

Benémerita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación



**Modelado de Organizaciones Jerárquicas usando
Sistemas Multiagentes Normativos**

TESIS PRESENTADA POR

Luis Erasmo Montealegre Vázquez

PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PUEBLA, PUEBLA.

OCTUBRE DE 2005

Índice general

1. Introducción	10
2. El Paradigma de Agentes	14
2.1. ¿Qué es un agente?	14
2.2. Arquitecturas de Agentes	14
2.3. Sistemas Multiagentes	16
2.4. Comunicación	17
2.5. Cooperación y Coordinación	17
2.5.1. Intenciones comunes	18
2.5.2. Compromisos y convenciones	19
2.5.3. Modelos y planes	19
2.5.4. Normas	21
3. Organizaciones Humanas	22
3.1. Teoría Clásica de la Administración	22
3.1.1. Funciones básicas de la administración	22
3.1.2. Concepto de administración	23
3.1.3. Diferencia entre administración y organización	24
3.1.4. Principios generales de la administración clásica	24
3.2. Teoría Neoclásica de la Administración	26
3.2.1. Principios generales de la administración neoclásica	26
3.2.2. Énfasis en los objetivos y los resultados	27
3.2.3. Principios básicos de la organización	27
3.2.4. Centralización y descentralización	31
3.2.5. Funciones del administrador	32
3.3. Tipos de Organizaciones	37
3.3.1. Organización lineal	37
3.3.2. Organización funcional	39

3.3.3.	Organización de tipo línea-staff	40
3.4.	Teoría de las Relaciones Humanas	41
3.4.1.	El hombre social	41
3.4.2.	Las necesidades humanas básicas	42
3.4.3.	Comunicación	42
4.	Organizaciones y Agentes	44
4.1.	Teoría Organizacional para Sistemas de Cómputo	44
4.1.1.	Elementos de una organización	46
4.1.2.	Estructuras organizacionales	46
4.1.3.	Dinámica	48
4.2.	Sistemas Distribuidos como Organizaciones	48
4.2.1.	Estructura organizacional	49
4.2.2.	Seleccionando una organización	50
4.3.	Estructuras de Coordinación para Organizaciones y Mercados	50
4.3.1.	Modelo	51
4.3.2.	Estructuras alternativas	51
4.4.	Análisis	53
5.	El Modelo	56
5.1.	Sistema Multiagentes Normativo	56
5.1.1.	Conjunto	57
5.1.2.	Atributo	58
5.1.3.	Estado del ambiente	59
5.1.4.	Meta	60
5.1.5.	Agente autónomo	60
5.1.6.	Norma	61
5.1.7.	Agente normativo	64
5.2.	Organización Jerárquica	66
5.3.	HIERARCHY	66
5.3.1.	Recurso	67
5.3.2.	Meta organizacional	67
5.3.3.	Contrato	68
5.3.4.	Servicio	69
5.3.5.	Norma organizacional	69
5.3.6.	Agente organizacional	69
5.3.7.	Proceso administrativo	71

5.3.8. Rol	72
5.3.9. Perfil de puesto	74
5.3.10. Estructuras de coordinación	78
5.4. Vista Global del Modelo	80
6. Aplicaciones del Modelo	81
6.1. Modelo de Casos de Uso	82
6.2. Modelo de Diseño	83
6.2.1. Vista estática	83
6.2.2. Vista de colaboración	101
7. Resultados y Conclusiones	114
7.1. Resultados Obtenidos	115
7.1.1. Un modelo para organizaciones jerárquicas	115
7.1.2. Una guía de desarrollo	117
7.2. Análisis Comparativo	119
7.2.1. El modelo MOISE	119
7.2.2. El modelo AALAADIN	121
7.2.3. Comparativos	123
7.3. Trabajo Futuro	126
A. Metodología y Lenguaje de Modelado	128
A.1. El Proceso Unificado de Desarrollo de Software	128
A.1.1. Características del proceso	129
A.1.2. El producto	130
A.1.3. Vida del proceso	131
A.2. El Lenguaje Unificado de Modelado	133
A.2.1. Características de UML	133
A.2.2. Vistas de UML	134

Índice de figuras

3.1. Diferentes niveles de la organización	28
3.2. Las dos terminales de comunicación de cada cargo	30
3.3. Área de autoridad en función del nivel jerárquico	31
3.4. Organizaciones centralizada y descentralizada	31
3.5. Correcciones y ajustes en el ciclo administrativo	32
3.6. Interacción dinámica de las funciones administrativas	32
3.7. Diferencias entre organización lineal y organización funcional .	40
3.8. Principales propósitos de la comunicación	43
4.1. Estructuras de recursos, habilidades, decisiones	47
4.2. Estructuras de autoridad, comunicación	47
4.3. Estructuras de requerimientos	48
4.4. Tipos de jerarquías	50
4.5. Estructuras de coordinación	53
5.1. Clase <code>Conjunto</code>	58
5.2. Clase <code>Atributo</code>	59
5.3. Clase <code>EdoAmbiente</code>	60
5.4. Clase <code>Meta</code>	60
5.5. Clase <code>AgenteAutonomo</code>	61
5.6. Clase <code>Norma</code>	61
5.7. Normas entrelazadas por no cumplimiento	62
5.8. Normas entrelazadas por cumplimiento	62
5.9. Norma de refuerzo	63
5.10. Norma de recompensa	63
5.11. Clase <code>AgenteNormativo</code>	64
5.12. Clase <code>SMAN</code>	65
5.13. Restricciones de la clase <code>SMAN</code>	65

5.14. Estructuras de coordinación	67
5.15. Clase Recurso	67
5.16. Clase MetaOrganizacional	68
5.17. Clase Contrato	68
5.18. Clase Servicio	69
5.19. Clase AgenteOrganizacional	71
5.20. Roles en HIERARCHY	73
5.21. Partes de un perfil de puesto	75
5.22. Clase PerfilPuesto	76
5.23. Norma que especifica una obligación	76
5.24. Norma que especifica un derecho	77
5.25. Clase PlanParcialGlobal	79
5.26. Diagrama de clases de HIERARCHY	80
6.1. Diagrama de casos de uso	82
6.2. Perfil de puesto perfilDirector	85
6.3. Conocimientos del director como métodos	86
6.4. Agente Director	87
6.5. Objeto director	87
6.6. Perfil de puesto perfilPrefecto	88
6.7. Conocimientos del prefecto como métodos	89
6.8. Agente Prefecto	89
6.9. Objeto prefecto	89
6.10. Perfil de puesto perfilCoordMat	90
6.11. Instancia del perfil de puesto perfilCoordFis	91
6.12. Instancia del perfil de puesto perfilCoordQuim	91
6.13. Conocimientos del coordinador como métodos	91
6.14. Agente Coordinador	92
6.15. Instancia del agente coordinadorMat	92
6.16. Instancia del agente coordinadorFis	92
6.17. Instancia del agente coordinadorQuim	92
6.18. Perfil de puesto perfilProfMat	94
6.19. Instancia del perfil de puesto perfilProfFis	94
6.20. Instancia del perfil de puesto perfilProfQuim	94
6.21. Conocimientos del profesor como métodos	95
6.22. Agente Profesor	95
6.23. Objeto profesorMat	95
6.24. Objeto profesorFis	96

6.25. Objeto profesorQuim	96
6.26. Jerarquía a nivel de roles	96
6.27. Jerarquía a nivel de perfiles de puesto	96
6.28. Norma crearContrato	97
6.29. Norma solicitarMensualidad	98
6.30. Norma coordinarCurso	98
6.31. Norma pagarPersonal	99
6.32. Norma entregarCertificado	99
6.33. Norma crearObjetivosCurso	100
6.34. Norma crearEstadisticasMat	100
6.35. Objeto salon	101
6.36. Objeto oficina	101
6.37. Contexto de la norma crearContrato	101
6.38. Objeto curso	102
6.39. Secuencia Inscribirse al curso	103
6.40. Objeto contratoEstudiante	104
6.41. Metas de la norma crearContrato	104
6.42. Contexto de la norma coordinarCurso	105
6.43. Meta organizacional coordinarCursoMat	105
6.44. Secuencia Crear el plan general	106
6.45. Plan planCurso	107
6.46. Contexto de la norma crearObjetivosCurso	108
6.47. Plan planMatematicas	108
6.48. Meta enseñarPolinomios	108
6.49. Plan planPolinomios	109
6.50. Metas de la norma crearObjetivosCurso	109
6.51. Secuencia Dirigir el curso	109
6.52. Secuencia Control del curso	110
6.53. Contexto de la norma crearEstadisticasMat	111
6.54. Insatisfacción de la meta enseñarPolinomios	111
6.55. Norma repetirExamen	111
6.56. Proceso normativo	112

Índice de cuadros

3.1. Funciones básicas de la empresa vs áreas de la administración	23
4.1. Características de una organización	54
4.2. Componentes de una organización	54
4.3. Tipos de estructuras	55
4.4. Factores para un modelo óptimo	55
7.1. Equivalencias entre MOISE, AALAADIN y HIERARCHY	124
7.2. Equivalencias entre MOISE y HIERARCHY	125
7.3. Equivalencias entre AALAADIN y HIERARCHY	126
A.1. Vistas y diagramas de UML	136

Agradecimientos

Quiero expresar todo mi agradecimiento a la doctora Fabiola López y López por su excelente e invaluable supervisión y apoyo. Gracias por su tiempo, comentarios, sugerencias y correcciones.

No existen palabras para describir la inmensa gratitud que siento, y la increíble fortuna que he tenido al conocerla y trabajar con usted.

Muchas gracias.

Agradezco también a mis revisores: la doctora Cora Beatriz Excelente Toledo, por tomarse la molestia de venir hasta Puebla durante este último año. Gracias también, al doctor Mario Rossainz López y a la doctora Darnes Vilariño Ayala. Mi agradecimiento a todos por sus sugerencias y observaciones, que hicieron de ésta una mejor tesis.

Quiero dedicar este trabajo a mi familia. Gracias por su amor, apoyo y comprensión. Gracias a mi hermana Angeles por creer siempre en mí. Hermanita, sabes que eres gran parte de mi inspiración. Agradezco tu valiosa opinión sobre mi trabajo. Gracias a mis padres Erasmo y María Luisa por apoyarme en todo momento.

Los amo a los tres.

Gracias a todos mis profesores de la maestría por compartirme de su conocimiento y enriquecer esta experiencia.

Gracias a mis amigos: Erick, Alma, Sonia, Marcos, Sugehi y Rocío. Un agradecimiento muy especial a Siboney, este trabajo también es tuyo.

Gracias a todos los que contribuyeron al logro de este objetivo y que tal vez nunca lo supieron: a mi familia, amigos (viejos y nuevos) y alumnos, de no habernos encontrado esto no hubiera sido lo mismo.

Gracias también al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico otorgado para la realización de mis estudios de maestría.

*El bosque sería muy triste si sólo cantaran
los pájaros que mejor lo hacen.*

Rabindranath Tagore

Capítulo 1

Introducción

Los grandes avances tecnológicos en redes, nos brindan la posibilidad de diseñar sistemas de software que permiten la interacción entre aplicaciones que pertenecen a varias empresas o usuarios, donde a pesar de sus diferencias, es posible coordinar sus acciones con el fin de lograr la solución cooperativa de problemas de gran escala.

Los sistemas de software han evolucionado desde las aplicaciones que se ejecutan en una sola computadora hasta sistemas que funcionan en ambientes complejos. Estos ambientes se caracterizan por ser: distribuidos, abiertos y dinámicos.

- *Ambientes distribuidos.* En este tipo de ambientes los sistemas presentan una interacción constante.
- *Ambientes abiertos.* En estos ambientes las aplicaciones se encuentran implementadas en diferentes plataformas de hardware y de software.
- *Ambientes dinámicos.* En un ambiente dinámico los sistemas pueden ser anexados, dados de baja, estar fuera de servicio temporalmente, etc.

Hoy en día requerimos que los sistemas no actúen de forma aislada sino que se comuniquen y que cooperen entre sí para lograr la solución de problemas que van más allá de su funcionamiento individual.

En la actualidad existen escenarios donde las tareas pueden ser *delegadas y automatizadas* por medio de computadoras, algunos ejemplos de estos escenarios son: las compras electrónicas, los organizadores de viajes y los asistentes de vuelo, sólo por mencionar algunos. Por lo tanto, necesitamos que

este tipo de aplicaciones respondan de forma similar a como respondemos los humanos (*human orientation*). Esto implica la necesidad de crear sistemas de cómputo que actúen a nombre de los humanos. Tales sistemas deben tener la capacidad de buscar y pedir ayuda, en caso de tener que llevar a cabo tareas que no puedan realizar debido a sus propias limitaciones (de recursos y procesamiento). Esto último trae como consecuencia el diseño de sistemas capaces de *actuar independientemente* y de tal forma que representen adecuadamente los intereses de los usuarios para los cuales trabajan.

Todas estas necesidades han provocado el surgimiento de una nueva rama de las Ciencias de la Computación: los Sistemas Multiagentes [1]. Los Sistemas Multiagentes (SMA) han dejado de ser una área de interés de la Inteligencia Artificial para convertirse en una área de tecnología de cómputo. Esta nueva área ha recibido aportaciones de teorías provenientes de otras ciencias para crear la llamada: *tecnología basada en agentes*. La tecnología basada en agentes, es un paradigma que proporciona una herramienta para el desarrollo, conceptualización e implementación de sistemas de software. Un agente es una entidad de software inteligente cuya característica principal es la autonomía sobre sus acciones, es decir, es capaz de tomar decisiones para alcanzar sus objetivos.

Por otro lado, el creciente uso de Internet y el middleware disponible para construir aplicaciones distribuidas, nos permite *automatizar* muchas de las *tareas* que se llevan a cabo en una *organización*. Con este propósito, necesitamos un paradigma que nos permita representar cada uno de los *elementos* que conforman una organización, además de los *procesos* que ocurren en ella. Dado que los *humanos* son un elemento clave en una organización y los *agentes* son concebidos como entidades de software que *actúan a favor de un usuario*, los SMA se consideran como el paradigma más adecuado para representar cualquier tipo de sociedad [1]. Un sistema multiagentes se encuentra formado por un conjunto de agentes, los cuales interactúan entre sí.

Existen muchos tipos de modelos para organizaciones, pero las *organizaciones jerárquicas* son las más comunes en nuestras sociedades. Sus principales características son: las *relaciones de autoridad* entre sus miembros y las restricciones en las *líneas de comunicación*. Por otra parte, en una organización, sin importar su tipo, se definen *roles* para los miembros, se establecen *objetivos organizacionales* y se proveen *servicios* a otras organizaciones o individuos. Por otro lado, en una organización debe existir cierto orden social, por lo tanto, las *normas* son un elemento primordial que debe representarse en una organización. Del lado computacional, los Sistemas Multiagentes Nor-

mativos [16] son SMA, donde las normas son usadas como un mecanismo para regular la conducta de sus miembros. Es por esta razón que podemos usarlos para modelar cualquier tipo de grupo social donde sus miembros deban cumplir con ciertas normas.

El objetivo de este trabajo es presentar un modelo para *organizaciones jerárquicas abiertas*, donde los agentes que las conforman *no se conocen de antemano* y están implementados por *diferentes diseñadores*. El modelo debe permitir la creación de organizaciones virtuales formadas por *agentes que trabajen para diferentes usuarios* y que sean *capaces de ofrecer servicios*. Este tipo de organizaciones jerárquicas basadas en agentes resulta útil para automatizar las actividades en una organización humana, ya que *por cada humano se puede tener uno o más agentes que lo representen*.

El modelo que se presenta, utiliza como base un Sistema Multiagentes Normativo [13]. Además, nuestro modelo incorpora conceptos de diferentes teorías de las Ciencias Sociales tales como: las teorías *clásica* y *neoclásica de la administración* [3, 4] y la teoría de las *relaciones humanas* [5].

Nuestro modelo toma en cuenta el *dinamismo* que se produce en las organizaciones debido a las *normas* que las rigen y las *relaciones y obligaciones* que resultan de tales normas. Nuestro enfoque brinda la posibilidad de implementar el modelo con agentes que *no pertenecen* a una sola persona o empresa. Además, permite a los agentes *ingresar y salir* de la organización *de manera voluntaria*, conservando así su autonomía.

El modelo presenta suficiente detalle de forma que la implementación es casi inmediata. El modelo es representado usando el Lenguaje Unificado de Modelado (*Unified Modeling Language, UML*). Elegimos UML [19] y el Proceso Unificado de Desarrollo de Software (*Unified Software Development Process, USDP*) [20], porque ambos son ampliamente usados y se han convertido en estándares de facto dentro de la industria del software. Utilizando UML creamos un modelo con una representación gráfica estándar, la cual es ampliamente conocida y fácil de entender. Por otro lado, tanto UML como el USDP tienen varias herramientas que facilitan la implementación del diseño de un sistema.

El resto de esta tesis está organizado de la siguiente manera. En el Capítulo 2 presentamos los términos básicos relacionados con *agentes* utilizados a lo largo de este trabajo. El Capítulo 3 describe la *teoría sobre organizaciones humanas* en la que basamos nuestro modelo. El Capítulo 4 describe como han sido representadas las *estructuras de coordinación* que componen una organización desde el punto de vista computacional y presentamos un *análisis*

comparativo. El Capítulo 5 describe nuestro *modelo* elemento a elemento, y explica los *diagramas UML* correspondientes. En el Capítulo 6 mostramos la *aplicabilidad* del modelo mediante un ejemplo. El último capítulo de este trabajo presenta nuestras *conclusiones*. Partes de esta tesis han sido presentadas y publicadas [18].

Capítulo 2

El Paradigma de Agentes

2.1. ¿Qué es un agente?

De acuerdo con Wooldridge [21], un agente es un sistema de cómputo situado en algún ambiente, capaz de actuar de manera *autónoma* en ese ambiente con el fin de alcanzar sus objetivos de diseño.

Se dice que un agente es flexible si posee tres características: reactividad, proactividad y habilidad social.

- *Reactividad.* Un sistema reactivo es aquel que mantiene una interacción con su ambiente y *responde a los cambios* que ocurren en él con el fin de seguir siendo útil a sus fines.
- *Proactividad.* Se dice que un agente es proactivo si su comportamiento está *dirigido a alcanzar metas* y a *reconocer oportunidades*, y no que sólo responda a los eventos que ocurran en su ambiente.
- *Habilidad social.* A la habilidad de *interactuar con otros agentes* por medio de algún lenguaje de comunicación para agentes y tal vez cooperar, se le denomina habilidad social.

2.2. Arquitecturas de Agentes

Hasta ahora hemos explicado lo que es un agente y las características que debe poseer. Dada la necesidad de construir agentes que soporten dichas características, surge el área de arquitecturas de agentes. Una arquitectura

de agente describe el arreglo particular de datos, algoritmos y control de flujo que definen el comportamiento del agente. Existen 4 vertientes principales que describen las arquitecturas de agentes.

- *Agentes reactivos*. Este tipo de agentes no posee una representación de su mundo y presentan un comportamiento dirigido por eventos.
- *Agentes deliberativos*. Son agentes con características antropomórficas. Como su nombre lo indica, estos agentes deliberan antes de tomar cualquier decisión. Su comportamiento los lleva a alcanzar metas, es decir, elaboran planes con el fin de lograr sus objetivos.
- *Agentes interactivos*. Este tipo de arquitectura incluye elementos para tratar con la presencia de otros agentes, es decir, poseen mecanismos de comunicación, cooperación y coordinación.
- *Agentes híbridos*. Los agentes híbridos combinan varias de las características ya mencionadas.

En esta tesis se utilizan agentes deliberativos del tipo DBI (*Beliefs, Desires and Intentions*), ya que los agentes en los que estamos interesados deben deliberar acerca de las consecuencias de pertenecer a una organización o a un grupo social.

Los *agentes BDI* tienen su origen en una rama de la psicología social, denominada psicología de las actitudes, la cual establece que el comportamiento humano puede ser predecido y explicado a través de la atribución de actitudes, tales como, creencias, deseos, anhelos, miedos, etc. A este tipo de actitudes se les llama nociones intencionales. De esta manera, podemos diseñar *agentes como sistemas intencionales*. Los agentes BDI trabajan con tres tipos de actitudes mentales

- *Informativas*. Conocimiento, creencias, suposiciones.
- *Motivacionales*. Deseos, motivaciones, metas.
- *Deliberativas*. Intenciones y planes.

Las arquitecturas BDI definen a las *creencias* como la representación que un agente tiene acerca de su mundo, y pueden representarse como un conjunto de estados, que a su vez se representan por medio de predicados. Los *deseos*

o *metas* se definen como los estados a los que un agente quiere llegar, y pueden representarse como un conjunto de estados que no han ocurrido. Las *intenciones* se definen como los medios para lograr una meta, algunas veces son representadas como planes.

2.3. Sistemas Multiagentes

De acuerdo con la definición dada en [1], un *Sistema Multiagentes* (SMA) está formado por un conjunto de agentes los cuales se encuentran situados en un ambiente, en el cual interactúan y a veces cooperan con otros agentes, con los que posiblemente tienen metas en conflicto.

Tipos de sistemas

La *interacción* es el proceso por medio del cual las actividades de los agentes tienen efectos entre sí. Para diseñar un SMA es muy importante entender que tipo de interacciones tendrán los agentes. Con este propósito distinguiremos dos tipos de sistemas multiagentes.

Sistemas de agentes benevolentes. En estos sistemas, los agentes tienen una *meta común global*, donde la prioridad de todos los agentes es satisfacer esa meta. Sus metas individuales serán aquellas que se generen como parte de su actividad hacia la satisfacción de la meta común global. Este tipo de agentes son diseñados explícitamente para trabajar en conjunto, y por lo general son diseñados por el mismo grupo de trabajo. La interacción entre estos agentes es para cooperar y coordinar sus actividades.

Sistemas de agentes interesados en sí mismos. En este tipo de sistemas los agentes *trabajan a nombre de un usuario* o de otro agente al cual representan. Estos agentes se diseñan de manera independiente utilizando tecnologías heterogéneas, tienen metas individuales las cuales deben satisfacer, ya que son su prioridad. Tales metas pueden estar en conflicto con las metas de otros agentes. Este tipo de agentes también tienen capacidades limitadas y necesitan la cooperación de otros agentes para satisfacer sus metas, por lo que necesitan convencer a otros agentes que también son interesados en sí mismos para que se

les proporcione ayuda. A partir de este momento supondremos que las interacciones se darán entre agentes interesados en sí mismos.

2.4. Comunicación

Para que los agentes puedan operar efectivamente dentro de un ambiente multiagentes, dicho ambiente debe proveer de *protocolos de comunicación* para permitir que los agentes puedan intercambiar y comprender mensajes.

La intención de un mensaje es influenciar al receptor del mensaje con el fin de cambiar sus creencias, metas y por lo tanto sus planes para hacer lo que el emisor quiere. Por tal motivo, los agentes utilizan un lenguaje de comunicación para agentes (*Agent Communication Language ACL*) para transmitirse información y conocimiento.

Se ha decidido, dentro de la teoría, separar la *semántica del protocolo de comunicación* de la *semántica del mensaje* en sí, ya que la primera es independiente del dominio, mientras que la segunda no lo es.

Un protocolo de comunicación debe ser compartido por todos los agentes. Éste debe ser conciso y tener un número limitado de primitivas de actos del habla, es por este motivo que surgen KQML y KIF. KQML (*Knowledge Query and Manipulation Language*) es un protocolo de comunicación que define un formato de envoltura para la transmisión de mensajes. KIF (*Knowledge Interchange Format*) es un lenguaje que permite la representación de conocimiento de un dominio particular.

2.5. Cooperación y Coordinación

Cooperación es un término usado en los sistemas concurrentes para describir aquellos sistemas que deben interactuar con otros a fin de realizar alguna tarea. Para lograr con éxito la implementación de una sociedad artificial de agentes, debemos considerar dos criterios: coherencia y coordinación

- La *coherencia* es la medida que nos dice que tan bien se comporta un sistema actuando como una unidad sobre algún criterio de evaluación. La coherencia se puede medir en términos de: calidad de la solución, eficiencia en el uso de los recursos, claridad conceptual de operación, en que medida el sistema se degrada en presencia de incertidumbre o fallas, etc.

- La *coordinación* es el grado al cual los agentes pueden evitar actividades extrañas al momento de sincronizar sus actividades.

En un SMA de agentes interesados en sí mismos, no existe una meta global. Por lo tanto, los agentes *acuerdan trabajar juntos* en la satisfacción de una meta. Una vez que esto sucede el problema clave es la coordinación.

Coordinación es el proceso de hacer que las actividades de un agente no afecten negativamente las acciones de otro. El problema de la coordinación es manejar las dependencias entre las actividades de los agentes. Necesitamos mecanismos de coordinación cuando las actividades de los agentes pueden interactuar. Por ejemplo:

- Cuando los agentes usan el mismo recurso en sus actividades.
- Cuando las actividades de un agente dependen de las actividades de otro.
- Cuando los agentes tiene recursos que podrían facilitar el trabajo de otros agentes.

Siempre que exista una relación entre los agentes será necesario un mecanismo de coordinación. Se asume que la coordinación debe suceder en tiempo de ejecución (coordinación dinámica), es decir, el agente debe ser capaz de reconocer sus relaciones y de manejarlas como parte de sus actividades. Existen varias formas de coordinación dinámica: *intenciones conjuntas/comunes, compromisos y convenciones, modelos de agentes, planes, normas y leyes sociales*.

2.5.1. Intenciones comunes

Esta forma de coordinación está basada en los modelos de trabajo humano en equipo (*human team work*) y consiste en atribuirle actitudes mentales a un grupo de agentes.

- Creencias conjuntas (*joint beliefs*)
- Metas conjuntas (*joint goals*)
- Intenciones conjuntas (*joint intentions*)

El significado de estas actitudes mentales es similar a las de un agente. Si se tiene una intención común los agentes tratan de coordinar sus actividades alrededor de esas intenciones. Partimos del hecho de que las intenciones son el elemento clave, ya que proporcionan la estabilidad y predictibilidad necesarias para la interacción social. También proveen la flexibilidad y reactividad que se necesitan para sobrevivir a los cambios en el ambiente.

Podemos *diferenciar* una meta individual de una meta conjunta en base a los *criterios para abandonar la meta*. Una intención para una meta individual puede abandonarse cuando: la meta se ha alcanzado, cuando la meta es imposible de alcanzar, o cuando la meta ya no existe. Una intención común no puede abandonarse cuando se cree que la meta es imposible de alcanzar, ni cuando la motivación de la meta deja de existir.

2.5.2. Compromisos y convenciones

Para dar soporte a las intenciones conjuntas es necesario utilizar *compromisos* y sus respectiva *convenciones* asociadas.

- Los *compromisos* representan las promesas de un agente por hacer algo.
- Las *convenciones* son los medios para monitorear los compromisos, éstas especifican bajo que circunstancias un compromiso puede abandonarse y como deben comportarse los agentes tanto localmente como con otros agentes cuando esto pase.

Cuando un grupo de agentes se engancha en una actividad cooperativa se dice que tienen un compromiso conjunto hacia el objetivo general. Al mismo tiempo, los agentes adquieren compromisos individuales para realizar las tareas específicas que se les han sido asignadas. Los compromisos sociales son persistentes y tienen asociadas convenciones sociales. Las convenciones sociales proporcionan las guías generales y un marco común de referencia en el cual los agentes pueden trabajar. Es a través de las convenciones sociales que los agentes saben lo que se espera de ellos y lo que pueden esperar de otros agentes.

2.5.3. Modelos y planes

La coordinación usando *modelos* se basa en el hecho de que los agentes pueden tener modelos internos de otros agentes para predecir su comportamiento. Los modelos de otros agentes pueden incluir sus creencias, metas,

intenciones, planes, etc. Este tipo de coordinación también es llamado coordinación sin comunicación.

La coordinación usando *planes*, tiene como finalidad la creación de un plan conjunto. En un plan conjunto las actividades de los agentes deben estar coordinadas. El problema entonces es quién va a construir dicho plan. Se tienen 3 alternativas.

- *Planeación centralizada para planes distribuidos.* Utiliza un sistema de planeación centralizada (puede ser un agente maestro), el cual desarrolla un plan para un grupo de agentes. En dicho plan se define la división y el orden de las actividades. Este agente maestro distribuirá el plan entre los esclavos quienes ejecutarán la orden.
- *Planeación distribuida.* Un conjunto de agentes coopera para formar un plan centralizado. Los agentes que participan en la construcción del plan son agentes especialistas en diferentes aspectos del plan general y en tal medida proporcionarán su ayuda. Los agentes que generan un plan no necesariamente son los que lo ejecutan. Su función es sólo construir el plan.
- *Planeación distribuida para planes distribuidos.* Un grupo de agentes coopera para formar planes individuales de acción. Se debe asegurar que los planes individuales no tengan conflictos. Los agentes serán interesados en sí mismos así que podría ser necesaria la negociación si es que se produce algún problema de coordinación.
- *Planes parciales globales.* El principio fundamental es que los agentes cooperantes intercambian información para analizar conclusiones comunes sobre el proceso que da solución al problema. Se dice que un plan es parcial si las actividades para solucionar un problema no están completamente definidas. Se dice que un plan es global si fue formado entre varios agentes a partir de sus planes locales. Este tipo de planeación involucra tres estadios.
 1. Cada agente decide cuáles son sus metas y genera planes a corto plazo para satisfacerlas.
 2. Los agentes intercambian información para determinar donde es que interactúan los planes y las metas.
 3. Los agentes alteran sus planes locales para coordinar mejor sus actividades.

2.5.4. Normas

Las normas representan restricciones a la conducta de los agentes, las cuales, establecen un balance entre las metas individuales y las metas sociales. Son una forma de establecer el orden y evitar que los agentes ejecuten acciones que puedan dañar a los demás. Además, simplifican la toma de decisiones porque dictan las acciones que deben efectuarse en ciertas situaciones. Por lo tanto, pueden predecir la conducta de los agentes. Existen tres formas de incluir normas en un SMA.

- *Diseño fuera de línea.* Las normas son diseñadas y ensambladas dentro de los agentes.
- *Emergencia de normas.* Las decisiones dentro de un grupo de agentes pueden convertirse en normas si son benéficas para el grupo.
- *Agentes normativos autónomos.* Los agentes adoptan y cumplen normas de acuerdo a sus propias metas y motivaciones.

Capítulo 3

Organizaciones Humanas

Este capítulo presenta los conceptos de la teoría sobre organizaciones humanas. Algunos de estos conceptos son tomados e incorporados a nuestro modelo. La Sección 3.1 trata los conceptos generales de las organizaciones en base a la teoría clásica de la administración [3], la Sección 3.2 explica los conceptos y aportaciones de la teoría neoclásica de la administración [4], la Sección 3.3 describe los diferentes tipos de organizaciones que surgen a partir de los principios de la teoría neoclásica [6]. Por último la Sección 3.4, explica la teoría de relaciones humanas en organizaciones [5].

3.1. Teoría Clásica de la Administración

Según el trabajo hecho en [3], la teoría clásica se distingue por el *énfasis en la estructura* que debe tener una organización para lograr la eficiencia. La teoría clásica parte del todo organizacional y de su estructura para garantizar la eficiencia en cada una de las partes involucradas, sean órganos (secciones, departamentos, etc.) o personas (los ocupantes de algún cargo o los ejecutores de las tareas).

3.1.1. Funciones básicas de la administración

Según la teoría clásica de la administración [3] toda empresa se puede dividir en seis grupos de funciones, a saber:

1. *Funciones técnicas.* Producción de bienes o servicios de la empresa.

2. *Funciones comerciales.* Compra, venta o intercambio.
3. *Funciones financieras.* Búsqueda y gerencia de capitales.
4. *Funciones de seguridad.* Protección y preservación de los bienes y las personas.
5. *Funciones contables.* Inventarios, registros, balances, costos y estadísticas.
6. *Funciones administrativas.* Integración de las cinco anteriores por parte de la dirección. Las funciones administrativas coordinan y sincronizan las demás funciones de la empresa.

Actualmente las seis funciones de la empresa se denominan *áreas de la administración*, como se puede observar en el Cuadro 3.1.

Antes	Ahora
Funciones técnicas	Area de producción, manufactura u operaciones
Funciones comerciales	Area de ventas o de marketing
Funciones financieras y contables	Area financiera
Funciones administrativas	Administración general
Funciones de seguridad	Pasan a un nivel inferior
	Surge el área de recursos humanos

Cuadro 3.1: Funciones básicas de la empresa vs áreas de la administración

3.1.2. Concepto de administración

Las *funciones administrativas* se definen como el acto de *planear, organizar, dirigir, coordinar y controlar* las actividades de una organización. Las funciones administrativas definen las actividades del administrador.

- *Planear:* visualizar el futuro y programa de acción
- *Organizar:* construir las estructuras material y social de la empresa.
- *Dirigir:* guiar y orientar al personal.
- *Coordinar:* enlazar, unir y armonizar actos y esfuerzos colectivos.

- *Controlar*: verificar que todo suceda de acuerdo a las reglas establecidas

Estos elementos en su conjunto constituyen el *proceso administrativo* y se hallan presentes en cualquier nivel o área de actividad de la empresa. Las *funciones universales* de la administración son:

- *Planeación*. Implica evaluar el futuro y tomar previsiones en función de él. Unidad, continuidad, flexibilidad y valoración son los aspectos principales de un buen plan de acción.
- *Organización*. Proporciona todos los elementos necesarios material y social para el funcionamiento de la empresa.
- *Dirección*. Pone en marcha a la organización. Su finalidad es alcanzar el máximo rendimiento de los empleados, de acuerdo con los aspectos generales.
- *Coordinación*. Armoniza todas las actividades de la empresa. Sincroniza los recursos y actividades en proporciones adecuadas y ajusta los medios a los fines.
- *Control*. Verifica si todo marcha en conformidad con lo establecido (plan trazado). Su objetivo es identificar las debilidades y los errores para rectificarlos y evitar que se repitan.

3.1.3. Diferencia entre administración y organización

Administración es un todo y la organización es una de las partes que la conforman. La administración es un conjunto de procesos estrechamente relacionados y unificados que incluyen: la planeación, dirección y control. La organización es estática y limitada, puesto que se refiere al establecimiento de la estructura y la forma. Por lo que la palabra organización se usa con dos significados: como *entidad social* y como *función administrativa* y parte del proceso administrativo.

3.1.4. Principios generales de la administración clásica

La administración como cualquier ciencia se debe basar en leyes o principios. Para su estudio, la teoría clásica de la administración [3] establece los siguientes principios fundamentales.

- *División del trabajo.* Se refiere a la especialización que debe existir en las tareas y las personas que forman la organización, lo cual aumenta su eficiencia.
- *Autoridad y responsabilidad.* Autoridad es el derecho de dar órdenes; la responsabilidad es una consecuencia natural de la autoridad, e implica el deber de rendir cuentas.
- *Disciplina.* Obediencia, dedicación, energía, comportamiento y respeto a las normas.
- *Unidad de mando.* Cada empleado debe recibir órdenes de un sólo superior. A esto último también se le conoce como *principio de autoridad única*.
- *Unidad de dirección.* Se debe establecer un jefe y un plan para cada grupo de actividades que tengan el mismo objetivo.
- *Subordinación de los intereses particulares a los generales.* Los intereses generales deben estar por encima de los particulares.
- *Remuneración del personal.* Debe haber una satisfacción justa y garantizada para los empleados y para la organización en términos de retribución.
- *Centralización.* Concentración de la autoridad en la cúpula de la jerarquía de la organización.
- *Jerarquía o cadena escalar.* Línea de autoridad que va desde el escalafón más alto al más bajo, conocido también como: *principio de mando*.
- *Orden material y humano.* Establece que debe existir un lugar para cada cosa y cada cosa debe estar en su lugar.
- *Equidad.* Amabilidad y justicia para conseguir la lealtad del personal.
- *Estabilidad del personal.* Cuanto mas tiempo permanezca una persona en un cargo, tanto mejor, ya que la rotación tiene un impacto negativo en la eficiencia de la organización.
- *Principio de iniciativa.* Capacidad de visualizar un plan y asegurar personalmente su éxito.

- *Principio de trabajo en equipo.* La armonía y unión entre las personas constituyen grandes fortalezas para la organización.

Estos principios representan un enfoque normativo y prescriptivo ya que determinan cuales son los elementos de la administración (funciones del administrador) y cuales principios generales debe seguir el administrador en su actividad.

3.2. Teoría Neoclásica de la Administración

La teoría neoclásica es una *actualización* de la teoría clásica, redimensionada en los problemas administrativos y el tamaño de las organizaciones actuales. Las características principales de la teoría neoclásica son: énfasis en la *práctica de las funciones administrativas*, énfasis en los *principios generales de la administración*, y el énfasis en los *objetivos y los resultados*.

3.2.1. Principios generales de la administración neoclásica

En la administración, los principios son equivalentes a las leyes en física, pues buscan demostrar una relación causa-efecto. Un *principio* es una proposición general aplicable a determinados fenómenos, que *proporciona una guía de acción*. Los once principios más utilizados, según la teoría neoclásica [4] son:

1. Los objetivos de la empresa y sus departamentos deben estar definidos y establecidos claramente por escrito. La organización debe ser sencilla y flexible.

En cuanto a las actividades

2. Las responsabilidades asignadas a una posición deben reducirse al desempeño de una función sencilla.
3. Las funciones asignadas a los departamentos deben ser homogéneas para que la operación sea más eficiente y económica.

En cuanto a la autoridad

4. En la organización deben existir líneas claras de autoridad de arriba hacia abajo y de responsabilidad de abajo hacia arriba.
5. La responsabilidad y la autoridad de cada posición deben estar claramente definidas por escrito.
6. La responsabilidad debe ir siempre acompañada de la correspondiente autoridad.
7. La autoridad para emprender o iniciar una acción debe delegarse al nivel más próximo posible al escenario de la acción.
8. El número de niveles de autoridad debe ser el mínimo posible.

En cuanto a las relaciones

9. El número de subordinados que un individuo debe supervisar con eficiencia tiene un límite.
10. Cada individuo debe reportarse con un sólo supervisor.
11. La responsabilidad de la autoridad más elevada es absoluta en relación con los actos de sus subordinados.

3.2.2. Énfasis en los objetivos y los resultados

Toda organización existe no para sí misma, sino para alcanzar objetivos y producir resultados. La organización debe estar determinada, estructurada y orientada en función de éstos. De allí, nace el énfasis en los objetivos organizacionales y en los resultados que deben alcanzarse, como un medio para evaluar el desempeño de las organizaciones. Los objetivos son valores buscados o resultados deseados por la organización, los cuales espera alcanzar a través de la eficiencia en su operación. Si esta operación falla, los objetivos o resultados se alcanzan parcialmente o simplemente se frustran. Los *objetivos justifican la existencia y operación de una organización.*

3.2.3. Principios básicos de la organización

La teoría neoclásica [4] añade algunos elementos al concepto de organización formal y la define como *un conjunto de posiciones funcionales y jerárquicas orientadas a la producción de bienes y servicios.* Los principios

fundamentales de la organización formal son; *especialización, jerarquía, y distribución de la autoridad-responsabilidad.*

División del trabajo

Para su eficiencia, la producción debe basarse en la división del trabajo, la cual consiste en descomponer un proceso complejo en una serie de pequeñas tareas. La división del trabajo obliga a que la propia empresa se divida en tres niveles que conforman el aparato administrativo necesario para dirigir la ejecución de tareas y operaciones.

1. *Nivel institucional*, compuesto por los dirigentes y directores de la organización.
2. *Nivel Intermedio*, conformado por los gerentes.
3. *Nivel operacional*, formado por los supervisores que administran la ejecución de las tareas y operaciones de la empresa.

En la Figura 3.1 podemos observar los tres niveles antes mencionados, los integrantes de la organización que conforman cada nivel, así como las actividades que deben realizar.

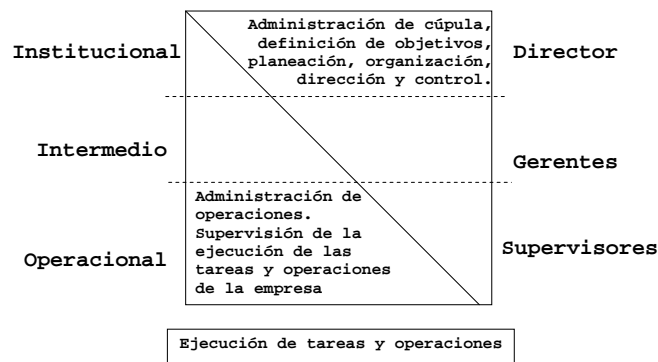


Figura 3.1: Diferentes niveles de la organización

Especialización

La especialización es una *consecuencia de la división del trabajo*: cada órgano tiene a su cargo funciones y tareas específicas y especializadas. La

simplificación de las tareas y la asignación a cada puesto de trabajo de tareas sencillas y repetitivas que requieren poca experiencia y pocos conocimientos previos, reducen los periodos de aprendizaje y *facilitan la sustitución de unos individuos por otros*, lo cual *mejora los métodos de incentivos en el trabajo* y aumenta de modo decisivo el rendimiento de la producción.

Jerarquía

En toda organización formal existe una jerarquía por escalas, estratos o *niveles de autoridad*, de los cuales los superiores tienen cierta autoridad sobre los inferiores. Según la teoría neoclásica, la *autoridad es el derecho formal y legítimo de tomar decisiones, dar órdenes, y asignar recursos para conseguir los objetivos previstos por la organización*. La autoridad distingue tres características:

1. *La autoridad descansa en los cargos de la organización, no en las personas*. La autoridad de los administradores se deriva de los cargos que ocupan.
2. *La autoridad es aceptada por los subordinados*. Los subordinados aceptan la autoridad de los superiores porque creen que tiene derecho legítimo de dar órdenes que deben ser cumplidas.
3. *La autoridad fluye hacia abajo por la jerarquía vertical*. La autoridad va de la cúpula a la base de la organización; las posiciones de arriba tienen más autoridad que las posiciones de la base.

La responsabilidad, es el otro lado de la moneda. Responsabilidad significa el deber de desempeñar la tarea o actividad para la que fue designada la persona. Según los autores clásicos y neoclásicos, la responsabilidad proviene de la relación superior-subordinado y del hecho de que alguien debe tener autoridad para exigir la ejecución de determinadas tareas a otras personas.

Es la *relación contractual* por la cual el subordinado *acepta prestar ciertos servicios a cambio de una recompensa* en dinero u otras formas de retribución. La autoridad fluye del superior hacia el subordinado, cuando se asignan deberes; la responsabilidad es la obligación del subordinado de realizar esos deberes, como podemos observar en la Figura 3.2.

La *delegación*, es el proceso de transferir *autoridad y responsabilidad* a posiciones inferiores de la jerarquía. Las técnicas de delegación de autoridad son las siguientes:

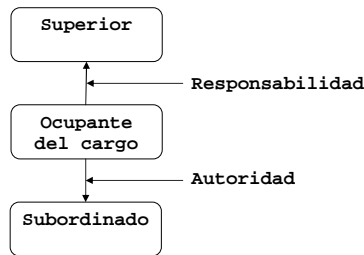


Figura 3.2: Las dos terminales de comunicación de cada cargo

- *Delegar la tarea completa.* El gerente debe delegar una tarea entera a una persona, en vez de subdividirla entre varias personas.
- *Delegar en la persona adecuada.* Todas las personas no tienen las mismas *capacidades* ni *motivaciones*. El gerente debe adecuar el talento de la persona a la tarea para que la delegación resulte eficaz.
- *Delegar responsabilidad y autoridad.* Asignar tareas no es delegar completamente. El individuo debe responsabilizarse de la ejecución de la tarea y la autoridad para llevarla a cabo de la manera que juzgue mejor.
- *Proporcionar información adecuada.* El individuo debe *comprender la tarea y los resultados esperados*, disponer de las provisiones y los recursos necesarios y *saber a quién y cuándo debe presentar los resultados*.
- *Mantener retroalimentación.* Retroalimentación significa abrir líneas directas de comunicación con el subordinado para responder preguntas y proporcionar orientación, sin ejercer control.
- *Evaluar y recompensar el desempeño.* Al finalizar la tarea, el gerente no debe evaluar los métodos empleados sino los resultados. Cuando los resultados no cumplan las expectativas, el gerente debe analizar los errores y las consecuencias. Cuando cumplan o sobrepasen las expectativas, el gerente debe *recompensar* el trabajo bien hecho con estímulos psicológicos y sociales, recompensas financieras y delegación de nuevas actividades.

Distribución de la autoridad y la responsabilidad

La jerarquía de la organización formal representa la *distribución de la autoridad y la responsabilidad* entre los diversos niveles de la estructura. El nivel jerárquico que está por encima de otros tiene mayor peso en las decisiones. En general, el derecho de mandar disminuye a medida que se desciende en la estructura jerárquica. Esta característica puede representarse como una pirámide invertida (Figura 3.3) en la que el supervisor ocupa la parte inferior, lo cual indica que su derecho a mandar está rigurosamente limitado a todas las direcciones. A medida que se asciende en la cadena de mando de la estructura de la organización, el área de autoridad se amplía de manera gradual a cada nivel.

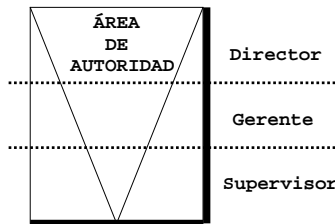


Figura 3.3: Área de autoridad en función del nivel jerárquico

3.2.4. Centralización y descentralización

La centralización y descentralización se refieren al nivel jerárquico en que se toman las decisiones. Centralización significa que la facultad de tomar decisiones se localiza cerca de la cúpula de la organización; descentralización indica que la facultad de tomar decisiones se desplaza a los niveles inferiores de la organización. Esto podemos observarlo en la Figura 3.4, donde, el nivel con la facultad de decidir incluye elementos de color negro.

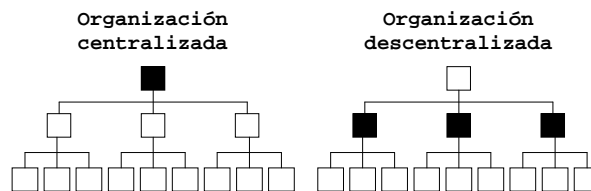


Figura 3.4: Organizaciones centralizada y descentralizada

3.2.5. Funciones del administrador

De modo general, en la actualidad se acepta que las funciones básicas del administrador son la planeación, la organización, la dirección y el control. El desempeño de estas cuatro funciones básicas, constituye el denominado *proceso administrativo*, visto como una secuencia lineal. El desempeño continuo de las funciones del administrador conforma un ciclo, al cual se le denomina *ciclo administrativo*. A medida que se repite, el ciclo administrativo permite la corrección y el ajuste continuos, mediante la retroalimentación. El desarrollo de un ciclo permite hasta cierto punto, definir cuáles serán las correcciones que habrán de introducirse en el ciclo siguiente, y así sucesivamente. De manera gráfica, la representación de esas correcciones y ajustes se muestran en la Figura 3.5.

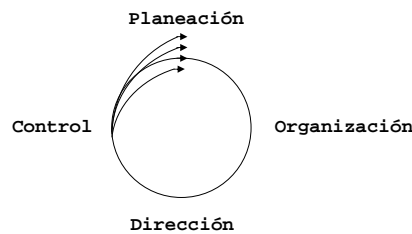


Figura 3.5: Correcciones y ajustes en el ciclo administrativo

Las funciones del administrador, forman algo más que una secuencia cíclica, pues están íntimamente relacionadas en interacción dinámica. El proceso administrativo es *cíclico*, *dinámico* e *interactivo*, como lo muestra la Figura 3.6.

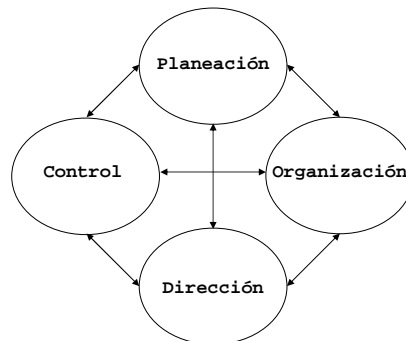


Figura 3.6: Interacción dinámica de las funciones administrativas

Planeación

Planear es *definir los objetivos y escoger anticipadamente el mejor curso de acción para alcanzarlos*. La planeación determina a dónde se pretende llegar, lo que debe hacerse, cuándo, cómo y en qué orden. Existen 3 niveles diferentes de planeación: estratégica, táctica y operacional.

1. *Planeación estratégica*: es la planeación más amplia de la organización. Sus principales características son:
 - Proyectada a largo plazo, a varios años, con sus efectos y consecuencias.
 - Cobija la empresa como una totalidad, abarca todos los recursos y áreas de actividades, y se preocupa por alcanzar los objetivos organizacionales.
 - Definida por la cúpula de la organización (en el nivel institucional), y corresponde al plan general, al cual están subordinados todos los demás.
2. *Planeación táctica*: planeación efectuada en los departamentos. Sus principales características son:
 - Proyectada a mediano plazo, generalmente para el ejercicio anual.
 - Cobija cada departamento, abarca sus recursos específicos y se preocupa por alcanzar los objetivos del departamento.
 - Se define en el nivel intermedio, en cada departamento de la empresa.
3. *Planeación operacional*: planeación de cada tarea o actividad. Sus principales características son:
 - Proyectada a corto plazo para lo inmediato.
 - Cobija cada tarea o actividad aisladamente y se preocupa por alcanzar metas específicas.
 - Está definida por cada tarea o actividad.

La planeación produce un resultado inmediato: *el plan*. Todos los planes tienen un propósito común: la previsión, la programación y la coordinación de

una secuencia lógica de eventos que, si se aplican con éxito, deberán conducir a la consecución de los objetivos que los orientan. Como un plan describe un curso de acción, debe responder a las preguntas qué, quién, cómo, dónde y por quién. Existen cuatro clases diferentes de planes, los cuales a su vez, pueden ser estratégicos, tácticos u operacionales, según el alcance que tengan.

1. *Procedimientos*. Son los planes relacionados con los *métodos de trabajo o de ejecución*. Casi siempre los objetivos son planes operacionales. Se representan por gráficas denominadas flujogramas.
2. *Presupuestos*. Son los planes relacionados con el *dinero* durante determinado periodo, ya sea por ingresos o gastos. Según sus efectos y dimensiones, los presupuestos se consideran *planes estratégicos* cuando cobijan la empresa como una totalidad y abarcan un largo periodo. Son *planes tácticos* cuando cubren determinada unidad o departamento de la empresa a mediano plazo. Son *planes operacionales* cuando su dimensión es local y a corto plazo.
3. *Programas o programaciones*. Los programas se basan en la correlación de dos variables: *tiempo y actividades* que deben ejecutarse. Los métodos de programación varían ampliamente, y van desde programas sencillos (por ejemplo, programar actividades usando una agenda) hasta programas complejos (por ejemplo, usar técnicas matemáticas avanzadas o procesamiento de datos por computadora).
4. *Reglas o reglamentos*. Son planes relacionados con el comportamiento solicitado a las personas. Especifican cómo deben comportarse las personas en determinadas situaciones. Buscan sustituir el proceso de decisión individual, restringiendo de antemano, el grado de libertad de las personas en situaciones determinadas.

Organización

La palabra organización puede adoptar varios significados. En esta tesis, la utilizamos de dos maneras.

1. *Organización social, orientada a alcanzar objetivos específicos y estructurada deliberadamente*. La organización es una entidad social porque la conforman personas; está orientada a objetivos porque se halla diseñada

para conseguir resultados, generar utilidades (empresas en general), proporcionar satisfacción social (clubes), etc. y estructurada deliberadamente porque propone la división de trabajo y asigna su ejecución a los miembros. Esta definición cubre todos los tipos de organizaciones, tengan o no ánimo de lucro. Desde este punto de vista, la organización presenta dos aspectos diferentes:

- *Organización formal.* Es la organización planeada o la que está definida en el organigrama.
 - *Organización informal.* Surge a partir de las relaciones humanas establecidas al desempeñarse en los cargos. Se configura a través de las relaciones de amistad (o de antagonismo) y del surgimiento de grupos informales que no aparecen en el organigrama u otro documento formal.
2. *Organización como parte administrativa y parte integrante del proceso administrativo.* En este sentido, la organización significa el acto de organizar, estructurar e integrar los recursos y los órganos involucrados en la ejecución y establecer las relaciones entre ellos y las atribuciones de cada uno.

En esta sección nos referimos a la organización como la *segunda función administrativa*, que depende de la planeación, la dirección y el control para formar el proceso administrativo. Para alcanzar los objetivos, ejecutar los planes y lograr que las personas trabajen con eficiencia, las actividades deben agruparse de manera lógica y se debe distribuir la autoridad para evitar conflictos y confusiones. La organización consiste en:

- Determinar las actividades específicas necesarias para el logro de los objetivos planeados (*especialización*).
- Agrupar las actividades en una estructura lógica (*departamentalización*).
- Asignar las actividades a posiciones y personas específicas (*cargos y tareas*).

Dirección

La dirección, tercera función administrativa, sigue a la planeación y a la organización. La función de dirección se relaciona directamente con la manera de *orientar la actividad de las personas* que componen la organización para alcanzar el objetivo o los objetivos. La dirección es la función administrativa que se refiere a las relaciones interpersonales de los administradores y sus respectivos subordinados en todos los niveles de la organización. Para que la planeación y la dirección puedan ser eficaces, necesitan ser dinamizadas y complementadas por la orientación que se dé a las personas mediante una adecuada comunicación y habilidad de *liderazgo y motivación*.

Control

La palabra control tiene varios significados. Los tres principales son:

1. *Control como función restrictiva y coercitiva.* Es utilizado con el fin de cohibir o limitar ciertos tipos de desvíos indeseables o de comportamientos no aceptados.
2. *Control como sistema automático de regulación.* Se utiliza con el fin de mantener un sistema en funcionamiento. El mecanismo de control detecta posibles desviaciones o irregularidades e introduce de modo automático las regulaciones necesarias para volver a la normalidad.
3. *Control como función administrativa.* El control forma parte del proceso administrativo, del mismo modo que la planeación, la organización y la dirección.

Este último significado es al que daremos explicación. La finalidad del control es asegurar que los resultados de aquello que se planeó, organizó y dirigió, se ajusten tanto como sea posible a los objetivos previamente establecidos. La esencia del control reside en *comprobar si la actividad controlada está alcanzando o no los objetivos o los resultados deseados*. El control es un proceso cíclico compuesto de cuatro fases:

1. *Establecimiento de los estándares o criterios:* Los estándares representan el desempeño deseado. Los criterios representan las normas que guían las decisiones. Ambos representan las disposiciones que proporcionan los medios para establecer que deberá hacerse y cuál es el desempeño o resultado que se aceptará como normal o deseable.

2. *Supervisión del desempeño*: El proceso de control permite ajustar las operaciones a determinados estándares previamente establecidos, y funciona basado en la información que recibe. La información o verificación del desempeño o del resultado busca obtener información precisa sobre la operación que se está controlando.
3. *Comparación del desempeño con el estándar establecido*: Toda actividad experimenta alguna variación, error o desviación. Es importante determinar los límites en esa variación, para poder aceptarla como normal o deseable. No toda variación exige correcciones, sólo aquellas que sobrepasan los límites normales.
4. *Acción correctiva*: El objetivo del control es mantener las operaciones dentro de los estándares establecidos para conseguir los objetivos de la mejor manera. La acción correctiva busca lograr que lo realizado esté de acuerdo con lo que se pretendía realizar.

3.3. Tipos de Organizaciones

No existen dos empresas idénticas aunque existen ciertos principios y características básicas empleadas como directrices en el estudio de la organización empresarial. Todas las empresas poseen una organización propia específica e individual. La mejor manera de comprender la organización formal es comparar los tres modelos tradicionales de organización ampliamente divulgados por los autores neoclásicos. Para nuestros efectos, podemos distinguir tres modelos tradicionales de organización: *organización lineal*, *organización funcional* y *organización línea-staff*.

3.3.1. Organización lineal

La denominación lineal obedece al hecho de que entre el superior y los subordinados existen *líneas directas y únicas de autoridad y de responsabilidad*. En esta organización sencilla piramidal cada jefe recibe y transmite todo lo que sucede en su área, puesto que las líneas de comunicación se establecen con rigidez.

Características de la Organización Lineal

La organización lineal presenta cuatro características:

1. *Autoridad lineal o única.* Cada subordinado se reporta única y exclusivamente ante su superior y tiene un sólo jefe, del cual recibe órdenes y ante quien se reporta.
2. *Líneas formales de comunicación.* Las comunicaciones entre los órganos o cargos existentes en la organización lineal se efectúan únicamente a través de las líneas presentes en el organigrama.
3. *Centralización de las decisiones.* La autoridad lineal que dirige la organización se centraliza en la cima del organigrama y los canales de responsabilidad convergen hacia allí a través de los niveles jerárquicos.
4. *Configuración piramidal.* La organización lineal presenta una conformación típicamente piramidal. A medida que se asciende la escala jerárquica, disminuye el número de cargos u órganos.

Campo de aplicación de la organización lineal

La organización lineal es aplicable específicamente en los siguientes casos:

- Cuando la organización es pequeña y no requiere ejecutivos especialistas en tareas altamente técnicas.
- Cuando la organización está comenzando su desarrollo.
- Cuando las tareas llevadas a cabo por la organización *están estandarizadas, son rutinarias y tienen pocas modificaciones.*
- Cuando la organización tiene vida corta y la rapidez en la ejecución del trabajo se hace más importante que la calidad del mismo.
- Cuando la organización juzga más interesante invertir en consultoría externa u obtener servicios externos, que establecer órganos internos de asesoría.

3.3.2. Organización funcional

La organización funcional es el tipo de estructura organizacional que aplica el principio funcional o principio de la especialización de las funciones para cada tarea.

Características de la organización funcional

La organización funcional presenta cuatro características:

1. *Autoridad funcional o dividida.* En la organización funcional, la autoridad es relativa y se basa en la especialización. Cada subordinado reporta ante muchos superiores de manera simultánea, pero sólo en los asuntos concernientes a la especialidad de ellos.
2. *Líneas directas de comunicación.* La comunicación entre los órganos o cargos de la organización es directa, sin intermediarios.
3. *Descentralización de las decisiones.* Las decisiones se delegan a los órganos o cargos especializados que posean el conocimiento necesario para implementarlas mejor.
4. *Énfasis en la especialización.* La organización se basa en la especialización de todos los órganos o cargos en todos los niveles de la organización. Cada órgano o cargo contribuye con su especialidad a la organización.

Campo de aplicación de la organización funcional

Este tipo de gerencia múltiple (*multiple management*), creado por los ingenieros estadounidenses originó algunos problemas, en especial de coordinación y de control, pues ni el entrenamiento, ni el sentido común, consiguen evitar la confusión que se presenta cuando hay líneas de autoridad divididas. La aplicación de la organización funcional debe restringirse a los siguientes casos:

- Cuando la organización por ser pequeña, tiene un equipo de especialistas bien compenetrado, que reporta ante un dirigente eficaz y está orientado hacia objetivos comunes muy bien establecidos y definidos.

- Cuando en determinadas circunstancias, la organización delega durante un periodo determinado la autoridad funcional a un órgano especializado, sobre los demás órganos, para implantar una rutina o procedimiento, o evaluar y controlar alguna actividad.

Las principales diferencias entre la organización funcional y la lineal se muestran en la Figura 3.7. Del lado izquierdo podemos observar la estructura jerárquica que las representa y del lado derecho sus principales características.

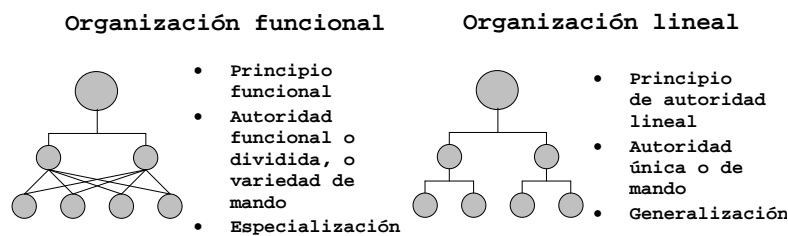


Figura 3.7: Diferencias entre organización lineal y organización funcional

3.3.3. Organización de tipo línea-staff

La organización del tipo línea-staff comparte características lineales y funcionales, que al combinarse crean un tipo de organización más completa y compleja. En la organización línea-staff existen órganos de línea (*órganos de ejecución*) y de asesoría (*órganos de apoyo y consultoría*) que mantienen relaciones entre sí. Los órganos de línea se caracterizan por la autoridad lineal y el principio jerárquico mientras que los órganos de staff prestan asesorías y servicios especializados.

Características de la organización línea-staff

La organización línea-staff posee características mucho más complejas que los otros tipos de organización, entre las cuales podemos señalar:

1. *Fusión de la estructura lineal con la estructura funcional, con predominio de la primera.* Cada órgano responde ante un sólo y único órgano superior. Cada órgano recibe asesoría y servicios especializados de diversos órganos de staff.

2. *Un órgano de staff* no puede obligar a los demás órganos a que acepten sus servicios y recomendaciones, por cuanto no tiene autoridad de línea, es decir, *sólo da asesoría y presta servicios especializados*.

Todos los órganos sean de línea o de staff tienen autoridad de línea sobre sus subordinados, y autoridad funcional en relación con los demás órganos de la empresa, en su especialidad.

Campo de aplicación de la organización línea-staff

Si consideramos que los especialistas pueden incorporarse a la organización cuando sus servicios sean necesarios y su contratación corresponde a un costo razonable, es evidente que el staff se convierte en un hecho muy deseado, siempre y cuando estos especialistas no interfieran las líneas de autoridad de los órganos a los cuales prestarán asesoría.

3.4. Teoría de las Relaciones Humanas

La teoría de las relaciones humanas critica con dureza los antiguos conceptos de autoridad, jerarquía, principios generales de la administración, etc. El ingeniero y el técnico ceden su lugar al psicólogo y al sociólogo. El énfasis en las tareas y la estructura es sustituido por el *énfasis en las personas*.

3.4.1. El hombre social

La teoría de las relaciones humanas inicia una nueva concepción de la naturaleza del hombre en su desempeño dentro de una organización: *el hombre social*.

1. Los trabajadores son criaturas sociales complejas que tienen sentimientos, deseos y temores. Su *comportamiento* en el trabajo, como en cualquier lugar, es consecuencia de muchos *factores motivacionales*.
2. Las personas se ven motivadas por ciertas necesidades y logran sus satisfacciones primarias con la ayuda de los grupos con los que interactúan.

3. El comportamiento de los grupos puede manejarse mediante un estilo adecuado de *supervisión* y liderazgo. El supervisor eficaz posee capacidad para influir en sus subordinados, obtiene lealtad de éstos, consigue estándares elevados de desempeño y un gran *compromiso* con los objetivos de la organización.
4. Las *normas* de una organización sirven de mecanismos reguladores del comportamiento de sus miembros y controlan de modo informal los niveles de producción. Este modo de control social puede adoptar *sanciones* positivas (estímulos, aceptación social, etc.) o negativas (burlas, rechazo por parte del grupo, sanciones simbólicas etc.).

3.4.2. Las necesidades humanas básicas

Para comprender que motiva el comportamiento, es necesario el conocimiento de las necesidades humanas. El comportamiento humano está determinado por causas que escapan al propio entendimiento y control del ser humano. Dichas causas se denominan *necesidades o motivos*, las cuales son fuerzas conscientes o inconscientes que determinan el comportamiento del individuo. La *motivación* se refiere al comportamiento causado por las necesidades internas del individuo, el cual *se orienta a conseguir objetivos* que pueden satisfacer tales necesidades.

3.4.3. Comunicación

La comunicación es una actividad administrativa que cumple dos propósitos esenciales:

1. Proporcionar la información y la explicación necesarias para que las personas puedan desempeñar sus tareas,
2. Proporcionar las actitudes necesarias que promuevan la motivación, la cooperación y la satisfacción en los cargos.

En conjunto estos dos propósitos sirven para promover un ambiente que conduce a crear sentido de equipo y lograr mejor desempeño en las tareas. En la teoría de las relaciones humanas, la comunicación es importante para la relación entre las personas y para explicar con claridad, a los subalternos, las razones de las decisiones tomadas. Éstos deben recibir de los superiores

un flujo de comunicación capaz de satisfacerles las necesidades. A su vez, los superiores deben recibir de aquellos un flujo de comunicación capaz de darles una idea precisa de lo que está sucediendo. Podemos reflejar los objetivos de una buena comunicación mediante la Figura 3.8.

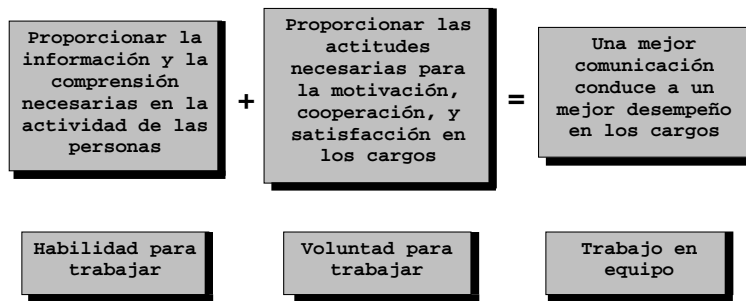


Figura 3.8: Principales propósitos de la comunicación

Dentro de este contexto surge la necesidad de que el administrador revise con periodicidad el trabajo de sus subordinados, para evaluar su desempeño y habilidades. Esa necesidad de evaluación se fundamenta en tres aspectos:

1. El hombre trabaja mejor cuando conoce los *estándares* de su trabajo.
2. La organización opera con más eficiencia cuando el obrero y el jefe saben cuáles son las *responsabilidades* y los estándares de desempeño que la empresa espera de ellos.
3. Cada obrero puede ser ayudado para que dé la máxima contribución a la empresa y utilice el máximo de sus capacidades y habilidades.

Capítulo 4

Organizaciones y Agentes

Este capítulo presenta una descripción de las estructuras que componen una organización desde el punto de vista de varios autores, además de un análisis comparativo de las estructuras propuestas por cada uno. Dicho análisis nos proporcionará los aspectos más importantes a considerar de una organización jerárquica, los cuales, nos servirán para crear el modelo de una organización formada por agentes normativos. La organización de este capítulo es la siguiente: en la sección 4.1 se presenta la investigación hecha por Carley y Gasser [2] sobre teoría organizacional, la sección 4.2 describe la analogía entre los sistemas distribuidos y las organizaciones humanas hecha por Fox [9], la sección 4.3 presenta el análisis hecho por Malone [17] acerca de las estructuras de coordinación de una organización. En la sección 4.4 se hace un análisis comparativo de los tres trabajos antes mencionados.

4.1. Teoría Organizacional para Sistemas de Cómputo

El primer trabajo que se describe es el de Carley y Gasser [2], donde se analiza a los sistemas de cómputo a partir de la perspectiva de la teoría organizacional. Según los autores, las organizaciones poseen las siguientes características:

- Resuelven problemas de gran escala.
- Están formadas por varios agentes.

- Se ocupan de una o más tareas.
- Alcanzan metas.
- Son capaces de afectar su ambiente y viceversa.
- Tienen un conocimiento, cultura, memoria, historia y capacidades distintas para cada agente.

Las organizaciones existen para superar las capacidades individuales de los agentes. Décadas de investigación en esta área [2, 9, 17] han mostrado repetidamente que no hay un diseño individual apropiado o correcto de una organización. La decisión de qué modelo organizacional es óptimo, depende de una variedad de factores que incluyen:

- La tarea o tareas específicas a realizar.
- La inteligencia, capacidades cognitivas, o entrenamiento de los agentes.
- La volatilidad del ambiente.
- Restricciones legales o políticas en el diseño de la organización.
- El tipo de salida deseado (por ejemplo, eficiencia, efectividad, exactitud o costos mínimos).

La teoría organizacional se ha enfocado en la búsqueda de principios organizacionales y las condiciones bajo las cuales tales principios aplican o no. Por ejemplo, dos principios relacionados son la especialización y la división de labor.

- La *especialización* es el principio que dice que los agentes se pueden volver más efectivos cuando son expertos en actividades particulares que requieren tipos de conocimiento particulares y limitados.
- La *división de labor* es el principio que dicta que una división apropiada de tareas, conocimiento y habilidades entre agentes en una organización puede mejorar el rendimiento organizacional; por ejemplo, limitando las dependencias del conocimiento y las tareas.

En general, las organizaciones que emplean instancias productivas de estos dos principios, son capaces de superar las limitaciones individuales de los agentes.

4.1.1. Elementos de una organización

Carley y Gasser afirman que a fin de representar una organización como un sistema computacional los siguientes factores deben modelarse a cierto nivel de detalle:

- Los *agentes* que comprenden la organización. Los agentes tienen un nivel de capacidad cognitiva y ocupan una posición en una organización.
- La *estructura* o diseño de la organización. Incluye los procedimientos y reglas que conforman el conocimiento de la organización y el conjunto de conexiones entre agentes y tareas.
- Las *tareas* que la organización realiza. Comprende aspectos como la dependencia entre tareas. Existen 3 tipos de dependencia: la secuencial (donde el resultado de una tarea es la entrada de otra), la recíproca (donde los resultados dependen uno del otro) y la conjunta (donde se da la unión de resultados)
- La *tecnología* para la transformación del material y/o la información.
- Alguna *restricción* para la organización (volumen de ventas, fechas límite, cambios en la legislación, etc.).

Una de las mayores cuestiones en el área de organización computacional es determinar que diseño organizacional tiene sentido y cuales son los costos y beneficios de tal diseño.

4.1.2. Estructuras organizacionales

Puede haber muchas de tales estructuras en una organización. La estructura más familiar es la estructura de comunicación y autoridad. Éstas están frecuentemente representadas en términos específicos por un organigrama, el cual, es la representación de la estructura formal. A continuación se describirán, los tres tipos de estructuras que comprenden el diseño en general de una organización.

- La *estructura de acceso a recursos, habilidades, o decisiones*, es la que liga a los agentes con sus recursos, las posibles decisiones elegibles, y sus habilidades. Tales estructuras muestran lo que cada agente tiene a su disposición.

1. Una *estructura segregada o especializada* implica que cada agente es único. En este tipo de estructura no existen dos agentes que usen los mismos recursos o que tengan las mismas habilidades. (Fig. 4.1a).
2. Una *estructura jerárquica* a este nivel implica que hay un agente que tiene acceso general o, conocimiento sobre, los recursos, habilidades o las decisiones de todos los agentes (Fig. 4.1b).

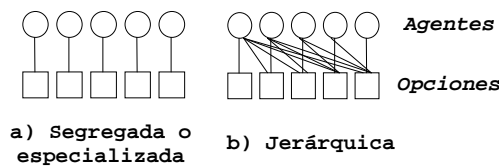


Figura 4.1: Estructuras de recursos, habilidades, decisiones

- *Estructura de autoridad-comunicación.* En la estructura de autoridad los enlaces muestran quién tiene autoridad sobre quién y quién se reporta con quién. En la estructura de comunicación los enlaces muestran quién habla con quién.
 1. En una *estructura colaborativa* todos los enlaces son posibles, es decir, todos los agentes se comunican entre sí y nadie posee autoridad sobre nadie (Fig.4.2a).
 2. En una *jerarquía* hay un agente central o en la cima. Dicho agente funge como jefe y los agentes del nivel inferior se comunican únicamente con él (Fig. 4.2b).

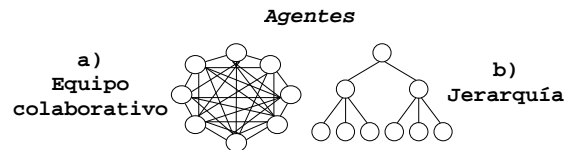


Figura 4.2: Estructuras de autoridad, comunicación

- *Estructura de requerimientos.* Liga los recursos, o habilidades con las tareas que los necesitan, es decir, esta estructura muestra que se necesita para alcanzar una meta por medio de una tarea.

1. En la *estructura segregada o especializada* cada tarea posee requerimientos diferentes (Fig. 4.3a).
2. En la *estructura sin segmentos* cada tarea requiere todos los recursos o habilidades (Fig. 4.3b).

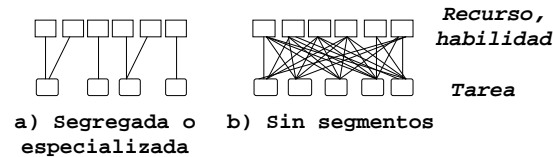


Figura 4.3: Estructuras de requerimientos

4.1.3. Dinámica

Las organizaciones son potencialmente dinámicas de muchas formas. Las organizaciones son capaces de ser rediseñadas, o entrar en un proceso de reingeniería, haciendo cambios tales como, qué agentes se reportan con cuales otros agentes, o qué agentes manejan cuales tareas. Existen varios enfoques para modelar la dinámica organizacional, pero sólo mencionaremos dos.

- El primero se basa en la idea de agentes flexibles (agentes que pueden reestructurarse en respuesta a los cambios en su ambiente).
- El segundo enfoque está basado en la idea de un aprendizaje a nivel dual (organizaciones en los que el aprendizaje a nivel agente y a nivel estructural ocurren como respuesta a cambios en el ambiente)

4.2. Sistemas Distribuidos como Organizaciones

El trabajo de Fox [9], intenta construir parte de la teoría de diseño de sistemas distribuidos basado en el diseño organizacional, describiendo una analogía entre los sistemas distribuidos y las organizaciones humanas.

4.2.1. Estructura organizacional

Una organización se define como la composición de una estructura y un régimen de control. El conjunto de posibles estructuras organizacionales va desde las jerarquías estrictas hasta las jerarquías no uniformes.

- Una *jerarquía simple* tiene dos niveles. Encontramos un sólo tomador de decisiones en el nivel superior que coordina los esfuerzos de las personas en el nivel inferior. La información completa sólo debe estar disponible a esa persona, y debe tener la autoridad de efectuar cambios en el comportamiento de la organización. Una coordinación apropiada, implica relaciones de empleo y autoridad, y de distribución de información, lo anterior es necesario para que la organización tenga un desempeño efectivo (Fig. 4.4a).
- Conforme el tamaño de la organización de un sólo producto aumenta, las jerarquías simples suelen fallar debido a la racionalidad limitada. El tomador de decisiones es incapaz de procesar toda la información (se satura). Lo anterior hace necesaria la transición de una jerarquía simple a lo que se conoce como una *jerarquía uniforme de varios niveles*. En ésta, se crean varios niveles de manejo para asegurar que la toma de decisiones centralizada sea apropiada. Cada nivel de la jerarquía actúa como un filtro de información y las decisiones se propagan hacia las partes superiores de la jerarquía. Las decisiones se toman en el nivel más inferior de la jerarquía, en donde se tiene la información necesaria para tomar la mejor decisión y la autoridad para ejecutarla (Fig. 4.4b).
- Conforme la jerarquía uniforme aumenta, en el tamaño y el número de productos, los procesos de control se deterioran. Debido al aumento de productos, se produce una competencia entre unidades por los mismos recursos. Asignar recursos de modo que se encuentren disponibles para poder procesar los productos en tiempo y forma, es una tarea bastante compleja. Un enfoque que reduce este efecto es la *jerarquía multidivisional*. La organización se divide en líneas de producción. Cada división tiene el control total sobre las tácticas concernientes al desarrollo de su producto. Por lo tanto, el control está situado localmente donde la información que permite el control esté disponible. El control estratégico se concede a un grupo (staff) asignado a una oficina general. La oficina general se ocupa de la planeación estratégica, valoración y control de la organización, incluyendo la asignación de recursos (Fig. 4.4c).

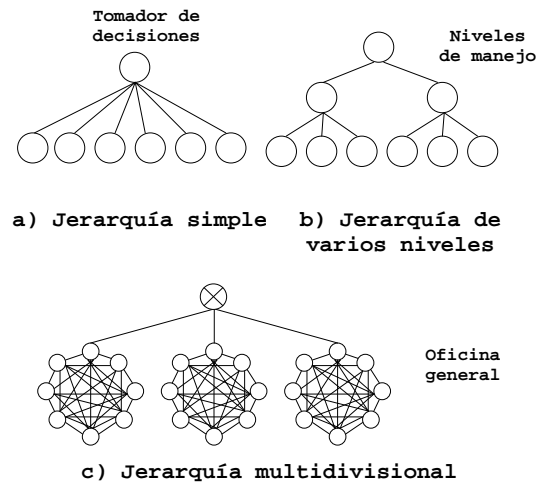


Figura 4.4: Tipos de jerarquías

4.2.2. Seleccionando una organización

La elección de que tipo de organización se ajusta mejor a que tarea, es un proceso bastante complejo. No existe una teoría analítica que nos diga como seleccionar una organización. Pero existe una aproximación descriptiva, la cual contiene tres partes:

1. Una descripción de las características de una tarea, por ejemplo, su complejidad e incertidumbre.
2. Un conjunto de estructuras organizacionales y regímenes de control que pueden hacer frente a esas características.
3. Una forma de medir el desempeño de la organización. La meta de la tarea proporciona un buen criterio de medición.

4.3. Estructuras de Coordinación para Organizaciones y Mercados

Malone [17] describe un conjunto de estructuras simples de coordinación, que modelan los tipos de procesamiento de información en organizaciones y mercados. En este trabajo nos centraremos en analizar únicamente a las

organizaciones. Esta perspectiva parece particularmente prometedora para comprender como la información puede afectar la estructura organizacional, debido a que los mecanismos de coordinación son críticos al determinar la estructura organizacional.

El enfoque de Malone ve a cada actividad como una tarea que debe realizarse en algún procesador y su realización requiere de una cantidad de tiempo. Este punto de vista, subraya la importancia de asignar las tareas a los procesadores como uno de los componentes fundamentales de la coordinación y subraya el tiempo de retraso y la capacidad de procesamiento como un componente importante del costo total de la salida.

4.3.1. Modelo

Malone [17] define una estructura de coordinación como un patrón de toma de decisiones y de comunicación entre un grupo de actores, quienes realizan una tarea a fin alcanzar una meta.

Su trabajo entonces es analizar los tipos de costos para cada estructura de coordinación: costos de producción, costos de coordinación, y costos de vulnerabilidad. Los *costos de producción* incluyen la capacidad de producción y los costos de los retrasos al realizar las tareas. En los *costos de coordinación* se incluyen los costos de mantener los enlaces de comunicación (o canales) entre actores y el costo de intercambiar “mensajes” sobre esos enlaces. Los *costos de vulnerabilidad* se refieren a los costos inevitables que surgen cuando ocurre un cambio de situación antes que la organización pueda adaptarse a la nueva situación. Se modelan esos costos en términos de los costos esperados, que resultan, cuando un actor falla al realizar su tarea.

4.3.2. Estructuras alternativas

Malone considera dos estructuras de coordinación: las jerarquías de productos y las jerarquías funcionales.

- *Jerarquía de producto* (Fig. 4.5a). En una jerarquía de producto, existen divisiones separadas para cada línea de producción diferente. Cada división tiene un jefe (llamado manejador de producto) y tiene diferentes departamentos para diferentes funciones.

Esta forma se representa mediante la siguiente estructura de coordinación: Cada división consta de un manejador de producto y un proce-

sador especializado (por ejemplo, un departamento) para cada tarea distinta. El manejador de producto decide que tareas deben hacerse para realizar un producto (o para alcanzar una meta, viéndolo de forma generalizada). A cada tarea que llega, el manejador del producto le asigna el procesador especializado que realiza esa tarea.

Se asume que es necesario enviar un “mensaje” al procesador que se le asigna una tarea, esto significa que debe haber líneas de comunicación entre cada uno de los manejadores de productos y sus propios procesadores, pero no con las otras divisiones. Malone asume que cuando un procesador falla, la división del producto en la cual ocurre la falla es interrumpida, pero esto no afecta las otras divisiones.

Dado que la estructura de “jerarquía de producto” no incluye interacciones entre las divisiones, puede utilizarse para modelar una compañía completa, o de hecho un conjunto de compañías que no comparten recurso alguno.

- *Jerarquía funcional* (Fig. 4.5b). En esta clase de jerarquía, una cantidad de procesadores de tipo similar se distribuyen en departamentos funcionales compartidos entre productos. Los procesadores compartidos pueden reducir la duplicación de errores y balancear la carga de procesos sobre todos los productos.

En la estructura de coordinación que se usa para representar esta forma organizacional, los procesadores del mismo tipo se agrupan en departamentos funcionales, cada uno tiene un jefe (“manejador funcional”). También existe una “oficina ejecutiva”, la cual, decide que tareas deben hacerse para producir todos los productos de la organización (dicho de manera más general, para alcanzar la meta global de la estructura). La oficina ejecutiva juega entonces, el rol del “manejador del producto” para todos los productos. Asignar tareas en esta estructura es más complicado que en la jerarquía de producto porque involucra una capa de manejo más. Se asume que siempre que una tarea de cierto tipo debe hacerse, la oficina ejecutiva delega la tarea al manejador funcional del tipo apropiado, el cual a su vez, asigna la tarea a uno de los procesadores en el departamento funcional del manejador. A fin de realizar esta asignación de manera inteligente, el manejador funcional debe mantener un seguimiento de la carga y las capacidades de los procesadores de su departamento.

En la definición de esta estructura de coordinación, se hace la suposición de que se necesitan 4 mensajes para asignar una tarea: uno para delegar la tarea al manejador funcional apropiado, uno para asignarla al procesador, y dos para notificar que la tarea está completa; uno para el manejador funcional y uno para la oficina ejecutiva. A fin de intercambiar estos mensajes la oficina ejecutiva debe tener enlaces de comunicación con cada manejador funcional, y éste a su vez debe tener un enlace con los procesadores en sus departamentos. Cuando un procesador falla, la tarea que se va a realizar se retrasa hasta que sea reasignada a otro procesador en el mismo departamento. Cuando un manejador funcional o la oficina ejecutiva fallan, los procesos de la organización completa fallan.

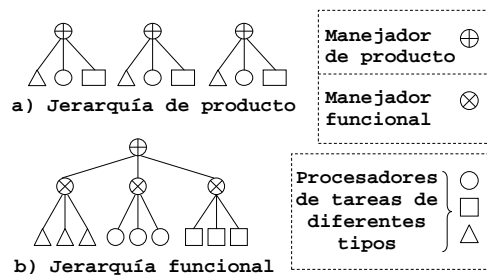


Figura 4.5: Estructuras de coordinación

4.4. Análisis

Nuestro análisis se basa en cuadros comparativos, donde la primera columna representa a los autores y la segunda nos proporciona los elementos a comparar.

Cada autor inicia definiendo la palabra organización, Carley y Gasser concluyen que no existe una sola definición para ésta, pero que podemos definirla mediante un conjunto de características generales, las cuales se resumen en el Cuadro 4.1. Fox define una organización como la composición de una estructura y un régimen de control, mientras que Malone no define el término, pero en su trabajo analiza a los miembros de la organización, las tareas y las estructuras que la conforman.

Autor	Características
[2]	Tareas, agentes (capacidades y roles), metas, estructuras.
[9]	Estructura y régimen de control.
[17]	Actores, tareas y estructuras.

Cuadro 4.1: Características de una organización

Los tres autores incluyen en su análisis 3 elementos organizacionales: la estructura de la organización, sus miembros y las tareas que realizan. En [2] se define a los miembros de la organización como agentes, además se hace especial énfasis en las capacidades y los roles de los agentes, en el sentido de que un buen desempeño individual puede lograr un mejor desempeño organizacional. Para Fox los miembros de la organización son un conjunto de tomadores de decisiones, a su vez, Malone los llama actores, pero, un actor no es más que un procesador. Todos los autores coinciden en incluir a las tareas y las diferentes estructuras (de comunicación, autoridad, recursos, etc.) como partes de una organización. En [2] y [9], se incluye el concepto de meta organizacional, el cual es un parámetro que nos sirve para medir el buen o mal desempeño de la organización, esto depende de si se cumple o no la meta, mientras que en [17], el desempeño se mide en términos de los costos de producción y de coordinación. Lo anterior se resume en el Cuadro 4.2.

Autor	Componentes
[2]	Agentes (capacidades y roles), tareas, metas, estructuras.
[9]	Tomadores de decisiones, tareas, meta organizacional, estructura y régimen de control.
[17]	Actores (procesadores), tareas y estructuras.

Cuadro 4.2: Componentes de una organización

Cada uno de los autores define sus propias estructuras organizacionales las cuales se describen en el Cuadro 4.3. Las estructuras descritas por Malone y Fox son equivalentes a las estructuras de autoridad-comunicación definidas por Carley y Gasser. La estructura de jerarquía funcional definida por Malone coincide con la estructura multidivisional definida en el trabajo hecho por Fox, aunque Fox no define una estructura de comunicación-autoridad jerárquica, y sólo menciona que se trata de un grupo. Como podemos obser-

var Carley y Gasser agregan a sus modelos estructuras de recursos (las cuales muestran los recursos disponibles para cada agente) y de requerimientos (las cuales asocian a los recursos con las tareas que los necesitan).

Autor	Estructuras
[2]	Recursos, autoridad, requerimientos.
[9]	Jerarquía simple, de varios niveles y multidivisional.
[17]	Jerarquía funcional y jerarquía de producto.

Cuadro 4.3: Tipos de estructuras

Por último, la decisión sobre que modelo organizacional es óptimo depende de varios factores, los cuales, se resumen en el Cuadro 4.4. El desempeño de una organización se ve limitado por un conjunto de restricciones, Malone toma en cuenta los costos de coordinación y de producción, Fox menciona la complejidad de las tareas y la incertidumbre, Carley y Gasser hablan de dependencia entre tareas.

Autor	Factores
[2]	Tareas, ambiente, inteligencia de los agentes, restricciones.
[9]	Tareas, estructuras, restricciones, meta organizacional.
[17]	Costos de coordinación y de producción.

Cuadro 4.4: Factores para un modelo óptimo

Capítulo 5

El Modelo

Este capítulo presenta nuestro modelo para organizaciones jerárquicas, el cual toma como base un sistema multiagentes normativo. Nuestro modelo utiliza varios conceptos de diferentes teorías sociales [3, 4, 5], los cuales se explicaron en el Capítulo 3. Estas teorías describen las características, los elementos y los procesos que se llevan a cabo dentro de una organización humana.

La organización de este Capítulo es la siguiente. En la Sección 5.1 modelamos un sistema multiagentes normativo. En la Sección 5.2 explicamos la definición de organización utilizada en esta tesis. En la Sección 5.3 enumeramos los componentes del modelo, posteriormente presentamos su definición en base a la teoría de organizaciones humanas y por último, explicamos la definición y la representación que le hemos dado en nuestro modelo utilizando diagramas de clase.

5.1. Sistema Multiagentes Normativo

De acuerdo con López y Luck [13], un Sistema Multiagentes Normativo (SMAN) puede definirse como un conjunto de agentes normativos, los cuales son controlados por medio de un conjunto de normas comunes, las cuales van desde las obligaciones y compromisos sociales, hasta los códigos sociales. Este control puede observarse desde tres perspectivas. Primero, los agentes miembros deben reconocerse a sí mismos como parte de una sociedad. Segundo, el control no puede ser ejercido si no se aplican sanciones positivas o negativas cuando se cumple o se infringe una norma. Tercero, se deben permitir cam-

bios en la normatividad actual como un medio para resolver conflictos entre normas y/o agentes. Un SMAN debe incluir los siguientes elementos [13].

- Un conjunto de agentes *miembros* capaces de razonar acerca de las normas.
- Un conjunto de agentes *legisladores*.
- Un conjunto de agentes *defensores*.
- Un conjunto de *normas* dirigidas a regular el comportamiento de esos agentes.
- Un conjunto de normas cuya finalidad es *fortalecer* y determinar el cumplimiento del más reciente conjunto de normas.
- Un conjunto de normas dirigidas a *promover* el cumplimiento de las normas a través de recompensas.
- Un conjunto de normas emitidas para permitir la *creación y abolición* de normas.

Dado que éste es nuestro modelo base, lo restante de esta sección está dedicado a representar cada uno de sus componentes¹ usando el Lenguaje Unificado de Modelado.

5.1.1. Conjunto

Tanto el modelo de SMAN [13], como nuestro modelo de organización jerárquica utilizan conceptos de la teoría de conjuntos y de la lógica de primer orden. Es por esta razón que consideramos necesario modelar la clase que representa un conjunto de elementos. La Figura 5.1 muestra la clase que utilizamos para representar un conjunto. Esta clase implementa las operaciones básicas sobre conjuntos. Los conjuntos en nuestro modelo son listas

¹Cabe mencionar que, en lo restante de este capítulo, en algunas ocasiones, por motivos de espacio, se suprimen los métodos que nos permiten modificar o recuperar la información de los atributos de algunas clases. Nos centraremos en explicar únicamente los métodos que describen mejor la funcionalidad de una clase y su rol dentro del modelo. Se hace la aclaración ya que en el Capítulo 6 se utilizan métodos que se suprimen en los diagramas de éste capítulo.

ligadas formadas por nodos, como lo muestra la Figura 5.1. La notación que usaremos en lo restante de la tesis es la siguiente. Las clases que representan conjuntos llevan nombres del tipo: `ConjNombreClase`, cuyos elementos son nodos con nombres del tipo `NodoNombreClase`, los cuales contienen información del tipo `NombreClase`. La Figura 5.1 muestra la relación entre clases que indica que un conjunto contiene de cero o más nodos.

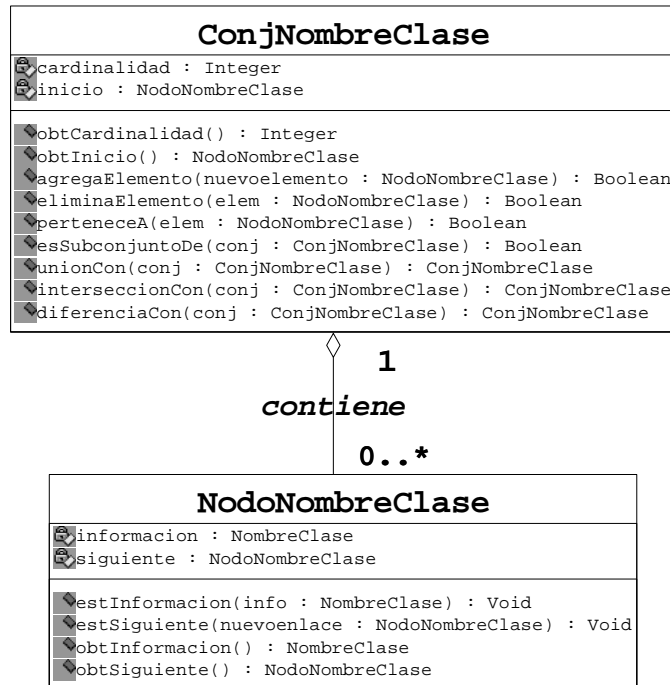


Figura 5.1: Clase Conjunto

5.1.2. Atributo

En el modelo de López y Luck [13], un *atributo* representa una característica perceptible del ambiente del agente, lo cual a su vez se representa por medio de un predicado o su negación. Por ejemplo, podemos decir *el carro es rojo* o *el libro está en la mesa*. Por lo tanto, un predicado está representado mediante la clase `Predicado`, la cual consta de un símbolo de predicado, el número de argumentos y sus valores. Para ilustrar la funcionalidad de esta clase, supongamos que deseamos representar el predicado *enLaMesa(libro)*,

el símbolo de predicado es “enLaMesa”, el cual sólo tiene un argumento, cuyo valor es la constante “libro”. Por otro lado, la clase **Atributo** consta de dos elementos: 1) la situación que modela, la cual se representa como un predicado, y 2) una variable booleana. La variable **signo** nos permite representar la negación de un predicado, cuando su valor es **false**. Además cuenta con un método, el cual evalúa el predicado. En la Figura 5.2 se muestra la relación entre clases que indica que un atributo consiste de un predicado o su negación.

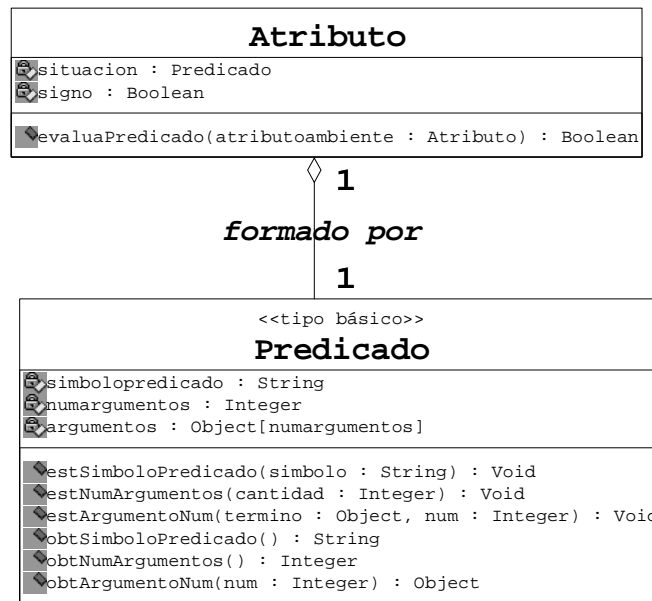


Figura 5.2: Clase Atributo

5.1.3. Estado del ambiente

Un *estado del ambiente* se representa por medio de un conjunto de atributos. Podemos observar en la Figura 5.3 el diagrama de clase que modela el *estado del ambiente* en el que se encuentra un agente. Éste consiste de un conjunto no vacío de atributos y dos métodos, uno para hacer cambios en el ambiente y otro para obtener información de él.

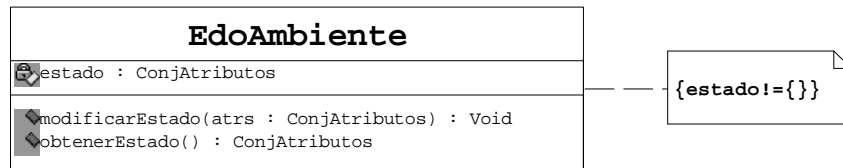


Figura 5.3: Clase EdoAmbiente

5.1.4. Meta

Una *meta* representa alguna situación a la que un agente quiere llegar. Las *motivaciones* son los deseos o preferencias que afectan el razonamiento previsto para satisfacer la meta del agente. Nosotros representamos una motivación como un número entero que asigna un peso a cada meta del agente, como lo muestra la Figura 5.4. La clase *Meta* utiliza dos métodos. El método *establecerPreferencia* asigna un valor a la motivación del agente y el método *obtenerPreferencia* devuelve la prioridad de la meta.

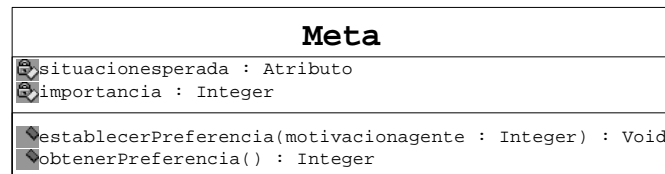


Figura 5.4: Clase Meta

5.1.5. Agente autónomo

Un *agente autónomo* se representa mediante un conjunto no vacío de atributos que representan sus *características permanentes*, un conjunto de *metas* a las que el agente quiere llegar (las cuales tienen asociada su respectiva *motivación*), un conjunto de creencias y un conjunto de *acciones* que es capaz de realizar (Figura 5.5). La clase *AgenteAutonomo* implementa los métodos *actualizaCreencias* y *generaMetas*, y *creaIntencion*, lo cual le permite 1) renovar sus creencias, 2) obtener el conjunto de metas que es factible perseguir y, 3) elegir que meta perseguir en base a sus preferencias. En [12] se encuentra una mejor descripción de tal proceso. Los métodos que representan las habilidades del agente dependen del perfil de puesto, como lo veremos en el Capítulo 6.

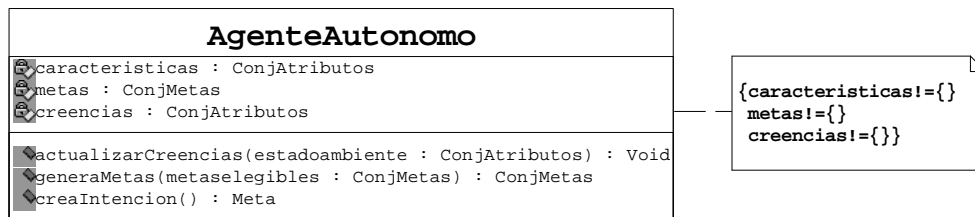


Figura 5.5: Clase AgenteAutonomo

5.1.6. Norma

Las normas de un SMAN se prescriben para ser cumplidas por un conjunto de agentes *destinatarios*. El cumplimiento de estas normas puede beneficiar otro conjunto de agentes, llamados *beneficiarios* de la norma. Las normas especifican algo que debe cumplirse, lo cual es representado en el modelo mediante *metas normativas*. Las normas no siempre son aplicables, su aplicación depende del *contexto* en el que los agentes están situados. También existen excepciones, en las cuales los agentes no están obligados a cumplir con la norma. En algunos casos, las normas sugieren la existencia de un conjunto de *castigos* que se imponen a los agentes que no cumplen con las metas normativas. También existe un conjunto de *recompensas* para los agentes que sí las cumplen. Las recompensas y los castigos se representan como conjuntos de metas (posiblemente vacíos) que deben ser satisfechas por otros agentes. La representación de una norma se muestra en la Figura 5.6.

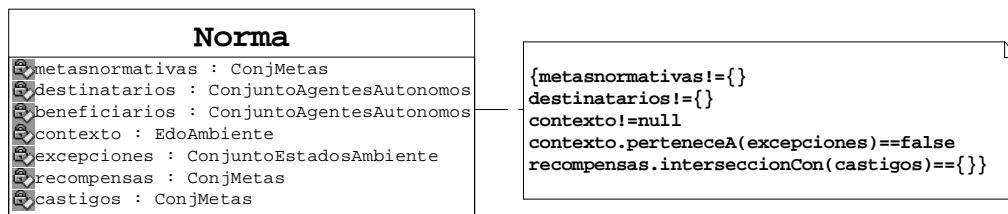


Figura 5.6: Clase Norma

Normas entrelazadas

Algunas veces el *cumplimiento* de unas normas es una *condición para activar otras*. Cuando la activación de una norma depende del cumplimiento pasado de otra, llamamos a esas normas: *normas entrelazadas*. La norma que

provoca la activación de otra es llamada *norma primaria*, mientras que la norma activada como resultado del cumplimiento o violación de la primera es llamada *norma secundaria*. De acuerdo con López y Luck [13], el *contexto* es un estado que se debe mantener para cumplir con una norma. Dado que el cumplimiento de una norma está determinado por medio de sus metas normativas, *el contexto de la norma secundaria debe incluir la satisfacción (o no satisfacción) de todas las metas normativas primarias*.

Formalmente, una norma está *entrelazada por no cumplimiento* si, en el contexto de la norma secundaria, una instancia de la norma primaria puede considerarse violada. La Figura 5.7 muestra gráficamente la relación, las partes de la norma que se encuentran resaltadas indican los conjuntos donde se dan las coincidencias.

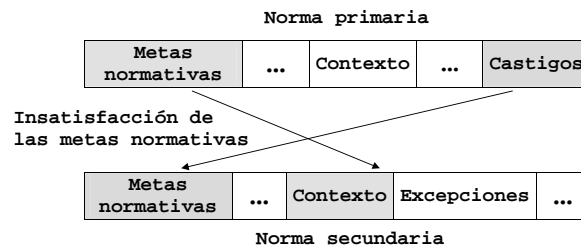


Figura 5.7: Normas entrelazadas por no cumplimiento

Similarmente una norma está *entrelazada por cumplimiento* si, en el contexto de la norma secundaria, una instancia de la norma primaria se considera cumplida, como lo muestra la Figura 5.8.

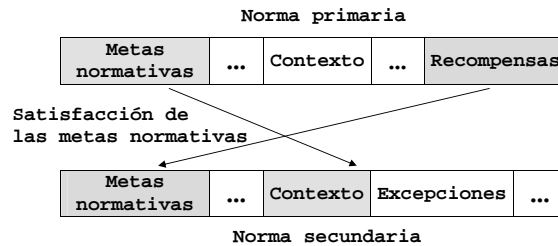


Figura 5.8: Normas entrelazadas por cumplimiento

La Figura 5.9 muestra la relación que existe entre dos normas entrelazadas por *incumplimiento*, representadas mediante un diagrama UML de *restricción*. Una restricción es una expresión booleana representada como

una cadena interpretable en un determinado lenguaje [19]. Los primeros dos renglones de la restricción especifican que para cada meta normativa de la instancia de la norma **n1** debe existir un estado de la norma **n2**, tal que la negación de la meta de la norma primaria **n1** sea igual a un estado del contexto de la norma secundaria **n2**. La segunda condición establece que los castigos de la norma infringida **n1** deben ser un subconjunto de las metas normativas de la norma de refuerzo **n2**.

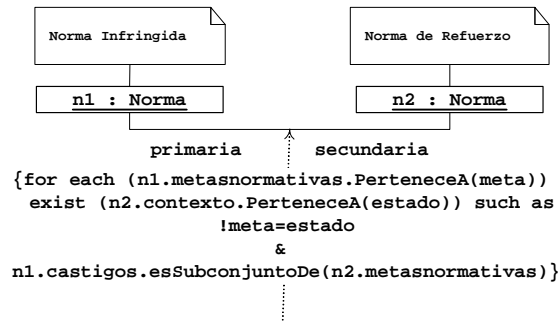


Figura 5.9: Norma de refuerzo

El diagrama UML de la Figura 5.10 muestra la relación que existe entre dos normas relacionadas por cumplimiento. La primera condición de la restricción, establece que las metas normativas de la norma primaria **n1** deben ser un subconjunto del contexto de la norma secundaria **n2**. La segunda condición indica que las metas normativas de la norma de recompensa **n2** deben ser un subconjunto de las recompensas de la norma cumplida **n1**.

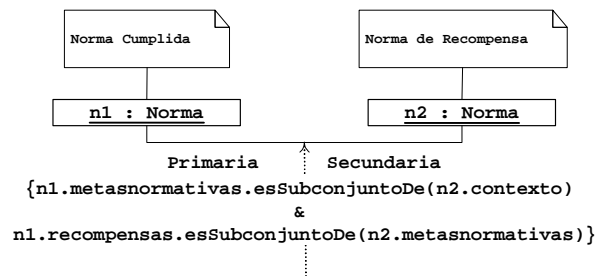


Figura 5.10: Norma de recompensa

5.1.7. Agente normativo

Un agente normativo se define como un agente autónomo cuyo comportamiento está determinado por las obligaciones que debe cumplir, las prohibiciones que limitan los tipos de metas que puede perseguir, los compromisos sociales que han sido creados durante su vida social y los códigos sociales que podrían no tener castigos, pero cuyo cumplimiento es un medio para ser identificado como parte de una comunidad.

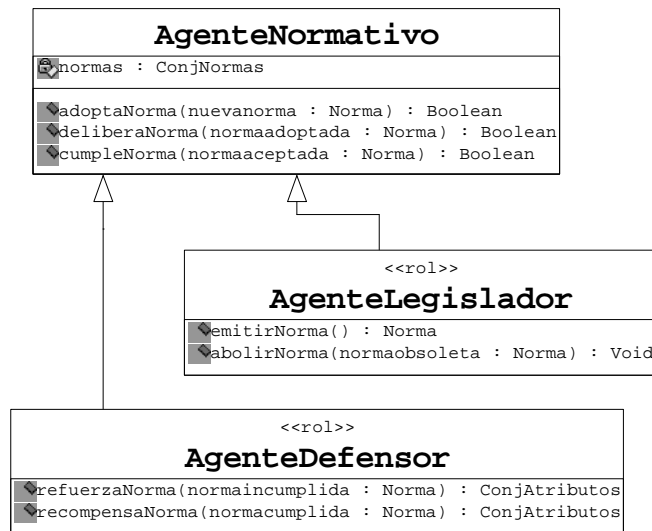


Figura 5.11: Clase **AgenteNormativo**

La clase **AgenteNormativo** es una especialización de **AgenteAutónomo** (ver Figura 5.11). Un agente autónomo tiene asociadas un conjunto de normas. Además, es capaz de *adoptar* nuevas normas y *deliberar* sobre *cumplir* o no cumplir con éstas. En [15] encontramos una mejor descripción de agentes normativos.

Como se mencionó anteriormente, en un SMAN podemos encontrar varios roles. Hay agentes que tienen derecho a *legislar* y por lo tanto crear nuevas normas, también hay agentes autorizados para *castigar* o *recompensar* a otros agentes. La representación de este tipo de agentes también se muestra en la Figura 5.11.

Ya que hemos representado los componentes individuales de un Sistema Multiagentes Normativo, podemos usarlos para crear el diagrama de clases que muestra la Figura 5.12. Un SMAN incluye: un conjunto de *agentes nor-*

mativos, un conjunto de agentes *defensores* y otro de *legisladores*, así como un conjunto de *normas*, que incluye las normas de *legislación*, las de *recompensa* y las de *castigo*.

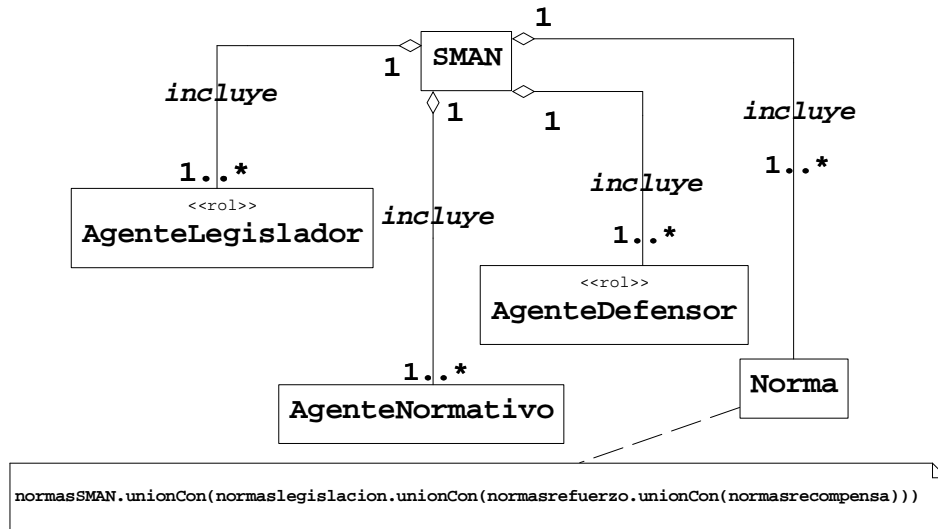


Figura 5.12: Clase SMAN

La Figura 5.13 muestra las restricciones existentes en un SMAN. La primera restricción indica que *todos* los miembros deben adoptar *alguna* de las normas del sistema. La segunda indica, que los agentes destinatarios de *cada* norma del sistema deben ser parte de éste.

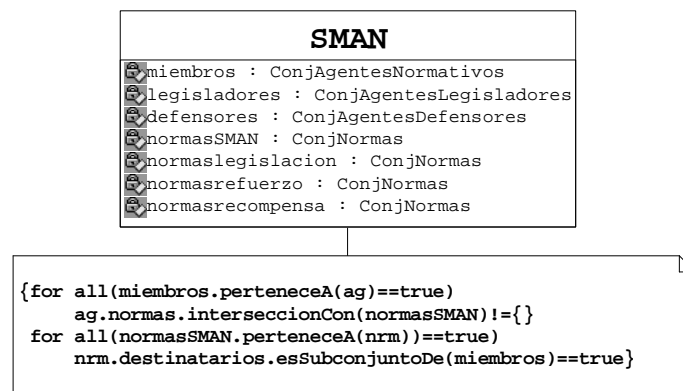


Figura 5.13: Restricciones de la clase SMAN

5.2. Organización Jerárquica

En esta sección expandemos el modelo de SMAN para crear un modelo para organizaciones jerárquicas, al cual denominamos HIERARCHY². Antes de hacerlo, introducimos algunos conceptos tomados de varias teorías de la administración, los cuales nos serán de utilidad para crear el nuevo modelo.

Una *organización humana* [3] es una entidad social orientada a objetivos, conformada por personas, diseñada para conseguir resultados, generar utilidades, proporcionar satisfacción social y estructurada deliberadamente debido a que propone la división del trabajo y asigna su ejecución a sus miembros. La teoría neoclásica [4] añade algunos elementos al concepto de organización formal y la define como *un conjunto de posiciones funcionales y jerárquicas orientadas a la producción de bienes y servicios*.

HIERARCHY se encuentra formada por un conjunto de agentes con habilidades específicas que se coordinan con el fin de alcanzar las metas organizacionales, y de este modo, brindar algún servicio. Además posee una estructura formal que define los roles y las posiciones funcionales de sus miembros. Esta estructura también es utilizada para coordinar las actividades de los agentes de la organización.

5.3. HIERARCHY

A fin de modelar una organización jerárquica se deben considerar los siguientes elementos:

- Las *metas* de la organización.
- Los *servicios* que proporciona.
- Las *normas* que regulan la conducta de sus miembros.
- Los *roles* de los agentes que forman la organización.
- Las *funciones del proceso administrativo* y los *perfiles de puesto*, los cuales nos ayudarán a representar las siguientes estructuras de coordinación.

²**HIER**archical **AR**chitecture for the **C**oordination, **H**eterogeneity and normative **a**utonom**Y** in open multiagent systems.

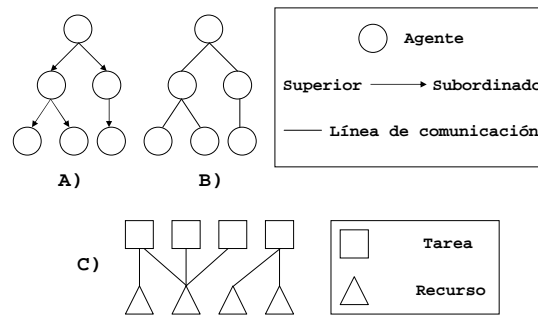


Figura 5.14: Estructuras de coordinación

- *Estructura de autoridad.* Los enlaces muestran quién tiene autoridad sobre quién y quién se reporta con quién, como se muestra en la Figura 5.14 A).
- *Estructura de comunicación.* Establece las líneas de comunicación permitidas, véase la Figura 5.14 B).
- *Estructura de requerimientos.* Liga los recursos o habilidades con las tareas que los necesitan, como lo muestra la Figura 5.14 C).

5.3.1. Recurso

A fin de alcanzar una meta o prestar un servicio, la organización necesita utilizar uno o varios recursos. Los recursos de HIERARCHY se modelan en la clase `Recurso`. Un recurso consiste de un nombre, un tipo, su localización y su disponibilidad, como lo muestra la Figura 5.15.

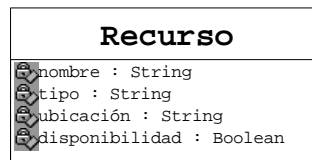


Figura 5.15: Clase Recurso

5.3.2. Meta organizacional

Toda *organización humana* existe no para sí misma, sino para alcanzar objetivos, producir resultados (prestar algún servicio) y obtener ganancias

[4]. Por lo tanto, la organización debe estar determinada, estructurada y orientada en función de sus objetivos. Los objetivos son los valores buscados o los resultados deseados por la organización. Los objetivos son alcanzados a través de la eficiente operación de la organización.



Figura 5.16: Clase MetaOrganizacional

Del mismo modo, una *organización de agentes* está diseñada para alcanzar metas, prestar servicios y obtener ganancias. Las *metas* en HIERARCHY se modelan como un conjunto de estados a los que la organización, como un todo, quiere llegar. La Figura 5.16 ilustra la representación de una meta organizacional, la cual consta de una meta y un plan para alcanzar la meta. Los planes se describirán en la Sección 5.3.7.

5.3.3. Contrato

Un contrato es utilizado para formalizar y garantizar el servicio que se va a proveer. En HIERARCHY se representa un contrato como un conjunto de cláusulas, las cuales definen las condiciones generales del contrato (fechas límite de entrega, pagos, etc.), el objetivo del servicio (el resultado esperado), el tipo de contrato (laboral, de servicio, de transacción, renta, etc.) y el identificador del cliente. Podemos observar el diagrama de clase para un contrato en la Figura 5.17. Los detalles restantes de un contrato, no son discutidos en esta tesis.

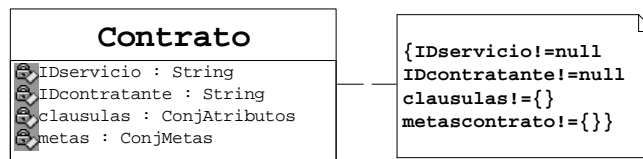


Figura 5.17: Clase Contrato

5.3.4. Servicio

Una *organización humana* está orientada a prestar servicios a otras personas u organizaciones. Su diseño debe permitirle generar utilidades y/o proporcionar satisfacción social [4].

Para una *organización de agentes* un servicio se define como el conjunto de capacidades o recursos de una entidad de software, los cuales pueden ser accedidos por entidades externas (agentes, personas u organizaciones de agentes) a través de una red.



Figura 5.18: Clase Servicio

En HIERARCHY un servicio consiste de un identificador, un conjunto de metas y una descripción del servicio, lo cual se modela como un conjunto de atributos como lo muestra la Figura 5.18.

5.3.5. Norma organizacional

Las normas de una *organización humana* sirven como mecanismos reguladores del comportamiento de sus miembros y controlan de modo informal los niveles de producción [5]. Este modo de control social puede adoptar sanciones *positivas* (estímulos, aceptación social, etc.) o *negativas* (burlas, rechazo por parte del grupo, sanciones simbólicas etc.).

Debido a que las organizaciones se rigen mediante normas, los agentes deben ser capaces de reconocer las normas de su organización. Las normas de HIERARCHY se representan del mismo modo que se hace en [13] (ver Sección 5.1.6). La representación en UML de una norma puede observarse en la Figura 5.6.

5.3.6. Agente organizacional

Los principios generales de la administración [3] afirman que la *especialización* en las tareas es una de las consecuencias de la división del trabajo. Esto, obliga a las personas a ser *especialistas* en determinadas funciones.

Sabemos también que debido a la jerarquía de la organización, sus miembros tienen *derecho a dar órdenes* (autoridad) y el *deber de desempeñar la tarea que se les designe* (responsabilidad). Otra cualidad que deben tener los integrantes de la organización es la de mostrar *respeto a las normas* que rigen su conducta dentro de la organización.

Por otro lado, el principio del orden material y humano dicta que: debe existir un lugar para cada cosa (persona o recurso material) y cada cosa debe estar en su lugar [3]. A partir de esto, podemos decir que los miembros de la organización deben ser capaces de reconocer los recursos a los que tiene acceso (su disponibilidad, ubicación, etc., véase la Figura 5.15), y los agentes con los que tiene una relación de autoridad o responsabilidad, también debe tener conocimiento de las personas con las que debe comunicarse. Por lo tanto, un *agente organizacional* debe tener las siguientes características.

- Un agente debe reconocer y cumplir con las normas de la organización.
- Debe poseer habilidades para realizar ciertas actividades o conseguir metas.
- *Debe ser capaz de adoptar uno o varios roles.*
- *Debe adoptar una o varias posiciones funcionales, las cuales se definen mediante perfiles de puestos.*
- *Debe tener acceso a algún recurso.*
- *Debe reconocer las estructuras de la organización.*

Podemos entonces modelar un agente organizacional, basándonos en un agente normativo [13]. Sin embargo, un agente organizacional debe poseer características adicionales. Debe tener acceso a un conjunto de recursos, además de un mecanismo para reconocer las estructuras de coordinación de la organización a la que pertenece.

En HIERARCHY un *agente organizacional* es un agente normativo que posee un conjunto específico de capacidades, al cual se le asocia un conjunto de recursos, además es capaz de reconocer las autoridades de una organización mediante un perfil de puesto.

La Figura 5.19 muestra como podemos representar un agente organizacional. La clase `AgenteOrganizacional` es una especialización de la clase

AgenteNormativo. Es importante mencionar que en el caso de los agentes organizacionales que también son autoridades, deben representarse heredando de las clases **AgenteLegislador** o **AgenteDefensor**. La clase que representa a un agente organizacional modela los perfiles de puesto que un agente puede desempeñar y los recursos que puede usar.

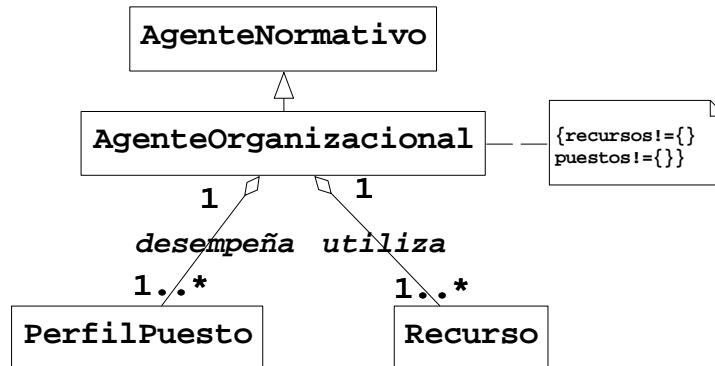


Figura 5.19: Clase **AgenteOrganizacional**

5.3.7. Proceso administrativo

Una *organización humana* puede brindar más de un servicio. La noción de servicio necesita del conocimiento de las funciones del proceso administrativo dentro de una organización. Cada servicio define un proceso administrativo. El proceso administrativo define cuatro actividades [4]:

1. *Planeación.* Determina los objetivos y detalla los planes necesarios para alcanzarlos de la mejor manera posible.
2. *Organización.* Determina las actividades específicas necesarias para el logro de los objetivos planeados (especialización). Agrupa las actividades en una estructura lógica (departamentalización). Asigna las actividades a posiciones y personas específicas (cargos y tareas).
3. *Dirección.* La dirección implica orientación, asistencia en la ejecución, comunicación, motivación, en fin, todos los procesos por medio de los cuales los administradores procuran influir en sus subordinados para que se comporten según las expectativas y consigan los objetivos de la organización

4. *Control*. La esencia del control reside en comprobar si la actividad controlada está alcanzando o no, los resultados deseados. El control es un proceso que guía la actividad hacia un fin determinado. El control es un proceso cíclico compuesto de cuatro fases: establecimiento de los estándares o criterios de desempeño, supervisión del desempeño, comparación del desempeño con el estándar establecido, acción correctiva.

En HIERARCHY representamos las estructuras de coordinación usando las funciones del proceso administrativo y los perfiles de puesto (Ver Sección 5.3.9).

5.3.8. Rol

En toda *organización humana* existe una jerarquía por escalas, estratos o niveles de autoridad, en dicha jerarquía los superiores tienen cierta *autoridad* sobre los inferiores [4]. La *responsabilidad*, es el otro lado de la moneda, responsabilidad significa el deber que tiene una persona de desempeñar la tarea o actividad que le fue designada. En general, el grado de autoridad es proporcional al grado de responsabilidad asumida por la persona.

La jerarquía de una organización define los roles de sus miembros. Un *rol* define el comportamiento esperado de un agente dentro de un grupo. Debemos entonces reconocer los tipos de roles en nuestro modelo. En HIERARCHY distinguimos los roles que muestra la Figura 5.20.

- *Administrador*. Es un agente que posee autoridad sobre los agentes trabajadores. Un administrador implementa las dos primeras funciones del proceso administrativo, como lo ilustra la clase `AgenteAdministrador`.
- *Supervisor*. Este agente verifica que los agentes cumplan con sus obligaciones, a fin de que la meta de la organización se cumpla. Es también, el encargado de efectuar acciones correctivas en caso de ser necesario. La clase `AgenteSupervisor` implementa este rol.
- *Trabajador*. Este tipo de agente debe ser capaz de reconocer la autoridad de los administradores y debe cumplir con sus obligaciones, de modo que beneficien la meta de la organización. Este rol es implementado mediante la clase `AgenteOrganizacional`.

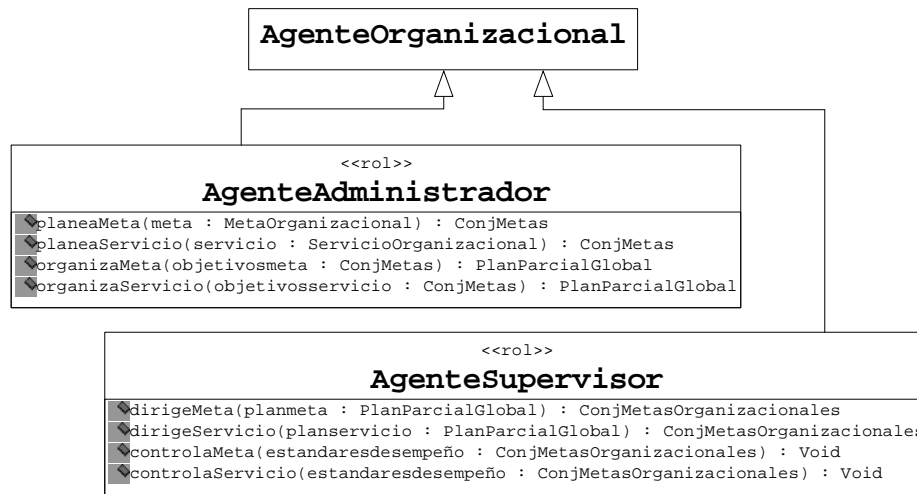


Figura 5.20: Roles en HIERARCHY

Los trabajadores de una *organización humana* son criaturas sociales complejas que tienen sentimientos, *deseos* y temores. Su comportamiento en el trabajo, como en cualquier lugar, es consecuencia de muchos *factores motivacionales* [5]. Las personas se ven motivadas por ciertas *necesidades* y logran sus satisfacciones primarias con la ayuda de los grupos con quienes interactúan. El *comportamiento* de los miembros de una organización puede manejarse mediante un estilo adecuado de supervisión y liderazgo. Un *supervisor* eficaz posee la capacidad de influir en sus subordinados, obtener lealtad de éstos, conseguir estándares elevados de desempeño y un gran *compromiso* con los objetivos de la organización.

Los modelos computacionales anteriores no toman en cuenta el rol de un supervisor, el cual es necesario debido a que los agentes miembros de nuestro modelo de organización son agentes *interesados en sí mismos*, y por lo tanto al igual que los trabajadores de una organización humana van en busca de la satisfacción de sus propias *necesidades*. Las metas individuales y las motivaciones (preferencias) les permiten elegir que normas cumplir. Necesitamos de un medio para controlar el comportamiento de este tipo de agentes, con el fin de preservar el buen desempeño de la organización. Esta es la meta de un agente supervisor.

Cabe mencionar que cada agente en HIERARCHY puede ocupar más de un rol en la organización. Por ejemplo, un agente administrador, puede a su vez, ser un agente trabajador o supervisor. Lo anterior depende de la jerarquía

de la organización que se esté modelando y del perfil de puesto que cubra el agente.

5.3.9. Perfil de puesto

Como un medio para representar la jerarquía de la organización, haremos uso del *análisis de puestos* descrito en [10]. Las partes en que se divide el análisis de puestos son:

1. *Encabezado*. En el encabezado se deben anotar los datos generales de la organización, tales como: la razón social o nombre de la empresa, su rama o giro y su domicilio legal.
2. *Identificación del puesto*. En este apartado se almacenan todos los datos necesarios para la fácil identificación de los puestos que conforman una organización. Estos datos son: el nombre del puesto, su nivel jerárquico y su ubicación dentro de la jerarquía, el número de empleados en el puesto, el puesto inmediato superior, el puesto inmediato inferior, y la supervisión recibida.
3. *Descripción genérica*. En la descripción genérica se definen en forma de objetivos todas las actividades que se realizan en el puesto de forma general.
4. *Descripción específica*. En la descripción específica se hace una exposición detallada de las actividades que el trabajador realiza en un puesto determinado. Las actividades se deben registrar en orden cronológico según la importancia de cada una. Además es necesario registrar el tiempo en el cual se realizaran las mismas.
5. *Especificación del puesto*. La especificación se deriva de la descripción del puesto que pone énfasis en los requisitos mínimos óptimos para desempeñar el trabajo. A estos requisitos también se les llama *factores* y se clasifican en 4 categorías: conocimientos, habilidades, capacidades y condiciones de trabajo.
6. *Perfil*. En el perfil de puesto se especifica el grado en que se requieren las principales características que debe poseer un individuo para desempeñarse en un puesto. Es decir, se recopila la información señalada en los puntos anteriores.

Podemos representar gráficamente las partes que componen un perfil de puesto³, como lo muestra la Figura 5.21.

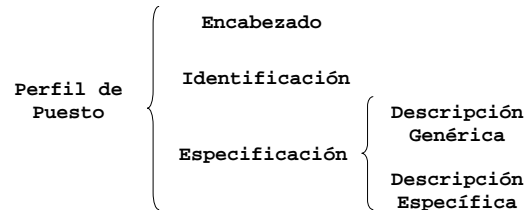


Figura 5.21: Partes de un perfil de puesto

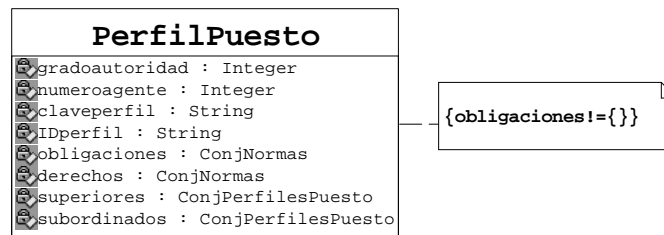
A fin de representar de manera sencilla un perfil de puesto, hacemos uso del formato que se propone a continuación. Este formato consta de dos partes, las cuales se describen a continuación.

Parte I Identificación del puesto

- *Localización en la jerarquía.*
 - *Grado de autoridad-responsabilidad.*
 - *Clave.*
 - *Número.*
 - *Identificador del puesto.*
- *Misión del puesto.*
 - *Obligaciones.*
 - *Derechos.*
- *Relaciones de autoridad.*
 - *Ascendentes.*
 - *Descendentes.*

³El encabezado no es utilizado como parte de la clase `PerfilPuesto`, ya que ésta sólo incluye los elementos que nos permiten coordinar las actividades de un agente en una organización.

La parte I de este formato proporciona los datos necesarios para la creación de la clase `PerfilPuesto`. La clase `PerfilPuesto` (ver Figura 5.22), muestra los elementos que identifican un perfil de puesto mediante los atributos `gradoautoridad`, `numeroagente`, `claveperfil` e `IDperfil`. Se modela también, el conjunto de derechos y el conjunto de obligaciones de los agentes por medio de normas. Podemos diseñar normas que especifiquen autoridad (derechos) y responsabilidad (obligaciones). Por último se muestra el conjunto de perfiles de puestos, que representan a los subordinados y superiores de los agentes.

Figura 5.22: Clase `PerfilPuesto`

La característica principal de una norma que prescribe una *obligación* es que siempre castiga a aquellos agentes que la infringen. Formalmente, una obligación es una norma cuyo incumplimiento siempre se castiga [14], es decir, tiene un conjunto no vacío de castigos. La Figura 5.23 muestra dicha restricción. Por lo tanto, una obligación *siempre* tendrá una norma de refuerzo que castigará su incumplimiento.

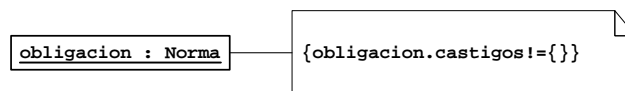


Figura 5.23: Norma que especifica una obligación

Un derecho es otorgado gracias al cumplimiento de una obligación, esto quiere decir, que el cumplimiento de una norma de obligación activará el derecho de un agente `ag1`. Por otro lado, un derecho es una obligación que tiene un agente `ag2` con un agente `ag1`. Esta norma de obligación del agente `ag2` se activa debido al cumplimiento de la obligación del primer agente `ag1`. Resulta claro que la privación de un derecho produce un castigo sobre el agente `ag2`, debido a que no cumple con la obligación que tiene de conceder un derecho al agente `ag1`. El castigo aplicado a el agente `ag2` puede

ser proporcionado por un tercer agente **ag3**. Por otro lado, las normas que representan los *derechos* de un agente no deben imponer un castigo al agente. Todas estas relaciones se resumen en la Figura 5.24, la cual muestra la cadena de normas entrelazadas asociadas a un derecho, así como la restricción que presenta la norma que lo representa.

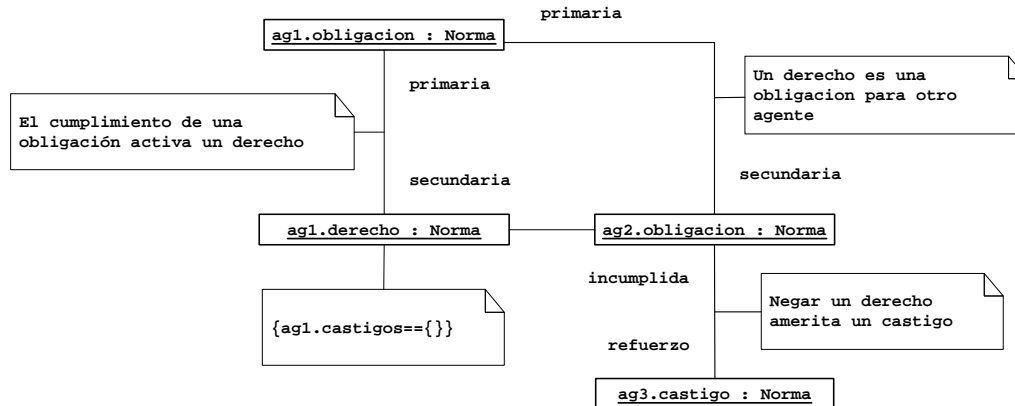


Figura 5.24: Norma que especifica un derecho

Parte II Especificación del puesto

- *Especificación de conocimientos/habilidades/capacidades.*
 - *Conocimientos.*
 - *Habilidades.*
 - *Capacidades.*
- *Condiciones de trabajo.*

Una vez que se ha modelado un perfil, debemos diseñar e implementar un agente organizacional que pueda ocupar el puesto. La Parte II del perfil, describe los conocimientos, habilidades, capacidades y las condiciones de trabajo que debe poseer un agente para desempeñarse en un puesto. Por lo tanto, esta parte nos sirve para modelar un agente organizacional en cuatro pasos. Este proceso se implementa en el Capítulo 6.

1. Los *conocimientos* en un perfil de puesto, definen los métodos que el agente debe implementar. Para implementar las habilidades de un agente podemos heredar de la clase `AgenteAutonomo`, e implementar los métodos que sean necesarios.
2. Las *habilidades*, hacen referencia al *rol normativo* del agente. Es decir, especifican si el agente se encarga de defender normas, ya sea reforzando o recompensando una norma, o si se trata de un agente legislador.
3. Las *capacidades* describen el *rol organizacional* del agente, ya que definen las *capacidades* que tiene el agente con respecto al proceso administrativo. En esta parte se define a un agente como administrador, supervisor o trabajador según corresponda.
4. Las *condiciones de trabajo*, proporcionan información sobre los recursos que necesita un agente para poder desempeñarse en su puesto.

5.3.10. Estructuras de coordinación

En la Sección 5.3 mencionamos las estructuras de coordinación que se encuentran presentes en nuestro modelo. En esta Sección describimos cual es su representación en `HIERARCHY`, y como podemos utilizarlas para coordinar las actividades de una organización. Las estructuras de coordinación que describimos son las siguientes: estructura de autoridad-comunicación y estructura de recursos. Incluimos también una explicación sobre la representación y la utilidad que tiene el proceso administrativo dentro de nuestro modelo.

En nuestra *organización de agentes*, la estructura de autoridad-comunicación se encuentra implícita en la clase `PerfilPuesto`, dado que establece los puestos inmediatos superiores e inferiores para cada cargo en la organización. La estructura de requerimientos, se encuentra presente en la clase `AgenteOrganizacional`. Esta clase contiene la información sobre los recursos a los que el agente tiene acceso.

En una *organización humana* [4] la división del trabajo obliga a que la empresa se divida en tres niveles que conforman el aparato administrativo necesario para dirigir la ejecución de tareas y operaciones.

1. *Nivel institucional*, compuesto por los dirigentes y directores de la organización (*los administradores*).

2. *Nivel Intermedio*, conformado por los gerentes (*los administradores y supervisores*).
3. *Nivel operacional*, formados por los ejecutores de las tareas y operaciones de la empresa (*los trabajadores*).

En HIERARCHY, los agentes que adoptan el rol de administrador deben ser capaces de planear y organizar servicios y metas (Figura 5.20). Por lo tanto, un administrador debe poseer la habilidad de establecer los objetivos para una meta o servicio, como lo indican los métodos `planeaMeta` y `planeaServicio`. Un administrador también debe generar *planes* usando los métodos `organizaMeta` y `organizaServicio`. Nuestro modelo utiliza planes parciales globales como mecanismo de coordinación de las actividades de los agentes de la organización.

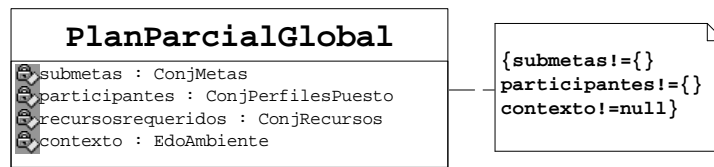


Figura 5.25: Clase `PlanParcialGlobal`

La representación de un plan en HIERARCHY puede observarse en la clase `PlanParcialGlobal` (Figura 5.25). Un *plan* consiste de un conjunto de submetas y el conjunto de perfiles de puesto que puede alcanzarlas. El contexto de un plan describe bajo que circunstancias el plan es aplicable. Por último, la clase `PlanParcialGlobal` especifica que recursos se van a utilizar para llevar a cabo el plan.

Las funciones de dirección y control son implementadas por un supervisor. La dirección puede implementarse usando los métodos `dirigeMeta` y `dirigeServicio`. Estos métodos toman el plan del servicio o la meta y establecen los estándares de desempeño que se esperan de los agentes. El supervisor puede reforzar o castigar el cumplimiento de las normas y estándares de desempeño dado que es el encargado de monitorear el comportamiento de los agentes para asegurarse que los objetivos se estén alcanzando. La función administrativa de control es implementada por los métodos `controlaMeta` o `controlaServicio`, verificando si los estándares de desempeño se están logrando.

Capítulo 6

Aplicaciones del Modelo

Para mostrar la aplicabilidad de HIERARCHY *desarrollamos un ejemplo* de organización jerárquica. Para la realización de este ejemplo, utilizamos actividades de la administración de personal, las cuales facilitan el análisis y desarrollo de un proceso de ingeniería de software. En este capítulo proponemos *una guía* que permite una *fácil implementación* de nuestro modelo.

Del *lado computacional*, retomamos partes del Proceso Unificado (Apéndice A.1), como son la creación de un modelo de casos de uso y un modelo de diseño (Apéndice A.1.2), los cuales representaremos usando la vista de casos de uso y la vista estática de UML (Apéndice A.2.2). Por el *lado organizacional* utilizamos el análisis de puestos, el cual nos permitirá definir los perfiles de puesto de la organización, así como modelar los agentes que desempeñarán las actividades de cada perfil. Ambas actividades nos permiten modelar la *parte estática* del modelo, la cual, una vez concluida, nos permitirá modelar la *parte dinámica* del sistema, la cual se representa en la vista de interacción mediante diagramas de secuencia (Apéndice A.2.2).

Aunque la complejidad del ejemplo aparenta ser *mínima*, el nivel de detalle pretende ser los más *extenso* posible. Este capítulo está organizado de la siguiente manera. La Sección 6.1 describe a grandes rasgos la funcionalidad del sistema por medio de casos de uso. En la Sección 6.2 describimos el sistema desde los puntos de vista estático y dinámico. La Sección 6.2.1 proporciona una guía para el modelado de perfiles de puestos, agentes y normas. La Sección 6.2.2 muestra las colaboraciones que implementan los casos de uso, al mismo tiempo se modelan los elementos que son creados dinámicamente en la organización como son los planes, así como también los cambios que sufre el ambiente de la organización en tiempo de ejecución.

6.1. Modelo de Casos de Uso

Cualquier sistema de software se crea para dar servicio a un conjunto de usuarios. Una *interacción* de un usuario con el sistema genera un caso de uso. Un caso de uso representa una parte del sistema que proporciona al usuario algún resultado importante [20]. El modelo de casos de uso se encarga de representar este tipo de interacciones usuario-sistema mediante diagramas.

Nuestro ejemplo de organización jerárquica es una escuela primaria. La escuela ofrece un curso que consta de tres materias. Las materias que se imparten en el curso son: matemáticas, física y química. El plantel se encuentra formado por un director, un prefecto, un coordinador y un profesor para cada materia.

Por lo tanto, la organización que se quiere modelar es una escuela y el servicio que ésta presta es el curso. Identificamos también a los usuarios del sistema. En este caso, los actores son los estudiantes, los profesores, los coordinadores, el prefecto y el director de la escuela. Encontramos varias relaciones actor-escuela, las cuales describen a grandes rasgos la *funcionalidad del sistema*, las cuales se modelan como casos de uso¹. Este análisis produce el diagrama de casos de uso que se muestra en la Figura 6.1. Por cuestiones de espacio, nos centraremos en explicar únicamente los casos de uso relacionados con el actor Estudiante.

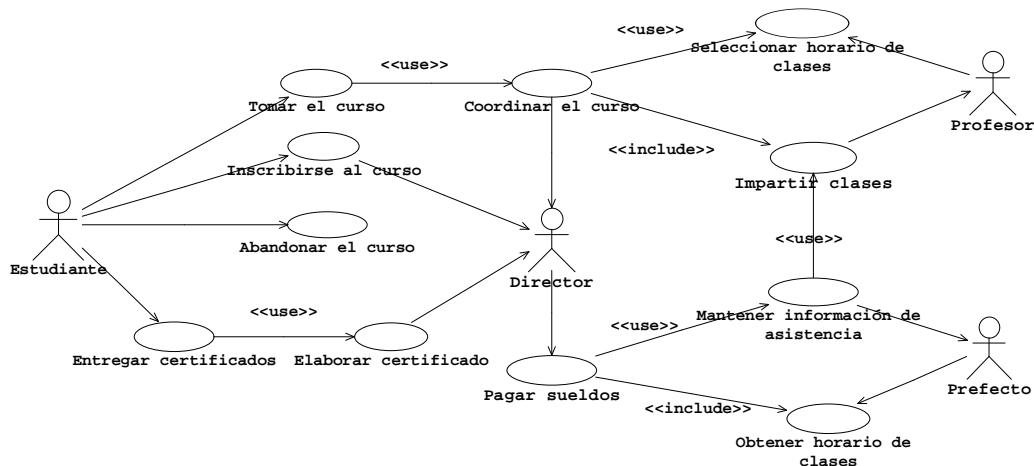


Figura 6.1: Diagrama de casos de uso

¹Algunos casos de uso son actividades humanas, sin embargo las modelamos como actividades realizadas por agentes, a fin de ejemplificar la creación de los perfiles de puesto.

6.2. Modelo de Diseño

Un modelo de diseño se ilustra mediante la vista estática y la vista de colaboración de un sistema. Los elementos principales de la vista estática son las *clases y sus relaciones*: asociación, generalización y varios tipos de dependencia. La vista de colaboración proporciona una visión integral del *comportamiento de un sistema* a través del tiempo, es decir, muestra el flujo de control a través de muchos objetos. La Sección 6.2.1 describe la vista estática de nuestro ejemplo de organización, así como las actividades a realizar para su creación. En la Sección 6.2.2 se modelan los aspectos dinámicos de la organización.

6.2.1. Vista estática

La estructura estática del sistema ya ha sido modelada en el Capítulo 5. El propósito de esta sección es el de representar dicha estructura para un dominio específico por medio de objetos.

Es sabido que los casos de uso deben encajar con la arquitectura cuando se crean y por otro lado, la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos [20]. Por lo tanto, debemos crear objetos y clases que modelen los casos de uso de la escuela.

Análisis de puestos

Una de las partes fundamentales de la arquitectura de nuestro sistema es la *jerarquía de la organización*. Para representar la jerarquía de la organización hacemos uso del análisis de puestos [10].

A continuación presentamos los perfiles de puesto que surgen como resultado del análisis de puestos de la organización. Presentamos también, el modelo de los agentes que cubren los perfiles de puestos, utilizando el formato y el proceso propuesto en la Sección 5.3.9. La Parte I del formato es utilizada para representar fácilmente objetos de la clase `PerfilPuesto`. Por otro lado, la Parte II nos guía durante el proceso que permite modelar los agentes que pueden ocupar tales puestos.

Perfil de Puesto Director

Parte I Identificación del puesto

- *Localización en la jerarquía.*
 - *Grado de autoridad-responsabilidad.* Máximo.
 - *Clave.* DIR
 - *Número.* 01
 - *Identificador del puesto.* 01DIR01.
- *Misión del puesto.*
 - *Obligaciones.* Establecer los contratos con los estudiantes, solicitar las mensualidades a los alumnos inscritos, entregar certificados, pagar y hacer descuentos a los sueldos de los trabajadores de la escuela y coordinar el curso.
 - *Derechos.* Recibir su sueldo.
- *Relaciones de autoridad.*
 - *Ascendentes.* No existe ninguna relación de este tipo.
 - *Descendentes.* Se dan con los coordinadores de matemáticas, física y química.

Para representar el perfil de puesto del director de la escuela como un objeto, lo único que debemos hacer es tomar la información del perfil de puesto y llevarla a su equivalente computacional. Por ejemplo, el *ID* para el perfil de puesto del director es 01DIR01, el cual es una *composición* del *número de agente*, el *identificador* para el perfil de puesto y el *nivel de autoridad*.

Podemos establecer una equivalencia entre el nivel de autoridad y un número entero. Para efectos del presente ejemplo tomaremos las siguientes equivalencias: *máximo=1*, *medio=2* y *bajo=3*. En general, pueden existir n niveles de autoridad en una organización, por lo tanto el diseñador puede definir la cantidad de niveles y equivalencias que le convengan.

El identificador de perfil del director significa que el ocupante del cargo es el primer y único director (el uno de **númeroagente**), y su nivel de autoridad es el máximo dentro de la organización (el uno de **nivelautoridad**). La

misión del puesto es transformada en un conjunto de normas que establecen las obligaciones y derechos del director, y por último las relaciones de autoridad se convierten en dos conjuntos de perfiles de puestos. El conjunto **superiores**, corresponde a las relaciones de autoridad ascendentes, y el conjunto **subordinados** corresponde a las relaciones de autoridad descendentes. La Figura 6.2 muestra el objeto `perfilDirector`.

perfilDirector : PerfilPuesto
<pre> gradoautoridad = 1 numeroagente = 1 claveperfil = DIR IDperfil = 01DIR01 obligaciones = {negociarContrato,solicitarMensualidad entregarCertificado,pagarPersonal coordinarCurso} derechos = {hacerDescuento,recibirPago} superiores = {} subordinados = {perfilCoordMat,perfilCoordFis, perfilCoordQuim} </pre>

Figura 6.2: Perfil de puesto `perfilDirector`

Parte II Especificación del puesto

- *Especificación de conocimientos/habilidades/capacidades.*
 - *Conocimientos.* Sobre la elaboración de contratos y de certificados, y de entrega y solicitud de pagos.
 - *Habilidades.* Manejo y motivación de grupo.
 - *Capacidades.* De planeación y organización de sus subordinados.
- *Condiciones de trabajo.* Una oficina.

Hasta este momento, hemos definido el perfil de puesto para el director, pero una vez que se ha establecido dicho perfil, debemos diseñar e implementar un agente organizacional que pueda desempeñar el puesto. La segunda parte del perfil, describe los conocimientos, habilidades y capacidades que debe poseer un agente para desempeñarse en su puesto.

1. Los *conocimientos* que especifica el perfil de puesto, definen los métodos que el agente debe implementar. La clase `AgenteDirector` implementa los métodos que representan las actividades del director. Tales métodos, le permiten crear un contrato, recibir la mensualidad de un estudiante,

pagarle a un empleado y entregar un certificado, como lo muestra la Figura 6.3.

AgenteDirector	
◆	creaContrato(IDEstudiente : String, IDservicio : String) : Contrato
◆	recibeMensualidad(IDEstudiente : String, mensualidad : Double) : ConjAtributos
◆	pagaEmpleado(Empleado : String) : ConjAtributos
◆	entregaCertificado(IDEstudiente : String) : ConjAtributos

Figura 6.3: Conocimientos del director como métodos

2. Las *habilidades*, se refieren al *rol normativo* del agente. El perfil señala las siguientes habilidades: manejo y motivación de grupo. En este caso, debemos modelar un agente defensor, ya que el director tiene derecho de castigar a otros agentes (sus subordinados), cuando estos no cumplen con sus obligaciones. El director es un defensor, ya que es el destinatario de la norma de castigo *hacerDescuento*, como lo muestra la Figura 6.2.
3. Las *capacidades* se refieren al *rol organizacional* del agente. El agente debe ser capaz de crear un plan global y dividir la meta global en submetas, las cuales son asignadas a sus subordinados. Por lo tanto, debemos modelar al director como un agente administrador.
4. Las *condiciones de trabajo*, nos dan información sobre los recursos que un agente necesita para poder desempeñarse en su puesto. El director necesita de una oficina.

Este proceso puede verse de manera gráfica en la Figura 6.4, la cual muestra del lado izquierdo la(s) clase(s) que se utiliza(n) en cada paso y del lado derecho se muestra el número de apartado, de la Parte II del perfil de puesto, que sirve como guía al diseñador.

La Figura 6.5 muestra el objeto director. Como características, el agente posee un nombre, su meta es ganar \$5000, la motivación para alcanzar dicha meta es de uno, siendo uno la máxima prioridad del agente. Sus creencias son: el día y la fecha actuales, y aún no recibe su sueldo. Decidió aceptar la norma *asistirTrabajo*, ocupa el recurso *oficina* y desempeña las labores que especifica el perfil de puesto *perfiledirector*.

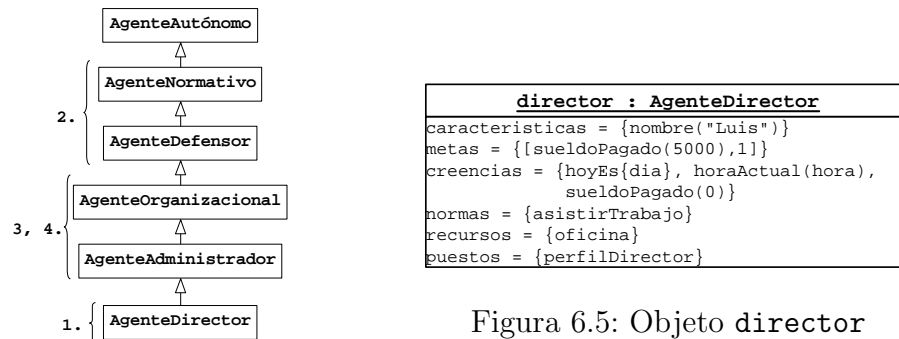


Figura 6.5: Objeto director

Figura 6.4: Agente Director

Perfil de Puesto Prefecto

Parte I Identificación del puesto

- *Localización en la jerarquía.*
 - *Grado de autoridad-responsabilidad.* Medio.
 - *Clave.* PRF
 - *Número.* 01
 - *Identificador del puesto.* 02PRF01.
- *Misión del puesto.*
 - *Obligaciones.* Verificar asistencia.
 - *Derechos.* Recibir su sueldo.
- *Relaciones de autoridad.*
 - *Ascendentes.* Se dan con el director.
 - *Descendentes.* Se dan con los profesores de matemáticas, física y química

El perfil de puesto para el prefecto indica que es el único prefecto en la escuela, dado su *número de agente*. El *nivel de autoridad* con el valor 2 nos indica que el nivel de autoridad del prefecto es medio. La *clave de perfil* es PRF y su *identificador* es 02PRF01. La misión del puesto está conformada por la norma `verificarAsistencia`, la cual establece las obligaciones del

prefecto. El prefecto tiene derecho de recibir su sueldo. Las relaciones de autoridad se encuentran dadas por el conjunto de **superiores**, el cual contiene al director, el conjunto de **subordinados** corresponde al conjunto de profesores. El diagrama de objeto correspondiente se muestra en la Figura 6.6.

perfilPrefecto : PerfilPuesto
gradoautoridad = 2
númeroagente = 1
claveperfil = PRF
IDperfil = 02PRF01
obligaciones = {verficarAsistencia}
derechos = {recibirPago}
superiores = {perfilDirector}
subordinados = {perfilProfMat, perfilProfFis, perfilProfQuim}

Figura 6.6: Perfil de puesto perfilPrefecto

Parte II Especificación del puesto

- *Especificación de conocimientos/habilidades/capacidades.*
 - *Conocimientos.* Sobre la elaboración y manejo de listas de asistencia.
 - *Habilidades.* Motivar al máximo el desempeño individual de sus subordinados.
 - *Capacidades.* De mejorar su desempeño individual.
- *Condiciones de trabajo.* Una oficina.

Una vez representado el perfil de puesto, procedemos a modelar el agente organizacional correspondiente. Repetimos los cuatro pasos utilizados para crear la clase **AdministradorDirector**.

1. Los *conocimientos* requeridos para desempeñar el puesto de prefecto, son modelados mediante el método **pasarLista**, como se muestra en la Figura 6.7.
2. El agente prefecto debe poseer la *habilidad* de motivar al máximo el desempeño individual. Por lo tanto, el prefecto es un agente normativo, que adopta la norma verificar asistencia como su obligación.



Figura 6.7: Conocimientos del prefecto como métodos

3. Las *capacidades* del agente son de mejora del desempeño individual. Por lo tanto, estamos hablando de un agente organizacional.
4. Las *condiciones de trabajo* establecen, que el recurso que utiliza el prefecto es una oficina.

La Figura 6.8 muestra gráficamente el proceso anterior a nivel de clases, el número a la izquierda de cada clase indica el paso en el que se elige utilizarla. La Figura 6.9 muestra una instancia de la clase **AgentePrefecto**. El nombre del prefecto es Joel, su meta es recibir un sueldo de \$3000, sus creencias son la fecha y hora actuales y que no ha recibido sueldo, decidió aceptar la norma *asistirtrabajo*, y posee acceso a una oficina.

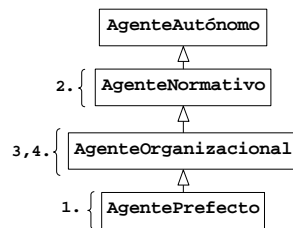


Figura 6.8: Agente Prefecto

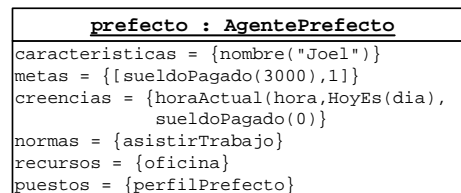


Figura 6.9: Objeto prefecto

Perfil de Puesto Coordinador de Matemáticas

Parte I Identificación del Puesto

- *Localización en la jerarquía.*
 - *Grado de autoridad-responsabilidad.* Medio.
 - *Clave.* CDM
 - *Número.* 01
 - *Identificador del Puesto.* 02CDM01

- *Misión del puesto.*
 - *Obligaciones.* Crear el temario de la materia de matemáticas y llevar las estadísticas de la materia.
 - *Derechos.* Modificar el temario y recibir un sueldo.
- *Relaciones de autoridad.*
 - Las *descendientes* se dan con el profesor de matemáticas
 - Las *ascendentes* se dan con el director.

De acuerdo con el diagrama de objeto de la Figura 6.10, el coordinador de matemáticas es el primer coordinador en la escuela, dado su *número de agente*. El *nivel de autoridad* con el valor 2 nos indica que el nivel de autoridad del coordinador es medio. La *clave de perfil* es CDM y su *identificador* es 02CDM01. Su único *superior* es el director y su *subordinado* es el profesor de matemáticas. La misión del puesto está conformada por las normas `crearTemarioMat` y `llevarEstadística`, la cuales definen las obligaciones del coordinador. El coordinador tiene derecho a recibir un sueldo.

perfilCoordMat : PerfilPuesto
gradoautoridad = 2
númeroagente = 1
claveperfil = CDM
IDperfil = 01CDM02
obligaciones = {crearObjetivosCurso, crearEstadísticasMat}
derechos = {modificarTemario, recibirPago}
superiores = {perfilDirector}
subordinados = {perfilProfMat}

Figura 6.10: Perfil de puesto `perfilCoordMat`

Los perfiles de puesto que corresponden a los coordinadores de física y química se muestran en las Figuras 6.11 y 6.12 respectivamente. Note que ambos perfiles poseen un identificador, un conjunto de subordinados y un conjunto de obligaciones diferente. Estas diferencias dependen de la materia en la que el agente es experto, así como de la ubicación en la jerarquía del perfil de puesto.

perfilCoordFis : PerfilPuesto
gradoautoridad = 2
númeroagente = 2
claveperfil = CDF
IDperfil = 02CDF02
obligaciones = {crearObjetivosCurso, crearEstadisticasFis}
derechos = {modificarTemario, recibirPago}
superiores = {perfilDirector}
subordinados = {perfilProfFis}

Figura 6.11: Instancia del perfil de puesto perfilCoordFis

perfilCoordQuim : PerfilPuesto
gradoautoridad = 2
númeroagente = 3
claveperfil = CDQ
IDperfil = 03CDQ02
obligaciones = {crearObjetivosCurso, crearEstadisticasQuim}
derechos = {modificarTemario, recibirPago}
superiores = {perfilDirector}
subordinados = {perfilProfQuim}

Figura 6.12: Instancia del perfil de puesto perfilCoordQuim

Parte II Especificación del Puesto

- *Especificación de conocimientos/habilidades/capacidades.*
 - *Conocimientos.* Sobre la creación y modificación de temarios, manejo de estadísticas.
 - *Habilidades.* Manejo de grupo y de la motivación.
 - *Capacidades.* De planeación y organización de actividades.
 - *Condiciones de trabajo.* Una Oficina.
1. Los *conocimientos* que debe implementar un agente de este tipo se describen mediante los métodos: `creaTemario`, el cual establece como submetas los temas que el alumno debe aprender en base al servicio que la escuela está ofreciendo; `obtPorcentajeAprobados` y el método `obtPromedioGeneral`, permiten al agente llevar las estadísticas para cada materia (Ver Figura 6.13).

AgenteCoordinador
◀ <code>creaTemario(curso : ServicioOrganizacional) : ConjMetas</code>
◀ <code>obtPorcentajeAprobados(calificaciones : ConjAtributos) : Atributo</code>
◀ <code>obtPromedioGeneral(calificaciones : ConjAtributos) : Atributo</code>

Figura 6.13: Conocimientos del coordinador como métodos

2. Este tipo de agente debe tener *habilidades* de supervisor, ya que debe vigilar el buen desempeño de los profesores que tenga a su cargo. Por lo tanto, el rol normativo de un coordinador es el de legislador ya que este agente tendrá derecho de crear normas que prescriban sanciones negativas a los agentes que no cumplan con su trabajo.

3. Un coordinador debe tener la *capacidad* de dirigir y controlar el trabajo de sus subordinados, para que su comportamiento laboral corresponda al plan establecido.
4. *Condiciones de trabajo*. El coordinador necesita también de una oficina para laborar.

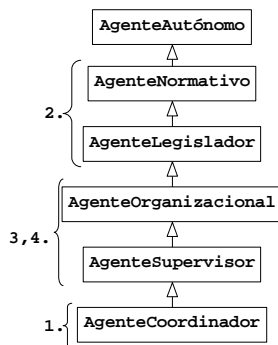


Figura 6.14: Agente Coordinador

```

coordinadorMat : AgenteCoordinador
caracteristicas = {nombre("Virginia")}
metas = {[sueldoPagado(3500),1]}
creencias = {HoyEs(dia),horaActual(hora),
             sueldoPagado(0)}
normas = {asistirTrabajo}
recursos = oficina
puestos = {perfilCoordMat}
    
```

Figura 6.15: Instancia del agente coordinadorMat

El proceso anterior se ilustra en la Figura 6.14. La Figura 6.15 muestra la instancia de la clase `AgenteCoordinador` que corresponde al coordinador de Matemáticas. El nombre del coordinador es *Virginia*, desea recibir un sueldo de \$3500, lo cual es su meta más importante, es destinataria de la norma `asistirTrabajo`, y utiliza una oficina para realizar su trabajo.

```

coordinadorFis : AgenteCoordinador
caracteristicas = {nombre("Angeles")}
metas = {[sueldoPagado(3500),1]}
creencias = {HoyEs(dia),horaActual(hora),
             sueldoPagado(0)}
normas = {asistirTrabajo}
recursos = {oficina}
puestos = {perfilCoordFis}
    
```

Figura 6.16: Instancia del agente coordinadorFis

```

coordinadorQuim : AgenteCoordinador
caracteristicas = {nombre("Eugenia")}
metas = {[sueldoPagado(3500),1]}
creencias = {hoyEs(dia),horaActual(hora),
             sueldoPagado(0)}
normas = {asistirTrabajo}
recursos = {oficina}
puestos = {perfilCoordQuim}
    
```

Figura 6.17: Instancia del agente coordinadorQuim

Un análisis similar se hace con los perfiles de puesto de los otros dos coordinadores, el de física y el de química, ya que realizan actividades similares. Las instancias para estos agentes se muestran en las Figuras 6.16 y 6.17

Perfil de Puesto Profesor de Matemáticas

Parte I Identificación del Puesto

- *Localización en la jerarquía.*
 - *Grado de autoridad-responsabilidad.* Bajo.
 - *Clave.* PRFM
 - *Número.* 01
 - *Identificador del puesto.* 01PRFM03
- *Misión del puesto.*
 - *Obligaciones.* Impartir clases, evaluar a los alumnos, preparar clases y llevar el control de los estudiantes.
 - *Derechos.* Recibir sueldo.
- *Relaciones de autoridad.*
 - No tiene relación *descendente* alguna.
 - Las *ascendentes* se dan con el coordinador de matemáticas.

De acuerdo con la Parte I del perfil de puesto para el profesor de matemáticas, creamos el diagrama de objeto de la Figura 6.18. El profesor de matemáticas es el primer profesor en la escuela, dado su *número de agente*. El *nivel de autoridad* tiene un valor de 3, lo cual indica que el nivel de autoridad del profesor es bajo. La *clave para el perfil* es PRFM y su *identificador* es 01PRFM03. Sólo tiene un *superior*: el coordinador de matemáticas. El profesor no tiene *subordinados*. La misión del puesto está conformada por las normas `impartirClase`, `pasarLista` y `evaluarEstudiantes`. Tales normas definen las obligaciones del profesor. El agente que ocupe el puesto del profesor de matemáticas tiene derecho a recibir un sueldo y a utilizar un salón de clases.

Los perfiles de puesto para los profesores de física y química se muestran en las Figuras 6.19 y 6.20 respectivamente. El análisis para estos perfiles de puestos es análogo al realizado con los coordinadores de cada materia. Note que ambos perfiles poseen un identificador diferente y cada uno posee un conjunto de subordinados diferente, el cual depende de la materia.

perfilProfMat : PerfilPuesto
<pre> gradoautoridad = 3 numeroagente = 1 claveperfil = PRFM IDperfil = 01PRFM03 obligaciones = {impartirClase, pasarLista evaluarEstudiantes} derechos = {recibirPago, utilizarSalon} superiores = {perfilCoordMat} subordinados = {} </pre>

Figura 6.18: Perfil de puesto perfilProfMat

perfilProfFis : PerfilPuesto	perfilProfQuim : PerfilPuesto
<pre> gradoautoridad = 3 numeroagente = 2 claveperfil = PRFF IDperfil = 02PROFF03 obligaciones = {impartirClase, pasarLista evaluarEstudiantes} derechos = {recibirPago, utilizarSalon} superiores = {perfilCoordFis} subordinados = {} </pre>	<pre> gradoautoridad = 3 numeroagente = 3 claveperfil = PRFQ IDperfil = 03PRFQ03 obligaciones = {impartirClase, pasarLista evaluarEstudiantes} derechos = {recibirPago, utilizarSalon} superiores = {perfilCoordQuim} subordinados = {} </pre>

Figura 6.19: Instancia del perfil de puesto perfilProfFis Instancia del perfil de puesto perfilProfQuim

Parte II Especificación del puesto

- *Especificación de conocimientos/habilidades/capacidades.*
 - *Conocimientos.* Sobre la impartición de clases de matemáticas métodos de evaluación y de elaboración de listas.
 - *Habilidades.* Manejo de grupo, de la motivación y de la presión.
 - *Capacidades.* Motivar al máximo el desempeño individual de los alumnos de manera que se mejore al máximo su rendimiento y se logre la satisfacción total de los usuarios de sus servicios.
 - *Condiciones de trabajo.* Un salón de clases.
1. Los *conocimientos* necesarios para desempeñarse como profesor se describen mediante tres métodos. El método `imparteClase` recibe como parámetro el tema que se debe impartir y devuelve como resultados un conjunto de atributos, el cual representa los conocimientos que el profesor genera; el método `evaluaAlumno` recibe como parámetros el

nombre del alumno y el tema que va a evaluarse y obtiene como resultado la calificación del alumno, la cual se representa como un conjunto de atributos; y por último, el método `verificaAsistencia`, este método ayuda al coordinador a llevar la estadística de la materia. La Figura 6.21 muestra el diagrama de clase correspondiente.

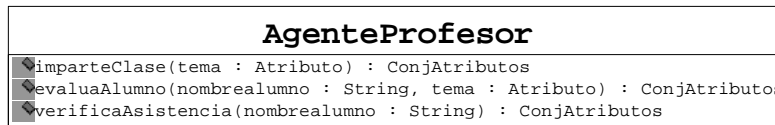


Figura 6.21: Conocimientos del profesor como métodos

2. Dado que el perfil establece un nivel de autoridad bajo. Este tipo de agente sólo necesita tener las *habilidades* básicas de un agente normativo.
3. Debido a que el agente no posee subordinados, un profesor sólo debe tener la *capacidad* de adoptar el rol de trabajador.
4. *Condiciones de trabajo*. El profesor necesita un salón de clases para poder trabajar.

El proceso de modelado para este tipo de agente se ilustra en la Figura 6.22. La Figura 6.23 muestra una instancia de la clase `AgenteProfesor`. Su nombre es `Manuel`, su meta es recibir un sueldo de \$3000, sus creencias incluyen el día y la fecha actuales. La norma que decidió aceptar es `asistirTrabajo`. Un análisis similar se hace con los profesores de Física y Química. Dicho análisis se ve reflejado en las Figuras 6.24 y 6.25.

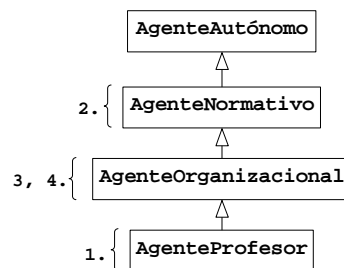


Figura 6.22: Agente Profesor

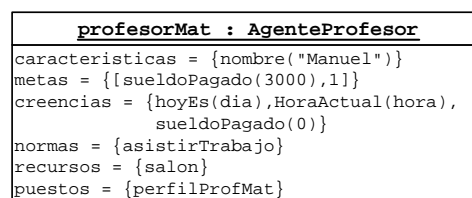


Figura 6.23: Objeto profesorMat

profesorFis : AgenteProfesor
características = {nombre("Natividad")} metas = {[sueldopagado(3000),1]} creencias = {hoyEs(dia),horaActual(hora), sueldoPagado(0)} normas = {asistirTrabajo} recursos = {salon} puestos = {perfilProfFis}

Figura 6.24: Objeto profesorFis

profesorQuim : AgenteProfesor
características = {nombre("Alma")} metas = {[sueldopagado(3000),1]} creencias = {hoyEs(dia),horaActual(hora), sueldopagado(0)} normas = {asistirTrabajo} recursos = {salon} puestos = {perfilProfQuim}

Figura 6.25: Objeto profesorQuim

A continuación presentamos la estructura jerárquica de la escuela vista desde 2 perspectivas. La Figura 6.26 muestra las relaciones de autoridad que existen entre roles. En cada extremo de la línea se especifica cual es el rol que desempeña la clase. Por otro lado, la Figura 6.27 muestra la misma jerarquía, pero tomando en cuenta los perfiles de puesto que se definen en ella. En este diagrama se muestra quién tiene autoridad sobre quién, entre más cerca de la cima de la jerarquía mayor es el nivel de autoridad.

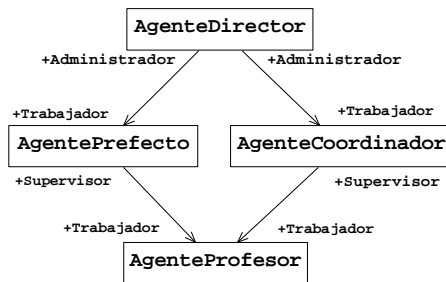


Figura 6.26: Jerarquía a nivel de roles

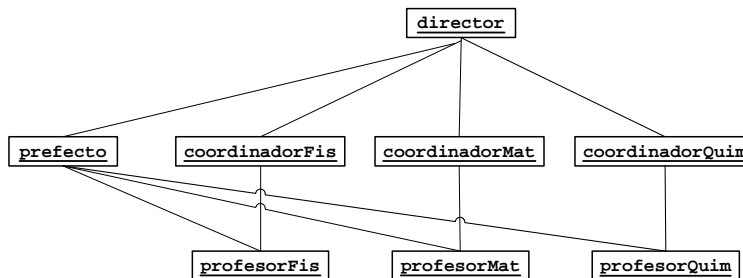


Figura 6.27: Jerarquía a nivel de perfiles de puesto

Normas

En esta sección se describen algunas de las normas asociadas con los perfiles de puesto. Es importante describirlas detalladamente ya que definen el comportamiento esperado de los agentes que forman la organización. Debido a las restricciones de tiempo sólo describimos las normas que nos ayudan a coordinar las actividades de la organización, las cuales, complementan y colaboran, con la elaboración de los planes y las metas globales. Esto último se explicará en la Sección 6.2.2.

Perfil de Puesto Director

Comenzaremos describiendo las obligaciones del director. La Figura 6.28 muestra el diagrama de objeto para la norma `crearContrato`. Esta norma se activa cuando un alumno desea inscribirse al curso, en otras palabras, existe un nuevo aspirante, como lo indica el contexto. Las metas² de la norma especifican que el director debe crear un contrato para cada alumno inscrito, es decir, para cada nuevo estudiante. Si no se cumple con esta norma el castigo³ es un descuento al salario.

<code>crearContrato : Norma</code>
<code>beneficiarios = {}</code>
<code>destinatarios = {director}</code>
<code>metasnormativas = {esEstudiante(nuevoEstudiante), tieneContrato(nuevoEstudiante)}</code>
<code>recompensas = {}</code>
<code>castigos = {hacerdescuento(director)}</code>
<code>contexto = {esAspirante(nuevoEstudiante)}</code>
<code>excepciones = {}</code>

Figura 6.28: Norma `crearContrato`

Solicitar la mensualidad de los estudiantes inscritos es una más de las obligaciones del director. La Figura 6.29 muestra el diagrama UML, de la instancia de dicha norma. La norma `solicitarMensualidad` se activa los días de pago(`contexto`). La norma no aplica si el alumno cubrió su mensualidad con anterioridad (`excepciones`). Si la mensualidad no se cobra, se aplica un descuento al salario del director (`castigos`).

²En los diagramas de objeto de las normas se omite la prioridad de las metas normativas, castigos y recompensas.

³Debido a las limitaciones de tiempo y espacio, no se modelan las normas de refuerzo de las obligaciones descritas en este capítulo.

<u>solicitarMensualidad : Norma</u>
beneficiarios = {}
destinatarios = {director}
metasnormativas = {mensualidadPagada(estudiante)}
recompensas = {}
castigos = {hacerDescuento(director)}
contexto = {esEstudiante(estudiante), hoyEs(FechaPago)}
excepciones = {mensualidadPagada(estudiante)}

Figura 6.29: Norma solicitarMensualidad

Una de las tareas más importantes del director es la de coordinar el servicio organizacional, es decir, el curso. La norma `coordinarCurso` (Figura 6.30) es una más de las obligaciones⁴ de este perfil de puesto. La meta de esta norma es la dejar el curso “listo” para iniciarse. Esta norma se activa una vez que existen al menos cinco estudiantes escritos, y todo el personal se encuentra en disposición de iniciar el curso en una fecha estipulada con anterioridad.

<u>coordinarCurso : Norma</u>
beneficiarios = {}
destinatarios = {director}
metasnormativas = {estadoCurso("listo")}
recompensas = {}
castigos = {}
contexto = {cantidadEstudiantes(5), inicioCurso(fechainicio), porcentajePersonalDisponible(100)}
excepciones = {}

Figura 6.30: Norma coordinarCurso

Una más de las misiones del director es pagar los sueldos de todos los trabajadores de la escuela, quienes evidentemente son los beneficiarios. Esta norma se activa siempre y cuando no se haya violado la norma `asistirTrabajo`. Si el director cumple con esta norma se la paga su sueldo. Esta norma no cuenta con excepciones.

La norma `pagarPersonal` (Figura 6.31) representa la *obligación* que tiene el agente director de *cumplir con el derecho que tienen el resto de los agentes*, de recibir un sueldo. Esta situación ejemplifica la cadena de normas entrelazadas que genera una norma que especifica un derecho, lo cual fue explicado en el Capítulo 5.

⁴Observe que no todas las obligaciones descritas en este capítulo tienen un castigo asociado, pero lo correcto es definir al menos un castigo por cada obligación, tal como lo establecimos en el Capítulo 5.

pagarPersonal : Norma
beneficiarios = {prefecto, coordMat, coordFis, coordQuim, profMat, profFis, profQuim}
destinatarios = {director}
metasnormativas = {sueldoPagado(prefecto), sueldoPagado(coordMat), sueldoPagado(coordFis), sueldoPagado(coordQuim), sueldoPagado(profMat), sueldoPagado(profFis), sueldoPagado(profQuim)}
recompensas = {sueldoPagado(director)}
castigos = {hacerDescuento(director)}
contexto = {asistenciaCompleta(Prefecto), asistenciaCompleta(coordMat), asistenciaCompleta(coordFis), asistenciaCompleta(coordQuim), asistenciaCompleta(profMat), asistenciaCompleta(profFis), asistenciaCompleta(profQuim)}
excepciones = {}

Figura 6.31: Norma pagarPersonal

Cada vez que un estudiante aprueba sus tres materias se le entrega un certificado, con la condición de que el alumno tenga cubiertas sus mensualidades (Figura 6.32).

entregarCertificado : Norma
beneficiarios = {}
destinatarios = {director}
metasnormativas = {certificadoEntregado(estudiante)}
recompensas = {}
castigos = {hacerDescuento(director)}
contexto = {aprobo(Mat), aprobo(Fis), aprobo(Quim), cursoPagado(estudiante)}
excepciones = {}

Figura 6.32: Norma entregarCertificado

Perfil de Puesto Coordinador

Las obligaciones del coordinador comprenden la creación de los objetivos de cada curso así como la obtención de las estadísticas finales. La Figura 6.33 muestra la instancia de la norma `crearObjetivosCurso`. Esta norma tiene como beneficiario al director, ya que el cumplimiento de esta norma refuerza la norma `coordinaCurso`, la cual está dirigida al director. La norma está destinada a los coordinadores de las tres materias. El objetivo de esta

meta es la de crear los objetivos de cada curso cada vez que el estado del curso sea “listo”.

crearObjetivosCurso : Norma
beneficiarios = {director}
destinatarios = {coordinadorMat,coordinadorFis, coordinadorQuim}
metasnormativas = {objetivosCreados(curso)}
recompensas = {}
castigos = {}
contexto = {estadoCurso("listo")}
excepciones = {}

Figura 6.33: Norma crearObjetivosCurso

La norma `crearEstadisticasMat`, está dirigida al coordinador de la materia de matemáticas, a diferencia de la norma anterior que estaba dirigida a todos los coordinadores. Esta norma establece que una vez terminado el curso el director debe obtener el promedio general y la cantidad de alumnos reprobados en la materia de matemáticas, como lo muestra la Figura 6.34.

crearEstadisticasMat : Norma
beneficiarios = {}
destinatarios = {coordinadorMat}
metasnormativas = {porcentajeAprobados(tema1,cantidad1), porcentajeAprobados(tema2,cantidad2), porcentajeAprobados(tema3,cantidad3), promedioGeneralMat(cantidad)}
recompensas = {}
castigos = {}
contexto = {estadoCurso("terminado")}
excepciones = {}

Figura 6.34: Norma crearEstadisticasMat

Recursos

Debido a las limitaciones de espacio, asumimos que sólo existen dos recursos en la escuela: un *salón* y una *oficina*. La Figura 6.35 muestra que el nombre del salón es d4-204. El tipo de recurso muestra que el salón es un edificio. La ubicación del salón es local y se encuentra disponible. La Figura 6.36 muestra las características de la oficina. El identificador para la oficina es d5-201, el resto de los datos son idénticos a los del recurso `salon`.

salon : Recurso
nombre = "d4-204"
tipo = "edificio"
ubicacion = "local"
disponibilidad = true

Figura 6.35: Objeto salon

oficina : Recurso
nombre = "d5-201"
tipo = "edificio"
ubicacion = "local"
disponibilidad = true

Figura 6.36: Objeto oficina

6.2.2. Vista de colaboración

La vista de colaboración provee una visión general del comportamiento de un sistema, dado que muestra el flujo de control del sistema a través de muchos objetos. La vista de interacción es mostrada en dos diagramas que se enfocan en diferentes aspectos: *los diagramas de secuencias y los diagramas de colaboración*.

A efectos de no repetir explicaciones mostraremos el comportamiento de nuestro sistema utilizando únicamente diagramas de secuencia. Un diagrama de secuencia *muestra un conjunto de mensajes ordenados en una secuencia de tiempo* [19]. Utilizamos este tipo de diagramas porque, se centran en las secuencias de tiempo de los mensajes [19], lo cual nos permite mostrar como se coordinan las actividades de los agentes en tiempo de ejecución.

Caso de uso: Inscribirse al curso

De acuerdo con nuestro diagrama de casos de uso el primer servicio que presta nuestro sistema es el de *inscribirse al curso*. Si un aspirante desea convertirse en alumno ocurre el siguiente proceso (Ver Figura 6.39).

1. El alumno solicita el servicio a la organización, mediante el método `solicitaServicio`, el cual recibe como parámetro el identificador del agente, en este caso, el nombre del agente. Este método devuelve como resultado el predicado `esAspirante("Jesús")`.

ambienteescuela : EdoAmbiente
estado = {enSalon(null), esEstudiante(null), tieneContrato(null), cantEstudiantes(0), esAspirante("Jesús"), mensualidadPagada(null), hoyEs("25/04/05")}

Figura 6.37: Contexto de la norma `crearContrato`

2. La organización actualiza su estado del ambiente mediante el método `establecerEstado`, las cuales corresponden al contexto de la norma `crearContrato` como lo muestra la Figura 6.37.
3. El director obtiene el estado del ambiente de la escuela haciendo una llamada al método `obtenerEstado`.
4. Posteriormente el director actualiza sus creencias mediante el método `actualizaCreencias`⁵.
5. Las creencias actuales del director activan la norma `crearContrato`, la cual forma parte de las obligaciones de su perfil de puesto. Por lo tanto, la norma debe ser cumplida. Con tal fin, se activa el método `cumplirNorma`.
6. El director obtiene el identificador, las metas y la descripción del servicio mediante los métodos `obtID`, `obtMetas`, `obtDescripción`. Esta información se encuentra ilustrado en la Figura 6.38. Tal objeto indica que las metas del servicio son terminar los cursos de matemáticas, física y química. También se proporciona una descripción del curso la cual incluye su duración en meses, el documento que se entrega y el monto de las mensualidades. La creación del plan para este servicio se muestra en la siguiente sección.

<u>curso : ServicioOrganizacional</u>
<pre> IDservicio = "S01" metasservicio = {[cursoTerminado("Matemáticas"), 1], [cursoTerminado("Física"), 2], [cursoTerminado("Química"), 3]} descripcionservicio = {duracionCurso(4), pagomensual(500), documentoEntregado("Certificado")} planservicio = plancurso </pre>

Figura 6.38: Objeto `curso`

7. El director envía al aspirante los informes sobre el servicio mediante el método `estInfoServicio`. Este método recibe como parámetro el identificador las metas y la descripción del servicio mencionados en el punto anterior.

⁵En éste y el resto de los ejemplos se asume la ejecución de los métodos `generaMetas` y `creaIntencion` después de que un agente actualiza sus creencias. Tales acciones completan el proceso de deliberación.

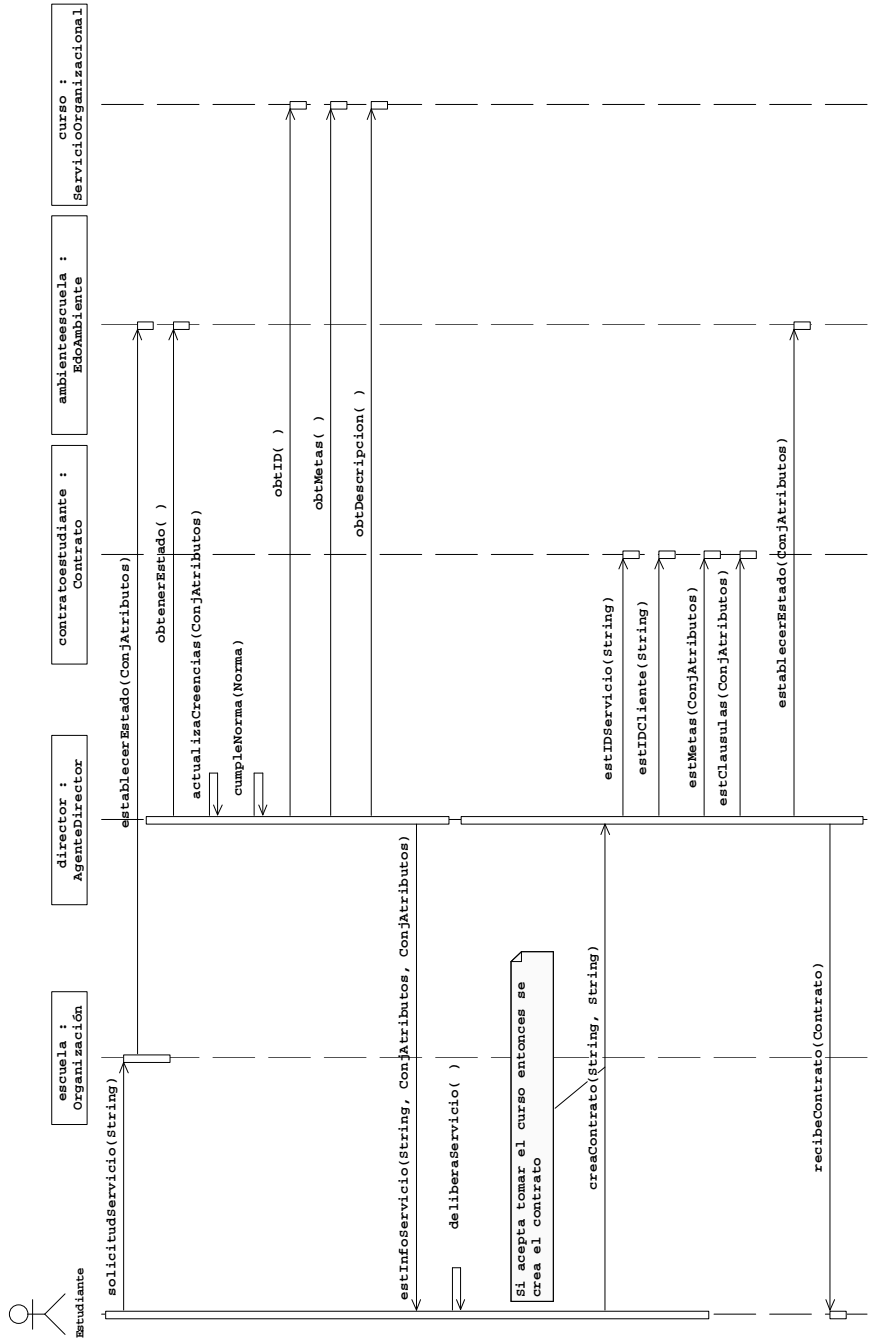


Figura 6.39: Secuencia Inscribirse al curso

8. El estudiante delibera si tomar o no el curso, basándose en la información que acaba de recibir, mediante el método `deliberaCurso`.
9. Si el alumno decide tomar el curso solicita al director la creación de su contrato. Mediante el método `creaContrato`, el cual tiene como parámetros el identificador del servicio y el del cliente.

contratoEstudiante : Contrato
<pre> IDservicio = "S01" IDcliente = "Jesus" metacontrato = {certificadoEntregado("Jesus")} clausulas = {porcentajeAsistenciaEstudiante(100), porcentajeMensualidadesPagadas(100), porcentajeClasesImpartidas(100)} </pre>

Figura 6.40: Objeto `contratoEstudiante`

10. El director establece cada uno de los atributos del contrato, creando así el objeto `contratoEstudiante` (Figura 6.40). El contrato establece que debe cubrirse el monto de las mensualidades al 100%. El mismo caso ocurre con el porcentaje de asistencias y el porcentaje de clases impartidas. El contrato también establece las metas que deben alcanzarse.
11. Una vez que se ha creado el contrato el director actualiza el estado del ambiente y crea los predicados que comprueban que se han alcanzado las metas de la norma `crearContrato`. Tal estado del ambiente se muestra en la Figura 6.41.

ambienteescuela : EdoAmbiente
<pre> estado = {esEstudiante("Jesús"), tieneContrato("Jesús"), cantEstudiantes(1), enSalon(null), mensualidadPagada(null), hoyEs("25/04/05")} </pre>

Figura 6.41: Metas de la norma `crearContrato`

Caso de uso: Tomar el curso

Este caso de uso será dividido en varias colaboraciones a fin de proporcionar una explicación clara.

Crear el plan general

Esta sección explica paso a paso la elaboración del plan que se implementa para impartir el curso. Este proceso se ilustra en la Figura 6.44.

1. Primero, el director verifica el estado del ambiente (`obtenerEstado`) y actualiza sus creencias (`actualizaCreencias`).

<u>ambienteescuela : EdoAmbiente</u>
estado = {cantidadEstudiantes(5), hoyEs("08/08/05"), porcentajePersonalDisponible(100)}

Figura 6.42: Contexto de la norma `coordinarCurso`

2. El director advierte que el estado del ambiente corresponde al contexto de la norma `coordinaCurso` y por lo tanto, se activa la norma, lo cual produce la ejecución del método `cumpleNorma`. La Figura 6.42 muestra dicho estado del ambiente. En éste se indica la cantidad de estudiantes mínima para iniciar el proceso administrativo que corresponde a la ejecución del servicio, la fecha de inicio del curso, y la disponibilidad de todo el personal. Omitimos los detalles restantes acerca del ambiente por cuestiones de brevedad y claridad.
3. El director define los objetivos del servicio haciendo uso del método `planeaServicio`. El resultado que devuelve este método consta de tres metas, una para cada curso.

<u>coordinarCursoMat : MetaOrganizacional</u>
meta = {objetivosCreados("Matemáticas"),1} planmeta = planMatematicas

Figura 6.43: Meta organizacional `coordinarCursoMat`

4. Las metas organizacionales que se generan en el paso anterior son instanciadas mediante el método `agregarMetasGlobales`, para posteriormente implementar el proceso administrativo apropiado para cada una de ellas. Por ejemplo, la meta que se genera para el curso de matemáticas, es la de establecer los objetivos del curso de matemáticas, como lo muestra la Figura 6.43.

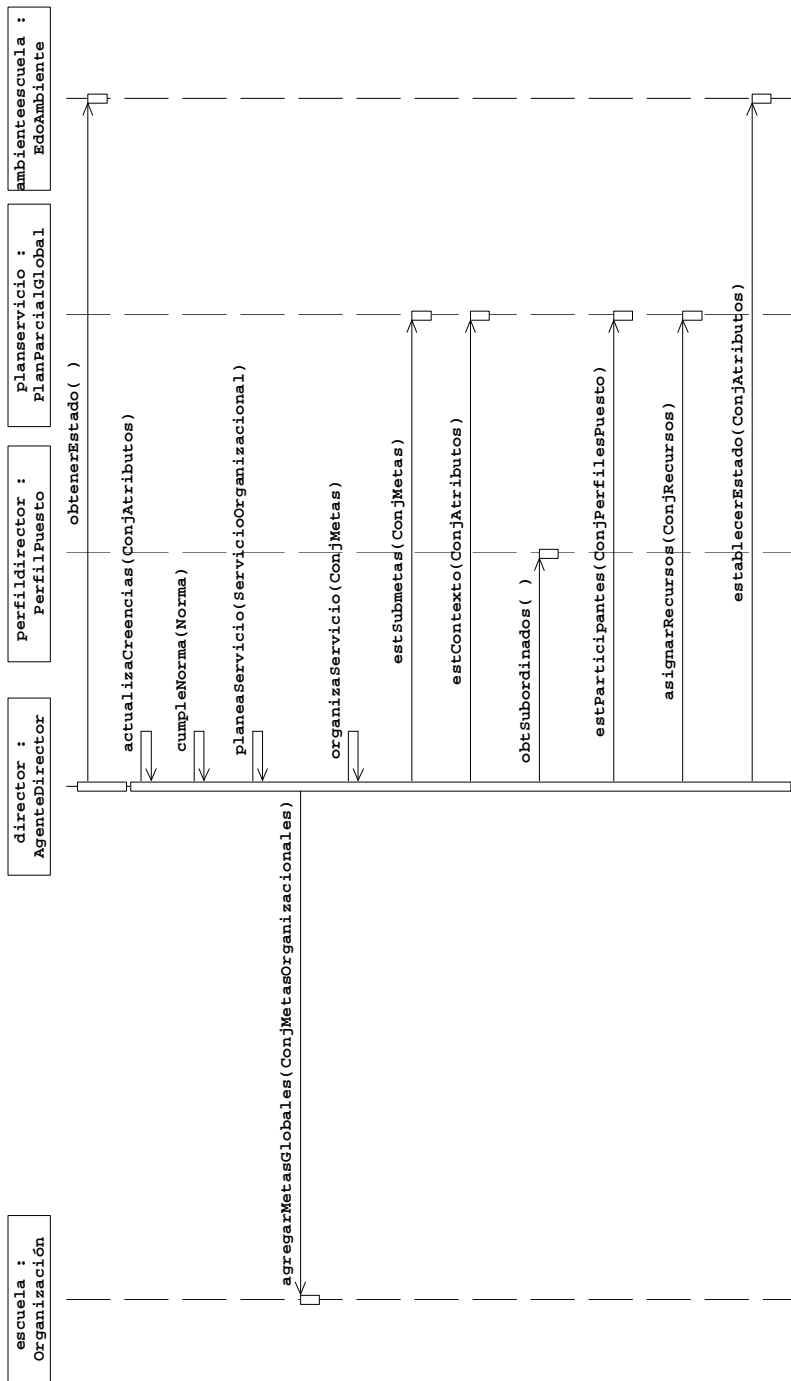


Figura 6.44: Secuencia Crear el plan general

5. Posteriormente, el director debe crear un plan para que el alumno pueda tomar el curso. Con este fin, recurrimos al segundo paso del proceso administrativo, el cual es implementado mediante el método `organizaServicio`.
6. Primero, se establece el identificador del servicio junto con las submetas del servicio. Para elegir a los participantes del plan, el director revisa el perfil de puesto de sus subordinados, a fin de determinar quien es el encargado de cada meta, y por último se asignan recursos. Para establecer el contexto bajo el cual el plan es aplicable, se utiliza el contexto de la norma `crearObjetivosCurso`, la cual forma parte de las obligaciones de los perfiles de puesto de los coordinadores. La instancia creada para el plan se muestra en la Figura 6.45.

<u>planCurso : PlanParcialGlobal</u>
<pre> submetas = {objetivosCreados("Matemáticas"), objetivosCreados("Física"), objetivosCreados("Química")} participantes = {perfilCoordMat, perfilCoordFis, perfilCoordQuim} contexto = {estadoCurso("listo")} recursos = {oficina} </pre>

Figura 6.45: Plan `planCurso`

7. En el último paso el director actualiza el estado del ambiente agregando el predicado `estadoCurso("listo")`. Este predicado activará la norma `crearObjetivosCurso`, lo cual explicaremos en la siguiente sección.

Dirigir el curso

La dirección, como parte del proceso administrativo, se encarga de establecer los estándares de desempeño. Esta sección tiene como propósito explicar como es que se lleva a cabo esta operación (Ver Figura 6.51).

1. El coordinador actualiza sus creencias, mediante la combinación de los métodos `obtenerEstado` y `actualizaCreencias` explicados en los ejemplos anteriores.
2. Enseguida el coordinador advierte la activación de una de las normas de su perfil: la norma `crearObjetivosCurso`, lo cual provoca la ejecución del método `cumpleNorma`. Véase la Figura 6.46.

```

ambienteescuela : EdoAmbiente
estado = {estadoCurso("listo")}

```

Figura 6.46: Contexto de la norma `crearObjetivosCurso`

3. El coordinador obtiene entonces, el plan que corresponde a la meta organizacional, el cual establece los temas que el profesor de matemáticas debe enseñar a sus alumnos (Figura 6.47).

```

planMatematicas : PlanParcialGlobal
submetas = {explicarTema("Polinomio"),
            explicarTema("Límite"),
            explicarTema("Derivada")}
participantes = {perfilProfMat}
contexto = {estadoCurso("listo")}
recursos = {salon}

```

Figura 6.47: Plan `planMatematicas`

4. Una vez que obtiene la información necesaria el coordinador utiliza el método `dirigeMeta`, el cual devuelve el conjunto de metas que debe alcanzar el profesor.
5. Se agregan las nuevas metas organizacionales dirigidas al profesor de matemáticas, las cuales indican al profesor los estándares de desempeño del curso. La Figura 6.48, indica que el profesor debe reportar el 100 % de alumnos aprobados. Con esto, se cumple con el cometido de la función de dirección del proceso administrativo.

```

enseñarPolinomios : MetaOrganizacional
meta = {porcentajeAprobados("Polinomios",100),1}
planmeta = planPolinomios

```

Figura 6.48: Meta `enseñarPolinomios`

Tales metas tienen un plan asociado cuya meta corresponde a una de las submetas de la instancia `planMatematicas` (Figura 6.47). La Figura 6.49 muestra el plan que el profesor de matemáticas debe seguir para alcanzar la meta `enseñarPolinomios`. Lo mismo ocurre para las submetas `explicarTema("Límite")` y `explicarTema("Derivada")`, ya que ambas producirán una meta organizacional nueva, con su respectivo plan asociado.

planPolinomios : PlanParcialGlobal
submetas = {explicarTema("Polinomios")}
participantes = {perfilProfMat}
contexto = {objetivosCreados("Matemáticas")}
recursos = {salon}

Figura 6.49: Plan planPolinomios

- Por último el coordinador actualiza el estado del ambiente, dado que ha alcanzado la meta de la norma, como lo muestra la Figura 6.50.

ambienteescuela : EdoAmbiente
estado = {estadoCurso("listo"), objetivosCreados("Matemáticas")}

Figura 6.50: Metas de la norma crearObjetivosCurso

La Figura 6.51 muestra gráficamente los seis pasos que acabamos de describir, los cuales corresponden a la fase de dirección del proceso administrativo.

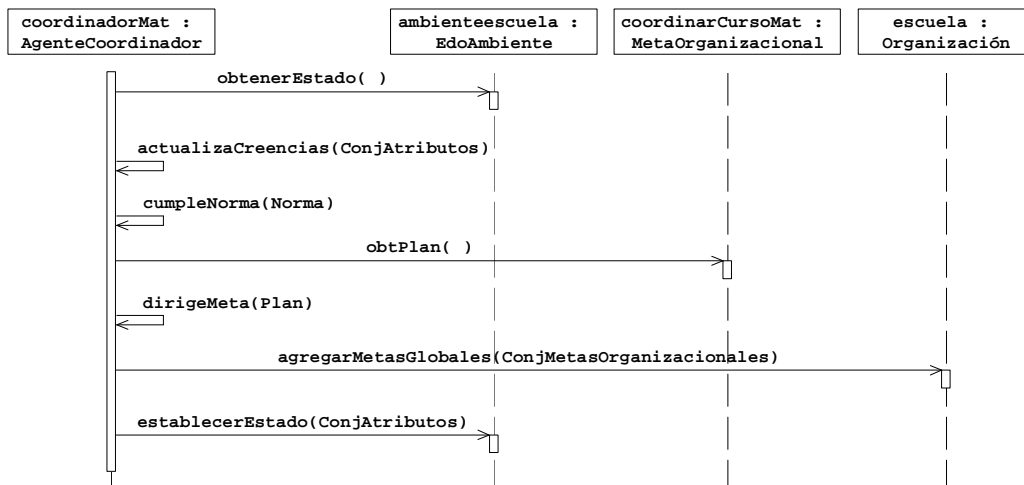


Figura 6.51: Secuencia Dirigir el curso

Control del curso

La misión de la función de control del proceso administrativo es verificar si las metas organizacionales están siendo alcanzadas. En caso de que las

metas no se alcancen o se alcancen parcialmente, se deberá efectuar una acción correctiva. Ejemplos de acciones correctivas son: la modificación de un plan, la generación de una nueva meta organizacional, o la creación de una nueva norma según sea el caso. En esta sección presentamos un ejemplo donde la acción correctiva es la creación de una nueva norma (Figura 6.52).



Figura 6.52: Secuencia Control del curso

1. El coordinador obtiene el estado del ambiente (`obtenerEstado`) y actualiza sus creencias (`actualizaCreencias`).
2. El coordinador ejecuta el método `cumpleNorma`, ya que el estado del ambiente coincide con el contexto de la norma `crearEstadisticasMat`, como lo muestra la Figura 6.53.
3. El coordinador obtiene las calificaciones de los estudiantes y obtiene la cantidad de alumnos aprobados en el primer parcial (Ver Figura 6.53). Cabe señalar que la calificación mínima aprobatoria es 7.

<u>ambienteescuela : EdoAmbiente</u>
<pre>estado = {califTemal("Jesus",5), califTemal("Oscar",7), califTemal("Arturo",6), califTemal("Ramon",5), califTemal("Dulce",6), estadoCurso("terminado")}</pre>

Figura 6.53: Contexto de la norma `crearEstadisticasMat`

- Posteriormente el coordinador obtiene las metas globales, ejecutando el método `obtMetasGlobales` y compara el estado del ambiente con las metas de la organización.
- El coordinador observa que la meta `enseñarPolinomios` (Figura 6.48) no se cumplió. Ya que el estado del ambiente refleja que sólo el 20 % de los alumnos aprobaron el examen de polinomios, como lo muestra la Figura 6.54.

<u>ambienteescuela : EdoAmbiente</u>
<pre>estado = {estadoCurso("terminado"), porcentajeAprobados("Polinomios",20)}</pre>

Figura 6.54: Insatisfacción de la meta `enseñarPolinomios`

- El coordinador decide entonces efectuar una acción correctiva y emitir una nueva norma dirigida al profesor de matemáticas. La norma se muestra en la Figura 6.55 e indica que el profesor debe explicar nuevamente el tema a fin de obtener un 60 % de alumnos aprobados en ese tema.

<u>repetirExamen : Norma</u>
<pre>beneficiarios = {} destinatarios = {profesorMat} metasnormativas = {explicarTema("Polinomios"), porcentajeAprobados("Polinomios",60)} recompensas = {} castigos = {hacerDescuento(profesorMat)} contexto = {estadoCurso("terminado"), porcentajeAprobados("Polinomios",20)} excepciones = {}</pre>

Figura 6.55: Norma `repetirExamen`

- La norma se agrega al conjunto de normas de la organización, a través del método `agregaNorma`.

8. El profesor recibe la notificación de la norma nueva, mediante el método `adoptaNorma`.
9. El profesor delibera si cumplir la norma (`deliberaNorma`) para posteriormente cumplirla (`cumplirNorma`).

El proceso de deliberación que ocurre para que el profesor decida *cumplir* o *no cumplir* la norma `repetirExamen`, se muestra en el diagrama de estados de la Figura 6.56.

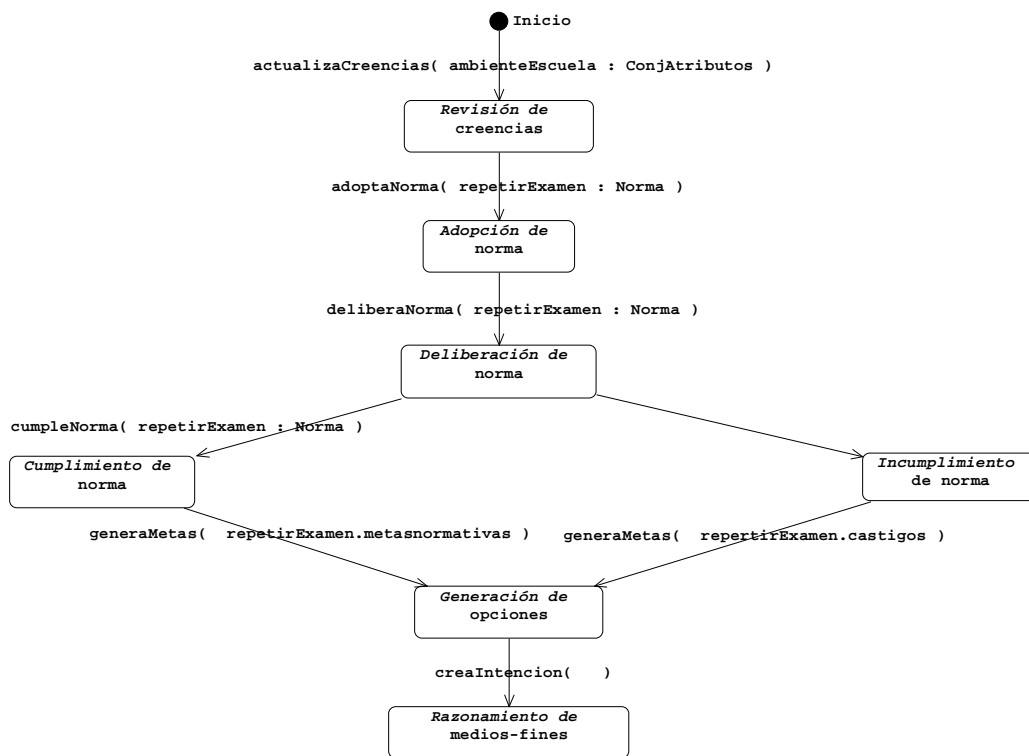


Figura 6.56: Proceso normativo

Primero, el profesor actualiza sus creencias, el resultado que arroja la actualización lo lleva a un estado de adopción de norma. El siguiente paso es deliberar si cumplir la norma `repetirExamen`. Enseguida pueden ocurrir dos situaciones: 1) El resultado de la deliberación lleva al agente a un estado de *cumplimiento* de norma, ó 2) El agente llega a un estado de *incumplimiento* de norma. En ambos casos, el siguiente paso es el de generar metas. En el

primer caso el agente crea su intención a partir de las metas normativas de la norma `repetirExamen`. En el segundo caso, el agente generará sus metas en base a los castigos asociados a la norma `repetirExamen`. La decisión de cumplir o no con la norma depende completamente de las metas individuales del agente del profesor de matemáticas. Por último, se elige la meta de mayor importancia para el agente (usando el método `creaIntención`), es decir, el agente creará un plan, para posteriormente ejecutar las acciones correspondientes.

Capítulo 7

Resultados y Conclusiones

A través del trabajo realizado en esta tesis hemos demostrado la viabilidad de automatizar y representar, no sólo los elementos que conforman una organización jerárquica, sino además, los procesos que en ella se llevan a cabo. Partimos del hecho, de que muchas de las actividades que se llevan a cabo en cualquier tipo de organización pueden ser automatizadas utilizando sistemas de software. Es a partir de este hecho, que el paradigma de agentes resulta ser adecuado, debido a que los elementos más importantes de cualquier organización son las personas que en ella laboran. Por lo tanto, por cada uno de los integrantes de la organización, podemos tener un agente que lo represente.

Por otro lado, el despliegue de la tecnología de Internet y el middleware disponible a la fecha, nos permiten implementar sistemas de software cada vez más complejos. Es por este motivo que podemos implementar sistemas multiagentes que representen sociedades abiertas y heterogéneas como las organizaciones jerárquicas que describimos en este trabajo. Donde, los miembros de la organización no se conocen de antemano y su comportamiento esta sujeto a sus metas individuales, por lo cual, dichos agentes pueden abandonar e ingresar a la organización según les convenga. Es este comportamiento individual el motivo por el cual, nuestro modelo incluye un mecanismo de control social como son las normas.

7.1. Resultados Obtenidos

A lo largo de este documento hemos analizado y descrito varias teorías sociales, como las de la administración clásica, neoclásica, de relaciones humanas [3, 4, 5] y de selección de personal [10], las cuales brindan el soporte teórico que sustenta nuestro modelo. Por otro lado, utilizamos un Sistema Multiagentes Normativo [13] como base de nuestro modelo de organización basado en agentes. El resultado que produce esta mezcla ha sido representado utilizando UML como lenguaje de modelado. UML provee una representación gráfica, fácil de comprender y ampliamente usada, facilitando así, a diseñadores y programadores, el proceso de implementación del sistema. Esta tesis no se limita a presentar un modelo que describa los elementos de una organización jerárquica, propone además, una guía de desarrollo que ejemplifica la implementación del modelo.

7.1.1. Un modelo para organizaciones jerárquicas

Desde el punto de vista estructural nuestro modelo se centra en definir, describir y representar, los elementos de una organización jerárquica y las relaciones que existen entre ellos. A continuación se da una breve descripción de cada uno de los elementos que conforman HIERARCHY.

- Modelamos un *agente organizacional* a partir de un agente autónomo, al cual se le agrega la habilidad de reconocer las normas de la sociedad a la que pertenece. Éste a su vez, puede adquirir un rol normativo [13] (legislador o defensor). A este agente normativo se le añaden características organizacionales, las cuales le permiten adoptar uno o varios perfiles de puesto, utilizar recursos, prestar servicios y alcanzar metas organizacionales. Un agente de este tipo puede además, adoptar roles organizacionales (trabajador, supervisor o administrador).
- Las *metas* de la organización constan de un conjunto de estados deseados, representados a su vez como atributos, los cuales representan situaciones a las que la organización quiere llegar. Estas metas tienen asociadas un plan parcial global.
- Los *servicios* que proporciona la organización, al igual que las metas, se representan como un conjunto de estados deseables que benefician las metas de la organización, y proporcionan a su vez, un resultado de

valor para los usuarios de la organización. Los servicios tienen también un plan parcial asociado, una descripción y un identificador.

- Un *contrato* es usado para garantizar el servicio que se va a brindar. Éste, consta del identificador del servicio y del identificador del cliente, el tipo de contrato, un conjunto de cláusulas, y el objetivo del contrato.
- Las *normas* que regulan la conducta de los agentes organizacionales son equivalentes a los reglamentos que rigen, de modo general, la conducta de los integrantes de una organización humana. Una norma está compuesta por un conjunto de beneficiarios y uno de destinatarios, un conjunto de metas, un conjunto de recompensas, un conjunto de castigos, el contexto bajo el cual la norma se aplica y un conjunto de excepciones.
- Las *funciones del proceso administrativo* son modeladas como métodos que ejecuta un agente dependiendo del rol que desempeñe. Las funciones de planeación y organización son implementadas por el agente que juegue el rol de administrador. El agente supervisor implementa las funciones de dirección y control. A continuación definimos los roles y explicamos en que consisten dichas funciones.
- Definimos tres *roles* para los agentes que forman la organización.
 1. *Administrador*. El agente que desempeña este rol tiene la habilidad de definir los objetivos de los servicios y las metas de la organización (*planeación*). Posee también la habilidad de crear un plan, el cual asigna metas y recursos a cada uno de sus subordinados (*organización*).
 2. *Supervisor*. La misión de un agente supervisor es la de establecer los estándares de desempeño, para las actividades que realizan sus subordinados (*dirección*). Así como también, la habilidad de ejecutar una acción correctiva en caso de no cumplir con los estándares, o brindar algún tipo de recompensa en caso de sobrepasarlos, según sea el caso (*control*). Podemos identificar varios tipos de acciones correctivas y de recompensa, por ejemplo, la creación o modificación de: un plan, una nueva meta global, una norma (de refuerzo de recompensa) o de algún estado del ambiente.
 3. *Trabajador*. Los agentes que ocupan el rol de trabajador poseen la habilidad de reconocer la autoridad de sus superiores, deben tam-

bién poseer alguna habilidad específica y acceso a algún recurso que les permita contribuir con las metas de la organización o con la ejecución de algún servicio.

- Los *perfiles de puesto*, constan de un identificador para el puesto, establecen además las obligaciones y los derechos que un agente adquiere al aceptar ocupar un puesto en una organización. Ambos, obligaciones y derechos, se modelan como un conjunto de normas. El perfil de puesto describe además el conjunto de superiores y de subordinados que un agente tiene al ocupar un puesto en la organización. Dichos conjuntos describen las líneas de autoridad y de comunicación de la organización.
- Un *plan* esta formado por un conjunto de submetas, el conjunto de perfiles de puesto encargado de alcanzar esas submetas y los recursos necesarios para alcanzarlas. Un plan es utilizado para coordinar las actividades cooperativas de los agentes. Los planes, en unión con las normas, modelan y sincronizan la interdependencia entre actividades.
- Un *recurso* es representado mediante su nombre, tipo, ubicación y disponibilidad.

7.1.2. Una guía de desarrollo

Además de los elementos que acabamos de describir, esta tesis desarrolla un ejemplo que propone una guía para la implementación del modelo, la cual utiliza y combina el análisis de puestos [10] con el Proceso Unificado de Desarrollo de Software [20]. Este proceso se puede describir en las siguientes fases:

1. Elaboración de un *modelo de casos de uso*. El proceso inicia obteniendo los casos de uso más relevantes para el sistema, los cuales describen a grandes rasgos la funcionalidad de la organización que se desea representar. Estos casos de uso definen los servicios que la organización proporciona a sus usuarios. En nuestro ejemplo obtuvimos tres casos de uso: inscribirse, tomar y abandonar el curso.
2. Creación de *perfiles de puesto* y *agentes organizacionales*. En este paso se parte de un formato que consta de dos partes, las cuales describen las actividades que un individuo realiza en una organización.

Parte I Es a partir de esta parte del formato que podemos crear las *instancias de los perfiles de puesto* de nuestro modelo. En el ejemplo desarrollado creamos los perfiles de puesto para el director, el prefecto, los coordinadores y los profesores de la escuela.

Parte II Esta parte nos permite *implementar el agente que reúne las características que define un perfil*. Es aquí donde modelamos el agente capaz de cumplir con el perfil de puesto creado en el paso anterior. Con este fin se realiza un análisis a partir de los conocimientos, habilidades y capacidades que debe poseer el agente, así como los recursos que debe tener a su disposición.

- Los *conocimientos* nos permiten crear los métodos que definen las acciones específicas que un agente debe poder realizar.
- Las *habilidades* describen el rol normativo que debe implementar el agente.
- Las *capacidades* definen el rol organizacional que debe desempeñar el agente.
- Las *condiciones de trabajo* establecen los recursos a los que el agente tiene acceso.

3. Una vez creados los perfiles de puesto, podemos crear las *normas* de los perfiles de puesto y las normas del reglamento de la organización. En nuestro ejemplo, explicamos las obligaciones descritas en los perfiles de puesto del director y de los coordinadores. La única norma del reglamento que modelamos es la de asistir al trabajo¹.
4. La información reunida hasta este momento nos permite modelar las instancias de los *recursos* establecidos en los perfiles de puesto.
5. Los elementos descritos en los pasos anteriores conforman la estructura estática de la organización. El siguiente paso es *describir el comportamiento* esperado de la organización y las colaboraciones que implementan los casos de uso. En el ejemplo elaborado en esta tesis, representamos las colaboraciones entre objetos como *diagramas de secuencia*, lo cual no es una restricción, ya que el diseñador puede elegir *diagramas de colaboración* si lo cree conveniente.

¹La norma `asistirTrabajo` es representada en los perfiles de puesto pero la instancia de esta norma no se elabora en el ejemplo del Capítulo 6

6. Una vez especificadas las colaboraciones y representadas como diagramas UML de secuencia o colaboración, procedemos a crear las instancias para los *planes parciales* y las *metas organizacionales* que surgen a partir de las submetas creadas en los planes, así como las instancias de los *servicios*, y sus respectivos *planes* y *contratos*.

El análisis hecho en esta tesis permite representar los elementos de una organización jerárquica. De este modo, proporcionamos un medio para automatizar las actividades que en ella se realizan. Además, proponemos un proceso que facilita la automatización de dichas tareas. Este proceso nos proporciona una guía de acción que nos permite implementar con mayor facilidad el modelo. Ambas actividades se dirigen a reducir el trabajo realizado por las personas que integran una organización jerárquica, ya que podemos implementar un agente que represente a cada individuo en la organización.

7.2. Análisis Comparativo

Existen otros trabajos relacionados con el nuestro que utilizan conceptos organizacionales similares a los utilizados en esta tesis. Entre los cuales destacamos el modelo MOISE de Hannoun et al. descrito en [11] y el meta-modelo descrito en [8] propuesto por Ferber y Gutknecht denominado AALAADIN.

7.2.1. El modelo MOISE

Hannoun et al. definen una organización como un conjunto de reglas normativas que restringen el comportamiento de los agentes. Cuando un agente entra en una organización de este tipo se vuelve parte de una red de obligaciones, prohibiciones y permisos [11]. En MOISE se elabora una especificación para agentes deliberativos capaces de razonar con metas, planes, acciones y recursos. El modelo MOISE esta estructurado en tres niveles.

- El *nivel individual* define las tareas que son responsabilidad de un agente. En el nivel individual los roles especifican los tipos de actividades que se le permiten a un agente.
- El *nivel de agregado* es el encargado de añadir agentes en estructuras grandes. En el nivel de agregado los grupos restringen el conjunto de agentes con los que se puede cooperar.

- El *nivel de sociedad* se refiere a la estructura global y la interconexión entre agentes y estructuras. En el nivel social, los enlaces limitan los tipos de interacción que un agente puede tener. Estos enlaces definen las estructuras de comunicación y control entre agentes y los grupos en el sistema

El nivel individual

En [11] se define una *tarea* como un proceso que involucra muchos agentes. Cada agente es responsable de una parte de la tarea, la cual recibe el nombre de *misión*. Una misión es un conjunto de metas prohibidas o autorizadas, un conjunto de planes a seguir, un conjunto de acciones a ejecutar y los recursos a usar.

Un *rol* se define como un conjunto de misiones. Un agente jugando un rol dentro de este tipo de organización *debe obedecer* los comportamientos permitidos en las misiones que constituyen el rol [11]. Esto último restringe fuertemente el comportamiento del sistema. Sin embargo, el modelo es flexible, en el sentido de que es posible definir roles que permitan adoptar cualquier meta pero que prohíban planear, o roles que permitan construir o ejecutar planes que correspondan únicamente a la meta que se delega al agente, por mencionar algunos ejemplos.

El nivel de agregado

Un *enlace organizacional* es un arco orientado entre un rol origen y un rol destino, el cual es etiquetado con un tipo y posee un conjunto de restricciones. Las restricciones dependen del tipo de enlace. Por último, un enlace organizacional define los conjuntos de misiones válidas, entre los roles origen y destino. El tipo de enlace puede tomar tres valores: *comunicación*, *autoridad*, y de *modelo de conocidos*. El conjunto de misiones establece el contexto bajo el cual ambos roles pueden usar el enlace [11].

Un *grupo* se encuentra formado por un conjunto de roles, un conjunto de misiones pertenecientes a esos roles que pueden activarse en el contexto del grupo, un conjunto de enlaces, el cual es un subconjunto de los enlaces entre los roles fuente y destino del grupo [11]. Existen dos tipos de enlaces en MOISE los *extra-grupo* y los *intra-grupo*. Los enlaces extra-grupo ligan un rol que pertenece al grupo de roles del grupo con uno que no pertenece. A diferencia de los intra-grupo, donde ambos roles pertenecen al mismo grupo.

El nivel de sociedad

Para Hannoun et al. [11], una *estructura organizacional* se define mediante un conjunto de roles, un conjunto de enlaces (de comunicación, de autoridad y de modelos de conocidos) y un conjunto de grupos. En [11] se hace una diferencia explícita entre una estructura organizacional y una entidad organizacional. Una *entidad organizacional* está definida, como un conjunto de pares que corresponden a un agente jugando un rol en la estructura organizacional, un conjunto de grupos instanciados. Un grupo en este conjunto está formado por un conjunto de agentes cuyos roles pertenecen a ese grupo. Y un conjunto de enlaces instanciados en esos agentes.

El cumplimiento de una tarea usando MOISE es un proceso que involucra la activación de las misiones correspondientes. Según Hannoun et al. [11] por activación se entiende la decisión de un agente de cumplir la tarea que especifica la misión y la posterior ejecución de las acciones respectivas. Lo anterior implica que el agente depende de otras tareas. Por lo tanto, el agente requerirá y negociará la activación de las misiones dependientes de otros. Esto implica, que el agente tiene la habilidad de reconocer las relaciones de dependencia considerando los roles, para lo cual en [11] se define una red de dependencia.

7.2.2. El modelo AALAADIN

En el trabajo hecho por Ferber y Gutknecht [8] se describe un meta-modelo, llamado AALAADIN, para construir organizaciones jerárquicas y mercados. El modelo describe dos niveles de análisis.

- El *nivel concreto* corresponde a los conceptos de agente grupo y rol, lo cual describe la organización en sí.
- El *nivel abstracto metodológico* define todos los roles, interacciones válidas y las estructuras de los grupos en la organización.

El nivel concreto

Un *agente* se define como una entidad de comunicación activa la cual desempeña roles dentro de los grupos. El diseñador es responsable de elegir el modelo de agente mas apropiado.

Un *grupo* se define como un conjunto de agentes. Un agente puede ser miembro de N grupos al mismo tiempo. Un grupo puede ser encontrado por cualquier agente. Un agente debe solicitar su admisión a un grupo existente. Los grupos deben estar distribuidos en varias máquinas.

Un *rol* es la representación abstracta de la función, servicio o identificación de un agente dentro de un grupo. Cada agente puede manejar varios roles y cada rol manejado por un agente es local para un grupo. Al igual que sucede con la admisión a un grupo, un agente candidato debe solicitar desempeñar un rol, aunque no siempre se le concede.

Un rol especial en un grupo es el de *manejador*, el cual es automáticamente asignado al creador del grupo. Este agente tiene la responsabilidad de manejar las solicitudes de admisión al grupo y las solicitudes de asignación de rol. También puede revocar roles y membresías.

Las solicitudes de admisión a un grupo para jugar un rol son evaluadas por una función booleana que devuelve verdadero si el agente es aceptado. En [8] se ilustran cinco posibles funciones que controlan la asignación de un rol dentro de un grupo.

1. *Aceptación o rechazo sistemáticos.* Un grupo local acepta o rechaza siempre a cualquier agente.
2. *Aceptación condicionada por competencia.* En este esquema un rol se asigna a un agente dependiendo de sus habilidades.
3. *Restringida por la implementación.* El agente candidato debe presentar alguna interfaz para poder unirse a un grupo.
4. *Condicionada a un diálogo de admisión.* La solicitud induce una interacción entre el manejador y el candidato para negociar la admisión.
5. *Restringida por el estatus del grupo.* Es posible, por ejemplo, definir un coeficiente de similitud entre los agentes del grupo y el agente candidato.

El nivel abstracto

Además de los conceptos descritos, en AALAADIN se añaden los conceptos de estructura de grupo y estructura organizacional para complementar el modelo.

La *estructura de grupo* es una descripción abstracta de un grupo. Identifica todos los roles e interacciones que pueden aparecer en un grupo. Una estructura de grupo está compuesta por un conjunto de identificadores de rol (donde se enumeran todos los roles posibles que los agentes pueden jugar en un grupo), un grafo de interacción (el cual identifica las interacciones válidas entre roles), y un lenguaje de interacción (para cada relación dentro de un grupo se asocia un protocolo de definición único).

En [8] se define una *estructura organizacional* como el conjunto de estructuras de grupo. Ésta, expresa el diseño de la organización multiagentes. Formalmente una estructura organizacional está formada por un conjunto de estructuras de grupo válidas y un grafo dirigido que especifica las relaciones reales entre roles y grupos. Del mismo modo que sucede con un grupo y la estructura de un grupo la organización verdadera es sólo una posible manifestación de la estructura organizacional.

Ferber y Gutknecht introducen tres mecanismos más para complementar su modelo. El mecanismo de *membresía cruzada*, permite a un agente pertenecer a un grupo relacionado y a un grupo de meta-nivel el cual manejará las actividades a nivel de sistema, tales como el manejo de recursos, el paso de mensajes remotos, la seguridad dinámica, la migración de agentes, etc. El mecanismo de *agentificación de servicios*, permite que los servicios a nivel de sistema se representen como agentes jugando roles en grupos de meta-nivel. El mecanismo de *uso de representante* permite que un agente pueda ser el representante de un grupo A en un grupo B, lo cual permite al agente delegar tareas a agentes del grupo B.

Para demostrar la efectividad de su modelo, Ferber y Gutknecht utilizan una plataforma llamada MADKIT, la cual añade tres principios de diseño. Una *arquitectura de micro-kernel*, la cual proporciona el conjunto mínimo de facilidades que permiten el desarrollo de los servicios de los agentes. La *agentificación de servicios* implementa los métodos encargados del manejo de roles y grupos, lo cual permite que los agentes trabajen transparentemente de manera distribuida, sin necesidad de modificar su código. El *modo gráfico* de la plataforma permite al agente ser independiente en su interfaz, la cual puede ser extremadamente sencilla o compleja.

7.2.3. Comparativos

Nuestro análisis comparativo inicia con un cuadro que enlista los elementos que componen nuestro modelo y su comparación con los modelos MOISE

y AALAADIN descritos en [11] y [8] respectivamente.

HIERARCHY	MOISE	AALAADIN
Agente Organizacional	Agente deliberativo capaz de razonar con metas, planes, acciones y recursos	Un agente es una entidad activa capaz de desempeñar roles dentro de grupos
Meta organizacional	Misión	No se modela
Servicio organizacional	No se modela	No se modela
Contrato	No se modela	No se modela
Norma	Conjunto de comportamientos permitidos o prohibidos en una misión	No se modela
Funciones del proceso administrativo	La activación de misiones es un proceso similar	No se modela
Roles	Es equivalente a un perfil de puesto en nuestro modelo	Es equivalente a un perfil de puesto en nuestro modelo
Perfil de puesto	Rol	Rol
Plan	Plan	No se modela
Recurso	Se menciona su uso pero no se modela	No se modela

Cuadro 7.1: Equivalencias entre MOISE, AALAADIN y HIERARCHY

El Cuadro 7.1 inicia describiendo la definición de agente que se da en los otros trabajos, en ninguno de los dos se dan detalles sobre la implementación del agente, y sólo se explican sus características. El término misión que se da en [11] es el más aproximado al término meta organizacional de nuestro modelo, ya que una misión esta compuesta por un conjunto de metas, un conjunto de planes un conjunto de recursos y un conjunto de acciones asociadas a un rol. AALAADIN utiliza el término agentificación de servicios, pero se refiere a un conjunto de agentes jugando un rol en un grupo específico y no a la descripción del servicio que se da en esta tesis. Una misión en MOISE describe el conjunto de acciones, metas y planes permitidos a un agente, lo cual es lo más parecido a una norma en nuestro modelo. Una diferencia es la membresía de un agente en una organización. En AALAADIN existen 5 formas de ocupar un rol, las cuales no dependen únicamente del agente solicitante, mientras que en HIERARCHY basta con que el agente acepte las normas del perfil de puesto. Es importante señalar que HIERARCHY no define un rol en-

cargado de la membresía de los miembros de la organización, ni los métodos que soporten dicha actividad. La activación de misiones es un proceso similar al de las funciones del proceso administrativo, ya que coordina las acciones conjuntas dentro de una organización. Los roles descritos en ambos modelos son equivalentes a un perfil de puesto en HIERARCHY. La diferencia entre rol y perfil de puesto, en nuestro modelo, es que el rol determina cuales son las funciones del proceso administrativo que realizará un agente, mientras que un perfil de puesto determina las actividades o acciones específicas que realiza un agente dentro de una organización. Los planes se modelan igual en MOISE que en nuestro modelo. En MOISE se menciona los recursos como una parte del modelo pero no se describen en ninguna parte en [11].

Mientras que en MOISE y AALAADIN se definen niveles de análisis, en HIERARCHY no se especifica un nivel de análisis como tal, pero aún así, podemos describir los elementos que corresponden a tales clasificaciones. El Cuadro 7.2 muestra, en la primera columna, los niveles de análisis que Hannon et al. especifican en MOISE, en la segunda columna se enlistan los elementos que componen ese nivel y en la tercera los elementos que poseen similitud con tales conceptos en HIERARCHY. Un análisis similar se hace en el Cuadro 7.3, pero ahora con los niveles y los elementos que describen Ferber y Gutknecht en AALAADIN [8].

Nivel	MOISE	HIERARCHY
Individual	Agente deliberativo capaz de razonar con metas, planes, acciones y recursos	Agente normativo con capacidades administrativas y de supervisión
	Mision	Meta organizacional
	Metas prohibidas y autorizadas	Obligaciones y derechos de un perfil de puesto
	Planes y recursos	Planes y recursos
	Rol	Perfil de puesto
De agregado	Enlace organizacional y grupo	Superiores y subordinados de un perfil de puesto
De sociedad	Estructura y entidad organizacional	Clase e instancia de una organización

Cuadro 7.2: Equivalencias entre MOISE y HIERARCHY

Nivel	AALAADIN	HIERARCHY
Concreto	Un agente es una entidad activa capaz de desempeñar roles dentro de grupos	Un agente es capaz de desempeñar un rol y un perfil de puesto para prestar servicios y contribuir con las metas de la organización
	Grupo	Unión de un agente y sus subordinados
	Rol	Perfil de puesto
	Rol de administrador	No existe un rol semejante en nuestro modelo
	Cinco tipos de admisión y asignación de rol	Adopción de las normas de la organización y de las normas de al menos un perfil de puesto
Abstracto	Estructura y entidad organizacional	Clase e instancia de una organización

Cuadro 7.3: Equivalencias entre AALAADIN y HIERARCHY

Una de las ventajas que tiene nuestro modelo sobre los que hemos descrito en esta sección, es que HIERARCHY proporciona una guía de desarrollo, lo cual facilita su análisis e implementación a nivel estructural y de comportamiento. Esta situación, reduce en gran medida las actividades de programadores y desarrolladores de sistemas de software. Una desventaja aparente en nuestro modelo es que se trata de un modelo más reciente. Por lo tanto, no se ha tenido la posibilidad de implementarlo para presentar resultados como las aplicaciones que se describen en [11] y [8]. En [8], se cuenta además, con una plataforma que brinda facilidades para la implementación del modelo. AALAADIN también describe, mecanismos de admisión a la organización, así como un rol encargado de las tareas de admisión y expulsión de elementos. Todas estas situaciones se encuentran fuera del alcance de la tesis, pero forman parte de nuestra propuesta de investigación en el futuro.

7.3. Trabajo Futuro

Considerando el *proceso de desarrollo* propuesto en este trabajo, el cual guía, en buena parte, la implementación del modelo, proponemos como futuro trabajo de investigación la creación de una metodología.

Existen metodologías para la creación de sistemas multiagentes que utilizan conceptos organizacionales como MaSE [7] o Gaia [22], las cuales resumimos brevemente a continuación.

- La *metodología MaSE* (Multiagent Systems Engineering, Ingeniería en Sistemas Mutliagentes) propuesta por DeLoach et al. en [7], proporciona guías para el desarrollo de SMAs que cubren las fases de análisis y diseño. MaSE sirve para crear sistemas estáticos y cerrados. Los agentes de estos sistemas son también estáticos y siempre están dispuestos a cooperar.
- La *metodología Gaia* desarrollada por Zambonelli et al. en [22] proporciona guías claras para el análisis y diseño de sistemas complejos y abiertos. Se asume que las organizaciones a desarrollar pueden ser abiertas o cerradas, dinámicas o estáticas y los agentes pueden ser bevolentes o interesados en sí mismos.

A pesar de los esfuerzos por desarrollar metodologías que utilicen conceptos organizacionales que nos permitan modelar y coordinar sistemas multiagentes como organizaciones, no existe una metodología para la implementación de organizaciones jerárquicas abiertas, heterogéneas y dinámicas, basadas en sistemas de agentes normativos como las que describimos en el modelo. La elaboración de esta metodología nos permitirá la automatización de las actividades de una organización por medio de nuestro modelo de una forma más sencilla y confiable, reduciendo así, el trabajo realizado y el tiempo invertido por los diseñadores y programadores de sistemas.

Apéndice A

Metodología y Lenguaje de Modelado

En este Capítulo presentamos una explicación breve de la metodología y el lenguaje de modelado utilizados para la elaboración de esta tesis. La organización de este Capítulo es la siguiente: en la Sección A.1 explicamos a grandes rasgos el Proceso Unificado de Desarrollo de Software, la Sección A.2 da una breve descripción de el Lenguaje Unificado de Modelado.

A.1. El Proceso Unificado de Desarrollo de Software

El Proceso Unificado es un proceso de desarrollo de software. De acuerdo con Rumbaugh, Jacobson y Booch [20], un *proceso de desarrollo de software* es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. El Proceso Unificado es más que un simple proceso. El proceso Unificado de Desarrollo de Software, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

A.1.1. Características del proceso

El proceso Unificado utiliza el Lenguaje Unificado de Modelado para representar todos los esquemas de un sistema de software. UML es parte esencial del Proceso Unificado, ya que sus desarrollos fueron paralelos. Los aspectos que definen al Proceso Unificado se resumen en tres frases [20].

- Dirigido por casos de uso.
- Centrado en la arquitectura .
- Iterativo e incremental.

Dirigido por casos de uso

Un sistema de software se crea para dar servicio a sus usuarios. Por lo tanto, es necesario saber lo que los futuros usuarios del sistema necesitan y desean. El término *usuario* representa a alguien o algo (como otro sistema fuera del sistema que se está construyendo) que interactúa con el sistema que se está desarrollando.

Una *interacción* de un usuario con el sistema genera un caso de uso. Un *caso de uso* es un fragmento de la funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los *requisitos funcionales* del sistema.

De acuerdo con [20], una especificación funcional contesta la pregunta ¿Qué debe hacer el sistema?. Los casos de uso añaden tres palabras a la pregunta: ¿... *para cada usuario?*. Los casos de uso *guían el proceso de desarrollo*, aunque no se desarrollan aisladamente, al contrario, éstos se desarrollan al mismo tiempo que la arquitectura. La arquitectura como los casos de uso maduran conforme avanza el ciclo de desarrollo.

Centrado en la arquitectura

La arquitectura de un sistema de software se describe mediante diferentes *vistas del sistema* en construcción. La arquitectura incluye los aspectos *estáticos y dinámicos* más significativos del sistema. La arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando de lado los detalles [20].

Debe haber interacción entre los caso de uso y la arquitectura [20]. Por un lado, los casos de uso deben encajar con la arquitectura cuando se crean

y por otro lado, la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos.

Iterativo e incremental

El desarrollo de un producto de software supone un esfuerzo que puede durar mucho tiempo. Es práctico dividir el trabajo en partes pequeñas o miniproyectos. Cada miniproyecto es una *iteración* que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos al crecimiento del producto.

En [20] se establece que, cada iteración genera la entrega interna o externa de varios *artefactos* que juntos constituyen una *línea base*. Un artefacto es una pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar sus actividades. Un artefacto puede ser un modelo, un elemento de un modelo o un documento. Una línea base es un conjunto de artefactos revisados y aprobados, los cuales proveen una base para las revisiones y desarrollo posteriores. Un *incremento* es la diferencia entre la entrega de una iteración y la entrega de la siguiente. El resultado de una iteración es un *incremento*. Se llega a la entrega del producto final a través de una secuencia de iteraciones.

A.1.2. El producto

Cada ciclo produce una nueva versión del sistema, y cada versión es un producto preparado para su entrega. El producto terminado no sólo debe ajustarse a las necesidades del usuario, sino también a las necesidades de toda la gente que trabajará con el producto. Para llevar a bien la ejecución del ciclo siguiente, los desarrolladores necesitan todas las representaciones del producto de software [20].

- Un *modelo de casos de uso*, con todos los casos de uso y su relación con los usuarios.
- Un *modelo de análisis*, el cual tiene dos propósitos: refinar los casos de uso y hacer una asignación inicial de la funcionalidad del sistema a un conjunto de objetos.
- Un *modelo de diseño* que define: la estructura estática del sistema en forma de subsistemas, clases e interfaces y los casos de uso reflejados como colaboraciones entre los subsistemas, clases e interfaces.

- Un *modelo de implementación*, que incluye componentes, los cuales representan al código fuente y la correspondencia de las clases con los componentes.
- Un *modelo de despliegue*, también llamado *modelo de distribución*, que define los nodos físicos (computadoras) y la correspondencia de los componentes con esos nodos.
- Un *modelo de prueba* que describe los casos de prueba que verifican los casos de uso
- Y por supuesto, una *representación de la arquitectura*.

Todos estos modelos están relacionados. Juntos representan al sistema como un todo. Los elementos de un modelo poseen *dependencias de traza*, mediante enlaces a otros modelos. Una *traza* es una dependencia que indica una relación histórica o de proceso entre dos elementos que representan el mismo concepto, sin reglas específicas para derivar una de otra [20].

A.1.3. Vida del proceso

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una *versión* del sistema para los clientes [20]. Una versión es un conjunto de artefactos relativamente completo y consistente, el cual se entrega a un usuario externo o interno. Cada ciclo consta de cuatro *fases*: inicio, elaboración, construcción y transición.

Fases del Proceso Unificado

Un proyecto que utiliza el Proceso Unificado se divide en cuatro fases principalmente.

- *Inicio*. Establece los objetivos del ciclo de vida para el proyecto. También se establece lo que deberá hacer el producto.
- *Elaboración*. Obtiene la línea base de la arquitectura, captura la mayoría de los requisitos y establece la arquitectura del ciclo de vida.
- *Construcción*. Desarrolla el sistema entero y garantiza que el sistema puede comenzar su transición a los clientes, en otras palabras, el sistema obtiene una funcionalidad operativa inicial.

- *Transición.* Garantiza la obtención de un producto preparado para entregarse a la comunidad de usuarios. En esta fase se enseña a los usuarios a usar el software.

Flujos de procesos

Las iteraciones contienen flujos de procesos. El proceso unificado describe cinco flujos de procesos base [20].

- *Requisitos.* Su propósito es guiar el desarrollo hacia un sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema lo suficientemente buena como para obtener un acuerdo entre el cliente y los desarrolladores sobre qué debe hacer el sistema.
- *Análisis.* Este flujo de basa en el modelo de análisis. Un modelo de análisis ofrece una especificación más precisa de los requisitos que obtenemos de la captura de requisitos. Este modelo se describe utilizando el lenguaje de los desarrolladores, produciendo un mayor formalismo, lo cual permite razonar sobre el funcionamiento del sistema.
- *Diseño.* En el diseño modelamos el sistema y encontramos su arquitectura, de modo que soporte todo los requisitos. Los propósitos del diseño son: adquirir una comprensión de los requisitos no funcionales y de las restricciones relacionadas con los lenguajes de programación, sistemas operativos, etc. y crear un punto de partida para las actividades de la implementación capturando los subsistemas individuales interfases y clases.
- *Implementación.* Los propósitos de la implementación son: implementar las clases y subsistemas encontrados durante el diseño, las clases se implementan como componentes de fichero que contienen código fuente; probar los componentes individualmente y posteriormente integrarlos, compilándolos y enlazándolos en uno o más ejecutables, antes de hacer las comprobaciones del sistema del siguiente flujo.
- *Prueba.* Los objetivos de la prueba son: planificar las pruebas necesarias en cada iteración, incluyendo las pruebas de integración y las pruebas de sistema, las pruebas de itegración son necesarias para cada construcción, mientras que las pruebas de sistema sólo son necesarias al final

de la iteración, una construcción es una versión ejecutable del sistema, por lo general, una parte específica del mismo; diseñar e implementar las pruebas creando los casos de prueba que especifican qué probar; realizar las diferentes pruebas y manejar los resultados de cada prueba sistemáticamente, las construcciones que presentan defectos se prueban nuevamente y tal vez se devuelvan a otro flujo de trabajo como el diseño o la implementación.

Cada iteración puede contener todos los flujos de procesos pero con diferente énfasis dependiendo en que iteración del ciclo de vida nos encontremos. Como hemos visto, el ciclo de vida del Proceso Unificado se divide en una secuencia de fases. Cada fase puede incluir muchas iteraciones. El número de iteraciones depende del tamaño del proyecto.

A.2. El Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual de propósito general que es usado para la especificación, visualización, construcción y documentación de artefactos de un sistema de software [19].

UML permite modelar sistemas de información, y su objetivo es lograr *modelos* que además de describir con cierto grado de formalismo tales sistemas, puedan ser *entendidos por los clientes o usuarios* de aquello que se modela.

A.2.1. Características de UML

UML captura decisiones y el conocimiento sobre los sistemas que se deben construir. Pretende ser usado con todos los métodos de desarrollo, en todas las etapas del ciclo de vida y en todos los dominios de aplicación. UML incluye conceptos semánticos, de notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizacionales. Además cuenta con el *soporte de herramientas de modelado visuales* que tienen generadores de código y de reportes. UML intenta ser un lenguaje de modelado de propósito general universal para sistemas discretos, tales como aquellos hechos de software, firmware o lógica digital.

La especificación de UML *no define un proceso estándar* pero pretende ser útil en un proceso de desarrollo iterativo. UML *no es un lenguaje de*

programación. UML no es un lenguaje altamente formal creado para probar teoremas. Hay una gran número de tales lenguajes, pero nos son fáciles de entender o usar para muchos propósitos.

Modelar un sistema desde varios puntos de vista separados pero relacionados nos permite entenderlo para diferentes propósitos [19]. Ahora bien, sabemos que un sistema es modelado como una colección de objetos discretos que interactúan para realizar algún trabajo que beneficia a un usuario externo. La *estructura estática* de UML es usada para definir los tipos de objetos importantes para un sistema y su implementación, así como las relaciones entre los objetos. Por otro lado, el *comportamiento dinámico* define la historia de los objetos a través del tiempo y la comunicación que existe entre los objetos para cumplir con una meta.

A.2.2. Vistas de UML

No hay una línea entre los diversos conceptos y construcciones de UML, pero, por conveniencia, lo dividimos en varias vistas. Una *vista* es simplemente un subconjunto de construcciones de modelos UML que representan un aspecto del sistema. La división en diferentes vistas es arbitraria, pero intuitiva [19].

De acuerdo al trabajo hecho en [19], en el nivel más superior las vistas se dividen en tres áreas: *clasificación estructural*, *comportamiento dinámico* y *gestión de modelo*.

- *Clasificación estructural*. Describe las cosas en el sistema y sus relaciones con otras cosas. Los clasificadores incluyen clases, casos de uso, componentes y nodos. Esta clasificación incluye: la vista *estática*, la vista de *casos de uso* y la vista de *implementación*.
- *Comportamiento dinámico*. Describe el comportamiento de un sistema a través del tiempo. El comportamiento puede describirse como una serie de cambios en una foto instantánea tomada desde la vista estática. El comportamiento dinámico incluye: la vista de *máquinas de estado*, la vista de *actividad* y la vista de *interacción*.
- *Gestión de modelo*. Describe la organización del modelo mismo dentro de unidades jerárquicas. El paquete es la unidad genérica de organización para los modelos. La vista de gestión de modelo, cruza las otras

vistas y las organiza para el trabajo de desarrollo y el control de la configuración.

UML también contiene varias construcciones que intentan proveer una capacidad de extensibilidad limitada pero útil. Estas construcciones incluyen restricciones, estereotipos y valores etiquetados. Estas construcciones son aplicables a los elementos de todas las vistas [19].

La tabla A.1 muestra las vistas de UML y los diagramas que las muestran, así como los conceptos relevantes de cada vista. La tabla no se debe tomar como un reglamento estricto, es simplemente *una guía* del uso común que se da a las vistas, dado que es permitido mezclar las vistas [19]. Debido a que el objetivo de este trabajo es el de crear un modelo y no el de implementar un sistema de software, sólo explicaremos la vista estática, la vista de casos de uso y la vista de interacción, las cuales utilizamos para mostrar la *aplicabilidad* de nuestro modelo.

Vista de casos de uso

La vista de casos de uso *modela la funcionalidad del sistema* como lo perciben los usuarios externos [19], también llamados actores. Un caso de uso es una unidad coherente de funcionalidad expresada como una transacción entre actores y el sistema. El propósito de la vista de casos de uso es listar los actores y los casos de uso y mostrar que actor participa en cada caso de uso.

Los casos de uso también pueden describirse a varios niveles de detalle. Pueden ser divididos y descritos en términos de otros casos de uso más simples. Un caso de uso es implementado como una colaboración en la vista de interacción.

Vista estática

Esta vista es estática porque no describe el comportamiento dependiente del tiempo del sistema [19]. Los elementos principales de la vista estática son las *clases y sus relaciones*: asociación, generalización y varios tipos de dependencia, tales como realización y uso. Una clase es la descripción de un concepto desde el dominio de la aplicación. Las clases son el centro alrededor del cual se organiza la vista de clases.

La vista estática se muestra en *diagramas de clase*. Las clases se dibujan en forma de rectángulo. La lista de atributos y operaciones se muestran en

Area	Vista	Diagrama	Conceptos Principales
Estructural	Vista estática	Diagrama de clases	Clase, asociación, generalización, dependencia, realización, interfaz
	Vista de casos de uso	Diagrama de casos de uso	Caso de uso, extensión, asociación, inclusión, actor, generalización de casos de uso
	Vista de implementación	Diagrama de componentes	Realización, interfaz, dependencia, componente
	Vista de despliegue	Diagrama de despliegue	Nodo, componente, dependencia, localización
Dinámica	Vista de máquina de estados	Diagrama de estados	Estado, evento, transición, acción
	Vista de actividad	Diagrama de actividad	Estado, actividad, transición de terminación, bifurcación, unión
	Vista de interacción	Diagrama de secuencia	Interacción, objeto, mensaje, activación
		Diagrama de colaboración	Colaboración, rol de colaboración, interacción, mensaje
Gestión del modelo	Vista de gestión del modelo	Diagrama de clases	Paquete, subsistema, modelo
Extensión a UML	Todas	Todos	Restricción, estereotipo, valores etiquetados

Cuadro A.1: Vistas y diagramas de UML

compartimientos separados. Los compartimientos pueden suprimirse cuando no sean necesarios los detalles completos. Una clase puede *aparecer en varios diagramas*. Las relaciones entre clases se dibujan como trayectorias que conectan los rectángulos de las clases.

Vista de interacción

La vista de interacción *describe el intercambio de mensajes entre roles* que implementan el comportamiento de un sistema [19]. Esta vista provee una vista integral del comportamiento de un sistema, es decir, muestra el flujo de control a través de muchos objetos. La vista de interacción es mostrada en dos diagramas que se enfocan en diferentes aspectos: *los diagramas de secuencias y los diagramas de colaboración*.

Diagrama de secuencia Un diagrama de secuencia *muestra un conjunto de mensajes ordenados en una secuencia de tiempo* [19]. Cada rol se muestra como una línea de vida, es decir, una línea vertical que representa el rol sobre el tiempo, a través de la interacción entera. Los mensajes son representados como flechas entre las líneas de vida.

Un uso de un diagrama de secuencia es *mostrar el comportamiento secuencial de un caso de uso*. Cuando el comportamiento es implementado, cada mensaje en un diagrama de secuencia corresponde a una operación en una clase, un evento disparador, o a una transición en una máquina de estados.

Diagrama de colaboración Una colaboración *modela los objetos y los enlaces significativos dentro de una interacción* [19]. Un diagrama de colaboración muestra los roles de una interacción en una disposición geométrica. Los mensajes se muestran como flechas, ligadas a las líneas de la relación, que conectan a los roles. La secuencia de mensajes, se indica con los números secuenciales que anteceden a las descripciones de los mensajes.

Un uso de un diagrama de colaboración es *mostrar la implementación de una operación*. La colaboración muestra los parámetros y las variables locales de la operación, así como las asociaciones más permanentes.

Bibliografía

- [1] Agent technology roadmap: Overview and consultation report. Technical report, AgentLink, December 2004.
- [2] K. M. Carley and L. Gasser. Computational organization theory. In G. Weiss, editor, *Multiagent systems: A modern Approach to Distributed Artificial Intelligence*, pages 299–330. MIT Press, 1999.
- [3] I. Chiavenato. *Introducción a la Teoría General de la Administración*, chapter Teoría Clásica de la Administración, pages 88–112. McGraw-Hill, 2000.
- [4] I. Chiavenato. *Introducción a la Teoría General de la Administración*, chapter Teoría Neoclásica de la Administración, pages 201–250. McGraw-Hill, 2000.
- [5] I. Chiavenato. *Introducción a la Teoría General de la Administración*, chapter Implicaciones de la Teoría de las Relaciones Humanas, pages 141–196. McGraw-Hill, 2000.
- [6] I. Chiavenato. *Introducción a la Teoría General de la Administración*, chapter Implicaciones del Enfoque Neoclásico: Tipos de Organizaciones, pages 251–286. McGraw-Hill, 2000.
- [7] S. A. DeLoach, M. F. Wood, and C. H. Sparkman. Multiagent systems engineering. *International Journal of Software Engineering and Knowledge Engineering*, 11(3):231–258, 2001.
- [8] J. Ferber and O. Gutknecht. A meta-model for the analysis and design of organizations in multi-agent systems. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98)*, pages 128–135, 1998.

- [9] M. S. Fox. An organizational view of distributed systems. *IEEE Transaction on Systems, Man, and Cybernetics*, SMC-11(1):70–80, January 1981.
- [10] E. Gama. *Bases para el Análisis de los Puestos*, chapter 4, pages 59–89. Manual Moderno, 1992.
- [11] M. Hannoun, O. Boissier, J. S. Sichman, and C. Sayettat. MOISE: An organizational model for multi-agent systems. In M. C. Monard and J. S. Sichman, editors, *Proceedings of the International Joint Conference IBERAMIA-SBIA*, pages 156–165. Springer, 2000.
- [12] F. López and A. Arenas. An architecture for autonomous normative agents. In *ENC '04: Proceedings of the Fifth Mexican International Conference in Computer Science (ENC'04)*, pages 96–103, Washington, DC, USA, 2004. IEEE Computer Society.
- [13] F. López and M. Luck. A model of normative multi-agent systems and dynamic relationships. In G. Lindemann, D. Moldt, and M. Paolucci, editors, *RASTA*, volume 2934 of *Lecture Notes in Computer Science*, pages 259–280. Springer-Verlag, 2002.
- [14] F. López and M. Luck. Modelling norms for autonomous agents. In E. Chavez, J. Favela, M. Mejia, and A. Oliart, editors, *ENC '03: Proceedings of the Fourth Mexican International Conference on Computer Science*, pages 238–245, Washington, DC, USA, 2003. IEEE Computer Society.
- [15] F. López, M. Luck, and M. d' Inverno. Constraining autonomy through norms. In *AAMAS '02: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 674–681, New York, NY, USA, 2002. ACM Press.
- [16] F. López, M. Luck, and M. d' Inverno. A normative framework for agent-based systems. In *NorMAS '05 : Proceedings of the Symposium on Normative Multiagent Systems*, pages 24–35. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour, 2005.
- [17] T. W. Malone. Modeling coordination in organizations and markets. *Management Science*, 33(10):1317–1332, 1987.

- [18] L. Montealegre and F. López. Representando estructuras organizacionales basadas en agentes. In M. Martín, M. Rubín, P. García, H. Jiménez, and M. León, editors, *Memorias del 2^o Congreso Nacional de Ciencias de la Computación*, pages 192–197. Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, 2004.
- [19] J. Rumbaugh, I. Jacobson, and G. Booch. *Unified Modeling Language Reference Manual*. Addison Wesley, 1998.
- [20] J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Software Development Process*. Addison Wesley, 1999.
- [21] M. Wooldridge. *An Introduction to Multiagent Systems*, chapter Intelligent Agents, pages 15–46. John Wiley & Sons, 2002.
- [22] F. Zambonelli, N. R. Jennings, and M. Wooldridge. Developing multi-agent systems: The gaia methodology. *ACM Transaction on Software Engineering Methodology*, 12(3):317–370, 2003.