

---

# ABC RAN: UNA ARQUITECTURA DE *BLACKBOARD* PARA EL CONTROL DE UN *ROBOT* AUTONOMO NAVEGANTE

ANA MARIA TREJOS CUARTAS  
JUAN DAVID VELASQUEZ FRANCO

## RESUMEN

Este artículo describe el uso de la arquitectura de blackboard para el control de un robot móvil autónomo, empleando un esquema conocido como navegación reactiva. La arquitectura propuesta ha sido utilizada en experimentos para controlar un robot simulado. Dichos experimentos o simulaciones han demostrado la flexibilidad y solidez del enfoque empleado.

**Palabras Claves:** Robots móviles;  
Navegación reactiva;  
Inteligencia artificial;  
Sistemas Blackboard.

## INTRODUCCION

Los robots móviles o robots autónomos navegantes tienen una larga historia dentro del área de la inteligencia artificial. Por mucho tiempo el anhelo de los investigadores en esta área ha sido construir máquinas que posean la habilidad de explorar y entender el ambiente que los rodea.

Como parte de los sistemas robóticos, los robots móviles autónomos constituyen un paso importante en la evolución de la inteligencia de máquinas. Este paso se presenta al proporcionar a un robot orientado a una tarea específica, la habilidad de navegar en el

espacio o ambiente, utilizando algún medio específico como las ruedas, patas o bandas y permitirle que tome decisiones de acuerdo con los cambios que ocurren en el ambiente.

Los robots móviles autónomos tienen múltiples aplicaciones potenciales. Dichas aplicaciones incluyen operaciones submarinas, exploraciones espaciales, operaciones mineras, manejo de materiales peligrosos o tóxicos, automatización de oficinas, manufactura y seguridad, entre otras. La característica principal de estas aplicaciones radica en que el ambiente es generalmente no predefinido o estructurado y es también impredecible. Por esto, es necesario construir robots que presenten un nivel real de autonomía, es decir, que dado un comando u objetivo, el robot realice todas las tareas y planeación necesarias para satisfacer ese objetivo. En otras palabras, se necesita construir un robot bastante flexible que incluya actuadores, sensores, procesadores y equipos auxiliares para que, como un humano, pueda realizar una variedad de tareas bajo las condiciones antes mencionadas.

---

ANA MARIA TREJOS CUARTAS. Analista de Auditoría, Universidad EAFIT, Ingeniera de Sistemas.

JUAN DAVID VELASQUEZ FRANCO. Investigador, Center for Educational Computing Initiatives - CECI - MIT. Ingeniero de Sistemas.

Para que un robot móvil logre alcanzar un nivel apreciable de autonomía, debe poseer un sistema de control lo suficientemente flexible que le permita desarrollar un comportamiento de navegación inteligente. En esta forma, el robot tiene la capacidad de tomar decisiones y reaccionar de una manera apropiada a los diferentes eventos que ocurren en el ambiente, mientras desarrolla una tarea específica.

## 1. ANTECEDENTES

En general, la navegación autónoma de un robot móvil puede definirse como el proceso de planeación y control del movimiento del robot en un ambiente determinado. Este proceso comprende varios problemas como percepción inteligente, planeación de trayectorias, construcción de mapas, mecanismos de locomoción, etc. Sin embargo, estos problemas o tareas no deben considerarse como una colección de problemas aislados sino que, por el contrario, debe dárseles un enfoque sistémico que permita enfrentar con éxito los problemas fundamentales de la navegación inteligente.

Por lo anterior, es necesario contar con un sistema que integre y controle de una manera eficiente y organizada todas las soluciones a los problemas antes mencionados. Este sistema es conocido como la arquitectura o sistema de control del robot.

El sistema de control, debe ser capaz de lograr un comportamiento de navegación inteligente si se desea lograr la autonomía total en el robot móvil. Este comportamiento proporciona al robot la habilidad de responder de una manera adecuada a los eventos que ocurren en el ambiente, mientras intenta alcanzar una meta específica.

Hasta la fecha se han desarrollado diferentes arquitecturas de control para robots móviles basadas en técnicas de computación, técnicas matemáticas, técnicas de inteligencia artificial, etc.

La arquitectura ABC RAN propone un sistema de control que emplea un esquema de navegación reactiva e integra todos los aspectos de esta navegación por medio del modelo de blackboard, ampliamente conocido en los sistemas basados en el conocimiento.

**1.1 Navegación reactiva.** La navegación reactiva es un esquema de control utilizado con bastante

éxito en algunos robots móviles. Se caracteriza por una relación de estímulo-respuesta entre el mundo y el robot, en la cual este último reacciona en cierta forma ante la información sensorial que percibe del ambiente [ARKIN, 1990] <sup>(1)</sup>.

En este tipo de control se logran comportamientos de navegación complejos a partir de la combinación de otros comportamientos sencillos y de bajo nivel que pueden describirse como reflejos o respuestas a los eventos que ocurren en el ambiente.

Existen varios ejemplos representativos de este tipo de navegación. Uno de ellos se conoce como la arquitectura de superposición (subsumption architecture) [BROOKS, 1986] <sup>(2)</sup>. Esta arquitectura se compone de varios niveles en capas distribuidos jerárquicamente, cada uno de los cuales representa un comportamiento permisible para el robot.

Para que un robot móvil logre alcanzar un nivel apreciable de autonomía, debe poseer un sistema de control lo suficientemente flexible que le permita desarrollar un comportamiento de navegación inteligente.

Otro enfoque similar propuesto por [NELSON, 1990] <sup>(3)</sup>, parte de las características de control y desarrollo observadas en los movimientos humanos. Este enfoque pretende lograr comportamientos inteligentes a partir de la interacción de partes no inteligentes que se relacionan de una manera especial.

- (1) Arkin, Ronald. Integrating Behavioral, Perceptual, and World Knowledge in Reactive Navigation. *En: Robotics and Autonomous Systems*. Vol. 6 (1990); p. 105-122.
- (2) Brooks, Rodney. A. Robust Layered Control System For A Mobile Robot. *En: IEEE Journal of Robotics and Automation*. Vol. RA-2, No. 1 (Mar. 1986); p. 14-23.
- (3) Nelson, Mark. An Adaptive Control System for Mobile Robot Navigation. p. 186-197. *En: National Conference on Robotics (3º: 1990: Melbourne)*. Proceedings Third National Conference on Robotics. Melbourne: s.n., 1990.

Igualmente, otro enfoque [ARKIN, 1990]<sup>(4)</sup> propone una arquitectura basada en esquemas relacionada con los modelos neurocientíficos del comportamiento de desvío (evasión de obstáculos) en los anfibios.

**1.2 El modelo de blackboard.** El modelo de **blackboard** es un modelo de solución de problemas que evolucionó a partir del proyecto del entendimiento del habla denominado HEARSAY-II y que posteriormente ha sido empleado en diferentes dominios como el análisis de señales en HASP/SIAP [NII *et al.*, 1982]<sup>(5)</sup>, el análisis de las estructuras de las proteínas en CRYSTALIS [TERRY, 1983]<sup>(6)</sup> y muchos otros.

La navegación reactiva se caracteriza por una respuesta a un estímulo externo. Es decir, un **robot** exhibiría determinado comportamiento como una combinación de respuestas a una gran variedad de estímulos del mundo.

La idea básica del modelo de **blackboard** consiste en dividir un problema complejo en una serie de sub tareas independientes, integrando así el conocimiento de varias fuentes para lograr una meta común. Un modelo de **blackboard** está compuesto por tres elementos básicos (Ver Figura 1): las fuentes de conocimiento o **Knowledge Sources**, la estructura de datos del **blackboard** y el mecanismo o unidad de control <sup>(7)</sup>

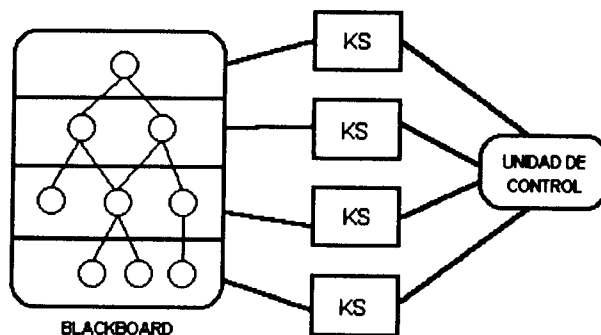


Figura 1. Arquitectura básica de **blackboard**.

## 2. LA ARQUITECTURA ABC RAN

La arquitectura propuesta se ha concebido con base en las tareas y componentes básicos de un **robot** móvil, incluyendo desde los sensores y actuadores hasta la planeación. No se ha desarrollado para una aplicación específica de la navegación de **robots** (ej.: exploraciones espaciales u operaciones submarinas) lo que le permite ser una arquitectura general para el área y considerarse como un marco de trabajo para una aplicación particular.

El enfoque propuesto es el diseño de un sistema de control bastante flexible que permita su evolución en etapas, incrementando en cada etapa, la complejidad del comportamiento de navegación. Es decir, cada nueva etapa se construye y adiciona al sistema ya existente para lograr una nueva clase de comportamiento de navegación. Todo este proceso converge hacia la obtención de un sistema de control bastante flexible y capaz de interactuar inteligentemente con su ambiente.

Los comportamientos de navegación que se han incluido en el diseño de la arquitectura propuesta son algunos de los que describe Brooks para la arquitectura de superposición [BROOKS, 1986]<sup>(8)</sup> y se enumeran de nuevo a continuación:

Nivel	Comportamiento	Descripción
0	Avanzar	Desplazarse hacia adelante.
1	Evitar Colisiones	Detenerse si existen obstáculos.
2	Divagar/Explorar	Rodear los obstáculos y continuar.

(8) Brooks, Rodney. *op.cit.*

(4) Arkin, Ronald C. *Autonomous Mobile Robots*. p. 116-160. En: ADELI, Hojjat. *Knowledge Engineering: Applications*. New York: Mc Graw Hill, 1990. 352 p.

(5) Nii *et al.* *Signal-to-Symbol Transformation: HASP/SIAP Case Study*. p. 135-157. En: Englemore, Robert and Morgan, Tony. *Blackboard Systems*. Wokingham: Addison-Wesley, 1988. 602 p.

(6) Terry, Allan. *Using Explicit Strategic Knowledge to Control Expert Systems*. p. 159-188. En: Englemore, Robert and Morgan, Tony. *Blackboard Systems*. Wokingham: Addison-Wesley, 1988. 602 p.

(7) Para profundizar al respecto del modelo de **blackboard** y sus aplicaciones puede consultarse: Jagannathan, V; Dodhiawala, Rajendra and Baum, Lawrence S. *Blackboard Architectures and Applications*. San Diego: Academic Press, 1989. 525 p.

Cada uno de estos comportamientos se logra a partir de la interacción de varios agentes que representan el conocimiento que posee el robot para reaccionar a los estímulos de su ambiente. Dada su naturaleza, cada nuevo comportamiento determina qué otros agentes requiere el sistema para lograr su nueva complejidad.

El modelo de *blackboard* consiste en dividir un problema complejo en una serie de subtarefas independientes, integrando así el conocimiento de varias fuentes para lograr una meta común.

En síntesis, la arquitectura resuelve de una manera flexible y evolutiva el problema de control para un robot autónomo navegante cuyo comportamiento de navegación está determinado por la cooperación de varios agentes y se vuelve cada vez más complejo e inteligente a medida que evoluciona y crece el sistema.

**2.1 Diseño de la arquitectura de control.** La arquitectura ABC RAN consta de tres componentes fundamentales: Blackboard del dominio, unidad de control y Knowledge Sources.

**2.1.1 El blackboard del dominio.** El blackboard es una base global de datos que contiene el estado actual de la solución del problema en un momento determinado. Esta base de datos está organizada jerárquicamente y separada en varios niveles de abstracción o niveles de análisis <sup>(9)</sup>, dependiendo del comportamiento de navegación que se desee lograr.

**2.1.2 Los knowledge sources.** Los KSs son los agentes que interactúan y poseen todo el conocimiento necesario para lograr conjuntamente el comportamiento de navegación del nivel deseado. Cada uno de ellos es competente en un dominio específico (ej: planeación de trayectorias, construcción de mapas, control de motores, etc). Los KSs son independientes, es decir, toda interacción o comunicación entre ellos ocurre sólo a través del blackboard. Cada KS responde oportunamente a los

(9) Estos niveles de análisis corresponden a diferentes aspectos de la solución del problema y no deben confundirse con los niveles de complejidad de los comportamientos de navegación deseados.

cambios que ocurren en el blackboard y que representan ocurrencias de eventos en el ambiente empleando un mecanismo de dependencia.

En el diseño de la arquitectura se consideraron los KSs o fuentes de conocimiento con la estructura que se muestra a continuación.

#### KNOWLEDGE SOURCE

<b>Nombre</b>	: <u>Label</u> identificador
<b>Descripción</b>	: Comportamiento característico del KS
<b>Condición</b>	
* <b>Trigger</b>	: Predicados basados en eventos que determinan la relevancia de activar un KS.
* <b>Precondiciones</b>	: Predicados basados en estados que determinan la factibilidad de ejecutar un KS.
* <b>Nivel origen</b>	: Nivel del <u>blackboard</u> que contiene los datos de entrada para el KS.
<b>Nivel de acción</b>	: Nivel del <u>blackboard</u> que cambia o modifica el KS
<b>Prioridad</b>	: Prioridad del KS con relación a otros KSs.
<b>Acción</b>	: Programa de los cambios en el <u>blackboard</u>

#### 2.1.3 Dependencia entre los KSs y el blackboard.

Como se había mencionado anteriormente, los KSs responden oportunamente a los cambios ocurridos en el blackboard gracias a un mecanismo de dependencia que existe entre ellos y los niveles de análisis. Este mecanismo consiste en lo siguiente: Un KS se activa dada la ocurrencia de un evento determinado en alguna parte del blackboard como, se mencionó antes. Esa parte del blackboard se referencia en el campo "Nivel de origen" del KS que indica en qué nivel del blackboard se produce el o los eventos que lo activan. Igualmente, un nivel de análisis conoce todos los KS que se activan si ocurren determinados eventos en ese nivel. Como puede verse, entre los KSs y los niveles de análisis existe una relación de dependencia ya que un KS depende de un evento ocurrido en un nivel determinado. El mecanismo de dependencia funciona entonces de la siguiente manera:

1. Se produce un evento  $E_i$  en el nivel de análisis  $N_k$ .
2. El nivel  $N_k$  informa a los KS que dependen del evento  $E_i$ .
3. Los KS que dependían del evento  $E_i$  informan a la unidad de control que pueden entrar a colaborar en el proceso de solución.

Este mecanismo se ha denominado de dependencia ya que fue diseñado con base en el mecanismo de dependencia que existe en el lenguaje Smalltalk donde un objeto informa a sus dependientes que ha cambiado [LALONDE and PUGH, 1990] <sup>(10)</sup>. La utilización de este mecanismo mejora el funcionamiento de la arquitectura de blackboard original ya que el control no tiene que hacer una búsqueda exhaustiva de todos los KSs que se pueden activar dada la ocurrencia de un evento, sino que son los KSs los que realmente se dan cuenta, a través de la dependencia con los niveles de análisis, que es el momento de contribuir a la solución.

**La arquitectura ABC RAN propone un sistema de control que emplea un esquema de navegación reactiva e integra todos los aspectos de esta por medio del modelo Blackboard.**

**2.1.4 La unidad de control.** La unidad de control administra los KS y el blackboard. Dicha unidad tiene que determinar cuál KS debe ejecutar y cuándo hacerlo según el estado del blackboard. Para la arquitectura ABC RAN, la unidad de control sigue el proceso de control descrito en la arquitectura de control BB1 [HAYES-ROTH, 1985] <sup>(11)</sup>.

La solución al problema de navegación en la arquitectura propuesta comienza con la entrada de información en alguno de los niveles del blackboard. La adición de esta información genera un evento en ese nivel lo cual ocasiona que algunos KSs avisen a la unidad de control pidiendo ser activados para entrar a contribuir en el proceso de solución. Por cada KS que informa de su capacidad para colaborar, la unidad de control crea un registro de la activación del mismo denominado blackboard Source Activation Record (KSAR) y lo adiciona a la lista de KSs activados. Cada KSAR es una estructura de cuatro campos como se muestra a continuación:

(10) Lalonde, Wilf R. and Pugh, John R. Inside Smalltalk. New Jersey: Prentice-Hall, 1990. v.1. 512 p. ISBN 0-13-468414-1.

(11) HAYES-ROTH, Barbara. A Blackboard Architecture for Control. p. 505-540. En: BOND, Alan H. and GASSER, Les. Readings in Distributed Artificial Intelligence. San Mateo: Morgan Kaufmann, 1988. 649 p.

## KNOWLEDGE SOURCE ACTIVATION RECORD

- **Nombre** : Número identificador
- **KS** : Nombre del KS activado
- **Ciclo de Activación** : Ciclo en que fue activado el KS
- **Evento de Activación**: Evento que activó el KS

Para cada uno de los KSAR existentes en la lista de activados, se evalúan sus preconditions con el estado actual del blackboard para determinar si es factible ejecutar el KS al cual representa. Si sus precondiciones se cumplen, la unidad de control retira ese KSAR de la lista de activados y lo agrega a la lista de invocables. En este punto, el control elige el KS por ejecutar, basándose para ello en la lista de invocable y las prioridades de cada uno de ellos. Por último, la unidad de control interpreta la acción de ese KS y lo ejecuta, ocasionando así que se generen nuevos eventos en ciertos niveles y de nuevo comience el ciclo de control.

**2.2 Evolución de la arquitectura.** El enfoque propuesto pretende conformar una arquitectura de control evolutiva que incrementalmente permita lograr diferentes comportamientos de navegación.

**2.2.1 Comportamiento de Nivel 0 (Avanzar).** Este comportamiento es el más sencillo de todos y consiste solamente en que el robot se desplace hacia adelante. Dado que aún no se ha incorporado ninguna capacidad sensorial al robot, se espera que este comportamiento termine cuando el robot no pueda avanzar más por encontrarse atascado contra un obstáculo.

Para lograr este comportamiento se ha diseñado la arquitectura de la siguiente forma:

El blackboard está dividido en dos niveles de análisis denominados:

Nivel 1: Estados del robot.

Nivel 2: Comandos de desplazamiento.

El primer nivel, denominado "Estados del Robot", contiene todos los datos relacionados con el estado del robot en un momento específico, es decir, su dirección (Norte, Sur, Este, Oeste, etc.) y el estado de sus motores (Atascados o en funcionamiento).

El otro nivel llamado "Comandos de desplazamiento" contiene los comandos que el sistema de locomoción debe ejecutar. Los comandos posibles son los siguientes: parada, avance, retroceso y giro. Los tres últimos deben estar acompañados de una magnitud. Por ejemplo "Avance 30" o "Giro 45°".

---

Los siguientes son los Knowledge Sources que se han definido para lograr este comportamiento de navegación.

- **KSNavegar:** Reacciona ante la existencia de nuevos estados del robot, generando comandos de avance.
- **KSMotoresAtascados:** Reacciona en el caso de que los motores del robot se atasquen con un obstáculo.
- **KSLocomoción:** Reacciona ante la existencia de nuevos comandos de desplazamiento, los cuales interpreta y convierte en señales motrices para su ejecución.

**2.2.2 Comportamiento de Nivel 1 (Evitar Colisiones).** Asume el comportamiento de navegación anterior, en que el robot se desplace hacia adelante. La única diferencia consiste en que el robot debe detenerse en caso de percibir algún obstáculo en frente suyo.

Dado que el robot posee el conocimiento suficiente para avanzar hacia adelante, sólo es necesario adicionar conocimiento que le permita percibir los posibles obstáculos y prevenir la colisión. Siendo así, es necesario incorporar dos nuevos agente a la arquitectura.

- **KSControlSensores:** Controla los sensores de detección de obstáculos que posee el robot e interpreta sus señales.
- **KSEvitarColisión:** Ordena al robot que se detenga en caso de encontrar un obstáculo en su camino.

Adicionalmente, es necesario almacenar en alguna parte las interpretaciones de los sensores que el agente de control de los mismos produce. Por tal motivo, se ha adicionado un nivel al blackboard denominado "Interpretaciones de los Sensores".

**2.2.3 Comportamiento de Nivel 2 (Divagar/ Explorar).** Es un comportamiento de navegación de más alto nivel y consiste en que el robot se desplace hacia adelante en cualquier dirección, pero que evite los obstáculos que se le presenten durante su trayectoria alterando su rumbo y continuando con su desplazamiento.

Para lograr este comportamiento no es necesario adicionar otro agente, es suficiente con adicionar conocimiento al KSEvitarColisiones para que en vez de ordenar que el robot se detenga, genere un comando de giro en el sentido que exija la menor rotación para robot.

**2.2.4 Otros comportamientos.** El proceso de evolución del sistema de control para alcanzar comportamientos de navegación cada vez más inteligentes es el mismo que se ha utilizado para lograr los comportamientos anteriores. Basta con adicionar nuevo conocimiento que le proporcione al robot las capacidades necesarias para lograrlo y probablemente realizar algunas modificaciones sobre la estructura de datos del blackboard.

Por ejemplo, si se quieren lograr comportamientos de navegación más avanzados como "Navegar e ir construyendo un mapa del ambiente" o "Identificar un objeto específico", es necesario robustecer el sistema de control adicionando nuevo conocimiento relativo a la construcción y representación de mapas en el primer caso, y/o características específicas del objeto a identificar en el segundo.

Puede apreciarse entonces la facilidad con que se pueden lograr otros comportamientos de navegación, demostrando así, la modularidad, solidez y flexibilidad de la arquitectura de control propuesta.

### 3. EVALUACION DE LA ARQUITECTURA

La evaluación de un sistema de control inteligente ha sido un aspecto de amplia discusión entre los investigadores de la inteligencia artificial. Realmente es difícil afirmar con certeza que el comportamiento exhibido por un sistema de control específico es inteligente o no lo es. Sin embargo, Barbara Hayes-Roth propone algunos criterios para evaluar un sistema de control inteligente <sup>(12)</sup>.

La siguiente tabla resume la confrontación de los criterios de evaluación con la arquitectura ABC RAN:

---

(12) La explicación detallada de estos criterios puede encontrarse en: HAYES-ROTH, Barbara. A Blackboard Architecture for Control. p. 505-540. En: BOND, Alan H. and GASSER, Les. Readings in Distributed Artificial Intelligence. San Mateo: Morgan Kaufmann, 1988. 649 p.

**CRITERIO DE EVALUACION****ABC RAN**

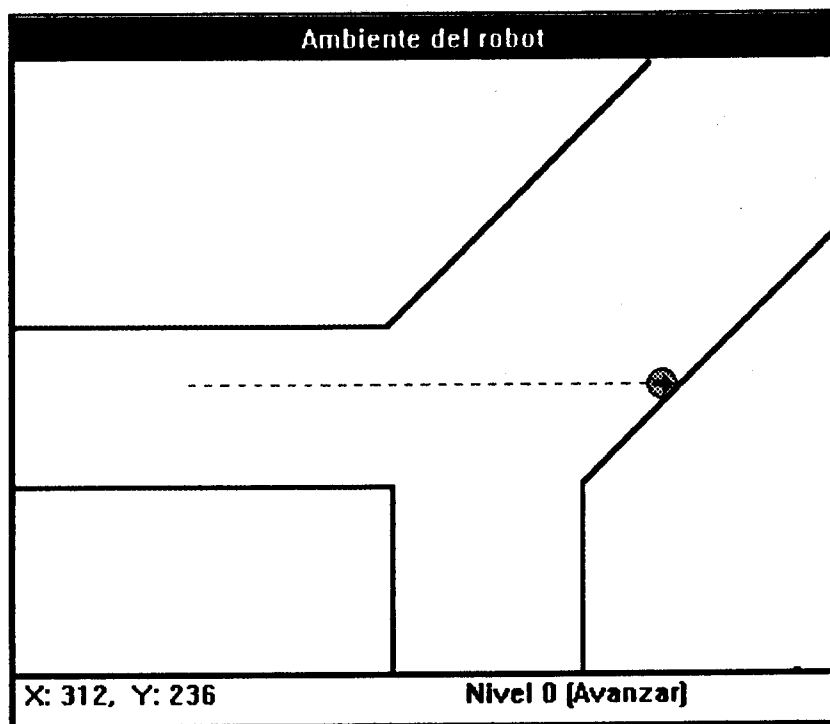
1. Tomar decisiones de control explícitas	SI
2. Decidir qué acciones realizar (deseables/factibles)	SI
3. Adoptar heurísticas de tamaño variable	SI
4. Adoptar la heurística de control más apropiada	POSIBLE
5. Variar la heurística de control dinámicamente	POSIBLE
6. Planear una secuencia estratégica de acciones	POSIBLE
7. Razonar entre las acciones del dominio y del control	SI
8. Funcionalidad	SI
9. Facilidad de implementación	SI
10. Facilidad de modificación	SI
11. Claridad de representación	SI
12. Explicación del razonamiento	POSIBLE
13. Aprendizaje	NO

Como puede apreciarse, de los trece criterios evaluados, la arquitectura ABC RAN satisface completamente ocho de ellos. De los cinco restantes, cuatro pueden lograrse con unos pequeños cambios en el diseño y sólo uno (criterio 13) requeriría de un re-diseño de la arquitectura para satisfacerlo.

Adicionalmente, y con el fin de obtener una evaluación operativa de la arquitectura propuesta, se ha diseñado e implementado un modelo computacional que refleja totalmente el desempeño de la arquitectura para lograr ciertos comportamientos de navegación en un *robot* simulado.

Este modelo se ha diseñado en su totalidad bajo un enfoque orientado a objetos empleando el lenguaje de programación C++ bajo el ambiente MS-Windows. Todos los aspectos relacionados con el ambiente y las capacidades sensoriales y motrices del *robot* fueron simulados.

Los resultados obtenidos en las simulaciones (Ver Figuras 2, 3 y 4) han demostrado que la arquitectura es lo suficientemente flexible para continuar con su utilización en las investigaciones de nuevos comportamientos de navegación inteligente.



**Figura 2. Ejecución del sistema para el Nivel 0.**

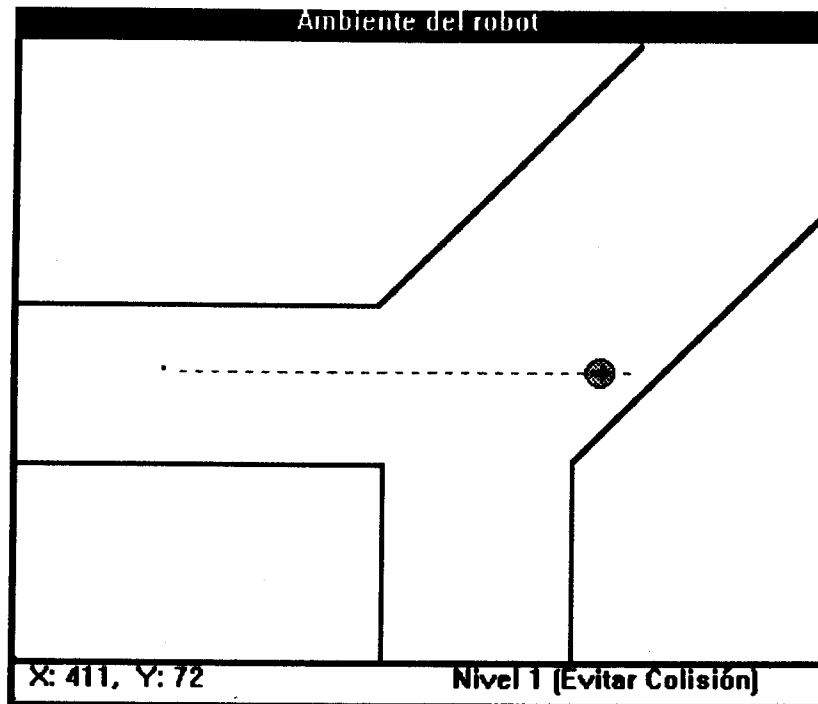


Figura 3. Ejecución del sistema para el Nivel 1.

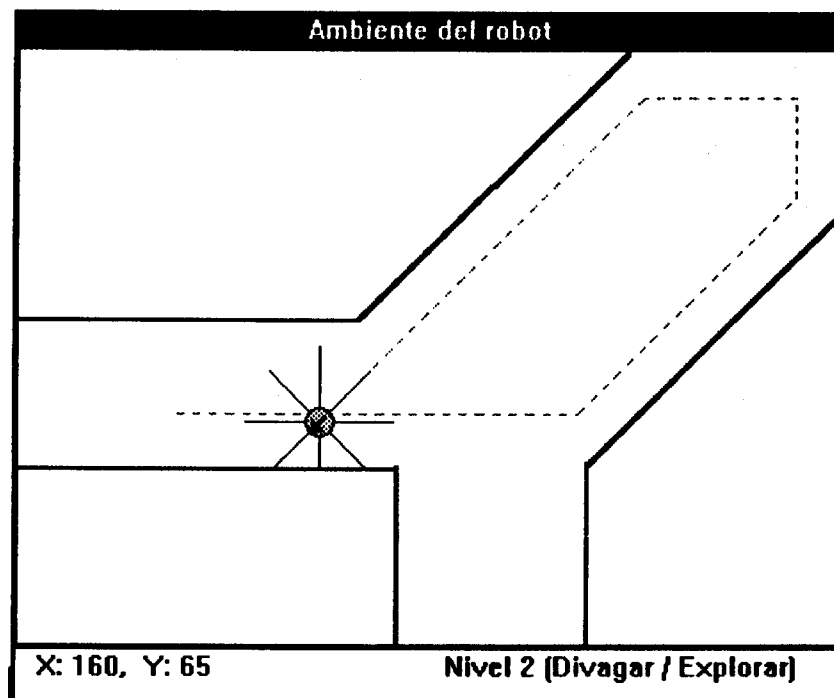


Figura 4. Ejecución del sistema para el Nivel 2.



---

En síntesis, dados los comportamientos de navegación logrados con el robot simulado y la confrontación con los criterios de evaluación propuestos por Hayes-Roth, puede concluirse que el modelo de la arquitectura propuesta es válido y por lo tanto puede emplearse con éxito en el control de un robot autónomo navegante.

#### 4. CONCLUSIONES

- Se ha propuesto un sistema de control para un robot autónomo navegante. Este sistema además de integrar de manera efectiva las tareas involucradas en la navegación inteligente, permite, gracias a su modularidad, su desarrollo evolutivo incorporando nuevo conocimiento de una manera ágil e incremental.
- En la presente investigación, se ha utilizado el modelo de blackboard como un aporte del área de la inteligencia artificial, el cual ha demostrado ser un enfoque bastante interesante para enfrentar problemas que requieran de un control inteligente.
- Se ha logrado poner en práctica los aspectos teóricos de la arquitectura de blackboard, desarrollando un modelo computacional para la misma. En la presente investigación se empleó este modelo en el área de la navegación autónoma de robots. Sin embargo, la generalidad obtenida en el diseño del mismo, permite su aplicación en otras áreas del conocimiento.
- La arquitectura ha sido evaluada en gran parte a través de las simulaciones realizadas. Una simulación es bastante útil, confiable y práctica y además permite identificar errores en el diseño que de ser construidos resultarían en costos elevados. Sin embargo, la evaluación de una arquitectura nunca va a ser igual con una simulación en vez de un robot físico. El robot es la realidad del problema y es en última instancia quien realmente prueba la arquitectura o sistema de control que gobierna sus acciones.

#### RECONOCIMIENTOS

Los autores expresan sus reconocimientos y agradecimientos al Ing. Luis Guillermo Restrepo Rivas, asesor del proyecto, por sus valiosas contribuciones durante la investigación, al Dr. Félix Londoño, por sus contribuciones con relación al modelo de blackboard, y a todas aquellas personas que en una u otra forma contribuyeron a la realización de la investigación.

#### BIBLIOGRAFIA

- Arkin, Ronald. Integrating Behavioral, Perceptual, and World Knowledge in Reactive Navigation. *En: Robotics and Autonomous Systems*. Vol. 6 (1990); p. 105-122.
- \_\_\_\_\_. Autonomous Mobile Robots. p. 116-160. *En: ADELI, Hojjat. Knowledge Engineering: Applications*. New York: Mc Graw Hill, 1990. 352 p.
- Brooks, Rodney. A Robust Layered Control System For A Mobile Robot. *En: IEEE Journal of Robotics and Automation*. Vol. RA-2, No.1 (Mar. 1986); p. 14-23.
- Hayes-Roth, Barbara. A Blackboard Architecture for Control. p. 505-540. *En: BOND, Alan H. and GASSER, Les. Readings in Distributed Artificial Intelligence*. San Mateo: Morgan Kaufmann, 1988. 649 p.
- Jagannathan, V; Dodhiawala, Rajendra and Baum, Lawrence S. Blackboard Architectures and Applications. San Diego: Academic Press, 1989. 525 p.
- Lalonde, Wilf R. and Pugh, John R. *Inside Smalltalk*. New Jersey: Prentice-Hall, 1990. v.1. 512 p. ISBN 0-13-468414-1
- Lang, Sherman Y. T. and WONG, Andrew K. C. Building Geometric world models with graph synthesis for sensor fusion in mobile robots. *En: Comput. Intell.* Vol. 6 (1990); p. 91-107.
- Nelson, Mark. An Adaptive Control System for Mobile Robot Navigation. p. 186-197. *En: National Conference on Robotics (3º : 1990: Melbourne)*. Proceedings Third National Conference on Robotics. Melbourne: s.n., 1990.
- \_\_\_\_\_. An Intelligent Navigational Control System for a Mobile Robot. p. 186-197. *En: The International Symposium and Exposition on Robots (19º : 1988: Sydney)*. Sydney: IFS. p. 756-766.
- NII, H. Penny. Introduction. *En: Jagannathan, V; Dodhiawala, Rajendra and BAUM, Lawrence S. Blackboard Architectures and Applications*. San Diego: Academic Press, 1989. 525 p.
- \_\_\_\_\_. The blackboard model of problem solving and the evolution of blackboard architectures. *En: the al mazazine*. (jul. 1986); p. 38-53
- NII et al. Signal-to-Symbol Transformation: HASP/SIAP Case Study. p. 135-157. *En: Engelmöre, Robert and MORGAN, Tony. Blackboard Systems*. Wokingham: Addison-Wesley, 1988. 602 p.