ISSN 1870-4069

Conformación dinámica de equipos colaborativos en un sistema multiagente ambiental

Manuel Hernández, Eduardo Sánchez-Soto

Universidad Tecnológica de la Mixteca,
Instituto de Computación,
México

{manuelhg, esanchez}@mixteco.utm.mx

Resumen. En este artículo se plantea la utilización de un formalismo de manejo de equipos en sistemas distribuidos aplicado a un conjunto de componentes autónomos que manejan una historia individual de sus propias acciones y representaciones del ambiente, incluidas las interacciones de los componentes autónomos entre sí. Tal historia (narrativa) es tratada a través de un cálculo de eventos, el cual no sólo registra los eventos que ocurren, sino que además reporta qué propiedades del sistema de agentes (vistos como componentes autónomos) son verdaderas en un instante dado. Como resultado de este trabajo mostramos la posibilidad de conformar equipos de trabajo de agentes para llevar a cabo tareas realizables colectivamente.

Palabras clave: Colaboración, autonomía, ambiente.

Dynamic Formation of Collaborative Teams in an Environmental Multi-agent System

Abstract. This article proposes the use of a team management formalism in distributed systems applied to a set of autonomous components that manage an individual history of their own actions and representations of the environment, including the interactions of the autonomous components with each other. Such a history (narrative) is treated through an event calculus, which not only records the events that occur, but also reports which properties of the agent system (seen as autonomous components) are true at a given moment. As a result of this work, we show the possibility of forming working teams of agents to carry out collectively achievable tasks.

Keywords: Teamwork, autonomy, environment.

1. Introducción

En este trabajo se propone abordar el problema de la colaboración entre agentes (componentes autónomos) a través de la conformación de equipos en un sistema distribuido ambiental.

Se plantea que tal problema puede abordarse mediante un formalismo conocido como SCEL [1] y un cálculo de eventos [8]. Este enfoque proporciona mecanismos computacionales para que algunos agentes autónomos se informen colectivamente de su entorno y, para nuestro caso, ésta información brinde un planteamiento de gestión de equipos de agentes. Así, se permite a los agentes el gestionar equipos basado en la percepción de los cambios en su entorno local y compartirlo con uno o más de los agentes autónomos existente para de esta manera tener mayores posibilidades de acciones colectivas o individuales, y suponiendo que tales acciones que modifican, a su vez, el ambiente en donde existen.

Nuestro planteamiento considera a los agentes (componentes autónomos) dentro de un ambiente sin predefinirles un modelo del mundo, y en su lugar la interacción directa del agente con su entorno genera modelos conforme pasa el tiempo. Aquí, empleamos una versión especializada del formalismo SCEL para aplicarlo a un sistema de agentes robóticos con interfaces, junto con un cálculo de eventos, y de esta manera garantizar una dinámica de formación de equipos y de posibles coaliciones (conjuntos de equipos).

Para obtener decisiones razonables de membresía a uno u otro equipo por parte de los agentes, cada agente puede recopilar, compartir o recibir información de su entorno, generando una noción de verdad consensuada y colectiva que puede aplicarse a la creación de equipos de trabajo como parte del diseño y ejecución de planes o bien como respuesta a labores que requieren atención inmediata. El cálculo de eventos aquí aplicado proporciona un marco computacional para representar y razonar sobre un ambiente dinámico, generando modelos lógicos que cambian con el tiempo. Al combinar éste cálculo con la noción de interfaces de SCEL, podemos controlar la interacción del agente.

Las interfaces permiten al agente percibir y actuar sobre su entorno, proporcionando un esquema de comunicación entre el agente y el mundo exterior [18], así como con otros agentes. De la misma forma, la percepción del ambiente posibilita, entre otras actividades, la gestión de equipos (la importancia de cómo el percibir el ambiente influye en la toma de decisiones grupales está descrito para un caso visual en [12] y en general en [3]).

El cálculo de eventos e interfaces proporcionan mecanismos computacionales adicionales para que los agentes se informen mejor de su entorno [5]. Al representar eventos y sus relaciones causales, los agentes (o en nuestro ejemplos, agentes robóticos) pueden crear equipos, actualizar sus membresías, y ampliar o eliminar tales equipos, para así anticipar cambios y tomar decisiones de acción basadas en un manejo efectivo de colaboración. Veremos que el conocimiento compartido es un aspecto fundamental que permite a los a los agentes alcanzar objetivos específicos para la conformación de equipos.

Panorama de este trabajo. En la Sección 2 se da una introducción al formalismo SCEL de Rocco de Nicola et al. [15], acentuando las características que un agente autónomo debe poseer para apoyar la formación de equipos; se menciona aquí que un concepto a resaltar es el de vector de interfaces. En la Sección 3 se plantea la aplicación especializada de un cálculo de eventos ideado por Robert Kowalski y Marek Sergot [8] como complemento a la decisión de cómo se debe gestionar el tema de equipos colaborativos entre agentes.

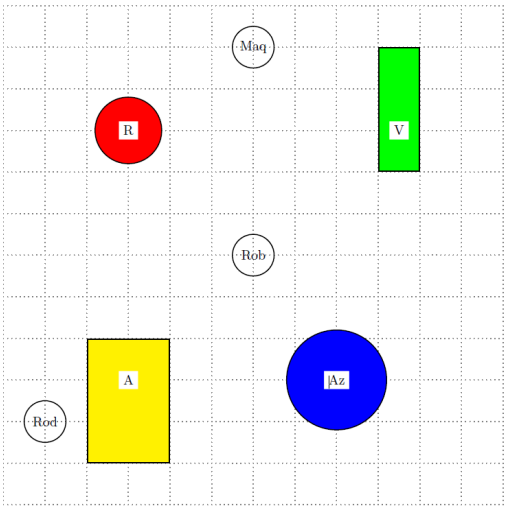

Fig. 1. Escenario inicial (V=verde, A=amarillo, R=rojo, Az=azul).

En la Sección 4 se ejemplifica nuestra propuesta tratando un par de casos de estudio de un conjunto de agentes robóticos explorando un ambiente. Al suponer objetivos a cumplir, y dada la información obtenida del entorno, se puede optar por formar equipos colaborativos para cumplir con sub-objetivos o los objetivos completos. Finalmente, se presentan algunas conclusiones en la sección final.

2. SCEL y la formación dinámica de equipos

Sea un conjunto de agentes autónomos conviviendo entre sí con la posibilidad de agruparse estructuradamente para llevar a cabo tareas grupales. Este tipo de agentes autónomos pueden considerarse como un tipo de robots inmersos en un ambiente [18]. Los agentes tienen posibilidades de comunicación, además de un apoyo informativo externo a cada agente pero interno al sistema (el equivalente a un depósito o nube local de almacenamiento compartido).

Se aprovechará la noción de interfaz de un componente autónomo para una aplicación de un cálculo lógico de eventos (que tiene a su vez una formulación natural en términos de programación lógica [7, 9, 2], con implementaciones en Prolog [10]) además del tratamiento de equipos (proponiendo la creación de un equipo, inscribiéndose a uno ya existente, abandonando uno, o eventualmente, eliminar otro).

Para alcanzar el objetivo planteado, se ha ideado una arquitectura modular en un sistema de agentes, en donde cada componente autónomo o agente (robótico) es un módulo computacional y reactivo (capaz de realizar físicamente acciones) basado en una computadora (Raspberry Pi) ejecutando Linux y otros módulos auxiliares más basados en microcontroladores. Dentro de este conjunto de módulos auxiliares basados en microcontroladores, se tienen 3 tipos: de recabado de datos sensoriales, de control de actuadores, y de gestores de comunicación Wi-Fi (por ejemplo, como puntos de acceso).

La técnica de comunicación mediante sockets y TCPs ha sido fundamental, ya que permiten la apertura de canales de comunicación con mensajes económicos y en tiempo real. Debido a estas posibilidades de comunicación, existe una dinámica de equipos: sea que un plan puesto en marcha considere el formar equipos, o sea que los equipos posibiliten el diseño de ciertos planes. En [16] se propone una formulación pragmática de tal formación de equipos basada en emitir o recibir notificaciones para organizar posible trabajo colaborativo. En este trabajo mostramos como las abstracciones tomadas del formalismo SCEL son lo suficientemente generales para apoyar también a la programación declarativa (representada en este caso por la programación lógica).

Los autores de [15] señalan una posibilidad de su formalismo: las interfaces contienen predicados que cambian con el tiempo. Esto nos lleva a ver que SCEL es adaptable también a una formulación particular y práctica de un cálculo lógico de eventos [8, 14], en donde los predicados que cambian con el tiempo son llamados fluentes; estos fluentes se utilizarían en la toma de decisiones de los componentes autónomos desde el punto de vista de narrativas peculiares o individuales (en contraste a narrativas generales), así mostrando que, con adecuadas restricciones, este cálculo de eventos tiene potencial para brindar racionalidad temporal e histórica en la toma de decisiones acerca de la gestión de equipos de agentes.

2.1. Descripción sucinta de un formalismo para el estudio de componentes autónomos: SCEL

Según SCEL, un componente autónomo (CA) dentro de un sistema distribuido se caracteriza por sus atributos e interfaces. (Tales componentes autónomos pueden ser vistos como agentes autónomos en el contexto de un agente que interactúa con su ambiente [18].) En la parte correspondiente a los atributos, se incluyen descripciones de los CAs tales como identidad, capacidades, coordenadas espaciales (ubicación), membresías a grupos, niveles de confianza, tiempos de respuesta, entre otros.

Particularmente, dentro de estos atributos y con miras a posibles integraciones grupales, cada CA consta de una o varias etiquetas que identifican su disponibilidad (o la ausencia de) para aceptar una membresía a un grupo de CAs a través de interfaces. Un CA puede pertenecer a varios equipos o a ninguno, y los mismos equipos pueden formar otros mayores llamados coaliciones. Esta conformación de super-equipos se detiene dependiendo de las necesidades de cada aplicación. Los CAs están capacitados para utilizar depósitos de información (DI).

Estos depósitos permiten compartir información que se obtiene de las actividades sensoriales de los agentes (o incluso información obtenida por deducción) para transmitirla previamente procesada o directamente a otros CAs. La ventaja de éste enfoque es el delegar labores a componentes especializados y complementar las fuentes de información sensorial o de deductivamente obtenida, con lo que podría equiparse con información detallada e indirecta a los demás CAs.

En nuestro enfoque, y por que el formalismo SCEL así lo permite (y lo promueve), las conductas de los CAs tienen una dualidad entre planes y secuencias de acciones, y a su vez entre planes y tratamientos de equipos. Estas conductas, entonces, son el resultado o de reaccionar a un ambiente o de racionalizar un estado (o estados del CA) para llegar a generar planes y acciones consecuentes.

El conocimiento que prevalece en una modelación de tipo SCEL tiene dos posibles fuentes: Cuando un CA adquiere tal conocimiento (información) directamente o bien cuando obtiene o complementa su información vía directa o vía depósitos existentes, para que un CA se informe de a cuáles equipos, si es el caso, el CA puede integrarse. Varias modalidades de interacción surgen entonces en la interacción con los depósitos: se puede contribuir con información propia, o bien se puede obtener información; o aún más, puede llegarse al caso de modificar (con los permisos apropiados) la información ya existente en un depósito (modificándola al grado de borrarla).

Las interfaces permiten que los agentes auto-examinen sus propios predicados "cambiantes" o fluentes de su propia situación o puedan adquirir información de otros agentes a los cuales tienen posibilidades de acceder y obtener su información. Con los fluentes, cada agente puede decidir qué es verdad en un tiempo t de su ambiente circundante. La información antes mencionada se ve plasmada a través de un vector de interfaces que consta de n entradas:

in_1	in_2	$\ldots$	in_k	out_1	out_2	$\ldots$	out_l	$pred_1$	$pred_2$	$\ldots$	$pred_m$

(1)

donde: $n = k + l + m$. Las entradas señaladas con *in* se utilizan para adquirir información del ambiente, sea de sensores internos, de servidores de sensores, e incluso de otros CAs. Las entradas señaladas con *out* son para emitir información ya sea de los sensores internos o de aquella otra información adquirida directa o indirectamente (por ejemplo, por medio de otros CAs). Por último, las entradas señaladas con *pred* sirven como aserciones temporales.

Estas aserciones son fórmulas lógicas que cambian con el tiempo, es decir, en la terminología del cálculo de eventos, fluentes. Quienes pueden modificar el valor de estos fluentes son o el mismo CA u otros componentes autónomos (teniendo permisos vía una política en vigor) o bien pueden ser deducidos del cálculo de eventos. SCEL permite restringir a quien se comparte la información de estas casillas, dependiendo la aplicación. Más información y definiciones formales extensas se encuentran en [15].

3. Un cálculo de eventos aplicado a agentes autónomos

Vemos ahora una sucinta descripción del cálculo de eventos. Mayor información puede ser obtenida de [17]. El cálculo de eventos tiene una formulación técnica basada en programas normales de la programación lógica. Algunos predicados del cálculo de eventos son fundamentales:

1. **Initiates** $(\alpha,\ \beta,\ \tau)$: El fluente β comienza a ser verdadero cuando acontece la acción α en el instante τ.

2. **Terminates** $(\alpha,\ \beta,\ \tau)$: El fluente β cesa de ser verdadero cuando acontece la acción α en el instante τ.

3. **InitiallyP** (β) El fluente β es verdadero desde el instante 0.

4. **Happens** $(\alpha,\ \tau)$ La acción α ocurre en el instante τ.

Tabla 1. Objetos seleccionados para nuestro caso de estudio.

Objeto	Características
Objeto 1 (Ob1):	Ciírculo azul (Ca)
Objeto 2 (Ob2):	Rectángulo amarillo (Ra)
Objeto 3 (Ob3):	Círculo rojo (Cr)
Objeto 4 (Ob4):	Rectángulo verde (Cv)

5. **HoldsAt** $(\beta,\ \tau)$ El fluente β es verdadero en el instante τ.
6. **Clipped** $(\tau_1,\ \beta,\ \tau_2)$: El fluente β cesa de ser verdadero en el intervalo $(\tau_1, \tau_2]$.
7. **Releases** $(\alpha,\ \beta,\ \tau_2)$: El fluente β no es ya más "inercial" después del evento α ocurrido en el instanteτ.
8. **InitiallyN** (β): El fluente β es supuesto como falso desde el instante 0.
9. **Declipped** $(\tau_1,\ \beta,\ \tau_2)$: El fluente β es verdadero desde algún instante que está en el intervalo $(\tau_1,\ \tau_2)$.

Damos a continuación una formulación de índole intuitiva que abarca los conceptos que utilizaremos posteriormente. Primero, se tiene una ley inercial tipo 0: Si suponemos que en un tiempo 0 la aserción A es verdadera (**HoldsAt**$(A,\ t)$), y dado un $t > 0$, la aserción A sigue siendo verdadera, a menos que un evento, que haya ocurrido entre 0 y t, haya cambiado su valor. Una generalización es una ley inercial tipo 1: Ahora un evento e ocurre en un tiempo t_1 (**Happens**(E, t_1)), que hace que A sea verdadera.

Para un tiempo t_2, con $t_2 > t_1$ A sigue siendo verdadera a menos que un evento e1 que ocurra entre t_1 y t_2 haga a A falsa. Finalmente, A es truncable (**Clipped**$(t_1,\ A,\ t_2)$) entre t_1 y t_2 si existe un evento entre t_1 y t_2 tal que haga a A falsa entre t_1 y t_2. En otras palabras, para una ley inercial 0, con $t_1 > 0$, A es verdadero en t_1 si A no fue truncable entre 0 y t_1. Para una ley inercial tipo 1, A no es truncable entre t_1 y t_2, con $t_1 < t_2$, y t_1 siendo el instante en que A comienza a ser verdadero.

Para un problema como el de agentes autónomos como el que estamos tratando, la fundamentación axiomática del cálculo de eventos es la siguiente: Cada CA ignora todo acerca de su ambiente en un tiempo inicial, pero puede tener información inicial propia por auto-inspección. Cada evento de un CA al adquirir información sensorial de un objeto O del ambiente hace que su información involucre a O. Cada vez que un CA se comunica con otro CA el acervo de información de ambos se comparte, haciendo que cada CA tenga mayor o igual o información a la previamente ya adquirida. El fluente de "contactar" un agente a otro es modificado después de que dos agentes se encuentran.

Notemos, entonces, que un CA siempre adquiere información incrementalmente. Notemos también que la información ya poseída tiene que compatibilizarse con la recibida. Cada CA, además, tiene un manejo de tiempo que es global. Es fundamental cerciorarse que éste cálculo de eventos es implementable según la programación lógica, pero reconociendo que el tipo de historia que cada CA maneje puede ocasionar problemas en la saturación de memoria, lo que obliga a tomar posibles estrategias para el almacenamiento de datos (como el borrar datos antiguos o bien algunos marcados

como de baja prioridad). De todas formas, algunos detalles pragmáticos representan de por sí sus propios desafíos. Se mencionan ahora algunas condiciones de aplicabilidad del cálculo de eventos, para lo cual es necesario identificar los siguientes conjuntos:

1. Un conjunto de acciones o eventos (ambos identificados como el mismo conjunto).
2. Un conjunto de fluentes, que son predicados con valor cambiante dependiendo de las acciones que van ocurriendo; cada fluente debe estar relacionado con al menos un evento, sea para que el fluente comience a ser verdadero por la ocurrencia del evento o el fluente comience a ser falso por tal ocurrencia (la liberación de los fluentes de la influencia de eventos es posible, para ciertas aplicaciones, pero tiene que señalarse explícitamente).
3. Un conjunto de valores de tiempo, los cuales pueden ser momentos o intervalos; tales valores deben ser comparables, discretizables (hasta un grado necesario), y de tal forma que toda acción tenga un momento de ocurrencia y todo fluente tenga un valor definido de veracidad dado un momento en el que el fluente se inspeccione. Hablamos de intervalos si tenemos dos momentos t_1 y t_2, con $t_1 < t_2$ tal que el conjunto $\{t,\ t_1 < t \text{ y } t < t_2\}$ sea no vacío.

También es necesario considerar restricciones de integridad, como aquellas suposiciones, leyes, o condiciones de la realidad que son explícitamente establecidas y que los eventos junto con los fluentes deben cumplir en cada ocasión (de otra manera, estarían ocurriendo inconsistencias o violaciones de temporalidad, por ejemplo) y una teoría de causalidad, que condiciona de forma causal a los eventos y a los fluentes. Para obtener una aplicación adecuada del cálculo de eventos, aquí se listan algunos temas que se deben abordar dependiendo la aplicación en mente:

Tipos de narrativas. Una narrativa es la identificación de un conjunto de ocurrencias de eventos en el tiempo. Las narrativas aquí elaboradas son individuales, por cada agente existente. Notaremos que el compartir tales narrativas permite una diseminación de información para una mejor toma de decisiones.

Granularidad. Se supondrá una descripción del tiempo discretizada, con respecto a un grado de granularidad, lo suficiente para no perder los instantes de la ocurrencia significativa de eventos que modifiquen fluentes. También manejaremos una granularidad espacial, para que la movilidad de los robots sea descrita por coordenadas enteras [13].

Intervalos. Todos los intervalos serán supuestos sobre el tiempo discretizado y serán abiertos por la izquierda y cerrados por la derecha; tales intervalos $(t_1, t_2]$ se describen como conjuntos discretizados con elementos t cumpliendo que $t > t_1$ y $t \leq t_2$.

4. Escenario de agentes robóticos como componentes autónomos

Seguimos algunas ideas de SCEL para la configuración de un escenario robótico (nombrando robot a un agente autónomo, en adelante) en donde nombraremos a los robots Rob, Maq y Rod como tres robots (que jugarían también el papel de componentes autónomos, en la terminología de SCEL) que exploran en un ambiente relativamente

controlado. Se supondrá que los agentes robóticos están conectados a una red local de alta confiabilidad, de forma inalámbrica mediante tecnología Wi-Fi. Notemos que esto conlleva una arquitectura de equipos de trabajo [6], cuando se considera la automatización de las interacciones entre los componentes autónomos. Como supuestos de movilidad (ver Figura 1) tenemos lo siguiente: Cada robot puede moverse a lo largo y ancho 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 $(6,\ 6)$). Consideremos un escenario como el de la Figura 1.

En este escenario existen los tres agentes robóticos mencionados, Rod, Rob y Maq, que bien pueden ser de naturaleza heterogénea, pero es fundamental que mantengan un conjunto uniforme de interfaces. En la Figura 1 las figuras sombreadas son obstáculos sólidos, detectables por cierto tipo de sensores (ultrasónicos, por ejemplo), e impenetrables e indeformables por los robots. Bajo la instancia de encuentro casual entre dos robots (encuentro considerado como un evento), los fluentes de cada robot cambian de "información no compartida" a "información compartida", de tal forma que los robots se comunican entre sí y se mandan la información de qué lugares están ocupados por objetos y qué objetos son. Planteamos los siguientes a antecedentes:

1. Hay tres robots en este mundo. Todos tienen una identidad.
2. Hay objetos que son circulares: C1, C2, ...
3. Hay objetos rectángulares, R1, R2, ...
4. Cada objeto tiene asociado un color (rojo, amarillo, verde o azul) y una ubicación (tomada como el centroide del objeto).

Por ejemplo, podríamos considerar una interfaz (tipo vector) que conste de un par de celdas del vector de interfaz siendo celdas recepción de información, in_1 e in_2; otro par de celdas del vector siendo celdas de emisión de información out_1 y out_2; y otras algunas entradas relacionadas con predicados que pueden cambiar, tales como el o los equipos a los que pertenece el robot ($pred_1$), el nivel de energía del robot ($pred_2$), su ubicación ($pred_3$), así como la información de los objetos que vayan adquiriendo conforme exploran el mundo en donde se hayan inmersos ($pred_4$). El vector de interfaz para estos robots quedaría así:

in_1	in_2	out_1	out_2	$pred_1$	$pred_2$	$pred_3$	$pred_4$

(2)

El fluente $pred_4$ se puede desglosar así (pues la celda en sí puede ser lo suficientemente compleja como fórmula lógica, incluidos los sujetos o estructuras de datos): un fluente es K, que consta de una lista de objetos conocidos por cuenta propia, de conocimiento "experimentado", K-DB, del tipo:

```
conozco1(objeto, características, ubicación, instante).
```

Y otra más, KA-DB, de objetos conocidos por comunicación con otros robos o de conocimiento "adquirido", de tipo:

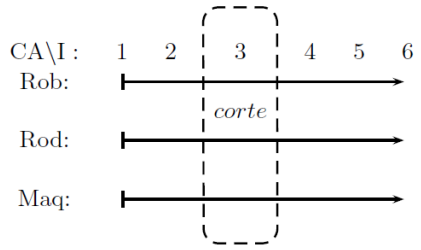

Fig. 2. Líneas de tiempo para los agentes robóticos Rob, Rod y Maq. Aquí, CA=componente autónomo, e I=instante.

```
conozco2(objeto, características, ubicación, instante).
```

Sea que los robots exploran su mundo, el cual contiene algunos objetos geométricos coloreados dispersos (ver la descripción en la Figura 1). La exploración de los robots se lleva a cabo en una área delimitada. Al estar dentro de una zona cercana a un objeto, lo reconocen, vía algún mecanismo sensorial (hemos utilizado robots con cámaras que reconocen códigos QR, por ejemplo). Del conocimiento propio experimentado K por un agente, no hay ninguna duda de su veracidad desde el punto de vista del propio agente.

Del conocimiento adquirido, KA, se pueden idear mecanismos de consenso grupal para así fortalecer la salvaguarda del sistema contra información falsa o maliciosa. Vamos a suponer que algunas entradas de la interfaz vectorial como la K-DB o la KA-DB. éstos fluentes contribuyen a la toma de decisiones con respecto a la creación, de equipos de trabajo que realizarían tareas grupales, así como a la modificación de una membresía, la modificación o eliminación de tales equipos.

Por ejemplo, se puede considerar como un atributo el número de objetos conocidos, el número de objetos con cierta forma geométrica o color, o bien el número total de objetos conocidos, que bien puede obtenerse de la suma de objetos conocidos por experiencia propia y de aquellos conocidos por conocimiento adquirido ajeno. Tomamos éste último criterio, para ejemplificar.

Debido a que no se conoce de antemano por los agentes el número total de objetos, no se tendría una condición de finalización en el tiempo, sino solo posibles "cortes" de información compartida en instantes determinados. En el diagrama de la Figura 2 (inspirado de técnicas descriptivas de cómputo y sistemas distribuidos [11]), se grafican alguas líneas de tiempo. Se muestra un corte (o instantánea) en el instante 3 encerrado en un rectángulo hecho de segmentos entrecortados.

Cada línea representa el tiempo de existencia de cada agente. Con mayores decoraciones gráficas sobre cada línea de tiempo, como veremos, se pueden representar eventos que acontecen a los agentes durante el transcurso del tiempo. Dos eventos nos interesan en particular: el evento en el que un agente conoce un objeto y las características de este objeto, y otro evento que consiste en que simultánemente un evento de envío de mensajes acontece con su recepción[1], así logrando que el conocimiento entre agentes sea diseminado.

[1]Esta simultaneidad no sería realista en general en sistemas distribuidos, por el retardo de tiempo entre el envío y recepción del mensaje, pero que que aquí abstraemos tal situación.

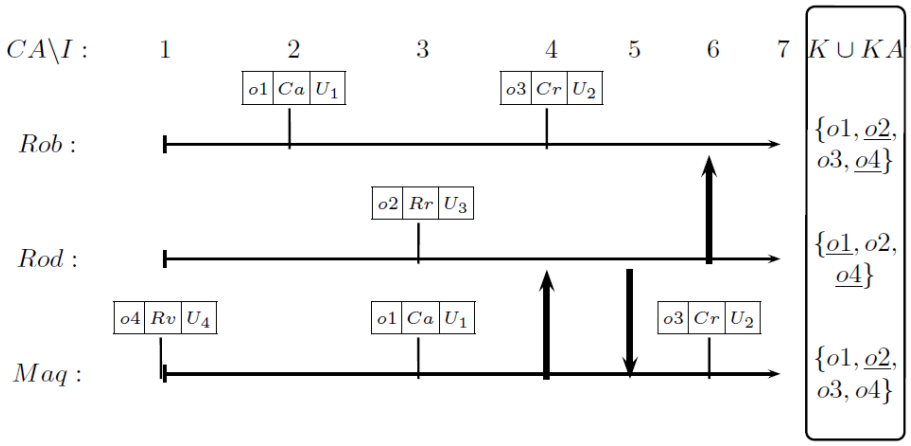

Fig. 3. Diagrama de temporalidad (timing diagram), con instantes cuando cada robot (agente) obtiene información de su entorno. En el rectángulo de la derecha está la información recopilada por cada agente hasta el instante 7.

El primer evento es representado por tres casillas anexas mediante un segmento a una línea de tiempo. El segundo evento es representado por un segmento grueso y dirigido (señalando el remitente y el destinatario (receptor) de un mensaje informativo acerca de los objetos conocidos) que conecta, ortogonalmente, dos líneas de tiempo. En la Figura 3 se muestra el uso de este tipo de diagramas para representar una historia de tiempo, que consta de una secuencia ordenada de instantáneas o (del inglés y de la terminología de sistemas distribuidos, snapshots) o cortes. En este caso particular, cada U_i es la ubicación de un objeto, que bien podría ser información relevante como un criterio a seguir para conformar equipos en otros contextos.

Consideremos una narrativa asociada al agente Rob, desde su punto de vista: En el instante 2 adquirí conocimiento acerca de la existencia del objeto 1, el cual es un círculo de color azul. En el instante 4 aquirí conocimiento del objeto 3, el cual es otro círculo de color rojo. En el instante 6 recibí informacion del agente Rod, quien me hizo conocer que existe un objeto 1, circular, de color azul (redundante, pero comprueba consistencia de informacion); tambien en ese mismo paquete de informacion se me informo del objeto 4, rectangular, de color verde, y del objeto 2, rectangular, de color rojo. Para el instante 7 conozco los objetos $\{o1,\ o2,\ o3,\ o4\}$ asi como sus formas y colores asociados. Este es un ejemplo de eventos identificables en esta narrativa:

a) AdquirirConocimiento/2,

b) EnviarInfo/2,

c) RecibirInfo/2 (aquí el número adjunto representa la aridad del predicado).

A su vez, algunos fluentes identificables serían:

a) NotificadoAgente(A,B): B es notificado de alguna información por A;

b) Conoce(A,Ob): el agente A conoce información (detallada) acerca del objeto Ob.

Habiendo ya tratado un ejemplo de colaboración general con el objetivo de recabar información de un ambiente ahora presentamos otro ejemplo, pero esta vez involucrando otro agente (robótico) llamado Ben. Sea que acontecen los siguientes eventos, en donde por cada instante de tiempo t se anexa de forma sufija su tiempo de ocurrencia (notemos que esto nos permite describir eventos simultáneos), consecutivamente:

- **Happens**(Hallar(Maq, Rv), t1).
- **Happens**(Hallar(Rob, Ca), t2).
- **Happens**(Hallar(Ben, Rr), t2).
- **Happens**(Hallar(Rod, Rr), t3).
- **Happens**(Hallar(Maq, Ca), t3).
- **Happens**(ComunicaHallazgos(Rob,Ben),t4).
- **Happens**(ComunicaHallazgos(Maq,Rod),t4).
- **Happens**(ComunicaHallazgos(Rod,Maq),t5).
- **Happens**(ComunicaHallazgos(Rod,Rob),t6).

Ahora, dos equipos pueden formarse, dependiendo de si se tiene información de la existencia de ciertos objetos. Por un lado, a partir del instante 7 los agentes Maq, Ben y Rob son quienes tienen información acerca de los objetos $\{o1,\ o2,\ o3\}$. Por otro lado, a partir también del instante 7 se tiene información de que los agentes Rob, Rod y Maq conocen los objetos $\{o1,\ o2,\ o4\}$:

- **Happens**(ConformanEquipo(Equipo1([Maq, Ben, Rob])),s).
- **Happens**(conformanEquipo(Equipo2([Rob, Rod, Maq])),u).

Notemos que aquí el ejemplo supone un corte hecho en el instante 7, para inspeccionar en eso momento qué agentes conocen qué. Bajo cortes previos, otros equipos podrían haberse también conformado, o bien, si el tiempo sigue fluyendo, es posible que algunos miembros abandonen sus equipos (bajo algún criterio) o bien algunos equipos desaparezcan como tales.

Aunque el corte lo hemos realizado a partir del instante 7, notemos que el criterio de información puede servirnos para formar equipos en instantes previos. Por ejemplo, como es mostrado en el diagrama 4, para el instante t_3 se sabe que Rod y Ben conocen el objeto o_2, y esto puede ya servir de guía para formar un equipo. Similarmente, para el instante t_4, Maq y Rob conocen el objeto o_1, y este también puede ser de guía para formar otro equipo.

Dada una propiedad de pertenencia (como parte de un vector de interacción), para tiempos $s > 7$ y $u > 7$ (suponiendo una ley inercial tipo 1) Equipo 1 y Equipo 2 son equipos existentes, y cualquiera de éstos equipos podrán realizar tareas colaborativa que involucren los objetos conocidos. Notemos que un fluente de pertenencia a un equipo es aplicable a un agente y equipo dados, y puede cambiar de falso a verdadero o de verdadero a falso con el tiempo.

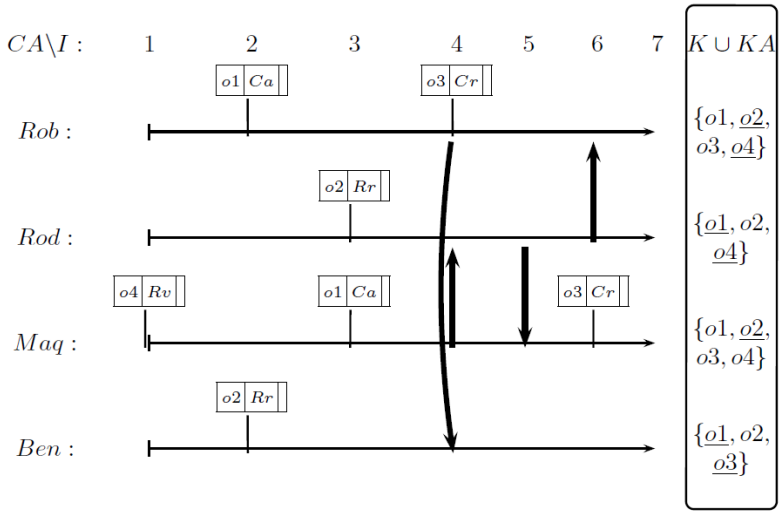

Fig. 4. Diagrama de temporalidad con un agente agregado, ilustrando el punto de conformación de equipos de acuerdo con el conocimiento adquirido por cada agente a través del tiempo. En el rectángulo a la derecha mostramos qué información tiene cada agente en el instante 7.

Ahora, para un tiempo t superior al instante 7, y de seguirse un criterio simple de intersección sobre los conjuntos de objectos conocidos, tendríamos:

– **HoldsAt**(Equipo1([Maq,Ben,Rob]),t).

Lo que implica que Maq, Ben y Rob son quienes conocen $\{o_1, o_2, o_3\}$, y:

– **HoldsAt**(Equipo2([Rod,Rob,Maq]),t).

Lo que implica que Rod, Rob y Maq son quienes conocen $\{o_1, o_2, o_4\}$. A partir de este instante t, los agentes Maq, Rod, Rob y Ben pueden, como ejemplos de decisiones y acciones:

a) Rechazar membresías, bajo algunas condiciones.

b) Optar por salir de un equipo (notemos que sería motivo de incorrección el salir de un equipo sin estar como miembro).

c) Conformar coaliciones, al conjuntar dos o más equipos.

d) Heredar las condiciones bajo las cuales los equipos son útiles (por ejemplo, si una tarea que requería un equipo ya fue hecha, es posible que el equipo carezca de sentido su existir).

Otras narrativas individuales pueden obtenerse de forma similar del diagrama de la Figura 4. De la unión de narrativas individuales surgirían narrativas grupales, mismas que pueden dar indicios más generales que las individuales para analizar el sistema, y así para el caso posible de una narrativa global.

Cabe destacar que una forma de seguridad (o de confiabilidad) del desarrollo (runtime) del sistema sería el cotejar (por algunos agentes, o todos) conocimientos acerca de los objetos y sus atributos, así como posibles marcas de tiempo (timestamps) para mostrar la consistencia de la información obtenida durante el tiempo transcurrido hasta un instante dado t_m, con $m > 1$. Otra índole de información tal como la ubicación de los objetos sería útil también, dependiendo la aplicación.

En la Figura 3, en la columna indicada con K, se ha recabado el total de objetos conocidos a partir del instante 7; se han señalado en los conjuntos de objetos conocidos aquellos que fueron conocidos por experiencia propia y aquellos otros conocidos por comunicación (indicados por nombres de objetos subrayados). Supongamos que HallazgoObjetos es un atributo que es visible desde las interfaces de los componentes autónomos del sistema. Para el instante 7, como hemos visto, es posible formar equipos que están determinados por aquellos agentes que conocen algunos objetos, asi logrando que el cálculo de eventos ayude en la toma de decisiones para conformar equipos [4, 15].

5. Conclusiones

Se ha afirmado en este trabajo que SCEL sí brinda un apoyo teórico a la formación de equipos y en principio, a coaliciones (conjuntos de equipos). Además, también se ha mostrado que a través de la noción de temporalidad se puede coadyuvar a la dinámica de equipos de agentes. A través de la definición de interfaces, SCEL propone un mecanismo de formación de equipos conforme a las necesidades del sistema que se presentan. La literatura indica la importancia de una adecuada comunicación entre agentes [19] para una acertada gestión de equipos.

Para una formulación de una jerarquización (no existente en nuestra propuesta) que sea útil en el consenso de datos, ver [13]. En la parte colaborativa, sería posible tratar el tema de acuerdo con la teoría de juegos, brindando recompensas a los agentes más proactivos y participativos, estableciendo esquemas de negociación, y en la parte competitiva o de adversarios, se esperaría integrar equipos defensivos o de ataque.

Referencias

1. De-Nicola, R., Latella, D., Lafuente, A. L., Loreti, M., Margheri, A., Massink, M., Morichetta, A., Pugliese, R., Tiezzi, F., Vandin, A.: The SCEL language: Design, implementation, verification (2015) doi: 10.1007/978-3-319-16310-9_1
2. Doets, K.: From logic to logic programming. The MIT Press (1994)
3. Dunin-Keplicz, B., Verbrugge, R.: Awareness as a vital ingredient of teamwork. In: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1017–1024 (2006) doi: 10.1145/1160633.1160815
4. Dunin-Keplicz, B., Verbrugge, R.: Teamwork in multi-agent systems: A formal approach. John Wiley and Sons (2010)
5. Hernández-Gutiérrez, M., Sánchez-Soto, E.: Actividad de agentes robóticos regulada a través de información de temporalidad vía un cálculo lógico de eventos. Research in Computing Science, vol. 6, no. 152, pp. 7–20 (2023)
6. Kaminka, G. A., Frenkel, I.: Towards flexible teamwork in behavior-based robots. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1355–1356 (2005) doi: 10.1145/1082473.1082770

7. Kowalski, R.: Logic for Problem Solving. North–Holland (1979)
8. Kowalski, R., Sergot, M.: A logic-based calculus of events. New Generation Computing, vol. 4, pp. 67–95 (1985) doi: 10.1007/BF03037383
9. Krzysztof, R. A.: Logic programming. Formal methods and semantics, Elsevier, vol. B, chapter 10, pp. 493–574 (1990)
10. Krzysztof, R. A.: From Logic Programming to Prolog. Prentice Hall (1997)
11. Kshemkalyani, A. D., Singhal, M.: Distributed computing: Principles, algorithms, and systems. Cambridge University Press (2008)
12. Kulyk, O., van-der-Veer, G., van-Dijk, B.: Situational awareness support to enhance teamwork in collaborative environments. In: Proceedings of the 15th European Conference on Cognitive Ergonomics: The Ergonomics of Cool Interaction, Association for Computing Machinery, pp. 1–5 (2008) doi: 10.1145/1473018.1473025 https://doi.org/10.1145/1473018.1473025
13. Luotsinen, L. J., Bölöni, L.: Role-based teamwork activity recognition in observations of embodied agent actions. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, vol. 2, pp. 567–574 (2008)
14. Mueller, E. T.: Chapter 17: Event calculus. Foundations of Artificial Intelligence, vol. 3, pp. 671–708 (2008)
15. 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
16. ROS: ROS - robotic operating system (2024) www.ros.org/
17. 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
18. Weyns, D., Schumacher, M., Ricci, A., Viroli, M., Holvoet, T.: Environments in multiagent systems. The Knowledge Engineering Review, vol. 20, no. 2, pp. 127–141 (2005) doi: 10.1017/S0269888905000457
19. Zhang, Y., Volz, R. A., Loerger, T. R., Yen, J.: A decision-theoretic approach for designing proactive communication in multi-agent teamwork. In: Proceedings of the ACM Symposium on Applied Computing, Association for Computing Machinery, pp. 64–71 (2004) doi: 10.1145/967900.967917