

Análisis Lenguajes de Especificación de Agente en Robótica Móvil

Sebastián Gottifredi Alejandro J. García

Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET),

Laboratorio de Investigación y Desarrollo de Inteligencia Artificial,

Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur,

Avenida Alem 1253,(B8000BCP), Bahía Blanca, Argentina

Tel: (0291) 459-5135 / Fax: (0291) 459-5136

Email: {sg,ajg}@cs.uns.edu.ar

Resumen

Esta línea de investigación involucra la implementación de sistemas multi-agente en robótica mediante el uso de lenguajes de especificación de agentes. En particular, en este trabajo se hará un análisis del desarrollo de agentes que controlan robots Khepera2 utilizando el lenguaje de especificación 3APL. Uno de los objetivos de esta línea de investigación es analizar las ventajas y desventajas del uso de un lenguajes de especificación de agentes en robótica móvil.

Palabras claves: Sistemas Multi-Agentes, Lenguaje de Especificación de Agentes, Robótica Móvil.

1. Introducción

En este trabajo se hará un análisis de sistemas mutli-agente (SMA) en robótica, implementados mediante el uso de lenguajes de especificación formal de agentes. A lo largo del trabajo se mostraran los beneficios del uso de lenguajes de especificación en robots móviles, a través de ejemplos. Por último, se realizará una evaluación de una implementación de agentes para manejar robots Khepera2 utilizando el lenguaje de 3APL.

Esta línea de investigación está orientada a sistemas multi-agente colaborativos, con agentes cognitivos que pueden llegar a poseer capacidades deliberativas. Por mas complejo que sea el agente la percepción sobre el entorno será parcial y al actuar la influencia sobre el entorno también será parcial. Por ende, una acción ejecutada por el agente dos veces puede tener efectos diferentes, el agente debe estar preparado para que la acción falle, o que los efectos no sean los previstos. En especial en un ambiente del mundo real utilizando robots móviles, estas acciones pueden tener muchas consecuencias de diferente índole y con diferentes efectos, dándole un grado de incertidumbre adicional al problema de desarrollar el agente que controla al robot. En este trabajo, trataremos las incertidumbres a nivel cognitivo respecto al agente. Por ejemplo, el error producido por el rozamiento de la ruedas en las diferentes superficies no será corregido a bajo nivel, sino que el agente conocerá esta situación y lo tomará en cuenta para decisiones futuras.

Financiado parcialmente por CONICET (PIP 5050), Universidad Nacional del Sur (PGI 24/ZN11) y Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096).

En general en la literatura se menciona que para conseguir buena performance en un agente, como un robot móvil, la habilidad del agente para reaccionar adecuadamente al ambiente externo debe ser el aspecto fundamental a ser considerado. Actualmente se reconoce la importancia de la reactividad, pero este aspecto sólo no es suficiente para crear agentes capaces de resolver tareas complicadas, para ello se necesita representar pensamiento de alto-nivel.

Los Sistemas Multi-Agente (SMA) son utilizados para resolver las problemáticas planteadas en robots físicos autónomos móviles en tareas tanto cooperativas como competitivas, como por ejemplo Robocup [2]. Sin embargo, en gran parte de las aplicaciones de SMA, los agentes que controlan a los robots son implementados en forma Ad-hoc utilizando lenguajes de propósito general más algún *plug in* para la comunicación o interacción. Esto hace muy heterogénea la gama de estructuras internas de los agentes, lo que produce que los agentes, y por ende los SMA, sean mucho más difíciles de diseñar, implementar, verificar. Una forma de aliviar este problema es utilizar reglas formales y estándares para la construcción de los agentes. Esto se puede conseguir si la implementación de los agentes del SMA se basa en reglas de sintaxis, semántica y pragmática de un lenguaje de especificación formal para el desarrollo de agentes. Las reglas de este lenguaje formal determinan la forma de las estructuras deliberativas generales del agente. La interacción y comunicación entre agentes del SMA es provista por reglas del lenguaje formal. Esto permite al desarrollador diseñar al agente a nivel de sus estados mentales y capacidades efecto-sensoriales. En [6] se enuncian las principales características de un lenguaje de especificación de agentes y las ventajas de desarrollar agentes de esta manera. A lo largo de la sección 2 se mostrarán las razones de por qué usar un lenguaje de especificación en agentes que controlan robots, y que elementos son necesarios para poder implementar estos agentes utilizando dicho lenguaje.

Existen varias plataformas de implementación de agentes que utilizan como base lenguajes de especificación formal de agentes, en este trabajo se tendrán en cuenta sólo este tipo de lenguajes de implementación. Un Ejemplo de ello es 3APL [1]. En la sección 3 se mostrará una implementación de los robots Khepera2 [7] utilizando este lenguaje.

2. Lenguajes de Especificación en Robótica Móvil

En esta sección se mostrarán las razones de por qué utilizar un lenguaje de especificar agentes que controlan robots móviles y cuáles son sus beneficios. A lo largo de la sección se utilizará como dominio de aplicación el *cleaning task problem* utilizado en [4], donde un conjunto de robots móviles, en una habitación con cajas y una zona almacén, tienen como objetivo llevar y depositar las cajas en el almacén.

El mundo de este dominio consiste de:

- Una área de trabajo cuadrada de 100 unidades por lado, dividida en células cuadradas de 10 unidades por lado.
- Robots encargados de llevar las cajas.
- Cajas de diferentes tamaños distribuidas a lo largo de la habitación.
- Una zona de faja que representa el Almacén.

El sistema es abierto, y cada robot posee las siguientes características:

- una visión global del cuarto.
- puede moverse, recoger o dejar cajas

- comunicarse con los otros robots, a diferencia de lo planteado en [4]
- Se pueden producir errores en las acciones del robot y Por ende, una acción ejecutada por el este dos veces puede tener efectos diferentes

Ejemplo 1: Considere 2 robots llamados Robot1 y Robot2, una zona de almacén de 3x3 células en la parte superior derecha de la habitación, y 2 cajas, una pequeña y una mediana llamadas caja1 y caja2 respectivamente (ver Figura 1).

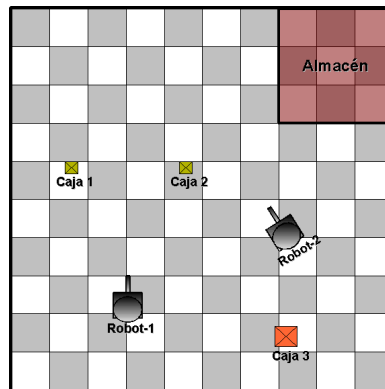


Figura 1: Ejemplo del dominio

En la literatura, se plantea que es adecuado desarrollar los robots móviles inteligentes a través de varios niveles de abstracción, que van desde el mas bajo nivel como los efectores físicos del robot, hasta el mas alto nivel como las capacidades cognitivas del agente. Ejemplos de estas arquitecturas son presentados en [3] y [5]. En la actualidad existen lenguajes como 3APL que siguen la estructura de niveles, gracias un soporte para robótica descrito en [8]. Este trabajo sólo considerarán los niveles mas altos de estas arquitecturas ya que los lenguajes de especificación se encuentran en el nivel cognitivo del agente. Cabe resaltar, que de no existir los niveles inferiores el lenguaje de especificación no se podría utilizar.

Las principales características con las que debe contar un lenguaje, según [6], de especificación de agentes son: arquitecturas mentales flexibles, reglas para modelar la relación comunicación-estados mentales, soporte para planning, manejar la base de conocimiento del agente y el razonamiento.

Como se menciona en [6], el lenguaje de especificación permite desarrollar los agentes a nivel de sus estados mentales, En robots móviles esto es importante, en especial para manejar la incertidumbre respecto del accionar del robot, ya que a este nivel es mas fácil tomar decisiones respecto al que acciones tomar. Siguiendo el Ejemplo 1, el robot1 y usando como base a el lenguaje de especificación de 3APL, estaría especificado del siguiente modo: como meta deberá ordenar la habitación, sus creencias provendrán de la percepción que recibe de la habitación y los mensaje enviados por los otros agentes, sus capacidades serán enviar mensajes, moverse y levantar las cajas, y por último tendrá reglas que le permitirán tomar decisiones respecto a que acciones tomar según la heurística elegida(cajas pequeñas primero, etc). A este nivel de abstracción es mas fácil razonar acerca de la incertidumbre de las posibles acciones que puede tomar el robot.

Una de las ventajas de usar un lenguaje de Especificación para desarrollar los agentes según [6], es que proveen una forma de comunicación entre agentes que está completamente integrada a las capacidades cognitivas de estos, tratándola como un componente mas en el razonamiento. En robots móviles, especialmente en sistemas abiertos, la comunicación para las tareas cooperativas entre agentes es importante debido a las necesidades de coordinación, optimización, etc. Si se observa la Figura 1 podemos ver que tanto el robot1 como el robot2 tienen cerca la caja1, que es la mas pequeña, si ambos están interesado en llevar las cajas mas pequeñas primero, se producirá una colisión. En

el caso de un SMA abierto, donde agentes con diferentes estrategias cooperan, el Robot1 debería avisarle al Robot2 que el se dirige en busca de esa caja, y esperar la confirmación, antes de dirigirse a buscar la caja. De no contar con comunicación, los Robots deberían predecir el comportamiento de sus compañeros.

3. Implementación de robots Khepera2 con 3APL

En esta sección se mostrará una implementación de agentes que controlan robots Khepera2 utilizando el lenguaje de especificación 3APL [1]. Para realizar esta implementación se utilizará la arquitectura por capas propuesta en [3], con la diferencia que la capa cognitiva estará representada por el lenguaje de especificación de 3APL, esta arquitectura permite a la capa cognitiva abstraerse de posiciones y detalles de bajo nivel. También se utilizaran la versión para robots móviles de 3APL planteada en [8], donde las creencias de los robots son actualizadas en base al entorno.

A continuación en la Figura 2 se muestra el código 3APL del Robot1 del Ejemplo 1. El comportamiento del agente esta dado por el siguiente orden de preferencia:

1. cajas cercanas a el.
2. cajas que cercanas al almacén.
3. cajas chicas.

En el ejemplo todos los elementos de la percepción del mundo en el BELIEFBASE son generados o actualizados por las capas de nivel inferior a la capa cognitiva en cada ciclo de deliberativo de 3APL.

```
Robot1
%-----
CAPABILITIES:
{} Dormir() {sleep(20)}
{} IrA(Lugar) {llamada a la capa de Sensorial/Efectorial}
{} Recoger(Objeto) {llamada a la capa de Sensorial/Efectorial, encima(Objeto)}
{} Soltar(Objeto) {llamada a la capa de Sensorial/Efectorial, NOT encima(Objeto)}
{} ElegirCaja(Caja) {mejorCaja(Caja)}
%-----
BELIEFBASE:
chica(cajal),media(caja3),otros([robot2]),yo(robot1)cerca(cajal, robot1),
cerca(cajal, robot2), cerca(caja2, robot2),cerca(cajal, almacen),CajasSueltas([cajal, caja2]),
CajasGuardadas([])
%-----
GOALBASE: Ordenar()
%-----
RULEBASE:
Ordenar <- CajasSueltas([]) |
    {Dormir(), Ordenar()},
Ordenar <- CajasSueltas([H|T]) |
    {MejorCajaPosible([H|T]), BuscarCaja(), Ordenar()},

MejorCajaPosible(L) <- (L = [H|T]) AND yo(R) AND cerca(H,R) AND NOT received(X,inform,H) |
    ElegirCaja(H),
MejorCajaPosible(L) <- (L = [H|T]) AND cerca(H,almacen) AND NOT received(X,inform,H) |
    ElegirCaja(H),
MejorCajaPosible(L) <- (L = [H|T]) AND chica(H) AND NOT received(X,inform,H) |
    ElegirCaja(H),
MejorCajaPosible(L) <- (L = [H|T]) |
    MejorCajaPosible(T),
MejorCajaPosible(L) <- (L = []) |
    SKIP

BuscarCaja() <- mejorCaja(B) AND yo(R) AND al_lado(B,R) |
    {Recoger(B), GuardarCaja()},
BuscarCaja() <- mejorCaja(B) |
    {Avisar(B), IrA(B), BuscarCaja() }
```

```

GuardarCaja() <- mejorCaja(B) AND encima(B) AND yo(R) AND al_lado(almacen,R) |
  Soltar(B),
GuardarCaja() <- mejorCaja(B) AND encima(B) |
  {IrA(Almacen), GuardarCaja()}

Avisar(Caja) <- TRUE |
  send(SistemaManejoAgentes,inform,Caja)

```

Figura 2: Implementación del Robot1 en 3APL

El código para el agente que maneja al Robot2 es similar, sólo cambian los hechos iniciales `yo(robot1)` por `yo(robot2)`, y `otros([robot2])` por `otros([robot1])`

4. Conclusión

En este trabajo se realizó una exposición de las razones del uso de Lenguajes de Especificación de agente en el desarrollo de agentes para Robots Móviles en un SMA y se mostraron sus beneficios a través de ejemplos, como por ejemplo, el uso de las arquitecturas de agente provistas por el Lenguaje y la integración del sistema de comunicación con dichas arquitecturas. Por último se realizó la implementación de agentes para manejar robots Khepera2 utilizando el lenguaje de 3APL, evaluar los beneficios del uso de lenguaje de especificación en robótica.

Del análisis de la implementación de Robots Khepera utilizando 3APL, se puede concluir que los lenguajes de especificación brindan un buen marco para el desarrollo de las capacidades mentales de los robots, lo que permite, además de todas las ventajas que esto trae respecto al desarrollo de agentes, corregir las posibles incertezas del mundo real a un alto nivel. Sin embargo, para que el lenguaje de especificación brinde todas sus ventajas debe estar implementado sobre un robusto sistema de niveles de abstracción, lo cual no siempre es simple.

Referencias

- [1] M. Dastani, B. van Riemsdijk, F. Dignum, and J. Meyer. A programming language for cognitive agents: Goal-directed 3apl, 2003.
- [2] The Robocup Federation. Robocup. <http://www.robocup.org>.
- [3] Edgardo Ferretti, Marcelo Errecalde, Alejandro García, and Guillermo Simari. Khedelp: A framework to support defeasible logic programming for the khepera robots. In *ISRA06*, 2006.
- [4] Edgardo Ferretti, Marcelo Errecalde, Alejandro García, and Guillermo Simari. An application of defeasible logic programming to decision making in a robotic environment. In *LPNMR*, 2007.
- [5] A. J. García, G. I. Simari, and T. Delladio. Designing an agent system for controlling a robotic soccer team. In *X Congreso Argentino de las Ciencias de la Computación*, 2004.
- [6] Sebastian Gottifredi and Alejandro García. Análisis de lenguajes de implementación de agente. In *Workshop de Investigadores en Ciencias de la Computación (WICC 2006)*, 2006.
- [7] K-Team. Khepera 2. <http://www.k-team.com>, 2002.
- [8] Marko Verbeek. 3apl as programming language for cognitive robots. Master's thesis, University of Utrecht, 2003.