

Coordinación en un Sistema de Gestión de Eventos Autónomo basado en Agentes de Software

Lorena Bearzotti, Enrique Salomone, Omar Chiotti
INGAR – CONICET, Avellaneda 3657, Santa Fe, Argentina, 3000
GIDSATD – UTN FRSF, Lavaise 610, Santa Fe, Argentina, 3000

Resumen

En este trabajo se propone la especificación de un mecanismo de coordinación de agentes, los cuales definen un sistema de gestión de eventos autónomo. Esta especificación se lleva a cabo mediante la definición de los procesos de negociación que se establecen entre los distintos agentes del sistema, los que tienen como objetivo elaborar una solución colaborativa que de respuestas a eventos del sistema que perturben los planes de los agentes. Un aspecto clave de la propuesta es el proponer un mecanismo distribuido para lograr la coordinación.

Keywords: Agentes de Software, gestión de eventos, sistema soporte, colaboración, coordinación, protocolo de interacción.

1. Introducción

En los últimos años, el software orientado a agentes ha experimentado un creciente interés en distintos campos, el cual se ha manifestado en el desarrollo de metodologías de análisis y diseño, así como también de plataformas de desarrollo de sistemas de agentes. (Bussman et al 2001, DeLoach et al 2001, Jennings et al 2001, Sycara 1998, Wooldridge et al 2000, Zambonelli et al 2000-2001)

Los agentes han sido aceptados y se tiende cada vez más a uso en ciertos ambientes, debido a que por su naturaleza permiten describir con ciertos dominios de aplicación, que con otras metodologías sería complejo. En determinados dominios, los agentes interactúan entre sí para lograr un objetivo común, sin embargo los agentes tienen su propia naturaleza y sus propios objetivos e intereses; en consecuencia existe la posibilidad de que surjan conflictos, los cuales pueden afectar el cumplimiento de los objetivos del sistema. Estos problemas requieren que exista un mecanismo adicional denominado coordinación, el cual emplea procesos de negociación para realizarse. (Paurobally et al 2003,)

La coordinación en un sistema multiagente (MAS) es un problema para el cuál aún no se ha hallado una solución específica. La necesidad de la coordinación entre los agentes surge debido a que éstos están sujetos a mutuas restricciones, las cuales determinan los estados internos de los agentes. Cuando un agente determina una acción a realizarse toma esta decisión en base a su Estado Interno. Los estados por los que pasa un MAS depende del scheduling de las acciones ejecutadas por cada agente, esto hace que los agentes deban coordinarse entre sí. (Dumas et al 2002, Bergenti)

En un MAS la negociación es un proceso que permite la solución de conflictos entre agentes por medio de la comunicación entre estos, buscando una solución global del conflicto lo menos comprometida posible para los agentes afectados. (Santacruz et al 96).

La coordinación implica comunicación, por lo que para que los agentes estén coordinados es necesario que puedan comunicarse entre ellos. Los lenguajes de comunicación de Agentes (ACL) tales como KQML y FIPA ACL, especifican las reglas de comunicación entre los agentes. Para la problemática de la comunicación entre agentes se ha llegado a niveles de estandarización, no así para el caso de la coordinación.

La idea subyacente tras el concepto de coordinación es que los agentes de un MAS puedan organizarse para tomar las decisiones en conjunto ante aquellas situaciones problemáticas que se

puedan plantear entre dos o más agentes cuando los intereses de uno se ven afectados por las acciones de otro.

En este trabajo sugerimos un concepto de coordinación en un sistema multiagente para la gestión de eventos en la cadena de suministro como el conjunto de los distintos procesos de negociación que en ella suceden. Por lo tanto describir la coordinación en este ambiente es equivalente a formalizar los distintos procesos de negociación que se presentan en el sistema. El sistema basado en agentes para la gestión de eventos en la cadena de suministro, para el que hemos efectuado el análisis y el diseño, es un sistema heterogéneo y distribuido en su naturaleza por lo que el problema de la coordinación de sus agentes tiene una mayor complejidad si se desea respetar la naturaleza del problema. En este tipo de ambiente la coordinación es descentralizada y esto es así porque no existe ningún agente en el modelo que asuma el rol de agente coordinador encargado de resolver los conflictos que pudieran aparecer entre los distintos agentes de la red.

En la Sección 2 se efectúa una presentación del problema de la gestión de eventos en general y se analiza un caso en particular, la gestión de eventos en la cadena de suministro, ya que la especificación de los procesos de negocios se realizará para ese ejemplo. En la sección 3 se analizará el problema de la coordinación en MAS. En la sección 4 se describirá en detalle nuestra propuesta para efectuar la coordinación en sistemas de gestión de eventos de la cadena de suministro. Por último en la sección 5 se enuncian las conclusiones de este trabajo y las tareas de investigación a seguir.

2. El problema de la gestión de eventos

Algunos sistemas de información tienen como funcionalidad principal soportar las actividades de planificación. Una vez que un plan ha sido definido, la siguiente actividad es monitorear que dicho plan se esté ejecutando tal y como es esperado con la finalidad de detectar excepciones (son producidas por eventos inesperados) que requieran de acciones correctivas. El proceso de la gestión de eventos consiste en el monitoreo de eventos para detectar excepciones al plan que se está ejecutando, si una excepción ocurre el sistema entonces realiza las acciones de control.

Como la ocurrencia de eventos inesperados es algo natural en el mundo real, los planificadores definen sus planes con holguras, lo cual le brinda robustez y flexibilidad para adaptarse a las nuevas condiciones. Es decir, los planes incluyen márgenes de seguridad los cuales pueden ser usados para mitigar el efecto de las excepciones.

Un sistema de gestión de eventos puede ser implementado con diferentes grados de automatización. En un primer nivel podemos plantear como EMS (Event Management System) un sistema manual de monitoreo del proceso en ejecución, en esta propuesta el tomador de decisión detecta una excepción y determina el curso de acción en base a su experiencia y conocimiento. Incrementando el nivel de automatización es posible plantear un EMS que funcione como un sistema de alarma, en este caso se automatiza la detección de los eventos que causan excepciones. Un nivel más alto de automatización puede ser planteado cuando al EMS se le permite cierta autonomía para definir acciones de control trabajando con los márgenes de seguridad especificados en los planes.

El problema de la gestión de eventos es naturalmente un problema de control. Es posible identificar los 3 tipos de variables: variable observada (es la variable que se observa con la finalidad de detectar eventos), variable de estado (esta variable define el punto de control, asociado con esta variable tenemos el plan de estado, al que definimos como una sucesión de valores que la variable de estado debe tomar en el horizonte de planificación) y variable de decisión (esta es la variable independiente cuyo valor puede ser ajustado para mitigar los efectos de una excepción con el propósito de regresar el sistema a los objetivos especificados)

Nuestra propuesta para EMS consiste en modelarlo como una red de puntos de control que representan las relaciones entre sus variables de estado. Podemos expresar lo mismo de la siguiente manera: El EMS es modelado como un conjunto de subsistemas interrelacionados, cada uno de

ellos representando un punto de control que tiene un plan y sus márgenes de seguridad definidos por el Sistema de Planificación. Cada subsistema tiene que monitorear sus variables observadas para detectar la ocurrencia de un evento y analizarlo para detectar si este produce una excepción. Cuando una excepción es detectada, el subsistema puede recurrir a los márgenes de seguridad del plan para definir las acciones de control para mitigar el efecto de esa excepción. Si las acciones de control no pueden ser implementadas solo con los límites definidos por los márgenes de seguridad de su plan, el subsistema puede negociar con los subsistemas relacionados. Si la negociación no encuentra una solución, entonces el subsistema que detectó el evento que ocasionó la excepción notifica de la situación al sistema de planificación.

Un evento para un subsistema puede ser, interno (si es detectado por el subsistema), externo (si proviene de otro subsistema) o bien simulado (no son eventos reales sino generados para evaluar posibles escenarios).

Los subsistemas del EMS pueden ser modelados por agentes autónomos o semiautónomos donde el conocimiento y las habilidades se encuentran distribuidos localmente y la comunicación es efectuada mediante el intercambio de mensajes. Desde esta perspectiva EMS puede ser visto como una Agencia ya que es una máquina desarrollada para soportar la gestión de eventos mediante la cooperación entre los agentes.

2.1. Ejemplo de aplicación: Gestión de Eventos en la Cadena de Suministro

A partir del EMS es posible derivar frameworks especializados, un ejemplo de esto es la especialización del EMS en el dominio de las cadenas de suministros.

Definimos a la cadena de suministro como una red de negocios autónomos o semiautónomas, colectivamente responsables de la obtención, producción y distribución de una o más familias de productos relacionadas. (Swaminathan et al, 1996). Entendemos como SCM (Supply Chain Management) al conjunto de propuestas utilizadas para integrar eficientemente a proveedores, productores, distribuidores, almacenes, tal que el producto sea producido y distribuido en la cantidad y tiempo correcto, minimizando los costos y cumpliendo con el nivel de servicio requerido. (Simchi Levi et al 2000). Dentro de SCM es posible destacar SCEM (Supply Chain Event Management) el cual es un componente que permite cerrar la brecha entre los procesos de planificación y ejecución.

Hemos definido a EMS como una red de puntos de control. En el contexto de la cadena de suministro proponemos que esta red esté compuesta por puntos de control de inventario, los cuales están interconectados por procesos de suministros, los que para su funcionamiento requieren recursos.

Así la SCEM como una red de puntos de control de inventario que están vinculados entre sí por puntos de control de procesos de suministro. Los procesos de suministro usan los recursos, los cuales son modelados como puntos de control de recursos. Entonces en este modelo definimos tres tipos de puntos de control:

SKU (Storage Keeping Unit): para la gestión de eventos asociado a los inventarios. Está definido por tres atributos principales: material, envase, ubicación.

RKU (Resource Keeping Unit): para la gestión de eventos asociados a recursos. En el sistema existirá un RKU por cada recurso de la cadena de suministro. Sus atributos representan la disponibilidad del recurso que representa.

SP (Supply Process): para la gestión de eventos asociado con un proceso de suministro. Un SP representa la transición de uno o más SKUs a otro u otros SKUs. Tenemos tres tipos de transiciones básicas, las cuales se pueden combinar para expresar transiciones compuestas. Estas transiciones básicas son: cambio de material, cambio de envase y cambio de lugar.

Aplicando la metodología GAIA de análisis y diseño orientado a agentes (Woolridge, 2000) determinamos la existencia de 3 tipos de agentes base, asociados con cada uno de los puntos de control. Así tendremos el Agente SKU, el agente RKU y el agente SP.

El **agente SKU** es responsable de la gestión de eventos de un punto de control específico: el inventario; está definido por 3 atributos: *producto*, *envase* y *lugar*. Cada SKU cuenta con una *lista de entrada/salida* que tiene la siguiente estructura: [I/O - SP - Start Time - Rate - Duration]. Además de esta lista, el SKU recibe los márgenes de seguridad, estos márgenes son empleados en la evaluación de eventos para determinar si ese evento conduce a una excepción. El **plan de estado** asociado a cada SKU es elaborado mediante la combinación de la lista de entrada/salida y los márgenes de seguridad. La variable monitoreada para el SKU es el valor actual del Inventario, cualquier evento interno significará una variación de ese valor, entonces el SKU detectará eventos que ocasionen variaciones en su inventario. Las variables controladas son para cada operación de I/O: ST, Rate, Duration. Estas variables son las que intervienen en la negociación, es decir sobre ellas se hacen las variaciones con la finalidad de distribuir las perturbaciones.

El **Agente RKU** monitorea la disponibilidad del recurso (todos los recursos existentes exceptuando a los materiales y a los energéticos) que representa. Existe un agente RKU para cada recurso relevante de la cadena de suministro. Cada RKU cuenta con una Agenda de Uso, a partir de la cual es posible elaborar un perfil de Carga de ese recurso. Se le definen los siguientes atributos: [SP - ST - Duration - Capacity]. La variable de Estado de un RKU es el Estado de disponibilidad de un Recurso. La variable de control, está definida por los períodos de tiempo existentes entre dos requerimientos de uso definidos en la Agenda, esto permite adelantar o atrasar una solicitud de uso de un recurso (no está permitido un cambio de ejecución de los pedidos). Las variables monitoreadas dependen del tipo de recurso, siendo indicadores de las posibles variaciones en la disponibilidad del recurso.

El **Agente SP** controla la extensión del proceso detectando variaciones en tiempo y en cantidad. Cuenta con un plan de actividades en el que se detallan las operaciones especificando el tiempo de inicio, la duración y BOM (Bill of Material). En la BOM se describe que tipo de operación es (si es de entrada o de salida), que SKUs intervienen y la cantidad del producto que entra o sale. Las variables de estado y la variable monitoreada es la extensión del proceso. Las variables controladas son: el tiempo de inicio, el tiempo de finalización, la cantidad ingresada/egreso de cada SKU.

En el siguiente gráfico vemos la comunidad de los agentes y los mensajes que ellos se envían, los cuales son los iniciadores de los procesos de negociación.

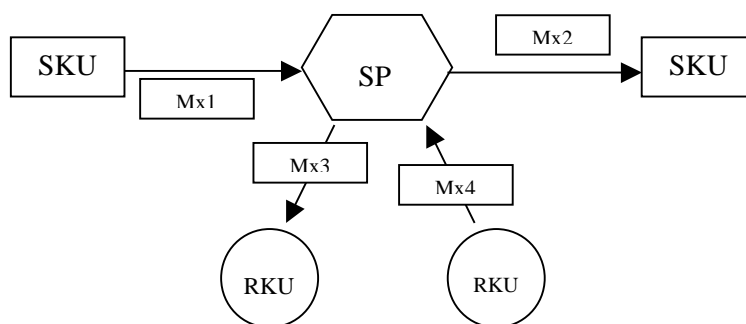


Figura 1: La comunidad de los Agentes

Mx1 (SKU-SP): en este mensaje un Agente SKU le notifica a un Agente SP de la necesidad de realizar un cambio en una o varias de las órdenes asociadas con él, esos cambios son de tiempo y/o de cantidad.

Mx2 (SP - SKU): Este mensaje es enviado por el agente SP al SKU indicándole que solicita un cambio en tiempo y/o cantidad de la orden o de las órdenes.

Mx3 (SP - RKU): el agente SP envía este mensaje a un RKU cuando solicita una variación en el uso del recurso que el RKU representa.

Mx4 (RKU-SP): el RKU notifica a los SPs, la indisponibilidad del recurso y el período de tiempo.

3. El problema de coordinación en Sistemas Multiagentes

Un agente puede ser considerado como una unidad en el universo del problema que tiene la capacidad de comunicarse con los otros agentes, interactuar con ellos por medio de protocolos y tener un comportamiento distintivo y bien definido.

En el ambiente de agentes, definimos a un sistema multiagente como un conjunto de agentes que interactúan entre sí para lograr un objetivo común, y lo logran cooperando entre sí. Sin embargo, entre estos agentes pueden existir conflictos que afecten la satisfacción de los objetivos planteados para el sistema, los cuales para su resolución necesitan de la coordinación. Coordinar es la acción de disponer las cosas metódicamente reuniendo esfuerzos para lograr un objetivo. La coordinación requiere que los agentes posean cierta información de las decisiones tomadas por los otros agentes y por esto la comunicación toma un papel primordial en el proceso de coordinación. (Santacruz et al 96)

Entonces, la coordinación en sistemas multiagentes es necesaria para resolver conflictos entre los agentes, entendemos por conflicto a una situación problemática en la que se contraponen intereses de distintos agentes. Para resolver el conflicto, los agentes hacen uso de los procesos de negociación.

En un proceso de negociación podemos plantear las siguiente etapas. **Inicio de negociación:** aquí se identifican los participantes de la negociación registrándolos e identificándolos. **Ronda de negociación:** interacción e intercambio de información entre los participantes de la negociación. **Terminación de la negociación:** una negociación puede terminar exitosamente, en cuyo caso se envía el resultado de los acuerdos tomados a los participantes para que estos los tengan en cuenta y los utilicen. La otra opción es que la negociación falle, en cuyo caso no hay una solución y el conflicto permanece.

Las propuestas de negociación en el ambiente de MAS pueden clasificarse en dos grandes grupos:

Negociación con moderador: en este esquema podemos observar un control centralizado para el intercambio de información, además una interacción indirecta de los diferentes agentes a través de este moderador, facilita el proceso de interacción entre agentes ya que solo requiere conocer como comunicarse con el moderador y no con todos los agentes del sistema, lo cual en ambientes heterogéneos es de gran utilidad.

Negociación sin moderador: en este esquema tenemos un control descentralizado, donde la comunicación entre los agentes se realiza de forma directa.

Un protocolo de coordinación define las posibles secuencias de mensajes entre agentes en un grupo. Así cuando un agente envía un mensaje a los otros miembros de su comunidad sabe el formato de las posibles respuestas a su mensaje.

El protocolo debe estar claramente especificado y validado, así como también correctamente implementado. Un protocolo debe ser *correcto, no ambiguo, completo y verificable*.

Además para garantizar la terminación de un protocolo en cualquier caso debemos obligar a que este verifique lo siguiente: un plazo máximo (deadline) para que los participantes puedan evaluar las propuestas y decidan si la aceptan o no; tener la posibilidad de que el iniciador cancele una propuesta (porque no se llega a un acuerdo o la situación que dio origen a la discusión haya cambiado) y que siempre se emita un juicio sobre una propuesta, ya sea aceptándola o rechazándola. (Santacruz et al 96, Rebollo et al 2001)

FIPA ha especificado diferentes Protocolos de Interacción de Agentes, de estos por sus características nos interesan el FIPA CNIP (Contract Net Interaction Protocol) y el FIPA ICNP (Iterated Contract Net Protocol). Estos protocolos se utilizan para permitir que un agente iniciador

solicite la ejecución de una tarea a uno o más agentes, la diferencia entre ambos es que el ICNP permite múltiples rondas iterativas de órdenes.

FIPA CNIP está basado en el trabajo de Contract Net Protocol desarrollado por Smith y Davis, (Smith 1980) varía en que esta propuesta incorpora los mensajes de rechazo y de confirmación. Este protocolo se utiliza para permitir que un agente iniciador solicite la ejecución de una tarea a uno o más agentes. Comienza cuando el iniciador hace una llamada para ver qué agentes pueden realizar la tarea. Los agentes receptores pueden rechazar el mensaje, no entenderlo o devolver al iniciador una propuestas con sus precondiciones para realizar la tarea. Cuando el iniciador recibe las propuestas, las evalúa, escoge a un agente (o a varios) para realizar la tarea y envía una aceptación al agente (o agentes) elegido y un rechazo al resto. Cuando los agentes contratados realicen la tarea informan al iniciador. Estos agentes pueden subcontratar la tarea a otros.

En el trabajo de Juhasz se determinó la performance y escalabilidad de CNP, llegando a la conclusión de que el protocolo es dependiente del tamaño del sistema, la carga de los agentes y del deadline que se asigne a cada diálogo.

CNP ha tenido una amplia difusión y aceptación. Sin embargo su aplicabilidad se ve reducida a procesos de negociación con reglas de comportamiento muy estricta donde además los agentes tienen modelos muy simples del mundo y de otros agentes. Otro punto observable es que la negociación no es iterativa. Para solucionar esto, FIPA propone el ICNP, este protocolo permite múltiples rondas iterativas.

ICNP comienza cuando el agente iniciador envía m mensajes llamando a propuestas. De los m agentes que recibieron el mensaje, n son los que responden al mismo, de los cuales j rechazan la propuesta, los k restantes ($k=n-j$) envían sus propuestas bajo las condiciones impuestas por el agente iniciador. Cuando el agente iniciador recibe las k propuestas puede decidir que se encuentra en la iteración final y acepta p de esas k propuestas donde p pertenece al intervalo $[0,k]$ y enviando un mensaje de rechazo a los $k-p$ otros participantes. Otra situación que se puede dar es que el agente decida iterar el proceso, enviando un nuevo llamado a propuestas a los l participantes que enviaron una propuesta, pudiendo a su vez rechazar a los $k-l$ otros participantes.

El objetivo que persigue este protocolo es que el agente iniciador por medio de propuestas y contrapropuestas puede determinar la mejor solución al problema que plantea.

Como este protocolo tiene un ciclo entre los agentes es necesario determinar como éste terminará. Las condiciones de terminación son: el iniciador rechaza todas las propuestas y no genera ningún nuevo llamado a propuesta; el iniciador acepta una o más propuestas; los participantes (todos) rechazan el llamado a propuestas.

Este protocolo contempla el manejo de dos excepciones, las cuales son cancelación de una propuesta y el no entendimiento de un mensaje.

4. El problema de coordinación en el sistema de gestión de eventos de la cadena de suministro.

El sistema basado en agentes para la gestión de eventos en la cadena de suministro es un sistema heterogéneo y distribuido en su naturaleza por lo que el problema de la coordinación de sus agentes tiene una mayor complejidad si se desea respetar la naturaleza del problema.

La idea de una coordinación centralizada, con la existencia de un Agente Coordinador cuya función es la de resolver conflictos entre los otros tipos de agentes es más fácil de implementar frente a una solución descentralizada como la que proponemos. Sin embargo en el modelo propuesto la coordinación es descentralizada debido a que no existe un único agente coordinador. El punto aquí es determinar quien realiza la coordinación. Nuestra propuesta consiste en que *el rol de agente coordinador sea asumido por el agente que inicia el proceso de negociación al detectar un evento interno*. Terminado el proceso de negociación, el Agente que lo inició deja de ser Coordinador.

Consideremos el siguiente ejemplo, sea SKU1 un agente el cual en el tiempo ti detecta un evento interno, luego de analizar este evento el SKU1 concluye que para resolver la situación ocasionada por el evento necesita iniciar un proceso de negociación con algunos de los SP con los que se encuentra vinculado. Por lo tanto inicia un proceso de negociación se convierte en el agente coordinador correspondiente a ese evento.

Sin embargo en el tiempo ti n -eventos internos se pueden producir, los cuales son detectados por n -agentes, de los n -eventos j de ellos pueden ocasionar la necesidad de iniciar un proceso de negociación, entonces en el tiempo $ti+Dt$ pueden coexistir j -Agentes coordinadores, lo cual es equivalente a afirmar que en simultáneo podemos tener j -procesos de negociación.

Por lo tanto si definimos a la coordinación como un conjunto de procesos de negociación que tienen como objetivo resolver los conflictos entre los agentes planteados por la ocurrencia de excepciones, lo que tenemos que determinar es lo que negocian los agentes entre sí, es decir cuales son los parámetros de la negociación y en que magnitud estos parámetros pueden ser variados. La magnitud de la variación está dada por los márgenes de seguridad definidos para cada uno de los agentes, es decir no se puede variar un parámetro tal que el valor de la variable de estado sea llevado a una región infactible (fuera de los márgenes de seguridad).

4.1. Definición de los procesos de negociación

El proceso de negociación completo puede ser visto como la unión de subprocesos de negociación entre solo dos agentes; teniendo 3 tipos de agentes que tienen la habilidad de negociar entre sí para lograr el fin común, y sabiendo que agentes de un mismo tipo no pueden estar relacionados, planteamos los siguientes subprocesos: SKU – SP, SP- SKU, SKU-SP, SP-RKU.

Este concepto de negociación visto como la unión de subprocesos entre agentes vecinos ha sido planteada en el trabajo de Santacruz (96) y otros en el cual se proponen dos modelos, donde uno de ellos plantea la negociación de un agente con sus vecinos (modelo del Panal).

Entonces desde esta perspectiva un proceso de negociación será la unión de subprocesos de negociación entre dos agentes. Por lo tanto la especificación del proceso de negociación completo consiste en definir y caracterizar estos subprocesos y analizar como el agente reacciona y trabaja cuando está cumpliendo el rol de coordinador y cuando es un mero eslabón de la cadena.

Analicemos un proceso de negociación genérico que hemos propuesto para la negociación de los agentes de nuestra agencia, el cual es una extensión del protocolo FIPA ICNP (Ver Figura 2). Los agentes se encuentran monitoreando cada uno un punto de control de interés y reciben eventos que suceden relacionados con el punto de control que cada agente gestiona. Por cada evento el agente determina a que situación lo lleva, es decir evalúa como ese evento impacta en su Plan de Estado. La evaluación tiene dos estados posibles, que el evento conduzca a una excepción (es decir que escape de los márgenes de seguridad) o bien que el evento conduzca a la variable de estado del agente fuera del valor esperado pero dentro de los márgenes de seguridad, es decir que puede ser absorbido por el propio agente sin escaparse de los márgenes definidos.

Analicemos el caso para un evento interno, independientemente de la situación a la cual lo lleve el evento interno que acontece, el Agente coordinador generará una serie de propuestas o posibles soluciones contemplando a los agentes vecinos que potencialmente pueden ser afectados, estas soluciones las llamamos *EspacioSolución*, en función de esas posibles soluciones el Agente inicia un proceso de negociación con los agentes que conoce de su mundo. El inicio está marcado cuando el Agente Coordinador envía un mensaje a sus agentes vecinos proponiéndoles una posible solución (es decir modificaciones en algunos de los parámetros definidos para ese clase de relación entre agentes). De los n vecinos el coordinador puede elegir enviar el mensaje iniciador a m de ellos donde $m \leq n$. De los m agentes vecinos que recibieron el mensaje iniciador, p de ellos responden (con $p \leq m$) pudiendo ser su respuesta: rechaza la solución propuesta por el iniciador, no

comprende el mensaje, acepta la solución o bien en función de la propuesta original le contestan con una contrapropuesta.

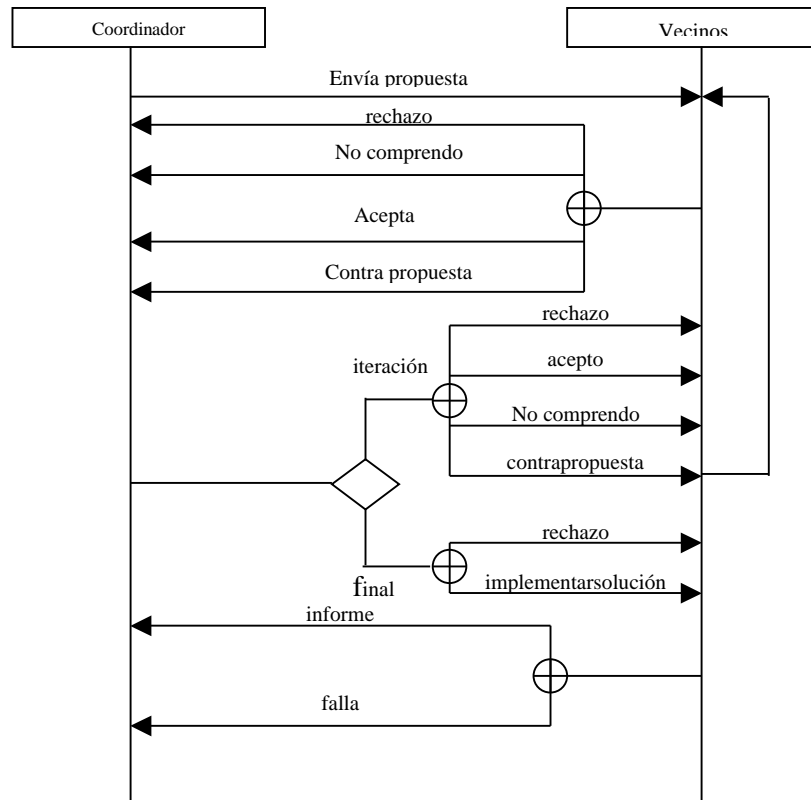


Figura 2: Diagrama del protocolo genérico

Para los mensajes de rechazo y no comprensión el Agente Coordinador elimina a esos vecinos de su proceso de negociación; la finalidad de esto es reducir el número de agentes en la negociación y ser un punto de finalización del proceso, ya que si se produce la eliminación sucesiva de los agentes vecinos y no queda ninguna agente con quien negociar el proceso termina.

En caso de que el agente vecino acepte la propuesta, el agente coordinador guarda la propuesta y su aceptación como solución, con su costo asociado. Si recibe un mensaje de contrapropuesta, el agente coordinador debe evaluarla, para determinar si esta contrapropuesta pertenece al EspacioSolución, si pertenece al EspacioSolución esa propuesta es rechazada debido a que ya ha sido contemplada por el agente coordinador, sino pertenece a este conjunto, el Coordinador evaluará la propuesta de la misma forma que lo hace con sus propias propuestas. Las posibles respuestas a una contrapropuesta son: rechazo, no comprende, acepta y contrapropuesta, y así iterando. Cuando se llega al estado final las respuestas son rechazo o implementación de una solución. Con el mensaje de rechazo se le indica a los agentes vecinos que el compromiso de mantener su estado se ve liberado, con el de solución se le confirma la implementación de la misma.

Luego de implementar una solución, el o los vecinos deben notificar si la implementación fue exitosa o bien si hubo fallas en la implementación de la misma. En caso de problemas, el agente coordinador tiene dos opciones, iniciar el proceso de negociación de nuevo o bien asumir que no hay solución para el evento que disparó el proceso de negociación.

El manejo de excepciones se hace con los mensajes de falla en el caso de implementación de una solución y de no comprensión de un mensaje. Además de estas consideraciones, el agente coordinador maneja el deadline, el cual comienza a correr a partir del momento en el que se envía la propuesta, pasado el deadline, si no se reciben respuestas, el agente coordinador asume que no hay

solución y que la negociación ha fallado. Por lo tanto, dependiendo si se está en presencia de una excepción el Agente coordinador notificará al Sistema de Planificación que no puede resolver la excepción, si en cambio el evento lleva al sistema a una zona de márgenes de seguridad, el agente coordinador absorbe el desvío. Además a partir del vencimiento del deadline todos los mensajes asociados a ese proceso de negociación son rechazados cuando le lleguen y todos los mensajes asociados son cancelados.

Comencemos a analizar las características de los subtipos de protocolos de negociación. En el proceso de negociación SKU-SP, un SKU es el que toma el rol de agente iniciador de la negociación. Podemos encontrar dos casos posibles, el primero es cuando el SKU detecta un evento interno, por lo tanto se transforma en el Agente Coordinador del proceso de negociación.. El segundo caso es cuando el SKU forma parte un proceso iniciado por otro Agente. Un SKU solo está enlazado con SPs, a los cuales él los conoce a partir de su lista de Entrada/Salida. Entonces dado el evento, evalúa a cuales SP puede afectar más directamente y con ese conocimiento genera una lista de posibles soluciones al evento modificando los parámetros de tiempo y cantidad asociados a las órdenes de esos SP seleccionados. Las propuestas de solución son elaboradas mediante un algoritmo, escogiendo solamente a aquellas que no conducen a escenarios no factibles. Cada una de estas propuestas son almacenadas en su Base de Conocimiento de forma tal que se puede efectuar un seguimiento de las mismas. Una vez elaborado el EspacioSolución, el siguiente paso que efectúa el SKU es enviar mensajes a los SP elegidos para participar en el proceso de negociación con las propuestas, un mensaje a cada SP por cada propuesta de solución. Dentro del deadline definido para la negociación el SKU comienza a recibir las respuestas de los SP, en caso de que pasado el deadline no haya recibido ninguna respuesta tiene que decidir si reinicia la negociación o asume que los SP no desean negociar. Los pasos sucesivos son determinados por el protocolo genérico. El ciclo de negociación termina cuando las contrapropuestas no mejoran las soluciones existentes de acuerdo al costo que cada una de ellas tiene asociadas.

El proceso de negociación SP- SKU es iniciado por el SP, si el evento es interno se convierte en el Agente Coordinador, por otro lado si el evento es externo significa que la perturbación le llega por medio de un mensaje proveniente de uno de sus SKUs relacionados o bien desde alguno de los RKUs. El SP negocia con los SKUs variaciones en el tiempo y la cantidad de sus órdenes asociadas, esas variaciones son cambios en sus variables controladas, las cuales son de su lista de actividades: tiempo_inicio, tiempo_final y cantidad. Entonces ante un evento que lo aleje de su Plan de estado, el SP inicia un proceso de negociación con los SKU que se vean total o parcialmente afectados por el mismo.

Cuando el proceso de negociación es iniciado por el SP con un RKU, le está solicitando a este un adelantamiento, atraso o extensión en el uso de un recurso. El RKU evalúa las propuestas sobre su agenda de uso y determina si esos desplazamientos son posibles usando sus márgenes, si es posible envía un mensaje aceptando la propuesta del SP. En caso de no serlo debido a que los márgenes no son suficientes rechaza la propuesta del SP o bien le hace contrapropuestas de acuerdo a sus capacidades. El ciclo sigue hasta que se ponen de acuerdo en un nuevo valor o bien cuando todas las propuestas son rechazadas.

Un proceso de negociación iniciado por un RKU tiene como finalidad notificar a los SPs sobre la no disponibilidad de un recurso. A diferencia de los previos, este es siempre iniciado por un evento interno detectado por el RKU que manifiesta que un recurso por un período de tiempo no estará disponible, entonces el RKU envía un mensaje notificando de esta nueva situación a todos los SPs que se ven afectados proponiéndoles variaciones si estas son posibles. Éstos responden aceptando la proposición o con una contrapropuesta. Con las contrapropuestas recibidas el RKU elabora una lista y decide cual de ellas es la más conveniente para implementar como solución, si eso es posible.

3. Conclusiones

La coordinación en un sistema multiagente es un aspecto de particular importancia en el análisis y diseño de sistemas multiagentes, ya que los agentes en la mayoría de los casos necesitan interactuar unos con otros, y esa interacción se basa en la comunicación Interagente.

Existen determinados ambientes esa interacción de los agentes persigue la realización de un fin global o del sistema, recordemos que los agentes son entidades autónomas con sus propios objetivos e intereses y que por ende esa diferencia de objetivos puede ocasionar conflictos. En consecuencia los agentes que interactúan para lograr un fin del sistema necesitan un mecanismo de negociación o coordinación que le permitan resolver los conflictos que pueden surgir ante diferencias de intereses. La finalidad de este trabajo ha sido presentar un mecanismo de coordinación en el ámbito de un sistema multiagente para la gestión de eventos en la cadena de suministro. En este contexto hemos modelado la cadena de suministro como una red de agentes (RKU, SKU, SP), cada uno de ellos realiza la gestión de eventos en cada punto de control (recurso, producto, proceso), a su vez cooperan entre sí con la finalidad de mantener a la cadena de suministro dentro de los planes que se han definido. La cooperación es llevada a cabo por medio de la especificación de los protocolos de negociación. Definimos un protocolo base y luego lo especificamos acorde al tipo de relación que se está presentando (RKU-SP, SP-SKU). Este protocolo toma como base el FIPA Iterated Contract Net Interaction Protocol)

Como trabajos futuros relacionados con este que presentamos, podemos mencionar la formalización del protocolo, el objetivo de formalizar el protocolo de interacción es el de subsanar la semiformalidad, vaguedad que traen consigo un desarrollo intuitivo. Debido a la importancia que revisten los protocolos de interacción en un ambiente multiagente este debe ser claramente especificado, validado y correctamente implementado. A causa de esto como futuro trabajo se prevee el estudio de herramientas de implementación y especificación formal de protocolos, el algoritmo de generación del espacio solución, definición de la función de costo, así como los métodos para verificar y validar el protocolo de interacción. [Paurobally et al, 2003][Dumas et al, 2002]

Referencias

- Paurobally, S., Cunningham J., Jennings N. Developing Agent Interaction Protocols Using Graphical and Logical Methodologies. *In Workshop on programming MAS, AAMAS, 2003.*
- Dumas M., Governatori G., Hofstede A. A formal approach to negotiating agents development. *Electronic Commerce Research and Application 1(2002) (193-207).*
- Rebollo M., Carrascosa C., Julián V. Un protocolo para co-edición argumentativa entre agentes. *Actas de V Congreso ISKO-España. Alcalá de Henares. ISBN 84-8138-435-6 2001.*
- Juhasz Z., Paul P. Scalability Analysis of Contract Net Protocol.
- Bergenti F., Poggi A., Somacher M. A Contract Decommitment Protocol for Automated Negotiation in Time Variant Environments
- Santacruz F., Casanova R., Quintana F. Negociación Distribuida y Semi-Centralizada en Sistemas Multiagentes: dos modelos propuestos. *In CLEI Panel 96, Bogotá.*
- Foundation for intelligent physical agents, fipa, agent communication language. In <http://www.fipa.org>
- Monostori L, E. Szelke, B. Kadar. Management of changes and disturbances in manufacturing systems. *Annual Reviews in Control 22 (1998) (85-97).*
- Odrej N, G. Mejía. A reconfigurable multiagent system architecture for error recovery in production systems. *Robotics and Computer Integrated Manufacturing 19 (2003) (35-43).*
- Bussman S., K Schild. An Agent-based approach to the control of Flexible Production Systems. *8th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA, 2001) (169-174).*
- Swaminathan J, S. Smith, N. Sadeh. Modeling Supply Chain Dynamics: A Multiagent Approach. *Decision Sciences Volumen 29 Number 3 (1998) (607-632).*

- Jennings N. An agent-based approach for Building Complex software systems. *Communications of the ACM*. April 2001/Vol. 44 No 4. (35-41).
- Sycara K. Multiagent Systems. *American Association for Artificial Intelligence*. (1998). (79-92).
- Wooldridge M., N.R. Jennings, D. Kinny. The GAIA methodology for agent-oriented analysis and design. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3): 285-312, 2000.
- Zambonelli F., N. Jennings, A. Omicini, M. Wooldridge. Agent-oriented software engineering for internet applications. In A. Omicini, F Zambonelli, M Klusch and R. Tolksdorf editors, *Coordination of Internet Agents: Models, Technologies and Applications*. Springer-Verlag, 2000.326-346.
- Zambonelli F., N. Jennings, M. Wooldrige. Organizational rules as an abstraction for the analysis and design of multiagent systems. *International Journal of Software Engineering and Knowledge Engineering*. Vol. 11, no. 3. (2001) (303-308).
- DeLoach S., M. Wood, C. Sparkman. Multiagent Systems Engineering. *Int. Journal of Software Engineering and Knowledge Engineering*. Vol 11m no 3. (2001) (231-258).
- Nwana H., D. Ndume, L. Lee and J. Collis. A toolkit for building Distributed Multiagent Systems. *Applied Artificial Intelligence Journal*, vol 1, no. 13, (1999) (129-185).
- Villarreal P., M. Alesso, S. Rocco, Ma R. Galli, O. Chiotti. Approaches for the Analysis and Design of Multi-Agent Systems. *Proceedings of the Argentine Symposium of Artificial Intelligent (ASAI 02)*, 31 JAIIO, Santa Fe, Argentina (2002) (50-61).
- R. G. Smith. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Trnasactions on Computers*, C-29 (12), pages 1104-1113, 1980.