

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
DOCTORADO EN CIENCIAS APLICADAS

COORDINACIÓN DE SISTEMAS MULTI-AGENTE USANDO
ALGORITMOS CULTURALES. CASO DE ESTUDIO: PROBLEMA DE
INTEGRACIÓN EN AUTOMATIZACIÓN

POR:

JUAN CARLOS TERÁN PICÓN

TESIS DE GRADO PRESENTADA COMO REQUISITO PARCIAL PARA LA OBTENCIÓN
DEL DOCTORADO EN CIENCIAS APLICADAS

TUTORES:

DR. JOSÉ AGUILAR,
DRA. MARIELA CERRADA

MÉRIDA, VENEZUELA

Marzo, 2015

RESUMEN

La automatización industrial basada en agentes permite resolver el problema de integración usando la teoría de Inteligencia Artificial Distribuida, pero presenta nuevos retos que requieren de sus estudios y modelados, para ser abordados con óptimos resultados. Uno de esos retos es estudiado en este trabajo: la determinación de la coordinación óptima entre los agentes. Por otra parte, la computación evolutiva nos permite crear modelos con el fin de resolver problemas de optimización combinatoria, por lo que puede ser usada para enfrentar dicho reto en la automatización industrial basada en SMA. Este trabajo tiene como objetivo presentar un modelo de aprendizaje de esquemas de coordinación en SMA, basado en Algoritmos Culturales (AC). En este modelo, los individuos (uno de los componentes principales del AC) son las diferentes conversaciones que pueden ocurrir en el SMA bajo estudio, y estos individuos pueden aprender que tipo de mecanismo de coordinación es el más apropiado dado el contexto de una conversación. Cada conversación se identifica con un tipo particular de conversación, asociada con ciertos patrones de interacción, según lo establecido en la literatura y en los estándares. Además, los mecanismos de coordinación clásicos en la literatura, de negociación y de planificación, son formalizados matemáticamente, de tal manera de poder ser manipulados por el AC. Basado en dicho modelo, se desarrolló una herramienta computacional, llamada CLEMAS, que se ha utilizado para aplicar el modelo a varios casos de estudio del problema de integración en automatización industrial, con el fin de analizar sus capacidades para resolver dicho problema, tanto a nivel de la integración de servicios como de datos, presentes en ellos.

TABLA DE CONTENIDOS

	Pág.
RESUMEN	ii
LISTA DE FIGURAS	vi
LISTA DE TABLAS	vii
PARTE A. MARCO TEÓRICO	
CAPÍTULO	
I. INTRODUCCIÓN	1
1.1 Generalidades	1
1.2 Motivación e importancia del problema	3
1.3 Objetivo general	4
1.3.1 Objetivos específicos	4
1.4 Estado de arte	4
1.5 Organización de la tesis	9
II. MARCO TEÓRICO	10
2.1 Sistemas multi-agente (SMA)	10
2.1.1 Coordinación en SMA	11
2.1.1.1 El mecanismo de subasta en SMA	12
2.1.1.2 El protocolo de licitación en SMA	14
2.1.1.3 El mecanismo de planificación en SMA	15
2.1.2 Aprendizaje colectivo en SMA	16
2.1.2.1 Algunas técnicas de aprendizaje colectivo	17
2.2 Algoritmos culturales (AC)	18
2.3 Automatización industrial	22
2.3.1 Integración en automatización	26
2.3.2 Sistemas multi-agente en automatización	27
PARTE B. DESARROLLO	
III. INTEGRACIÓN EN AUTOMATIZACIÓN	31
3.1 La automatización desde la teoría de agentes	31
3.2 Integración en automatización basada en la coordinación de SMA	33
IV. FORMALIZACIÓN DE LOS MECANISMOS DE COORDINACIÓN	36
4.1 Modelo formal para el mecanismo de subasta	36
4.2 Modelo formal para el mecanismo de licitación	39
4.3 Caracterización del proceso de planificación e los SMA	40
4.3.1 Modelo formal del mecanismo de planificación multi-agente	41

V. MODELO CULTURAL PARA EL APRENDIZAJE DE SISTEMAS MULTI-AGENTE	46
5.1 Principios básicos para la caracterizar los componentes del AC	46
5.2 Modelo formal basado en algoritmos culturales	50
5.2.1 Individuos	50
5.2.2 Función objetivo o fitness (FO)	52
5.2.3 Espacio de creencias	55
5.2.4 Funciones de aceptación e influencia	56
VI. HERRAMIENTA COMPUTACIONAL PARA EL APRENDIZAJE DE COORDINACIÓN EN SISTEMAS MULTI-AGENTE: CLEMAS	60
6.1 Arquitectura de CLEMAS	60
6.2 Diseño de CLEMAS	61
6.2.1 Individuos	61
6.2.2 Espacio de creencias	61
6.2.3 Sistema principal de la numeración	62
6.2.4 Interfaz	63
6.2.5 Panel de resultados	66
VII. CASOS DE ESTUDIO	67
7.1 Casos de estudio	67
7.1.1 Caso A: Sistema manejador de fallas basado en un Sistema multi-agente	67
7.1.2 Caso B: Planificación de un proceso de producción petrolera por levantamiento por gas basado en SMA	71
7.1.3 Caso C: SMA de una unidad de explotación de Yacimientos de producción petrolera	72
7.2 Uso de la formalización de los mecanismos de coordinación en los casos de estudio	75
7.2.1 Ejemplo de una subasta inglesa para la conversación de tareas urgentes del caso de estudio A	75
7.2.2 Ejemplo de licitación en la conversación de mantenimiento por condición del caso de estudio A	79
7.2.3 Formalización del mecanismo de planificación para el caso de estudio B	81
7.3 Protocolos de experimentación con CLEMAS en casos de estudio A y C	84
7.3.1 Detalles del caso de estudio A	84
7.3.1.1 Especificación del caso de estudio en CLEMAS	84
7.3.1.2 Diseño de los experimentos	85
7.3.1.3 Análisis de resultados	87
7.3.2 Verificación de la capacidad de integración en los sistemas de automatización en SMA usando CLEMAS y el caso de estudio C	88
7.3.2.1 Detalles del caso de estudio	88
7.3.2.2 Diseño de los experimentos	91
7.3.2.3 Análisis de los resultados	92

CONCLUSIONES	94
REFERENCIAS	99

LISTA DE FIGURAS

2.1 Planificación centralizada para planes distribuidos	15
2.2 Planificación distribuida para un plan centralizado	16
2.3 Planificación distribuida para planes distribuidos	16
2.4 Estructura de un AC.	20
2.5 Modelo de Automatización Piramidal	24
2.6 PROSA	25
2.7 SADIA	28
2.8 SCDA	29
3.1 SADIA con ejemplos de instanciación	32
3.2 Interacciones entre SMA	34
4.1 Niveles de Coordinación y sus Mecanismos	40
4.2 Modelos clásicos de planificación	43
4.3 Matrices de detalle	44
5.1 Actos de habla	46
5.2 Pseudocódigo del algoritmo cultural	51
5.3 Estructura interna del individuo	51
5.4 Ejemplos de cruce y mutación	52
5.5 Cambio del mecanismo de coordinación de manera aleatoria	57
5.6 Modificación de los parámetros de manera aleatoria	58
5.7 Modelo cultural de aprendizaje SMA	58
6.1 Arquitectura de CLEMAS	60
6.2 Diagrama UML para la clase individuo	62
6.3 Diagrama en UML de la clase EspacioCreencias	63
6.4 Diagrama UML de la clase CruzaIndividuos	64
6.5 Panel de configuración inicial	64
6.6 Diagrama UML de las clases de los paneles principales	65
6.7 Ventana para crear una sub-conversación	65
6.8 Diagrama UML de las clases que muestran los resultados	66
7.1 SMA del SMF	68
7.2 SMA para la planificación en un lazo de producción de pozos LAG	71
7.3 SMA de la unidad de explotación de yacimientos petrolera	72
7.4 Modelo de referencia del agente supervisión	74
7.5 Modelo de referencia del agente de control	74
7.6 Diagrama de interacción de la conversación tareas urgentes	77
7.7 Diagrama de interacción para una subasta Inglesa en la conversación tareas urgentes	79
7.8 Conversación con sus TC	85
7.9 Evolución de la función objetivo para el escenario uno	86
7.10 Diagrama de interacción de la conversación monitorear proceso	89
7.11 Diagrama de interacción de la conversación diseñar control	90
7.12 Diagrama de interacción de la conversación evaluar desempeño del plan de control y sus TC	91

LISTA DE TABLAS

4.1 Matriz resumen de variables de subastas	38
5.1 Actos de habla Vs. Actos Comunicativos (mensajes)	47
5.2 Tipos de conversación y Protocolos de interacción FIPA	48
5.3 Patrones característicos en los tipos de conversación:	50
5.4 Definición de los parámetros de la ecuación (2)	53
5.5 Valores cualitativos de los parámetros PI, PE y $A_{l,q}$	54
5.6 Definición de los parámetros de la ecuación (3)	54
5.7 Valores cualitativos de los parámetros CEP, CEO y CS	55
5.8 Conocimiento Circunstancial	56
5.9 Conocimiento Normativo	56
7.1 Modelo de Tareas	69
7.2 Modelo de Coordinación	69
7.3 Ejemplo matriz de la Ofertas	78
7.4 Ejemplo de la matriz de Orden	78
7.5 Ofertas para TDi_1	80
7.6 Ofertas para TDi_2	80
7.7 Ofertas para TDi_3	80
7.8 Mensaje para los ganadores de cada tarea TDi_j (hipotéticos)	80
7.9 Reporte parcial de cada tarea TDi_j	81
7.10 Matriz T(PCpD)	83
7.11 Matriz SSP	83
7.12 Matriz ASP	84
7.13 Matriz T	84
7.14 Resultados (porcentaje de uso de cada MC en cada TC, basado en el 20% de la población, caso A, 35 gen.)	86
7.15 Resultados (porcentaje de uso de cada MC en cada TC, basado en el 20% de la población, Caso A, 50 gen.)	87
7.16 Resultados (porcentaje de uso de cada MC en cada TC, basado en el 20% de la población, caso C)	92

CAPÍTULO I

INTRODUCCIÓN

1.1 Generalidades

La puesta en operación de sistemas informáticos en procesos de alta complejidad requiere un esfuerzo significativo, de manera especial en las fases de modelado y construcción. La evolución en el desarrollo de la Ingeniería de Software ha contribuido de manera fundamental para la puesta en operación de sistemas informáticos de alta complejidad [1]. En particular, se han hecho esfuerzos por plantearse formas para la solución de problemas cooperativamente. La Inteligencia Artificial Distribuida (IAD) es una de las áreas que ha estudiado ese problema, en este caso, integrando conceptos provenientes tanto de la Inteligencia Artificial (IA) como de los Sistemas Distribuidos [1]. La IAD se caracteriza por una arquitectura formada por componentes inteligentes y modulares que interactúan de forma coordinada [2]. El campo de la Inteligencia Artificial intenta construir entidades inteligentes, mientras que el campo de los Sistemas Distribuidos estudia las propiedades de conjuntos de procesos autónomos, que cooperan para alcanzar objetivos colectivos, comunicándose por medio del envío de mensajes sobre una red de comunicación o compartiendo memorias [3].

Dentro de las áreas de la IAD se encuentran los Sistemas Multi-Agente (SMA), que estudian la coordinación de la conducta entre un conjunto de agentes inteligentes autónomos [3]. Uno de los estudios más importante actualmente en los SMA tiene que ver con los problemas de cooperación y coordinación entre los elementos que componen dichos sistemas. La coordinación es considerada como un proceso clave para que los agentes de una comunidad aseguren acciones coherentes y logren objetivos colectivos determinados [4]. En particular, se requiere de nuevas metodologías, técnicas y entornos de soporte informático para el desarrollo de sistemas que consideren estos aspectos [2].

Por otra parte, la teoría de agentes puede ser vista como una evolución de la IA en la búsqueda de aportar autonomía a los sistemas computacionales. Existe el acuerdo de que la autonomía es la característica principal que describe un agente, entendiendo como autonomía la capacidad del agente de actuar sin la intervención de un usuario o de otro sistema [5]. Dentro de las propiedades que poseen los agentes está la inteligencia, a través de la cual ellos son capaces de aprender. Este aprendizaje puede darse a nivel individual o multi-agente (colectivo). En este trabajo estudiamos el problema de aprendizaje colectivo (SMA). En el aprendizaje colectivo o distribuido, el aprendizaje es llevado a cabo por los agentes como un grupo (e. g., mediante el intercambio de conocimientos, observación de otros agentes, etc.).

Por otro lado, dentro de la computación evolutiva se encuentran los Algoritmos Culturales (AC), desarrollado por Robert G. Reynolds [7, 8, 9, 10, 11, 12]. Los AC fueron

desarrollados para modelar la evolución de los sistemas culturales basados en principios humanos de la evolución social [10]. Los AC operan en dos espacios. Primero en el espacio de la población, como en todos los métodos de computación evolutiva, en el que se tiene un conjunto de individuos que van evolucionando genéticamente. Cada individuo tiene un conjunto de características genéticas con las que se es posible determinar su aptitud. A través del tiempo, tales individuos podrán ser reemplazados por algunos de sus descendientes, obtenidos a partir de un conjunto de operadores genéticos aplicados a la población. El segundo espacio es el de creencias, donde se almacenaran los conocimientos que han adquirido los individuos en generaciones anteriores. La información contenida en este espacio debe ser accesible a cualquier individuo, quién pueda utilizarla para modificar su comportamiento. Para ello se establece un protocolo de comunicación que dicta las reglas del tipo de información que se debe intercambiar entre los espacios. Este espacio también va evolucionando con el nuevo conocimiento que se va adquiriendo [7].

Ahora bien, con respecto al desarrollo industrial, las empresas han ido evolucionando en sus esquemas de manejo de los procesos de negocio, desde sistemas de información, control y decisión aislados, hacia sistemas que manejan el conocimiento completo de lo que sucede en la empresa [13]. Gran parte de esta evolución tiene que ver con la automatización, a partir de la cual se establecen nuevos paradigmas y criterios para el control y supervisión de los procesos [1]. En particular, la Automatización Industrial, es un conjunto de técnicas que involucran la aplicación e integración de sistemas mecánicos, eléctricos, electrónicos, fluidos, soportados por tecnologías de información y comunicación (TIC), todos ellos unidos para operar, supervisar y controlar diferentes tipos de procesos técnicos de forma autónoma [14]. Los sistemas de automatización industrial son aplicaciones que se caracterizan por requerimientos bien específicos de productividad y de seguridad operacional, así como de confiabilidad, eficiencia y calidad [14]. En específico, las tendencias en el desarrollo informático moldean las tendencias en automatización industrial, originando nuevos paradigmas en el ámbito de la automatización de procesos [1]. Pero el gran desarrollo actual de componentes de hardware y de las TIC, ha generado una fuerte necesidad de integración en los sistemas de automatización industrial [14]. Este conocimiento completo de lo que sucede en la empresa, se alcanza mediante la integración de las plataformas computacionales que componen su entorno. Esa integración puede ser vertical como horizontal, entendiendo como vertical aquella que está vinculada con el flujo de decisiones entre niveles gerenciales y su retroalimentación de la información. En este caso el manejo de las decisiones tiene una estructura totalmente jerárquica, en donde a las distintas funciones de la empresa se le asocia un orden de jerarquía. La integración horizontal está vinculada con el flujo de productos e información en un mismo nivel organizacional (no entre niveles). En este caso, distintas unidades de negocio (en un mismo nivel) realizan actividades complementarias, donde la información de unas depende de las otras. Para el caso de la integración horizontal, los modelos más utilizados son los flujos de trabajo y los modelos basados en cadenas de valor [1].

En general el problema de integración en automatización industrial es un problema abierto, que ha obligado a pensar en nuevos paradigmas, por ejemplo los basados en la teoría de IAD. De esa teoría, los SMA permiten diseñar sistemas sofisticados y complejos [14]. Ahora bien, un aspecto fundamental de los SMA para resolver los problemas de integración en automatización industrial, tiene que ver con los procesos de aprendizaje colectivo implícitos en ellos. En este trabajo nos dedicaremos a estudiar los mismos.

1.2 Motivación e importancia del problema

La interacción entre aplicaciones, el intercambio de datos heterogéneos, los diferentes servicios involucrados en un proceso productivo, entre otras cosas, implica la necesidad de lograr una buena integración. [1]. En particular, el número, la variedad, y la sofisticación de las aplicaciones informáticas avanzadas para la automatización de procesos continua creciendo. La arquitectura de automatización influye en las aplicaciones a implantar, porque determina sus características no funcionales, tales como los tipos de datos y, los mecanismos de comunicación entre procesos soportados, la interoperabilidad de los bloques, entre otras cosas. En este sentido, la integración industrial estudia esos aspectos, con la finalidad de establecer estrategias que permitan poner a trabajar en conjunto a aplicaciones heterogéneas en un entorno de automatización. Fundamentalmente la integración se basa en el desarrollo de modelos de intercambio de información, de cooperación y de representación común de los datos, para posibilitar la ejecución de las acciones u objetivos planteados. La integración implica la interrelación entre las distintas áreas del conocimiento existentes en una empresa, entre éstas: (i) investigación de operaciones, (ii) computación, (iii) control de procesos, y (iv) redes de comunicaciones. La necesidad de integración tiene su origen en distintos factores que se presentan tradicionalmente al momento de coordinar el trabajo de los diversos elementos involucrados en las distintas fases del proceso productivo, entre estos factores están [1]:

- Diversidad de marcas, sistemas operativos, protocolos de comunicación y bases de datos. Además, algunos de estos protocolos de comunicación son primitivos, ineficientes y específicos para la transmisión de una información predeterminada.
- Dispersión de datos con eventual redundancia. En algunos casos la misma información es representada en diversas formas (nomenclaturas diferentes), lo cual obliga a mantener tablas de conversión, por lo que se hace necesario implantar sistemas complejos de mantenimiento y actualización.
- La interacción con los diversos sistemas, a través de interfaces de usuario disímiles y poco amistosas, a veces incluso incompatibles entre sí.
- Poca integración entre las áreas de control de procesos, informática, gerencia empresarial e instrumentación; esto debido a la rápida evolución de cada área.

En ese contexto, la integración de las diferentes tecnologías existentes en una empresa se ha vuelto un factor determinante en el proceso productivo, de manera especial cuando estos

procesos se desarrollan bajo alguna estrategia de automatización. En ese escenario se requiere el manejo óptimo de la información disponible, desde el nivel operacional, donde se ejecutan acciones de control sobre la planta, hasta los niveles de planificación corporativa, donde se toman decisiones que afectan el desempeño global de la empresa [1].

En nuestro caso, proponemos resolver el problema de integración en automatización, a través de la coordinación. Eso implica modelar el problema de automatización usando SMA. Además, implica analizar las estrategias de coordinación y comunicación, tal que un sistema automatizado basado en agentes pueda escoger que mecanismo de coordinación resulta el más idóneo en un momento dado, para sus diferentes conversaciones. Para realizar esa tarea se requiere, entre otras cosas, la capacidad de aprendizaje colectivo, de tal manera de explotar las experiencias y conocimientos que se vayan generando en el SMA derivados de sus procesos de coordinación.

1.3 Objetivo general

Proponer un esquema de aprendizaje de mecanismos de coordinación en una sociedad de agentes (Sistema Multi-agente) usando Algoritmos Culturales, y aplicar dicho esquema en el problema de integración en automatización industrial.

1.3.1 Objetivos específicos

1. Definir un esquema genérico de aprendizaje de mecanismos de coordinación para SMA.
2. Caracterizar el esquema de aprendizaje de mecanismos de coordinación usando los AC.
3. Modelar el proceso de integración en automatización desde la teoría de SMA
4. Incorporar el esquema de aprendizaje de mecanismos de coordinación basados en AC en el modelo de integración basado en agentes.

1.4 Estado de arte

Los antecedentes más relevantes para este trabajo se han organizado por secciones, según los aspectos de interés para el desarrollo de la tesis. Así, inicialmente consideraremos trabajos vinculados al problema de coordinación en los SMA, después trabajos que usen los AC en SMA, y finalmente trabajos que usen SMA en automatización y en problemas de integración.

Problema de Coordinación en Agentes

Dentro del marco de la coordinación en SMA existe una gran cantidad de trabajos orientados a su estudio [4, 6, 15]. En particular, presentaremos algunos trabajos que han tratado de formalizar el problema de coordinación en SMA, por el interés en nuestro

trabajo. Los modelos propuestos en estos trabajos son detallados en el capítulo II. Comenzamos con [15], en donde abordan el problema de la coordinación ad hoc. En este problema de coordinación el objetivo es diseñar agentes autónomos denominados agentes ad hoc. Estos agentes son capaces de alcanzar una óptima flexibilidad y eficiencia. Los agentes ad hoc no requieren de una previa coordinación para interactuar con otros agentes. La flexibilidad describe la habilidad que tienen los agentes ad hoc para realizar sus tareas junto con otros agentes en el sistema. La eficiencia es el tiempo que necesitan para realizar sus tareas. Para lograr esto basan su modelo en la teoría de juegos, y derivan una solución llamada *Harsanyi-Bellman Ad hoc coordination* (HBA), la cual utiliza el equilibrio Bayesiano de Nash¹ para planear procedimientos que lleven a encontrar acciones óptimas en el sentido del control óptimo de Bellman (programación dinámica). En [16] estudian la coordinación distribuida en sistemas multiagente con dinámicas no-lineales caracterizando las interacciones entre los agentes con grafos no dirigidos. En ese trabajo abordan dos problema de coordinación, aquella que se logra por consenso de todos los agentes, es decir, sin un líder que los guíe, y aquella que posee un agente líder que guíe a los demás agentes para lograr la coordinación. Establecer una coordinación con o sin líder lo logran introduciendo un filtro distribuido en cada agente o seguidor, a través de un algoritmo de control el cual usa una medida de posición relativa cuando hay ausencia de comunicación. Finalmente, otro trabajo que citaremos es [17], en el que se propone una abstracción unificada para la coordinación de agentes, basada en las ideas de entornos y de objetos de coordinación. La idea principal es captar en un sistema multi-agente conceptos como servicios, herramientas y artefactos compartidos y explotados por el colectivo o por individuos, para lograr sus objetivos globales o individuales. El trabajo se basa en un conocimiento distribuido del entorno - en particular de las herramientas disponibles - a partir del cual se establecen acciones cooperativas de la comunidad. Los objetos de coordinación son vistos como abstracciones que ayudan a los agentes a cooperar y coordinar sus actividades. Es decir, existen entidades especializadas en proveer servicios de coordinación en SMA.

Algoritmos Culturales en Sistemas Multi-agente.

Los algoritmos culturales (AC) han servido como estructura fundamental para modelar la evolución de comportamientos cooperativos en grupos sociales [9, 10]. Un ejemplo de uso en ese contexto es dado en [11], en el cual se propone su uso en la coordinación de robots. Allí, los AC proveen una estructura para establecer un marco de cooperación multiagente entre un grupo de jugadores de fútbol (robots). El sistema es utilizado para aprender varios tipos de jugadas: ofensivas y defensivas. El AC es utilizado para enseñar a los agentes

¹ En teoría de juegos, un equilibrio Bayesiano de Nash es un juego donde la información sobre las características de los otros jugadores es incompleta. Se denomina Bayesiano por el análisis probabilístico inherente en el juego, debido a que los jugadores tienen creencias iniciales sobre cada jugador, y pueden actualizarlas de acuerdo con la regla de Bayes, es decir, la creencia que tiene un jugador sobre el tipo de otro jugador podría cambiar en función de las acciones que han jugado [18]

robots a identificar la mejor acción a ejecutar dependiendo de su posición a jugar en el campo, y su relación con el jugador más cercano. Para ello, las acciones de cada agente son codificadas en una cadena de valores enteros. Estas acciones son evaluadas a través de la función objetivo la cual está basada en la cantidad de goles que el equipo puede conseguir a raíz de dichas acciones.

La aplicación de los AC para la gestión de conocimiento ha sido también explorada [19, 20]. Por ejemplo, en [20] aplican los AC para actualizar redes semánticas. Las redes semánticas son utilizadas generalmente para representar conocimiento, y para el caso de ese trabajo, son usadas para modelar toda la información relevante acerca del sistema de planificación del proceso de manufactura de automóviles de la compañía Ford Motor. En ese trabajo sugieren, que los AC pueden ser utilizados para identificar los atributos que son más significativos, para la recuperación de nodos (de la red semántica), y permiten utilizar el conocimiento implícito en las redes semánticas de manera más eficiente y menos complejas [20]. En específico, los AC permiten rediseñar las redes semánticas y derivarlas en redes menos complejas y más eficientes. Así, la información que poseen los individuos son conceptos de ingeniería y atributos divididos por secciones o cluster. Uno de los requerimientos principales en algunas redes semánticas es un algoritmo que ubique en el lugar apropiado los nodos nuevos o modificados. Este proceso se conoce como clasificador, y está basado en los atributos de cada nodo, esa es la tarea que hacen los AC.

Como se puede constatar, para el caso de la coordinación en SMA prácticamente no existen trabajos, y más que todo se enfoca en la coordinación de robots, como es el caso antes expuesto de [11].

Automatización e Integración usando SMA

Dentro del marco de automatización de procesos basados en SMA, han sido diversos los estudios basados en la IAD [13, 16, 21, 22]. Generalmente, los sistemas automatizados son distribuidos, y cada subsistema puede operar con diferentes tipos de aplicaciones. Es por esto que su integración (cooperación y trabajo conjunto entre subsistemas) y comunicación es compleja, pero necesaria para optimizar los procesos productivos (producción y costos operacionales, entre otros aspectos). A continuación presentamos algunos de los trabajos que han propuesto modelar la Automatización Industrial, y resolver el problema de integración usando SMA. En [23] presentan una arquitectura de integración para un sistema automatizado basada en IAD, denominada ARCHON, la cual es implementada en una compañía de energía eléctrica. Esta arquitectura parte del principio de que los componentes del sistema son vistos como agentes. Además, la arquitectura propone los siguientes módulos: el de Monitoreo, nivel de cada agente, donde se encuentran las unidades de monitoreo de las tareas de los diferentes componentes del sistema; el de Planificación y Coordinación a nivel de cada agente, donde se decide cuales acciones deben tomar los agentes y sus interacciones con otros agentes; y el de Gerencia de Información sobre los Agentes, el cual es global y provee la información de los servicios disponibles para los

agentes cooperativos; y finalmente el de Comunicación, que permite que los agentes se comuniquen con otros usando el protocolo TCP/IP. En [23] se detallan los dos primeros módulos de ARCHON.

Otra propuesta es el modelo de referencia SADIA (Sistemas de Automatización Distribuida Inteligente basada en Agentes) [21], la cual propone un conjunto de niveles (3 en total) para definir un proceso industrial, donde cada nivel es descrito por un comunidad de agentes, los cuales son compuestos por los agentes del nivel inferior inmediato. Cada agente tiene sus tareas de automatización bien definidas. Así, en el primer nivel se modelan los objetos de negocio como agentes, esto es, el proceso productivo es visto como un SMA, donde las diversas unidades de producción del proceso son vistas como agentes. Los agentes de este nivel (unidades de producción) deben negociar entre sí para llegar a acuerdos que permitan cumplir con las metas de producción establecidas. En el segundo nivel, cada agente del primer nivel es visto como un SMA, compuesto por agentes que se ocupan de las actividades necesarias en un proceso automatizado: control de procesos, ingeniería de mantenimiento, manejo de situaciones anormales, manejo de los factores de producción, y planificación de la producción. Las actividades antes listadas son comunes para cada agente del primer nivel (objetos de negocio). Sin embargo, existen actividades que son exclusivas de un objeto de negocio en particular, las cuales son modeladas por medio de agentes especializados, que complementan la arquitectura base de cada agente del primer nivel. Finalmente, ya que las actividades que desempeñan los agentes del segundo nivel son complejas, se propone un tercer nivel de abstracción en donde cada agente del segundo nivel sea visto como un SMA. En el tercer nivel de abstracción se hace uso del marco de referencia SCDIA [22] para modelarlo, así, cada agente del segundo nivel de abstracción es modelado usando el SCDIA. SCDIA es un SMA que modela un sistema de control de procesos, caracterizando los elementos presentes en un lazo de control. Estos agentes son los agentes coordinador, controlador, de medición, y de actuación [22]. El agente de medición y actuación se encuentran en un mismo nivel, e interactúan con el proceso productivo. Ya que el SCDIA está inspirado en los componentes de un lazo de control de procesos, cada una de las actividades desempeñada por los agentes del segundo nivel se modelará como un lazo de control.

En [24] se presenta un enfoque de diseño para sistemas de control y supervisión de procesos industriales basados en agentes inteligentes. En él se define una plataforma general, la cual soporta un SMA para el control y la supervisión del proceso productivo, y a partir de aquí, las aplicaciones son concebidas como SMA especializado, para coordinar, ejecutar y evaluar tareas de control y supervisión, que son necesarias para procesar la información distribuida y tomar decisiones de control. En ese sentido, las diferentes tareas de control y supervisión son modeladas como SMA. Las unidades de producción en entornos industriales también son modeladas como SMA, implementando una abstracción lógica y funcional de los procesos reales. De esta manera la cooperación y negociación para

mejorar el desempeño de la producción es alcanzado a través de la interacción adecuada entre la comunidad de agentes, agentes que proveen servicios de control y supervisión a los agentes que representan las unidades de producción.

En [25] presentan un enfoque basado en SMA para la integración de operaciones de manufactura. Ellos definen una plataforma de decisión integrada, donde concurren las decisiones de control y planificación conjuntamente con las decisiones de reconfiguración del sistema, para responder a los cambios de la demanda. El sistema propuesto permite modelar sistemas complejos heterogéneos, con sus restricciones físicas. La lógica de funcionamiento del sistema propuesto se centra en detectar cuando la estructura actual del sistema de manufactura no puede responder a las demandas; luego, el modelo debe ser capaz de identificar las opciones de reconfiguración a través de la relajación gradual de las restricciones actuales. En esta arquitectura se identifican tres capas: *Capa de modelado y planificación*: en la cual se modelan el sistema de manufactura y las órdenes de producto; y se realiza el control, la planificación y la reconfiguración del sistema de manufactura. Por ejemplo uno de los problemas clásicos de optimización en esa capa es la asignación de un trabajo que considera cierta cantidad de operaciones que tienen que ser completadas con ciertos recursos, en una secuencia dada con un mínimo costo. *Capa de flujo de proceso*: describe los enfoques que permiten identificar las opciones de reconfiguración, a partir de los resultados de planificación. *Capa de simulación*: aloja un ambiente de simulación integrada de sistemas a eventos discretos, para generar y evaluar escenarios operativos. Para cada una de estas capas ha sido propuesta una arquitectura de agentes, responsables de llevar a cabo las funcionalidades de cada capa [25].

Otra arquitectura basada en agentes es propuesta en [26], denominada Arquitectura de Planta Automatizada basada en Sistemas Distribuidos (PABADIS, por sus siglas en inglés) Esta arquitectura permite desarrollar Sistemas de Ejecución de Manufactura² (MES, por sus siglas en inglés) distribuidos basados en la tecnología de agentes con el fin de enfrentar las limitaciones de los MES clásicos: complejidad, flexibilidad y robustez [26]. Es una arquitectura que refuerza la descentralización de los sistemas basándose en agentes móviles, resaltando las siguientes características:

- La programación de un MES distribuido se reduce a la ejecución del proceso de producción, como respuesta a las órdenes de otros agentes.
- No hay necesidad de una comunicación exhaustiva con los sistemas MES y los Sistemas de Gestión de Negocios³ (ERP, por sus siglas en inglés).
- Las funcionalidades del MES, implementadas con agentes, pueden cambiarse con sólo ajustar el código de los agentes.

² Manufacturing Execution System. Responsables de ejecutar y hacer seguimiento a las recetas de producción requeridas a nivel de los equipos de control de plantas (PLCs, SCADAs, etc.)

³ Enterprise Resource Planning. A través de los cuales se lleva el control de inventario de producción, órdenes de producción, órdenes de compra, órdenes de mantenimiento, nómina y recursos humanos, entre otras cosas.

- La integración con los sistemas ERP se realiza a través de un supervisor.

Estas características permiten procesos de toma de decisiones locales, disminuyendo los tiempos de reacción debido a los eventos ocurridos a nivel de campo. De esta manera los sistemas ERP son los responsables de la planificación estratégica, y los sistemas MES son los responsables de la planificación táctica, realizada por unidades distribuidas y cooperativas [26]. En general, el enfoque PABADIS propone disolver el MES en una parte centralizada adjunta al sistema ERP, y otra parte descentralizada que es implementada por agentes.

Existen otros trabajos más que usan la teoría de agentes en automatización industrial, en [1] hacen un análisis de algunos otros.

1.5 Organización de la tesis

La tesis está estructurada de la siguiente manera: en el capítulo dos se aborda el marco teórico de la tesis, en donde se estarán presentando los conceptos que sirven de base a la tesis como: coordinación en SMA, aprendizaje colectivo en SMA, AC, e integración y automatización industrial. En el capítulo tres se trata el tema de la integración en SMA, en donde se abordan dos tópicos importantes para nuestro trabajo: automatización industrial desde la teoría de agentes, y el problema de integración en automatización industrial, presentando nuestra propuesta para abordar ese problema. En el capítulo cuatro se describen las formalizaciones matemáticas de los mecanismos de coordinación, caracterizando sus parámetros. Esto con la finalidad de poder modificar su desempeño a través de los AC. El capítulo cinco describe el modelo cultural de aprendizaje en SMA, es decir, se presentan los formalismos previos de los mecanismos de coordinación estructurados dentro de los componentes de un AC, para así poder evolucionar a través de él. El capítulo seis se refiere a la implementación de la herramienta computacional CLEMAS, este modelo computacional permite modelar un SMA en nuestro modelo cultural de aprendizaje, y finalmente presenta cual mecanismo de coordinación resulta más idóneo para cada conversación que compone al SMA, además de mostrar gráficamente la evolución del aprendizaje colectivo sobre los mecanismos de coordinación más idóneos para ese SMA. El capítulo siete aborda tres casos de estudio que se refieren al problema de integración en automatización industrial, en cada caso se comprueba el comportamiento de las formalizaciones propuestas, y posteriormente se realiza un análisis de los resultados obtenidos; al final del capítulo se realiza una comparación con trabajos previos similares a los nuestros. Finalmente, se presentan las conclusiones.

CAPÍTULO II

MARCO TEÓRICO

Dentro del marco teórico, abordaremos los SMA, haciendo hincapié en los aspectos de coordinación y aprendizaje, los AC, la integración y automatización industrial.

2.1 Sistemas multi-agente (SMA)

Un SMA está formado por un grupo de agentes. Un agente puede ser “una entidad virtual o física capaz de actuar, percibir su entorno, y comunicarse con otros, es autónomo y posee habilidades para alcanzar metas y tendencias” [17]. La interacción entre dos o más agentes compone un SMA, el cual contiene entornos, objetos, relaciones entre los agentes, y tareas que pueden ser ejecutadas por ellos [17]. Así, un SMA tiene que ver con el comportamiento de una colección de agentes autónomos tratando de resolver un problema dado, por lo cual comparten conocimiento acerca del problema y su(s) solución(es). Para interactuar los agentes utilizan protocolos y lenguajes de comunicación de alto nivel, para resolver problemas que están más allá de la capacidad o del conocimiento de cada uno. Algunas de las características de los SMA son [1]:

- Cada agente tiene capacidad para solucionar parcialmente el problema
- No existe un sistema global que los controla
- Los datos no están centralizados
- La computación es asíncrona

Un SMA puede ser visto como un sistema distribuido compuesto por agentes, donde la conducta combinada de ellos produce un resultado en conjunto inteligente. En general, en un SMA se pueden distinguir tres niveles de organización [1]:

- El nivel micro social, que se interesa esencialmente por las interacciones y conexiones que existen entre los agentes.
- El nivel de grupo, que se interesa en las estructuras que se producen en la composición de una organización compleja. En este nivel se estudian los roles y actividades de los agentes, la emergencia de las estructuras organizacionales entre los agentes, y el problema general de agregación de agentes para la constitución de organizaciones.
- El nivel de la sociedad global (o poblaciones), que se interesa en la dinámica de un gran número de agentes, así como en la estructura general del sistema y su evolución.

Los SMA se caracterizan por la interacción de varios agentes en un mismo entorno, ya sea este físico o virtual. Así, un concepto principal de los SMA es la interacción y coordinación

entre agentes, que no se limita a la comunicación ni al envío de mensajes entre estos, sino a la forma como un agente se relaciona con otros agentes. Existen diferentes técnicas o mecanismos de coordinación en los SMA.

2.1.1 Coordinación en SMA

Los SMA, se caracterizan por la interacción de varios agentes en un mismo entorno, ya sea este físico o virtual. Un concepto principal de la coordinación entre los agentes de un SMA, es que no se limita al envío de mensajes entre ellos estos, sino a la forma como un agente se relaciona con otros agentes.

En los SMA la coordinación puede ser vista como un proceso, en el cual los agentes participan para asegurar que sus actos comunitarios se realicen de manera coherente, entendiendo por coherencia a que tan bueno es el comportamiento de un SMA como unidad. Existen diferentes razones por la cual múltiples agentes necesitan coordinarse [27].

- **Prevención de anarquía o caos:** debido a la descentralización en sistemas basados en agentes, la anarquía puede presentarse fácilmente. Ningún agente posee una visión global del entorno al cual pertenece. Consecuentemente, los agentes solo poseen visión local, metas y conocimientos que puedan tener conflictos con otros. A través de la coordinación pueden lograr todo tipo de arreglo entre ellos, y así evitar conflictos.
- **Limitaciones globales:** existen restricciones o limitaciones individuales que los agentes pueden superar al agruparse, si quieren alcanzar satisfactoriamente sus objetivos. Para ello, deben coordinar su comportamiento.
- **Experiencia, recursos o información distribuida:** los agentes pueden tener diferentes capacidades y conocimientos especializados. Alternativamente, pueden tener diferentes fuentes de información, recursos, niveles de fiabilidad, responsabilidades, limitaciones, etc. Los agentes, necesariamente, deben estar coordinados para compartir sus capacidades y habilidades en su debido momento.
- **Eficiencia:** incluso cuando los individuos pueden funcionar independientemente, obviando así la necesidad de coordinación, la información descubierta por un agente puede ser tan útil para otro agente que ambos pudieran quizás solventar el problema dos veces más rápido.

La coordinación puede abordarse desde dos principales puntos de vista, la cooperación y la negociación. La cooperación es un tipo de coordinación que consiste en muchos agentes con interacciones no antagonistas (las acciones de un agente no perjudiquen a los otros) con el fin de alcanzar objetivos comunes. La negociación puede lograr acuerdos entre los agentes de un sistema, aun cada uno defiende sus propios intereses, lo que lleva a una situación que beneficia a todos [1]. En particular, la negociación es esencial para llegar a los acuerdos entre los agentes. En este caso, no hay objetivos comunes entre los agentes,

por esta razón es necesario los mecanismos de negociación para alcanzar sus metas. La subasta y la red de contrato (también llamada licitación) son protocolos de negociación comunes.

En la siguiente sección estaremos abordando los tres mecanismos de coordinación utilizados en esta tesis, estos son: subasta, y licitación (negociación), y planificación.

2.1.1.1 El mecanismo de subasta en SMA

Los mecanismos de subasta se han utilizado en muchas aplicaciones prácticas de la informática que implican la asignación de bienes, tareas y recursos. Una de las razones de por qué la subasta se ha vuelto tan popular es que se trata de un escenario de interacción extremadamente simple, y por lo tanto fácil de automatizar. Un mecanismo de subasta está compuesto por un grupo de agentes, donde un agente tiene el papel de subastador y los agentes restantes son ofertantes [28]. El escenario clásico supone que el subastador quiere vender un artículo en el precio más alto posible, mientras que los postores quieren comprar al precio más bajo posible. Vamos a considerar algunas de las dimensiones en las que los protocolos de subastas pueden variar [28]:

- i. *La asignación del ganador:* este proceso define quién se lleva el bien para los cuales los ofertantes están haciendo una oferta. El bien es asignado al agente con la mejor oferta. Estos protocolos son conocidos como subastas de primer precio. Una segunda forma es asignar el bien al agente que tiene la oferta más alta, pero este agente paga sólo la cantidad de la segunda oferta más alta. Estas subastas son conocidas como las subastas de segundo precio.
- ii. *La visibilidad de las ofertas:* la segunda dimensión en la que los protocolos de subastas pueden variar es, si las ofertas de los agentes se conocen entre sí o no. Si cada agente puede ver lo que cualquier otro agente está ofertando (las ofertas son de conocimiento común), entonces la subasta se dice que es a viva voz. Si los agentes no son capaces de determinar las ofertas presentadas por otros agentes, la subasta se dice que es una subasta de sobre sellado.
- iii. *Presentación de ofertas:* una tercera dimensión es el mecanismo mediante el cual las ofertas preceden. La forma más sencilla es tener una sola ronda de ofertas, después del cual el subastador asigna el bien para el ganador. Estas subastas son conocidas como subastas de una sola ronda. La segunda posibilidad es que el precio comienza a aumentar (generalmente un precio de reserva) y las ofertas sucesivas son de cantidades cada vez mayores. Estas subastas son conocidas como subastas ascendentes. La alternativa - descendente - es cuando el subastador comienza con un valor alto para disminuir el precio en las rondas sucesivas. El valor del elemento (valoración) subastado es:
 - *Valor privado:* el valor del elemento depende sólo de preferencias del agente. El adjudicatario no podrá revender o hacer buena exhibición contra el otro (en caso

contrario el valor dependerá de las valoraciones de otros agentes). Un ejemplo es la subasta de un pastel el ganador va a comer.

- *Valor común:* el valor que un agente da a un elemento, depende enteramente en el valor que tienen otros agentes de ese elemento. Por ejemplo, el valor de los bonos del tesoro depende por completo de su potencial para la reventa.
- *Valor correlacionado:* El valor asignado por un agente depende en parte de sus propias preferencias y en parte de las valoraciones de otros agentes. Por ejemplo, un agente de compra una pintura (o coche) que tal vez revende en el futuro.

A continuación se presentan diferentes protocolos de subastas, que se pueden obtener mediante la variación de las dimensiones anteriormente expuestas:

Subasta inglesa:

Es el mecanismo de subasta más popular [28]. Es de primer precio, a viva voz, y ascendente. Las características principales son:

- El subastador comienza la subasta con un precio para el recurso inicial (que puede ser cero); si no hay agentes que ofrecen más que el precio inicial, entonces el recurso está asignado al subastador.
- Las Ofertas son hechas por los agentes de los ofertantes, quienes deben ofrecer más que la oferta anterior. Los agentes pueden ver todas las ofertas realizadas, y son capaces de participar en el proceso de oferta cuando lo consideren.
- Cuando no hay agentes para mejorar la oferta actual (el más alto), la subasta ha terminado y el agente debe pagar el precio más alto ofrecido. La Fig. 1(a) muestra el protocolo de subasta propuesto por FIPA.

Subasta holandesa:

Sus principales características son:

- Este es un ejemplo de viva voz y subasta descendente [28]:
- El subastador comienza ofreciendo un precio de salida ligeramente por encima de su valor esperado.
- El subastador comienza a bajar el precio de forma continua, hasta que un agente ofrece este precio, que será el precio final a pagar.

Subasta de primer precio y a sobre cerrado:

Sus principales características son:

- Son ejemplos de subasta de una sola ronda (One Shut), y son, quizás, las más simples de todos los tipos de subastas.

- Se ejecutan en una sola ronda, en la que los postores envían sus ofertas al subastador, sin hacerlas públicas a otros agentes, (es decir, que son privadas).
- No hay siguientes rondas, y el bien se le asigna al agente que ofrece el máximo valor.

2.1.1.2 *El protocolo licitación en SMA*

La red de contratos, diseñado por Randall Davis y Reid Smith [29], son mecanismos utilizados para resolver los problemas de manera colaborativa. En este caso, el objetivo es que varios agentes se comuniquen y coordinen entre sí, para que puedan llevar a cabo una tarea cuya complejidad hace que sea difícil de realizar por un solo agente. Cada agente en la red adopta uno o dos roles relacionados con la ejecución de una tarea individual: gerente o contratista [29]. Cuando un agente tiene la función de gerente o administrador debe llevar a cabo ciertas actividades, que son: (i) dividir una tarea compleja en sub-tareas menos complejas, y por lo que será más fácil de resolver, (ii) anunciar a otros agentes que existe una sub-tarea esperando a ser ejecutada. Estos anuncios se distribuyen a través de un mensaje de difusión a todos los agentes o se dirigen a un grupo específico de agentes, (iii) cuando recibe la respuesta a sus peticiones, el administrador selecciona la oferta más adecuada y asigna una sub-tarea a este agente, para lo cual se crea un contrato, (iv) garantizar el seguimiento del contrato, pudiendo pedir información, informes, etc., es libre para reasignar la sub-tarea si el contratista no cumple, finalmente, (v) integrar los resultados parciales producidos por los contratistas en una solución completa. Mientras tanto, los agentes contratistas realizan las siguientes actividades: (i) recibir anuncios de tareas y evaluar sus propias habilidades y disponibilidad para llevarlas a cabo, (ii) si son capaces de llevar a cabo satisfactoriamente la tarea, hacen una oferta, y (iii) si esta oferta es aceptada, se reservan los recursos necesarios para su ejecución. Cuando los contratistas han ganado licitaciones, deben realizar las tareas que se les asignan, y generar informes sobre los avances de estas tareas y sus resultados finales. Los agentes contratistas pueden convertirse en gerentes gestores si la sub-tarea es demasiado compleja para sus habilidades, entonces ellos se subdividen y se repite la sub-tarea y el proceso de asignación de tarea. En resumen, el protocolo utiliza un número de mensajes básicos [30]:

- Notificación de la tarea, enviado por el agente administrador para anunciar la disponibilidad de una tarea y la necesidad de contratistas para realizarla
- Oferta, enviado por los agentes contratistas cuando están disponibles para realizar una tarea
- Ganador, enviado por el agente contratista afirmando que ahora es responsable de la tarea

- Informe, presentado por el agente contratista para el agente administrador / gerente indicando sí o no se ejecuta la tarea con éxito y los resultados (también, envía los resultados parciales).

Estos mensajes proporcionan la funcionalidad necesaria para hacer que los contratos se cumplan entre los agentes de ambos roles [30].

2.1.1.3 El Mecanismo de planificación en SMA

Particularmente, en la planificación multiagente puede haber varios agentes planificadores, varios agentes responsables de la asignación de los planes, así como varios ejecutores. Generalmente, los modelos de planificación en los SMA son [1]:

Planificación centralizada para planes distribuidos (PCpD): es un proceso de planificación centralizado que genera planes individuales para un grupo de agentes, en los cuales la asignación de planes depende de los agentes que están disponibles en un momento dado, y de sus capacidades. En este caso se realiza un plan general (el planificador) y se descompone en sub-planes, para asignarlos a los agentes según sus capacidades y disponibilidades, ver figura 2.1 Se supone que existe un solo agente planificador, capaz de planear y organizar las acciones. El resto de agentes simplemente ejecutan los planes asignados. También, puede que quien asigne los sub-planes al resto de los agentes no sea el mismo que planificó.

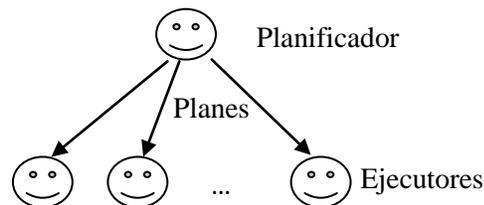


Figura 2.1. Planificación centralizada para planes distribuidos

Planificación distribuida para un plan centralizado (PDpC): Un grupo de agentes coopera para formar un plan centralizado. Para lograr esto es necesario que exista una fusión de todos los planes parciales hechos por las agentes planificadores, ver figura 2.2. La fusión de planes multiagente consiste en [1]: Cada agente crea un plan que satisface su propio objetivo, luego, los planes son analizados para detectar y resolver conflictos entre ellos, y algunas veces también, para explotar interacciones positivas. Sin embargo, los agentes que forman el plan no lo ejecutan, su papel es simplemente generar el plan. Algún agente del SMA lo ejecutará (eventualmente, alguno de ellos).

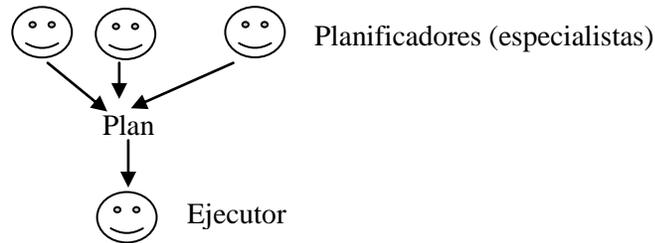


Figura 2.2. Planificación distribuida para un plan centralizado

Planificación distribuida para planes distribuidos (PDpD): un grupo de agentes posee planes de acción individuales. No existe ningún agente centralizador, ni para planear planes globales, ni para coordinar planes parciales, ni para ejecutar los planes, ver figura 2.3 En la planificación distribuida cada agente planea individualmente las acciones que se deben realizar, en función de unos objetivos. La dificultad se refiere a resolver los conflictos potenciales que pueden presentarse en la ejecución de los planes (por ejemplo por el uso de recursos), como también en reconocer las situaciones sinérgicas que se pueden presentar cuando las acciones de unos pueden ser útiles a los otros.

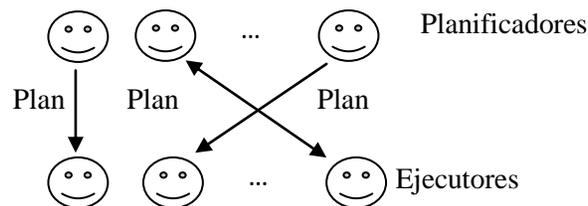


Figura 2.3. Planificación distribuida para planes distribuidos

2.1.2 Aprendizaje colectivo en SMA

En años recientes, los SMA han tenido gran atención en la comunidad de la inteligencia artificial. El estudio de los SMA involucra la investigación sobre comportamientos autónomos, flexibles y racionales de entidades tales como, programas de software o robots, así como sus interacciones y coordinación. Al diseñar un sistema de agentes, es imposible prever todas las situaciones potenciales que un agente pueda encontrar, y así especificar un comportamiento óptimo. Por esto, los agentes deben ser capaces de aprender y adaptarse a su entorno [31]. Tanto los agentes como los SMA, pueden ser vistos como otro dominio de aplicación de los sistemas de máquinas de aprendizaje, donde el aprendizaje puede darse tanto a nivel individual como a nivel colectivo. En los SMA se plantea el problema de aprendizaje distribuido/colectivo, i. e., varios agentes aprendiendo separadamente para adquirir una hipótesis conjunta. En el aprendizaje distribuido los agentes necesitan cooperar y comunicarse para aprender efectivamente.

Aprendizaje de un Agente Individual vs Aprendizaje Multi-Agente

El aprendizaje de un agente individual se enfoca en cómo éste mejora sus habilidades, independientemente del dominio en el que se desenvuelve. Ahora bien, no podemos hablar de aprendizaje multiagente si lo que un agente aprende no afecta ni es afectado por otros agentes vecinos. Pero ¿un agente perteneciente a un entorno multiagente puede aprender sin afectar o ser afectado por otros agentes? Aun cuando un agente no está consciente de otros agentes, este los percibe como parte de su entorno y su comportamiento será parte de lo aprendido [28]. Por otra parte, el aprendizaje de un agente individual puede no siempre alcanzar comportamientos óptimos en entornos multiagente, y puede haber dominios donde el aprendizaje multiagente coordinado es una metáfora más natural y mejora la efectividad.

2.1.2.1 Algunas técnicas de aprendizaje colectivo

En esta sección definiremos las diferentes técnicas de aprendizaje colectivo, que tiene como característica de base que involucra a múltiples agentes, donde los espacios de búsqueda pueden ser inusualmente grandes, y debido a la interacción de los agentes, pequeños cambios en el comportamiento a menudo pueden generar cambios impredecibles en el resultado a nivel macro (“emergente”), es decir, el SMA como un todo [31].

Aprendizaje reactivo

En sistemas reactivos, el comportamiento general emerge de las interacciones de los componentes. En lugar de diseñar protocolos de coordinación, o proveer agentes con complejos modelos de reconocimiento, los individuos asumen trabajar sobre valores de información del entorno (e. g., la distancia que deben mantener con sus vecinos) que producen un comportamiento social. Dado que se evita el procesamiento interno, esta técnica permite a los sistemas de agentes responder a cambios en el entorno en el momento oportuno. En el aprendizaje reforzado, los agentes reactivos poseen una descripción del estado actual y tienen que escoger la siguiente acción tal que maximicen un valor escalar reforzado el cual reciben después de cada acción [31]. Así, la tarea de los agentes es aprender de manera indirecta, basados en recompensas retardadas, para escoger las acciones que produzcan un acumulativo considerable de recompensas. Esto puede traer como efecto secundario, que los agentes se encuentren despojados del conocimiento del dominio, lo que es esencial para tomar las acciones correctas en escenarios dinámicos y complejos.

Aprendizaje basado en lógica

Una herramienta que permite superar esta limitación, es el método de programación lógica inductiva (IPL, por sus siglas en inglés) [31]. Este método calcula una hipótesis inductiva con la ayuda de datos de entrenamiento (ejemplos positivos y negativos) y conocimiento previo. Los agentes recogen muestras (ejemplos) de acciones y planes ejecutados en la comunidad, es decir del colectivo, con sus respectivos resultados. Estos datos, junto con la base del conocimiento del dominio y objetivos como predicados, forman la base del

proceso de aprendizaje inductivo, el cual genera una hipótesis (i. e., una definición del objetivo predicado), que puede ser utilizado en sus planes subsecuentes. Los objetivos predicados vienen dados por el diseñador del sistema. Una vez que cierto número de muestras sean clasificadas de manera incorrecta (i. e., los agentes cometen un cierto número de errores en sus predicciones de acciones resultantes), una nueva hipótesis será generada basada en el extenso conjunto de datos o muestras para entrenamiento. Es decir, el aprendizaje de cada individuo está basado en las experiencias de acciones y resultados previos, que se encuentran en el entorno o en la comunidad de cada uno de ellos.

Aprendizaje social

El aprendizaje en SMA, generalmente ha sido etiquetado como aprendizaje social. Este aprendizaje se basa en que los agentes aprenden del comportamiento de otros agentes más experimentados. Es decir, considérese un SMA cuya población esté formada por individuos experimentados, y esta comunidad de agentes sea explorada por nuevos agentes. En un primer momento, los nuevos agentes llegan prácticamente en blanco, ya que no han tenido la oportunidad de aprender de su entorno. A pesar de esto, los nuevos agentes no necesitan aprender todo lo que necesitan por sí solos: es posible que se beneficien del aprendizaje acumulado por la población de agentes más experimentados. Esta situación, la podemos visualizar en los escenarios de internet, donde agentes de software altamente autónomos podrían ser nuevos agentes, y al principio no habrán aprendido cual motor de búsqueda utilizar primero, o cual página de subasta posee la mejor oferta [32]. Ese agente probablemente comenzará utilizando las páginas o motores más utilizadas por sus vecinos.

Existen diferentes mecanismos de aprendizaje social, algunos de ellos son:

- Comportamiento contagioso: los estímulos producidos por un comportamiento particular de un agente, sirven como disparo para que otros se comporten de la misma manera.
- Seguir una conducta o comportamiento: el cual queda ejemplificado por “seguir a alguien mayor y luego aprender de lo que sucede”. La diferencia con el mecanismo anterior es que aquí el agente no necesita de estímulo alguno. 
- Aprendizaje por observación: una manera de ejemplificarlo es “preste atención a lo que otros hacen o experimentan, y si los resultados para ellos son buenos o malos, entonces aprenda de eso”

Modelos de transmisión cultural y transmisión genética entre generaciones, ayudan a delinear las condiciones bajo las cuales es ventajoso el aprendizaje desde otros individuos, en lugar de lograrlo por sí mismos. 

2.2 Algoritmos culturales (AC)

Los métodos de la computación evolutiva (EC) han sido exitosos en la resolución de diversos problemas de búsqueda y optimización, aún más en situaciones con poco o ningún conocimiento del dominio [10]. Sin embargo, ellos pueden mejorar su ejecución cuando se utiliza un conocimiento específico del problema para guiarlos en su resolución. Por otro lado, en las sociedades humanas, la cultura puede ser un medio para almacenar información, de manera tal que impacte a los individuos que la componen. Esta información es accesible completamente a todos los miembros de la sociedad. Así, la cultura es de gran utilidad para guiar la solución de problemas a través de la interacción social de los individuos en la población [10].

Basándose en estos fundamentos culturales, Robert Reynolds propuso un tipo de algoritmo evolutivo, en el que el conocimiento del dominio no se integra “*a priori*” (de manera previa) a la técnica, sino que se extrae durante el mismo proceso de búsqueda: estos son los algoritmos culturales [12]. Los AC fueron desarrollados por Robert G. Reynolds, como un complemento a la metáfora que usan los algoritmos de computación evolutiva, que se habían concentrado en conceptos genéticos y de selección natural. Estos están basados en las teorías de algunos sociólogos y arqueólogos sobre la evolución cultural [12]. Tales investigadores indican que la evolución cultural puede ser vista como un proceso de herencia a dos niveles: el nivel micro-evolutivo, que consiste en el material genético heredado por los padres a sus descendientes, y el nivel macro-evolutivo, que es el conocimiento adquirido por los individuos a través de las generaciones, y que una vez codificado y almacenado, sirve para guiar el comportamiento de los individuos que pertenecen a una población [12].

Reynolds intenta captar ese fenómeno de herencia doble en los AC. Estos algoritmos operan en dos espacios. Primero el espacio de la población, como en todos los métodos de computación evolutiva, en el que se tiene un conjunto de individuos. Cada individuo tiene un conjunto de características independientes de los otros, con la que es posible determinar su aptitud. A través del tiempo, tales individuos podrán ser reemplazados por algunos de sus descendientes, obtenidos a partir de un conjunto de operadores aplicados a la población. El segundo espacio es el de creencias, donde se almacenarán los conocimientos que han adquirido los individuos en generaciones anteriores. Este espacio se divide a su vez en cinco dominios, que son [10]:

- Conocimiento normativo: colección de rangos de valores deseables para los individuos de la población, por ejemplo, comportamiento aceptable para un agente en la población
- Conocimiento del dominio específico: información acerca del dominio del problema

- Conocimiento situacional: ejemplos específicos de eventos importantes, por ejemplo, soluciones exitosas y no exitosas
- Conocimiento temporal: historial del espacio de búsqueda (registra lo recorrido del espacio de búsqueda), por ejemplo, patrones temporales del proceso de búsqueda
- Conocimiento espacial: información acerca de la topografía del entorno o espacio de búsqueda.

La información contenida en este espacio debe ser accesible a cualquier individuo, quien pueda utilizarla para modificar su comportamiento. Para unir ambos espacios se establece un protocolo de comunicación, que dicta las reglas de intercambio y el tipo de información que se puede transferir entre los espacios.

Un AC formalmente se denota de la siguiente manera:

$$AC = (P, S, V_C, f, B, Acepte, Ajuste, Influencia)$$

donde: P representa la población; S es el operador de selección; V_C son operadores de variación (e. g., los operadores genéticos de cruce y mutación); f es la función objetivo o de aptitud; B es el espacio de creencias; $Acepte$ es la función de aceptación, la cual está encargada de aceptar los mejores individuos de la población (algunos autores han determinado que debería ser el 20%), para adquirir sus experiencias y llevarlas al espacio de creencias; $Ajuste$ es un operador del espacio de creencias (B), que actualiza el conocimiento en él; e $Influencia$ es un conjunto de funciones de influencias que determinan como el espacio de creencias influye en la población (normalmente eso se consigue modificando o introduciendo nuevos operadores de variación V_C , en este caso, usando la información contenida en el espacio de creencias). $Acepte$ e $Influencia$ juntos representan el protocolo de comunicación del AC. Los AC no se rigen por un protocolo en particular, ellos son particulares para cada problema a estudiar. En la figura 2.4 se muestra como está estructurado un algoritmo cultural, y la relación que existe entre sus componentes.

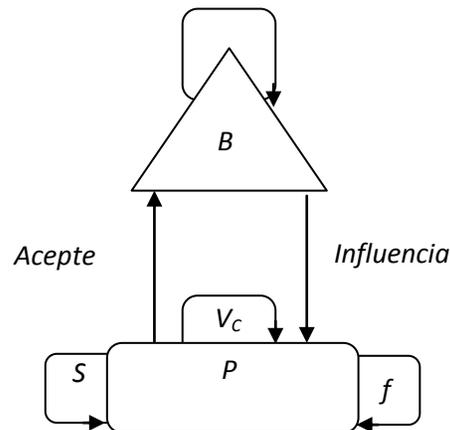


Figura 2.4. Estructura de un AC.

El pseudo-código de un AC se muestra a continuación [8].

Begin

$t = 0;$

Iniciar P^t

Iniciar B^t

Repetir

 Evaluar P^t

 Ajuste (B^t , Acepte (P^t))

 Variación (P^t , influencia (B^t))

$t = t + 1;$

 Seleccionar P^t de P^{t-1}

Hasta (haber alcanzado la condición de finalizar)

End

En el pseudo-código, t es el tiempo inicial, P^t y B^t son la población y el espacio de creencias al tiempo t ; luego, el algoritmo entra en un bucle en donde se evalúa a los individuos de la población, se actualiza el espacio de creencias con las nuevas experiencias seleccionadas de los individuos (a través de la función aceptación), y se influencia a la población con este conocimiento adquirido; finalmente, el bucle termina con una condición de parada (generalmente un número máximo de generaciones).

Para modelar la población se pueden utilizar los modelos clásicos de la computación evolutiva como los algoritmos genéticos, la programación evolutiva, la programación genética entre otros. De la misma manera para modelar el espacio de creencias se pueden utilizar redes semánticas, mapas conceptuales, modelos basados en lógica o reglas, entre otros. Los modelos utilizados por Reynolds, para representar la población en diferentes trabajos han sido [12]:

- Algoritmos Genéticos en problemas de aprendizaje y optimización
- Programación Genética para definir estrategias en agentes
- Estrategias Evolutivas en juego de fútbol con robots
- Modelos Meméticos⁴ para modelar la evolución de la agricultura

Los modelos utilizados por Reynolds para representar el espacio de creencias, en diferentes trabajos han sido [12]

- Esquemas

Valores binarios para el problema de Boole, minería de datos

⁴ Los modelos meméticos están basados en una protociencia incluida en el campo de la sociología, un acercamiento a la propuesta de los modelos de evolución de transferencia de información cultural basado en el concepto de meme, el cual representa una unidad de evolución cultural humana análoga a los genes, argumentando que la replicación también ocurre en la cultura en un sentido diferente.

Esquematas de intervalos de valores reales para optimización sin restricción

Esquematas regional para optimización restringida

- Redes semánticas
- Modelos gráficos 
- Modelos basados en reglas y lógicas para detección de fraudes

2.3 Automatización industrial

La sustitución en los procesos técnicos, del quehacer humano por equipos mecánicos, eléctricos, electrónicos o sistemas con el fin de optimizar el uso de los recursos existentes y la continuidad de los procesos, recibe el nombre de automatización [1].

Según [1], la Automatización Industrial (AI) es un conjunto de técnicas que involucran la aplicación e integración de sistemas mecánicos, eléctricos, electrónicos, fluidos, soportados por tecnología de información y comunicación (TIC), todos ellos unidos para operar, supervisar y controlar diferentes tipos de procesos técnicos de forma autónoma. Así, la AI es un área en donde confluyen diferentes disciplinas para la solución de diferentes problemas industriales.

La automatización debe proveer una infraestructura que cubra todas las fases y aspectos del proceso productivo, con el fin de lograr un incremento de la producción, manteniendo o reduciendo de los costos, por esta razón, en estos modelos se encuentran presentes todas las actividades inherentes al proceso productivo: Optimización, Planificación, Supervisión, y Control. En función del proceso industrial se organizan e integran estas actividades, definiéndose para ello diferentes arquitecturas (jerárquicas, heterárquicas e inclusive híbridas) [1].

Modelos Jerárquicos

Dentro del contexto de automatización la palabra jerarquía, o sistema jerárquico, debe ser interpretada como un sistema de subsistemas débilmente interrelacionados, cada uno pudiendo ser a su vez jerárquicos hasta llegar a algún nivel más bajo de subsistemas elementales. En arquitecturas jerárquicas, los diferentes niveles no pueden tomar la iniciativa, tienen tareas muy precisas, de esta manera su autonomía y reactividad ante disturbios son débiles. La arquitectura resultante es muy rígida, y por tanto, costosa de desarrollar y difícil de mantener. La modificación de las estructuras automatizadas para agregar, quitar o cambiar recursos es difícil, ya que requiere la actualización de todos los niveles. Además, los fallos ocurridos en niveles inferiores se propagan hacia los superiores, afectando el funcionamiento de las demás actividades inherentes a la automatización [33].

Entre sus ventajas se pueden nombrar las siguientes: Los modelos jerárquicos disuelven el vínculo entre el tamaño y la complejidad, en virtud de la jerarquía, siendo la complejidad de una organización casi independiente de su tamaño total. Además, se reduce la necesidad

de transmisión de información entre los diferentes elementos que conforman la organización. Un nivel solo necesita información detallada sobre las actividades correspondientes a su nivel, e información sumaria adicional sobre el comportamiento medio en otros niveles. Es suficiente saber cuáles son las salidas que el subsistema debe proporcionar a otros entes, sin importar a que procesos serían sometidas por este último [33].

Como ejemplo clásico de estos modelos jerárquicos, está el modelo de Automatización Piramidal, el cual consta de cinco niveles que abarcan las diferentes funciones de una planta coordinada de manera jerárquica (figura 2.5), cubriendo desde los aspectos de control de los procesos físicos en su nivel más bajo, hasta los niveles donde se realizan las funciones corporativas de la planta. Cada nivel se caracteriza por un tipo de información y de procesamiento diferente, siendo necesaria la integración para manejar la comunicación en cada nivel y entre niveles, con el fin de lograr sistemas que permitan ejecutar las diferentes tareas en una empresa. Este es el modelo de automatización más difundido en el ambiente de producción continua por la ISO, y sus 5 niveles son [33]:

- Empresa: Nivel más alto, donde se fijan las estrategias y metas a cumplir.
- Planificación: se realiza la planificación acorde con las estrategias fijadas, así como la gestión de la producción.
- Supervisión: se realiza el monitoreo de las diferentes unidades de producción, y se verifica la sincronización de los procesos interdependientes.
- Control Local: se compone de todos los elementos destinados al control regulatorio de los procesos.
- Proceso: abarca tanto el fenómeno físico como los instrumentos terminales en contacto con los mismos.

El esquema de actividades por niveles permite estandarizar procedimientos y soluciones a problemas. En este modelo, la granularidad de la información crece a medida que se va bajando en los niveles, y la toma de decisiones es jerárquica [33].

Modelos heterárquicos

Se basan en la fragmentación del sistema en pequeñas unidades completamente autónomas. Las arquitecturas heterárquicas se basan en una total autonomía local, resultando en un entorno en el cual los componentes cooperan para alcanzar objetivos globales gracias a la toma de decisiones locales (control distribuido).

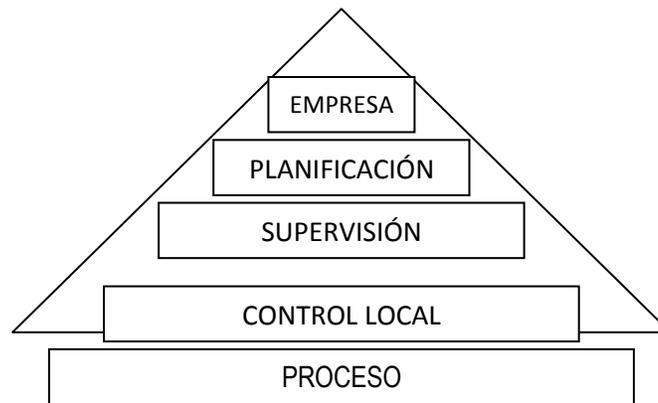


Figura 2.5. Modelo de Automatización Piramidal

Estos componentes autónomos pueden ser agentes u holones y la cooperación se estructura a través de protocolos de negociación [1]. El enfoque heterárquico prohíbe todo tipo de jerarquía, lo que le da cierta autonomía a los componentes y obliga al trabajo colaborativo entre ellos. Así, al eliminar las relaciones de jerarquía en el sistema los componentes cooperan como iguales, dando lugar a una arquitectura plana en lugar de asignar relaciones de subordinación y de supervisión. Para proveer la robustez y flexibilidad necesaria se establecen esquemas de comunicación y cooperación, tal que al existir fallas en un componente otro pueda ejecutar sus actividades [33]. Normalmente, la Automatización Inteligente se asocia al modelo heterárquico, para lo cual se basa en los conceptos de agentes u holones. Un ejemplo de modelos de Automatización Inteligente, son SADIA y PABADIS, los cuales son presentados en detalles más adelante. Tienen un buen desempeño ante cambios y pueden adaptarse continuamente a su entorno.

Particularmente, un modelo de Automatización Inteligente basado en holones consiste en la descentralización de las decisiones por medio de la atribución de cierto nivel de independencia a cada unidad que compone la arquitectura. Por ejemplo PROSA (Process Resource Order Staff Architecture) básicamente se compone de tres unidades semi-dependientes y cooperantes denominadas holones (ver figura 2.6), con capacidad de tomar decisiones y ejecutar acciones, las cuales son [1]: holón de recurso, que representa los recursos disponibles, holón de producto, que representa los productos generados y holón orden de trabajo, que establece las órdenes de trabajo.

En general, un sistema de fabricación holónico es una holarquía⁵ que integra un rango completo de actividades de fabricación, desde reserva de pedidos hasta diseño, producción, control y mercadeo, para obtener una empresa de fabricación ágil.

⁵ Son organizaciones que permiten la construcción de sistemas muy complejos que son, no obstante, eficientes en el uso de recursos, resistentes a perturbaciones (tanto internas como externas), y adaptables a cambios en su entorno.

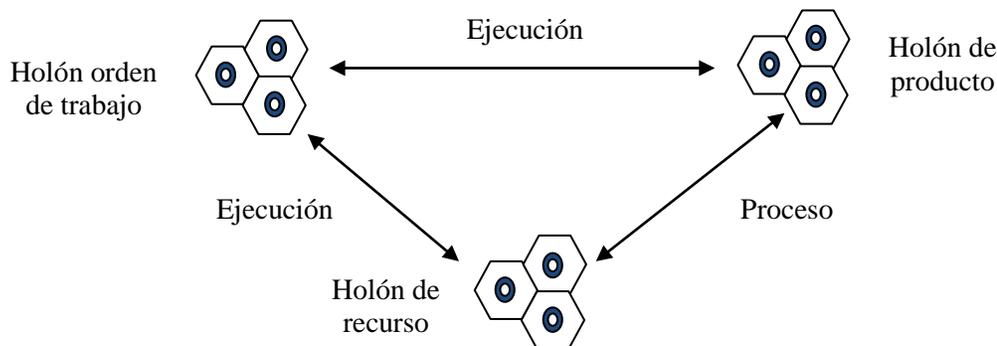


Figura 2.6. PROSA

La potencia de las organizaciones holónicas, u holarquías, es que permiten la construcción de sistemas muy complejos que son, no obstante, eficientes en el uso de recursos, resistentes a perturbaciones (internas y externas), y adaptables a cambios en su entorno. Los holones pueden participar en múltiples holarquías al mismo tiempo. La idea de los sistemas de integración holónica es proporcionar procesos dinámicos y descentralizados, en los cuales los holones son la base del procesamiento, y en donde los distintos problemas pueden ser resueltos a través de la integración y distribución de los diferentes componentes de un sistema utilizando mecanismos de agregación y desagregación [1]. Se ha estado aplicando la tecnología de holones en la integración de empresas de fabricación, en la gestión de cadenas de suministro, en la planificación de la producción, para la asignación de recursos, entre otras cosas [1].

Modelos Híbridos

Constituido por aquellos modelos que combinan las ventajas de los sistemas jerárquicos y heterárquicos al mismo tiempo, evitando sus desventajas. Algunos autores colocan dentro de esta clasificación las arquitecturas holónicas antes nombradas, caracterizadas por:

- Otorgar autonomía a los módulos individuales para evitar las estructuras rígidas de los sistemas jerárquicos. Esto dota al sistema de un mecanismo para generar respuestas rápidas ante perturbaciones y la habilidad para reconfigurarse a sí mismo ante nuevos requerimientos.
- Los elementos de la arquitectura pueden pertenecer a múltiples jerarquías u holarquías, o formar temporalmente una.
- La existencia de jerarquías flexibles asegura el desempeño controlable y predecible de la arquitectura 

Independientemente de la arquitectura utilizada, la automatización debe permitir implantar un sistema capaz de realizar la captación, adquisición, control y supervisión de todos los elementos, procesos y servicios llevados a cabo por las industrias, fábricas y empresas, posibilitando al personal encargado la monitorización de los procesos. El desarrollo de la AI ha estado ligado al desarrollo de los sistemas computacionales. Inicialmente, la automatización solo ocurría a nivel de máquinas y supervisión, siendo incorporada posteriormente al proceso de planificación y optimización, dando cabida a nuevos modelos de automatización [1].

2.3.1 Integración en automatización

Una definición de integración propuesta en [1], presenta la integración como la coordinación de las operaciones de todos los elementos de la empresa que trabajan en forma conjunta para lograr el cumplimiento, en forma óptima, de la misión de la empresa. Fundamentalmente, la integración se basa en el desarrollo de modelos de intercambio de información, de cooperación y de representación común de datos, para poder coordinar y ejecutar las acciones u objetivos planteados.

En particular, podemos definir la integración en dos importantes arquitecturas: integración orientada a datos (DOI, por sus siglas en inglés) e integración orientada a servicios (SOI, por sus siglas en inglés). DOI involucra la combinación de datos que residen en diferentes fuentes y plataformas, proveyendo a los usuarios una vista unificada de ellos [34]. Este proceso es significativo en varias situaciones, en el caso comercial cuando compañías similares necesitan unir sus bases de datos, y científicamente para combinar y compartir resultados de investigaciones diferentes e. g., provenientes de repositorios bioinformáticos. Es por ello que, en la integración los elementos fundamentales son: los productores de datos, la infraestructura de comunicación, y los mecanismos de procesamiento de datos; componiendo así los aspectos necesarios en las estrategias para la automatización integrada.

Por su parte, la integración basada en una arquitectura orientada a servicios está definida por [35] como la integración de entidades computacionales que utilizan servicios, basado en la arquitectura orientada a servicios (SOA, por sus siglas en inglés). SOA es una arquitectura de software basada en piezas de software, las cuales proveen funcionalidades a aplicaciones como servicios. Un servicio es una representación lógica autónoma de una función o actividad repetible [35]. Los servicios pueden ser combinados con otro software de aplicación, y juntos proveer una funcionalidad completa dando lugar a un macro software de aplicación. Por otra parte, SOI maneja los problemas legales y sistemas heterogéneos flexiblemente, permitiendo a organizaciones TIC ofrecer aplicaciones existentes como servicios reusables. En contraste con la tradicional integración de aplicaciones empresariales (EAI), las características de base de SOI son:

- Interfaces estandarizadas bien definidas, tal que los consumidores disponen de un consistente acceso a los servicios subyacentes.

- Opacidad - La tecnología y la ubicación de la aplicación que proporciona la funcionalidad están ocultas detrás de la interfaz de servicio. De hecho, no hay necesidad de un proveedor de servicios fijo.
- Flexibilidad - Tanto los proveedores de servicios y consumidores de servicios pueden cambiar - la descripción del servicio es la única constante. Tanto  proveedor como el consumidor deben continuar apegados a la descripción del servicio, para que las aplicaciones continúen funcionando.

2.3.2 *Sistemas multi-agente en automatización*

En la industria, la necesidad de adoptar diferentes tecnologías, a fin de responder a las demandas del mercado, ha hecho que la complejidad de los procesos de producción se incremente, requiriendo, además, la integración de todos los niveles de la pirámide de automatización. Este incremento de la complejidad está basado en los cambios constantes y drásticos, debido a la aparición de eventos inesperados, cambios impredecibles en los proveedores y consumidores, posicionamiento en el mercado ante altos niveles de competitividad y, en general, cambios en los paradigmas de producción. El concepto de reconfigurabilidad surge como una nueva característica que debe tener la industria para responder adecuadamente a estos cambios. Para desarrollar el concepto de reconfigurabilidad en la industria, es necesario garantizar la comunicación entre los diferentes niveles de automatización, tanto en la línea vertical como en la horizontal, lo que permite el intercambio oportuno de la información entre dichos niveles. Esta necesidad de flexibilidad en la comunicación hace que los agentes sean los entes capaces de satisfacer este requerimiento, principalmente debido a la posibilidad de [1]:

- Reducción de la carga de la red industrial, puesto que cada agente evalúa la información localmente, sin necesidad de estar conectados con los sistemas del nivel de planificación (Enterprise Resources Planning, ERP).
- Conocimiento local de la planta por cada agente, a partir de donde actúa, integrando sus decisiones con la de los otros agentes.
- Adaptación de las características de un agente o grupo de agentes determinado, según los requerimientos del entorno, llevando así a cabo la reconfiguración del sistema.

En la literatura existen muchos trabajos sobre Automatización Industrial basada en agentes, en este apartado solo nombraremos algunos, para una presentación más detallada al respecto puede verse el libro [1]. A continuación se presentan los siguientes trabajos, debido a que los casos de estudios se basan en algunos aspectos presentes en ellos.

En [36] se propone una arquitectura basada en SMA que permite la toma de decisiones aprovechando las capacidades de negociación de los agentes, para responder en forma dinámica a los cambios en las diferentes actividades de manufactura (integración basada en

servicios). El sistema tiene la capacidad de monitorear todas las actividades del proceso de producción, ejecutar simulaciones en tiempo real, analizar y planificar las actividades de producción, y mantener un sistema de alarma siempre activo. Las principales funciones de este SMA son:

- Generación, basada en tiempo, del programa de operaciones y producción.
- Monitoreo y control activo de la ejecución de las actividades de la planta, incluyendo el control de las operaciones para alcanzar la programación planificada, incorporando capacidades de detección de condiciones anormales, para luego identificar alternativas de operación factibles.
- Evaluación a tiempo real de las actividades de producción, tanto en efectividad como en desempeño.

Otros ejemplos de arquitectura para Automatización Industrial basada en SMA son SADIA y PABADIS. Como vimos antes, SADIA propone un conjunto de niveles para definir un proceso industrial, donde cada nivel es descrito por una comunidad de agentes, los cuales son compuestos por los agentes del nivel inferior. En particular, el modelo de referencia SADIA está compuesto por tres niveles de abstracción (ver figura 2.7) [22].

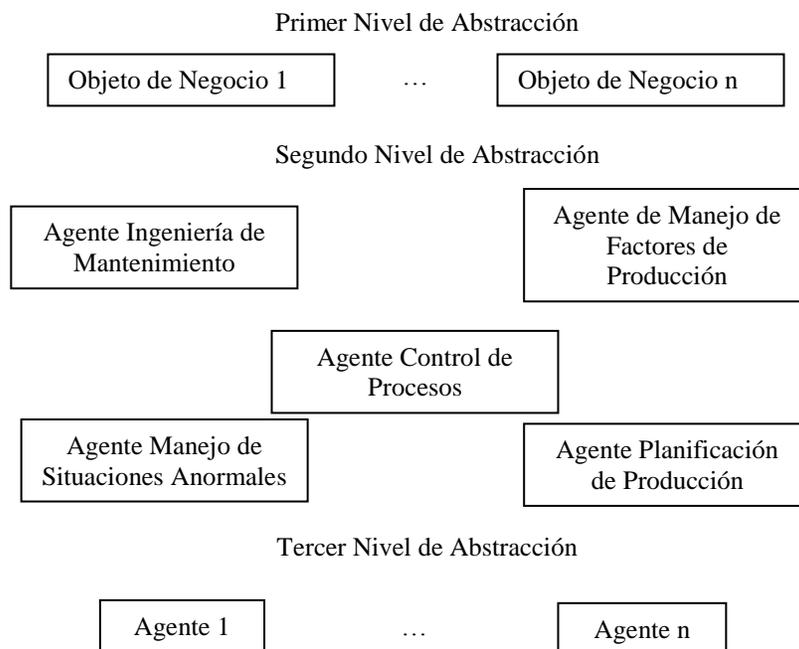


Figura 2.7. SADIA

En el primer nivel se modelan los objetos del negocio como agentes. El segundo nivel se encuentra compuesto por agentes que se ocupan de las actividades necesarias en todo proceso de automatización industrial: control de proceso, ingeniería de mantenimiento, manejo de situaciones anormales, manejo de los factores de producción y planificación de la producción. En el tercer nivel de abstracción se hace uso del marco de referencia

“Sistema de Control Distribuido Inteligente basado en Agentes” (SCDIA) [37] para modelarlo, así, cada agente del segundo nivel de abstracción es modelado usando el SCDIA. El SCDIA permite especificar un sistema de control distribuido inteligente basado en agentes. Sus agentes colaboran entre sí para llevar a cabo las tareas de supervisión y control relacionadas con la Automatización Industrial. Como se dijo antes, dicha comunidad está compuesta por 5 agentes, que se asemejan a los elementos de un lazo de control cerrado, estos son: Agente Coordinador, Agente Controlador, Agente de Medición, Agente de Actuación y Agente Especializado (ver figura 2.8).

Este modelo (SADIA) apunta a la integración basada en servicios, ya que modela todo un sistema de automatización industrial distribuido orientado a servicios, abarcando desde el proceso productivo (como objetos de negocio) hasta una comunidad de agentes encargada del control y la gestión de servicios.

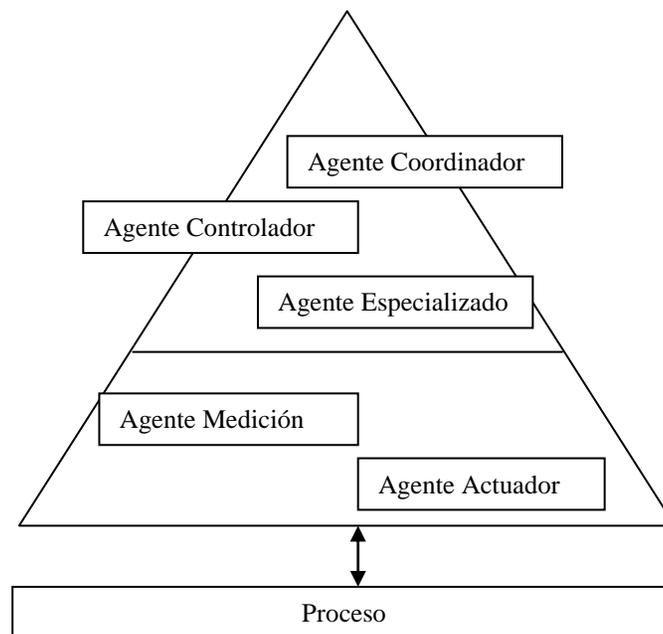


Figura 2.8. SCDIA

En el caso de PABADIS (Plant Automation based on Distributed Systems, por sus siglas en inglés), permite desarrollar sistemas MES (Manufacturing Execution Systems) distribuidos basados en la tecnología de agentes [1]. Esta arquitectura está basada en la automatización piramidal, pero utilizando sistemas distribuidos y agentes inteligentes, se reduce la jerarquía a dos capas, disuelve la capa de supervisión y divide su funcionalidad en una parte centralizada que puede ser ubicada dentro del sistema de planificación y una parte descentralizada que puede ser implementada por agentes móviles. Cada agente está asociado a una pieza de trabajo que tiene la información necesaria del producto, y se mueve a través de la planta de la misma forma como lo hace la pieza física. Esta información puede ser relativa a:

- Secuencia y programación de producción

- Datos de producción de las máquinas asociadas a las piezas
- Estado de procesamiento
- Información administrativa sobre la orden
- Uso de los recursos
- Información del control de calidad

CAPÍTULO III

INTEGRACIÓN EN AUTOMATIZACIÓN

3.1 La automatización desde la teoría de agentes

Según [38], la palabra automatizar significa “Técnica que hace que un dispositivo, un proceso, o un sistema opere automáticamente”. Esto nos lleva a la definición de [1] la cual dice que es la sustitución, en los procesos técnicos, del que hacer humano por sistemas mecánicos, eléctricos, electrónicos, con el fin de optimizar el uso de recursos existentes y la continuidad de los procesos (operatividad permanente del proceso). Se puede decir entonces que un sistema automatizado es aquel que es capaz de ejecutar tareas repetitivas, y de controlar operaciones, con la mínima intervención de un operador humano, permitiendo, a su vez, la disposición de elementos para supervisar y controlar cada uno de los procesos dentro de la cadena de producción, almacenamiento y distribución, así como los diferentes servicios ofrecidos a los clientes por parte de las fábricas, empresas y proveedores. Todo esto implica la implantación de instrumentación, dispositivos de maniobra, máquinas mecánicas o hidráulicas móviles, displays o paneles, SCADAs y redes de comunicación [1]. La tendencia de estos sistemas automatizados apunta a arquitecturas planas (distribuidas), cuyos componentes inteligentes, de software y hardware heterogéneos, interactúan entre sí buscando alcanzar objetivos comunes que optimicen el desempeño de los procesos. Es importante también acotar que la integración de todos los componentes, es prioridad en dichos sistemas.

Dicho esto, comenzaremos a visualizar el tema de automatización desde la teoría de agentes. Para esto es necesario enmarcar el modelo de automatización con las características expuestas en el párrafo anterior en una arquitectura multiagente. Algunas de esas posibles arquitecturas fueron presentadas en el capítulo 2, como SADIA o PABADIS. Particularmente, en el caso de SADIA su objetivo es modelar una arquitectura de automatización como SMA. Es importante señalar antes de continuar, que las características de SADIA pueden ser extrapoladas a otros casos de automatización. SADIA está conformada por tres niveles de abstracción, donde cada nivel es visto como un SMA. La figura 3.1 tiene del lado izquierdo el modelo arquitectónico de SADIA, y del lado derecho un ejemplo de instanciación de SADIA en un proceso de producción de petróleo, en específico, el lazo de producción de pozos. El proceso de producción petrolera requiere un conjunto de instalaciones que permiten la extracción, tratamiento, distribución y transporte de hidrocarburos [11]. Así, del lado derecho, en el primer nivel, se encuentran los diferentes componentes/instalaciones (objetos del negocio) del proceso productivo (pozo, patios de tanques, plantas compresoras de gas, etc.) y en los siguientes niveles los agentes propuestos por SADIA (para una detallada presentación de esa arquitectura ver [22]).

La principal razón de haber escogido esta arquitectura, es porque ella contempla una integración, tanto vertical como horizontal, entre las diferentes comunidades de agentes (SMA) que la conforman. Es un modelo de integración orientada a servicios (SOI) bastante natural. La figura 3.1 muestra eso, una integración horizontal entre los agentes que conforman un nivel dado (por ejemplo, entre los agentes que representan los diferentes objetos de negocio del primer nivel, como el agente pozo, el agente MLAG, etc.), y una integración vertical entre los agentes que conforman niveles distintos (por ejemplo, entre el agente pozo y los agentes del segundo nivel de abstracción, tales como el agente de Ingeniería de Mantenimiento). Veamos en detalle varios ejemplos de integración

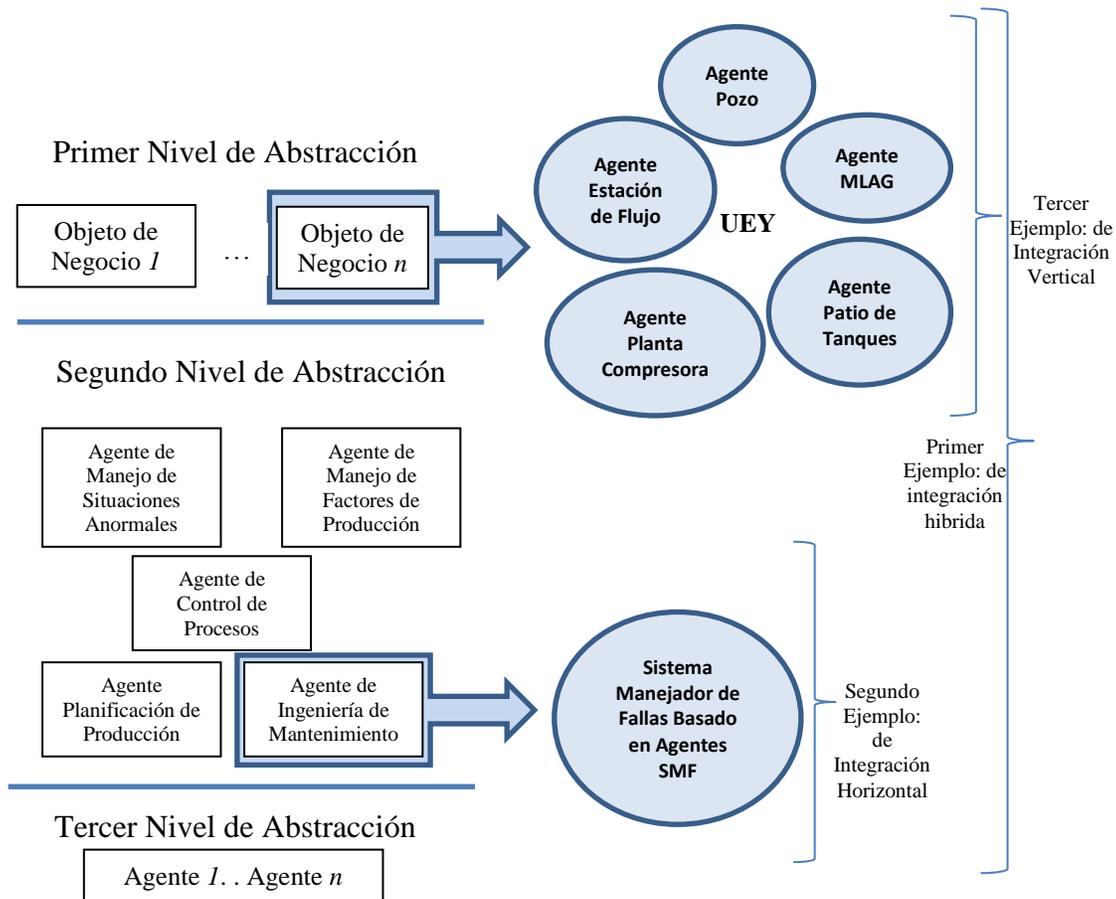


Figura 3.1. SADIA con ejemplos de instanciación

Un primer ejemplo de integración es la descripción del proceso de producción petrolera de pozos petroleros que se basan en el método de levantamiento artificial por gas (LAG). En este ejemplo, el lazo de producción petrolera se centra en la Unidad de Explotación de Yacimiento (UEY), y sus componentes principales son (ver [39] para los detalles de una UEY para pozos LAG): pozos, estaciones de flujo, patios de tanques, plantas compresoras de gas y múltiples. De esta manera, la comunidad de agentes del primer nivel de SADIA es: Agente Pozo, Agente Estación de Flujo, Agente Planta Compresora de Gas, Agente Múltiples de Levantamiento Artificial por Gas (MLAG), y Agente Patio de Tanques. Así,

en el primer nivel se presentan los objetos de negocio, en donde cada objeto de negocio puede representar cualquier agente de la UEY. Este ejemplo es general, ya que trata todo un lazo completo de un sistema de producción, en donde se dan conversaciones que permiten a los SMA realizar tareas a nivel de una unidad de producción, o entre unidades de producción, o entre unidades de producción y aplicaciones de soporte técnico (niveles inferiores). De esta manera se da tanto una integración vertical como horizontal global (ver sección 3.2 para más detalles).

El segundo ejemplo de integración se enfoca exclusivamente en analizar el agente de ingeniería de mantenimiento del segundo nivel. Este agente ejecuta tareas de detección, localización, diagnóstico y predicción de fallas en el sistema. Cabe recordar que cada objeto de negocio del primer nivel está integrado por los cinco agentes del segundo nivel de abstracción de SADIA, pero cada agente del segundo nivel es un SMA. Así, este ejemplo está enfocado en solo uno de esos agentes del segundo nivel, el agente de ingeniería de mantenimiento, lo cual representa un caso de integración horizontal las interacciones que se dan en la comunidad de agentes que lo constituyen (ver sección 3.2).

Finalmente, un último ejemplo de integración es cuando solo se consideran las interacciones globales entre los niveles de abstracción de SADIA. Este ejemplo caracteriza el proceso de integración vertical (ver sección 3.2 para más detalles).

3.2 Integración en automatización basada en la coordinación de sistemas multi-agente

En esta sección abordaremos el problema de integración vertical (IV), horizontal (IH) e híbrida, y como se pueden modelar a través de los mecanismos de coordinación de los SMA, y en particular, los que son usados en las conversaciones entre agentes. La figura 3.2 permite visualizar estas interacciones para el ejemplo anterior.

La figura 3.2 nos muestra algunas (no todas) de las posibles interacciones entre los diferentes SMA (flechas de doble dirección). Las siglas indican en donde se da la integración vertical (IV) y horizontal (IH). La integración horizontal se da entre los SMA del mismo nivel, es decir, cuando hay comunicación entre agentes del mismo nivel, mientras que la vertical se da entre los diferentes niveles de abstracción.

Ahora presentamos el problema de integración en automatización en ese modelo de agentes:

- La comunidad de agentes del tercer nivel debe ser capaz de comunicarse con el segundo nivel de abstracción de la arquitectura, es decir, debe poder intercambiar información con los agentes de control de procesos, ingeniería de mantenimiento, manejo de factores de producción, planificación y situaciones anormales como un todo, ya que estos representan las diferentes actividades a realizar para cumplir con las metas de los objetos de negocio.

que se da a nivel de los mecanismos de coordinación y comunicación usados para orquestar la ejecución de los servicios brindados por los agentes. De esta manera, la idea es proponer un modelo que permita a los agentes poder escoger colectivamente que mecanismo de coordinación resulta el más idóneo al momento en que ellos se comuniquen (en sus conversaciones).

Partiendo de ese análisis específico sobre SADIA, a continuación se plantean diferentes hipótesis acerca de los mecanismos de coordinación de las conversaciones a nivel general para cualquier SMA, con respecto al problema de la integración:

1ra Hipótesis: Los agentes pertenecientes a diferentes comunidades de agentes, al coordinarse en sus diferentes conversaciones, logran *integrarse tanto horizontal como verticalmente*, lo que les permite ejecutar plenamente sus tareas.

2da Hipótesis: Al garantizarse la comunicación entre los diferentes de agentes, sin importar el nivel de abstracción al que pertenecen (comunidades de agentes), se permite que los agentes hablen entre sí, enviando y recibiendo datos. Esto posibilita la *integración de datos*

3ra Hipótesis: Cualquier modelo de SMA Orientado a Servicios (como SADIA), al obtener los mecanismos de coordinación adecuados entre los agentes (bien sean de negociación o planificación), permite orquestar la ejecución de los servicios entre los agentes con un mínimo de conflicto entre ellos. Esto permite la *integración de servicios*.

4ta Hipótesis: Al trabajar conjuntamente todos los agentes que conforman el sistema automatizado, realizando cada uno sus tareas específicas, se permite la *interoperabilidad entre las comunidades*.

En general, la coordinación garantiza el buen desempeño de los sistemas computacionales de automatización basados en agentes, posibilitando la ejecución coherente de sus actividades. Los diversos mecanismos de coordinación existentes en la literatura, tales como los mecanismos de negociación (subasta, licitación, etc.) y de planificación [28], dan un amplio espectro de posibilidades para que estos sistemas se puedan integrar de forma óptima.

Estas hipótesis vinculadas al problema de integración en sistemas de automatización basados en SMA serán comprobadas en el capítulo siete, a través de diferentes casos de estudio. Esos casos permiten analizar las capacidades adaptativas, a nivel de los mecanismos de coordinación, de los SMA (propuesta principal de este trabajo).

CAPÍTULO IV

FORMALIZACIÓN DE LOS MECANISMOS DE COORDINACIÓN

En este capítulo presentamos el primer aporte fundamental de nuestro trabajo, la formalización matemática de los principales mecanismos de coordinación en los SMA. Este capítulo está basado en los trabajos [40, 41].

4.1 Modelo formal para el mecanismo de subasta

Supongamos un grupo de agentes $A = (a_s, A_i)$, donde a_s es el agente subastador y A_i es el conjunto de agentes ofertantes definido por $A_i = \{A_1, A_2, A_3, \dots, A_n\}$.

Se define una subasta mediante la tupla:

$$S = \langle C_0, Of_i^j, \vec{\varepsilon}_i, \alpha_i^j, C_p, C \rangle \quad (4.1)$$

- C_0 es el precio inicial de la subasta, $C_0 \in \mathfrak{R}^+$
- Of_i^j es una matriz de ofertas $\in \mathfrak{R}^{(n+1) \times m}$ conformada por el de número de agentes $i = \{1, \dots, n\}$ más un vector fila A_G adicional donde se muestra al ganador, y el número de rondas $j = \{1, \dots, m\}$.
- $\vec{\varepsilon}_i$ es un vector, donde cada elemento representa la cantidad máxima que puede ofertar cada agente i ofertante $\vec{\varepsilon}_i = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n\}$.
- α_i^j especifica una propuesta dada, siendo α el valor propuesto en la ronda j , e i el agente ofertante.
- C_p es la condición de parada de la subasta y puede ser de diferentes tipos:

$$C_p = \begin{cases} t \text{ (tiempo)}, \text{ si la subasta se rige por tiempo} \\ j \text{ (número de rondas)}, \text{ si se rige por rondas} \\ \alpha_i^j = 0 \forall i = 1, \dots, n \text{ cuando no hay más ofertas a partir de una ronda } j \text{ dada} \\ x_r \text{ (umbral)}, \text{ cuando se alcanza un precio mínimo umbral del subastador (para el caso de subasta Holandesa)} \end{cases}$$
- C es el precio final del recurso, este puede ser el valor máximo o mínimo del vector fila A_G , (para el caso de la subasta Holandesa es el mínimo), $C = A_G^j$, $C \in \mathfrak{R}^+$ y el ganador puede ser cualquier elemento del conjunto A_i , i. e., $\exists A_i : A_i = \text{ofrecio } A_G$.

Por otro lado, la matriz de ofertas (Of_i^j), almacena las diferentes propuestas de los agentes participantes y la mayor oferta (oferta ganadora) hecha en una ronda dada j por un agente i (ganador). Cada celda de la matriz contiene:

$$Of_i^j = C_0^j + \alpha_i^j$$

donde, $C_0^j = C_0$ si $j = 1$ y A_G^{j-1} si $j > 1$

Es decir, en la primera ronda cada celda representa el valor del precio inicial más la propuesta hecha por el agente; mientras que en las siguientes rondas ese valor será el valor ganador Of_i^j para una ronda j dada, el cual es A_G^j , más la propuesta hecha por cada agente.

Al final de la matriz se define el vector fila A_G , el cual identifica el ganador en cada ronda.

En la matriz se presentan las siguientes condiciones:

- El valor pagado por una agente i debe ser menor o igual a la cantidad máxima que puede ofertar el agente, i. e., $\forall A_i \max Of_i^j \leq \varepsilon_i$.
- Para las filas se cumple que el valor propuesto por un agente debe ser mayor a su propuesta anterior, $\alpha_i^j < \alpha_i^{j+1} \quad \forall i = 1, \dots, n$ y $\forall j = 1, \dots, m$ (caso Inglesa, One Shut, etc.), o menor (Caso Holandesa)
- Para cualquier agente en la siguiente ronda su propuesta debe mejorar al valor de la oferta ganadora en la ronda anterior, i. e., $A_G^j < \min(Of_i^{j+1})$ para el caso de la subasta Inglesa, etc. En el caso de la subasta Holandesa ocurre lo contrario ($A_G^j > \max(Of_i^{j+1})$), es decir, el valor debe siempre ser inferior al anterior, $\forall i = 1, \dots, n$ y $\forall j = 1, \dots, m$
- En una ronda dada, cada propuesta de un agente debe mejorar a la propuesta anterior de otro agente. Para ello debemos definir otra matriz, llamada matriz de orden Or , tal que si $Or_i^j < Or_k^j \Rightarrow Of_i^j < Of_k^j$, para el caso de la subasta Inglesa, etc. En el caso de la subasta Holandesa se tiene que si $Or_i^j < Or_k^j \Rightarrow Of_i^j > Of_k^j \quad \forall i, k = 1, \dots, n$ e $i \neq k$. Esta matriz Or_i^j , es una matriz dinámica que indica el orden en que los participantes proponen sus ofertas en una ronda dada. La dimensión de esta matriz es $i = \{1, \dots, n\}$ agentes por $j = \{1, \dots, m\}$ rondas.

Las condiciones para esta matriz son:

- No pueden haber ofertas simultáneas, i. e., $\forall i, k = 1, \dots, n$ y $\forall j = 1, \dots, m \quad Or_i^j \neq Or_k^j$
- Cualquier número de posición de orden en una ronda dada j debe ser menor o igual al número de ofertantes en la subasta, i. e., $Or_i^j \leq n$

A continuación se muestra en la tabla 4.1, el resumen de las variables de cada protocolo de subasta que estudiaremos en este trabajo: subasta Inglesa, subasta Holandesa y subasta de

Una Ronda (One Shut) (este último protocolo, a pesar de que no posee un modelo estándar FIPA, puede ser emulado por el protocolo de red de contrato de la FIPA [42] 

Tabla 4.1 Matriz resumen de variables de subastas

Subasta/ Variables	C_0	Of_i^j	ε_i	α_i^j	C_p	C
Subasta Inglesa	x	$x + \alpha_i^j$	$f(i)$	(+)	j, t	max
Subasta Holandesa	x	$x - \alpha_i^j$	-	(-)	t, x_r	min
Subasta de Una Ronda	x	α_i^j	$f(i)$	(+)	$j=1$	max

En la tabla 4.1 podemos observar que:

- C_0 en los protocolos de las subastas Inglesa y Holandesa cuando éstas superan una ronda, pasan a ser el valor ganador de A_G de cada ronda.
- Of_i^j en la subasta Inglesa muestra un incremento continuo, mientras que para la subasta Holandesa muestran un decremento continuo. Para la subasta de una ronda es una sola propuesta por cada agente.
- ε_i es sólo utilizada en los casos de la subasta Inglesa y de Una Ronda. Para la subasta Holandesa, este valor máximo no presenta riesgo de ser superado siempre y cuando sea menor al valor inicial, ya que éste se va reduciendo hasta un valor umbral mínimo (x_r).
- α_i^j para la subasta Inglesa y de Una Ronda es ascendente, y para la subasta Holandesa es descendente.
- C_p en la subasta Inglesa, aparte del número de rondas j y tiempo t , puede ser cuando no existan más ofertas ($\alpha_i^j = 0$). Para la subasta Holandesa puede darse en función del tiempo t , cuando no existan más ofertas o/y cuando se llegue al valor umbral mínimo x_r . Para el caso de la subasta de Una Ronda es sencillamente en base a una sola ronda.
- C (el precio final a pagar) para la subasta Inglesa y la de Una Ronda es un valor máximo. Para la subasta Holandesa es el valor más pequeño (mínimo) de las ofertas.

4.2 Modelo formal para el mecanismo de licitación

Sea T la tarea que un agente desea licitar. Se define un grupo de agentes $A = (a_a(T), A_c)$, donde $a_a(T)$ es el agente administrador de la tarea T , y A_c es el conjunto de agentes contratistas potenciales, $A_c = \{A_1, A_2, \dots, A_n\}$. Además, se supone que $|A| > a_a(T)$, lo que garantiza que al menos dos contratistas participan en la licitación. Finalmente, habrá un agente $A_i \forall i = 1, n$, que será el agente contratista ganador para realizar la tarea, i. e., $\forall T \exists A_i \in A : A_i = A_i(T)$. Este protocolo puede ser visto como una tupla L :

$$L = \langle M, f(T), \vec{O}_c, g(O_c), M_p, RP, h_c(RP), RF \rangle \quad (4.2)$$

donde:

- M es el mensaje inicial, $M = (T, I_a, F)$ donde T es el tipo de tarea a ser licitada. I_a es la información del agente administrador (que puede incluir su dirección virtual (DIR_a), etc.), y F es la fecha de expiración para licitar (que viene dado por un par numérico $F = [DD, MM]$, donde $DD \in \{1, 31\}$ y $MM \in \{1, 12\}$).
- $f(T)$ es una función que permite a los posibles contratistas evaluar sus capacidades para responder a la notificación de la solicitud de realización de la tarea T .
- \vec{O}_c es un vector, $O_c \in M_{m \times 1}$ donde $O_i \in O_{c,i=\{1,\dots,m\}}$ contiene la oferta de cada contratista i para realizar la tarea T .
- $g(O_c)$ es una función que representa los criterios de evaluación de la oferta del agente administrador.
- M_g es el mensaje enviado al contratista ganador. Es una tupla $M_g = \langle A_g, [acepta - propuesta] \rangle$, donde A_g es el nombre del agente ganador, con el acto comunicativo "accept-proposal". Para el resto de los agentes, el agente administrador difunde un mensaje notificándoles que no ganaron la licitación.
- RP es un vector, $RP \in M_{1 \times r}$ donde cada celda $rp_k \in RP$ representa el informe de progreso k en la realización de la tarea T , del agente ganador A_g (para $k = \{1, \dots, r\}$).
- $h_c(RP, k)$ es la función de evaluación del agente administrador del informe de avance k por la ejecución de la tarea T realizada por el del agente ganador A_g .
- RF representa el informe final del agente ganador (A_g) por la culminación de la realización de la tarea T , y es el último elemento del vector RP , i. e., $RF \in RP$.

4.3 Caracterización del proceso de planificación en los sistemas multi-agente

La planificación multi-agente es diferente a la de un agente individual, ya que se debe tomar en consideración el hecho de que las actividades de los agentes pueden interferir entre ellas; por lo tanto, sus actividades deben ser coordinadas [1]. Existen diversas variantes en cuanto al problema de planificación en SMA [43]. Según [43], la solución al problema de planificación en SMA es la definición de una secuencia parcial o totalmente ordenada de tareas, la cual, cuando es ejecutada exitosamente, resulta en un conjunto de objetivos logrados por algún(os) agente(s). Particularmente, la planificación en SMA está vinculada estrechamente a los problemas de coordinación en agentes, la cual tiene dos niveles: un nivel de planificación y un nivel de comunicación. En trabajos previos se han desarrollado formalismos de algunos de los mecanismos de coordinación orientados al nivel de comunicación, basados en los tipos de conversación que pueden ocurrir en cualquier Sistema Multiagente (SMA) (ver capítulo cinco). La planificación corresponde al nivel superior de un proceso de coordinación en un SMA (nivel de planificación), ya que se considera el problema de planificar las tareas a realizar, y quienes las ejecutan (sub-planes) en un determinado orden. El nivel de planificación determina al segundo nivel del proceso de coordinación, el nivel de comunicación (conversaciones y actos de habla presentes), y además integra los otros mecanismos de coordinación (subasta, licitación, entre otras.).

Dentro del proceso de planificación están las tareas de crear los planes, asignarlos y ejecutarlos. Así, el nivel de planificación utiliza el nivel de comunicación de manera implícita, ya que este proceso involucra los tipos de conversación presentes en el nivel de comunicación, los cuales son: solicitar (la creación de un plan), asignar (sub-planes o tareas), y finalmente informar (ejecución de los sub-planes o tareas). En la figura 4.1 se aprecia que ambos niveles (planificación y comunicación) están interconectados, es decir, interactúan entre sí. La planificación es responsable de realizar los planes y su asignación (fase 1 y 2 del proceso de planificación), mientras que la ejecución (fase 3) aparece en las conversaciones (negociación), es decir, consiste en una serie de conversaciones y servicios invocados en las mismas.

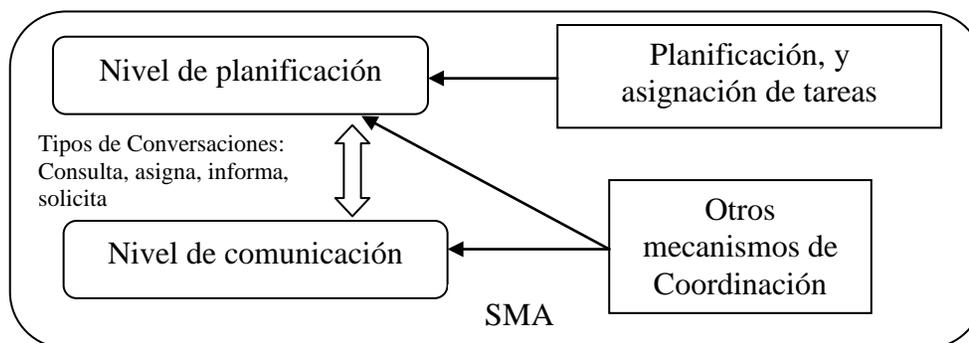


Figura 4.1. Niveles de Coordinación y sus Mecanismos

Recordemos que habíamos dicho en el capítulo dos que en la planificación multiagente puede haber varios agentes planificadores, varios agentes responsables de la asignación de los planes, así como varios ejecutores. Además, en ese capítulo presentamos los modelos de planificación en los SMA: planificación centralizada para planes distribuidos (PCpD), planificación distribuida para un plan centralizado (PDpC), y planificación distribuida para planes distribuidos (PDpD).

Con esa introducción, podemos pasar a formalizar el mecanismo de planificación.

4.3.1 Modelo formal del mecanismo de planificación multi-agente

Para la definición formal de los mecanismo de planificación, partiremos de la hipótesis de SMA donde cada uno puede brindar servicios a los otros para alcanzar sus objetivos, u objetivos globales de la comunidad. En ese sentido, cada agente es caracterizado por los servicios que puede brindar en los diferentes modelos de planificación.

Los diferentes modelos de planificación en los SMA, están caracterizados por:

(i) Los roles que juegan los agentes. Se pueden definir cuatro roles:

1. Agente iniciador: es aquel que dispara el proceso de planificación completo (sus tres etapas).
2. Agentes planificadores: son los encargados de crear el plan o los sub-planes que ejecutarán los agentes ejecutores.
3. Agente de asignación de planes: son los que asignan a otros agentes  responsabilidad de llevar a término el plan o los sub-planes creados por los agentes planificadores. Este rol puede ser desempeñado tanto por los mismos planificadores, como por otros agentes denominados mediadores o brokers.
4. Agentes ejecutores: son aquellos que ejecutan un plan, bien sea de manera parcial o absoluta.

(ii) Un proceso de asignación T . Puede ser visto como una matriz binaria de agentes y servicios, donde los *unos* (1's) representan que ese agente está activo en ese servicio, y los *ceros* (0's) indican que ese agente no realiza ese servicio. En la matriz, un agente puede realizar uno o varios servicios, o viceversa, un servicio puede ser realizado por uno o varios agentes. Sea $A = \{A_i\} \forall i = 1, \dots, n$, un conjunto de n agentes, y $S = \{S_v\} \forall v = 1, \dots, w$ el conjunto de los w servicios. Se define el problema de asignación como la matriz: $T = T_{A \times S}$ descrita en 4.3:

		Servicios (S)				
		T	S_1	S_2	\dots	S_w
Agentes (A)	A_1					
	A_2					
	\dots					
	A_n					

(4.3)

Según esa matriz, un agente i ofrece un servicio específico v , varios agentes ofrecen uno o más servicios, y un servicio puede ser hecho por varios agentes. Estos servicios pueden ser manejar una base de datos, acceder a repositorios, generar alarmas, etc., así como servicios específicos, como por ejemplo, para el caso de un sistema manejador de fallas, servicios de detección de fallas, localización de fallas, etc.

(iii) un objetivo general G que se busca alcanzar con el desarrollo del plan

Así, formalmente un mecanismo de coordinación basado en planificación puede verse como una 6-tupla, cuyos elementos son:

$$PL = \langle AP, AA, AE, T, G, P \rangle \quad (4.4)$$

donde:

- AP, AA y $AE \subset A$ son los conjuntos de agentes que planifican, asignan y ejecutan un plan, o sub-planes, respectivamente.
- T es la matriz de asignación
- G es el objetivo general,
- P es el conjunto de sub-planes sp_e , $\forall e = 1..q$, donde q es el número de sub-planes

Además cada sub-plan (sp_e) es una tupla

$$sp_e = \langle S, R \rangle, \quad (4.5)$$

donde:

- $R = \{R_k\} \forall 1..r$ es el conjunto de los r tipos de recursos requeridos para ejecutar los planes o sub-planes.

Específicamente, como se dijo antes, generar un plan, asignarlo o ejecutarlo, son vistos también como servicios, los cuales pueden ser realizados por uno o más agentes, dependiendo del modelo de planificación que se esté usando (PCpD, PDpC, PDpD).

Así, cada modelo de planificación puede ser representado a través de una matriz $P_{A \times S}$, donde el primer servicio (S_1) será planificar un plan (o sub-planes), el segundo (S_2) asignar

un plan (o los sub-planes), y el tercero (S_3) ejecutar un plan (o los sub-planes). De esta manera tenemos:

Agentes		Servicios (S)		
(A)	$T(PCpD)$	S_1	S_2	S_3
A_1		0	...	1
A_2		0	...	1
...		1	1	0
A_n		0	...	1

(a)

Agentes		Servicios (S)		
(A)	$T(PDpC)$	S_1	S_2	S_3
A_1		1	...	0
A_2		1	...	0
...		...	1	1
A_n		1	...	0

(b)

Agentes		Servicios (S)		
(A)	$T(PDpD)$	S_1	S_2	S_3
A_1		1	...	1
A_2		...	1	...
...		...	1	...
A_n		1	...	1

(c)

Figura 4.2. Modelos clásicos de planificación

En la figura 4.2(a) se tiene:

- Los ceros indican que no hay agente asignado para tal servicio, los unos indican que hay agentes en tales servicios, y los puntos suspensivos indican que ese agente agente podría estar asignado a tal servicio, puede ser el mismo agente o un broker
- $|AP| = |AA| = 1, |AE| > 1$
- Quien genera el plan (S_1 , planificar) es un solo agente (denotado por 1 en la matriz T), i. e., $\exists A_i : \{A_i\} = AP$
- Quien lo asigna (S_2 , asignar) es un agente, que puede ser quien lo planificó o éste puede delegar a otros agentes (mediadores o brokers), eso lo indica el 1 en los puntos suspensivos, i. e., $\exists A_j : \{A_j\} = AA$
- Quienes lo ejecutan son un grupo de agentes, excluyendo a quien lo planificó.

En la figura 4.2(b) se tiene:

- $|AP| > 1, |AA| = |AE| = 1$
- Quienes generan el plan (S_1 , planificar) son varios agentes.
- Quien lo asigna (S_2 , asignar) es un agente (mediador o broker) $A_i : \{A_i\} = AA$
- Quien lo ejecuta es un solo agente, i. e., $A_j : \{A_j\} = AE$

En la figura 4.2(c) se tiene:

- $|AP| = |AA| = |AE| > 1$
- Quienes generan los planes (S_1 , planificar) son varios agentes.
- Quienes lo asignan (S_2 , asignar) pueden ser quienes lo planificaron o éstos pueden delegar a otros agentes (mediadores o brokers) para que presten ese servicio.

- Quienes lo ejecutan pueden ser los mismos agentes planificadores u otros agentes.

También existen diferentes tipos de asignación: asignación directa (el planificador asigna) y asignación con mediador. En el último caso el mediador maneja el proceso de asignación, centralizando las solicitudes de los planificadores y las ofertas de servicios, con el fin de poner en correspondencia estas dos categorías. En general, el proceso de asignación (con o sin negociación), es caracterizado en los diferentes modelos de planificación de la siguiente forma:

- Existe un solo agente planificador y es quien asigna ($|AP| = 1$ y $AP = AA$), o lo realiza otro agente, denominado mediador ($AP \neq AA$).
- Existen varios agentes planificadores y la asignación centralizada la realiza un agente mediador, quien realiza la fusión de los planes, resuelve los conflictos entre ellos, etc. $|AA| = 1$
- Existen varios agentes quienes asignan los planes, resuelven en conjunto los conflictos, etc. $|AA| > 1$

Adicionalmente, es posible formular otras representaciones matriciales complementarias para detallar el proceso de planificación, que sirven para determinar la consistencia del proceso de planificación como se muestra a continuación

Hasta los momentos se ha formalizado el proceso de planificación a nivel de sus tres fases básicas: planificación, asignación y ejecución (vistos como servicios). Ahora bien, en una comunidad de agentes ellos brindan otros servicios, que son los requeridos para la ejecución de los sub-planes, los cuales requieren de ciertos recursos. A continuación en la figura 4.3, se presentan tres matrices: una que permite observar quién o quiénes ejecutan un determinado plan o sub-planes, (matriz agentes-subplanes, ASP), otra que identifica qué servicios son requeridos en los sub-planes (plan) (matriz que se denomina servicios - subplanes, SSP), y finalmente otra matriz que muestra que recursos se necesitan para ejecutar los servicios (a esta matriz se le denomina recursos-servicios, RS). La matriz T se relaciona con la matriz ASP, a través de la matriz SSP (las tres deben ser consistentes entre sí); así, un agente puede ejecutar un plan/sub-plan (ASP) si le asignaron ejecutar los servicios (T) que lo componen (SSP).

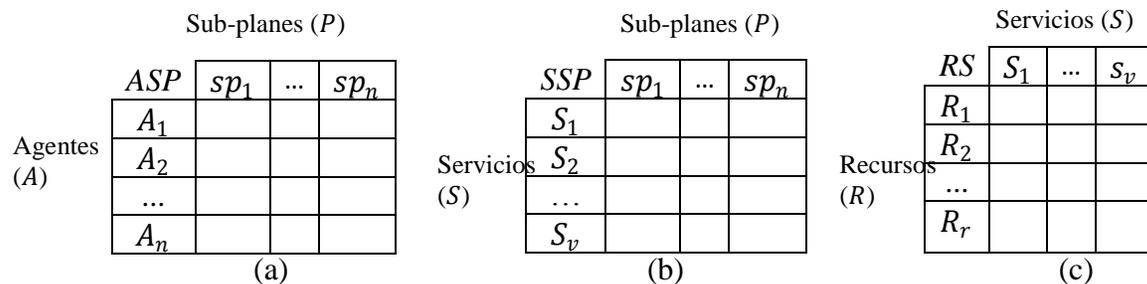


Figura 4.3. Matrices de detalle

RS y SSP son matrices representativas/descriptivas, que sirven para determinar la consistencia del proceso de planificación, y pueden o no ser consideradas al formalizar procesos de planificación en los casos de estudio. Particularmente, en la matriz SSP, sus celdas son representadas por “unos” identificadores que indican la prioridad con que ser ejecutado un servicio en el Sub-plan. Por otra parte, estas matrices se adaptan a cualquiera de los tres métodos clásicos de planificación:

- Para *PCpD*, donde $|AP| = |AA| = 1, |AE| > 1$
- Para *PDpC*, donde $|AP| > 1, |AA| = |AE| = 1$, la matriz (b) tendrán una sola columna, y la matriz (a) un solo elemento, ya que se tiene un solo plan ejecutado por un solo agente
- Para *PDpD*, donde $|AP| = |AA| = |AE| > 1$

Las formalizaciones de los diferentes mecanismos de coordinación, son fundamentales para ser utilizadas por un modelo de optimización de esquemas de coordinación para comunidades de agentes. El siguiente capítulo presenta su implementación computacional usando algoritmos culturales. 

CAPÍTULO V

MODELO CULTURAL PARA EL APRENDIZAJE DE SISTEMAS MULTI-AGENTE

En este capítulo presentamos un modelo de optimización de mecanismos de coordinación para SMA. Nuestro modelo es una forma de aprendizaje colectivo en SMA. En específico, para que los mecanismos de coordinación (MC) puedan ser manipulados por el proceso evolutivo de los algoritmos culturales (AC), estos deben ser formalizados, como se hizo en el capítulo anterior. A partir de esa formalización, el modelo de optimización podrá determinar los esquemas de coordinación adecuados para las interacciones entre agentes de una comunidad dada. En particular, estas interacciones entre agentes son vistas como conversaciones, que a su vez pueden tener sub-conversaciones. En la siguiente sección se presenta el diseño de nuestro modelo de optimización usando AC.

5.1 Principios básicos para caracterizar los componentes del AC

Una sociedad de agentes es un sistema con un determinado número de agentes, los cuales interactúan unos con otros a través de la comunicación por mensajes. En estos entornos pueden existir agentes que cooperan entre sí para lograr sus objetivos y tareas, así como agentes con intereses propios que necesitan negociar para lograr sus objetivos. Así, en todo SMA existen intercambios de mensajes entre agentes que son necesarios para poder interactuar, coordinarse, cooperar, negociar, etc. Para la fundación de agentes físicos inteligentes (FIPA, por sus siglas en inglés) [44], cada mensaje enviado o recibido por un agente es visto como un acto comunicativo (CA, por sus siglas en inglés). En particular FIPA propone una extensa librería de ellos [44]. Cuando ocurre el envío de dos o más CAs entre dos o más agentes, se dice que existe una conversación [45]. Estas conversaciones entre agentes - según FIPA - generalmente presentan patrones típicos que las diferencian unas de las otras. Esos patrones son determinados por las secuencias de CAs esperadas a seguir. Así, FIPA establece estos patrones específicos de intercambio de CAs como protocolos de interacción (IPs, por sus siglas en inglés) [44]. Algunos de los IPs propuestos por FIPA son: protocolo de proposición (propose), de consulta (query), de licitación (contract net), de solicitud (request), entre otros. Por otro lado, en [3] se proponen un grupo de actos de habla que tratan de enmarcar la gran mayoría de los actos comunicativos propuestos por FIPA, entre los cuales se tienen los que se muestran en la figura 5.1:

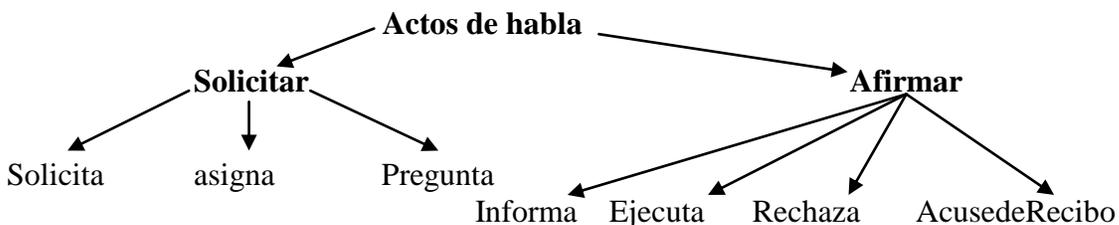


Figura 5.1. Actos de habla

Tabla 5.1 Actos de habla Vs. Actos Comunicativos (mensajes)

Actos de Habla según [3]	Actos comunicativos FIPA
Solicitar	
Solicita	request, request when.
Asigna	accept proposal, call for proposal (cfp), propose, request.
Pregunta	query if, query ref, proxy.
Afirmar	
Informa	agree, failure, not understood, inform, inform if.
Ejecuta	request, inform, inform ref, propagate.
Rechaza	cancel, disconfirm, refuse, reject proposal.
AcusedeRecibo	confirm, inform ref, subscribe.

La tabla 5.1 muestra algunos de los actos comunicativos de FIPA para cada acto de habla, cada IP tiene un conjunto de atributos que los caracterizan, estos son:

- (i) Roles: los agentes cumplen o desempeñan diferentes roles en la conversación, (por ejemplo, en algunos IPs un agente adopta el rol de iniciador y el resto de los agentes el rol de participantes).
- (ii) Motivo u objetivo de la conversación: si el motivo es una consulta se opta por el IP de consulta (query), si es proponer está el IP de propuesta (propose), por citar algunos⁶.
- (iii) Secuencia de actos de habla: cada IP o conversación se inicia con un CA en particular, a su vez uno que le sucede en respuesta a éste, y así sucesivamente hasta que se alcance el objetivo de la conversación. Por ejemplo, en el protocolo de licitación (contract net) se inicia la conversación con un ‘cfp’, el cual es sucedido por un ‘refuse’ o un ‘propose’, y así sucesivamente.

Basado en lo anterior podemos clasificar los diferentes IPs propuestos por FIPA, es decir, podemos generalizarlos. Así, tomando como base la propuesta de la FIPA, se proponen algunos tipos de conversación (TC) característicos en todo SMA. Para poder enmarcar inicialmente gran parte de la extensa literatura de IPs y CAs propuestos por FIPA, se proponen los siguientes TC que nos permitan generalizar las conversaciones que se puedan dar en cualquier SMA. Los TC que se proponen inicialmente son los siguientes:

- TC1: Consulta. Un agente busca cualquier tipo de información, ya sea en base de datos, repositorios, e Internet.

⁶ Existen algunos CAs que también tienen su propio IP, como es el caso de query y propose, entre otros [47]

- TC2: Asignación. A través de esta conversación un agente ordena la ejecución de tareas a otros agentes.
- TC3: Información. Un agente puede informar a otros la ocurrencia de un evento determinado, información que acaba de procesar.
- TC4: Solicitud. Un agente gestiona la realización de un servicio

La tabla 5.2 muestra los TC que se proponen, y algunos de los IPs más importantes propuestos por FIPA que podrían ser usados por los TC propuestos. Esto nos permite comprobar cuan genéricos son con respecto a los IPs propuestos por FIPA.

Tabla 5.2 Tipos de conversación y Protocolos de interacción FIPA

<u>Tipos de conversación</u>	<u>Protocolos de interacción FIPA</u>
TC1: Consulta	Brokering, Query, Request,
TC2: Asignación	Brokering, Contract Net, English and Dutch Auction
TC3: Informar	Brokering, propose
TC4: Solicitud	Request, Query

A continuación se definen los atributos (roles, motivos y secuencia de actos de habla) observados en los IPs propuestos por FIPA, que nos permitirán caracterizar a nuestros tipos de conversación.

Roles. Sus valores pueden ser:

- Agente Usuario (AU): tiene la capacidad o habilidad de ordenar acciones.
- Agente Informante (AI): da a conocer la ocurrencia de un evento determinado o proporciona una información dada.
- Agente Solicitante (AS): solicita o gestiona con otros agentes la realización de una acción.
- Agente Ejecutor (AE): Ejecuta cualquier acción solicitada por los agentes usuario o solicitante.

Motivo. El motivo es el objetivo de la conversación. Inicialmente se proponen:

- Asignar tareas
- Informar un evento/información
- Buscar información
- Solicitar la ejecución de una acción

Secuencia de actos de habla. Especifica el orden de cada acto comunicativo, es decir la secuencia de mensajes entre ellos.

Definidos cada uno de los criterios característicos, a continuación describimos los patrones de los cuatro tipos de conversación inicialmente propuestos.

Tipo de Conversación uno (TC1): Consulta información.

- Descripción: este tipo de conversación está relacionado con todo lo referente a buscar información, consulta en bases de datos, búsqueda de historiales, etc.
- Roles: Agente Usuario (AU) y Agente Ejecutor (AE)
- Motivo: Búsqueda de información
- Secuencia de actos de habla: solicita (AU) – { acepta, rechaza } (AE) – informa (AE)

Tipo de conversación dos (TC2): Asignar tareas

- Descripción: está relacionado con todo tipo de asignación, orden de ejecución, órdenes directas de cualquier tipo de acción que se requiera.
- Roles: Agente Usuario (AU) y Agente Ejecutor (AE)
- Motivo: asignar una determinada tarea o grupo de ellas
- Secuencia de actos de habla: asigna (AU) - informa (AE)

Tipo de conversación tres (TC3): Informar eventos/información

- Descripción: tiene que ver con dar a conocer la ocurrencia de cualquier evento o dar una información
- Roles: Agente Informante (AI) y Agente Usuario (AU)
- Motivo: informar la ocurrencia de un evento o una información
- Secuencia de actos de habla: informa (AI) – { acusederecibo – informa } (AU)

Tipo de conversación cuatro (TC4): Solicitar acción

- Descripción: solicita o gestiona la ejecución de acciones para realizar tareas o servicios.
- Roles: Agente Solicitante (AS) y Agente Ejecutor (AE)
- Motivo: solicitar una acción
- Secuencia de actos de habla: solicita (AS) – { acepta, rechaza } (AE) – informa (AE)

En la secuencia de actos habla de los TC, aquellos encerrados entre llaves y separados por una coma (,) significan que son opcionales, es decir, puede darse tanto uno como el otro, y los actos de habla encerrados entre llaves y separados por un guion (-) significan que son

subsecuentes, es decir, uno le sucede al otro. Otra acotación importante es que un agente dado puede asumir más de uno de los roles durante una conversación, e. g., un agente puede asumir el rol de usuario al inicio de una conversación y más adelante el rol de ejecutante. En la tabla 5.3 se presentan los TC y sus patrones conversacionales. Cada TC es particular en sus patrones. En TC1 y TC2, a pesar de que sus roles son iguales, sus motivos y actos de habla son diferentes. En general, se pueden diferenciar claramente cada uno de los tipos de conversación.

Tabla 5.3 Patrones característicos en los tipos de conversación:

TCs	Roles	Motivo	Actos de habla
TC1: Consultar	AU, AE	Búsqueda de información	solicita – {acepta, rechaza} – informa
TC2: Asignar	AU, AE	Asignar una o grupo de tareas	asigna – {informa}
TC3: Información	AI, AU	Informar	informa – {acusederecibo-informa}
TC4: Solicitar	AS, AE	Solicitar una acción	solicita – {acepta, rechaza} – informa

5.2 Modelo formal basado en algoritmos culturales

El modelo formal de aprendizaje involucra la definición de cada uno de los componentes de un AC [8]. En específico, los componentes del modelo de aprendizaje son: la población, los MC, el espacio de creencias, la función objetivo, la función de aceptación y la función de influencia. En la figura 5.2 se muestra el pseudo-código básico del modelo.

5.2.1 Individuos

La población, como en todo método de Computación Evolutiva (EC), está formada por individuos. En nuestro modelo, cada individuo es un SMA conformado por las diferentes n conversaciones involucradas en la comunidad de agentes (es una instanciación del SMA, usando en cada caso diferentes mecanismos de coordinación en sus conversaciones). Recordemos que cada conversación, a su vez, eventualmente posee sub-conversaciones.

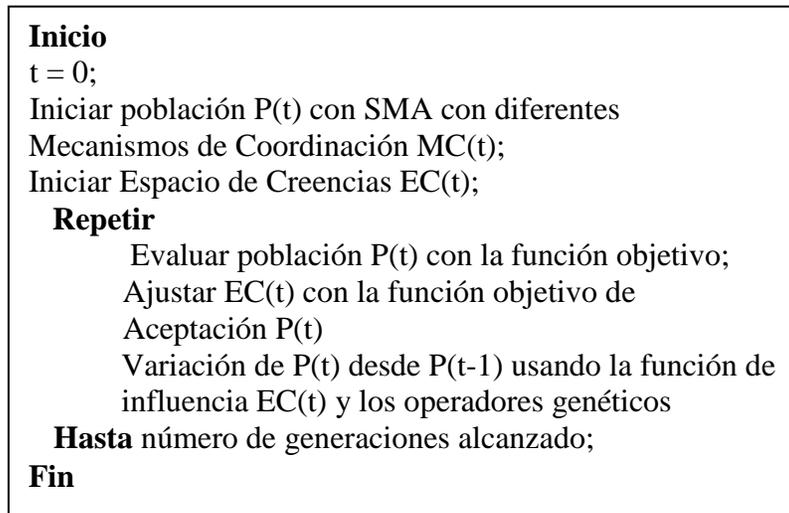


Figura 5.2. Pseudocódigo del algoritmo cultural

Cada conversación es caracterizada por algunos de los TC definidos previamente. En la figura 5.3 se muestra la estructura general de un individuo y su configuración interna.

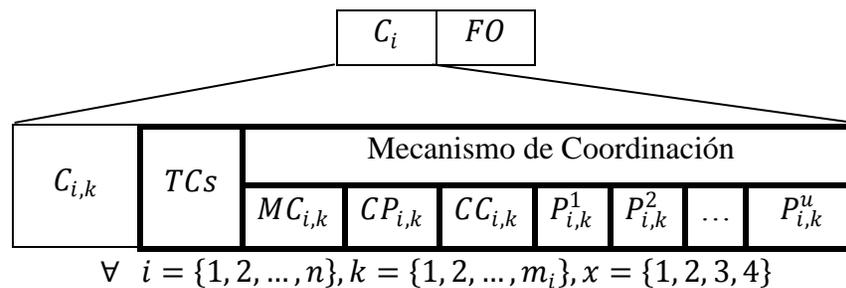


Figura 5.3. Estructura interna del individuo

En la figura 5.3, C_i denota la conversación i existente en ese SMA, en el zoom, $C_{i,k}$ denota la sub-conversación k y de la conversación i , siendo m_i el número de sub-conversaciones asociadas a la conversación, TCs es el tipo de conversación, siendo s : consultar (1), asignación (2), informar (3), y solicitar (4). $MC_{i,k}$ es el mecanismo de coordinación, $CP_{i,k}$ y $CC_{i,k}$ son los costos que implican emplear ese mecanismo (de procesamiento y de comunicación, respectivamente), $P_{i,k}^u$ son los u parámetros para cada mecanismo (ver capítulo 4). La parte remarcada del individuo representa el conocimiento o la experiencia que este aporta.

Por otro lado, en los AC se definen operadores genéticos que permiten la reproducción de nuevos individuos en el espacio de la población, éstos son el operador de cruce y mutación. Un ejemplo de cruce y mutación de individuos se muestra a continuación (ver figura 5.4): suponemos dos individuos con una conversación cada uno, uno de ellos usa como mecanismos de coordinación subasta inglesa (SI) y licitación (L) y el otro subasta

holandesa (SH) y planificación, en cualquiera de sus modelos (PXpY), donde X e Y pueden ser centralizada (C) o distribuida (D).

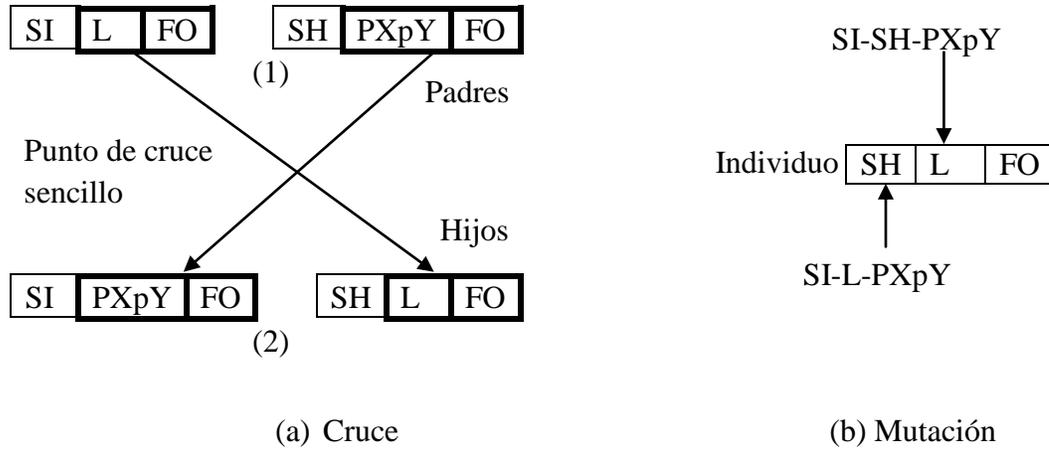


Figura 5.4. Ejemplos de cruce y mutación

En la figura 5.4 (a) se da el cruce por punto sencillo, el cual consiste en escoger un punto aleatorio (un gen que representa un mecanismo de coordinación) común para ambos individuos y en intercambiar la información restante. En la parte (b) se muestra un ejemplo de mutación, en donde cualquier gen representado un mecanismo de coordinación es sustituido de manera aleatoria por cualquier otro mecanismo. Tanto el cruce como la mutación se dan con cierta probabilidad, generalmente la probabilidad de cruce (P_m) es mucho mayor que la de mutación (P_m), pero ambas siempre menor que uno (1).

5.2.2 Función objetivo o fitness (FO)

La función objetivo o función fitness permite evaluar el performance o el desempeño de cada individuo. Esta función está basada en el costo de procesamiento (CP) y el costo comunicacional (CC) del SMA representado en la estructura del individuo. Para nuestro caso, el mejor individuo será aquel que minimice esta función objetivo. Entonces se tiene que:

$$FO = \sum_{i=1}^n \sum_{k=1}^{m_i} (a * CP_{i,k} + b * CC_{i,k}) \quad (5.1)$$

En la ecuación 5.1, a y b son constantes definidas por el usuario, además de permitir normalizar las unidades en la función. El número de conversaciones es n , m_i es el número de sub-conversaciones en una conversación i , $CP_{i,k}$, es el costo de procesamiento del mecanismo de coordinación empleado en la conversación i y la sub-conversación k . Este costo viene dado en bytes, y es determinado por los agentes que participan y los algoritmos de procesamiento del mecanismo de coordinación. $CC_{i,k}$ es el costo comunicacional, el cual

está basado en los diferentes actos comunicativos (mensajes) de los mecanismo de coordinación usados en cada una de las conversaciones del SMA.

En particular, $CP_{i,k}$ viene dado por:

$$CP_{i,k} = PI_k + PE_k + \sum_{l=1}^j \sum_{q=1}^{n_j} A_{l,q} \quad (5.2)$$

En la ecuación 5.2 n_j es el número de participantes en la ronda j , $CP_{i,k}$ viene dado por tiempo de ejecución. En la siguiente tabla se especifican el resto de las variables para cada mecanismo

Tabla 5.4 Definición de los parámetros de la ecuación (2)

Mecanismo de Coordinación (MC)	PI_k	j	n_j	$A_{l,q}$	PE_k
Licitación	Especificación de las condiciones en las que se requiere un servicio	Solo una ronda ($j=1$)	Número de licitadores de la ronda j	Tiempo para preparar propuesta de licitación de los agentes	Proceso de selección del agente a realizar el servicio
Subasta	Fijación de precio inicial e inicio de la subasta	Número de rondas	Número de ofertantes por ronda	Tiempo para preparar propuesta de subasta de los agentes participantes	Proceso de selección del agente ganador
Planificación centralizada	Generación de planes para un grupo de agentes	No aplica	No aplica	No aplica	Proceso de asignación de planes
Planificación distribuida	No aplica	No aplica	Número de agentes planificadores	Tiempo para generar los planes parciales por parte de los agentes participantes en la planificación	Proceso de construir el plan global

En la tabla 5.4 j y n_j son parámetros cuantificables y PI , PE y $A_{l,q}$ son parámetros cualitativos de entrada. Para cuantificar estos valores, nos basamos en [48] en donde proponen una simple escala cualitativa entre bajo, medio y alto, la cual permite dar una idea del peso de estos parámetros para cada mecanismo. Para poder medir con números estos parámetros (las ecuaciones 5.2 y 5.3 así lo requieren) tomaremos como referencia la escala de Likert [49], que permite asignar valores numéricos a este tipo de parámetros. Así le asignaremos el valor de 0,2 al valor bajo; 0,6 al valor medio, y 1 al valor alto. Los demás parámetros son cuantificables y fueron predeterminados en la estructura del individuo. La tabla 5.5 muestra los valores para cada variable.

Tabla 5.5 Valores cualitativos de los parámetros PI , PE y $A_{l,q}$

Mecanismo de Coordinación	PI	PE	$A_{l,q}$
Licitación	Medio	Medio	Medio
Subasta	Bajo	Medio	Bajo
Planificación Centralizada	Alto	bajo	No aplica
Planificación Distribuida	No aplica	Alto	Medio

El costo computacional para la licitación al principio (PI) es medio, ya que el licitador debe preparar y proponer a todos los potenciales agentes contratistas el servicio a solicitar. En la subasta simplemente se anuncia el inicio y se comienza con un precio inicial, sin importar quienes ofertaran. Por otro lado, la planificación centralizada es mucho más compleja, ya que es aquí donde se generan los planes para un grupo de agentes. En el caso de PE , tanto en la licitación como en la subasta el proceso es un poco más complejo, ya que se deben seleccionar en cada caso los ganadores a quienes se les adjudicará el premio. En el caso de PE , tanto en la licitación como en la subasta el proceso es un poco más complejo, ya que se deben seleccionar en cada caso los ganadores a quienes se les adjudicará el premio. En la planificación centralizada este proceso es considerado como bajo, ya que simplemente se da un proceso de asignación de planes entre los agentes. Para el caso de la planificación distribuida este proceso es más complicado, ya que en ese momento se debe realizar la integración de los planes locales en un plan global. Con respecto a $A_{l,q}$, el costo es (medio en la licitación), debido a que cada agente ofertante debe procesar las exigencias del licitador para enviar sus ofertas, mientras que en la subasta es simplemente ofertar sin necesidad de procesar tanta información. Para la planificación distribuida también se presume un poco de complejidad por la necesidad de generar los planes parciales.

Por su parte, $CC_{i,k}$ viene dado por

$$CC_{i,k} = \sum_{l=1}^j \left(\sum_{r=1}^{N-1} CEP_{l,r} + \sum_{s=1}^{n_j} CEO_{l,s} \right) + \sum_{r=1}^{N-1} CS_r \quad (5.3)$$

En la ecuación 5.3 j es el número de rondas y $N - 1$ el número de agentes menos el que envía el mensaje. CEP es el costo de envío de propuesta (costo inicial), CEO es el costo de envío de las ofertas y CS es el costo por informar quien gana. En la tabla 5.6 se especifican el resto de las variables para cada mecanismo.

Tabla 5.6 Definición de los parámetros de la ecuación (3)

Mecanismo de Coordinación (MC)	$CEP_{l,r}$	$CEO_{l,s}$	CS_r	j
Licitación	Costo de envío informando que se licita	Costo de envío de ofertas	Costo de informar quien hace servicio	Solo una ronda ($j=1$)

Subasta	Costo de envío de la propuesta inicial	Costo de envío de ofertas	Costo de informar quien hace servicio	Número de rondas
Planificación centralizada	Costo de envío de plan	No hay	No hay	Solo una ronda (j=1)
Planificación distribuida	No aplica	Costo de envío de los planes parciales	Costo por enviar el plan global (cuando sea necesario, si no, No aplica)	Solo una ronda (j=1)

El costo de servicio (CS) en licitación implica los mensajes enviados para reportar quien ganó el servicio. En la subasta son los mensajes enviados por el subastador durante todo el proceso, hasta enviar el mensaje al ganador para que pague el precio final de la subasta y a los otros para decirle que perdieron. En el caso de P_D , solamente tiene sentido cuando ese plan global debe ser distribuido entre los distintos agentes. En la tabla, CEP y CEO se miden de manera similar a las variables de PI y PE del costo de procesamiento. La tabla 5.7 muestra estos parámetros (CEP y CEO) para cada mecanismo de coordinación.

Tabla 5.7 Valores cualitativos de los parámetros CEP , CEO y CS

MC	CEP	CEO	CS
Licitación	Bajo	Medio	Bajo
Subasta	Bajo	Medio	Bajo
Planificación Centralizada	Alto	No aplica	No aplica
Planificación Distribuida	No aplica	Alto	Alto

En licitación y subasta, el primer mensaje es uno difundido a los agentes (CEP) con la información de que se subasta/licita, y la respuesta (CEO) implica el envío de las ofertas por parte de los agentes ofertantes, lo que hace un poco más complejo (medio) este proceso. Para la planificación centralizada CEP es alto, ya que ha de enviar a todos los agentes involucrados en la planificación sus planes parciales. Para la planificación distribuida, la comunicación de planes parciales (CEO) para la construcción de un plan global es alta. Con respecto a CS , para licitación y subasta se considera bajo, debido a que basta con el envío de un solo mensaje para notificar a todos quien fue el ganador. Para la planificación distribuida es alto, ya que implica el costo de enviar el plan global a todos los agentes (cuando sea necesario).

5.2.3 Espacio de creencias

En el espacio de creencias se tienen dos categorías de conocimiento: Circunstancial y Normativo.

En el conocimiento circunstancial se muestran ejemplos de éxitos y fracasos, éste incluye el tipo de conversación TCs , el mecanismo de coordinación MC , un índice de ocurrencia (IO) de cada mecanismo empleado en ese TC , y finalmente $TO_{(TC)}$, que es el total de

ocurrencias de todos los mecanismos empleados para ese tipo de conversación. En la tabla 5.8, se representa la estructura de ese conocimiento.

Tabla 5.8. Conocimiento Circunstancial

TCs	MC	$IO_{(TC, MC1, t-1)}$	\dots	MC	$IO_{(TCx, MC4, t-1)}$	$TO_{(TCx)}$
$\forall s = \{1, 2, 3, 4\}$						

El valor de s es igual a 4, ya que los TCs son cuatro inicialmente, es decir: consultar, asignar, información y solicitar.

El conocimiento normativo son rangos de valores idóneos de cada una de las variables que conforman a los mecanismos de coordinación (MC) empleados en el conocimiento circunstancial. En la tabla 5.9 LI y LS son los límites inferiores y superiores de cada variable que conforma al mecanismo.

Tabla 5.9. Conocimiento Normativo

MC	P^1		P^2		\dots	P^u	
	LI	LS	LI	LS		LI	LS

Estos rangos son normalizados mediante la ecuación (5.4):

$$z_{norm} = li + \frac{[(z - LI)(ls - li)]}{(LS - LI)} \quad (5.4)$$

En la ecuación 5.4 z_{norm} es el valor normalizado, li es el límite inferior del rango normalizado, ls es el límite superior, z es el valor dentro del rango normal, y LI y LS son los límites inferiores y superiores del rango normal (no normalizado).

5.2.4 Funciones de aceptación e influencia

Las funciones de aceptación e influencia son los protocolos que permiten la interacción entre el espacio de creencias y el espacio de la población. La función de aceptación actualiza ambos conocimientos de la siguiente manera:

Función de Aceptación para el Conocimiento Circunstancial

Esta función toma un porcentaje de la población (un 20% de los individuos es suficiente según Reynolds [9]) para nutrir con sus experiencias el espacio de creencias. La función de aceptación actualiza el conocimiento circunstancial de la siguiente manera: para un tipo de conversación específico, cada uno de los mecanismos involucrados en ese tipo de conversación es actualizado mediante la siguiente ecuación:

$$IO_{(TC, MC, t)} = IO_{(TC, MC, t-1)} + \frac{NO_{(TC, MC, t)}}{TO_{(TC, t)}} \quad (5.5)$$

En la ecuación 5.5 $IO_{(TC, MC, t)}$ es el índice de ocurrencia nuevo (t) para un tipo de conversación (TC), y un mecanismo de coordinación (MC); $IO_{(TCx, MCy, t-1)}$ es el índice de ocurrencia anterior ($t-1$), es decir, el que se encuentra actualmente en el espacio de

creencias; $TO_{(TC)}$ es el total de ocurrencias para un tipo de conversación específica y $NO_{(TC, MC, t)_i}$ es el número de ocurrencias en la actual instanciación del SMA de cada mecanismo de coordinación para ese TC; También es necesario actualizar el total de ocurrencias TO, para eso se emplea la ecuación 5.6, en donde k es el número de mecanismos de coordinación utilizados para ese tipo de conversación:

$$TO_{(TCx)} = TO_{(TCx)} + \sum_{i=1}^k NO_{(TCx, MCy, t)_i} \quad (5.6)$$

Función de Aceptación para el Conocimiento Normativo

La función de aceptación actualiza el conocimiento normativo mediante la siguiente ecuación:

$$Lac = \left[\frac{Lv * \bar{m} + \bar{P} * m}{2} \right] \quad (5.7)$$

En la ecuación 5.7 Lac es el límite actual (ya sea LI o LS), Lv es el límite anterior, \bar{m} es el complemento del momento, es decir, $(1 - m)$, \bar{P} es el promedio del valor del límite de todos los individuos dentro del 20% aceptados proveniente de la población. Finalmente m es el momento, que viene dado por la ecuación:

$$m = \frac{\mu}{t} \quad (5.8)$$

En la ecuación 5.8 μ es una constante de momento entre 0 y 1, y t es el número de iteraciones, ($t = 1, 2, 3, \dots$). De esta manera, cada vez que llega una nueva experiencia de la población, los límites de cada parámetro de ese mecanismo se actualizan.

Función de Influencia para el Conocimiento Circunstancial

La función de influencia determina como el conocimiento del sistema influye en los individuos de la población. El conocimiento circunstancial influye en los individuos como lo muestra la figura 5.5:

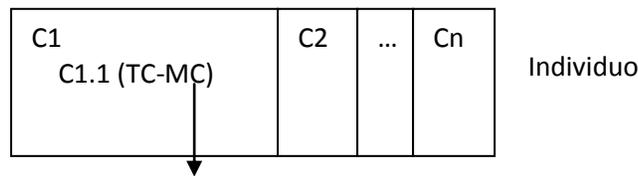


Figura 5.5. Cambio del mecanismo de coordinación de manera aleatoria.

Básicamente consiste en cambiar los mecanismos de coordinación en un individuo. Para el cambio de estos mecanismos (planificación, licitación, algún protocolo de subasta, etc.) se emplea un tipo especial de operador genético de mutación. Ese operador de mutación usa el conocimiento circunstancial para cambiar el mecanismo de coordinación actual de una conversación dada, acorde a una regla probabilística basada en el parámetro IO de cada TC (a eso llamamos mutación dirigida).

Función de Influencia para el Conocimiento Normativo

En este caso, la función influencia afecta los parámetros de los mecanismos de coordinación del individuo. Esta modificación se puede realizar a través de un tipo de operador genético de mutación como el anterior (mutación dirigida), para el caso del conocimiento circunstancial, solo que aquí no se varían los mecanismos de coordinación, sino solo se alteran sus parámetros específicos (LI, LS), como lo muestra la figura 5.6.

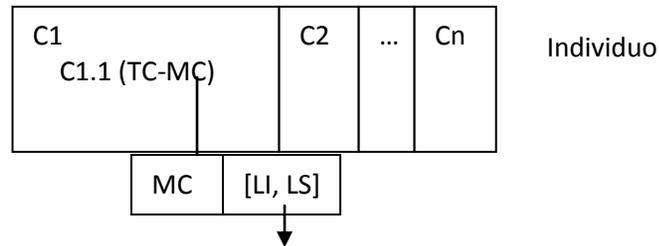


Figura 5.6. Modificación de los parámetros de manera aleatoria

La siguiente figura (figura 5.7) muestra como todos los componentes anteriores se integran en el AC.

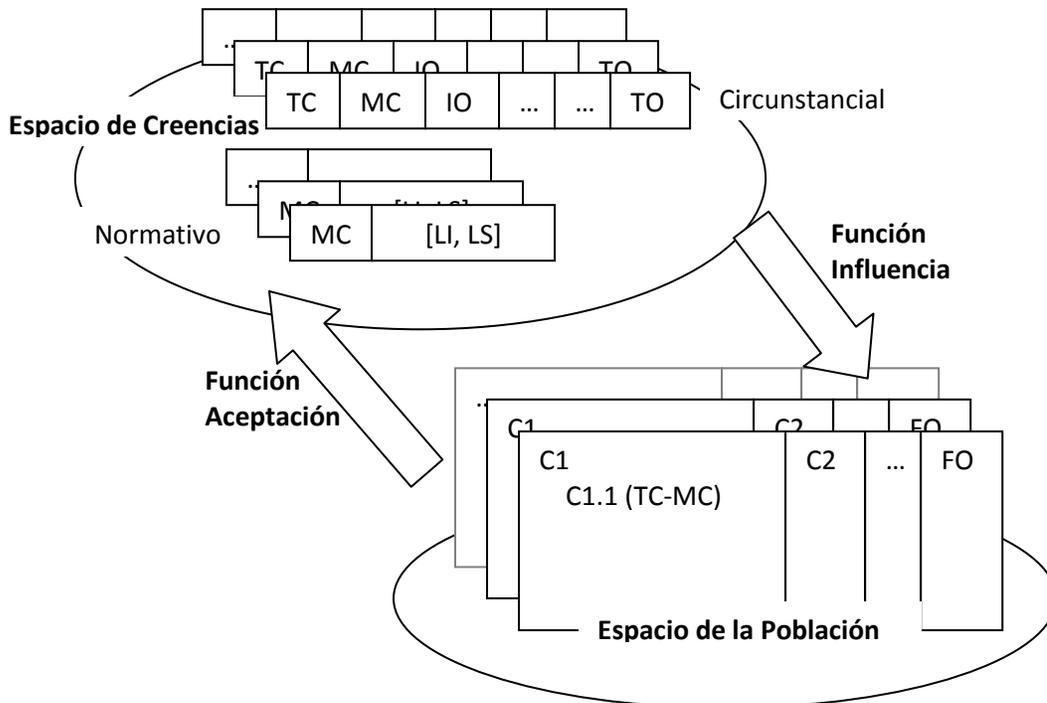


Figura 5.7. Modelo cultural de aprendizaje SMA

Así, la figura 5.7 muestra la estructura general de AC, de nuestro modelo de optimización. En el *espacio de la población* se muestran los individuos y su estructura interna, conformada por conversaciones y sub-conversaciones, que a su vez, poseen tipos de conversación que usan mecanismos de coordinación. Al final del individuo se presenta su

función objetivo. El *protocolo de comunicación* es representado por las funciones de aceptación e influencia, la función de aceptación toma un 20% de sus mejores individuos para depositar sus experiencias en el espacio de creencias, y la función de influencia enriquece a los individuos de la población con el conocimiento provisto por el espacio de creencias, usando para ello el operador de mutación dirigida. Finalmente, el *espacio de creencias* va actualizando sus conocimientos circunstanciales y normativos con las experiencias de las nuevas generaciones provistas por la función aceptación.



CAPÍTULO VI

HERRAMIENTA COMPUTACIONAL PARA EL APRENDIZAJE DE COORDINACIÓN EN SISTEMAS MULTI-AGENTE: CLEMAS

Este capítulo describe el diseño e implementación de un sistema computacional, que describe el comportamiento de los SMA desde el punto de vista de los mecanismos de coordinación en sus conversaciones. El sistema computacional está basado en el AC presentado en la sección anterior, tal que a través de sus generaciones, los individuos van aprendiendo a seleccionar cual mecanismo de coordinación resulta el más adecuado para una determinada conversación. Como el sistema está basado en la estructura de un AC (ver capítulos 2 y 5), tiene como componentes una población, un espacio de creencias y un protocolo de comunicación.

6.1 Arquitectura de CLEMAS

CLEMAS (del acrónimo en inglés Cultural LEarning for Multi-Agent Systems) es el sistema computacional propuesto. En particular, es la herramienta computacional que implementa el modelo de aprendizaje propuesto en el capítulo V. Está integrado por cuatro componentes principales: el motor de ejecución, el AC, una interfaz gráfica para configurar el sistema inicialmente, y una base de datos que almacena el conocimiento adquirido por los individuos en el espacio de creencias. La figura 6.1 muestra la arquitectura de CLEMAS.

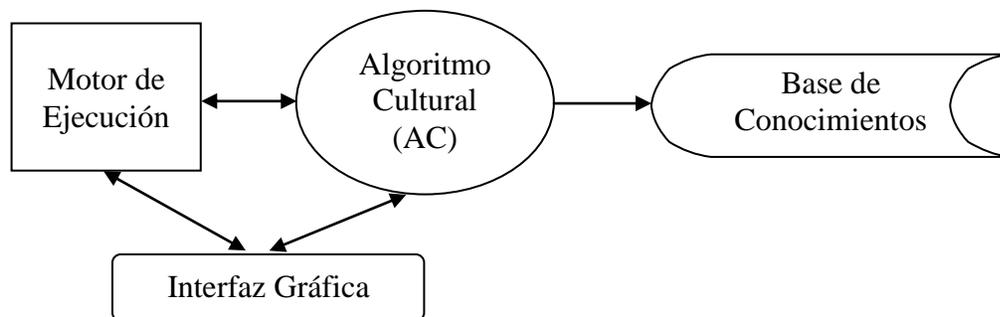


Figura 6.1. Arquitectura de CLEMAS

El *motor de ejecución* inicia el proceso de aprendizaje a través de una clase denominada ‘Simulación’, utilizando para esto la configuración inicial del sistema computacional. Esta configuración se realiza a través de la *interfaz gráfica*, y consiste en lo siguientes: se debe definir el tamaño de la población y la estructura de los individuos inicialmente, especificando sus conversaciones, tipos de conversaciones, y mecanismos de coordinación que usaran al principio de la simulación. Luego se debe configurar el número de generaciones, y la probabilidad de cruce y mutación. El componente del AC toma los datos cargados y crea la población, además inicializa el espacio de creencias. A medida que evoluciona el algoritmo (lo hacen el motor de ejecución y AC), el componente *base de*

conocimientos almacena el conocimiento circunstancial y normativo adquirido (es el espacio de creencias del AC). Además, CLEMAS guarda un archivo de extensión .ccg, por si se desea usar este conocimiento en otros SMA. A continuación presentamos su diseño e implementación.

6.2 Diseño de CLEMAS

CLEMAS es una herramienta que permite, bajo diversas funciones de ajustes hecha por el usuario determinar:

- Los mecanismos de coordinación más idóneo para un tipo de conversación (TC).
- Los mecanismos de coordinación más empleado por los individuos.
- El valor de la función objetivo (FO) del mejor individuo.
- Un nuevo conocimiento colectivo (en el espacio de creencias) a través de las generaciones, como producto de un proceso de aprendizaje del SMA.

CLEMAS fue desarrollado en Java. A continuación se presenta su diseño utilizando en algunos casos el modelado de lenguaje unificado (UML). Además, el diseño es orientado a clases.

6.2.1 Individuos

La representación de los individuos en CLEMAS se hace mediante una clase llamada ‘individuo’, cuyos atributos principales son un conjunto de conversaciones, una función objetivo y un identificador. En cada iteración de la simulación se vuelve a calcular el valor de la función objetivo de los individuos, todos los cálculos realizados en este módulo son realizados utilizando las ecuaciones mencionadas en el capítulo 5.

Todas las clases utilizadas para la definición del individuo se muestran en la figura 6.2, la clase individuo se relaciona con la clase que describe las conversaciones individuales, y esta, a su vez, se relaciona con la clase sub-conversaciones, que su vez se relación con la clase mecanismos de coordinación. Finalmente, la clase mecanismo de coordinación se relación con la clase limites, para establecer los límites inferior y superior de cada mecanismo de coordinación. Como lo indica el diagrama UML, la clase ‘individuo’ es la clase central. Dentro de los atributos de esta clase está el id, que aunque no pertenece formalmente al individuo del modelo, sirve para identificar cada uno de ellos en el algoritmo.

6.2.2 Espacio de creencias

Las clases utilizadas para especificar el espacio de creencias se muestran en la figura 6.3. En el diagrama UML se aprecia que se requieren tres clases para definir el espacio de creencias.

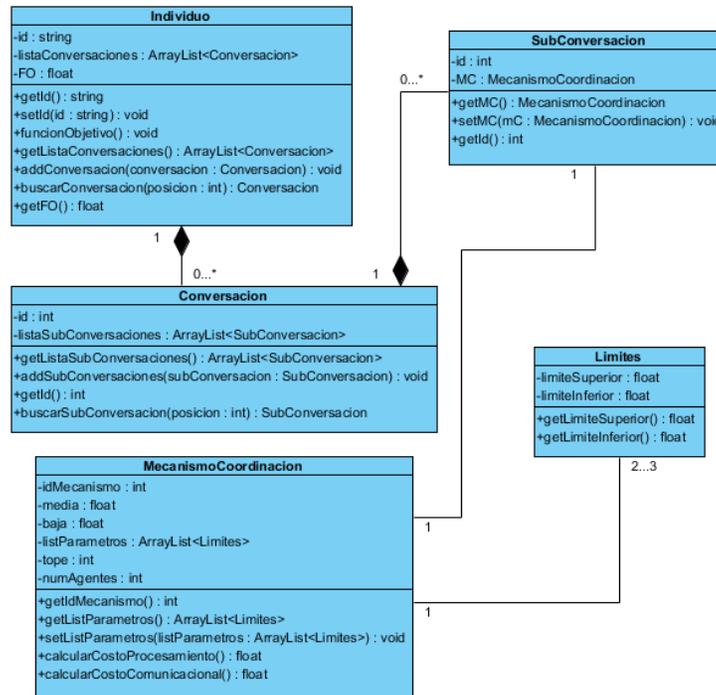


Figura 6.2. Diagrama UML para la clase individuo

EspacioCreencias: es la clase que modela el espacio de creencias, sus principales atributos son el conjunto de conocimientos circunstanciales y normativos (por eso se relaciona con esas clases). Este se encarga de ejecutar las funciones de aceptación e influencia.

ConocimientoCircunstancial: Esta clase representa el conocimiento circunstancial. Sus atributos son: un identificador del tipo de conversación y del mecanismo de coordinación, su índice de ocurrencia y la ocurrencia total. Estos atributos son suficientes para modelar este conocimiento, donde se muestra la experiencia en cuanto a qué tipo de mecanismo se usa en qué tipo de conversación, y cuanto se ha utilizado.

ConocimientoNormativo: Esta clase representa el conocimiento normativo. Sus principales atributos son: el identificador del mecanismo de coordinación, y los límites de sus parámetros. Esta clase modela el rango de cada parámetro que contienen los mecanismos de coordinación.

6.2.3 Sistema principal de simulación

Para llevar a cabo las simulaciones del sistema se diseñó una clase llamada ‘simulacion’, cuyos atributos principales son: máximo número de generaciones, el tamaño de la población, la tasa (en porcentaje) de aceptación, la probabilidad de cruce, la probabilidad de mutación (dirigida o normal), el id del espacio de creencias y un conjunto de estados.

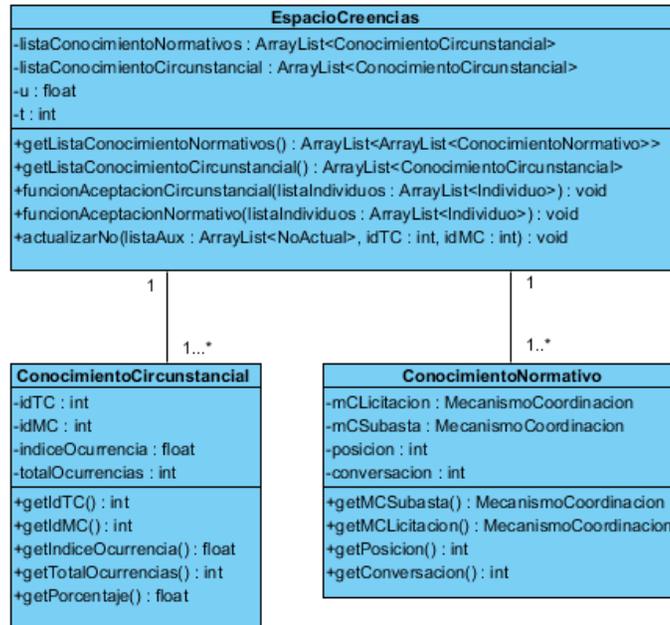


Figura 6.3. Diagrama en UML de la clase EspacioCreencias

Es responsable de iniciar el sistema, determina cuáles son los mejores individuos que puedan influir en el espacio de la población, qué mecanismo se utilizará para crear la próxima generación, y lleva un registro de lo que sucede a lo largo de la simulación. La figura 6.4 muestra el diagrama con las clases que participan en este módulo.

Simulación: Esta clase es responsable de la simulación del sistema.

CruzarIndividuos: es responsable de crear un nuevo individuo a través del método de cruce, el cual escoge dos sub-conversaciones al azar para cruzarlas entre sí, y obtener así dos nuevos individuos (ver capítulo V).

MutarIndividuo: es responsable de crear un nuevo individuo por el método de mutación. En este módulo se determina qué tipo de mutación es (dirigida o normal), y aleatoriamente selecciona qué conversaciones o sub-conversaciones serán mutadas.

Estado: Esta clase se encarga de almacenar la información en un estado de la simulación.

6.2.4 Interfaz

El panel de control permite al usuario establecer las condiciones iniciales de la simulación. Este panel tiene varios controles necesarios para implementar y diseñar el SMA. La figura 6.5 muestra los componentes del panel. Las diversas clases requeridas para el manejo de las acciones son (ver figura 6.6):

PanelCentral: Este es responsable de crear el panel principal del sistema. Se puede dividir en dos regiones, una región superior para el diseño del individuo, y una inferior para la configuración de la simulación total (ver figura 6.5).

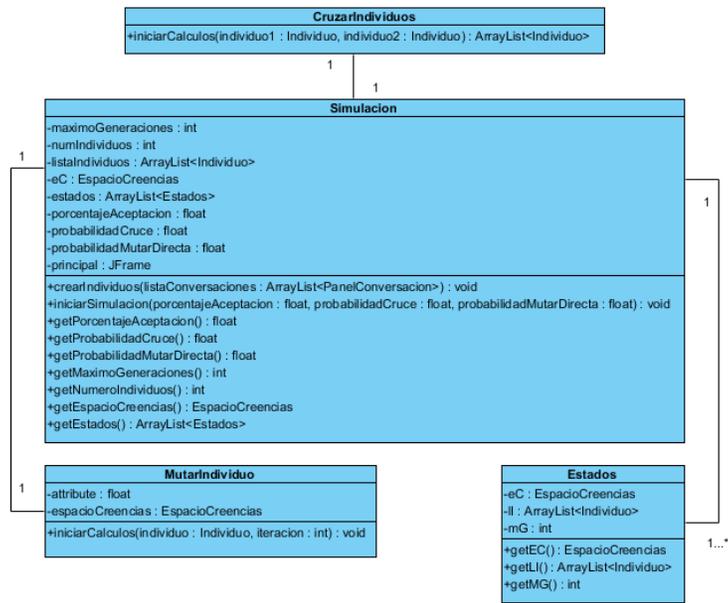


Figura 6.4. Diagrama UML de la clase CruzaIndividuos

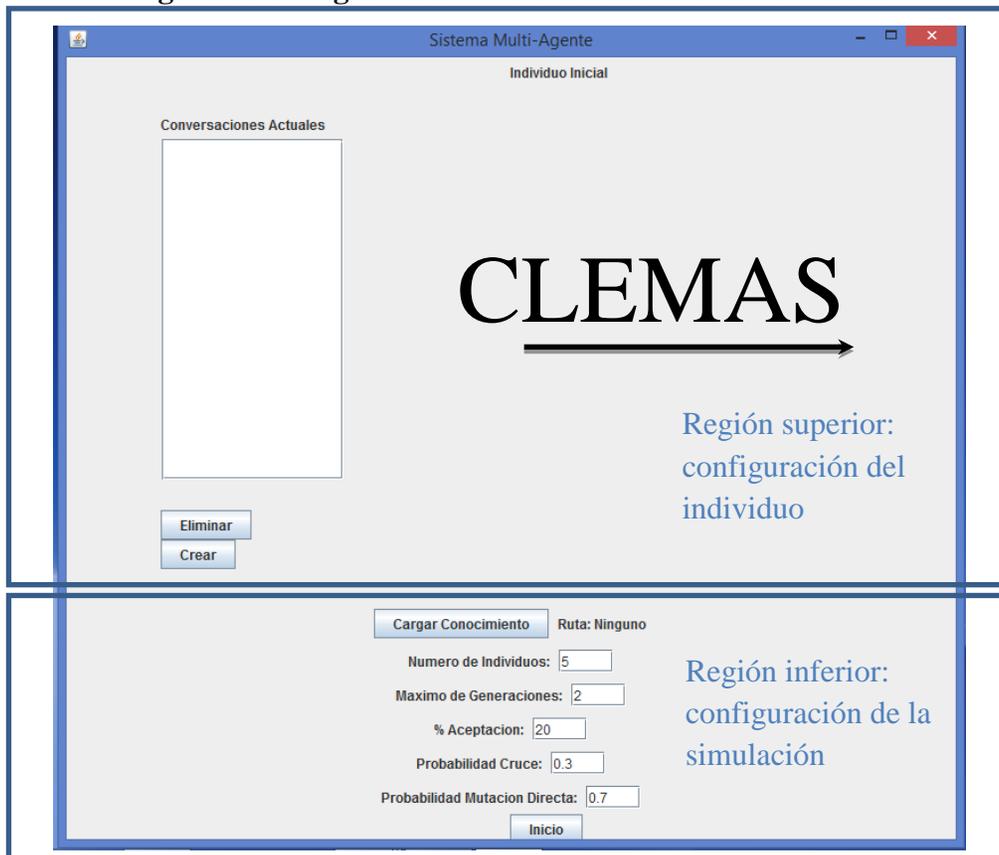


Figura 6.5. Panel de configuración inicial

PanelSupConversacion: Esta clase es responsable de crear la ventana para el diseño de una sub-conversación, permite al usuario establecer el tipo de conversación, los parámetros de cada mecanismo, un máximo de rondas (en el caso de ser la subasta) y el número de los actores involucrados en la sub-conversación (ver figura 6.7).

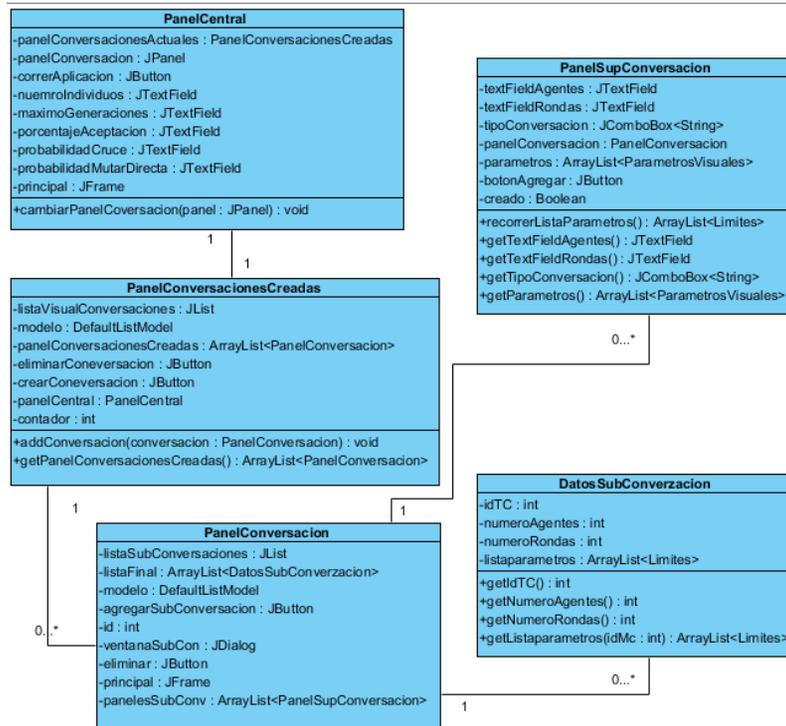


Figura 6.6. Diagrama UML de las clases de los paneles principales

The screenshot shows a window titled "Sub-Conversion" with the following elements:

- Tipo de conversacion:** A dropdown menu set to "Consulta".
- Numero de Agentes:** A text input field containing the value "1".
- Numero de Rondas:** A text input field containing the value "1".
- Subasta Inglesa:** Three pairs of input fields labeled X1, X2, and X3, each with values "1" and "2".
- Licitacion:** Two pairs of input fields labeled X1 and X2, each with values "1" and "2".
- Subasta Holandesa:** Three pairs of input fields labeled X1, X2, and X3, each with values "1" and "2".
- Aceptar:** A button at the bottom.
- Legend:**
 - Para Subasta: X1: Precio Inicial, X2: Maxima Cantidad, X3: Numero Rondas
 - Para Licitacion: X1: Fecha de expiracion de oferta, X2: Capacidad para ejecutar una tarea

Figura 6.7. Ventana para crear una sub-conversación

6.2.5 Panel de resultados

Permite al usuario ver los datos finales de la simulación. Se necesita una ventana para la visualización de los resultados de la simulación en cada iteración, para llevar a cabo esto se diseñó la clase PanelResultados (ver figura 6.8).

Clase PanelResultados: es responsable de crear la ventana de resultados, se divide en tres regiones: superior, media e inferior, cada región muestra una información específica de la simulación. Fue diseñado en la forma más amplia posible, ya que permite al usuario ver tanto el comportamiento de los mecanismos como el histórico de la función objetivo durante toda la simulación.

PanelHistorial: Esta clase es responsable de la creación de una ventana que puede mostrar una historia detallada de lo sucedido en cada iteración de la simulación.

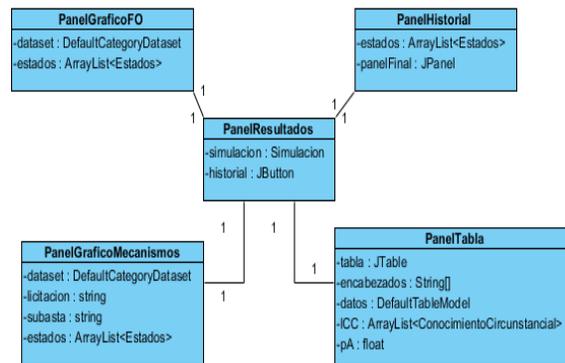


Figura 6.8. Diagrama UML de las clases que muestran los resultados



CAPÍTULO VII

CASOS DE ESTUDIO

Una vez obtenido el modelo de aprendizaje de esquemas de coordinación basado en AC, y luego de haber desarrollado un modelo computacional que permita implementarlo (CLEMAS), queda como siguiente paso aplicar la propuesta a casos de estudio, para evaluar su desempeño en sistemas de automatización industrial. Todos los casos de estudio tratan el problema de integración en automatización. Además, todos son implementados en nuestro modelo siguiendo la misma metodología. El primer caso de estudio es un sistema manejador de fallas basado en SMA, el segundo es un SMA que modela el proceso de planificación de una unidad de explotación petrolera, y el tercer caso es un SMA de una unidad de explotación de yacimientos de producción petrolera. A continuación se presentan cada uno de ellos. Después se describe como nuestros formalismos matemáticos pueden ser usados para describirlos, y finalmente se usa CLEMAS en ellos para analizar los mecanismos de coordinación que usan.

7.1 Casos de estudio

7.1.1 Caso A: Sistema manejador de fallas basado en un sistema multi-agente.

El Sistema para el Manejo de Fallas (SMF) está compuesto por dos módulos, el primero realiza tareas de monitoreo y análisis de la Falla, y el segundo realiza tareas de apoyo para la gestión del mantenimiento del sistema [50]. El SMF interactúa con la Gerencia de Ingeniería para todo lo relacionado con los índices de productividad del proceso, manejo de recursos, etc., y con el Proceso Controlado Tolerante a Fallas. El módulo de Monitoreo y Análisis de la Falla comprende las tareas de: Detección y Diagnóstico de las fallas, el módulo de Apoyo a las Tareas de mantenimiento, comprende las tareas de Predicción de la ocurrencia de una falla funcional, Planificación de las tareas de mantenimiento preventivo y Ejecución de Mantenimiento [50].

Modelo Basado en Sistemas Multi-Agente:

El SMF puede verse como un sistema compuesto por agentes inteligentes, capaces de cooperar para la solución de problemas relacionados con el manejo de las fallas en el sistema. Por otro lado, algunas actividades realizadas por el SMF deben ajustarse a un modelo computacional distribuido, como por ejemplo, las realizadas para la detección de la falla en equipos o en procesos, la estimación de índices de funcionamiento, entre otras. Para la especificación del SMA se usó la metodología MASINA [21], la cual prevé varios modelos. En nuestro trabajo sólo mostraremos los que son de nuestro interés: modelo de agentes, modelo de tareas y modelo de coordinación; para una presentación más detallada de los mismos ver [50].

Modelo de Agentes: El SMF provee las siguientes funcionalidades: monitorear, detectar, localizar, analizar, predecir la ocurrencia de una falla, y corregirla en el sistema de control. Estas funciones representan los roles de los actores, quienes permanecen como agentes en el sistema, aunque algunos de ellos fueron divididos posteriormente en varios agentes.

El SMA del SMF está basado en el modelo SCDIA (capítulo 2, figura 2.6), el cual es un modelo de referencia de SMA que ha sido diseñado para sistemas de control (ver capítulo 2). El SCDIA ha sido adaptado al problema de manejo de fallas, visualizando dicho problema como un problema genérico de control en lazo cerrado. Así pues, se definieron ocho agentes denominados: Agente Especializado Detector, Localizador, Diagnosticador, Predictor, Agente Coordinador, Agente Controlador, Agente Actuador, Agente Observador. Los Agentes Especializados y el Agente Coordinador se encuentran en el nivel Supervisor, y en el nivel de Proceso se encuentran los Agentes Controlador, Observador y Actuador (ver figura 7.1). La especificación detallada de estos agentes se encuentra en [50].

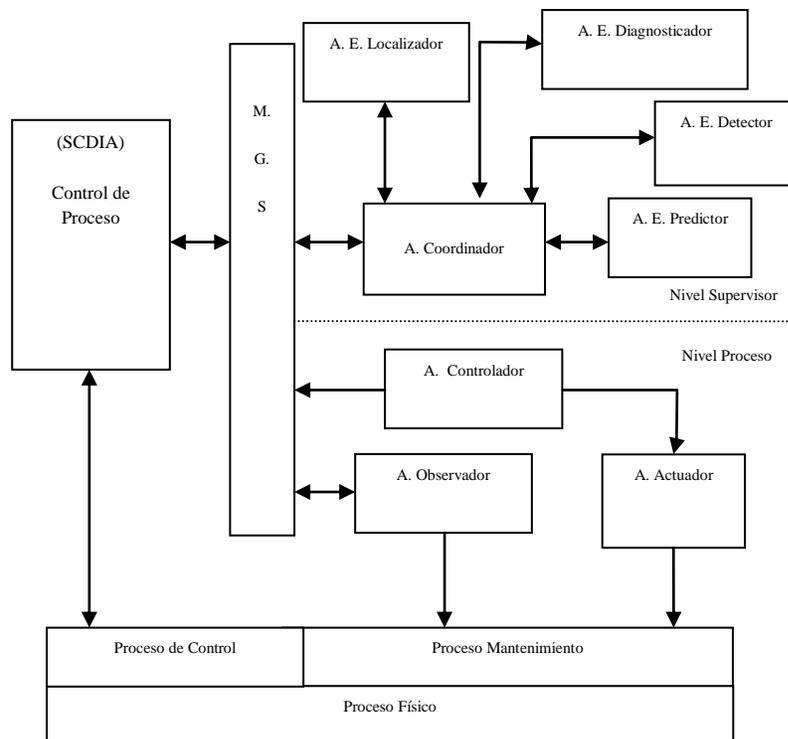


Figura 7.1. SMA del SMF

Además, el SMF interactúa con el SCDIA de control de proceso a través del Medio de Gestión de Servicios (MGS), compuesto a su vez por otros agentes: Agente Base de Datos, Agente Administrador de Agentes, entre otros.

Modelo de tareas: las tareas identificadas en el SMF son mostradas en la tabla 7.1:

Tabla 7.1. Modelo de Tareas

Tareas de Observación	(i) Identificar fallas funcionales abruptas, (ii) Calcular índices de funcionamiento, (iii) Determinar el estado del mantenimiento.
Tareas de Detección	(i) Llevar estadísticas sobre la ocurrencia de fallas, (ii) Seleccionar técnica de detección, (iii) Incorporar nuevos métodos de detección
Tareas de Localización	(i) Ubicar falla
Tareas de Diagnóstico	(i) Llevar estadística de los modos de falla, (ii) Llevar estadística sobre las causas de las fallas, (iii) Realizar análisis sobre las consecuencias de las fallas en el sistema, (iv) Reajustar modelos de diagnóstico, (v) Incorporar nuevos modelos de diagnósticos, (vi) Incorporar nuevos modos de fallas, (vii) Incorporar nuevas causas de fallas, (viii) Identificar modos de fallas y sus causas.
Tareas de Predicción	(i) Calcular curvas de confiabilidad en equipos, (ii) Generar índices de confiabilidad del proceso, (iii) Incorporar nuevos modelos de predicción
Tareas de Control	(i) Proponer plan de mantenimiento, (ii) Procesar plan de mantenimiento
Tareas de Coordinación	(i) Proponer macro plan de mantenimiento, (ii) Evaluar recursos, (iii) Ordenar realización de tarea de detección, localización, diagnóstico y predicción (DLDP), (iv) Ordenar mantenimiento correctivo, (v) Replanificar el mantenimiento preventivo
Tareas de Actuador	(i) Ejecuta las tareas de mantenimiento, (ii) Ejecuta planes de contingencia

Modelo de coordinación: a través de este modelo se describen las conversaciones, los objetivos y participantes en las mismas [50]. La tabla 7.2 muestra las diferentes conversaciones en ese SMA.

Tabla 7.2. Modelo de Coordinación

<p>Conversación: Mantenimiento por Condición</p> <p>Objetivo: Realizar mantenimiento por condición en el sistema.</p> <p>Agentes: Coordinador, Controlador, Base de Datos, Detector, Localizador, Predictor, Diagnosticador y Humano.</p> <p>Descripción: Mediante esta conversación, el agente controlador envía la</p>
--

<p>información al agente coordinador, para que éste construya el plan de mantenimiento, basado en tareas DLDP.</p>
<p>Conversación: Tareas de Mantenimiento</p> <p>Objetivo: Realizar tareas de mantenimiento en el sistema.</p> <p>Agentes: Controlador, Base de Datos, Actuador y Observador.</p> <p>Descripción: Mediante esta conversación, el agente controlador envía la información al agente actuador, para que éste realice las tareas de mantenimiento en el sistema, y a partir de la realización o no del mantenimiento, el agente observador reporta las tareas que no pudieron ser realizadas.</p>
<p>Conversación: Tareas Urgentes</p> <p>Objetivo: Realizar tareas urgentes en el sistema.</p> <p>Agentes: Coordinador, Controlador, Base de Datos, Actuador, Observador y Humano.</p> <p>Descripción: Mediante esta conversación, el agente coordinador envía la información al agente controlador, para que éste ordene el mantenimiento, y si no se pudo realizar esta tarea urgente, se da una alarma.</p>
<p>Conversación: Replanificación de Tareas</p> <p>Objetivo: Replanificar tareas que no se efectuaron en el sistema.</p> <p>Agentes: Coordinador, Base de Datos y Humano.</p> <p>Descripción: Mediante esta conversación, el agente coordinador busca la información proveniente del agente base de datos para replanificar las tareas de mantenimiento pendientes en el sistema, y realizar un nuevo plan de mantenimiento. Si la tarea es urgente y no se puede replanificar se da una alarma.</p>
<p>Conversación: Estado de Mantenimiento</p> <p>Objetivo: Realizar tareas urgentes en el sistema</p> <p>Agentes: Observador, Base de Datos y actuador.</p> <p>Descripción: Mediante esta conversación, el agente observador busca la información proveniente del agente base de datos y del agente actuador, para almacenar las tareas de mantenimiento pendientes en el sistema.</p>
<p>Conversación: Identificar Falla Funcional</p> <p>Objetivo: Realizar tareas urgentes en el sistema</p> <p>Agentes: Observador, Base de Datos, Coordinador y Controlador.</p> <p>Descripción: Mediante esta conversación, el agente observador busca la información proveniente del agente base de datos, para reportar la presencia de una falla funcional abrupta en el sistema, el agente coordinador busca este reporte y ordena urgentemente la realización de una tarea de mantenimiento en el sistema.</p>

En este caso de estudio analizaremos el problema de integración horizontal, así como la forma de instanciar los formalismos matemáticos desarrollados en este trabajo para los mecanismos de negociación.

7.1.2 Caso B: Planificación de un proceso de producción petrolera por levantamiento artificial por gas basado en SMA.

Para este caso se toma como estudio un proceso de producción petrolera, en específico, el lazo de producción de pozos. Particularmente, el proceso de producción petrolera está compuesto por un conjunto de instalaciones, las cuales son: pozos, estaciones de flujo, patios de tanques, plantas compresoras de gas y múltiples. Existen diversos métodos de producción de hidrocarburos, clasificados según los mecanismos utilizados para llevar hidrocarburos desde el yacimiento hasta la superficie [39]. Este caso de estudio se concentra en el método de levantamiento artificial por gas (LAG).

Modelo de agentes: los potenciales agentes para este caso de estudio son: pozo (PZ), Estaciones de Flujo (EF), Plantas Compresoras de Gas (PC), Múltiples de Levantamiento Artificial por Gas (MLAG), Patios de Tanques (PTQ). Además, según [39] el SMA específico para el proceso de planificación está compuesto por los agentes: Agente Administrador de Requerimientos (AAR), Agente Administrador de Productos (AAP), Agente Administrador de Recursos (AAR), Agente Identificador de Escenarios (AIE), Agente Planificador (AP), Agente Identificador de Riesgos (AIR), y Agente Monitor de la Ejecución (AME). En la figura 7.2 se muestra el SMA del lazo de producción de pozo

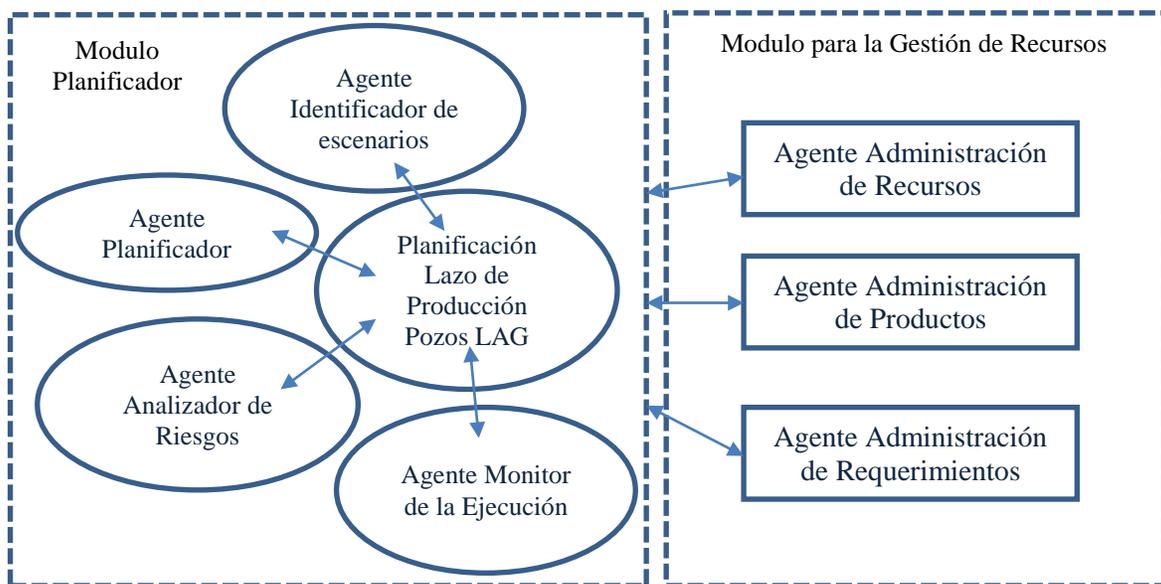


Figura 7.2. SMA para la planificación en un lazo de producción de pozos LAG

Con este caso de estudio analizamos la forma de instanciar el formalismo matemático del protocolo de planificación desarrollado en este trabajo.

7.1.3 Caso C: SMA de una unidad de explotación de yacimientos de producción petrolera

El tercer caso de estudio es tomado de un proyecto real de una industria petrolera, en el que se estudió la implantación de funcionalidades inteligentes a los sistemas de producción petrolera, para lo cual se propuso el uso de agentes en la automatización de procesos [51]. Desde el punto de vista de uso de agentes inteligentes, en ese proyecto se definió una arquitectura de automatización tal como se muestra en la figura 7.3, donde se distinguen tres niveles funcionales: *nivel de campo*, donde se encuentran los elementos actuadores y sensores, *nivel Medio de Gestión de Servicios* (MGS), y *nivel de aplicaciones de agentes inteligentes*. El nivel MGS representa el conjunto básicos de módulos de software que implantan las abstracciones mínimas para la especificación, implantación y manipulación de los agentes, y su interacción con los dispositivos de campo objeto. Este nivel está compuesto a su vez por tres niveles: Nivel Interfaz, Nivel Medio, y Nivel de Acceso a Recursos [51]. *El nivel interfaz* está compuesto por los agentes definidos por la especificación FIPA para brindarles servicios a las comunidades de agentes (SMA). Está conformado por cinco agentes: Agente Administrador de Agentes (AAA), Agente Gestor de Recursos (AGR), Agente Gestor de Aplicaciones (AGA), Agente Gestor de Datos (AGD) y Agente de Control de Comunicación (ACC) [52]. *El Nivel Medio* constituye el núcleo del sistema distribuido, provee servicios de software que requieren los agentes para poder interactuar entre sí y con el nodo de ejecución [52]. *El Nivel de Acceso a Recursos* representa el conjunto de servicios que provee el sistema operativo, incluyendo el manejo de tareas de tiempo real.

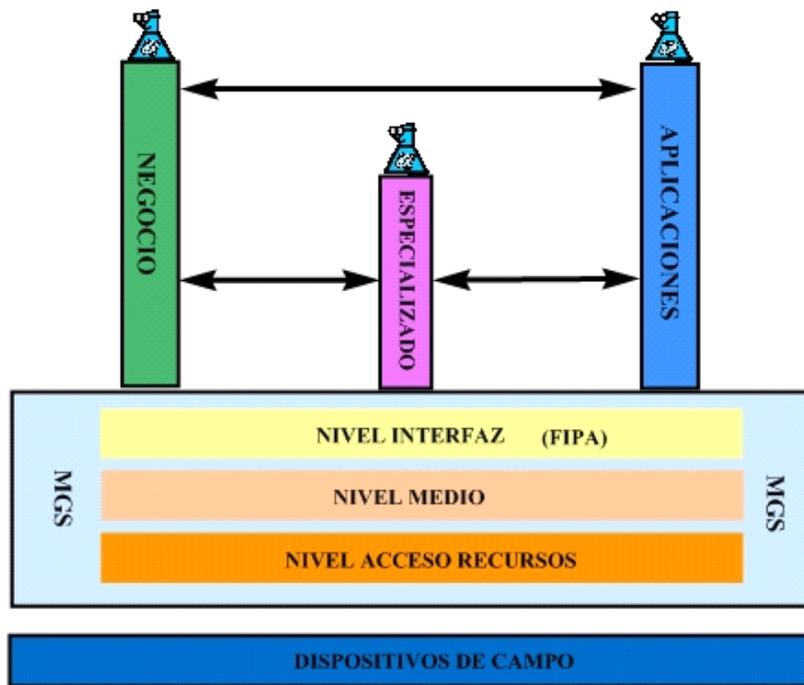


Figura 7.3. SMA de la unidad de explotación de yacimientos petrolera

Ahora bien, el nivel que nos interesa en este trabajo es el nivel superior, ya que allí es donde se realizan todas las tareas de automatización de procesos. Ese nivel está compuesto por dos comunidades de agentes. Una de ella es la comunidad de *Agentes de Negocio*, los cuales describen los diferentes objetos de negocio (abstracciones lógicas y funcionales de los procesos reales). La otra comunidad es la de los *Agentes de Aplicaciones y Especializados*, conformada por todos los agentes que realizan funciones específicas a nivel de supervisión, control, optimización, visualización, y otras aplicaciones especializadas y/o legadas. Esta comunidad brinda servicios a la comunidad de agentes de negocios. Ambas interactúan con el MGS a través de la capa de interfaz. Además, cada uno de los agentes específicos de supervisión y control, están compuesto por otras comunidades de agentes (más adelante se detallan).

Modelo de agentes: Para el modelo de agentes haremos referencia a los agentes del nivel superior solamente (porque es donde se modela el proceso de automatización, como se dijo previamente). A continuación se presentan las comunidades de agentes que conforman el nivel superior, que son: Agentes de negocio, Agentes de aplicaciones y Agentes de aplicaciones especializadas.

Agente de Negocio: Los Agentes de Negocios modelan los elementos de las unidades de producción, cada unidad de producción está representada por un Agente de Negocio. La composición de un Agente de Negocio está basada, por un lado, por una división física del proceso y por otro lado, por una división funcional de las tareas del agente.

Agente de Aplicaciones: Los Agentes de Aplicaciones fueron concebidos como SMA especializados en coordinar, ejecutar y evaluar las tareas de control, supervisión, optimización y gestión, necesarias en el procesamiento de la información del proceso y la toma de decisiones. Dicho SMA está conformado por los siguientes agentes [52]:

- Agente de Supervisión
- Agente de Control
- Agente de Gestión de Optimización
- Agente de Gestión
- Agente de Manejo de Situaciones Anormales: forma parte de la comunidad de agentes de Aplicaciones de Supervisión, en particular es una instancia del Agente Supervisor de Confiabilidad.
- Agente Repositorio
- Agente de Visualización

Algunos de esos agentes, a su vez, está formado por otros SMA. Así, tenemos que el *agente de supervisión* se diseña como un SMA conformado por los siguientes agentes (ver figura 7.4):

- Agente Supervisor del Control del Proceso.
- Agente Supervisor de Confiabilidad.
- Agente Supervisor de Tareas.

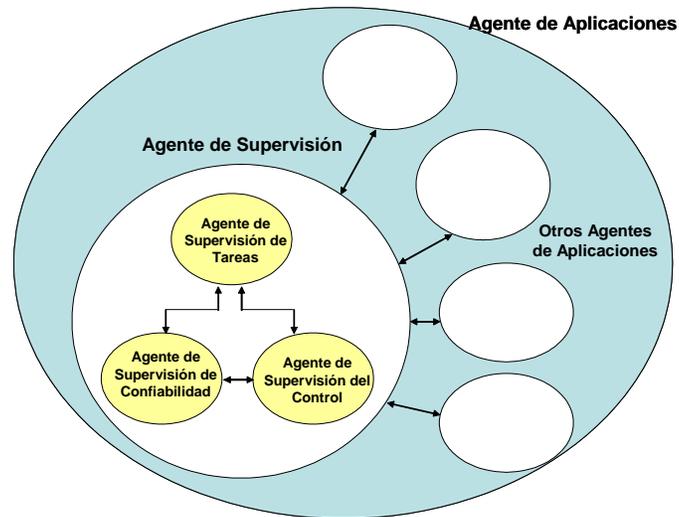


Figura 7.4. Modelo de referencia del agente supervisión

El agente de control se diseña como un SMA conformado por los siguientes agentes (ver figura 7.5):

- Agente Diseñador del Control.
- Agente Ejecutor del Control.
- Agente Evaluador del Control.

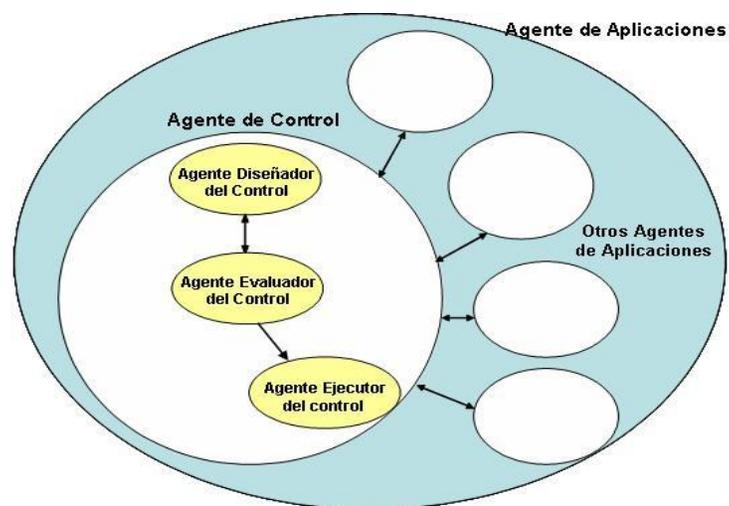


Figura 7.5. Modelo de referencia del agente de control

El agente de gestión de optimización se encarga de suministrar las mejores soluciones de estado, a fin de lograr la máxima eficiencia y eficacia en los procedimientos, en su coordinación y en su coherencia.

El agente gestión tiene tareas orientadas a actividades gerenciales y organizacionales, por consiguiente, definen todos los planes estratégicos de producción y su evaluación durante la ejecución.

El Agente de Manejo de Situaciones Anormales propuesto en este proyecto, se enmarca dentro de los Agentes de Aplicaciones de Supervisión, de esta forma, ha sido definido como una instancia del Agente Supervisor de Confiabilidad.

El Agente Repositorio es un Agente de Aplicaciones que se encarga de establecer los atributos básicos que pueden tener los distintos repositorios de datos requeridos por la plataforma de automatización.

El Agente de Visualización es el encargado de presentar la interfaz humano-computadora de los distintos módulos programados. Este agente tiene la capacidad de configurarse en función de los distintos roles de los usuarios, permitiendo el despliegue de los datos e información entre los diferentes dispositivos de salida y/o aplicaciones de una manera transparente.

Los Agentes Especializados: se encarga de ejecutar tareas específicas de búsqueda y procesamiento de datos e información a partir de los datos proporcionados.

Los detalles de este proyecto, y de los agentes propuestos, pueden ser revisados en [52]. En este caso de estudio analizaremos el problema de integración híbrida usando CLEMAS.

7.2 Uso de las formalizaciones de los mecanismos de coordinación en los casos de estudio

En esta sección se presenta el uso de los formalismos de los mecanismos de coordinación, propuestos en este trabajo, en los casos de estudio. Se presenta el uso de la subasta, y licitación para el caso de estudio A, y luego el formalismo de planificación para el caso de estudio B.

7.2.1 Ejemplo de una subasta inglesa para la conversación de tareas urgentes del caso de estudio A.

La conversación tareas urgentes [50] es modelada usando el mecanismo de subasta Inglesa anteriormente expuesto, ya que en ella existen agentes y requerimientos que pueden ser asociados con los escenarios de subasta. Para esto, se detalla a continuación su diagrama de interacción (ver figura 7.6) y el modelo de coordinación según la metodología MASINA.

Conversación: Tareas Urgentes

- Agentes participantes: Coordinador (ACoo), Controlador (ACon), Base de Datos (ABD), Actuador (AA), Observador (AO) y Humano (AH).
- Iniciador: Coordinador
- Actos de habla: órdenes urgentes, orden de mantenimiento, reporte de tareas de mantenimiento, tareas pendientes, buscar tareas pendientes, y alarma.
- Precondición: tener que realizar tareas urgentes en el sistema.
- Condición de terminación: si no se pudo realizar la tarea urgente se da una alarma.

Esquema de Coordinación

- Objetivo a seguir: Planificar las interacciones entre los agentes Coordinador, Base de Datos, Controlador, Actuador, Observador y Humano.
- Tipo: predefinido
- Coordinación por Defecto: Centralizada, a través de mecanismos de pases de mensajes para relacionar los agentes que participan en la conversación [50].

Planificación

Tipo: Predefinido

Técnica: Existen protocolos, consulta, requerimientos y contratación entre el agente Base de Datos y los agentes Coordinador y Observador. Los agentes Observador, controlador con el agente actuador. Finalmente el agente Coordinador con el agente Humano.

Mecanismo de Comunicación

- Tipo: Directa
- Técnica empleada: Pase de mensajes
- Metalenguaje: KQML

Como se expuso en párrafos anteriores, en este SMF los agentes deben cumplir con diversas tareas de mantenimiento, para mantener un nivel óptimo en el sistema. En algunas circunstancias, bien sea por falta de presupuesto o falta de material, muchas de estas tareas no se llevan a cabo, quedando pendientes en los agentes bases de datos del sistema (como se dijo antes, estos agentes forman parte del Medio de Gestión de Servicios (MGS)).

Uno de los mecanismos que sirve para gestionar la realización de estas tareas puede ser la subasta, cuyo iniciador o subastador sería un agente coordinador del sistema, y los participantes serían los agentes bases de datos que poseen tareas de mantenimiento pendientes por realizar.

Instanciación de Subasta Inglesa en la Conversación de Tareas Urgentes

Se plantea el siguiente caso: existen cuatro (4) plantas agrupadas en una unidad de producción, gestionadas por un único agente Coordinador de la unidad de producción. Además, cada planta que integra esta unidad posee un agente Base de Datos que almacena las tareas de mantenimiento pendientes de cada planta. Por otro lado existe limitación de recursos para ejecutar las tareas de mantenimiento, por lo que los agentes Base de Datos deben “competir” entre ellos por un lugar de preferencia en la lista de ejecución de tareas urgentes del Coordinador.

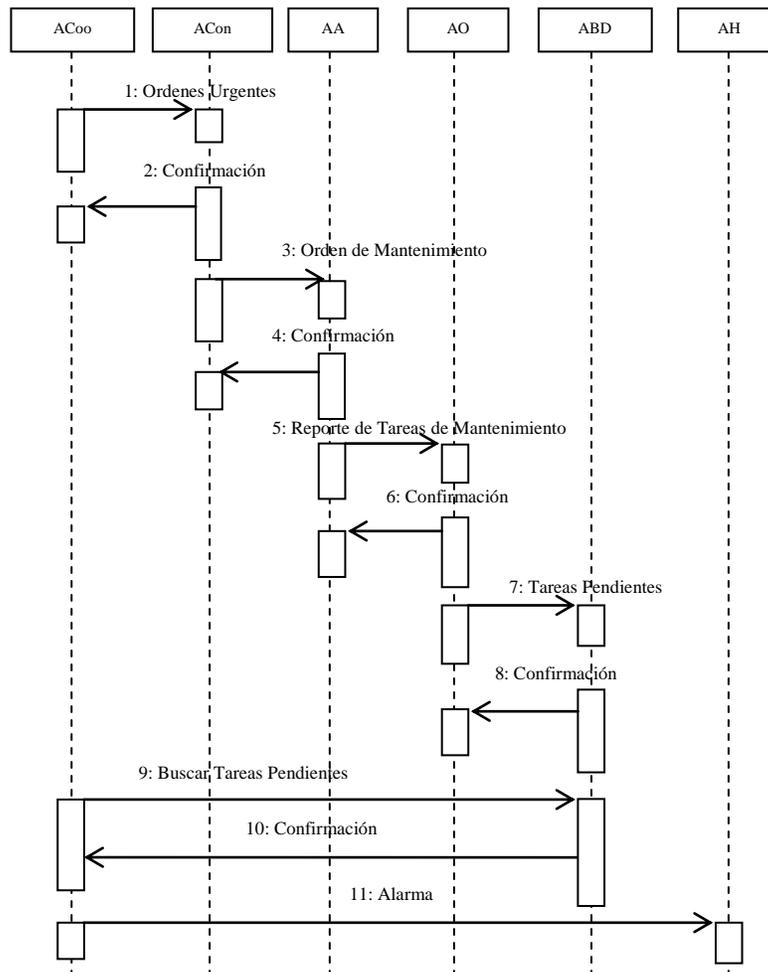


Figura 7.6. Diagrama de interacción de la conversación tareas urgentes

Este proceso de competencia puede ser visto como una subasta, donde el agente Coordinador es el agente subastador, los agentes Base de Datos serían los ofertantes y el ítem ofertado es un lugar de preferencia para la ejecución de sus tareas urgentes. Las variables de nuestro modelo formal tendrán los siguientes valores:

- Agente iniciador (a_s): agente Coordinador

- Agentes participantes $A_i = \{A_1, A_2, A_3, A_4\}$, 4 agentes Bases de Datos.
- Precio inicial: $C_0 = x$, así cada celda es

$$Of_i^j = \begin{cases} x + \alpha_i^j & \text{para } j = 1 \\ \max A_G^{j-1} & \text{para } j > 1 \end{cases}$$

$$\forall i = 1, \dots, n \quad \forall j = 1, \dots, m.$$

- $\vec{\varepsilon}_i = \{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4\}$. Para este caso en particular llamaremos a ε necesidad de cada agente.
- Condición de parada, $C_p = j > 1$ y $Of_i^j = 0$, es decir, ocurre después de la primera ronda y cuando no hayan más ofertas
- Subasta ascendente

Un ejemplo de la matriz Of sería:

Tabla 7.3. Ejemplo matriz de la Ofertas

Agentes/ Rondas	1 ^{ra}	2 ^{da}
A_1	Of_1^1	$\max A_3^1 + \alpha_1^2$
A_2	Of_2^1	$\max A_3^1 + \alpha_2^2$
A_3	Of_3^1	$\max A_3^1 + \alpha_3^2$
A_4	Of_4^1	-
A_G	$\max = A_3^1$	$\max = A_3^2$

Un ejemplo de la matriz de Or sería:

Tabla 7.4. Ejemplo de la matriz de Orden

Agentes\Rondas	1 ^{ra}	2 ^{da}
A_1	3	2
A_2	4	1
A_3	2	3
A_4	1	-

Las tablas 7.3 y 7.4 muestran que:

- En la primera ronda el agente tres (3) ofertó de segundo (Tabla 7.4), y propuso el mayor valor (Tabla 7.3).
- El agente cuatro (4) no ofertó en la segunda ronda.
- El último elemento del vector fila A_G muestra que el agente ganador es el agente Base de Datos tres (3).

- El precio que debe pagar el agente Base de Datos 3 es $C = \max A_3^2$, y tiene la preferencia del agente Coordinador para realizar sus tareas urgentes.

En la 7.7 se muestra el protocolo FIPA para la subasta inglesa pero con la nomenclatura adoptada para el formalismo, donde se aprecia que el agente iniciador en el agente coordinador y los agentes ofertantes son los agentes bases de datos.

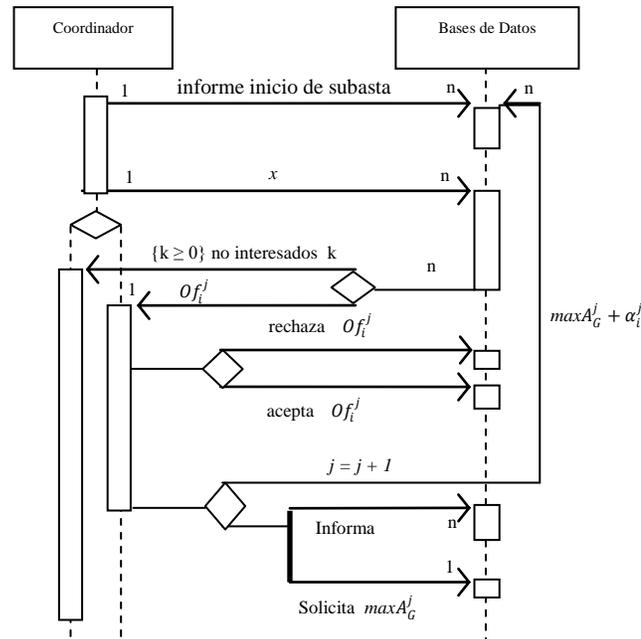


Figura 7.7. Diagrama de interacción para una subasta Inglesa en la conversación tareas urgentes

7.2.2 Ejemplo de licitación en la conversación de mantenimiento por condición del caso de estudio A.

Para este caso se utiliza la conversación de Mantenimiento por condición del SMF [17]. El caso de estudio es un subsistema de bombeo en un área específica. El subsistema tiene el siguiente equipamiento: bomba principal, filtro principal, el filtro de bloqueo, temporizadores electromecánicos, tuberías y válvulas. En un momento dado, el agente controlador envía una solicitud para realizar una tarea de diagnóstico de tres equipos del subsistema de bombeo: bomba principal, temporizadores electromecánicos y válvulas, en una fecha específica. Luego de esta solicitud, dada la magnitud de la tarea, el agente controlador envía una solicitud de realizar dicha tarea a tres agente coordinadores. Estos, posteriormente, envía tres ofertas (a través de un mensaje de difusión) a un grupo de agentes diagnosticadores. A este mensaje responden varios agentes diagnosticadores. Este escenario puede ser visto desde la perspectiva de una red de contrato o licitación, en el que los agentes coordinadores serían los agentes responsable de la tarea de diagnóstico de tres



equipos, y los agentes diagnosticadores existentes será potenciales contratistas. El proceso de licitación es:

- Tipos de tareas: tres tareas de diagnóstico, TDi_1 (bomba principal), TDi_2 (temporizadores), TDi_3 (válvulas).
- Tres agentes administradores (a_a): Tres agentes coordinadores (ACoo1, ACoo2, ACoo3), uno por cada dispositivo.
- Potenciales contratistas para las tareas TDi_1 $A_c = \{A_{11}, A_{12}, A_{13}, A_{14}\}$, para TDi_2 $A_c = \{A_{21}, A_{22}, A_{23}\}$, y para TDi_3 $A_c = \{A_{31}, A_{32}, A_{33}, A_{34}, A_{35}\}$ todos agentes diagnosticadores. Para cada agente el primer índice representa el dispositivo y el segundo el número del agente.
- Mensaje inicial: $M = (TDi, DIR_a, [DD, MM])$
- $f(TDi_j)$, es la función para evaluar la habilidad para realizar la tarea por parte de los contratistas (determina si este oferta o no). Como criterio para ofertar ellos consideran sus habilidades y disponibilidad.
- Vector \vec{O}_c

Tabla 7.5. Ofertas para TDi_1

Ofertas TDi1
$O_{11} = [TCE, TS, TPO, TB]_{11}$
$O_{12} = [TCE, TS, TPO, TB]_{12}$
$O_{13} = [TCE, TS, TPO, TB]_{13}$
$O_{14} = [TCE, TS, TPO, TB]_{14}$

Tabla 7.6. Ofertas para TDi_2

Ofertas TDi2
$O_{21} = [TCE, TS, TPO, TB]_{21}$
$O_{22} = [TCE, TS, TPO, TB]_{22}$
$O_{23} = [TCE, TS, TPO, TB]_{23}$

Tabla 7.7. Ofertas para TDi_3

Ofertas TDi3
$O_{31} = [TCE, TS, TPO, TB]_{31}$
$O_{32} = [TCE, TS, TPO, TB]_{32}$
$O_{33} = [TCE, TS, TPO, TB]_{33}$
$O_{34} = [TCE, TS, TPO, TB]_{34}$
$O_{35} = [TCE, TS, TPO, TB]_{35}$

Las tablas 7.5, 7.6 y 7.7 muestran las habilidades de los agentes (las abreviaciones TCE, TS, TPO y TB significan tareas de cálculo de estadísticas, tareas de selección de técnicas de detección, tareas de procesamiento y ejecución de cálculo de índices y tareas de búsqueda de métodos de detección, están detalladas en [17])

- $g(O_c)$, es el criterio de evaluación de las ofertas por parte del agente administrador. Asumimos como criterio de evaluación la disponibilidad de tiempo de cada agente para realiza la tarea.
- Mensajes a los agentes ganadores (M_p):

Tabla 7.8. Mensaje para los ganadores de cada tarea TDi_j (hipotéticos)

T_j	A_c	M_p
TDi1	A_{13}	$M_p = \langle A_{13}, [accept - proposal] \rangle$
TDi2	A_{21}	$M_p = \langle A_{21}, [accept - proposal] \rangle$
TDi3	A_{34}	$M_p = \langle A_{34}, [accept - proposal] \rangle$

- El vector RP de cada tarea TDi_j

Tabla 7.9: Reporte parcial de cada tarea TDi_j

Tarea	Reportes			
TDi_1	$rp_{3,1}$	$rp_{3,2}$	$rp_{3,3}$	
TDi_2	$rp_{1,1}$	$rp_{1,2}$		
TDi_3	$rp_{4,1}$	$rp_{4,2}$	$rp_{4,3}$	$rp_{4,4}$

Cada reporte tiene dos índices: el primero el número de los reportes. y el segundo el número de los reportes.

- $h(RP, k)$ es una función que evalúa la ejecución de cada agente contratista en cada tarea asignada, tal que cada vector RP corresponde la tarea TDi_j en evaluación.
- El valor de RF para cada vector RP de cada TDi_j sería:

$$RF = \begin{bmatrix} rp_{Total} \text{ para } TDi_1 \\ rp_{3,Total} \text{ para } TDi_2 \\ rp_{45,Total} \text{ para } TDi_3 \end{bmatrix}$$

En resumen, se tiene que:

- El número de agentes ofertantes para cada una de las tareas (de acuerdo a la función $f(TDi_j)$) fueron: para $TDi_1 = 4$ agentes (tabla 7.5), $TDi_2 = 3$ agentes (tabla 7.6), $TDi_3 = 5$ agentes (tabla 7.7), todos diagnosticadores.
- La Tabla 7.8 muestra que cada administrador evalúa las ofertas recibidas, dando como ganadores: Agente 3 para TDi_1 , agente 1 para TDi_2 y agente 4 para TDi_3 , para los cuales se asume que tienen disponibilidad inmediata.
- Para los vectores de reportes parciales (Tabla 7.9), vemos que el agente 4 reporta mayor progreso en su tarea asignada (cuatro veces).
- Finalmente, cada administrador obtiene el reporte final con el diagnóstico de cada uno de los equipos solicitados.

7.2.3 Formalización del mecanismo de planificación para el caso de estudio B.

Para el mecanismo de planificación se utiliza el caso de estudio del SMA para la planificación del proceso de producción petrolera, ya que en él se presenta claramente un caso de planificación.

Especificación del Problema de Coordinación

En este caso, el usuario crea un mensaje con una solicitud a realizar, y se la envía al agente administrador de requerimientos. Este lo recibe, e inmediatamente busca en el directorio al agente administrador de productos; se comunica con este agente a través de un mensaje, donde envía el código de los tipos de productos presentes en la solicitud del usuario. El agente administrador de productos, al recibir la solicitud, busca si están disponibles esos productos. En este caso vamos a suponer que no existen los productos solicitados. Con la información obtenida, este agente crea un mensaje de respuesta para el agente

administrador de requerimientos. Ahora, el agente administrador de requerimientos analiza si el lazo de producción puede satisfacer la solicitud, para lo cual se apoya en el agente administrador de recursos. Suponemos que el agente determina que el lazo de control puede satisfacer la solicitud. Llegado a este punto, el agente administrador de requerimientos se comunica con el agente planificador para que elabore el plan respectivo del lazo de producción. El agente planificador interactúa con los siguientes agentes: agente de identificador de escenarios para recopilar la información de la situación actual, agente analizador de riesgos para determinar los riesgos del plan, y agente de asignación de recursos para resolverlos. A partir de esas interacciones el agente planificador construye el plan del lazo de producción, lo asigna a los diferentes componentes del lazo de producción, y le informa al agente monitor de ejecución para que observe y evalúe la ejecución del plan.

Formalización del Proceso de Planificación

Para formalizar el proceso de planificación del caso de estudio que se ha descrito, comenzamos por identificar cada uno de los roles para este caso.

- Agente iniciador: agente Analizador de requerimiento (AAR)
- Agentes planificador: uno (1) agente planificador (AP), $|AP| = 1$
- Agente que asigna los sub-planes: el mismo planificador, $|AP| = |AA| = 1$
- Agentes ejecutores: varios, cada uno de los componentes del lazo de producción (cada uno visto como un agente), i. e., $|AE| > 1$
 - Pozo (AE_1)
 - Estación de flujo (AE_2)
 - Planta compresora de gas (AE_3)
 - Múltiples de levantamiento artificial por gas (AE_4)
 - Patios de tanque (AE_5)

Dada la cardinalidad de los roles $|AP| = |AA| = 1, |AE| > 1$, se presenta un caso de planificación centralizada para planes distribuidos (PCpD). Antes de mostrar la matriz PCpD, se presenta la lista de servicios y recursos para realizar los sub-planes.

Objetivo general (G):

- Planificar un lazo de producción petrolera por LAG

Sub-planes (P):

- Son varios, uno por cada agente ejecutor

Servicios (S): S_1, S_2, S_3 , siguen siendo planificar, asignar y, ejecutar, respectivamente, pero además, se tienen los siguientes servicios según los requerimientos de producción:

- Tareas de control (S_4)
- Tareas de mantenimiento (S_5)
- Tareas de gestión de inventarios (S_6)
- Tareas de manejo de anomalías (S_7)
- Tareas de supervisión (S_8)

Recursos (R):

- Petróleo (R_1)
- Gas (R_2)

La matriz T que representa el proceso general de planificación según nuestro formalismo es dado en la tabla 7.10

Tabla 7.10. Matriz T(PCpD)

		Servicios (S)		
		$T(PCpD)$	S_1	S_2
Agentes (A)	AP_1	1	1	0
	AE_1	0	0	1
	AE_2	0	0	1
	AE_3	0	0	1
	AE_4	0	0	1
	AE_5	0	0	1

En la matriz 7.10 se aprecia que hay 6 agentes involucrados en cumplir el objetivo. Sólo uno provee el servicio de planificar y asignar, mientras que el resto se encargan de ejecutar el plan que garantiza la gestión del lazo de producción petrolera para la solicitud que está siendo atendida.

La estructura de los sub-planes a ser ejecutada por los agentes ejecutores es la misma. Esa estructura general es mostrada en la matriz SSP en la tabla 7.11:

Tabla 7.11. Matriz SSP

		Plan (P)
		SSP
Servicios (S)	$sp_i - LAG$	
	S_4	1^1
	S_5	1^1
	S_6	1^1
	S_7	1^1
	S_8	1^1

Para $i = 1 \dots 5$ (sub-planes). La matriz que permite representar los agentes que ejecutan los sub-planes (ASP), se muestra en la tabla 7.12, y la matriz que representa el proceso de

asignación de los servicios a los agentes (matriz T de servicios específicos) se muestra en la tabla 7.13.

Tabla 7.12. Matriz ASP

		Sub-planes (P)				
ASP		sp ₁	sp ₂	sp ₃	sp ₄	sp ₅
Agentes (A)	AE ₁	1	0	0	0	0
	AE ₂	0	1	0	0	0
	AE ₃	0	0	1	0	0
	AE ₄	0	0	0	1	0
	AE ₅	0	0	0	0	1

La matriz ASP (ver tabla 7.12) muestra la asignación de sub-planes a los agentes, que en este caso son los mismos para todos, como se indicó en la matriz SSP.

Tabla 7.13. Matriz T

		Servicios (S)				
T		S ₄	S ₅	S ₆	S ₇	S ₈
Agentes (A)	AE ₁	1	1	1	1	1
	AE ₂	1	1	1	1	1
	AE ₃	1	1	1	1	1
	AE ₄	1	1	1	1	1
	AE ₅	1	1	1	1	1

En este caso, la matriz T muestra que todos los agentes de ejecución proveen todos los servicios requeridos por los sub-planes.

7.3 Protocolos de experimentación con CLEMAS en casos de estudio A y C

Para experimentar con CLEMAS se utilizan los casos de estudio A y C. A continuación detallamos los aspectos que nos interesan de esos casos de estudio [52, 53].

7.3.1 Detalles del caso de estudio A

7.3.1.1 Especificación del caso de estudio en CLEMAS

Modelo de coordinación: El SMF tiene seis conversaciones que son: mantenimiento por condición (C1), tareas de mantenimiento (C2), tareas urgentes (C3), replanificación de tareas (C4), estado de mantenimiento (C5), e identificar la falla funcional (C6).

En particular analizaremos C4 y C5

Conversación 4 (C4): Esta conversación se compone de tres sub-conversaciones: C4.1 de tipo TC1, C4.2 de tipo TC3 y C4.3 de tipo TC4.

Descripción: A través de esta conversación, el agente coordinador busca la información del agente de base de datos para reprogramar las tareas de mantenimiento pendientes en el sistema, y hacer un nuevo plan de mantenimiento. Si la tarea es urgente y no puede reprogramar, se da una alarma.

Conversación 5 (C5): Esta conversación se compone de cuatro sub-conversaciones: C5.1 de tipo TC1, C5.2 de tipo TC1, C5.3 de tipo TC3 y C5.4 de tipo TC4.

Descripción: A través de esta conversación, el agente observador solicita información de la base de datos y el agente de actuador, para almacenar las tareas de mantenimiento en circulación en el sistema.

En la figura 7.8, se presenta el diagrama de interacción de la conversación C5, con el fin de mostrar la caracterización de los TC en las sub-conversaciones. En esa conversación el agente observador (AO) hace una consulta (TC1) en la base de datos dos veces (información del proceso (TC1) e información de mantenimiento (TC1)), reporta (informar, TC3) al agente actuador (AA) las tareas de mantenimiento, y si estas no se han realizado, solicitar (TC4) al agente base de datos (ABD) para que las incorpore.

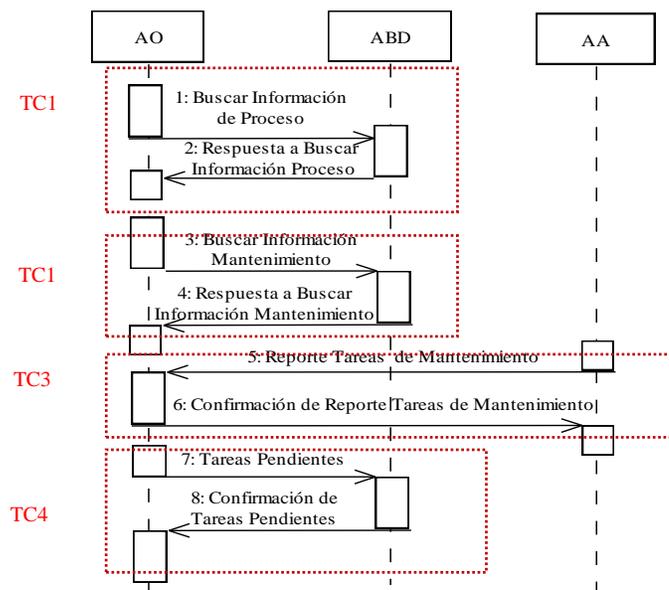


Figura 7.8. Conversación con sus TC

7.3.1.2 Diseño de los experimentos

Para el caso de estudio, se propone el siguiente escenario:

Escenario para las conversaciones C4 y C5: Para este escenario suponemos que C4 optimiza sólo la sub-conversación (C4.1), que tiene 4 agentes (3 agentes de base de datos y 1 agente coordinador), y C5 optimiza solo sub-conversación (C5.3) que tiene 3 agentes (un agente observador y 2 agentes actuadores). El objetivo de la simulación es mostrar cómo el número de iteraciones influye en el proceso de aprendizaje. Para lograr esto, simplemente

se configura CLEMAS con un bajo número de generaciones. El máximo número de rondas de subasta es de 5 y de licitación 1. Los valores iniciales de los parámetros de los mecanismos son, para subasta: $C_0 = [5...15]$ $\varepsilon = [5...20]$, $C_P(j) = [1...5]$. Para licitación $M(F) = [1...3]$, y $f(T) = [5...20]$. Para planificación: $AP = AA = AE = 4$, $PDpD$, con 5 sub-planes. Para esta simulación la población es de 20 individuos, 35 generaciones (35 iteraciones), y las probabilidades de cruce es de 0,7 y de mutación es de 0,5. Posteriormente se aumenta el número de generaciones a 50 para mostrar como mejora el valor de la función objetivo (a través del aprendizaje).

Los resultados de las simulaciones para 35 generaciones, proporcionados por CLEMAS son los siguientes:

Tabla 7.14. Resultados (porcentaje de uso de cada MC en cada TC, basado en el 20% de la población, caso A, 35 gen.)

TC	Licitación	Subasta Inglesa	Subasta Holandesa	Planificación
Consulta (TC1)	96,42%	1,02%	2,14%	0,40%
Asignación (TC2)	0,0%	0,0%	0,0%	0,0%
Informa (TC3)	4,85%	34,65%	59,48%	1,02%
Solicitud (TC4)	0,0%	0,0%	0,0%	0,0%
Total de ocurrencias	48,92%	18,21%	32,14%	0,71%

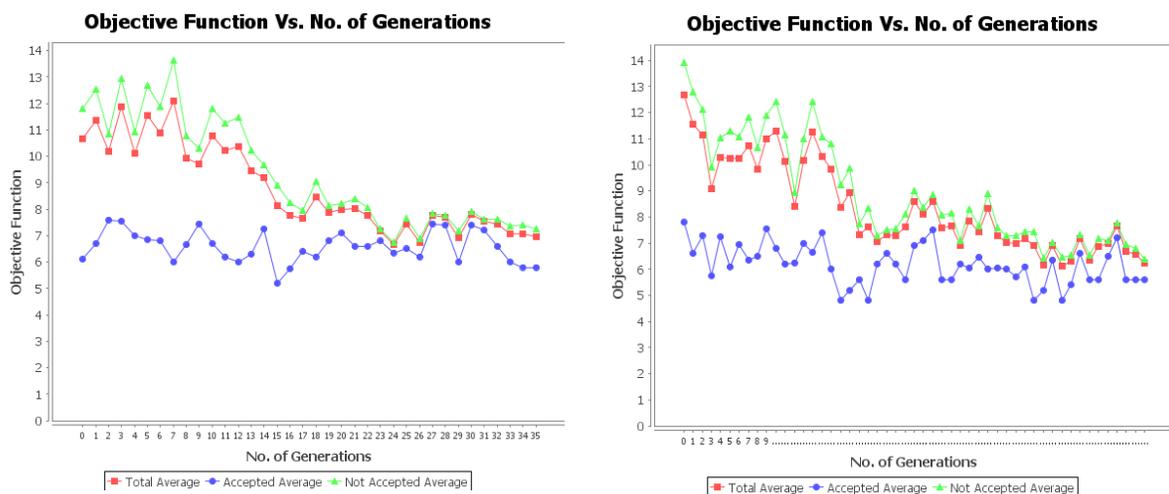


Figura 7.9. Evolución de la función objetivo para el escenario uno

La figura 7.9 muestra la evolución de la función objetivo para este escenario. Los resultados de las simulaciones para 50 generaciones proporcionados por CLEMAS son los siguientes:

Tabla 7.15. Resultados (porcentaje de uso de cada MC en cada TC, basado en el 20% de la población, caso A, 50 gen.)

TC	Licitación	Subasta Inglesa	Subasta Alemana	Planificación
Consulta (TC1)	74.20%	2.60%	22.18%	1%
Asignación (TC2)	0.0%	0.0%	0.0%	0.0%
Informa (TC3)	16.04%	35.13%	68.02%	0,8%
Solicitud (TC4)	0.0%	0.0%	0.0%	0.0%
Total de ocurrencias	41.02%	18.57%	40.10%	0.3%

7.3.1.3 Análisis de resultados

En la tabla 7.14, se observa que para el escenario uno, los TC a optimizar fueron consultar (TC1) e informar (TC3), que son precisamente los TC usado por C4.1 y C5.3 respectivamente. La tabla también muestra para este escenario que licitación ha prevalecido con 49,64% con respecto a otros mecanismos (ocurrencias totales). Esto es porque el TC1 utiliza 96,42% este mecanismo, mientras que TC3 utiliza 60,01% la subasta holandesa. La subasta Inglesa solo se utiliza el 34,65% para el TC3. Para el caso de la planificación, es prácticamente nula su utilización debido a que los casos no ameritan planificación como tal, dando costos de comunicación y de procesamiento muy elevados.

La figura 7.9 por su parte, muestra la evolución de la función objetivo a través de las generaciones. En esa figura, la curva roja (cuadrados) representa el valor promedio de la función objetivo de toda la población. La curva azul (círculos) representa la función objetivo del 20% de la población (individuos seleccionados por la función objetivo, que representa el comportamiento deseado). Finalmente, la curva verde (triángulos) representa el promedio del 80% restante de la población. Este gráfico puede ser visto como el aprendizaje de los individuos, ya que su refleja cómo los individuos reducen sus costos (FO) mediante el uso de los mecanismos de coordinación apropiados en el tiempo (van aprendiendo). Así, vemos cómo las curvas tienden a seguir el de los mejores individuos (círculos), a pesar de que el número de generaciones es bastante pequeño (no influye en número de generaciones a partir de un momento dado).

En la tabla 7.15 se muestra cómo cuando se aumenta el número de generaciones los resultados cambian poco, a pesar que con el número de generaciones los individuos

adquieren más conocimiento. Como análisis final de resultados, para ambos escenarios, a pesar de que TC1 inicia con subastas y TC3 con licitación, los individuos eligen licitación para TC1 y subasta para TC3. Además, el aumentar el número de generaciones no mejora significativamente los resultados.

7.3.2 Verificación de la capacidad de integración en los sistemas de automatización basados en SMA usando CLEMAS y el caso de estudio C

Par realizar esta verificación se utiliza el caso de estudio C, ya que representa un sistema completo con diferentes niveles, en donde en cada nivel existe un SMA. En este caso de estudio se busca verificar los dos tipos de integración estudiados durante toda la tesis, la integración vertical y horizontal.

7.3.2.1 Detalles del caso de estudio

El caso de estudio C propone un SMA de varios niveles para modelar una unidad de explotación de yacimiento (UEY) de un proceso de producción petrolera. Supongamos que se necesitan realizar las siguientes tareas (cada una genera una conversación en el SMA):

1. Monitorear el proceso del agente patio de tanques
2. Diseñar el control de un agente pozo
3. Evaluar el desempeño del plan de control del agente estación de flujo.

Escenario de la tarea 1: En este escenario, como en los restantes, se presentan los dos tipos de integración, es decir, a medida que los agentes de un mismo nivel requieran comunicarse (y lo logren), se estará dando la integración horizontal. Por otro lado, cuando se comuniquen los SMA entre el nivel superior y los niveles inferiores (por jerarquía) se estará dando la vertical. En este escenario, el agente supervisor de confiabilidad inicia el monitoreo del agente patio de tanques (objeto de negocio). Para esta tarea, el agente supervisor inicia la conversación que permite el acceso a los datos del proceso en tiempo real, estimar sus tendencias futuras, analiza los resultados y genera información de manera clara e interpretable. Si es necesario, el agente genera alarmas y solicita manejo de fallas. Estos resultados son transmitidos al agente de visualización para que sean presentados al humano.

La figura 7.10 muestra el diagrama de interacción de esta conversación, los agentes involucrados, los diferentes TC, así como el tipo de integración que se da. A diferencia de lo mostrado en el capítulo 3, aquí se detalla específicamente en qué TC se da la integración, y su tipo. Por ejemplo, en el primer TC1 se presenta una integración vertical (IV) ya que se comunican el agente supervisor de confiabilidad (nivel superior) con el agente gestor de datos el cual pertenece al nivel MGS (nivel inferior). Para el caso de los siguientes TC1 se da una integración horizontal (IH), ya que el agente patio de tanques y el agente repositorio pertenecen al mismo nivel. Lo mismo sucede con los agentes supervisor de confiabilidad y visualización.

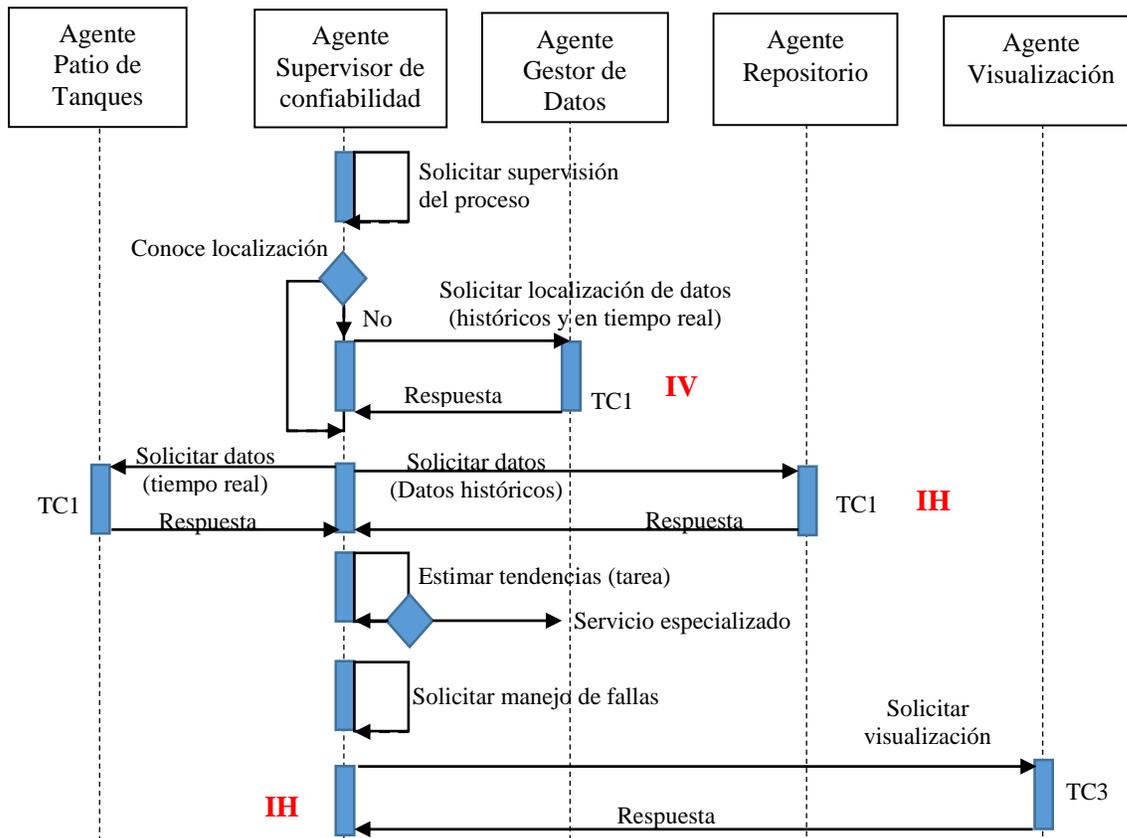


Figura 7.10. Diagrama de interacción de la conversación monitorear proceso

Escenario de la tarea 2: En este escenario, el agente pozo solicita al agente diseñador de control diseñar una estrategia de control. Para esta tarea el agente debe localizar los datos, solicitarlos al agente pozo y gestor de datos, determinar requerimientos de procesamiento de control, planificar el control, para posteriormente transmitirlos al agente pozo.

En la figura 7.11 muestra esa conversación, la cual se inicia con una integración horizontal (IH) cuando el agente pozo solicita al agente diseñador de control realizar la tarea de diseño de control. El agente diseñador de control se comunica con el gestor de datos dando lugar a una integración vertical (IV). Finalmente existe comunicación entre los agentes pozo, diseñador de control, y repositorio quienes se encuentran en el mismo nivel superior.

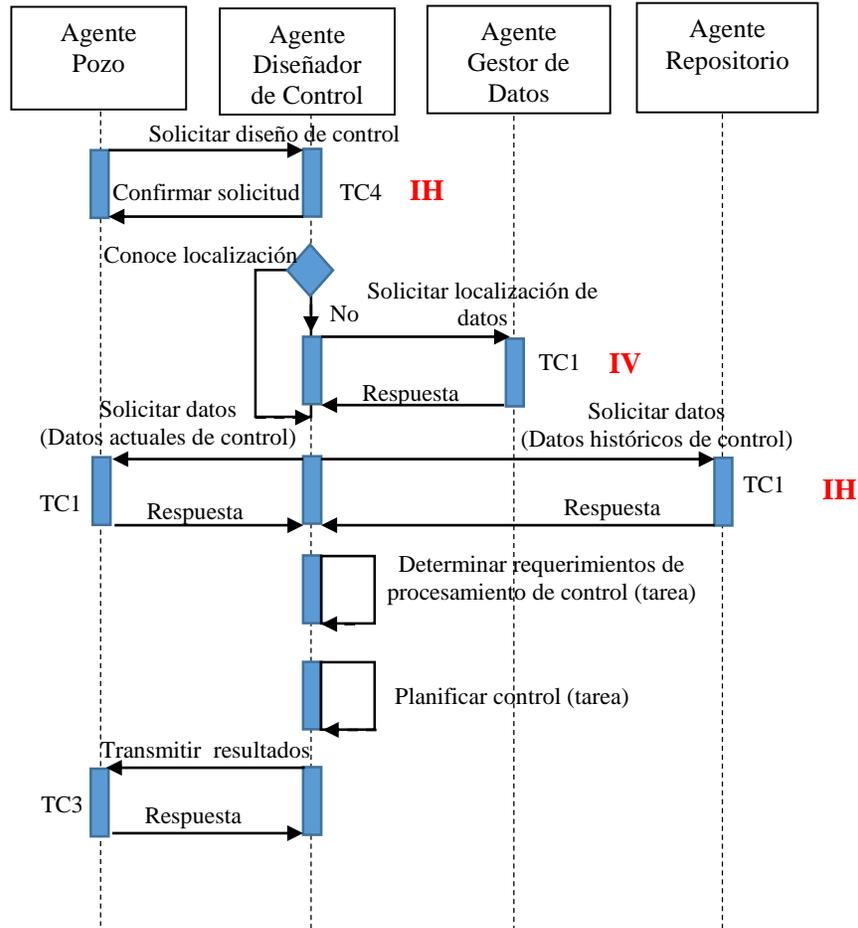


Figura 7.11. Diagrama de interacción de la conversación diseñar control

Escenario de la tarea 3: En este escenario, el agente estación de flujo solicita al agente evaluador de control, realizar una evaluación al desempeño del plan de control. Para esto, el agente evaluador de control solicita datos actuales de control al agente estación de flujo y datos históricos de control al agente repositorio. Cuando finalizada su tarea este transmite la información al agente estación de flujo.

En la figura 7.12 se presenta un la conversación respectiva, con los tipos de integración que se dan.

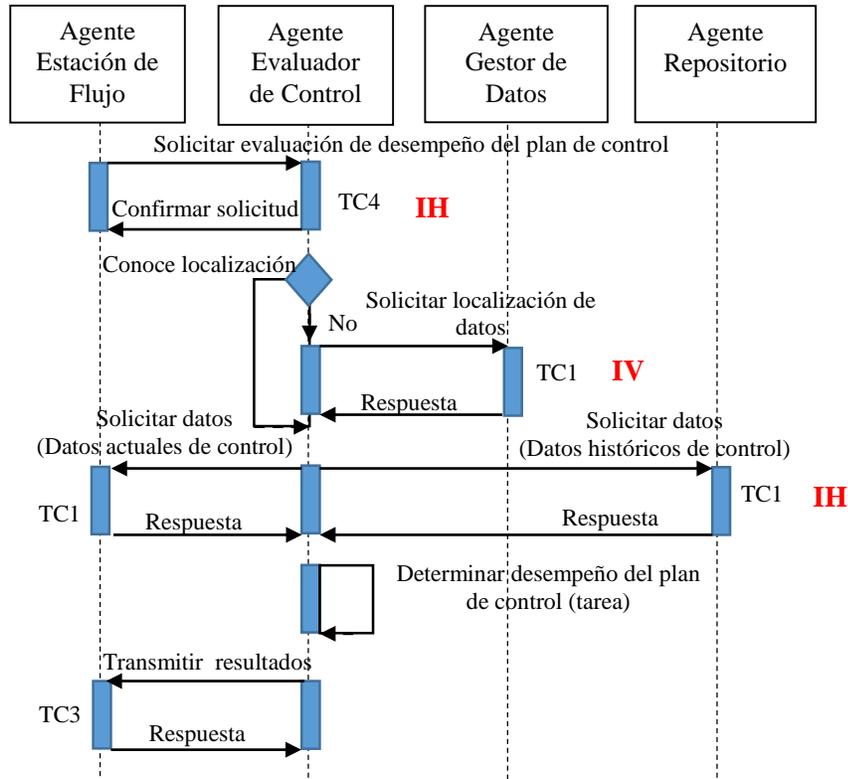


Figura 7.12. Diagrama de interacción de la conversación evaluar desempeño del plan de control y sus TC

7.3.2.2 Diseño de los experimentos

El individuo está conformado por las tres conversaciones mostradas en las figuras 7.10, 7.11, y 7.12. Al inicio de la simulación a cada TC se le asigna un mecanismo de coordinación. Los valores iniciales de los parámetros de los mecanismos son, para subasta: $C_0 = [4...10]$, $\varepsilon = [5...20]$, $C_P(j) = [1...5]$. Para licitación $M(F) = [4...10]$, y $f(T) = [5 ... 20]$. Para planificación: $AP = 1$, $AA = 1$, $AE = 3$, $PCpD$, con 6 sub-planes El resto de los parámetros son:

- Población: 50 individuos
- Numero de generaciones: 65
- Probabilidad de cruce: 0.7
- Probabilidad de mutación: 0.07

Los resultados de las simulaciones proporcionados por CLEMAS son los siguientes:

Tabla 7.16. Resultados (porcentaje de uso de cada MC en cada TC, basado en el 20% de la población, caso C)

TC	Licitación	Subasta Inglesa	Subasta Holandesa	Planificación
Consulta (TC1)	55,36%	5,20%	37,90%	1,53%
Asignación (TC2)	0,0%	0,0%	0,0%	0,0%
Informa (TC3)	30,22%	33,18%	34,80%	1,78%
Solicitud (TC4)	90,39%	5,21%	1,32%	3,07%
Total de ocurrencias	55,01%	12,55%	30,32%	2,12%

En la tabla 7.16, el mecanismo con mayor número de ocurrencias fue licitación con 55,01%. La tabla no muestra resultados para TC2, ya que ninguna conversación la incluye. Ahora bien, para consultar el mecanismo más escogido fue licitación, para informar fue la subasta holandesa, mientras que para solicitar fue licitación.

7.3.2.3 Análisis de resultados

El análisis de los resultados en esta parte se hará en base a las hipótesis propuestas en el capítulo tres. A continuación se mostrará la verificación de cada hipótesis, se hará contrastando los resultados obtenidos en este capítulo con dichas hipótesis:

1ra Hipótesis: Los agentes pertenecientes a diferentes comunidades de agentes, al coordinarse en sus diferentes conversaciones, logran *integrarse tanto horizontal como verticalmente*, lo que les permite ejecutar plenamente sus tareas.

Verificación.- CLEMAS optimiza los mecanismos de coordinación usados en las diferentes conversaciones de un SMA, sin importar a que nivel/comunidad pertenecen los agentes. De esta manera, al coordinarse eficientemente los SMA (optimización de los mecanismos de optimización a usar, por parte de CLEMAS), se garantiza la integración de cualquier tipo (horizontal y vertical), lo que posibilita la ejecución plena de sus tareas.

2da Hipótesis: Al garantizarse la comunicación entre los diferentes de agentes, sin importar el nivel de abstracción al que pertenecen (comunidades de agentes), se permite que los agentes hablen entre sí, enviando y recibiendo datos. Esto posibilita la *integración de datos*.

Verificación.- En los diagramas de interacción presentados en este capítulo se pueden observar los intercambios de mensajes entre los individuos. En particular, las conversaciones están compuestas por actos de habla (comunicaciones) entre los agentes. En algunos casos la comunicación es entre agentes del mismo nivel, o entre agentes de diferentes niveles jerárquicos, enviándose y recibándose información entre ellos. Así, los actos de hablan develan la integración de datos presente en las conversaciones. CLEMAS

los considera implícitos en las conversaciones, cuyos mecanismos de coordinación optimiza.

3ra Hipótesis: Cualquier modelo de SMA Orientado a Servicios (como SADIA), al obtener los mecanismos de coordinación adecuados entre los agentes (bien sean de negociación o planificación), permite orquestar la ejecución de los servicios entre los agentes con un mínimo de conflicto entre ellos. Esto permite la *integración de servicios*.

Verificación.- Los mecanismos de negociación (subasta, licitación) y planificación son bastante utilizados a la hora de resolver conflictos de uso de recursos y de orden de ejecución de servicios entre agentes. En específico, ellos permiten orquestar (planear) la ejecución del conjunto de servicios presentes en una conversación. CLEMAS determina cuál de ellos debe ser usado en cada conversación presente en el SMA, para garantizar una buena integración de los servicios a ejecutar durante cada conversación.

4ta Hipótesis: Al trabajar conjuntamente todos los agentes que conforman el sistema automatizado, realizando cada una de sus tareas específicas, se permite la *interoperabilidad entre las comunidades*.

Verificación.- El caso de estudio C presenta varias comunidades de agente en diferentes niveles. Sus conversaciones presentan interacciones entre agentes en un mismo nivel, o agentes de diferentes niveles (ver figuras 7.10 a 7.12). La optimización que hace CLEMAS de las conversaciones presentes en el caso de estudio C, tienen implícito la resolución del problema de integración, y en general de interoperabilidad entre todos los agentes de ese SMA, sin importar en los niveles/comunidades en donde se encuentren (pertenezcan).



CAPÍTULO VIII

CONCLUSIONES

Dentro de las características que poseen los agentes como entidades de software, están las de comunicación, sociabilidad, e inteligencia, entre muchas otras. Cuando estos agentes comparten y desarrollan estas características con otros agentes, forman sociedades denominadas SMA. Estos sistemas generalmente conforman sociedades que pueden relacionarse con otras sociedades, es decir, con otros SMA. Múltiples conversaciones e interacciones se llevan a cabo dentro de los SMA, o con otros SMA, con el fin de lograr sus objetivos. Muchas de estas interacciones pueden presentar conflictos con respecto al uso distribuido y coherente de recursos y servicios, por lo que es necesario abordar dicho problema. Este problema puede ser descrito como la necesidad de coordinación. La coordinación no solo se limita el envío de mensajes entre los agentes (conversar), sino también como se deben relacionar unos con otros, como deben ejecutarse los servicios requeridos en el SMA en un momento dado, entre otras cosas. En los SMA se han venido estandarizando protocolos a seguir, para lograr la coordinación, estos protocolos se denominan mecanismos de coordinación. Así, la coordinación es un problema fundamental en los SMA, que requiere de su estudio. En nuestro trabajo se estudiaron dos tipos de mecanismos de coordinación de SMA, los de negociación y planificación.

Por otro lado, los SMA se presentan muy convenientes para dar soporte al rápido crecimiento de los procesos industriales. Dentro de estos procesos surge la automatización industrial, la cual cada vez es más compleja y requiere de la integración de diferentes módulos operacionales, sin importar su arquitectura, sea jerárquica o distribuida (plana). El problema de integración en la automatización industrial radica generalmente (no siempre) en la dificultad de comunicación entre los procesos y servicios, que conforman una arquitectura de automatización. En este trabajo se ha propuesto abordar el problema de integración en automatización industrial usando SMA, y en particular, analizando los mecanismos de coordinación de las diferentes conversaciones presentes en el SMA. Esto lo hemos logrado sub-dividiendo las conversaciones en sub-conversaciones, las cuales hemos llamado tipos de conversación (TC). Así, cada tipo de conversación es caracterizada por una performativa que describe qué se quiere hacer: consultar, asignar, informar, o solicitar, y para lograr su objetivo usa algún mecanismo de coordinación.

En cuanto a los AC, son un método derivado de la computación evolutiva, la cual está basada en búsquedas aleatorias dentro de poblaciones integradas por posibles soluciones. Los individuos (posibles soluciones) que integran estas poblaciones, pueden ser sometidos a dos operadores genéticos, cruce y mutación, para procrear nuevas generaciones. En específico, los AC están basados en la evolución cultural, donde el proceso de herencia es visto en dos niveles, uno micro que es a nivel genético, y uno macro que es a nivel cultural, y los conocimientos son adquiridos por los individuos a través de generaciones. Así, los AC

se presentan como una poderosa herramienta de aprendizaje en diferentes sociedades de individuos. En este trabajo esa sociedad es representada por instancias de un SMA caracterizado a través de sus conversaciones. El modelo propuesto demuestra que los individuos del AC aprenden de las experiencias colectivas. Específicamente, en los casos experimentales los individuos “saben” que mecanismo seleccionar para obtener mejores resultados (reducción de costos de procesamiento y comunicación).

En particular, el modelo propuesto buscaba alcanzar el siguiente objetivo: proponer un modelo de aprendizaje de esquemas de coordinación para una sociedad de agentes basado en AC, y aplicarlo al problema de la integración en automatización. Para lograr esto fue necesario lograr los siguientes objetivos específicos:

1. *Definir un modelo genérico de coordinación para SMA*

Este modelo genérico se obtuvo con las formalizaciones de los mecanismos de coordinación de negociación y de planificación (ver capítulo cuatro). Se hizo una formalización matemática, con la idea de obtener variables manipulables en cada uno de ellos. Para el caso de la negociación, se parametrizaron los protocolos de subasta (inglesa, holandesa), y para la planificación, los métodos de planificación centralizada para planes distribuidos, planificación distribuida para planes centralizados, y planificación distribuida para planes distribuidos. Esto fue necesario para que los mecanismos pudieran ser manipulados por el proceso evolutivo de los AC.

2. *Extender el modelo obtenido de coordinación usando los AC*

Una vez obtenido el modelo genérico de los mecanismos de coordinación, se adaptó a la estructura de un AC. Vale recordar que estos algoritmos poseen tres componentes principales: población, espacio de creencias, y un protocolo que permite la comunicación entre los dos anteriores (ver capítulo 2, sección 2.2). Se estructuró cada individuo para describir las conversaciones presentes en los SMA, y al final del individuo está el valor de la función objetivo que indica la calidad de los mecanismos de coordinación usados en ellos. Cada conversación es clasificada a través de tipos de conversación (TC), escogiéndose los TC más comunes que se pueden dar en cualquier conversación de SMA. Estos TC están basados en los protocolos de interacción de FIPA, las que usamos en nuestro trabajo son:

1. Consulta (TC1)
2. Asignación (TC2)
3. Información (TC3)
4. Solicitud (TC4)

El espacio de creencias se nutre con dos categorías de conocimiento, el circunstancial y el normativo. El circunstancial contiene las experiencias exitosas y fallidas de los individuos. El conocimiento normativo contiene los rangos de valores idóneos de cada una de las

variables que conforman a los mecanismos de coordinación. Finalmente, se propuso un operador de mutación guiado, para nutrir a la población con el conocimiento en el espacio de creencias.

3. Modelar el proceso de integración en automatización desde la teoría de SMA.

El capítulo tres es dedicado a ese tema. Usando la teoría de SMA se describe como puede ser modelado y resuelto el problema de integración usando SMA, En particular, partiendo de esos principios, se propuso un modelo de aprendizaje colectivo de mecanismo de coordinación para SMA, que tiene implícito la resolución del problema de optimización (ver capítulo cinco).

Además, se desarrolló una herramienta computacional, denominada CLEMAS, que implementa el modelo de aprendizaje que resuelve tal problema de integración (ver capítulo 6). CLEMAS permite de forma interactiva, analizar diferentes escenarios y observar gráficamente los resultados. CLEMAS se presenta como una herramienta útil para el aprendizaje colectivo en las comunidades de los agentes. La ventaja que aporta CLEMAS con respecto a los trabajos previos es que no requiere planificar acciones, ni una información previa del dominio. Otras de las ventajas que presenta CLEMAS son la sencillez y flexibilidad a la hora de adaptarlo a cualquier escenario. El AC (uno de sus componentes principales) acumula todo el conocimiento en el espacio de creencias, para ser reusado por otros SMA para optimizar sus mecanismos de coordinación; es decir, ese conocimiento podrá ser reutilizado permanentemente para el diseño de los mecanismos de coordinación de otros SMA.

4. Incorporar las estrategias de coordinación basadas en AC en el modelo de integración basado en agentes

Para este objetivo se presentaron tres casos de estudio vinculados a problemas de integración en automatización industrial (ver capítulo siete). Los resultados fueron los siguientes:

- Para el caso de estudio A se comprobó que la formalización matemática propuesta se adapta fácilmente a este caso de estudio, pudiendo modelarse sus conversaciones con los mecanismos de negociación (subastas y licitación). Además, con CLEMAS fue posible optimizar los mecanismos de coordinación que usan sus conversaciones, sin ninguna dificultad.
- En el caso de estudio B se comprobó que sus conversaciones pueden ser modeladas con la formalización del mecanismo de coordinación de planificación, adaptándose esta sin ningún problema, siendo un caso de planificación distribuida para planes distribuidos.

- El caso C representa el proceso híbrido de integración, CLEMAS permitió modelarlo. CLEMAS propone la optimización de los mecanismos de coordinación a usar para la integración, tanto horizontal como vertical.

Aportes significativos de la tesis

El modelo de aprendizaje de esquemas de coordinación para SMA basado en AC, realiza varios aportes en las áreas de coordinación en SMA e integración en automatización industrial:

- Los tipos de conversación (TC) permiten generalizar cualquier conversación que se dé en un SMA, sin importar el dominio u objetivo. Los cuatro TC sugeridos en la tesis están basados en los protocolos de interacción de FIPA, y se adaptan a cualquier escenario.
- La formalización matemática  de los mecanismos de coordinación les permite ser adaptado y usado en cualquier contexto. Por ejemplo, modificando los valores de los parámetros para el formalismo hecho para la subasta, se pueden obtener diferentes protocolos, como la inglesa u holandesa. Igual para los demás mecanismos.
- La herramienta computacional CLEMAS se presenta como una herramienta de aprendizaje colectivo de SMA, para discernir cual mecanismo de coordinación es mejor en un contexto dado, basado en las interacciones entre los agentes.
- El modelo de aprendizaje de esquemas de coordinación para SMA basado en AC puede ser adaptado a cualquier ambiente de automatización basado en SMA, sin importar su arquitectura de base (jerárquica, plana, etc.).

Algunos de los trabajos futuros podrían ser

- Incorporar más mecanismos de coordinación a CLEMAS, ampliando así el abanico de opciones para escoger, para otras situaciones de mayor complejidad.
- Aplicar CLEMAS a otros procesos aparte del de automatización, con la finalidad de verificar aún más su adaptabilidad a cualquier proceso basado en SMA.
- Automatizar a CLEMAS para captar y reconocer en línea a los TC que conforman las conversaciones de un SMA, para que en tiempo real pueda optimizar sus mecanismos de coordinación.

Recomendaciones

El modelo de aprendizaje está basado en las conversaciones, así, tiene la limitante de que en los sistemas en donde los agentes no necesitan comunicarse (o conversar) con otros para realizar sus tareas CLEMAS no aplica. Para solventar esto, sería necesario incorporar mecanismos de coordinación como las leyes sociales, en donde una de las entradas de CLEMAS serían las reglas o normas a seguir por los agentes. Esto en general aplica para

las formas de comunicación indirectas, típicos de los sistemas emergentes. En ese sentido, en esos casos se estaría evaluando si los mecanismos de coordinación emergente son adecuados, requiriéndose formalizar, como se hizo con otros mecanismo de coordinación en este trabajo (subasta, planificación, etc.), para poder ser optimizado por nuestro modelo de aprendizaje.

También, sería interesante rediseñar la carga de datos en CLEMAS, agregando la opción de si se desea trabajar fuera de línea o en línea con el proceso. Así, los datos serían cargados por el usuario o los captaría automáticamente del mismo proceso. Esto amerita lógicamente cumplir con los trabajos futuros propuestos para ello.

REFERENCIAS

- [1] Aguilar, J., Rios, A., Hidrobo, F., y Cerrada, M., (Ed.). (2012). *Sistemas Multiagentes y sus Aplicaciones en Automatización Industrial*, Talleres Gráficos Universitarios, Mérida: Universidad de Los Andes.
- [2] García, S. y Sascha, O. (1998). Inteligencia Artificial Distribuida y Sistemas Multiagentes. *Revista Iberoamericana de Inteligencia Artificial*, 6, 1-12.
- [3] Quintero, A., Rueda, S., y Ucros, M. (1995). Agentes y Sistemas Multiagente: Integración de Conceptos Básicos, Grupo de Investigación HYDRA, Dpto. de Ing. de Sistemas y Computación, Universidad de los Andes, Reporte Técnico.
- [4] O'Hare, G. and Jennings N. (1996). Coordination Techniques for Distributed Artificial Intelligence, (pp. 187-210) en *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons.
- [5] De Andrés, T. (Ed.). (2002). *Homo Cybersapiens. La Inteligencia artificial y la humana*, EUNSA: Universidad de Navarra.
- [6] Weiss G. (2002). Learning coordinate actions in multi-agent systems, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, 311-316.
- [7] Reynolds, R. (1999). New Ideas in Optimization (Ed.), *An Overview of Cultural Algorithms* (pp. 367-378). McGraw Hill Press.
- [8] Becerra, R. (2002). Algoritmos Culturales Aplicados a Optimización con Restricciones y Optimización Multiobjetivo. Tesis M.Sc, CINVESTAV, Inst. Pol. Nac., México, D. F.
- [9] Reynolds, R. (2004). Agent-Based Modeling of Cultural Change in Swarm Using Cultural Algorithms. *Proceedings of SWARMFEST* May 9-11.
- [10] Reynolds, R. (1994). An Introduction to Cultural Algorithms. *Proceedings of the Third Annual Conference on Evolutionary Programming*, February 24-26.
- [11] Reynolds, R. and Chung C. (1997). A cultural algorithm framework to evolve multiagente cooperation with evolutionary programming. *Lectures Notes in Computer Science*, 1213, pp. 323-333.
- [12] Reynolds, R. (1999). New Ideas in Optimization (Ed.), *Cultural Algorithms: Theory and Applications* (pp. 367-377). McGraw Hill Press.
- [13] Montilva, J. (2002). Enfoques de automatización e integración de sistemas heterogéneos en empresas de producción". V *Jornadas Científico Técnicas de la Facultad de Ingeniería*.
- [14] Rios, A., Cerrada, M., Narciso, F., Hidrobo, F., y Aguilar, J. (2008). Implantando Sistemas de Control con Agentes Inteligentes, *Rev. Ciencia e Ing.* 29, 249-260.
- [15] Stefano, A. and Ramamoorthy, S. (2013). A game theoretic model and best-response learning method for ad hoc coordination in multiagente systems," The University of Edinburgh Technical Report.

- [16] Mei, J. (2013). Distributed coordination for second-order multi-agent systems with nonlinear dynamics only relative position measurement. *Automatica*, 49, 1419-1427.
- [17] Ricci, A. and Viroli, M. (2005). Coordination Artifacts: A Unifying Abstraction for Engineering Environment-Mediated Coordination in MAS. *Informatica*, 29, 433-443.
- [18] Harsanyi, J. (2003). Games with Incomplete Information Played by Bayesian Players, I-III. *Management Science*. Disponible en http://en.wikipedia.org/wiki/Bayesian_game.
- [19] Rychtyckyj, N. (1999). Using cultural algorithms in industry. In *Proceedings Swarm Intelligence Symposium*, 187-192
- [20] Rychtyckyj, N. and Reynolds, R. (1998). Learning to re-engineer semantic networks using cultural algorithms. *Lecture Notes in Computer Science*, 1447, 179-190.
- [21] Aguilar, J., Bessembel, I., Cerrada, M., Hidrobo, F., y Narciso, F. (2008). Una Metodología para el Modelado de Sistemas de Ingeniería Orientado a Agentes. *Revista Iberoamericana de Inteligencia Artificial*, 12, 39-60.
- [22] Aguilar, J., Hidrobo, F., Rios, A., y Leon, L. (2005). An Architecture for Industrial Automation Based on Intelligent Agents. *WSEAS Transactions on Computers*, 4, 1808-1815.
- [23] Jennings, N. (1995). Developing Industrial Multi-Agent Systems. In *Conf. ICMAS '95*, 423-430.
- [24] Ríos, A., Cerrada, M., Hidrobo, F., y Aguilar, J. (2006). Towards Control and Supervision Systems based on Intelligent Agents. In *Proceedings of the 2nd. WSEAS International Conference on Dynamical Systems and Control*, 36-41.
- [25] Quintero, L. (2009). Un modelo de control inteligente para sistemas de manufactura basado en los paradigmas holonicos y multi-agente. Tesis de Maestría, Universidad Nacional de Colombia.
- [26] Luder, A., Peschke, J., Bratukhim, A., Treytl, A., Kalogeras, A., and Gialelis, J. (2006). The PABADIS PROMISE – Architecture. In *Proceedings of International Congress ANIPLA 2006*
- [27] Nwana, H., Lee, L., and Jennings, N. (1996). Co-ordination in software agent systems. *BT Technol J.* 14, 79-88.
- [28] Wooldridge, M. (Ed.). (2002). *An Introduction to Multi Agent Systems*. WILEY & SON, LTD: University of Liverpool.
- [29] Smith, R. (1980). The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, C-29, 1104-1113.
- [30] La Fournie, C. (2011). Cooperations using Negotiations and the Contract Net Protocol. Disponible en: [pages.cpsc.ucalgary.ca/~laf/601.72/Negotiaion&ContractNet.pdf](http://pages.cpsc.ucalgary.ca/~laf/601.72/Negotiation&ContractNet.pdf).

- [31] Alonso, E. (2001). Learning in Multi-Agent Systems, en *Third Workshop of the UK's Special Interest Group on Multi-Agent Systems*.
- [32] Noble, J. and Franks, D. (2004). Social Learning in Multi-Agent System. *Computing and Informatics*, 22, 1001-1015.
- [33] Vento, J. (2006). Optimización en Automatización. Técnicas Emergentes para la Automatización Integrada de Procesos Industriales. Reporte Técnico #3. Proyecto 2005000170, Fonacit.
- [34] Lenzerini, M. (2002). Data integration: A theoretical perspective. Tutorial at *PODS 2002*, Madison, Wisconsin, USA.
- [35] Thomas, E. (2004). Service-Oriented Architecture. *A Field Guide to Integrating XML and Web Services*, Prentice Hall.
- [36] Wang, L. and Lin., S. (2009). A multi-agent based agile manufacturing planning and control system. *Computers & Industrial Engineering*, 57, 620-640.
- [37] Aguilar, J., Cerrada, M., Mousalli, G., Rivas, F., e Hidrobo, F (2005). A Multiagent Model for Intelligent Distributed Control Systems. *Lecture Notes in Artificial Intelligence*, 3681, 191-197.
- [38] AF Member Organization (2014, October 22). Disponible en: http://www.automationfederation.org/Content/NavigationMenu/General_Information/Alliances_and_Associations/The_Automation_Federation/About1/What_is_Automation_/What_is_Automation_.htm.
- [39] Martínez, F., Aguilar, J., y Bravo, C. (2011). Planificación en Automatización basado en Sistemas Multiagente. *Revista Avances en Sistemas e Informática*, 8, 959-966.
- [40] Terán, J., Aguilar J., y Cerrada, M. (2013). Mathematical Models of Coordination Mechanisms in Multi-Agent Systems. *Clei electronic journal*, 16, 1125-1137.
- [41] Terán, J., Aguilar, J., y Cerrada, M. (2014). Formalización del Proceso de Planificación en Sistemas Multiagente. *Revista de Ingeniería, Investigación y Tecnología*. UNAM (en revisión).
- [42] FIPA (2002). Contract Net Interaction Protocol Specification, Document number: SC00029H.
- [43] de Weerd, M. (2009). Introduction to Planning in Multiagent Systems, *Journal Multiagent and Grid Systems*, 5, 345-355.
- [44] FIPA (2002). Interaction Protocol Library Specification, Document number: XC00025E.
- [45] FIPA (2002). Communicative Act Library Specification, Document number: SC00037J.
- [46] Van Dyke, H. (1996). Visualizing Agent Conversations: Using Enhanced Dooley Graphs for Agent Design and Analysis. In proceedings of the second international conference on MAS (ICMAS'96). 1-8.
- [47] FIPA (2002). Query Interaction Protocol Specification. Document number SC00027H.

- [48] Terán, J., Aguilar, J., y Cerrada, M. (2013). Modelo cultural para la coordinación de sistemas multiagente. Informe Técnico. Universidad de Los Andes.
- [49] Disponible en <http://www.socialresearchmethods.net/kb/scallik.php>
- [50] Cerrada, M., Aguilar, J., Cardillo, J., y Faneite, R. (2007). Agents-Based Design for Fault Management Systems in Industrial Processes. *Journal Computers in Industry*, 58, 313-328.
- [51] Aguilar, J., León, L., Rios, A., Cerrada, M., Hidrobo, F., Narciso, F., Hernández, D., y Bessembel, I. (2005). Informe 4, Documento de Especificación del Nivel Superior. Informe Técnico, FUNDACITE-ULA.
- [52] Terán, J., Aguilar, J., y Cerrada, M. (2014). Cultural Learning for Multi-Agent System and its Application to Fault Management. In *IEEE Congress on Evolutionary Computation (CEC'2014)*, 2188-2195.
- [53] Terán, J., Aguilar, J., y Cerrada, M. (2014). Collective Learning in Multi-Agent Systems Based on Cultural Algorithms. In *Clei Electronic Journal*, 17, 265-276.