

Tutorial

Arquitectura de Subsunción para el Desarrollo de Controladores Basados en Comportamiento para Robótica e Inteligencia Artificial

Miguel Angel Astor R.

Universidad Central de Venezuela, Facultad de Ciencias
Escuela de Computación, Centro CICORE, Laboratorio ICARO
Urb. Valle Abajo, San Pedro, Caracas, Venezuela
Email: miguel.a.astor@ucv.ve

Resumen—En este tutorial se presentará la arquitectura de subsunción utilizada para definir controladores reactivos en los campos de la robótica y la inteligencia artificial. Esta arquitectura es un patrón de diseño arquitectónico que estructura el controlador de un agente inteligente como una serie de capas conectadas entre sí por líneas de control, de manera que cada capa represente el comportamiento del agente ante una situación concreta. El enfoque del tutorial es teórico-práctico, y durante el mismo los participantes podrán practicar la implementación de un controlador basado en la arquitectura de subsunción, utilizando el entorno de simulación provisto por el juego Robocode.

Palabras clave—Arquitectura de subsunción, controladores reactivos, robótica, inteligencia artificial, agentes inteligentes, Robocode.

I. INTRODUCCIÓN

La arquitectura de subsunción es una manera de estructurar controladores para robots y agentes inteligentes propuesta por Rodney Brooks en 1986 [1]. Como sugiere su nombre, la arquitectura de subsunción se define como un patrón de diseño arquitectónico para la estructuración e implementación de controladores reactivos para robots móviles, asociada a una filosofía de diseño enfocada en el principio de emerger comportamientos complejos mediante las interconexiones de comportamientos simples [2]. Se dice que un controlador es reactivo cuando este acopla fuertemente la percepción del agente con las acciones que este toma en base a dichas percepciones, en contraposición con el uso de representaciones abstractas del mundo que permitan al agente planificar o deliberar antes de actuar [2].

A pesar de que la arquitectura de subsunción fue propuesta originalmente para el desarrollo de controladores reactivos para robots, esta también puede ser utilizada para la implementación de controladores para agentes inteligentes en otros dominios, como por ejemplo la creación de jugadores automáticos para distintos tipos de juegos [3], [4] o la generación de patrones de ritmo para música sintética [5].

Así mismo, esta arquitectura ha servido de base para el desarrollo de arquitecturas reactivas más complejas, como la arquitectura dinámica de subsunción [3], e incluso para la definición de arquitecturas de control híbridas que combinan

comportamiento reactivo para la ejecución de acciones a corto plazo (en particular a lo referente a auto-preservación y evasión de obstáculos [6]), junto a módulos deliberativos para planificación de acciones y cumplimiento de metas a largo plazo [7].

El objetivo de este tutorial es presentar la arquitectura de subsunción como una manera de estructurar controladores para agentes inteligentes móviles, haciendo especial énfasis en sus usos en la robótica y la Inteligencia Artificial en tiempo real, practicando la implementación de un agente inteligente basado en subsunción en el lenguaje Java, mediante el uso del juego didáctico Robocode, tomando como base la implementación provista por el sistema operativo LeJOS (LEGO Java Operating System) para los robots móviles LEGO Mindstorms NXT™.

II. OBJETIVOS

A continuación se presentan los objetivos general y específicos del Tutorial planteado.

II-A. *Objetivo general*

Conocer, comprender y practicar el uso de la arquitectura de subsunción como patrón de diseño para Inteligencia Artificial reactiva.

II-B. *Objetivos específicos*

- Presentar el paradigma de Inteligencia Artificial reactiva.
- Presentar la arquitectura de subsunción como patrón de diseño para Inteligencia Artificial.
- Practicar la implementación de un controlador reactivo basado en la arquitectura de subsunción en el lenguaje Java.

III. CONTENIDOS

Durante el Taller se desarrollará el siguiente contenido temático:

III-A. La Arquitectura de Subsunción

Inteligencia Artificial deliberativa y reactiva. La arquitectura de subsunción. Autómatas finitos aumentados. Inhibición y supresión de señales de control. Ejemplo de un controlador de subsunción para búsqueda y exploración.

III-B. Implementación de la Arquitectura de Subsunción

Introducción a Robocode. Diseño e implementación del árbitro de subsunción. Diseño e implementación de los comportamientos. Integración del árbitro y los comportamientos en un controlador reactivo.

III-C. Arquitecturas derivadas de la Arquitectura de Subsunción

Arquitectura dinámica de subsunción. Arquitectura híbrida de tres (3) capas.

IV. METODOLOGÍA

La metodología del taller es teórico-práctica, basada en una presentación de conocimiento teórico para el desarrollo del Tema III-A, seguida por una serie de ejercicios prácticos de programación para el desarrollo del Tema III-B, culminando con una breve presentación teórica para el desarrollo del Tema III-C.

Los ejercicios prácticos de programación serán realizados en el lenguaje de programación Java utilizando la herramienta y juego didáctico de simulación Robocode. Este juego permite a los usuarios el ejecutar batallas simuladas entre robots programados en el lenguaje Java y sirve como herramienta de apoyo para la práctica de programación en Inteligencia Artificial [8].

V. DURACIÓN

Se estima una duración de 4 horas para el Taller (medio día).

VI. PÚBLICO OBJETIVO

Estudiantes, profesionales y personas interesadas en la implementación de controladores para Inteligencia Artificial de tiempo real. Los participantes deben tener experiencia básica a intermedia con el lenguaje de programación Java.

VII. REQUERIMIENTOS

Para poder llevar a cabo el tutorial se requiere lo siguiente:

- Un proyector.
- Acceso a Internet en la sala.
- Computadoras donde puedan trabajar hasta dos personas por computadora, las cuales deben poseer lo siguiente:
 - Sistema operativo basado en Windows o (preferiblemente) Linux.
 - Compilador para Java 6 o superior.
 - El software Robocode (paquetes robocode y robocode-doc en sistemas basados en Debian).

VIII. IDIOMA

La exposición, dictado, ejercicios y ejemplos serán dados en español.

REFERENCIAS

- [1] R. Brooks, "A robust layered control system for a mobile robot," *IEEE journal on robotics and automation*, vol. 2, no. 1, pp. 14–23, 1986.
- [2] R. C. Arkin, *Behavior-based robotics*. MIT press, 1998.
- [3] H. Nakashima y I. Noda, "Dynamic subsumption architecture for programming intelligent agents," in *Multi Agent Systems, 1998. Proceedings. International Conference on*. IEEE, 1998, pp. 190–197.
- [4] A. Krzywinski, W. Chen, y A. Helgesen, "Agent architecture in social games-the implementation of subsumption architecture in diplomacy," in *AIIDE*, 2008, pp. 100–104.
- [5] A. Levisohn y P. Pasquier, "Beatbender: Subsumption architecture for autonomous rhythm generation," in *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*. ACM, 2008, pp. 51–58.
- [6] T. Gomi y P. Volpe, "Collision avoidance using behavioral-based ai techniques," in *Intelligent Vehicles Symposium (1993: Tokyo, Japan). Proceedings of the Intelligent Vehicles' 93 Symposium*, 1993.
- [7] S. J. Russell y P. Norvig, *Artificial intelligence: a modern approach (3rd edition)*. Prentice Hall, 2009.
- [8] K. Hartness, "Robocode: using games to teach artificial intelligence," *Journal of Computing Sciences in Colleges*, vol. 19, no. 4, pp. 287–291, 2004.

PRESENTADOR

Miguel Ángel Astor Romero

Profesor investigador, categoría Instructor, de la Escuela de Computación, de la Universidad Central de Venezuela (UCV). Es Licenciado en Computación de la Universidad Central de Venezuela (2014). Actualmente cursa estudios de Maestría en Ciencias de la Computación en la UCV bajo la tutoría del prof. Wílmer Pereira Gonzales.

Dirección: Paseo los Ilustres, Urb. Valle Abajo. Facultad de Ciencias, Cdad. Universitaria de Caracas, Venezuela.

Tlf: +58-212-6050984. Correo: miguel.a.astor@ucv.ve