

# Integración de la Arquitectura Cognitiva SOAR en un Entorno ROS sobre un Parrot AR.Drone 2.0\*

Sai Kishor Kothakota, Cecilio Angulo

Universitat Politècnica de Catalunya · UPC BarcelonaTech

Pau Gargallo 5. ESII – Departament d'Automàtica, Barcelona, Spain 08028

saisastra3@gmail.com, cecilio.angulo@upc.edu

## Abstract

En la vida diaria, el uso de aplicaciones de Inteligencia Artificial (IA) está creciendo. Sin embargo, todavía se necesita de un mayor apoyo para su integración en los robots, lo que conllevará el desarrollo de seres inteligentes capaces de tomar decisiones. Los algoritmos de inteligencia artificial hacen el trabajo más fácil y eficaz. En este documento integramos una Arquitectura Cognitiva (SOAR) desde el dominio de la IA computacional en un Parrot AR.Drone 2.0, un drone cuatrimotor. El principal objetivo es hacerse cargo de la misión con el Parrot AR.Drone 2.0 mediante el establecimiento de una conexión entre el sistema operativo ROS, con la arquitectura cognitiva SOAR, en el que la información de los sensores de aviones no tripulados se utiliza para llevar a cabo la tarea requerida. Este trabajo tiene como objetivo la comunicación bidireccional entre SOAR y ROS, a la vez que interactuar con los seres humanos.

## 1. Introduction

La robótica móvil es un área de investigación de rápido desarrollo tecnológico. Los robots móviles se utilizan para aplicaciones específicas, pero en general carecen de la habilidad para realizar acciones donde se producen toma de decisiones en escenarios en tiempo real, así como también la interacción con humanos en que existe ambigüedad [Canal *et al.*, 2015].

El enfoque habitual para la planificación en robots móviles inteligentes se basa principalmente en la tecnología de máquinas de estado, de forma que se consigue un objetivo determinado con un plan completo adecuado disponible antes de su aplicación. Este plan completo es una combinación secuencial de habilidades sencillas del robot para lograr el objetivo. Puesto que ninguna retroalimentación desde el entorno se considera, el robot se comporta de la misma manera en diferentes situaciones del entorno. Así, utilizando una planificación completa desde el principio el robot procede con comandos que tienen poco conocimiento sobre el entorno [Puigbo

*et al.*, 2015]. El principal inconveniente de este enfoque es que se necesita demasiada información anterior al inicio de la toma de decisiones, que impide al robot reaccionar ante situaciones nuevas.

La alternativa más común a las máquinas de estado son los planificadores. Los planificadores suelen basarse en criterios probabilísticos que determinan la mejor combinación de habilidades en tiempo real para un objetivo específico. Un enfoque diferente a los planificadores es el uso de razonadores, es decir el uso de arquitecturas cognitivas, que ha atraído una renovada atención tanto de los académicos como de la industria [Besold *et al.*, 2014]. Así, una arquitectura cognitiva se ha aplicado a un robot de servicio general en [Puigbo *et al.*, 2015] mediante el establecimiento de una comunicación unidireccional desde la arquitectura al robot.

La comunicación bidireccional entre entorno y robot y el uso de una arquitectura cognitiva ayudarían a procesar la meta con planes incompletos en base a información retardada procedente del entorno. La arquitectura cognitiva con planificación incompleta también podría utilizar la información disponible por parte del usuario para construir planes. Por lo tanto, los robots que utilizan este tipo de arquitectura podrían volver a planificar de acuerdo a los datos ambientales y también podría establecerse un entorno interactivo de usuario, lo que hace innecesario el suministro de datos enormes de antemano y completar el plan a medida que se alcanza el objetivo especificado.

Existen varias arquitecturas cognitivas en la literatura [Langleya *et al.*, 2009] como SOAR [Cho *et al.*, 1991; Laird *et al.*, 2004; Young and Lewis, 1997], ACT-R [Anderson *et al.*, 2004; Anderson, 2007; Trafton *et al.*, 2005], CRAM, SS-RICS [Kelley, 2006; Kelley and Avery, 2010]. Todas ellas fueron evaluadas en [Puigbo *et al.*, 2015] en un escenario robótico de propósito general en función de su generalización, el razonamiento, el aprendizaje, la capacidad de cumplir subobjetivos y su escalabilidad. Utilizando esa discusión previa, también se ha seleccionado SOAR como arquitectura más adecuada para este estudio.

SOAR es una arquitectura cognitiva que ha estado en continuo desarrollo desde 1980 que ofrece la capacidad de comunicarse con el ambiente externo. SOAR dispone de otras características especiales que, en caso de no existir un plan inicial para lograr el objetivo, permite poner en práctica varios mecanismos y conocimientos disponibles, tales como frag-

\*Este trabajo ha sido financiado en parte a través del proyecto de investigación PATRICIA (TIN2012-38416-C03-01) por el Ministerio de Economía y Competitividad español.

mentación, aprendizaje por refuerzo, y la capacidad de subobjetivos. Esta arquitectura cognitiva permite seleccionar la habilidad necesaria para la situación actual de cara a lograr el objetivo especificado sin necesidad de ninguna lista predefinida de situaciones y planes. Este será un elemento clave para establecer una comunicación bidireccional con el entorno [Laird, 2008]. Por lo tanto, SOAR proporciona una buena plataforma para establecer dicha comunicación y la interacción con el usuario.

Por otra parte, SOAR se ha aplicado recientemente para el control de un robot humanoide de servicio de propósito general [Puigbo *et al.*, 2015] para resolver tareas complejas usando las habilidades básicas de robot humanoide de servicio como la navegación, captar el entorno y reconocimiento de objetos. Sin embargo, solamente la comunicación unidireccional entre SOAR y ROS está disponible, y no se está considerando la intervención del usuario.

Por lo tanto, el objetivo principal de este trabajo es el desarrollo de una comunicación bidireccional entre SOAR y ROS. La bidireccionalidad puede ser utilizada en entornos interactivos, de forma que el robot actúe de manera diferente en situaciones particulares. El resto del trabajo se organiza de la siguiente manera: la arquitectura implementada se introduce en la Sección 2; en la Sección 3 se describe la plataforma robot; la Sección 4 destaca los principales resultados obtenidos durante la experimentación; por último, se ofrecen algunas conclusiones y futuras líneas de investigación.

## 2. Comunicación SOAR-ROS bidireccional

SOAR ha sido seleccionada como arquitectura cognitiva en nuestro enfoque para la planificación del robot. En SOAR, un *estado* es una representación de la situación del entorno ante la resolución del problema en curso, un *operador* transforma un estado (produciendo cambios en la representación) y un *objetivo* es un resultado deseado en la actividad de resolución de problemas, es decir, la solución virtual que representa la manera de solucionarla.

El sistema global propuesto se compone de tres módulos que están conectados mutuamente. La arquitectura cognitiva envía comandos de acuerdo a los datos en tiempo real recibidos. A continuación, la interfaz entre SOAR y ROS, que es nuestra implementación, recibe órdenes de SOAR, las procesa en comandos de ROS y luego toma los datos del entorno de ROS o de otras fuentes, los procesa para ser enviados a la arquitectura cognitiva SOAR y así procesar futuras decisiones basadas en los datos de entrada.

La interfaz para comunicarse con el ambiente externo o robots mediante el envío y recepción de comandos se programa con SML (Soar Markup Language), donde los comandos se empaquetan como paquetes XML y se envían. El entorno y el depurador que soporta se conocen como *clientes*.

La comunicación bidireccional propuesta entre SOAR y ROS será posible considerando el entorno SOAR como el elemento principal. Por lo tanto, los comandos enviados desde ROS se procesan en SOAR y los comandos de SOAR a ROS se procesan correctamente en SOAR y formatean para una evaluación fácil en ROS. Mediante el uso de la comunicación en esta forma, tanto el usuario como las entradas

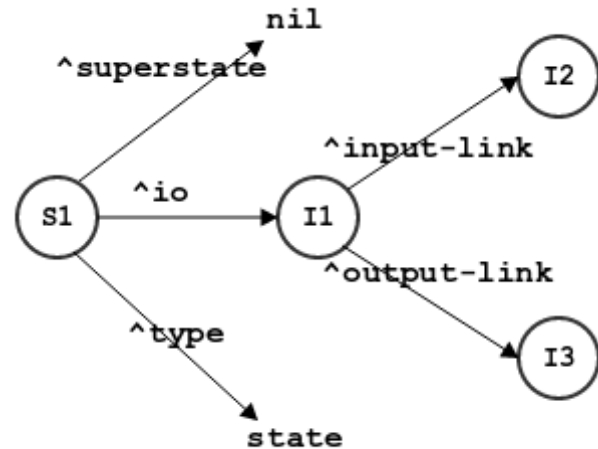


Figura 1: Imagen gráfica de las estructuras de memoria de trabajo que SOAR genera automáticamente.

desde el entorno del robot se pueden procesar en SOAR para la toma de decisiones. Por lo tanto, esta interfaz orientada a SOAR permitirá bidireccionalidad, así como la intervención del usuario.

### 2.1. La memoria de trabajo

La información almacenada de corto plazo y las estructuras probadas de las reglas generadas serán colocadas en la *memoria de trabajo*. Así se construye la estructura de grafos de los elementos, donde se crea cada elemento a partir de las reglas y de los sensores o los datos del entorno externo. La arquitectura SOAR crea automáticamente algunas de las estructuras de memoria de trabajo para todos los agentes, cuyos elementos se construyen como tripletas: identificador, atributo y valor. SOAR automáticamente crea un atributo *io*, que significa *input-output* para comunicarse con el mundo exterior. Hay dos atributos para *io* (véase la Figura 1): *input-link* y *output-link*:

- El atributo *output-link* en la estructura de la memoria de trabajo se utiliza para enviar comandos al mundo exterior con el fin de realizar algunas acciones con cada decisión de la arquitectura SOAR. El *output-link* es donde los comandos de acción deben ser creados para los agentes en su mundo.
- El atributo *input-link* en la estructura de la memoria de trabajo se utiliza para obtener la información del mundo exterior o de los sensores. Normalmente, para la tarea que involucra la información desde el entorno externo la mayoría de los estados se construyen a partir de la información perceptual disponible de los sensores. Esta información se crea en el *input-link* de la estructura del estado. Puede también ser utilizado para tomar la entrada del usuario y hacer que esté disponible en la estructura *input-link*.

## 2.2. Envío y de recepción de comandos en SOAR

SOAR ayuda al usuario a comunicar la información al proporcionar las estructuras `input-link` y `output-link` de la memoria de trabajo. Usando estas estructuras es posible enviar comandos de SOAR y hacer operaciones con éxito. En forma similar, podemos usar la estructura `input-link` para obtener datos en SOAR (desde ROS en nuestro caso) con fines computacionales o de toma de decisión. Así, se pueden obtener los datos de los sensores de a bordo en el robot y utilizarlos para la toma de decisiones.

## 2.3. El cliente de interfaz

Con el fin de construir la comunicación con el entorno ROS, un cliente de interfaz debe ser diseñado de manera que pueda acceder a las estructuras `input-link` y `output-link` para el procesamiento de la comunicación de envío / recepción de datos en SOAR. Esta conexión de la interfaz a la arquitectura cognitiva SOAR se puede establecer mediante clientes SML (Soar Markup Language). Estos clientes ayudan a la interfaz a establecer una conexión correcta para la transferencia de datos entre ellos.

SOAR ofrece diferentes archivos de clientes SML para diferentes lenguajes de programación (C++, Python, Java) y estos archivos se pueden utilizar para establecer la conexión entre la interfaz programada y la arquitectura cognitiva SOAR.

Acciones seleccionadas por la arquitectura SOAR se traducirán en la activación de una habilidad en el entorno ROS por medio de la interfaz. Estos comandos ROS luego se reflejan en las acciones del robot, que son los pasos para lograr el objetivo específico.

Los clientes SML utilizados desde el archivo de interfaz ayudan a la interfaz a establecer una conexión con el núcleo de SOAR. El cliente puede ayudar ya sea en la creación de un kernel local de SOAR o una conexión remota a un kernel existente SOAR. Este objeto núcleo se puede utilizar para crear un agente de SOAR. Los agentes creados en SOAR se encargan de la gestión de los elementos de memoria de trabajo. Este agente ayuda a obtener información de la arquitectura SOAR y también del envío de información hacia él (ver Figura 2).

## 3. Configuración de la experimentación

Para la experimentación de la comunicación bidireccional entre la arquitectura cognitiva SOAR y el entorno ROS se ha empleado el UAV (Unmanned Aerial Vehicle) Parrot AR.Drone 2.0 (ver Figura 3). Este UAV está equipado con numerosos sensores a bordo y un controlador para mantener un vuelo estable, por lo que puede mantenerse en un lugar de espera antes de recibir el siguiente comando.

El Parrot AR.Drone es una interesante plataforma de investigación para la visión por ordenador y la exploración robótica [Krajník *et al.*, 2011]. En el proceso de experimentación, SOAR empleará habilidades simples del robot, como el aumento / disminución de la altitud, movimiento hacia adelante / atrás, 'rolling' a la derecha / izquierda, giro hacia la derecha / izquierda, con el fin de lograr un objetivo específico (véase el Cuadro 1. El Parrot AR.Drone se controlará mediante entorno ROS por lo que la ejecución de órdenes de ROS en la interfaz

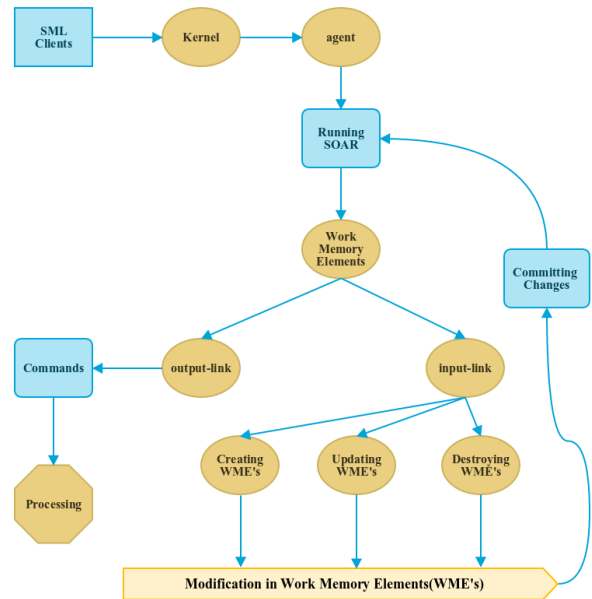


Figura 2: Desarrollo de interfaz del cliente SML.



Figura 3: Parrot AR.Drone 2.0.

se procesa a partir de los comandos decididos en SOAR para que el avión no tripulado pueda llevar a cabo las habilidades o acciones.

Parrot AR.Drone 2.0 responde a los comandos ROS con el uso de los paquetes `ardrone_driver` y `ardrone_autonomy` en el entorno ROS<sup>1</sup>. Estos controladores ayudan en la interfaz a que las órdenes recibidas desde la arquitectura cognitiva SOAR sean procesadas en el entorno de ROS. Este entorno proporciona los datos de navegación de los sensores de a bordo, antes, durante y después del vuelo. Las lecturas de los sensores disponibles son, entre otros, las señales de altitud, magnetómetro, giroscopio y acelerómetro, las lecturas del barómetro, el valor de la temperatura, el ángulo de viento. Estos pueden ponerse a disposición de SOAR y ayudar en la toma de decisiones.

<sup>1</sup><http://autonomylab.org/>

Habilidad	Acción
Adelante	<i>Pitch</i> adelante
Atrás	<i>Pitch</i> atrás
Arriba	Incrementa altitud
Abajo	Decrementa altitud
Lateral	<i>Rolling</i> derecha
Lateral2	<i>Rolling</i> izquierda
Derecha	<i>Yaw</i> derecha
Izquierda	<i>Yaw</i> izquierda

Cuadro 1: Habilidades del robot disponibles y sus acciones asociadas.

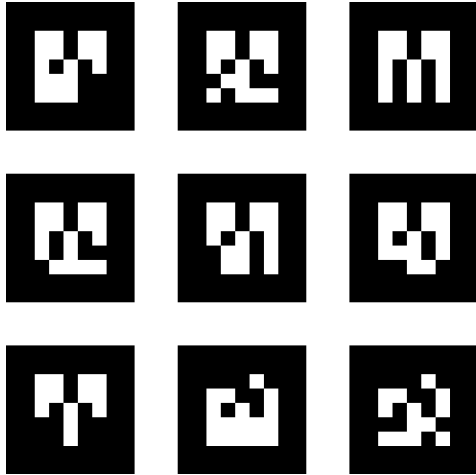


Figura 4: Augmented Reality (AR) markers.

El objetivo en la experimentación será el de encontrar un marcador de Realidad Aumentada (AR) durante el vuelo del UAV y, eventualmente, aterrizar en base a la aceptación por parte del usuario. La detección de marcadores AR se llevará a cabo con el paquete *ar\_pose*, que es un estimador de pose de marcadores de realidad aumentada que utiliza ARToolkit [Amin and Govilkar, 2015]. El UAV buscará un marcador de AR durante su vuelo y enviará la información sobre su presencia a la arquitectura SOAR a través de la interfaz. Estos datos son utilizados por SOAR para la toma de decisiones. Algunos de los marcadores AR empleados se muestran en la Figura 4.

Con este experimento podemos mostrar la comunicación bidireccional entre SOAR y ROS, el envío y recepción de comandos entre SOAR y ROS en la interfaz desarrollada, así como la forma en que el usuario también puede enviar comandos a fin de completar el plan, aceptando el aterrizaje propuesto por el dron o eventualmente rechazarlo. La experimentación se ha hecho disponible en <http://tiny.cc/ardrone>.

La aplicación ha sido diseñada en un ambiente cerrado pero inconcontrolado, un pasillo en un edificio público. Para alcanzar el objetivo, el robot puede trabajar en dos tipos de interacciones:

- **Categoría 1.** Uso de comunicación bidireccional entre la arquitectura cognitiva SOAR y el entorno ROS, como por ejemplo los datos de presencia de marcadores AR re-

lacionados con los subcomandos permitidos en el robot, que son los definidos en la Tabla 1.

- **Categoría 2.** Intervención del usuario, cuyos datos pueden ser utilizados para la toma de decisiones.

En la Categoría 1, en función de la presencia o no de un marcador AR, el módulo SOAR toma decisiones en forma de retroalimentación del sistema: el módulo SOAR modifica el plan en tiempo real de manera que se logre el objetivo tras sentir el entorno. Durante la operación, se busca la presencia de marcadores AR y se envía al módulo de SOAR el resultado antes de tomar la siguiente decisión; a su vez, esta información es utilizada por SOAR para tomar decisiones ya que así conoce si ha encontrado un marcador AR en la operación anterior. El robot, tras encontrar un marcador AR, aterrizará en esa posición, si así lo determina el usuario.

En la Categoría 2, se le pregunta al usuario si debe aterrizar el UAV después de haber encontrado el marcador AR: “*Enter 1 for landing, 0 for hovering some time before landing*”. Dependiendo de la entrada del usuario, el robot o bien aterriza inmediatamente o puede hacerlo tras flotar durante un determinado período de tiempo. Así se introduce la intervención del usuario en la comunicación SOAR-ROS.

## 4. Resultados

La comunicación bidireccional SOAR-ROS ha sido probada en un entorno cerrado para el proceso de búsqueda por parte de un Parrot AR.Drone 2.0 de un marcador AR. En esta experimentación, el robot realiza operaciones basadas en los comandos de ROS que se procesan en la interfaz a partir de los comandos de decisión recibidos de la arquitectura SOAR y luego devuelve información acerca de la detección de un marcador de AR a SOAR para la toma de decisiones. En función de la decisión del usuario, el plan cambia con el tiempo y las habilidades están determinadas por el módulo SOAR para lograr el objetivo de una manera apropiada. Durante la ejecución han sido probadas las dos categorías.

La experimentación implica proveer el sistema robot con una entrada de usuario por medio de línea de comandos, y la comprobación de que el robot ha sido capaz de realizar las acciones necesarias para completar el objetivo. Situaciones en las que se prueba el robot son:

**Categoría 1.** :“*Búsqueda de marcador AR y aterrizaje*”.

La secuencia de las acciones realizadas por el robot es: *hacia adelante, verifica si existe marcador AR x 5, lado, verifica si existe marcador AR, hacia atrás, verifica si existe marcador AR x 5, lateral, verifica si existe marcador AR, hacia adelante, verifica si existe marcador AR x 5* y así sucesivamente (hasta un cierto límite de operaciones laterales) lo que demuestra que no es necesaria la planificación completa inicial en este caso.

**Category 2.** :“*Intervención del Usuario*” Cuando se encuentra un marcador AR, entonces SOAR es notificado sobre su presencia a través de un sistema de retroalimentación en la interfaz. A continuación se pedirá la entrada del usuario a escoger entre las opciones disponibles. Cuando el usuario escoge una opción, SOAR llevará a cabo una acción de acuerdo con la interacción del usuario / intervención en el sistema.

El sistema cognitivo presentado en este artículo garantiza que las acciones propuestas darán lugar a conseguir la meta, por lo que el robot encontrará una solución basada en la retroalimentación desde el entorno ROS y en la intervención del usuario. Esta situación, sin embargo, no puede ser declarada como óptima. Por ejemplo, en algunas situaciones, el robot podría trasladarse a un lugar que no era camino correcto para el comando ejecutado; antes de pasar a una segunda acción necesita dar un paso en la dirección correcta. En todo caso, la realización de la tarea está garantizada ya que la arquitectura proporcionará continuamente pasos hasta que se logre el objetivo específico.

## 5. Conclusiones y futuras investigaciones

La comunicación bidireccional entre la arquitectura cognitiva SOAR y el entorno ROS se ha introducido y probado en un UAV (Unmanned Aerial Vehicle) Parrot AR.Drone 2.0 con objeto de resolver una tarea compleja expresada como la combinación de habilidades simples. La comunicación bidireccional es un tema clave para aumentar la información en la toma de mejores decisiones por parte de un razonador como SOAR. De esta forma, los planes incompletos pueden ser considerados, información del entorno se puede introducir en el razonador y el usuario puede interactuar en el proceso de planificación.

La comunicación bidireccional introducida ha permitido superar dos desventajas principales del anterior enfoque de comunicación SOAR-ROS. En primer lugar, la finalización de los planes incompletos de un objetivo especificado basándose en los datos en tiempo real disponibles en el entorno. En segundo lugar, la intervención del usuario que ayuda a las arquitecturas a comunicarse directamente con los usuarios y tomar sus sugerencias o entradas para utilizarlos en la toma de decisiones.

Como inconveniente, ya que se permiten planes parciales cuando se inicia la ejecución del plan, la arquitectura cognitiva no puede conocer si el objetivo solicitado al robot será completamente accesible o no. Sin embargo, ya que la acción del usuario y la retroalimentación del entorno son ahora accesibles, volver a planificar se hace más fácil, así como la intervención del usuario en base a la información del entorno ayuda en la consecución de la meta.

Se abren además diversas líneas de investigación futuras. Así, con el uso de esta comunicación bidireccional, sólo una arquitectura es necesaria sobre las habilidades simples de robots bien definidas y la retroalimentación de los sensores de a bordo. Por lo tanto, el robot puede comportarse de diferentes maneras ante diferentes ambientes, modificando los planes y ayudando a lograr el objetivo.

Por otra parte, la implementación actual se puede mejorar en términos de robustez mediante la resolución de un problema conocido. Principalmente, si una de las acciones no está completamente lograda (por ejemplo, el robot no es capaz de llegar a una posición en el espacio, porque está ocupada o el robot no puede encontrar un objeto que está delante de él), la activación de la habilidad fallará. En la implementación actual, así como en los anteriores, el robot no tiene medios para descubrir la razón del fallo. De ahí que el robot detecte que

el estado del mundo no ha cambiado y seleccionará de nuevo la misma acción (reintento) para el logro de la meta. Este comportamiento podría dar lugar a un bucle infinito de reintentos. Para la seguridad del robot UAV, cuando se produce esta situación el dron aterriza en el momento que reconoce la existencia de comandos repetidos.

## Acknowledgments

Este trabajo fue posible gracias al apoyo de la Universitat Politècnica de Catalunya (UPC) y la SASTRA University, con su programa de intercambio.

## Referencias

- [Amin and Govilkar, 2015] Dhiraj Amin and Sharvari Govilkar. Comparative study of Augmented Reality SDK's. *International Journal on Computational Sciences & Applications (IJCSA)*, 5, Feb 2015.
- [Anderson *et al.*, 2004] John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological review*, 111(4):1036–60, October 2004.
- [Anderson, 2007] John R. Anderson. How can the human mind occur in the physical universe? *New York: Oxford University Press*, 2007.
- [Besold *et al.*, 2014] T. R. Besold, A. D'Avila Garcez, K. U. Kühnberger, and T. C. Stewart (eds.). Neural-symbolic networks for cognitive capacities (Special issue). *Biologically Inspired Cognitive Architectures (BICA)*, 9, 1–122, 2014.
- [Canal *et al.*, 2015] Gerard Canal, Cecilio Angulo, and Sergio Escalera. Human multi-robot interaction based on gesture recognition. In *International Joint Conference on Neural Networks, IJCNN*, 2015. In Press.
- [Cho *et al.*, 1991] Bonghan Cho, Paul S. Rosenbloom, and Charles P. Dolan. Neuro-Soar: A neural-network architecture for goal-oriented behavior. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 673–677, 1991.
- [Kelley and Avery, 2010] Troy Dale Kelley and Eric Avery. A cognitive robotics system: the symbolic and subsymbolic robotic intelligence control system (SS-RICS). *SPIE Proceedings Vol. 7710: Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2010*, 25:460–470, April 2010.
- [Kelley, 2006] Troy Dale Kelley. Developing a Psychologically Inspired Cognitive Architecture for Robotic Control: The Symbolic and Subsymbolic Robotic Intelligence Control System. *International Journal of Advanced Robotic Systems*, 3(3):219–222, 2006.
- [Krajník *et al.*, 2011] Tomáš Krajník, Vojtěch Vonásek, Daniel Fišer, and Jan Faigl. AR-Drone as a Platform for Robotic Research and Education. *Springer Berlin Heidelberg: Communications in Computer and Information Science*, pages 172–186, June 2011.
- [Laird *et al.*, 2004] John E Laird, Keegan R Kinkade, Shiwali Mohan, and Joseph Z Xu. Cognitive Robotics using

- the Soar Cognitive Architecture. In *Proc. of the 6th Int. Conf. on Cognitive Modelling*, pages 226–230, 2004.
- [Laird, 2008] John E Laird. Extending the Soar Cognitive Architecture. In *Artificial General Intelligence Conference*, 2008.
- [Langleya *et al.*, 2009] Pat Langleya, John E. Lairdb, and Seth Rogersa. Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, June 2009.
- [Puigbo *et al.*, 2015] J.-Y. Puigbo, A. Pumarola, C. Angulo, and R. Tellez. Using a Cognitive Architecture for General-purpose Service Robot Control. In *Connection Science - The Society for the Study of Artificial Intelligence and the Simulation of Behaviour*, 2015.
- [Trafton *et al.*, 2005] J.G. Trafton, N.L. Cassimatis, M.D. Bugajska, D.P. Brock, F.E. Mintz, and A.C. Schultz. Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man and Cybernetics*, 25:460–470, July 2005.
- [Young and Lewis, 1997] Richard M Young and Richard L Lewis. The Soar Cognitive Architecture and Human Working Memory. In *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*, Nov 1997.