

# De simbólicos vs. subsimbólicos, a los robots etoinspirados

José María Cañas y Vicente Matellán

Grupo de Robótica  
Universidad Rey Juan Carlos, 28933 Móstoles (España),  
{jmplaza, vmo}@gsync.escet.urjc.es,  
<http://gsync.escet.urjc.es/robotica>

**Resumen** En la Inteligencia Artificial, desde sus orígenes, han existido dos corrientes básicas, la simbólica y la subsimbólica. Estas dos aproximaciones han tenido gran influencia también en la robótica. En este artículo queremos presentar un enfoque menos conocido, el de la etología, y en concreto su aplicación a la generación de comportamiento autónomo en robots móviles. Para ello presentamos los fundamentos de la "Jerarquía Dinámica de Esquemas", una arquitectura para el control de robots móviles, basada en la composición de unidades simples denominadas "esquemas" siguiendo las teorías etológicas de Arbib. Igualmente se presentan experimentos preliminares que validan esta aproximación y se discute su viabilidad y se presentan los trabajos previstos para continuar investigando en esta línea.

## 1. Inteligencia artificial y robótica

En 1950 el Dr. Grey Walter publicó "An imitation of life" [1] en el que describía su trabajo con las tortugas-robot<sup>1</sup> Elmer y Elsie que había construido durante 1949 y que pueden considerarse como los primeros robots móviles. Sus trabajos son por tanto prácticamente contemporáneos de los de Warren S. McCulloch y Walter Pitts, que publicaron en 1943 "*A logical calculus of the ideas immanent in nervous activity*" [2] y de los de Norbert Wiener, que publicó su trabajo "Cybernetics" [3] en 1948.

No sin discusión, podríamos considerar que todos estos trabajos supusieron el nacimiento de la robótica, la cibernética, y como consecuencia de la convocatoria de 1995<sup>2</sup> de J. McCarthy, M. L. Minsky, N. Rochester, y C.E. Shannon, de la Inteligencia Artificial (IA en adelante).

Simplificando mucho la compleja historia de la IA desde entonces, podemos decir que ha tenido y tiene dos familias básicas, la IA subsimbólica, interesada en modelar la inteligencia a un nivel similar al de la neurona, de forma que cosas como el conocimiento y la planificación "emergerán"; y la IA simbólica, que modela elementos como el conocimiento y la planificación en estructuras de datos que tienen sentido para los programadores que las construyen. Otra forma de explicar la diferencia entre ambas aproximaciones, es la fuente de su inspiración, la biología en el caso de la IA subsimbólica y la psicología en el caso de la IA simbólica.

La psicología cognitiva, que trata de entender el funcionamiento de la inteligencia humana, ha ejercido gran influencia sobre la IA, que a su vez persigue poder reproducirla en una máquina. En este sentido, quizá apoyado en un razonamiento introspectivo, el paradigma dominante en la IA ha

<sup>1</sup> La historia completa de estos ingenios puede consultarse en <http://www.ias.uwe.ac.uk/Robots/gwonline/gwonline.html>

<sup>2</sup> La convocatoria de lo que hoy conocemos como Conferencia de Dartmouth tuvo lugar en 1956 en la forma de convocatoria de un estudio durante dos meses por 10 personas del supuesto de que la inteligencia podría ser descrita de forma tan precisa, que una máquina pudiera simularla.

sido el de la descomposición funcional de la inteligencia en módulos especialistas (lenguaje, visión, razonamiento, etc.) y la lógica como motor de la deliberación, que se plasma como cierto proceso de búsqueda dentro de las alternativas.

También su ascendiente sobre la robótica móvil ha sido muy intenso. Por ejemplo, aportando la idea de que el comportamiento se puede generar como la ejecución de cierto plan calculado de antemano. Dicho plan es fruto de una deliberación sobre cierta representación del mundo, que tiene en cuenta los objetivos del robot. Por ejemplo, una ruta como secuencia de tramos intermedios que acaban llevando al robot al punto destino. La IA enfatiza la planificación y el modelado de la realidad como ingredientes fundamentales de la inteligencia en los robots.

El robot Shakey [4], un pionero dentro la robótica móvil, construido en el Stanford Research Institute (SRI) es el exponente más conocido de de la influencia de la *Inteligencia Artificial Simbólica*. Tenía una cámara, motores y sensores odométricos. Era capaz de localizar un bloque en su entorno y empujarlo lentamente. El procesamiento de las imágenes se realizaba en un ordenador fuera del robot. Dentro del mismo instituto, un digno sucesor fue el robot Flakey, construido en 1984 con numerosos avances tecnológicos sobre Shakey, y que ya incluía en su interior ordenadores para realizar a bordo cualquier cómputo necesario para su comportamiento.

Los orígenes epistemológicos de este enfoque están en la filosofía cartesiana que considera el alma como el ente que decide el comportamiento [5]. Hunde sus raíces en el cognitivismo y representa toda una teoría del funcionamiento de la inteligencia. Era natural que la Inteligencia Artificial clásica buscara refrendo probando su capacidad de generar comportamiento inteligente en cuerpos robóticos. Su aporte a la robótica ha sido fundamental en el desarrollo de esta última, introduciendo ideas como el manejo de símbolos, la planificación y la jerarquía para abordar la complejidad.

La aproximación subsimbólica, a pesar de haber sido empleada desde los principios de la robótica (las tortugas de Walter por ejemplo), virtualmente desapareció hasta finales de los 80. Hasta entonces los robots desarrollados, fundamentalmente basados en la IA simbólica, eran lentos y poco robustos, incapaces de realizar con soltura tareas aparentemente sencillas como navegar por un pasillo y reaccionar a obstáculos imprevistos. Motivado por estas limitaciones nació el paradigma reactivo, inspirado en las asociaciones estímulo respuesta típicas del conductismo. En él se genera comportamiento asociando una respuesta de actuación a ciertos estímulos sensoriales, bien directamente [6], bien a través de transformaciones sensorimotoras [7].

En los últimos años esta tendencia hacia la biología se ha intensificado y en particular hacia la etología, buscando claves que permitan simplificar y hacer más robustos los comportamientos de los robots [8]. Varios conceptos nacidos en biología como la homeóstasis se han propuesto como mecanismos para la selección de acción [9]. También los movimientos de las abejas, capaces de volver siempre a su colmena y orientarse con el sol, de las moscas equilibrando el flujo óptico en ambos ojos, sirven de ejemplo para las técnicas de navegación en robots. La percepción gestáltica y el uso de invariantes visuales perceptivas como en los picados de los cormoranes para pescar [5] puede facilitar el desarrollo de comportamientos en robots, en apariencia muy sofisticados. También son destacables los trabajos que reproducen artificialmente el comportamiento de una rana [10] o una mantis religiosa [11].

En esta línea es en la que se plantea la arquitectura para el control de robots autónomos que describimos en este artículo. JDE (Jerarquía Dinámica de Esquemas) es una arquitectura para la generación de comportamiento autónomo en robots móviles inspirada en las ideas de la etología. Se basa en la utilización de *esquemas*, que se describen en la siguiente sección, como unidades básicas. La organización de estos esquemas se realiza mediante jerarquías, que se analizan en la tercera sección. En la cuarta sección se describen algunos experimentos que muestran la organización

dinámica de los esquemas que es una de sus características más destacadas. Finalmente, la quinta sección resume algunas de las características principales así como los futuros trabajos que prevemos.

## 2. Esquemas

El encapsulamiento de funcionalidad en pequeñas unidades que luego pueden ser reutilizadas ha sido una constante desde los primeros intentos de comprender el comportamiento. A lo largo de estos años se han ideado diferentes unidades: módulos, habilidades, agentes, esquemas, comportamientos básicos, etc. cada una con sus matices y peculiaridades. En concreto, la aparición de los esquemas como parte explicativa del comportamiento surge en el campo de la fisiología y neurofisiología. Su instalación en el campo de la robótica ha recibido el apoyo de muchos investigadores, entre los que se podría destacar a Michael Arbib y Ronald Arkin<sup>3</sup>.

Dentro de JDE definimos *esquema* como un flujo de ejecución independiente con un objetivo; un flujo que es modulable, iterativo y que puede ser activado o desactivado a voluntad. Distinguimos entre *esquemas perceptivos* y *esquemas motores o de actuación*. Los esquemas perceptivos producen piezas de información que pueden ser leídas por otros esquemas. Estos datos pueden ser observaciones sensoriales o estímulos relevantes en el entorno actual, y son la entrada para los esquemas motores. Los esquemas de actuación acceden a esos datos y generan sus salidas, las cuales son comandos a los motores o las señales de activación para otros esquemas de nivel inferior (nuevamente perceptivos o motores) y sus parámetros de modulación, tal y como se ilustra en la figura 1.

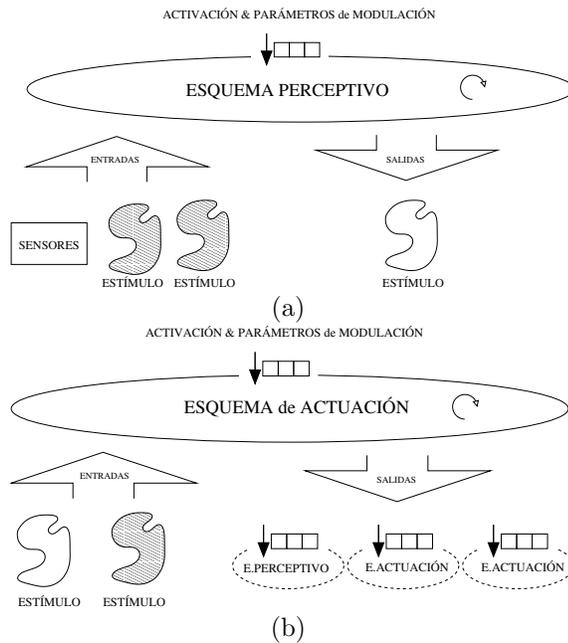
Los esquemas JDE son por definición *modulables*. Siguiendo el segundo principio de Arbib [7], referente a la evolución y la modulación, los esquemas pueden aceptar varios parámetros de entrada que modulan su propio funcionamiento haciendo que se comporte de diferentes maneras. Además todos los esquemas en JDE son procesos *iterativos*, realizan su misión en iteraciones que se ejecutan periódicamente. De hecho, el periodo de esas iteraciones es un parámetro principal de modulación de cada esquema, permitiendo que se ejecute muy frecuentemente o con menor cadencia. Los controladores digitales se pueden ver como un ejemplo de este paradigma, pues ellos entregan una acción correctora cada ciclo de control. Los esquemas son además *suspendibles*, de manera que pueden ser desactivados al final de cada iteración, y en ese caso no producirán ninguna salida hasta que sean activados nuevamente.

A los esquemas se les asocia cierto *estado*<sup>4</sup>, que ayuda a regular su coordinación, como veremos más adelante. Un esquema perceptivo puede estar en dos estados DORMIDO o ACTIVO. Cuando se encuentra ACTIVO, el esquema está actualizando las variables correspondientes al estímulo del cual se encarga. Cuando está DORMIDO las variables en sí existen, pero están sin actualizar, posiblemente desfasadas. El cambio de DORMIDO a ACTIVO o viceversa lo determinan los esquemas de nivel superior. Para los esquemas motores las cosas son un poco más elaboradas, pueden estar en cuatro estados: DORMIDO, ALERTA, PREPARADO y ACTIVO. Esto es debido a que cada esquema motor puede tener asociadas unas precondiciones y competir por el control con otros, según veremos en el apartado dedicado a la selección de acción.

Con JDE el sistema completo está formado por una colección de esquemas, siguiendo el primer principio de Arbib sobre computación cooperativa de esquemas [7]: “Las funciones del comportamiento (perceptivo-motor) y la acción inteligente de animales y robots situados en el mundo se pueden expresar como una red de esquemas o instancias de esquemas que interactúan”.

<sup>3</sup> <http://www.cc.gatech.edu/aimosaic/faculty/arkin/>

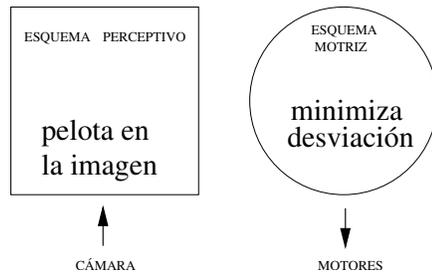
<sup>4</sup> Estado en el sentido de los autómatas, es decir, que pasa de uno a otro



**Figura 1.** Patrón típico de un esquema perceptivo (a) y de un esquema motor (b) en JDE

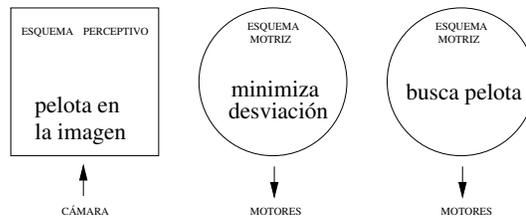
El empleo de este tipo de esquemas tiene dos implicaciones importantes: primero, la separación entre la parte perceptiva y la parte de actuación; y segundo, una fragmentación de ambas en pequeñas unidades que reciben el nombre de esquemas. La separación permite simplificar el diseño, porque la percepción y el control son dos problemas diferentes, relacionados pero distintos. Como ambos son complejos y distintos se resuelven en zonas del código separadas, eso facilita las cosas. La fragmentación en unidades pequeñas facilita la reutilización y permite acotar mejor la complejidad de cada uno de los subproblemas que aborda, haciéndolos manejables en una estrategia de divide y vencerás. Además esta separación permite dar cuerpo a cada esquema en un procesador distinto, posibilitando una implementación distribuida.

Esta división en esquemas motores y perceptivos se ha utilizado en [12,13,14,15]. Por ejemplo, en [13] se describen los comportamientos **sigue-pelota** y **sigue-pared** que se han conseguido como la conjunción de dos esquemas JDE. Tal y como muestra la figura 2 el comportamiento sigue pelota consta de un esquema perceptivo, mostrado con forma cuadrada, y un esquema motor, mostrado como un círculo (seguiremos esta notación a lo largo de todo el artículo: cuadrados para percepción y círculos para actuación). El perceptivo se encarga de buscar la pelota en la imagen y caracterizar su posición en ella, y el motor materializa un control proporcional que mueve el robot tratando de centrar la pelota en la imagen. Si la pelota aparece desviada a la izquierda, el esquema tratará de girar el robot hacia la izquierda, y de modo simétrico para desviaciones a la derecha. Igualmente, si la pelota aparece en la parte superior de la imagen, el esquema aumentará la velocidad de avance del robot porque la pelota está relativamente lejos. Si por el contrario aparece en la parte inferior significa que está demasiado cerca del robot, y el esquema hará retroceder al robot. Este esquema utiliza un control en velocidad sobre los motores de las ruedas y emplea la desviación de la posición



**Figura 2.** Comportamiento sigue pelota como activación simultánea de dos esquemas, uno perceptivo y otro motor

central de la pelota respecto del centro de la imagen como error a minimizar. La ejecución simultánea de estos dos esquemas permite al robot generar la conducta observable de seguimiento de la pelota.

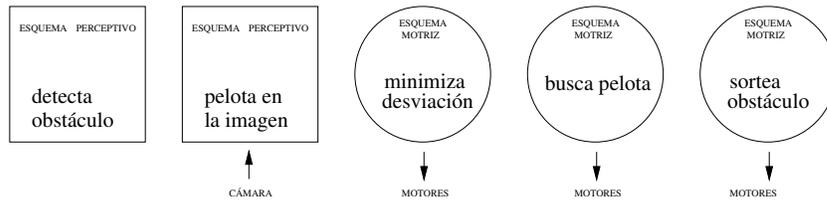


**Figura 3.** Sigue pelota con comportamiento apetitivo de búsqueda

Con estos dos esquemas, si no hay pelota alguna en la imagen entonces el esquema de control mantiene detenido los motores. Para ver la flexibilidad de esta descomposición, siguiendo con el sistema de ejemplo, una incorporación razonable podría ser un tercer esquema, también de control, que se activa cuando no aparece ninguna pelota en la imagen, y que se encargaría de mover el robot tratando de buscar las pelotas en su vecindad. Por ejemplo, girando el robot sobre sí mismo para barrer todos los alrededores. Tal y como muestra la figura 3, este esquema materializaría una conducta apetitiva, pues fomenta la aparición de la pelota en la imagen, que es el estímulo necesario para el comportamiento fundamental de seguimiento.

La pelota sería el estímulo clave que activa al esquema de seguimiento. En la figura 4 se añaden dos nuevos esquemas que enriquecen el comportamiento global con la capacidad de sortear obstáculos. Uno perceptivo para detectarlos y otro de control para sortearlos.

El siguiente paso sería englobar estos cinco esquemas como hijos de un esquema de nivel superior, que compite con otros, a su nivel, para hacer otra cosa, lo que introduce las jerarquías dinámicas que constituyen la principal aportación de JDE.



**Figura 4.** Sigue pelota complementado con sorteo de obstáculos

### 3. Jerarquía dinámica de esquemas

Una vez presentada nuestra unidad básica del comportamiento, el esquema, hay muchas opciones para su ensamblaje en un sistema completo. En este apartado daremos una visión global a la arquitectura JDE, que utiliza la jerarquía para regular el modo en el que se combinan los esquemas, y veremos también la manera en que éstos interactúan en ella y cómo unos utilizan la funcionalidad de otros para conseguir la conducta observable. Es decir, presentamos en esta sección los mecanismos de percepción y actuación que ofrece esta jerarquía como principio organizador.

Tal y como vimos anteriormente, cada esquema de JDE tiene un objetivo propio, realiza alguna funcionalidad específica en la que es experto (bien sea en control, bien sea en percepción). La jerarquía aparece por el hecho de que los esquemas pueden aprovechar la funcionalidad de otros para materializar la suya propia, y en JDE el modo que un esquema tiene de aprovechar la funcionalidad de otro es precisamente su activación y modulación.

Este patrón de activación puede repetirse recursivamente, de modo que aparecen varios niveles de esquemas donde los de bajo nivel son despertados y modulados por los del nivel superior. Las activaciones en cadena van conformando una jerarquía de esquemas específica para generar ese comportamiento global en particular.

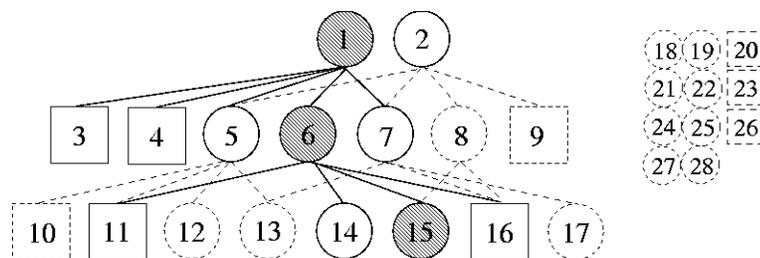
La colección de esquemas se organiza por tanto, en jerarquía para materializar cierta conducta. Esta jerarquía se reconstruye y modifica dinámicamente, según cambie el comportamiento a desarrollar o las condiciones por las que un esquema padre activa a cierto esquema hijo y no a otros.

La jerarquía que proponemos no es la clásica de activación directa, en la que el padre pone a ejecutar a un hijo durante cierto tiempo para que realice cierta misión, mientras él espera el resultado. En vez de que la misión del hijo sea un paso en el plan secuencial del padre, se entiende la jerarquía como una coactivación que expresa predisposición. En JDE un padre puede activar a varios hijos a la vez, porque no es una puesta en ejecución en la cual los hijos emitan directamente sus comandos a los actuadores, sino un situar en alerta. La activación final se deja en manos del entorno y de la competición con otros hermanos.

La forma en que se implementa ese proceso de activación y selección es usando cuatro estados: DORMIDO, ALERTA, PREPARADO y ACTIVO. El paso de DORMIDO a ALERTA lo determina la preactivación del padre. El paso de ALERTA a PREPARADO lo determinan las precondiciones del hijo, que él mismo evalúa periódicamente. El paso de PREPARADO a ACTIVO se pelea en una competición por el control entre los hermanos preparados en ese nivel.

En cada instante hay varios esquemas en ALERTA por cada nivel, ejecutándose concurrentemente, pero sólo uno de ellos es activado por la percepción del entorno. Cuando ningún esquema o más de uno quiere ser activado, entonces el esquema de nivel superior se invoca para que arbitre cual de ellos

toma realmente el control. Por ejemplo, la figura 5 muestra una instantánea de la arquitectura JDE en funcionamiento. Los esquemas motores en ALERTA aparecen como círculos con bordes continuos. Sería el caso de los esquemas 5, 6 y 7. Varios de estos pueden ser compatibles con la situación perceptiva del entorno, pasando a PREPARADO, pero sólo uno de ellos ganará la competición por el control en ese nivel y pasará a ACTIVO. Los esquemas en ACTIVO se muestran en la figura 5 como círculos rellenos, como el 1, el 6 y el 15.



**Figura 5.** Jerarquía de esquemas y batería de esquemas en DORMIDO.

Esta idea de predisposición ya aparece en las propuestas de jerarquía aparecidas en etología. Además esta interpretación es compatible con que se pongan en alerta varios esquemas que realizan la misma función, y dependiendo de la situación del entorno, sólo se activará el más adecuado a ese contexto. Tal y como señala Tinbergen para ejemplificar su jerarquía [16], el halcón que sale de caza tiene preactivados, predispuestos los módulos de *cazar-conejos*, *cazar-palomas*, etc. mientras sobrevuela su territorio de caza. Todos ellos satisfacen la finalidad de alimentarlo, se activará realmente uno u otro dependiendo de la presa que encuentre. Es lo que Timberlake [17] llama *variabilidad restringida*, y que tiene perfecta cabida en JDE. Esta capacidad de tener varias alternativas predispuestas de modo simultáneo contrasta con las alternativas que tiene RAP de Firby [18,19,20] (y por ello que hereda la arquitectura híbrida 3T [21]), que se ensayan una detrás de otra, pero sólo cuando la anterior ha fracasado.

Desde el punto de vista de los hijos, un esquema en estado activo sólo puede tener un único padre en un instante dado. Es decir, un hijo no puede tener varios padres en el mismo instante. Sí puede ocurrir que sea activado por distintos padres en diferentes momentos de tiempo. También puede suceder que se activen simultáneamente instancias diferentes del mismo esquema, probablemente con modulaciones distintas y en niveles diferentes. Esta posibilidad es un ejemplo de reutilización de esquemas.

Los esquemas que un momento dado no se usan para la tarea en curso descansan en una batería de esquemas, suspendidos en estado DORMIDO, pero preparados para la activación en cualquier momento. Éstos aparecen como cuadrados y círculos discontinuos en el lateral derecho de la figura 5 (esquemas 8, 9, 10, 12, 13, 17, 18, etc.).

Las jerarquías se construyen y modifican dinámicamente y son específicas de cada comportamiento. Si las condiciones del entorno o los objetivos del robot varían, puede ocurrir que entre los hermanos que están ALERTA en cierto nivel cambie la relación de cuáles de ellos están preparados para afrontar la nueva situación. Esto altera quién es el ganador en la competición por el control en ese nivel y fuerza una reconfiguración de la jerarquía desde ese nivel hacia abajo. Todos los que

estaban activos por debajo se desactivan y el nuevo árbol se establece a partir del nuevo ganador. Por ejemplo, los esquemas 10, 12 y 13 de la figura 5 están a un sólo paso de la activación, pues serán despertados si el esquema 5 pasa a ACTIVO en su nivel.

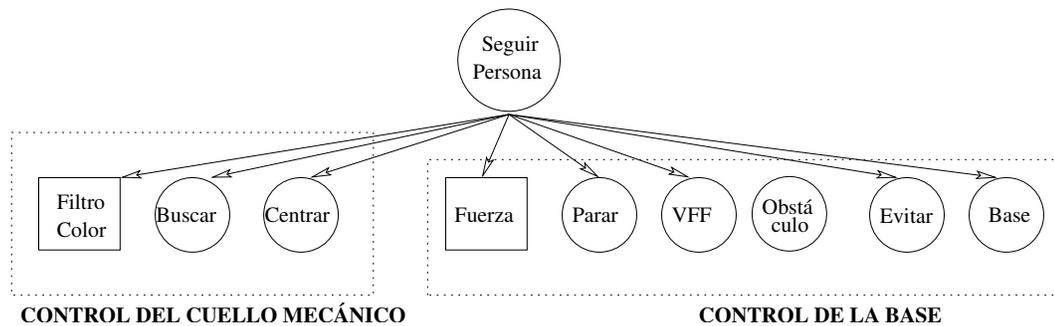
Del mismo modo, si la situación del entorno o los objetivos se alteran, cierto esquema en determinado nivel de la jerarquía actual puede decidir modificar los parámetros de sus hijos actuales, o incluso cambiar de hijos. De nuevo, esto fuerza una reconfiguración de la jerarquía desde ese esquema hacia abajo, pues los nuevos hijos al ejecutarse activarán a otros hijos suyos. Las reconfiguraciones en JDE suelen ser rápidas puesto que el arbitraje y las decisiones de cada esquema de control se toman periódicamente, a un ritmo suficientemente vivaz.

La jerarquía incluye la percepción como parte subsidiaria de los esquemas de actuación, y gracias a ello se facilita la coordinación, esto es, se da un contexto a la percepción. Para cada comportamiento hay que ligar los esquemas perceptivos a los de actuación, con ello se resuelve la coordinación percepción-actuación. Por ejemplo, la coordinación visuo-motora en el movimiento. La jerarquía determina qué percibir y cuándo.

Tal y como muestra la figura 5 hay reutilización de esquemas perceptivos puesto que el mismo esquema perceptivo puede asociarse a distintos esquemas motores. En realidad son distintas instancias del mismo esquema. Por ejemplo, el esquema 11 puede dar percepción al esquema motor 6 y al 5, o el 16 al 6, 7 y 8. Puede haber estímulos específicos, como el que elabora el 10, que sólo le interesa al esquema motor 5, pero en general la reutilización es la norma en percepción, para estímulos comunes. También puede haber reutilización de esquemas motores.

#### 4. Experimentos con JDE

Validar una arquitectura para la generación de comportamiento autónomo es muy difícil. En nuestro caso hemos decidido que el único camino es la construcción de diversas aplicaciones, de las más simples a otras más complejas. Así hemos ido desarrollando diferentes aplicaciones como las descritas en [12,13,14].



**Figura 6.** Jerarquía de esquemas para que un robot siga a una persona

Para ilustrar su uso vamos a describir brevemente dos aplicaciones. La primera es la reflejada en la figura 6. Se trata de la construcción del software para que un robot con visión monocular sea

capaz de seguir a un humano en un entorno de interiores con desconocido, es decir, con todo tipo de obstáculos.

En esta aplicación, al ser sencilla, se puede resolver con un único nivel. En él se distinguen dos entornos de control diferentes (marcados mediante dos cuadrados punteados). Uno para el control del cuello mecánico (*pan-tilt*) y otro para el control de la base. El primero está formado por tres esquemas: buscar, centrar y filtrar. Los dos primeros motores y el tercero perceptivo. El segundo está formado por un esquema perceptivo, encargado de proporcionar las fuerzas (denominado Fuerza en la figura) y cuatro motores para implementar la evitación de obstáculos, que detiene al robot y gira si aparece un obstáculo demasiado cerca, VFF para el control de robot con obstáculos a una distancia prudencial y el “Base” que se encarga de alinear la base con el cuello. Se ha añadido un esquema adicional de seguridad “Parar” para detener al robot al alcanzar el objetivo, o cuando se ha perdido. Se pueden encontrar los detalles de esta aplicación en [15].



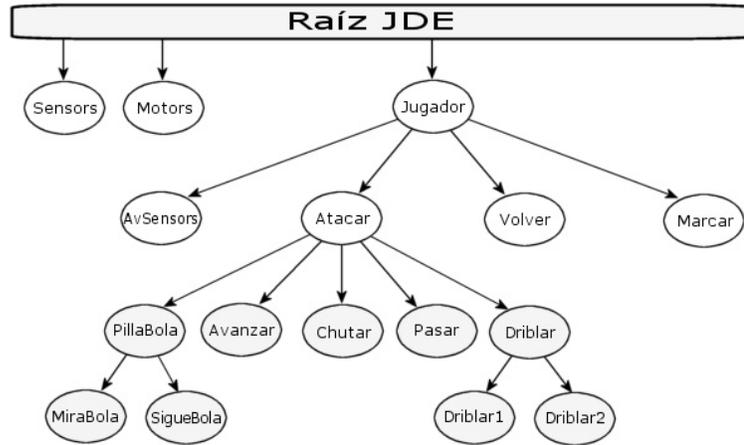
**Cuadro 1.** Ejemplo de uso de JDE en un robot móvil

Las imágenes del cuadro 1 resumen el funcionamiento de la aplicación. Se puede observar el robot, una plataforma Pioneer fabricada por ActivMedia controlada por un ordenador portátil estándar y con una cámara de vídeo-conferencia. Las cuatro imágenes de la parte izquierda de la figura reflejan la evitación de un obstáculo, mientras sigue al móvil. Las cuatro de la derecha muestran el control separado del cuello mecánico y de la base, así se aprecia como inicialmente el seguimiento se realiza mediante el cuello mecánico, y como posteriormente la base se alinea con el cuello.

Un ejemplo de uso de JDE en el que se evidencia la necesidad de la jerarquía es el descrito en [22] en el que se construyó la jerarquía necesaria para controlar un robot simulado capaz de jugar al fútbol.

La figura 7 refleja la forma en la que se organizan los esquema motores, en particular para un jugador ofensivo. Así, en el tercer nivel de la jerarquía, se puede observar como los esquemas “Chutar”, “Pasar”, “Avanzar”, etc. estarán concurrente evaluando sus posibilidades, siendo las condiciones del entorno las que decidan cual de ellos será el que tome el control.

De esta forma, la construcción de comportamientos autónomos se simplifica enormemente, como en los sistemas deliberativos clásicos, pero sin perder además las ventajas de la reactividad de los



**Figura 7.** Jerarquía de esquemas para un jugador robótico de fútbol

sistemas basados en comportamientos. Todas estas consideraciones se analizan con mayor detalle en la siguiente sección.

## 5. Conclusiones y Trabajo Futuro

La idea central presentada en este artículo, y subyacente en JDE, es la creencia en la composición de comportamientos [23]. En nuestro caso entendiendo la composición en dos sentidos: temporal y en abstracción conceptual. En sentido temporal con JDE se argumenta que *una gran variedad de tareas motoras se puede describir y conseguir en términos de secuenciación de varios esquemas*, exactamente como enuncia el primer principio de Arbib [7]. En sentido conceptual, JDE apuesta por la ejecución simultánea de varios niveles de abstracción, cada uno de los cuales se materializa en otro nivel inferior y puede en cada momento materializarse en otro diferente de bajo nivel para adaptarse a las distintas condiciones del entorno.

Este planteamiento separa la abstracción conceptual de la abstracción temporal, que son indisolubles en otras unidades de comportamiento como las unidades de acción [4] o los RAP [18]. En JDE los elementos de muy alto nivel de abstracción se pueden traducir en esquemas tan rápidos como se necesite. Se asciende en nivel de abstracción obligando a que las salidas de los esquemas de alto nivel sean exclusivamente activaciones de los esquemas de nivel inferior y su correspondiente modulación. Todos ellos funcionan en paralelo y toman decisiones en cada instante, cada uno en su nivel de abstracción. Un nivel más alto se distingue de otro inferior porque utiliza variables de estímulos que semánticamente son más abstractas, no por tener diferente velocidad.

La influencia de planteamientos etológicos en esta propuesta es muy grande. Por ejemplo, guarda muchas similitudes con la jerarquía que propone Tinbergen [16] para explicar la generación de comportamientos instintivos. Los esquemas de control son análogos a sus centros nerviosos. Como hemos señalado anteriormente, otra similitud es la utilización de la jerarquía para generar predisposición de ciertos comportamientos cuando se activa cierto instinto de nivel superior. Curiosamente

las propuestas modernas de Arkin [24] también se apoyan en esta idea de jerarquía, donde se admite además la influencia que han ejercido en su sistema las ideas nacidas en la etología. Como valor añadido sobre el trabajo de Tinbergen y Lorenz, JDE explicita también la organización de la percepción, tal y como hemos visto.

JDE además permite la implementación directa de conceptos como los *estímulos clave* o los *comportamientos apetitivos*, cuya definición nació en el campo de la etología. Por ejemplo, el *comportamiento apetitivo* o de apetencia lo acuñó el biólogo Craig en 1926 y se utiliza para designar a aquella conducta que no satisface directamente ninguna necesidad del animal, sino que busca la situación desencadenante de la acción final, del *comportamiento consumatorio*, que es el que realmente sacia la necesidad interna [25,26]. Una muestra de conducta consumatoria podría ser comer una presa, mientras que buscarla sería apetitiva.

Los *estímulos clave*, por su parte, se definen como aquellos que disparan la activación de cierto patrón de comportamiento. En JDE se puede implementar de modo sencillo un esquema perceptivo que busca en la realidad el estímulo que desencadena el paso de ALERTA a PREPARADO de cierto esquema motor, promoviendo su activación. En general cada esquema motor tiene su *estímulo clave* [10], y reacciona ante él cuando está presente.

Estos conceptos "prestados" de la etología son los que se han conseguido en los experimentos descritos que validan la aproximación propuesta. El ejemplo de seguimiento permite certificar la capacidad de JDE para enfrentarse a entornos reactivos", siendo capaz de enfrentarse a todo tipo de cambios en el entorno. De igual forma, el ejemplo del jugador de fútbol permite validar JDE en lo referente a la orientación a objetivos, haciendo que el sistema, a lo largo del tiempo, vaya utilizando una serie de comportamientos básicos dependiendo de la situación, pero siendo finalista, es decir, cuando mejor contribuyen al objetivo general del jugador.

Finalmente, los trabajos futuros están orientados a aumentar la experimentación con jerarquía y escalabilidad. El experimento presentado en este artículo para justificar la jerarquía ha sido probado únicamente en un entorno simulado. Estimamos necesario evaluarlo sobre un robot real, trabajo que está actualmente en curso. Igualmente, se están aprovechando estos nuevos experimentos para probar una versión ampliada del código (denominada JDE+) implementada en C++ (la descrita en este artículo se realizó en C) y con nuevas funcionalidades. Igualmente, se están estudiando los mecanismos para emplear eficazmente la memoria a largo plazo y las técnicas de visión.

## 6. Agradecimientos

El trabajo descrito en este artículo ha sido parcialmente financiado por el proyecto ACRACE, financiado por el Ministerio de Educación y Ciencia (DPI2004-07993-03-01) y por el proyecto RoboCity 2030 (S-0505/DPI/000176) de la Comunidad Autónoma de Madrid.

## Referencias

1. Walter, G.: An imitation of life. Scientific American (1950)
2. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in neural nets. Bulletin of Mathematical Biophysics (1943)
3. Wiener, N.: Cybernetics. Wiley and Sons (1948)
4. Nilsson, N.: A mobile automaton: an application of artificial intelligence techniques. In: Proceedings of the 1st International Joint Conference on Artificial Intelligence IJCAI, Washington, (USA) (1969) 509–520

5. McFarland, D., Bösser, T.: Intelligent behavior in animals and robots. The MIT Press (1993) ISBN-0-262-13293-1.
6. Brooks, R.A.: A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* **2**(1) (1986) 14–23
7. Arbib, M.A., Liaw, J.S.: Sensorimotor transformations in the worlds of frogs and robots. *Artificial Intelligence* **72** (1995) 53–79
8. Mallot, H.A., Franz, M.O.: Biomimetic robot navigation. *Autonomous Systems* **20** (1999) 133–153
9. Tyrrell, T.: An evaluation of Maes' "bottom-up mechanism for behavior selection". *Journal of Adaptive Behavior* **2**(4) (1994) 307–348
10. Corbacho, F.J., Arbib, M.A.: Learning to detour. *Adaptive Behavior* **5**(4) (1995) 419–468
11. Arkin, R., Kahled, A., Weitzenfeld, A., Cervantes-Prez, F.: Behavioral models of the praying mantis as a basis for robotic behavior. *Journal of Autonomous Systems* **32**(1) (2000) 39–60
12. Cañas, J.M., Martínez, M., Bustos, P., Bachiller, P.: Overt visual attention inside jde control architecture. In: *Proceedings of the IROBOT Workshop inside Portuguese Conference on Artificial Intelligence EPIA2005*. (2005)
13. Gómez, V., Cañas, J.M., Martín, F.S., Matellán, V.: Vision based schemas for an autonomous robotic soccer player. In: *Actas del IV Workshop de Agentes Físicos, WAF'2003, Universidad Alicante* (2003) 109–120 ISBN 84/607-7171-7.
14. Lobato, D.: Navegación local con ventana dinámica para un robot móvil. Proyecto fin de carrera, Universidad Rey Juan Carlos (2003)
15. Calvo, R., Cañas, J.M., García, L.: Person following behavior generated with jde schema hierarchy. In: *ICINCO, 2nd Int. Conf. on Informatics in Control, Automation and Robotics, Barcelona (Spain), INSTICC Press* (2005) 463–466
16. Tinbergen, N.: The hierarchical organization of nervous mechanisms underlying instinctive behavior. *Symposia of the Society for Experimental Biology* **4** (1950) 305–312
17. Timberlake, W.: Motivational models in behavior systems. In R.R. Mowrer, S.K., ed.: *Handbook of contemporary learning theories*. Hillsdale, NJ: Erlbaum Associates (2000) 155–209
18. Firby, R.J.: An investigation into reactive planning in complex domains. In: *Proceedings of the 6th AAAI National Conference on Artificial Intelligence, Seattle, WA* (1987) 202–206
19. Firby, R.J.: Buiding symbolic primitives with continuous control routines. In: *Proceedings of the 1st International Conference on AI Planning Systems AIPS'92, College Park, MD (USA)* (1992) 62–69
20. Firby, R.J.: Task networks for controlling continuous processes. In: *Proceedings of the 2nd International Conference on AI Planning Systems AIPS'94, Chicago, IL (USA)* (1994) 49–54
21. Bonasso, R.P., Firby, R.J., Gat, E., Kortenkamp, D., Miller, D.P., Slack, M.G.: Experiences with an architecture for intelligent reactive agents. *Journal of Experimental and Theoretical AI* **9**(2) (1997) 237–256
22. Martínez Gil, J.: Equipo de futbol con jde para la liga simulada robocup. Proyecto fin de carrera, Universidad Rey Juan Carlos (2003)
23. Aguirre, E., García-Alegre, M., González, A.: A fuzzy safe follow wall behavior fusing simpler fuzzy behaviors. In: *Proceedings of the 3rd IFAC Symposium on Intelligent Autonomous Vehicles IAV'98, Madrid* (1998) 607–612
24. Arkin, R.C., Fujita, M., Takagi, T., Hasegawa, R.: An ethological and emotional basis human-robot interaction. *Robotics and Autonomous Systems* **42** (2003) 191–201
25. Vogel, G., Angermann, H.: *Atlas de biología*. Ediciones Omega (1974)
26. Lorenz, K.: *Fundamentos de la etología*. Ediciones Paidós (1978)