IV. PROBLEMAS RELACIONADOS A LOS ENTORNOS COOPERATIVOS ROBÓTICOS

Cuando se desea hacer la integración de agentes robóticos de manera que se logre un buen nivel de acople en la comunidad, surgen diversos problemas a varios niveles, así como soluciones diversas a estos problemas [19]:

### A. A nivel de arquitecturas de software

El desarrollo de arquitecturas para este tipo de sistemas presenta muchas dificultades. Algunas de las formas más frecuentes de abordar este tipo de inconvenientes son las siguientes:

- Uso de una arquitectura centralizada: Un solo agente procesa toda la información adquirida y envía las acciones que deben de tomarse a los otros agentes.
- Uso de una arquitectura distribuida: la información es recuperada por todos los agentes pero las decisiones las toma un agente central.
- Uso de una arquitectura totalmente distribuida: Cada robot toma sus decisiones de acuerdo a la información que adquiere.

#### **B. A nivel de planificación y distribución de tareas**

Los planes deben tener en cuenta la naturaleza distribuida de la tarea a realizar. Las diferentes tareas deben de distribuirse en los distintos agentes de acuerdo a sus destrezas. Los planes deben de considerar sincronización y comunicación entre los diferentes miembros del equipo envueltos en la tarea. Si un agente no puede llevar a cabo su tarea, deberá haber un mecanismo que la reasigne.

#### **C. A nivel de coordinación de tareas**

La introducción de la coordinación de tareas a un equipo de múltiples agentes introduce problemas relacionados a la distribución y mantenimiento del conocimiento, y los problemas propios de los sistemas de comunicación (ruido, ancho de banda limitado). Dado que la comunicación puede ser implícita o explícita, se debe de tener en cuenta el mecanismo mediante el cual los agentes generan eventos cuando necesitan comunicar algo de relevancia para otros agentes (por ejemplo cuando un agente necesita comunicar que ha alcanzado un objetivo). La comunicación es uno de los mecanismos mas importantes para la coordinación de tareas en un equipo por lo tanto es uno de los problemas claves a atacar en el desarrollo de este tipo de sistemas.

#### **D. A nivel de modelamiento distribuido del mundo**

Dado que se pueden distribuir diferentes capacidades sensóricas en los diferentes agentes, la representación obtenida del mundo puede ser mas rica que la obtenida con un solo agente. El problema que surge es como lograr una fusión sensórica eficiente de manera que se logre una integración adecuada de los datos.

### **V. METODOLOGÍAS PARA EL DESARROLLO DE SISTEMAS MULTI-AGENTE**

Dada la relevancia que ha adquirido el área de los sistemas multi-agente se han llegado a problemas de envergadura tal que se ha hecho necesario el desarrollo de metodologías que brinden soporte a todas las etapas de desarrollo de este tipo de problemas. A continuación se citaran algunas de las metodologías de uso más común.

#### **A. MASCommonKADS**

Extiende del CommonKADS aplicando ideas de metodologías orientadas a objetos (OOSE, OMT) y de ingeniería de protocolos para la descripción de los protocolos de los agentes (SDL, MSC96) [3], [20]. Considera sólo dos agentes básicos: El usuario y el sistema.

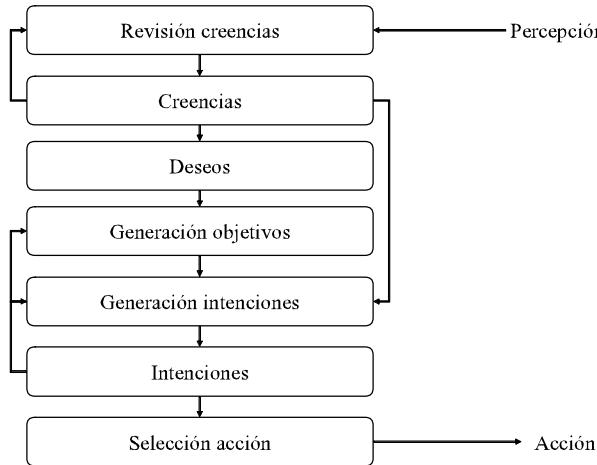
La metodología comienza con una fase de conceptualización que consiste en la recolección de los requisitos de usuarios y en un bosquejo de cómo será el sistema desde el punto de vista del cliente. Se definen siete modelos, para cada modelo se define un componente (entidad a ser modelada) y las relaciones entre los componentes. La metodología define una plantilla textual para la descripción de cada componente, y un conjunto de actividades para la realización de cada modelo. Una breve descripción de los modelos se da a continuación:

- **Modelo de Agente:** describe las características principales de los agentes, incluyendo las capacidades de razonamiento, destrezas (sensores/ efectores), servicios, objetivos, etc.
- **Modelo de Tareas:** describe las tareas (objetivos), llevadas a cabo por los agentes y la descomposición de dichas tareas.
- **Modelo de experiencia:** Contiene el conocimiento necesario para que los agentes puedan llevar a cabo sus tareas. Hay una distinción entre dominios, tareas, inferencia y conocimiento para la solución de problemas. También se diferencian los métodos de solución de problemas y los métodos de solución de problema cooperativos.
- **Modelo de Coordinación:** Alberga la información sobre las conversaciones entre agentes, sus interacciones y protocolos.
- **Modelo de Organización:** Describe la organización en la cual el sistema multi-agente será introducido y la organización de la sociedad de agentes -jerarquía de los agentes, relación de los agentes con su entorno, etc. -
- **Modelo de Comunicación:** Detalla las interacciones entre los humanos y el sistema y el factor humano para el desarrollo de interfaces de usuario.
- **Modelo de Diseño:** Agrupa los modelos anteriores y se divide en tres submodelos: diseño de la aplicación: composición y descomposición de los agentes obtenidos de la etapa de análisis de acuerdo a un criterio práctico. Diseño de la arquitectura: definición de los aspectos relevantes para el soporte de la red de agentes. Diseño de la plataforma: selección de la plataforma de implementación para los agentes del sistema.

#### **B. BDI**

Las arquitecturas conocidas como BDI (Belief-Desire-Intention) se inspiran en un modelo cognitivo proveniente del

área de la psicología [21]. Los agentes tienen un modelo del mundo que es modificado mediante las entradas sensóricas que posee el agente (Figura 4).



**Figura 4.** Arquitectura interna de un agente BDI.

La arquitectura de un agente BDI se compone de:

- Conjunto de creencias actuales que el agente tiene acerca de su entorno.
- Función de revisión de creencias (frc) que actualiza las creencias del agente de acuerdo a sus percepciones.
- Función de generación de deseos a partir de las creencias e intenciones actuales.
- Conjunto de opciones actuales.
- Función de filtro que permite determinar las nuevas intenciones a partir de las creencias, intenciones y deseos actuales.
- Conjunto de intenciones actuales.
- Función de selección de acciones que permite determinar la acción a realizar a partir de las intenciones que se tienen.

Para especificar el sistema de agentes, se emplea un conjunto de modelos que operan a dos niveles de abstracción: Externo e interno.

Desde el punto de vista externo un sistema se modela como una jerarquía de herencias de clases de agentes de los que los agentes individuales son instancias.

Desde el punto de vista interno, se emplea un conjunto de modelos que permiten imponer una estructura sobre el estado

de información y motivación de los agentes y las estructuras de control que determinan su comportamiento.

El refinamiento de los modelos internos conlleva a la realimentación de los modelos externos.

Se debe validar en todo momento en los modelos: que los eventos se respondan en su momento, que las creencias se mantengan consistentemente y que la selección y ejecución de planes se desarrolle de manera que refleje cierta racionalidad.

#### C. GAIA

La metodología trata con el nivel macro (sociedad) y micro (agente) del sistema. La metodología ve el sistema como una organización computacional que consiste de varios roles interactuando [22]. Su objetivo es obtener un sistema que maximice alguna medida de calidad global (la cual no se llega a detallar).

En la fase de análisis se busca comprender el sistema y su estructura sin referenciar ningún aspecto de implementación. Esto se consigue a través de la idea de organización. Para definir una organización es necesario definir los roles, sus relaciones e interacciones. Esta fase contiene dos modelos:

- **Modelo de roles:** Se identifican los roles claves en el sistema los cuales son documentados en una plantilla especial propuesta por la metodología.
- **Modelo de Interacciones:** Contiene las relaciones entre los roles centrándose más en el propósito de las interacciones que en el orden de estas.

En la fase de diseño se trabaja sobre los resultados de la etapa de análisis de manera que se llegue a niveles de abstracción sobre los que se puedan trabajar con cualquiera de las metodologías tradicionales para el desarrollo de software (OOSE, OMT). Consta de tres modelos: **Modelo de Agente:** Se identifican los agentes que harán parte del sistema y los agentes que serán instanciados de esos tipos; **Modelo de Servicio:** Permite identificar el servicio principal asociado a cada agente; **Modelo de Conocimiento:** Documenta el conocimiento para cada agente.

Las principales críticas a GAIA son: se queda en un nivel de abstracción demasiado alto; se resalta la importancia del uso de fórmulas lógicas sin explicar sus inconvenientes; omite las distintas dependencias entre los modelos propuestos; faltan herramientas de soporte.

#### D. STEAM

STEAM fue desarrollado originalmente en el contexto de construir equipos de agentes pilotos de helicópteros de ataque para simulaciones militares. [23] y reutilizado después para la RoboCup.

Hay dos aspectos clave de STEAM que son relevantes en la construcción de equipos de agentes:

Lo primero es que provee un marco de trabajo para desarrollo de equipos, basado en dos jerarquías separadas, cada una con restricciones clave.

- Jerarquía y roles de organización de equipos: En STEAM un equipo puede tener una organización plana o jerárquica (en la cual un equipo se compone recursivamente de subequipos). Una característica clave de este tipo de organización es que puede estar basada en roles particulares.

Existen dos tipos de roles: Roles persistentes: Son asignaciones a largo término de los roles a los individuos o subequipos en la organización. Esta asignación se espera que no cambie a corto plazo. Roles de tarea específica: Asignación de roles a corto plazo, basado en la tarea y situación actual.

Hay propiedades específicas asociadas con cada equipo, subequipo o individuo. La asignación de roles a individuos o subequipos se hace basándose en sus capacidades, si la capacidad de un individuo o subequipo se degrada durante la ejecución, los roles pueden ser reasignados.

- Jerarquía de actividades de equipo: STEAM depende de una representación explícita de actividades de equipo en la forma de operadores de equipo o planes de equipo reactivos. Los operadores de equipo expresan explícitamente el conjunto de actividades de un equipo, a diferencia de los "operadores individuales" comunes, los cuales expresan las actividades propias de un agente. La jerarquía de operadores de equipo está construida sobre operadores individuales.

Tanto operadores individuales como operadores de equipo consisten de: reglas de precondición, reglas de aplicación y reglas de terminación. Sin embargo, mientras que un operador individual se aplica al estado privado de un agente, un operador de equipo se aplica el estado del equipo.

El estado del equipo es el modelo abstracto que el agente tiene de las creencias mutuas que el equipo posee acerca del mundo.

El mapeo entre las dos jerarquías descritas está centrado en roles. El rol de un subequipo o individuo que hace parte de un equipo que está ejecutando un operador de equipo, puede restringir los suboperadores que puede ejecutar en servicio del operador de equipo. Por ejemplo, si un equipo ? esta ejecutando un operador de equipo OP, entonces la asignación del rol ?i a un miembro ? de ?, restringe ? a cierto subconjunto ?i de suboperadores de OP. Además, los roles pueden tener una relación de coordinación específica entre ellos. La clave en STEAM es la instanciación del concepto de rol como parte de la jerarquía de organización del equipo, con su uso como una restricción en la selección de los suboperadores de un operador de equipo y su uso en expresar las relaciones de coordinación.

La segunda, y quizás la más importante contribución de STEAM es que provee un modelo de equipos de trabajo explícito y de propósito general, el cual codifica axiomas y teoremas

independientes de dominio que perfilan explícitamente las responsabilidades y compromisos de los miembros del equipo en su trabajo cooperativo. Por lo tanto, apunta a una reducción substancial del esfuerzo de codificación y a una mejora en la flexibilidad del trabajo en equipo.

#### **E. CASSIOPEIA**

Cassiopeia esta orientado al tipo de problemas en los cuales los comportamientos colectivos son repartidos en operaciones que son asignadas a agentes. De acuerdo con la metodología, un sistema multi-agente puede ser diseñado en término de agentes compuestos con tres niveles de comportamiento: (1) elemental, (2) relacional, (3) organizacional [24].

El principio subyacente de Cassiopeia es que el problema de organizar individuos para lograr soluciones de una manera colectiva debe de ser enfrentado por el diseñador y/o los agentes. Los problemas organizacionales aparecen debido a las dependencias funcionales inherentes al trabajo cooperativo: el comportamiento de un agente afecta y es afectado por el comportamiento de otros agentes. El conjunto completo de dependencias determina el acople del problema organizacional. Hay dos tipos de acople: (1) Estático: cuando los comportamientos individuales no compiten, (2) Dinámico: Algunos comportamientos son compartidos por ciertos agentes y se generan situaciones de competencia.

Cassiopeia parte de las tareas colectivas para terminar en el diseño del MAS a través de tres pasos:

- Definición de los comportamientos elementales: Se realiza una lista con todos los comportamientos que se consideran necesarios para alcanzar la tarea colectiva. A partir de estos comportamientos el diseñador define los agentes para el sistema.
- Comportamientos relationales: Se analiza la estructura de la organización basándose en las dependencias entre los comportamientos elementales. Esas dependencias determinan el grafo de acople del sistema. Las dependencias entre diferentes tipos de agentes son llamadas influencias y el grafo final que contiene las relaciones entre agentes es llamado grafo de influencias. Es este grafo el que brinda una representación de la estructura de la organización.

A partir de este punto el diseñador puede determinar los comportamientos relationales que le permiten a los agentes identificar y manipular influencias.

- Comportamientos organizacionales: Aquí se definen los comportamientos que permitirán a los agentes administrar la formación, durabilidad y disolución de los grupos.

Los comportamientos relationales permiten la formación de grupos que pueden no ser tan óptimos y se deben de tener criterios para permitir la disolución de dichos grupos.

## VI. CONCLUSIONES

El campo de trabajo de la robótica cooperativa, como cualquier campo de la inteligencia artificial, cuenta aún con problemas tales como la falta de definición de un lenguaje común sobre el cual exponer los adelantos e investigaciones. Los problemas sin resolver en los sistemas robóticos individuales, como por ejemplo mecanismos más eficientes de visión, siguen siendo problemas abiertos en esta área. Quizá el mayor problema es la falta de una metodología concreta que de cobertura a todos los ciclos que permiten el desarrollo de este tipo de sistemas.

## VII. AGRADECIMIENTOS

El trabajo descrito en este artículo se enmarca dentro del proyecto de investigación "Modelo de Sensórica y Percepción en Agentes Robóticos para la Identificación de Materiales" auspiciado por la Vicerrectoría de Investigación de la Universidad Nacional de Colombia (Código DIME: 805010042). También se resalta la labor de los auxiliares de investigación: Juan José González España, Marcela Jiménez Sánchez, Juan Sebastián Álvarez Cavaría, Juan Ricardo Osorio Cadavid, Rubén Esneider Ruiz Pino, Ana Caterine Serna Marín.

## REFERENCIAS

- [1] Avouris N. and Gasser L. *Distributed A.I.: Theory and Praxis*. Kluwer Academic Publisher, 1992.
- [2] Werner E. and Demazeau Y. *Decentralized A.I.* 3 Ed.; North.
- [3] Iglesias C., Garijo M. and Gonzales J. *A Survey of Agent-Oriented Methodologies*. 1999; disponible en: <http://citeseer.ist.psu.edu/iglesias99survey.html> [Fecha de acceso: enero de 2008].
- [4] Kurabayashi, D. *Toward Realization of Collective Intelligence and Emergent Robotics*. Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on.
- [5] Fukuda T., Nadagawaka S., Kawauchi Y. and Buss M. *Structure Decision for Self Organizing Robots based on Cell Structures*. IEEE International Conference on Robotics and Automation, Arizona, pp 695-700.
- [6] Caloud P., Choi W., Latombe J., LePape C. and Yim M. *Indoor Automation with Many Mobile Robots*. IROS-90, Japan, pp 67-72.
- [7] Norelis F. *Toward a Robot Architecture Integrating Cooperation Between Mobile Robots: Application to Indoor Environment*. The International Journal of Robotics Research 12, pp 79-98.
- [8] Kube C., Zhang H. and Wang X. *Controlling Collective Task with ALN*. IEEE International Conference on Intelligent Robots and Systems; Japan; pp 289-293.
- [9] Kube C. and Zhang H. *Collective Robotic Intelligence: From Animal to Animats*. International Conference on Simulation of Adaptive Behavior.
- [10] Barman, R., Kingdon, J., Mackworth, A., Pai, D., Sahota, M., Wilkinson, H. and Zhang Y. *Dynamite: A Testbed for Multiple Robots*. Proceedings, IJCAI-93 Workshop on Dynamically Interactive Robots; France.
- [11] Parker, L. *Learning in Cooperative Robot Teams*. Proceedings, IJCAI93 Workshop on Dynamically Interacting Robots; 1993 Chambery, France, pp. 12-23.
- [12] Parker, L. *An Experiment in Mobile Robotic Cooperation*. 1993; disponible en: <http://citeseer.ist.psu.edu/parker93experiment.html> [Fecha de acceso: noviembre de 2007].
- [13] Donald, B., Jennings, J. and Rus, D. *Experimental Information Invariants for Cooperating Autonomous Mobile Robots*. 1993; Proceedings, International Symposium on Robotics Research, Hidden Valley, PA.
- [14] Altenburg, K. *Adaptive Resource Allocation for a Multiple Mobile Robot System using Communication*. 1994, Technical Report NDSU-CSOR-TR-9404, North Dakota State University.
- [15] Beckers, O., Holland, E. and Deneubourg, J. *From Local Actions to Global Tasks: Stigmergy and Collective Robotics*. R. Brooks and P. Maes, editors, *Proceedings of the Fourth Workshop on Artificial Life*, Boston, MA, 1994. MIT Press.
- [16] RoboCup. *A Brief History of RoboCup*. Disponible en: [www.robocup.org/overview/23.html](http://www.robocup.org/overview/23.html) [Fecha de acceso: junio de 2008].
- [17] Mataric, M. *Issues and Approaches in Design of Collective Autonomous Agents*. 1995; disponible en: <http://citeseer.ist.psu.edu/11341.html> [Fecha de acceso: noviembre de 2007].
- [18] Rueda, S., Ucrós, M. and Quintero, A. *Agentes y Sistemas Multi-Agente: Integración de Conceptos Básicos*. Memos de Investigación, no. 319, pp.1-14. Universidad de los Andes, Colombia.
- [19] Lima P. and Custodio L. *Artificial Intelligence and System Theory: Applied to Cooperative Robots*. 2002; disponible en: <http://citeseer.ist.psu.edu/578654.html> [Fecha de acceso: noviembre de 2007].
- [20] Iglesias, C., Garijo, M., González, J., Gonzales and Velásquez, J. *Analysis and Design of Multiagent Systems using MAS-CommonKADS*. Disponible en: <http://citeseer.ist.psu.edu/iglesias98analysis.html> [Fecha de acceso: noviembre de 2007].
- [21] Marcus J. and Huber, J. *JAM: A BDI-theoretic Mobile Agent Architecture*. Intelligent Reasoning Systems; 1999; International Conference on Autonomous Agents; Proceedings of the third annual conference on Autonomous Agents.
- [22] Wooldridge, M. Jennings, N. and Kinny, D. *A Methodology for Agent-Oriented Analysis and Design*. Disponible en: <http://www.cs.civ.ac.uk/~mjw/pubs/agents99.pdf> [Fecha de acceso: noviembre de 2007].
- [23] Tambe M. *Towards Flexible Teamwork*. Journal Artificial Intelligence Research; 1997.
- [24] Collinot A., Dragoul A. and Benhamou P. *Agent Oriented Design of a Soccer Robot Team*. ICMAS'96; 1996.



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA  
SEDE MEDELLÍN  
FACULTAD DE MINAS

**120** años   
TRABAJO Y RECTITUD