

Actividad de agentes robóticos regulada a través de información de temporalidad vía un cálculo lógico de eventos

Manuel Hernández Gutiérrez, Eduardo Sánchez Soto

Universidad Tecnológica de la Mixteca,
Instituto de Computación,
México

{manuelhg, esanchez}@mixteco.utm.mx

Resumen. Conforme las tecnologías del área de la Inteligencia Artificial se desarrollan, será en ocasiones de suma importancia para algunas aplicaciones el agregar a tales tecnologías mecanismos computacionales para tratar el paso del tiempo, según una noción humana de éste. En este artículo aplicamos un cálculo lógico de eventos a agentes robóticos para que tales agentes posean mecanismos computacionales básicos de tratamiento de tiempo. Estudiaremos el caso de una arquitectura multi-agente que depende de medios de comunicación remotos y de ciclos de agentes que se desarrollan en un ambiente cambiante. En esta arquitectura, cada agente pasa de observaciones a razonamientos y, de ahí, a acciones, con una noción adjunta del paso del tiempo.

Palabras clave: Temporalidad, lógica, agentes robóticos, robótica cognitiva.

Activity of Robotic Agents Ruled through Time Information via a Logic-Based Event Calculus

Abstract. With the advancement of Artificial Intelligence, there may be situations where it becomes necessary to incorporate time-processing mechanisms based on human perception into certain applications. Our article proposes a logical computation method for events in robotic agents, enabling them to possess these basic computational time processing mechanisms. We will focus on a multi-agent architecture that relies on remote communication media and agent cycles in a dynamic environment. Each agent in this architecture progresses from observations to reasoning and, ultimately, to actions, while taking into account the passage of time.

Keywords: Time, logic, robotic agents, cognitive robotics.

1. Introducción

La inteligencia humana está enmarcada dentro de una noción de tiempo. Sin embargo, para los humanos el tiempo es un concepto lineal, en tanto que para las inteligencias artificiales el tiempo tendría que ser considerado en su generalidad como

no-lineal [16, 14, 3]. Por ello, es indispensable considerar qué manejo de tiempo se asocia a una inteligencia artificial, sobre todo al permitir que inteligencias de este tipo tomen decisiones o lleven a cabo acciones. En este artículo se trata el tema de la deducción de estados temporales de agentes robóticos que realizan cierto conjunto de actividades dentro de un escenario físico.

Trataremos como caso de estudio a agentes robóticos móviles. Las deducciones así obtenidas son utilizadas para compartir información entre varios agentes y para tomar decisiones, ya sea grupales o individuales. Cada agente se interrelacionará con el mundo a través de un ciclo abductivo kowalskiano, en donde se realizan observaciones, se razona, y posteriormente se actúa, con posibilidades de actividades (mediante reglas de producción) emergentes o alternativas entre cada etapa [6].

También, aquí se muestra que las deducciones en relación a la espacio-temporalidad, un tema de conocimiento existencial fundamental, puede abordarse mediante un cálculo de eventos, así brindando algunos mecanismos computacionales extras a las reglas de producción para que, por un lado, los agentes 'conozcan' mejor su entorno y actúen en consecuencia, y por otro lado, su actuar sea enmarcado en una noción de tiempo apegada a la causalidad de la realidad y también sea asequible a la comprensión humana.

Panorama de este escrito. En la Sección 2 se tratará el tema de la programación lógica aplicada al cálculo de eventos, y de la descripción de este cálculo de eventos para conocer, con base en eventos y flujos, los estados del mundo. En la Sección 3 se describen algunos aspectos tecnológicos que subyacen a los agentes robóticos a los que se puede aplicar nuestros resultados. En la Sección 4 se da un ejemplo amplio de cómo aplicar el cálculo de eventos descrito a agentes que, como supuesto, sigan un ciclo kowalskiano abductivo. Finalmente, en la Sección 5 se brindan algunas conclusiones y trabajo a futuro.

2. Lógica computacional

Aquí utilizamos una lógica computacional que considera la ocurrencia de eventos y la forma en que el mundo cambia, en su modalidad de programación lógica. Tal lógica es aplicada a agentes robóticos para obtener deducciones acerca del espacio y el tiempo. En la programación lógica se mantiene un equilibrio delicado entre la eficiencia en la ejecución de sus programas, mediante resolución SLDNF, y las lecturas declarativas de éstos, mediante modelos de Herbrand.

La programación lógica tiene como notable ventaja el proveer modelos sintácticos de relativa naturaleza simple, logrando así que los programadores mantengan certeza de cuál es el objetivo general a lograr cuando se implementa un sistema.

En este punto, enfatizamos que no solo es necesario considerar que los modelos propuestos sean asequibles, dados los supuestos axiomáticos formales, y aquellos supuestos obtenidos de la realidad" (domain theory), sino que también es fundamental mantener a la vista un enfoque computacional, que además sea de una eficiencia en tiempo real, un tema, éste último, de naturaleza fundamental en el caso de la posible aplicación de la lógica computacional a agentes robóticos.

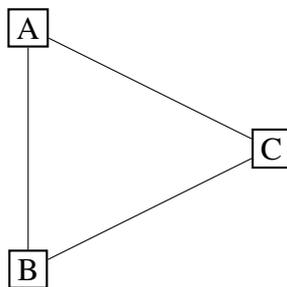


Fig. 1. Triángulo de Shanahan.

A continuación vemos un tema que tiene como origen el punto de vista de la programación lógica y se ha abierto paso en diversas aplicaciones que requieren el tratamiento del tiempo, a saber, el cálculo de eventos.

Cálculo de eventos. El cálculo de eventos basado en programación lógica fue ideado por Kowalski y Sergot hace ya algún tiempo [9], para incluir en el formalismo de la programación lógica la noción de predicados que cambian de valores de verdad conforme transcurre el tiempo.

Su relevancia en el tratamiento de temporalidad (dinámica) se muestra en publicaciones que aplican éste tipo de cálculo (o variantes y adaptaciones) a problemas que van desde temas de vigilancia por videos y sensores [12] hasta temas de salud pública [1]. Una excelente introducción al tema la da [15], en donde se aplica el cálculo de eventos a los temas de la inteligencia artificial, el no-determinismo, y a la abducción lógica, entre otros.

A continuación explicaremos los elementos esenciales requeridos del cálculo de eventos en la aplicación que tenemos destinada. De acuerdo con Shanahan [15] y basándose en la Fig 1, en donde de cada dos vértices conocidos se intenta estudiar un tercer vértice desconocido, se tiene:

A: Qué ha sucedido.

B: Qué efectos tienen algunos eventos o acciones sobre algunos flujos.

C: Qué es verdadero en un momento dado (qué flujos tienen el valor de verdad **Verdadero** en ese momento).

También de acuerdo con Shanahan, en la Fig. 1, dados dos vértices conocidos del triángulo, el tercero es posible de analizar con modalidades inductivas (B, si se conocen A y C), deductivas (C, si se conocen A y B) o abductivas (A, si se conocen B y C). Aquí nos enfocaremos en resultados deductivos (A y B conocidos, C desconocido): Hay una narración y se conoce qué hacen las acciones y cuáles son los flujos, de donde se tiene que deducir qué es verdadero en cierto momento.

Algunas restricciones de integridad (integrity constraints) obtenidas de la abstracción que se obtiene de la realidad forman parte de la información que debe tomarse en consideración al momento de que un agente interactúe con el mundo. Estas restricciones complementan posible información necesaria para justificar la inducción, deducción o abducción realizada, según sea.

Las personas interesadas aún más en el tema son remitidas a [15] dónde se explican algunas suposiciones extra que hacen formalmente rigurosa la aplicación del cálculo de eventos (leyes inerciales, de unicidad de nombramiento, suposición del "mundo cerrado", teoría de la igualdad, entre otras suposiciones rigurosas y fundamentales).

2.1. Una ontología para cálculo de eventos

Para tratar el tema de un agente existiendo en un mundo, es necesario considerar varios conceptos e interrelaciones de un cálculo de eventos, es decir, requerimos de una ontología de cálculo de eventos¹. Esta ontología consiste de identificar, en la medida de lo posible y en pro de tener un modelo adecuado a la ocurrencia de eventos en el tiempo y a la forma en que estas ocurrencias impactan en la veracidad del mundo en un instante dado, los siguientes conjuntos:

- Un conjunto finito de eventos, que se suponen instantáneos al ocurrir, completamente identificables, que afectan los valores de verdad de predicados llamados fuentes, descritos en el siguiente ítem.
- Un conjunto finito de fuentes, que son predicados con valor cambiante dependiendo de cómo están relacionados con la ocurrencia de eventos, y tales que está bien determinado su valor de verdad al momento de inspeccionarse, y en efecto, cada fuente debe estar relacionado con al menos un evento, sea para que el fuente comience a ser verdadero por la ocurrencia del evento o que el fuente comience a ser falso por tal ocurrencia.
- Un conjunto de valores de tiempo (instantes o momentos), los cuales deben ser comparables, discretizables (hasta un grado necesario), y de tal forma que todo evento tenga un momento de ocurrencia, y que todo fuente tenga un valor definido de veracidad dado un momento en el que el fuente se inspeccione. Hablamos de intervalos si tenemos dos momentos t_1 y t_2 , con $t_1 < t_2$ tal que el conjunto, intervalo entre t_1 y t_2 , $t \mid t > t_1$ y $t < t_2$ sea no vacío.
- Un conjunto de restricciones de integridad, como aquellas suposiciones, leyes, o condiciones de la realidad que son explícitamente establecidas en formulaciones lógicas, y que los eventos junto con los fuentes deben cumplir en cada ocasión (estipulando que, de otra manera, se incurriría en inconsistencias o violaciones de causalidad, por ejemplo).

La axiomática siguiente es identificada como "elemental" para el caso del tratamiento de un cálculo de eventos:

1. **Initiates**(α, β, τ): El fuente β comienza a ser verdadero cuando acontece el evento α en el instante τ .
2. **Terminates**(α, β, τ): El fuente β cesa de ser verdadero cuando acontece el evento α en el instante τ .

¹ Nota: La palabra ontología tiene en inteligencia artificial otra acepción a la aquí utilizada, de términos relacionándose con otros términos; no es la aquí aplicada.

$$\begin{aligned}
 \mathbf{HoldsAt}(f, t) &\leftarrow \mathbf{InitiallyP}(f) \wedge \neg \mathbf{Clipped}(0, f, t) & (1) \\
 \mathbf{HoldsAt}(f, t_3) &\leftarrow \mathbf{Happens}(a, t_1, t_2) \wedge \mathbf{Initiates}(a, f, t_1) \wedge & (2) \\
 & t_2 < t_3 \wedge \neg \mathbf{Clipped}(t_1, f, t_3) \\
 \mathbf{Clipped}(t_1, f, t_4) &\leftrightarrow \exists a, t_2, t_3 [\mathbf{Happens}(a, t_2, t_3) \wedge t_1 < t_3 \wedge t_2 < t_4 \wedge & (3) \\
 & [\mathbf{Terminates}(a, f, t_2) \vee \mathbf{Releases}(a, f, t_2)]] \\
 \neg \mathbf{HoldsAt}(f, t) &\leftarrow \mathbf{InitiallyN}(f) \wedge \neg \mathbf{Declipped}(0, f, t) & (4) \\
 \neg \mathbf{HoldsAt}(f, t_3) &\leftarrow \mathbf{Happens}(a, t_1, t_2) \wedge \mathbf{Terminates}(a, f, t_1) \wedge & (5) \\
 & t_2 < t_3 \wedge \neg \mathbf{Declipped}(t_1, f, t_3) \\
 \mathbf{Declipped}(t_1, f, t_4) &\leftrightarrow \exists a, t_2, t_3 [\mathbf{Happens}(a, t_2, t_3) \wedge t_1 < t_3 \wedge t_2 < t_4 \wedge & (6) \\
 & [\mathbf{Initiates}(a, f, t_2) \vee \mathbf{Releases}(a, f, t_2)]] \\
 \mathbf{Happens}(a, t_1, t_2) &\rightarrow t_1 \leq t_2 & (7)
 \end{aligned}$$

Fig. 2. Cálculo de eventos.

3. **InitiallyP**(β): El fuente β es verdadero desde el instante inicial (que uno puede convenir que sea 0 o 1).
4. **Happens**(α, τ): El evento α ocurre en el instante τ .
5. **HoldsAt**(β, τ): El fuente β es verdadero en el instante τ .
6. **Clipped**(τ_1, β, τ_2): El fuente β es terminado en el intervalo $(\tau_1, \tau_2]$.
7. **Releases**(α, β, τ_2): El fuente β no es ya más “inercial” después del evento α ocurrido en el instante τ : ya no es influido más en su valor de verdad por la ocurrencia del evento α .
8. **InitiallyN**(β): El fuente β no es verdadero desde el instante inicial (0 o 1, a convenir).
9. **Declipped**(τ_1, β, τ_2): El fuente β comienza a ser verdadero en algún instante del intervalo $(\tau_1, \tau_2]$.

Formalmente, colocamos esta axiomática en forma clausal en la Fig. 2. Se tratan a continuación cómo se manejarán algunos temas relacionados con el cálculo de eventos en este escrito, particularmente los de granularidad del tiempo, los intervalos y las narrativas.

2.2. Granularidad, intervalos y tipos de narrativas

El paso del tiempo es generalmente supuesto continuo; no obstante, para propósitos computacionales es mejor suponerlo discretizado, de donde surge el problema de qué tan fina o gruesa se necesita una subdivisión para modelar algunos fenómenos. Esto es conocido como el problema de la granularidad temporal [2]. La granularidad aquí propuesta hace que los eventos sean instantáneos, y que los flujos sean verdaderos o falsos por intervalos (posiblemente semi-abiertos).

Aún más, también impondremos una granularidad espacial, para que la movilidad de los agentes robóticos sea únicamente dada por coordenadas bidimensionales enteras, pero en una región acotada. Pasamos ahora a considerar intervalos de tiempo. En el enfoque de los relojes lógicos ([10]) son los mismos eventos los “motores” del paso del tiempo; aquí, en su lugar, el tiempo es medido por un reloj global único.

Todos los intervalos serán supuestos sobre el tiempo discretizado, y serán en efecto abiertos por la izquierda y cerrados por la derecha. Además, y como un eco de las marcas de tiempo (timestamps) de Lamport en [10], así como de un procedimiento lógico para deshacerse de los cuantificadores existenciales mediante la técnica de skolemización, utilizaremos una etiquetación consistente con el paso del tiempo, cuando sea necesario, para relacionar a los eventos con el instante de tiempo de su ocurrencia.

Pasamos ahora a definir algunos tipos de narrativas. Una narración consiste en enfocarse cuándo acontecen algunos eventos según una cronología y dentro de un escenario físico. Una narrativa es la identificación temporal de un conjunto de ocurrencias de eventos junto con la descripción de qué se ha afectado al ocurrir tales eventos. Se identificará como lo afectado a un conjunto de fuentes. Para un predicado de este tipo, de aridad n , se agregará el parámetro de tiempo al final de todos los argumentos, así haciendo un fuente de aridad $n + 1$.

Ahora bien, los eventos pueden ser descritos globalmente, por un superagente omnisciente que “sabe” qué ocurre en todo momento y para todo agente dentro de un mundo. Esto conlleva una narrativa global; este tipo de narrativas es adecuada para análisis teóricos de sistemas de agentes, sobre todo en ambientes artificiales que tengan una cantidad pequeña de eventos y de los fuentes afectados por el paso del tiempo. Otras dos opciones son: o narrativas grupales o narrativas individuales.

En una narrativa grupal el enfoque se da sobre los eventos que sean testimoniados (o realizados) por un subconjunto de agentes durante momentos o intervalos de tiempo bien definidos. Notar que una narrativa grupal es una proyección de una narrativa global.

La ventaja de las narrativas grupales es que son una más precisa descripción de los momentos de ocurrencia de los eventos o bien un estrechamiento de los intervalos narrados globalmente. Aquí se destaca el tema de la consistencia (ausencia de contradicciones) de las narraciones: ningún fuente de la narrativa global debe ser contradicho por una narrativa grupal.

En una narrativa individual se toman a agentes específicos para describir qué eventos y fuentes ocurren y se modifican, respectivamente, pero enfocándose en tales agentes. Cuando las narrativas son dadas por separado, se busca, en efecto, hallar consistencia entre las narrativas globales, grupales e individuales, ya que ello fortalece la consistencia de la descripción total del mundo.

2.3. Ciclos abductivos de agentes: Ciclo de agente Kowalskiano

En un ciclo abductivo kowalskiano [6] se tienen que cumplir ciclos del tipo: observar-razonar-actuar, con algunas partes esporádicas basadas en reglas de producción. Planeamos tratar tal enfoque que incluya reglas de producción en un trabajo a futuro. Por el momento, la parte de observar se traduce en un agente robótico como la capacidad de recabar información sensorial.

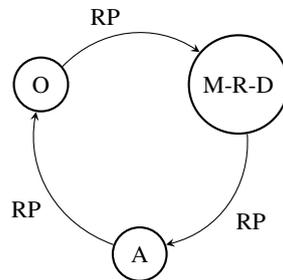


Fig. 3. Ciclo de agente kowalskiano: O: observaciones, M: metas, R: Razonamiento (diverso), D: Decisiones, A: Acciones, RP: Reglas de producción.

La parte de razonar es establecida, para el caso kowalskiano, con programas de la programación lógica. (El observar y el razonar se supondrán actividades realizables en tiempos acotados). No obstante, partes del actuar se ha considerado que es mejor manejarlo por reglas de producción (sobre todo acatando la exigencia de tiempo real para el funcionamiento de agentes), aunque las reglas de producción tienen el supuesto de que cada acción conlleva que el mundo cambie (actualización destructiva), en notoria contraposición a la referencia transpencial que la programación lógica ha propuesto a través de los años.

Lo que queda es tratar de realizar una entremezcla de estrategias computacionales: deducción en complemento a reglas de producción, o bien, la aplicación directa de reglas de producción preestablecidas cuando las deducciones auxiliares no sean posibles. Detallando, el ciclo abductivo kowalskiano, existen dos tipos de mecanismos computacionales posibles: de encadenamiento hacia adelante y hacia atrás.

El primer tipo de encadenamiento es típico de reglas de producción **if-then**, de la forma **if ANTECEDENTE(S) then CONSECUENTE**. Existen propuestas para manejar en lógica este tipo de reglas per se [8]. El segundo tipo permite manejar cláusulas de Horn, de la forma **CONSECUENTE if ANTECEDENTE(S)** [5]. La necesidad de considerar reactividad y racionalidad de forma intermezclada [7], sin que una domine sobre la otra, nace de dotar a agentes (individual o grupalmente) de racionalidad en ambientes realistas [4].

En este escrito nos enfocamos en la parte de razonamiento temporal y clausal, ya que nuestro objetivo primordial es la realización de deducciones auxiliares, y trataremos de mostrar que éstas deducciones pueden contribuir a una mejora en la toma de decisiones para activar reglas de producción y la planeación a largo plazo, entre otros aspectos.

3. Agentes robóticos y diseño de comunicación

En agentes robóticos, el tema de reactividad es manejado independientemente de algunos temas relacionados al conocimiento existencial dentro de un marco de referencia temporal. Pero frecuentemente la creación de planes, por ejemplo, es un producto resultante de “reflexionar” históricamente, y no nada más del conocimiento inmediato del entorno.

Una de las propuestas dadas en este artículo es que la incorporación de información temporal coadyuva en la mejora conductual de los agentes robóticos en términos espaciales. Se tendría que aceptar que el almacenar información espacial a lo largo del tiempo afecta las futuras decisiones y acciones del agente.

Luego, es de esperar que un agente espacial se beneficie de una habilidad de integrar observaciones locales en instantes específicos para derivar interrelaciones espaciales en una mayor escala.

Con propósitos de experimentación, se ha ideado una arquitectura modular, con módulos basados en una computadora ejecutando Linux (Raspberry Pi)² y otros módulos basados en microcontroladores Esp32³.

Dentro de este conjunto de módulos basados en microcontroladores, se tienen de 3 tipos: de recabado de datos sensoriales, de puesta en marcha de actuadores, y de gestores de comunicación Wi-Fi (por ejemplo, como puntos de acceso). Detallando:

1. Utilizamos una computadora Raspberry Pi junto con un robot PiCrawler⁴ modificado (agregándole llantas, básicamente) para tener un nodo robótico (programado en Python, con interacción de sockets y TCP a algunos programas de Prolog), y hemos llamado a este robot Tribot.
2. Utilizamos un microcontrolador Esp32 (en modalidades Esp32 Lolin mini y Esp32 Wroom) para establecer un punto de acceso ligado al nodo robótico (programado en Arduino).
3. Utilizamos un Esp32 como receptor de datos (servidor) de parte del nodo robótico (programado en Python).
4. Utilizamos otro controlador Esp32 como emisor de datos de un ambiente a el nodo robótico (programado en Arduino).

Se han utilizado programas en varios lenguajes de programación para completar el funcionamiento del sistema (Prolog, Python, MicroPython y Arduino). Algunos sensores que hemos utilizado son cámaras, sensores de detección de luz y de distancia.

Los actuadores utilizados son algunos motores (llantas), algunos servos (que conforman brazos robóticos), e indicadores lumínicos (leds) (para indicar algunas actividades o reportar situaciones especiales del entorno). Los microcontroladores Esp32 tienen capacidades de comunicación Wi-Fi.

Utilizamos este tipo de comunicación entre un robot y varios microcontroladores, dispersos, Esp32 S2 Mini, mismos que tienen diversas posibilidades relacionadas con la recepción sensorial o, a su vez, capacidades reactivas.

En el ambiente Linux tipo Ubuntu de Raspberry, se han creado diversos servidores de acciones, tanto atómicas como compuestas que nuestro robot Tribot acepta. La forma de controlar estos agentes robóticos es por codificación directa, por sockets, y por comunicación Wi-Fi.

² www.raspberrypi.org

³ www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

⁴ De la empresa Sunfounder.

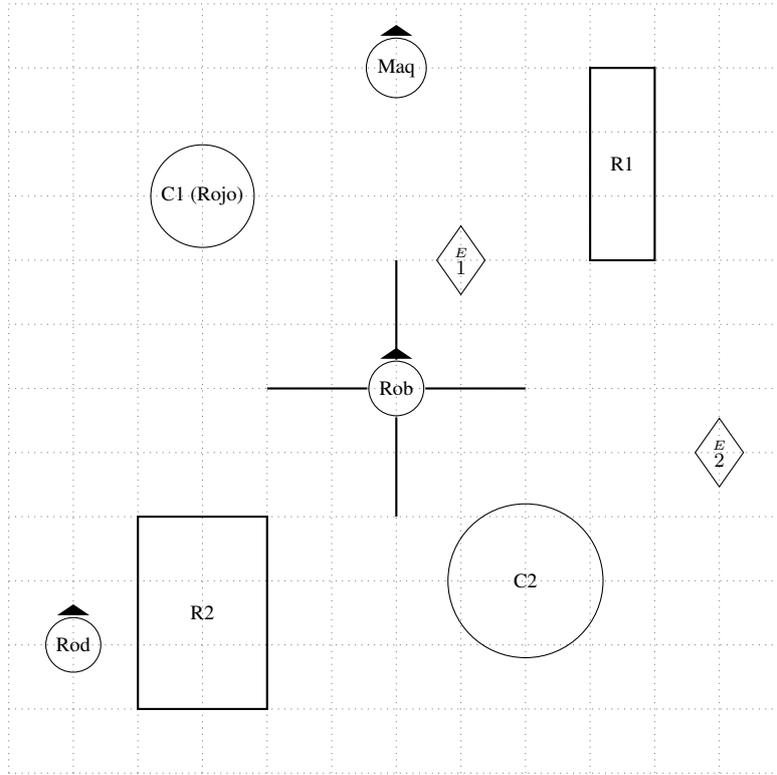


Fig. 4. Escenario inicial. Alcance de percepción.

4. Escenario robótico de estudio

Utilizaremos un conjunto de tres agentes robóticos (robots, para acortar). Nombraremos a los robots Rob, Maq y Rod. Los robots deambulan en un ambiente relativamente controlado pero semi-desconocido. En [17] se establece un problema parecido con métodos y objetivos diferentes, aunque con un punto de intersección esencial basado en la transferencia de información ocasional entre agentes.

Los robots tienen diversos sensores, algunos de los cuales son particulares o privados y otros más son irradianes o públicos. Consideraremos que cada robot tiene un vector de n entradas, con algunas entradas de información dedicadas a percepción sensorial, otras a la emisión, y otros más de información diversa. Todos los robots comparten un punto de acceso.

No se supondrá una red de Internet ya sea por razones de ausencia de señal, o bien de privacidad o bien por la oportunidad de los microcontroladores Esp32 de configurarse como puntos de acceso por sí mismos, así siendo esto una decisión de diseño. Dada la falla de un punto de acceso, otro más se activará inmediatamente, dando lugar a un sistema tolerante a desconexiones y autorreparable en relación a las fallas de intercomunicación.

Como supuestos de movilidad (ver Fig. 4) tenemos lo siguiente: Cada robot puede moverse a lo largo de la cuadrícula señalada en esta figura, en la dirección hacia adelante (f), hacia atrás (b), a la derecha (r) o a la izquierda (l), en unidades enteras (siendo la actual cuadrícula una de esquina inferior izquierda ubicada en $(-6, -6)$ y en esquina superior derecha ubicada en $(6, 6)$). La marca de un triángulito oscuro sobre cada robot señala su respectiva dirección hacia adelante, en el momento captado en la figura.

En la Fig. 4 se muestra la posición de los robots Rod, Rob y Maq. Las figuras sombreadas son obstáculos sólidos, detectables por cierto tipo de sensores (ultrasónicos, por ejemplo). Además, los objetos marcados por una E son microcontroladores Esp32 con capacidades sensoriales y de emisión de información (pero no actuadores), configurables como puntos de acceso.

Con estos puntos de acceso es posible compartir información remotamente. Por un diseño en meta-nivel, estipulamos que el círculo pequeño C1 es rojo, el círculo grande C2 es azul, el rectángulo pequeño R1 es verde y el rectángulo grande R2 es amarillo. En la Fig. 4 también se describe la “percepción” de objetos. Aquí nos bastará una abstracción de un umbral de cercanía para estudiar los efectos de cambios internos (fluentes) de la información de los robots (su “conocimiento”) a través del tiempo.

Mediante Wi-Fi es posible que cada robot tenga información de su propia ubicación. Bajo la instancia de interacción con un objeto (acción), el fluente asociado al “conocimiento” del robot cambia de “coordenada (x, y) ” vacía a ocupada.

Bajo la instancia de encuentro casual entre dos robots, cada robot cambia algunos fluentes de “información no compartida” a “información compartida”, de tal forma que los robots se comunican entre sí y se envían la información de qué lugares están ocupados por objetos. Describimos ahora algunas narrativas basadas en intervalos. Suponemos que los agentes robóticos están informados de lo siguiente ⁵:

1. Hay tres robots en este mundo.
2. Hay 4 objetos.
3. Dos objetos son círculos, C1, C2, de diferente tamaño, C1 siendo el pequeño y C2 siendo el grande.
4. Dos objetos son rectángulos, R1, R2, también de diferente tamaño, esta vez siendo R1 el pequeño y R2 el grande.

Establecemos ahora una narrativa global, basada en intervalos. Durante el primer intervalo (de 0 a 10 unidades de tiempo, digamos), tenemos:

1. Rod parte de la parte inferior izquierda y camina en línea recta hacia arriba.
2. Rod encuentra un rectángulo, pero no sabe si es el grande o el pequeño, aunque detecta que es de color amarillo.
3. Rod sigue su camino y se encuentra con un círculo, que puede detectar que es el pequeño, y que además detecta que es rojo.
4. Rod permanece quieto durante el resto de este periodo.

⁵ Notemos que es indispensable contar con bases de datos que manejen actualizaciones posiblemente destructivas. La recuperación de registros históricos longevos no está descartada si se requiere de, por ejemplo, del aprendizaje automático en alguna modalidad.

5. Rob permanece siempre quieto durante todo este intervalo.
6. Maq parte de su posición hacia la parte superior derecha del escenario. Encuentra un rectángulo, que sabe que es el pequeño y que su color es verde.

Definamos el predicado **conoce**/4, **conoce**(AR,NObj,Color,tamaño(Tam)), el cual describe que el agente robótico, AR, conoce un objeto con nombre NObj, color Color, y tamaño Tam, siendo Tam grande, pequeño o desconocido. Al final de este intervalo se tiene el siguiente conjunto de instancias del fluente **conoce**/4:

$$\{\text{conoce}(\text{Rod},\text{R2},\text{amarillo},\text{tamaño}(\text{desconocido})), \\ \text{conoce}(\text{Rod},\text{C1},\text{rojo},\text{tamaño}(\text{pequeño})), \\ \text{conoce}(\text{Maq},\text{R1},\text{verde},\text{tamaño}(\text{pequeño}))\}$$

Rob no ha hecho uso de su predicado **conoce**. Describamos lo que pasa durante el segundo intervalo (de 11 a 20 unidades de tiempo):

1. Esta vez Rod permanece quieto.
2. Rob, por su parte, camina del centro del escenario hacia la izquierda (desde nuestro punto de vista) y se halla tanto con el rectángulo R2, que sabe que es el grande y es amarillo, como con el círculo C1, que no sabe de su tamaño, pero sabe que es rojo.
3. Maq se desplaza hacia el centro de la parte izquierda del escenario, en donde es capaz de percibir la presencia del círculo C1, que detecta que es rojo pero ignora su tamaño.

Al final de este intervalo se tiene:

$$\{\text{conoce}(\text{Rob},\text{R2},\text{amarillo},\text{tamaño}(\text{grande})), \\ \text{conoce}(\text{Rob},\text{C1},\text{rojo},\text{tamaño}(\text{desconocido})), \\ \text{conoce}(\text{Maq},\text{C1},\text{rojo},\text{tamaño}(\text{desconocido}))\}$$

Describamos ahora qué acontece durante el tercer intervalo (de 21 a 30 unidades de tiempo). Los robots se reúnen en el punto medio de la izquierda del diagrama. (Otra variante es suponer que se utilizan sus capacidades Wi-Fi para comunicación remota). Se tiene, dado que las leyes inerciales se apliquen (los objetos permanezcan como objetos existentes, con sus colores y tamaños descubiertos en su momento):

$$\{\text{conoce}(\text{Rob},\text{R2},\text{amarillo},\text{tamaño}(\text{grande})), \\ \text{conoce}(\text{Rob},\text{C1},\text{rojo},\text{tamaño}(\text{desconocido})), \\ \text{conoce}(\text{Maq},\text{C1},\text{rojo},\text{tamaño}(\text{desconocido})), \\ \text{conoce}(\text{Maq},\text{R1},\text{verde},\text{tamaño}(\text{pequeño})), \\ \text{conoce}(\text{Rod},\text{R2},\text{amarillo},\text{tamaño}(\text{desconocido})), \\ \text{conoce}(\text{Rod},\text{C1},\text{rojo},\text{tamaño}(\text{pequeño}))\}$$

Supongamos que cada robot comparte su conocimiento del mundo con el resto de robots:

1. Primera conclusión: Hay 4 objetos (geométricos) en este mundo: C1, C2, R1 y R2.
2. Segunda conclusión, de los colores:

- a) C1 es rojo (descubierto por Rod).
 - b) C2 es azul (descubierto por Maq).
 - c) R1 es verde (descubierto por Maq).
 - d) R2 es amarillo (descubierto por Rod, confirmado por Rob).
3. Tercera conclusión, de los tamaños.
- a) C1 es el círculo pequeño (sabido por Rod).
 - b) C2 es de tamaño desconocido, según Maq, pero conociendo la información de Rod, ahora se sabe que es de tamaño grande.
 - c) R1 es de tamaño pequeño (según Maq).
 - d) R2 es de tamaño grande (según Rob, pero apoyándose en lo dicho por Maq).

Notemos que en el caso de una experiencia propia es posible anotar que un objeto se ha llegado a conocer de forma directa. En este caso describimos un evento que sería el de “cercanía”. Este evento dependería de algunos elementos que se involucren, tal como el agente robótico y el objeto:

Evento: **cercanía**(AR,Ob,conocer(Ob,Props*),t),

donde Props* indica una lista de atributos de un objeto, puestos como argumentos en el fuente. El fuente resultante con un argumento adicional de temporalidad sería el de **conoce**/5, con un argumento de cuándo el AR conoce tal objeto. Otro evento que destaca en la anterior descripción es el siguiente:

Evento: **traspasoInfo**(AR1,AR2,[ListaObjConocidos]).

En el caso de incorporar cómo se ha conocido el objeto, si por experiencia propia o ajena, definiríamos **conoceMedio**/6, que es como el fuente **conoce**, pero con un argumento extra para indicar si se llegó a conocer un objeto por experiencia propia o ajena:

Fuente: **conoceMedio**(AR,Obj,Props*,experienciaPropia,t).

Dado que un evento de traspasar información ocurra, para un agente receptor le es posible conocer la existencia de un objeto de una forma indirecta, a lo que nombramos, como ya mencionamos, experiencia ajena:

Fuente: **conoceMedio**(AR,Ob,Props*,experienciaAjena,t).

Notemos que hemos tomado el intervalo de 21 a 30 como un intervalo de actividad relacionado con compartir información. En un instante o intervalo posterior al momento 30 se puede utilizar la información sea de forma individual, grupal o global para decidir qué posibles actividades más podría realizar cada agente robótico.

En un formalismo que de énfasis al hecho de que los agentes se pueden organizar en agrupaciones, tal como [13], el tener criterios para realizar agrupaciones mediante la caracterización de los agentes (mediante algo conocido como atributos) permite que sea la base de trabajo realizable en equipo, por ejemplo. Dado el conocimiento geométrico grupalmente obtenido, es posible obtener planificaciones que optimicen el actuar espacial (planificando rutas óptimas) de los agentes robóticos [11].

5. Conclusiones y trabajo a futuro

En muchos sistemas multi-agente, las capacidades de los agentes se realzan si se toma en consideración el factor tiempo. En algunas descripciones de agentes, tales como [4]), las creencias, deseos e intenciones implícitamente tienen una noción de tiempo, pero para su manejo se requiere de algunas lógicas modales que parecen complicar los fundamentos teóricos.

En [6] se afirma que, desde la perspectiva del ciclo abductivo kowalskiano, las creencias son condiciones, los deseos son metas y las intenciones son planes, todo en un subsistema de la lógica de primer orden (aunque con preprocesamientos tipo cosificación (reification), en el que los flujos pasan a ser argumentos de predicados).

En un trabajo a futuro se planteará cómo las reglas de producción pueden ser temporalmente complementadas. En este escrito se ha diseñado una arquitectura distribuida basada en comunicación Wi-Fi y controladores Esp32, con agentes robóticos capaces de deambular en un ambiente artificialmente diseñado, y distinguir colores y tamaños de algunos objetos colocados al azar.

También, se ha diseñado un recabamiento parcial de datos que se desarrolla en el tiempo para obtener consenso y conocimiento de una realidad global. Para ello se han utilizado diversas instancias de narrativas, cada una aportando un grado de generalidad apropiado al número de agentes involucrados, así como el intercambio oportuno de información.

Queda como trabajo a futuro el considerar la utilización de bases de datos locales y globales, con actualización destructiva y dando entrada a la formulación de razonamiento no-monótono (vía la adquisición de nueva información). También, es trabajo a futuro integrar estos resultados con un formalismo de sistemas multi-agentes específico; tenemos ya seleccionado para tal fin a SCEL [13].

En un escenario que posteriormente estudiaremos, se planea utilizar un conjunto de agentes robóticos para realizar trabajo cooperativo que requiera deliberación y planeación; esto se hará posible con base en historiales individuales y mediante la gestión de información remota entre los agentes.

Agradecimientos. Agradecemos al Instituto de Computación de la Universidad Tecnológica de la Mixteca las facilidades brindadas para llevar a cabo nuestra investigación acerca de agentes robóticos. También, agradecemos las valiosas observaciones para la mejora de este trabajo por parte de los árbitros de COMIA 2023.

Referencias

1. Chaudet, H.: Extending the event calculus for tracking epidemic spread. *Artificial Intelligence in Medicine*, vol. 38, no. 2, pp. 137–156 (2006) doi: 10.1016/j.artmed.2005.06.001
2. Fisher, M., Gabbay, D. M., Vila, L.: *Handbook of temporal reasoning in artificial intelligence* (2005)
3. Furia, C. A., Mandrioli, D., Morzenti, A., Rossi, M.: Modeling time in computing. *ACM Computing Surveys*, vol. 42, no. 2, pp. 1–59 (2010) doi: 10.1145/1667062.1667063

4. Georgeff, M., Pell, B., Pollack, M., Tambe, M., Wooldridge, M.: The belief-desire-intention model of agency. *Intelligent Agents V: Agents Theories, Architectures, and Languages*, pp. 1–10 (1999) doi: 10.1007/3-540-49057-4_1
5. Kowalski, R.: *Logic for problem solving*. North-Holland (1979)
6. Kowalski, R., Sadri, F.: Integrating logic programming and production systems in abductive logic programming agents. *Web Reasoning and Rule Systems*, pp. 1–23 (2009) doi: 10.1007/978-3-642-05082-4_1
7. Kowalski, R., Sadri, F.: Reactive computing as model generation. *New Generation Computing*, vol. 33, no. 1, pp. 33–67 (2015) doi: 10.1007/s00354-015-0103-z
8. Kowalski, R., Sadri, F.: Programming in logic without logic programming. *Theory and Practice of Logic Programming*, vol. 16, no. 3, pp. 269–295 (2016) doi: 10.1017/s1471068416000041
9. Kowalski, R., Sergot, M.: A logic-based calculus of events. *New Generation Computing*, vol. 4, no. 1, pp. 67–95 (1986) doi: 10.1007/bf03037383
10. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, vol. 21, no. 7, pp. 558–565 (1978) doi: 10.1145/359545.359563
11. Lien, J. M., Lu, Y.: Planning motion in similar environments with obstacles. In: *Proceedings of Robotics: Science and Systems V* (2009)
12. McAreavey, K., Bauters, K., Liu, W., Hong, J.: The event calculus in probabilistic logic programming with annotated disjunctions. In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pp. 105–113 (2017)
13. Nicola, R. D., Loreti, M., Pugliese, R., Tiezzi, F.: A formal approach to autonomic systems programming: The SCEL language. *ACM Transactions on Autonomous and Adaptive Systems*, vol. 9, no. 2, pp. 1–29 (2014) doi: 10.1145/2619998
14. Pfeifer, R., Bongard, J. C.: *How the body shapes the way we think: A new view of intelligence*. MIT Press (2007)
15. Shanahan, M.: The event calculus explained. *Artificial Intelligence Today: Recent Trends and Developments*, pp. 409–430 (1999) doi: 10.1007/3-540-48317-9_17
16. Varela, F. J., Thompson, E. T., Rosch, E.: *The embodied mind: Cognitive science and human experience*. MIT Press (1992)
17. Wijmans, E., Savva, M., Essa, I., Lee, S., Morcos, A. S., Batra, D.: Emergence of maps in the memories of blind navigation agents (2023) doi: 10.48550/ARXIV.2301.13261